Università degli Studi della Basilicata

Ph.D. Program of National Interest in Robotics and Intelligent Machines (DRIM)

Administrative Headquarters: Università di Genova

Distributed control strategies for cooperative and collaborative object transportation via Multi-Robot systems

by

Graziano Carriero

PhD Program in Industry 4.0

Thesis submitted for the degree of *Doctor of Philosophy* (38° cycle)

February 2026

Fabrizio Caccavale	Supervisor
Francesco Pierri	Supervisor
Antonio Sgorbissa	Head of the PhD program

Thesis Jury:

Francesco Pierri, *Università degli Studi della Basilicata*	Internal examiner
Yiannis Karayiannidis, *Lund University*	External examiner
Alessandro Marino, *Università degli studi di Cassino*	External examiner

Dipartimento di Ingegneria

Abstract

This research deals with the development of distributed control strategies for object transportation within multi-robot frameworks, encompassing both fully autonomous cooperative manipulators and collaborative human-robot interaction.

The first part of the work investigates the cooperative transportation problem, where a team of robots is tasked with manipulating a shared load relying exclusively on local peer-to-peer communication. To address the scalability and robustness limitations inherent in centralized architectures, a novel methodology for the distributed estimation of desired trajectories and exchanged forces is introduced. Firstly, an appropriate kinematic formulation is established to describe coordinated motion tasks in terms of absolute and relative variables. Within this framework, consensus-based estimators enable each agent to compute the estimates of the desired trajectories, which are subsequently exploited for real-time end-effector reference calculations. Secondly, distributed control schemes are developed to regulate the internal forces exerted on the object while limiting external wrenches arising from the interaction with the environment via an admittance control framework. The effectiveness of the overall architecture is validated through both high-fidelity simulations and experimental campaigns on homogeneous and heterogeneous manipulator setups, demonstrating a significant reduction in internal stress on the transported object and robust handling of external environmental contacts. The convergence of the estimators and the stability of the distributed control schemes are rigorously assessed through Lyapunov-based analyses.

The second part of the work investigates Human-Robot Collaboration (HRC) by integrating the operator as an agent-in-the-loop through an advanced Augmented Reality (AR) framework. Leveraging the developed AR features, a bidirectional communication flow is established to address dynamic role allocation and the exchange of intentions during shared transportation tasks. The proposed methodology enhances the operator's perception by providing real-time feedback on the forces exerted by the robot, while allowing for intuitive feed-forward trajectory planning. A key contribution

of this research is the implementation of a flexible autonomy scheme, enabling the operator to dynamically switch between a leader-follower configuration and a shared-trajectory mode. In the latter, a distributed admittance control strategy ensures the minimization of interaction forces and guarantees operational safety. Furthermore, the framework incorporates safety-critical monitoring of both human and robotic signals, reducing the physical workload for the operator. Experimental validations demonstrate that while AR-based interaction might initially increase cognitive load for novice users, a properly tuned level of robot autonomy ensures a seamless and effective collaborative experience.

Table of contents

List of figures **vii**

List of tables **xi**

1 Introduction **1**

1.1 Overview . . . 1
1.2 Cooperative multi-robot system . . . 2
1.2.1 Multi-agent systems control . . . 2
1.2.2 Object transportation tasks . . . 6
1.3 Human-Robot Collaboration . . . 9
1.3.1 Safety in Human-Robot Interaction . . . 10
1.3.2 Human-Robot Object Transportation . . . 13
1.4 Motivation and contribution . . . 17
1.5 Outline . . . 19

2 Distributed control of a cooperative multi-robot system **21**

2.1 Notation . . . 21
2.2 Problem statement and assumptions . . . 22
2.3 Task-oriented formulation . . . 23
2.4 Multi-agent system and communication topology . . . 27
2.5 Distributed estimation of reference trajectories in coordinated motion tasks . . . 28
2.5.1 Cooperative task space trajectories estimation . . . 29
2.5.2 Robot reference trajectories estimation . . . 32
2.5.3 Robot reference trajectories: an alternative formulation . . . 34
2.6 Distributed cooperative object transportation . . . 38
2.6.1 Object dynamics . . . 39

2.7 Distributed wrench estimation 40
2.8 Distributed compliance control 42
2.8.1 Stability analysis 44
2.9 Distributed internal admittance control 46
2.9.1 Stability analysis 48
2.10 Distributed external and internal admittance control 49
2.10.1 Stability analysis 51

3 Simulations and experiments 57
3.1 Coordinated motion task 57
3.1.1 Simulation in CoppeliaSim 57
3.1.2 Experiment with two real and two simulated Franka Emika Panda manipulators 63
3.1.3 Experiment with three real Franka Emika Panda manipulators . 68
3.2 Cooperative object transportation task 72
3.2.1 Simulation in CoppeliaSim with compliance control 72
3.2.2 Simulation in CoppeliaSim with internal admittance control . . 76
3.2.3 Experiment with internal admittance control 78
3.2.4 Experiment with external and internal admittance control . . . 84

4 Human-Robot Collaboration 97
4.1 Platforms for Human-Robot Interaction 97
4.2 HumanTIX: an Augmented Reality platform for Human-Robot Interaction 101
4.3 Development Tools and Experimental Setup 102
4.4 Design of the Augmented Reality Platform 103
4.5 Augmented Reality Features 106
4.5.1 Calibration 106
4.5.2 Robot State 109
4.5.3 Robot Workspace Visualization 111
4.5.4 Trajectory generation 112
4.5.5 Trajectory validation 112
4.5.6 Trajectory simulation and execution 113
4.6 Experiments 114
4.6.1 Task 115
4.6.2 Surveys 116
4.6.3 Analysis of the results 117

4.7 Safety Developments . . . 121
4.7.1 Eye and head tracking . . . 122
4.7.2 Virtual wall . . . 125
4.8 Experiments . . . 127
4.8.1 Eye and head tracking . . . 127
4.8.2 Virtual wall . . . 128
4.9 Human-Robot collaborative object transportation task . . . 132

5 Conclusions and future work 137
5.1 Future work . . . 139

References 141
Appendices . . . 154
A1 Proof of Theorem 2 . . . 154
A2 Proof of Theorem 3 . . . 154
A3 Proof of Corollary 1 . . . 155
A4 Proof of Theorem 4 . . . 156
A5 Proof of Corollary 2 . . . 156
A6 Proof of Lemma 2 . . . 157
A7 Proof of Theorem 5 . . . 159
A8 Proof of Theorem 6 . . . 161

List of figures

2.1 Robotic cell composed of N manipulators. 23
2.2 Block scheme of the overall architecture of the compliance controller. . 43
2.3 Elastic model of the contact between the i-th end-effector and the object. 44
2.4 Block scheme of the overall architecture of the internal admittance controller. 47
2.5 Block scheme of the overall control architecture for the k-th agent. The bank of estimators enables the distributed information flow among the robots ($k = 1, \ldots, N$). Blue arrows indicate the estimations of the object desired trajectory and the wrench collective vector performed by each agent. 50

3.1 Screenshot of the simulation setup in CoppeliaSim. 58
3.2 The proposed ROS system architecture: the information flow is represented by the arrows. The virtual leader node communicates with *Agent 2* node, i.e., the leader, whereas all other robots are the followers. 58
3.3 Followers' position and orientation estimation errors, $\widetilde{\boldsymbol{p}}_{a,h}$ and $\widetilde{\boldsymbol{\phi}}_{a,h}$ ($h = 0, ..., 3$), of the absolute trajectories. 61
3.4 Followers' position and orientation estimation errors, $\widetilde{\boldsymbol{p}}_{r,h}$ and $\widetilde{\boldsymbol{\phi}}_{r,h}$ ($h = 0, ..., 3$), of the relative trajectories. The superscript a is omitted for compactness. 61
3.5 Norm of robots end-effector tracking errors (position $||\widetilde{\boldsymbol{p}}_i||$, orientation $||\widetilde{\boldsymbol{\phi}}_i||$ and their time derivatives, $||\dot{\widetilde{\boldsymbol{p}}}_i||$ and $||\dot{\widetilde{\boldsymbol{\phi}}}_i||$), with $i = 1, \ldots, 4$. . . . 62
3.6 Absolute trajectory tracking error (position $\widetilde{\boldsymbol{p}}_a$, orientation $\widetilde{\boldsymbol{\phi}}_a$ and their time derivatives, $\dot{\widetilde{\boldsymbol{p}}}_a$ and $\dot{\widetilde{\boldsymbol{\phi}}}_a$). 62
3.7 Simulation setup with two physical robots and two robots simulated in CoppeliaSim. 63

3.8 The proposed ROS system architecture: *Agents 1* and *3* communicate with two real Franka Emika Panda robots. 64
3.9 Followers' position and orientation estimation errors, $\widetilde{\boldsymbol{p}}_{a,h}$ and $\widetilde{\boldsymbol{\phi}}_{a,h}$ $(h = 0, ..., 5)$, of the absolute trajectories, along with the relative position estimation errors, $\widetilde{\boldsymbol{p}}_{r,h}$, where the superscript a is omitted for compactness. 66
3.10 Tracking errors of robots end-effector (top) and absolute trajectory (bottom). 67
3.11 The experimental setup: the base frame of the robot mounted on the mobile base is set as the world frame $\boldsymbol{\Sigma}_w$. The Vicon motion capture system is exploited to retrieve the matrices $\boldsymbol{T}^w_{b,2}$ and $\boldsymbol{T}^w_{b,3}$, by performing a calibration procedure. 68
3.12 The communication graph established between the robots. 69
3.13 Tracking errors of robots end-effector (top) and absolute trajectory (bottom). 71
3.14 Screenshot of the simulation setup. The robotic team is cooperative grasping a wooden board. 72
3.15 Communication graph between the agents. 72
3.16 Force estimation errors ${}^i\widetilde{\boldsymbol{f}}_e$. 74
3.17 Internal force contribution for Robot 2 with and without the compliance controller activated (top); robots end-effector tracking errors with the compliance controller activated (bottom). 75
3.18 Object position and orientation tracking errors. 75
3.19 Comparison of the internal forces and torques with (left) and without (right) the compliance control. 77
3.20 Robots and object position and orientation tracking errors. 77
3.21 The experimental setup: the piece is grasped and transported by the robots. The Aruco marker lays on the working table and is employed for the setup calibration phase: its pose is known with reference to the world frame $\boldsymbol{\Sigma}_w$ is exploited for retrieving the UR5e robot base frame pose $\boldsymbol{T}^w_{ur}$. 79
3.22 The communication graph consists of 3 nodes, each synchronizing the signal of the wrench sensed from its force-torque sensor, and the wrenches estimation of its neighbors, using the ROS *TimeSynchronizer* filter to account for delays and packet losses. 80
3.23 Wrench estimation errors $\widetilde{\boldsymbol{h}}_k$. 82

3.24 Internal force contribution (on the left side) and torque contribution (on the right side) for Robot 2, in both the case studies: pure positional controller (on top), internal admittance controller (on bottom). 83
3.25 Object tracking errors in the experiment with the pure positional controller (blue) and the admittance controller (orange). 83
3.26 The experimental setup: the robots grasp and move the board. The base frame of the robot mounted on the mobile base is set as the world frame $\mathbf{\Sigma}_w$. A cylindrical obstacle is placed within the robot team's workspace, firmly fixed to a heavy support. The communication graph consists of 3 nodes arranged in a cycle graph, as in Fig. 3.12. 84
3.27 *Approximate Time* policy working method: each line that links a set of bullets across the multiple lines indicates a synchronized message. . . . 85
3.28 Wrench estimation errors $\widetilde{\boldsymbol{h}}_k$. 91
3.29 *Positional controller* case study: external wrenches (top) and internal wrenches (bottom). 92
3.30 *Positional controller* case study: position errors (left) and orientation errors (right). 92
3.31 *External admittance controller* case study: external wrenches (top) and internal wrenches (bottom). 93
3.32 *External admittance controller* case study: position errors (left) and orientation errors (right). 93
3.33 *Internal admittance controller* case study: external wrenches (top) and internal wrenches (bottom). 94
3.34 *Internal admittance controller* case study: position errors (left) and orientation errors (right). 94
3.35 *External and internal admittance controller* case study: external wrenches (top) and internal wrenches (bottom). 95
3.36 *External and internal admittance controller* case study: position errors (left) and orientation errors (right). 95

4.1 Bi-directional communication stream between Human and Robot established through HumanTIX platform. 101
4.2 System Architecture. 102
4.3 HumanTIX virtual panels for Human-Robot Interaction. 105
4.4 Calibration procedure. 108

4.5 Joint spheres for the evaluation of joint limit distance. 110
4.6 Feasibility check of the end-effector target pose. 111
4.7 Linear point-to-point trajectory generation in AR. 113
4.8 Trajectory modification in presence of an obstacle. 113
4.9 Setup of both the AR and the kinesthetic interface. 114
4.10 The AR and the kinesthetic interfaces. 115
4.11 NASA TLX, mental demand and physical demand for kinesthetic and AR interfaces; error bars represents 95% confidence intervals (CIs). . . 118
4.12 Mental demand for kinesthetic and AR interfaces, with comparison between different levels of AR expertise; error bars represents 95% CIs. 119
4.13 SUS scores for kinesthetic and AR interfaces; error bars represents 95% CIs. 119
4.14 Main statistically significant results of the SUS survey; error bars represents 95% CIs. 120
4.15 Unity simulations of the human gaze tracking in both cases of human focus on the object (left) or not (right). 123
4.16 The virtual wall in the AR environment. 126
4.17 Desired (dashed lines) and the actual (solid lines) position trajectories. The vertical black lines denote when the operator looks away from the robot. 129
4.18 Desired (red dashed line) and actual (blue line) end-effector path. . . . 129
4.19 Desired (red dashed line) and actual (blue line) end-effector path. . . . 130
4.20 Desired (red dashed line) and actual (red line) end-effector for the three components x, y, and z. 130
4.21 Desired (red dashed line) and actual (red line) x components of the end-effector velocity. 131
4.22 A screenshot of the video showing the collaborative object transportation, available at https://web.unibas.it/automatica/multimedia.html. 134
4.23 Robot trajectory (bottom) and corresponding filtered sensed wrench (top). 135

List of tables

3.1 Trajectory parameters. 59
3.2 Control and Trajectory parameters. 73

4.1 Hardware and Development Tools. 103
4.2 Average and Standard deviation error out of 25 calibration executions. . 109
4.3 Comparative analysis of the proposed platform with other approaches in the literature. 121
4.4 Control parameters for Safety features. 127

Chapter 1

Introduction

1.1 Overview

In recent years, robotic systems have been increasingly tasked with executing complex operations to support human activities, often exceeding the capabilities of a single robotic unit. Within this framework, cooperative manipulation has emerged as a key advancement, as it allows multiple robots to jointly accomplish tasks that would be infeasible for individual agents.

Early implementations of cooperative systems relied on centralized control strategies, where all robots had complete knowledge of the task and their corresponding actions, while a central unit computed the reference signals for each robot based on their states [1–4]. Although centralized control ensures high levels of coordination and efficiency, it also introduces critical drawbacks: the system becomes susceptible to single–point failures, demands considerable computational resources from the central unit and requires a complete redesign whenever the number of robots in the team changes [5–7].

For these reasons, over the past two decades, decentralized and distributed control strategies have become increasingly widespread. In cooperative multi-robot systems, a key challenge lies in designing control and/or estimation algorithms that rely solely on local communication among neighboring robots. In many cases, some information may be accessible only to a leader robot or to a subset of agents, while the remaining robots must estimate it in a decentralized manner [8–10].

Furthermore, one of the emerging paradigms of the Industry 5.0 revolution is Human–Robot Co-working, in which humans and robots collaborate whenever and wherever possible. In this context, information exchange plays a crucial role—not only among robots, but also among all agents forming this heterogeneous system.

Communication between humans and collaborative robots (cobots) can occur either as a unidirectional flow (from human to robot or vice versa) or as a bidirectional interaction, enabling mutual adaptation and shared understanding [11–13].

This thesis is focused on the latter scenario described, being aimed at:

- developing novel control approaches for cooperative robotic systems composed of multiple manipulators;
- designing and implementing collaboration frameworks between humans and multi-robot systems through bidirectional communication and interaction.

1.2 Cooperative multi-robot system

Cooperation among multiple robots becomes essential when addressing tasks that exceed the capabilities of a single agent. Examples include handling large or heavy payloads, assembly of multiple components or manipulation of flexible materials — all of which necessitate the adoption of multi-manipulator systems. Since the early 1970s, considerable research efforts have been devoted to the study of cooperative multi-robot systems, aiming to establish suitable formulations for coordinated motion tasks [14, 1], disassembly operations [3] and, more generally, the manipulation or transportation of a common rigid object [2, 15, 16].

In the *symmetric formulation* proposed in [17], relationships are established between the object's forces and velocities and those acting at the end-effector contact points. Conversely, the *task-oriented formulation* presented in [18] introduces the concept of a cooperative task space, in which the task variables are expressed in terms of absolute and relative motion quantities derived from the end-effector coordinate frames. In subsequent works, the proposed framework has been extended and generalized for cooperative systems composed of N manipulators [1, 4].

1.2.1 Multi-agent systems control

A common approach is to model all robots and the object within a centralized architecture [19]. For example, [20, 14] address dual-arm manipulation by controlling the relative pose of the arms alongside the absolute position of the object. In [1], a task-oriented motion planning approach is formulated for cooperative multi-robot systems. The framework allows the user to assign the motion task variables at the work-

piece level, while the manipulator trajectories are computed automatically exploiting kinematic transformations between the relevant coordinate frames.

Although they achieve high performance, centralized strategies introduce significant communication demands, reducing robustness due to an increased susceptibility to faults or attacks, since they require a central unit governing the multi-agent system. Limited communication or central unit failures can be mitigated through decentralized and distributed approaches, which improve scalability and robustness, while reducing the computational burden. In these cases, agents share limited information with nearby peers according to a given communication topology. Such coordination can improve overall system performance while maintaining a certain level of autonomy.

As shown by [21], delays in communication networks can severely affect the stability and passivity of interactive systems. Distributed approaches mitigate this by avoiding the centralized bottleneck and the network-induced delays typical of master-slave architectures, which are extensively analyzed in the survey by [22]. The estimation and control loops ensure that the system's performance remains independent of the total number of agents N. In this scenario, the limited number of leaders is a direct consequence of hardware heterogeneity and connectivity constraints in field robotics. In many real-world applications, such as underwater or underground exploration [23], only a small subset of agents has access to global positioning (e.g., GPS-denied environments) or high-fidelity sensing. A distributed estimation framework is therefore essential to propagate the reference signals from these few "informed" agents to the rest of the fleet, ensuring global formation coherence even when global connectivity is impossible [24].

While decentralized architectures do not require information exchange among neighbors, as they assign full autonomy to each agent, coming at a cost of reaching limited performance, both centralized and distributed strategies rely on inter-agent communication. Since many robotic applications require real-time control inputs- with sampling frequencies of 1 kHz or higher- the resulting communication bandwidth demand is considerable. Moreover, in the presence of a continuous communication, the burden on the communication infrastructure scales accordingly the number of agents. Although Model Predictive Control (MPC) strategies represent a powerful tool for explicitly handling system constraints and non-linearities [25], their application in fully distributed settings presents significant structural challenges, as it requires solving a non-convex optimization problem at each sampling instant for every agent, which can

be computationally prohibitive for the high-frequency control loops necessary to ensure stability during physical force interaction.

For this purpose, *consensus problems* have been deeply investigated for multi-agent systems agreement upon certain variables of interest. In the consensus literature, a widely common investigated class of problems concerns coordination among multi-vehicle systems [26, 27]. Typically, it is assumed that the equilibrium state corresponds to a weighted average of the initial information states (average consensus), which is thereby constant over time [28]. However, this assumption may not hold in scenarios where each vehicle's information state varies dynamically over time and/or in n-dimensional space. Indeed, this framework is not well suited for applications in which a specific desired or correct information state must be achieved. In such cases, the convergence analysis must address not only the agreement among agents to reach a common value, but also toward target value.

Additional issues may regard link failure and creation, i.e., switching network topology, time delays, disturbances in data acquisition or communication noise. Directed and undirected networks with fixed or switching topologies have been explored in [29], where two consensus protocols stability have been assessed with and without time delays. In particular, directed graphs which satisfy specific properties have been shown to play a crucial role in solving average-consensus problems.

As previously analyzed, adopting distributed rather than centralized control strategies improves system scalability and robustness. However, in scenarios characterized by continuous data exchange among neighboring agents, challenges such as network congestion and packet loss become increasingly critical. To mitigate these effects, event-driven strategies have been investigated and compared in [30]: control actions are triggered only when a specific state-dependent quantity exceeds a predefined threshold. The approach was initially analyzed in a centralized setting and subsequently extended to a distributed framework, where each agent requires information only from its immediate neighbors. Simulation results experienced a significant reduction of the number of updates with event-triggered control, whereas the self-triggered approach offers greater robustness by ensuring a guaranteed upper bound on the time interval between consecutive updates.

Similarly, [31] proposes an event-triggered communication strategy, aimed at reducing the information load on the network, while preserving the most important properties of output-strictly passive systems. In this scheme, information stream is enabled only when a state-dependent error exceeds a predefined threshold, thereby

updating the shared information as needed. Simulation and experiments have been carried out, showing a drastic drop in communication rate, as well as good performance for both the cooperative trajectory tracking and grasp maintenance tasks.

In [32], a distributed cooperative control strategy is proposed for multiple Underwater Vehicle–Manipulator Systems performing joint manipulation and transportation tasks under strict communication bandwidth limitations of the underwater acoustic communications. The method requires minimal data exchange—limited to tool-frame velocity information—while ensuring coordinated motion, safety and stability.

The work in [33] proposes a fully distributed cooperative framework for networked mobile manipulators. By combining a task-oriented strategy with formation-based cooperative control, the framework ensures robust performance across various manipulation tasks, operating independently of the object's center of mass. In the first stage, each manipulator uses an adaptation-based estimation law to approximate the linearized trajectory of a virtual leader, while physical constraints on load distribution and internal forces, defined through wrench computation, guarantee cooperative object manipulation. In the second stage, distributed adaptive controllers synchronize the end-effector motions of the manipulators, compensating for kinematic and dynamic uncertainties, enhancing cooperative task tracking.

In [8], a decentralized controller–observer framework is presented for time-varying centroid and formation control in multi-robot systems. Each robot estimates the collective state using only local information, enabling cooperative tracking of global objectives defined at the team level in terms of the centroid and geometric formation. The proposed approach ensures convergence of both estimation and tracking errors for directed and undirected communication graphs, as well as for switching topologies. The same observer structure is then employed to handle actuator saturation while maintaining convergence. Although local state estimation increases computational and communication demands compared to centralized architectures, these requirements remain compatible with low-cost hardware.

In [34], a bank of cascaded consensus-based estimators is employed to estimate a time-varying vector signal that is initially known only to a subset of agents, referred to as *sources*. The remaining agents, denoted as *users*, have no direct access to the signal and run the estimators to compute it, assuming that the signal's time derivative of a given order is either zero or bounded. Convergence is guaranteed even under switching communication topologies and in the presence of measurement disturbances

or communication noise. This result relies on the *sub-graph connectivity property* [34] of the communication network, which is a weaker condition than strong connectivity.

1.2.2 Object transportation tasks

In industry, service and other application domains, physically demanding problems are commonly addressed through the deployment of cooperative robotic systems. For instance, mobile robot teams can be employed to cope with object transportation tasks, by exploiting more flexible system configurations, while taking advantage of a larger actuating power.

In [35], multiple omnidirectional robots equipped with 3 active-casters are adopted for cooperative transportation tasks. The overall system is composed of two layers: the central unit and the holonomic mobile robots. Leveraging on the knowledge of the environment map and the presence of any obstacle, the central computer computes and provides the reference trajectories to each robot.

Two decentralized coordination strategies—namely, the BeRoSH and Constrain-Move approaches— are integrated in a multi-robot system to address object transportation tasks in [36]. The convenient implementation of each one follows the requirements of the specific task: for instance, the Constrain-Move strategy is preferred for path-oriented tasks. The implementation of these behavior-driven cooperative strategies is developed by structuring and coordinating the predefined behaviors of each robot to achieve collective task execution.

In leader–follower schemes, one robot defines the object's trajectory while the others track it using compliant control [37], relying solely on implicit communication [38]. For tasks such as balancing items on a tray, a Model Predictive Control strategy has been proposed [39], while [40] estimates the object's centroid from relative positions and applies distributed control to achieve cooperative rotation. In [41], a cooperative transportation framework for multiple non-holonomic mobile robots is developed, where the transported object acts as a virtual leader and the robots operate as followers in a leader–follower formation. Smooth path generation and asymptotically stable tracking control, relying on Lyapunov-based design, are implemented through inter-robot communication.

The use of neural networks or deep reinforcement learning can be an effective approach when prior knowledge of the system dynamics is unavailable [42]. The study in [43] introduces a decentralized control framework for multi-robot cooperative object transportation, where each robot employs a Deep Q-Network controller to learn

coordination strategies through trial-and-error interactions within the environment, eliminating central decision bottlenecks and ensuring robust cooperative behavior.

In [44], the authors propose a distributed strategy for estimating both the grasp matrix and the desired robot trajectories in rigid object manipulation tasks. The approach integrates two distributed observers for estimating the actual and desired end-effector trajectories, along with a distributed neural adaptive impedance control scheme to handle time-varying impedance behavior under model uncertainties. The agents communicate over a spanning-tree network, where the root node acts as the leader, and overall system passivity is guaranteed through the adoption of a virtual energy tank. Similarly, [45] introduces a single-master–multiple-slave architecture for teleoperation applications. In this framework, the slave robots' motions are decoupled into task-space and null-space components, where the latter is dedicated to maintaining formation. A round-robin communication protocol is employed for data transmission and synchronization between the master and the object is achieved via a radial basis function neural network.

A key challenge in cooperative object manipulation tasks is managing the wrenches exchanged between the manipulators and the object, distributing them to reduce individual robot effort while minimizing their impact on the object to preserve its integrity. This can be addressed through impedance/admittance control [46–49].

In [50], convergence of both object pose and interaction wrenches is ensured through a decentralized fuzzy control scheme that integrates disturbance observers with parameter adaptation. In [51], underwater robots with limited communication employ a leader–follower scheme for cooperative transportation, where the leader applies impedance control to follow the trajectory. Collaborative quadrotors in [52] achieve coordination without external position tracking or inter-robot communication using an adaptive controller. Similarly, various adaptive protocols have been proposed to enhance transportation robustness, ranging from quaternion-based schemes that guarantee transient performance [53], to force-sensor-based frameworks designed to overcome kinematic uncertainties in decentralized settings [54]. Gradient projection is adopted in [55] to minimize internal wrenches on the manipulated object. In [56], cooperative human–robot manipulation and transportation are addressed in semi-structured environments through a decentralized framework. Object and human trajectories are estimated with a tracking system and supplied to an impedance/admittance control module.

The approach presented in [49] has been later revised in [57] to incorporate the concept of geometrically consistent stiffness, accounting for both translational and rotational displacements. It addresses the control of dual-arm robots cooperatively manipulating a rigid object in a centralized manner, combining an object level controller, designed to provide compliant behavior, with an end-effector level controller, aimed at limiting internal loads on the object. The overall control scheme is thereby arranged in a two-loop strategy:

- the outer loop, i.e. the *external impedance*, exploits the external wrenches calculation to distribute them for conferring a compliant behavior to the manipulated object;
- the inner loop locally minimizes the internal wrenches acting on the object, by recurring to an *internal impedance*.

The scheme flexibility lies in the decoupling of the two control loops, allowing individual activation/deactivation of external and internal impedance.

This method has been further extended in [15, 4] to teams of aerial robots. Specifically, in the latter, the geometry and dynamic parameters of the object are assumed unknown. Thus, in a first step, the robots cooperatively estimate the object parameters; then, the latter are used in an impedance control framework that aims to minimize both the contact wrenches arising from the object–environment interaction and the internal wrenches generated by the robot–object interaction.

In [58], a two-layer distributed architecture is proposed for the cooperative control of Euler–Lagrange systems. At the lower layer, each robot employs an observer–controller scheme to estimate the complete system state, which local controllers then exploit to compute the inputs for mobile manipulators; in particular, the upper level employs an adaptive control law capable of handling system uncertainties. The proposed method applies to both motion coordination and multi-robot object manipulation, where it controls object movement while minimizing internal stresses by decomposing each robot's wrench into external and internal components. A subsequent work, [59], introduces a distributed algorithm for cooperative manipulation of an unknown object rigidly grasped by multiple mobile manipulators, whose number is not pre-shared. The approach consists of two stages: cooperative estimation of the object's parameters, followed by a control framework that regulates the object's pose using the estimated data. The method is adaptable to various scenarios, such as enforcing impedance behavior or leader–follower zero-force control.

Any of these approaches focuses on a model-agnostic distributed estimation of the internal and external wrenches, adopting the Moore-Penrose pseudoinverse matrix of the grasp matrix, which takes into account the geometrical properties of the multi-robot system. Particular dynamic-based parameterizations of the generalized inverse of the grasp matrix [60], also incorporating the mass matrix [61], induce an implicit and "natural" load-sharing, even though requiring the knowledge of the dynamic model of the object and all manipulators.

To the best of my knowledge, existing decentralized or distributed control strategies for cooperative multi-robot transportation are not capable of simultaneously estimating the desired object trajectory and computing the internal and external generalized forces acting on the object. The problem of transporting a rigid object along a trajectory known only to a subset of the agents, while minimizing the forces exerted on the object, even under interactions with the external environment, remains an open research challenge.

1.3 Human-Robot Collaboration

Robots, sensors, devices and human resources are intelligent entities that can be integrated in a smart factory according to the paradigms of Industry 4.0. Smart factories employ their cooperation and real-time data sharing to reach high levels of flexibility and reconfigurability in order to dynamically adapt to the current markets changes.

The aim to improve flexibility of logistic and production systems is pursued not only with technological resources, e.g. smart devices and machines, but also with a new role of human operators, as the modern concept of the so-called *Operator 4.0* [62]. In smart factories, various augmentations of the human capabilities are adopted to empower the "smart operators" with new skills and gadgets to fully capitalize on the opportunities being created by recent technologies. For this reason, the increasingly widespread introduction of collaborative robots, called *cobots*, in industry provides an improvement of smart operator productivity, besides that relieving her/him from non-ergonomic and repetitive tasks: cobots and humans can interact establishing an information stream, e.g. by means of multi modal interfaces [62]. A comprehensive overview of the platforms enhancing Human-Robot Interaction (HRI) is detailed in Section 4.1.

1.3.1 Safety in Human-Robot Interaction

The European Commission defines Industry 5.0 as a complement to Industry 4.0: it represents a cultural revolution that relocates industry to the contemporary world, driving towards a sustainable, human-centric and resilient industry. Indeed, the current transition to Industry 5.0 builds upon the foundations of Industry 4.0. Research and innovation spearhead a shift towards an European industry that prioritizes humanity, sustainability and resilience. At the core of Industry 5.0 lies a human-centric philosophy that places paramount importance on addressing human needs and interests in the production process. This allows to move away from a solely technology-driven agenda to one that embraces both human and societal values [63].

Indeed, Industry 5.0 recognizes "the power of industry to achieve societal goals beyond jobs and growth, to become a resilient provider of prosperity, by making production respect the boundaries of our planet and placing the well-being of the industry worker at the center of the production process" [64]. The previous system, focused on quality improvement and cost reduction, gives way to a new system that places workers at the center of the manufacturing process, shifting from a technology-driven progress approach to a human-centric one.

One of the emergent visions of Industry 5.0 is *Human-Robot Co-working*. According to this, humans and robots will collaborate whenever and wherever possible. Some workers will be enthusiastic about collaborating with robots, while others will be firmly against this idea. As human-robot co-working is becoming a reality, but it is not yet, the real issue to tackle is to identify which jobs may be handled by robots. For instance, a good practice could be that humans will focus on tasks that require creativity and a superior cognitive capability, while robots will face the rest [11].

According to this paradigm, the adoption of innovative technologies, such as collaborative robots, sensors, HMD, is widespread in logistics and production systems for improving their flexibility. In this scenario, the human-machine understanding crucially matches the human-centric Human-Robot Collaboration (HRC): in the context of a manufacturing task, the purpose is making the cobots able to understand the short-term human intent [65, 66]. Hence, human-robot communication should be effective, since the human may provide real time information, namely what to do, when to do what, as well as the final objective of a manufacturing job, through instructions (gesture, gaze) and actions (motion prediction). Human-robot communication should also be dynamic and bi-directional, exploiting direct, such as gesture or indirect feedback,

for instance through gaze. Altogether the human-centric HRC is the cobot dynamic purpose of recognition of human states, aimed at proactively adapting to contingencies the task schedule.

In this context, productivity can be increased, without renouncing to relieve human operators of non-ergonomic and repetitive tasks, in order to ensure a safer and healthier work environment. In the last decades, much effort has been devoted to enforcing safety in industrial Human-Robot Interaction (HRI) applications. In [67], the problem of safety evaluation in human-robot collaborative manufacturing has been combined to production constraints: a methodology for generating velocity trajectories has been implemented, which simultaneously takes into account the maximization of the robot productivity, addressing the issue of human safety. The base concept is monitoring the real-time human-robot distance in order to modulate a suitable velocity scaling factor.

The same paradigm is suggested in [68], according to speed and separation monitoring, which is one of the methods monitoring the human-robot collaborative operations, where a robot system and humans share the same work-space. This ISO standard was published in 2016, in order to supplement the tighter [69] (recently replaced by [70]), conversely released in 2011, which was based on stringent constraints to regulate interaction between human and industrial robots. On the contrary, [68] meets the requirements of the previous standards and provides additional guidance, introducing four collaborative methods to be included – at least one of them – in the Human-Robot operations:

- safety-rated monitored stop,
- hand guiding,
- speed and separation monitoring,
- power and force limiting.

The first method consists in verifying the presence of the human operator in the collaborative workspace: if so, the robot has to cease its motion and then resume only after the operator exits; otherwise it can operate not-collaboratively and the operator is not allowed to enter inside the collaborative workspace. The safety-rated monitored stop method is applied also before the operator is permitted to enter in the collaborative workspace to transmit motion commands to the robot by a hand-operated device. This happens in the case of hand guiding mode. An alternative approach is to guarantee a proper protective separation distance between the robot and the human

operator, that are allowed to move inside the shared workspace. Indeed, according to speed and separation monitoring, the robot speed is directly proportional to its distance to the human operator: if this distance becomes smaller, robot speed should decrease.

None of the proposed methods allows a dynamic contact between the robot and the operator. Physical contact can occur intentionally or unintentionally only in power and force limiting operation method. To achieve risk reduction, the hazards associated to the robot are kept below the threshold values determined during the risk assessment. Active safety design measures can be adopted to address any type of contact. The ISO standard indicates some possible risk reduction measures, although admitting there could be many others and this is just one of the weakest points of ISO standards: there are several ambiguous issues remaining. No modern standards have been released since 2017 and no updates or integration have been discussed. Thus, for instance, there is no reference for human-robot interaction through Extended Reality (XR) systems, which are widespread nowadays. Indeed, XR can help to increase the productivity of logistics and production systems and relieve operators from non-ergonomic and repetitive tasks, making their work environment safer and healthier [71]. This lack of updating is merged to an increasing need for a new study that tracks which new human features or robot parameters should be detected for enhanced safety, thanks or due to the adoption of these technologies.

In [72], a new HRI taxonomy improving the safety recommendations of ISO standards has been developed. It classifies the HRI into four levels based on the interaction level and the required task. If each event is suitably detected, faulty events are avoided and human safety is enhanced. For instance, a physical Human-Robot interaction is dangerous if the robot system is not expecting it. Furthermore a new advanced safety function is proposed to guarantee safety: the robot recognizes human readiness for starting a collaborative task thanks to face detection algorithms.

Nowadays, advances in HRC are based on the creation of user-friendly interfaces that allow operators to effortlessly interact with cobots, regardless of their level of expertise. According to [73], visual augmentation can provide five different benefits regarding the HRI: facilitate programming, support real-time control, improve safety, communicate intent, increase expressiveness. For instance, it has been shown that the integration of XR technologies significantly improves human performance: in [74], compared to a kinesthetic teaching approach, the proposed Augmented Reality platform better supports users in robot programming tasks.

A human-centric integrated Mixed Reality (MR) system for safety-aware HRC is implemented in [75], in order to make safer and more effective the human operator's task execution, exploiting the XR features of Hololens 2. The system is deployed at three layers: cyber-physical, for the robot hologram calibration and for guaranteeing the minimum safety distance during the HRI; MR layer, for the robot programming through hand gesture; deep learning-based segmentation and 3D geometry, for seamless mapping between the real robot and its digital twin, minimum distance between human and robot evaluation and safety-related monitoring. Another example of safety increasing by exploiting AR is [76], where a projector-mirror and wearable HMD (Hololens 2) are used for building up a system for safe HRC. Although requiring still more technical maturity in order to be considered suitable for industrial environments, Hololens shows to be eligible to enhance safety.

The adoption of XR devices, e.g., Hololens HMD, broadens the panorama of possible advanced safety features and allows the tracking of some new human parameters. For instance, the introduction of a virtual wall that separates the human's and the robot's workspaces could guarantee a safe execution of the operator's task. This new safety feature can be easily and instantly implementable through the assistance of XR devices [77]. The core idea is defining this virtual entity that represents the splitting boundary of these two spaces: the robot can never penetrate it, hence the human can execute all her/his operations in absolute safety. Considering the safety-rated monitored stop method, no human function is explicated except for her/his presence in the shared workspace. Furthermore, no reference is provided about the operator's concentration and her/his monitoring of the robot task execution in ISO standards. Could there be a correlation between it and the robot speed during its task execution?

1.3.2 Human-Robot Object Transportation

In the field of physical Human–Robot Interaction (pHRI), safety, dependability and performance must be integrated within a unified framework of optimality. Such an approach aims to guide future advancements in the design of robotic systems that can operate safely and reliably in human-centered environments, in line with the vision of Industry 5.0 [78, 79]. Considerable research efforts have been devoted to the detection of unexpected collisions [80, 81] and their distinction from intentional contacts, either through frequency-domain analyses [82] or using neural-network-based methods [83].

This research direction is motivated by the growing deployment of robots for object transportation tasks in industrial contexts, as well as in other domains such as service

robotics and teleoperation. Such collaborative schemes enable the distribution of interaction forces among multiple agents, reduce the physical workload on humans and facilitate the manipulation of large or heavy objects while maintaining safe operating conditions.

This synergy leverages the complementary strengths of both agents — the robot's physical power generation capability and the human's cognitive abilities. For instance, humans' proactive capabilities can be exploited to oversee and optimize the physical power of robots when facing diverse task demands and unexpected disturbances. Consequently, achieving seamless integration between humans and robots represents a challenge of paramount importance in human–robot collaborative object transportation.

As detailed in [84], Human-Robot Collaboration interfaces establish a bi-directional information flow, which can be primarily distinguished into feed-forward and feedback modalities. Feed-forward interfaces manage the communication stream directed *to* the robot, originating from the human operator or the external environment. A prominent example is provided by visual interfaces, which assist the user in the intuitive generation and validation of trajectories. Moreover, in contexts involving physical Human-Robot Collaboration, haptic devices assume a crucial role by integrating the *human in the loop*. They facilitate a bi-directional signal stream, dealing with the force exchange, which is fundamental for tasks such as the co-transportation of objects. Conversely, feedback interfaces operate in the opposite direction- *from* robot to human- aiming to augment the operator's perception of the task and its environment. Mixed Reality devices, for instance, are particularly well-suited for this purpose, enhancing human awareness through the overlaying of contextual information supplied by the robot [85].

In both the scenarios, the effective integration of control inputs from both the human operator and the robot is essential for coordinating the multi-agent system synergy. According to [86], denoting as u_h and u_r the inputs originated by the human and the robot, respectively, and as μ the robot understanding of the incoming inputs, a critical aspect in this framework is the definition of the function $y(u_h, u_r)$ which modulates these inputs. For instance, an example of this function could be provided by a linear combination of the inputs, as:

$$y(u_h, u_r) = \alpha u_r + (1 - \alpha) u_h, \tag{1.1}$$

where the scalar $\alpha \in [0, 1]$ sets the level of autonomy within the overall system, thereby ranging from a Shared Autonomy (SA) to a Shared Control (SC). More in detail, in a

SA system α weights the inputs based on μ. The robotic agents adapt their behavior according to the inputs, including the human commands and external environment data, which are collected in μ. In turn, in a SC system, this dependence is removed and the human operator retains direct and continuous management of the input modulation by manually tuning the autonomy level over time.

The dynamic allocation of roles between human and robot agents is explored in [87] through the introduction of three distinct schemes for task effort distribution. A central concern in this framework is the robotic system's capacity to autonomously and appropriately adapt its own role during collaboration, a necessity dictated by task dynamics, environmental uncertainty and unpredicted variations in human intent.

In [88] a composite learning–based variable impedance control strategy is introduced that enables accurate online modeling of robot dynamics under interval excitation, requiring only limited data over time. A multi-mode adaptive control framework, encompassing robot-dominant, human-dominant and cooperative modes, ensures smooth transitions, stable interaction and precise trajectory tracking through adaptive stiffness and damping adjustment. Similarly, the work in [89] proposes a variable admittance control approach for the Human-Robot object manipulation. In this scheme, the admittance parameters are dynamically adapted based on the desired trajectory and the external forces sensed by the robot, enabling the system to effectively respond to potential changes of the manipulated object. This approach further incorporates robust inverse dynamics control to stabilize the motion control laws, ensuring accurate trajectory tracking even in the presence of uncertainties. A variable impedance control is implemented in [88] for HR collaborative transportation tasks, where three control modes are defined: robot-dominant, human-dominant and collaboration mode. A learning mechanism for parameter estimation is employed for enhancing safety, by actively adapting stiffness and damping throughout the human-robot interaction.

In [90] a reference model based on Dynamic Movement Primitives along with an Extended Kalman Filter observer is proposed to predict the target pose and time scaling, enabling proactive assistance during collaborative object transportation. The approach assumes that the weight of the object is known beforehand and accounts only for the forces exerted by the human partner. The stability analysis demonstrates that the proposed framework ensures bounded behavior of both the reference model and the observer.

A further application concerning the exchange of intentions during human-robot object transportation is explored in [91], where communicative signals are directly

encoded within actions that modify the object's state. This approach employs a Model Predictive Control framework to map the combined human-robot actions into cooperative strategies. The primary objectives of this control mechanism are the minimization of uncertainties and the simultaneous maximization of task execution efficiency.

The integration of humans into multi-agent systems represents a crucial issue in human–robot co-transportation tasks. Human can be fully integrated in the control loop solely by establishing a bi-directional exchange of intentions among the different entities involved. The development of novel XR platform can provide useful features, but dynamic role allocation, seamless data exchange and cognitive overload still represent relevant issues.

Another critical issue arises from the translation–rotation ambiguity resulting from the coupling of multiple degrees of freedom in the object's dynamics. The study in [92] addresses the difficulties encountered by human operators when applying large torques during collaborative object handling with a robot. By imposing a virtual nonholonomic constraint on the object, the robot's motion is effectively restricted, thereby preventing object's sideslip and enhancing the stability of the cooperative manipulation.

To address this challenge, further efforts have focused on the development of passive robotic systems, whose mechanical design specifically aims to mitigate the problem, allowing for efficient transmission of wrenches and smooth trajectory tracking. In [93], a system consisting of multiple mobile collaborative robots has been developed to enhance payload manipulability. In this case, the human operator exerts direct control over the trajectory and forces, acting as the primary guiding force to manipulate the object. In fact, these mobile robots are equipped with omnidirectional wheels and a 3-DOF Delta robot mounted on top. This configuration allows for the decoupling of object maneuverability from the robot's motion, with its passive compliance being inherent in its mechanical structure. Passive robotics is also explored in [94] and [95]. In the former, a passive robotic porter, comprising three omnidirectional wheels equipped with servo brakes and a control unit, is developed to assist humans in object co-manipulation. The additional passive joint is exploited to adjust the object's orientation, while its position is regulated through the servo brakes. In the latter, a set of distributed small mobile robots is employed for cooperative object transportation in collaboration with human operators.

In [96], the concept of employing an increasing number of small mobile robots, guided by a human operator, to perform object transportation tasks is taken to its

extreme. A swarm of cooperative mobile robots is commanded to push a common object toward a target position, specifically along the section that occludes the goal. No inter-robot communication is required, as their motion is governed by a fully decentralized scheme. When an obstacle is encountered, a designated goal robot is teleoperated by a human operator to manage the uncertainty introduced by the obstruction.

Recent developments have been focused on the manipulation of object with different rigidity and deformability. In such cases, the main challenges concern the transmission of wrenches and the estimation of the material's deformation state, which strongly depend on the physical properties of the manipulated object. In [97], information from a haptic device is combined with human motion data captured through a motion tracking system, enabling an accurate and intuitive co-transportation framework. In [98], a Convolutional Neural Network is employed to model and estimate the deformation state of the material based on depth images. The actual and desired deformation states are then compared to compute twist commands, which are subsequently fed to the robot controller.

In [99] and [100], cable-suspended manipulation is investigated in collaboration with a robotic manipulator and multiple aerial robots, respectively. In the former, cables are attached to a board that serves as a support for the transported objects, resulting in a partially deformable structure that enhances both safety and flexibility in human–robot interaction. The sensed wrenches are exploited to detect load variations or unintended collisions with the external environment. In the latter, the payload is suspended by cables and manipulated by a team of aerial robots. A six-DOF admittance controller is implemented and the system's redundancy is leveraged to accomplish secondary objectives such as load distribution, maximizing the separation distance between robots and humans and minimizing the estimated human wrench exerted on the object.

1.4 Motivation and contribution

In the field of cooperative robotics, the goal is to investigate new control strategies for multi-robot systems capable of performing collaborative transportation tasks of rigid objects in a distributed manner. The envisaged strategy addresses the estimation of the desired object trajectory, which is known only by a subset of the agents, and concurrently the estimation and mitigation of both internal and external wrenches acting on the object.

Firstly, an appropriate kinematic formulation is established to fully describe coordinated motion tasks of a multi-robot system, in terms of absolute and relative motion variables. In this context, consensus-based estimators enable each agent to compute an estimate of the desired absolute and relative trajectories in a distributed fashion which are subsequently exploited for the reference end-effector trajectory calculations (see Section 2.5). Secondly, control schemes for the cooperative manipulation and transportation of rigid objects are developed to regulate the internal forces exerted on the object, while limiting external forces arising from the interaction with the environment, and presented in Sections 2.8, 2.9 and 2.10. Unlike the work in [57], which is based on a centralized approach, these strategies also require the implementation of consensus-based estimators to distribute the calculation of the wrench sensed at each robot end-effector.

The convergence of the employed estimation algorithms, as well as the stability of the implemented distributed control schemes, is analyzed and assessed through Lyapunov-based approaches. Finally, the effectiveness of the overall control strategies has been evaluated in Chapter 3: initially, through simulations of multi-robot setups in robotic simulation environments; subsequently, through experimental campaigns on heterogeneous or homogeneous manipulator configurations aimed to reduce internal stress acting on the transported object and to limit the forces generated by the contact with the external environment.

As previously discussed, Augmented Reality applications represent effective interfaces for Human–Robot Interaction in collaborative transportation contexts. They provide a bi-directional information flow that enables full integration of the human operator into the control loop.

The development of human–robot platforms for collaborative transportation is part of an advanced project aimed at creating a general-purpose augmented reality framework on the HoloLens 2, entitled HumanTIX (see Section 4.2). The platform has been deeply enriched and developed so that the system is now capable of addressing the integration of the human operator as an agent-in-the-loop, exploring dynamic role allocation and intentions exchange issues, previously investigated: it has been designed to ensure safe interaction between robots and low-skilled operators during tasks, such as robot trajectory planning, pick-and-places and object transportation. As evaluated in the experimental test presented in Section 4.6.3, this approach reduces the physical workload associated with robot interaction, at the cost of a slight increase in cognitive load for novice users. The main aspect requiring attention concerns the potential

information overload during the user's first interactions. However, an appropriately configured level of robot autonomy can ensure a simple and effective user experience. Moreover, the platform implements novel features that address the need to incorporate safety guidelines by introducing the tracking of parameters and signals from both the human and the robot, with the aim of strengthening bidirectional communication (see Section 4.8).

The proposed application incorporates both feed-forward and feedback interaction modes. On one hand, it allows for the generation of an object-level trajectory; on the other, it enhances the operator's perception by providing information about the forces exerted by the robot on the object. In this heterogeneous system, composed of a human, a robot and an AR device, the operator can dynamically adjust the robot's level of autonomy, switching from a simple leader–follower scheme to a more advanced configuration in which the object trajectory is shared with the manipulator. In the latter case, the implementation of an admittance controller ensures the reduction of interaction forces and guarantees a safe collaboration (see Section 4.6). Additional AR functionalities—such as controlling the opening and closing of the robot's end-effector—are provided through an intuitive virtual control panel. Finally, thanks to its modular design, the application naturally supports the development and integration of new features that exploit the full capabilities of the augmented reality device (e.g., human gaze or hand movements tracking).

1.5 Outline

The remainder of this thesis is structured as follows. Chapter 2 presents the theoretical analysis of the task-oriented formulation for multi-robot systems, together with the distributed estimation of reference trajectories for coordinated motion, the distributed estimation of the wrenches exerted on the object and the distributed control strategies developed for cooperative object transportation. The stability and convergence analysis of the proposed schemes is reported in Appendices.

Simulation and experimental results evaluating the effectiveness of the proposed strategies are described in Chapter 3. Chapter 4 tackles the design of the Mixed Reality platform developed for Human–Robot Collaboration, the usability test conducted to assess its features and the collaborative modes explored for Human–Robot transportation tasks. Finally, the conclusions and directions for future work are discussed in Chapter 5.

Chapter 2

Distributed control of a cooperative multi-robot system

2.1 Notation

Consider a multi-robot system composed of N manipulators. Let $\boldsymbol{\Sigma}_w$ be an inertial reference frame and let $\boldsymbol{\Sigma}_{b_i}$ be the i-th robot base frame, with $i = 0, \ldots, N$. Hereafter, the notation $\boldsymbol{R}_{b_j}^{b_k}$ will denote the (3×3) rotation matrix expressing the orientation of the reference frame $\boldsymbol{\Sigma}_{b_j}$ with respect to frame $\boldsymbol{\Sigma}_{b_k}$. The Euler angles vector $\boldsymbol{\phi}_{j,k}$ describes the rotation expressed by $\boldsymbol{R}_{b_j}^{b_k}$, and is equivalent to the notation $\boldsymbol{\phi}(\boldsymbol{R}_{b_j}^{b_k})$, which indicates the Euler angles vector extracted from the rotation matrix $\boldsymbol{R}_{b_j}^{b_k}$ [101]. With regard to the geometric quantities expressed with respect to $\boldsymbol{\Sigma}_w$, the superscript w will henceforth be omitted for the sake of compactness. The pose, generalized velocity, generalized acceleration of the i-th robot end-effector reference frame $\boldsymbol{\Sigma}_i$ are expressed with reference to $\boldsymbol{\Sigma}_w$, as $\boldsymbol{x}_i = \left[\boldsymbol{p}_i^{\mathrm{T}}, \boldsymbol{\phi}_i^{\mathrm{T}}\right]^{\mathrm{T}}$, $\dot{\boldsymbol{x}}_i$, $\ddot{\boldsymbol{x}}_i$, respectively, with $\boldsymbol{\phi}_i$ denoting a suitable Euler-angle triplet.

As regards the kinematics of the single manipulator, its end-effector pose is linked to the joint position variables via

$$\boldsymbol{x}_i = \boldsymbol{k}_i(\boldsymbol{q}_i), \tag{2.1}$$

where $\boldsymbol{k}_i(\cdot)$ denotes the direct kinematic equation of the i-th robot and $\boldsymbol{q}_i$ is the $(m \times 1)$ vector defining the joint variables of the m degrees of freedom (DOFs) manipulator. In

turn, the end-effector velocity is retrieved as

$$\dot{\boldsymbol{x}}_i = \boldsymbol{J}_i(\boldsymbol{q}_i)\dot{\boldsymbol{q}}_i, \tag{2.2}$$

where $\boldsymbol{J}_i$ is the $(n \times m)$ matrix denoting the robot analytic Jacobian, and n represents the manipulator's DOFs in the operational space ($n = 6$ in the three-dimensional space). In general, $m \geq n$, and $m > n$, in the case of a redundant robot.

2.2 Problem statement and assumptions

The multi-robot system is in charge of executing a cooperative object transportation task. Since the task has to be performed in a distributed manner, hereafter the term *agent* will denote a robotic manipulator with its control unit, responsible for managing communication flows, planning the end-effector trajectories and computing the control action.

In this scenario, let us introduce a set of assumptions to be made for the purpose of this experiment:

Assumption 1. *The multi-robot system comprises all fixed-based robots. Without any loss of generality, if one robot is mounted on a mobile base, this latter is assumed to be mechanically braked.*

Assumption 2. *Robots do not know anything about the experimental setup a priori, but the relative pose of their base frames, $\boldsymbol{\Sigma}_{b_i}$, expressed with respect to the world reference frame, $\boldsymbol{\Sigma}_w$.*

Assumption 3. *The object cooperatively grasped by the multi-robot system is rigid and its geometry and weight are known by each agent.*

Assumption 4. *Only a subset of the agents knows a priori the trajectory assigned to the object. Robots can communicate each other according to a given topology, as an all-to-all communication can not be established.*

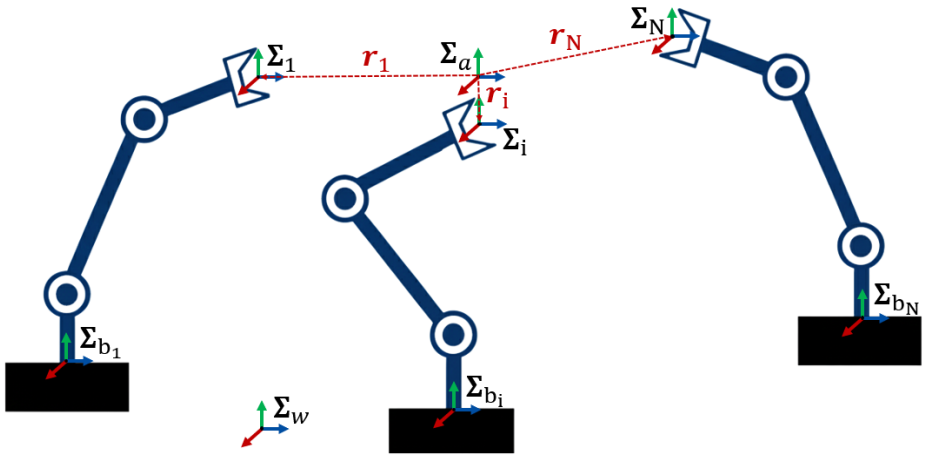


Figure 2.1 Robotic cell composed of N manipulators.

2.3 Task-oriented formulation

In order to devise a suitable formulation to describe the cooperative task, by resorting to the task-oriented formulation proposed in [102] and then extended for the case of N cooperative manipulators in [9], [103], a set of geometrically meaningful variables is designed with respect to the *cooperative task space*. This formulation aims at combining the trajectories of the robots' end-effector reference frames with the absolute and relative motion variables of the cooperative system.

Let $\boldsymbol{x}$ be the collective vector of the robots' end-effector poses $\boldsymbol{x} = \left[\boldsymbol{x}_1^{\mathrm{T}}, \ldots, \boldsymbol{x}_N^{\mathrm{T}}\right]^{\mathrm{T}}$. The absolute motion is described by the *absolute reference frame*, $\boldsymbol{\Sigma}_a$, whose pose can be defined as the barycenter of the end-effectors poses, as shown in Figure 2.1:

$$\boldsymbol{x}_a = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i = \boldsymbol{J}_a\boldsymbol{x}, \qquad \dot{\boldsymbol{x}}_a = \boldsymbol{J}_a\dot{\boldsymbol{x}}\,, \tag{2.3}$$

where $\boldsymbol{J}_a$ is the absolute Jacobian, given by

$$\boldsymbol{J}_a = \frac{1}{N}\left(\mathbf{1}_N^{\mathrm{T}} \otimes \boldsymbol{I}_n\right) \in \mathcal{R}^{n\times Nn}, \tag{2.4}$$

$\mathbf{1}_N$ is the $(N\times 1)$ vector of ones, $\otimes$ denotes the Kronecker product [104] and $\boldsymbol{I}_n$ the $(n\times n)$ identity matrix. The absolute Jacobian can be partitioned as $\boldsymbol{J}_a = \left[\boldsymbol{J}_{ap}^{\mathrm{T}}, \quad \boldsymbol{J}_{ao}^{\mathrm{T}}\right]^{\mathrm{T}}$, where $\boldsymbol{J}_{ap}$ and $\boldsymbol{J}_{ao}$ are the positional and orientational parts, respectively.

In this way, the position of the origin of $\boldsymbol{\Sigma}_a$ coincides with the centroid of the end-effectors of the multi-robot system, and similar considerations apply to the orientation, whose definition exploits the linearity of the Euler-angle representation.

Since the overall DOFs of a multi-robot system are $6N$, further kinematic relations must be considered in order to comprehensively describe its motion in the cooperative task space. Hence, define the relative position and orientation motion variables as:

$$\boldsymbol{x}_{i,r} = \boldsymbol{x}_{i+1} - \boldsymbol{x}_i, \quad i = 1, \ldots, N. \tag{2.5}$$

By defining the collective relative pose, $\boldsymbol{x}_r$, with simple algebraic operations, the relative task Jacobian can be introduced as follows

$$\boldsymbol{x}_r = \begin{bmatrix} \boldsymbol{x}_2 - \boldsymbol{x}_1 \\ \boldsymbol{x}_3 - \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_N - \boldsymbol{x}_{N-1} \end{bmatrix} = \boldsymbol{J}_r \boldsymbol{x}, \tag{2.6}$$

where $\boldsymbol{x}_r = [\boldsymbol{x}_{r_1}^{\mathrm{T}}, \ldots, \boldsymbol{x}_{r_{N-1}}^{\mathrm{T}}]^{\mathrm{T}} \in \mathcal{R}^{(N-1)n \times 1}$ and

$$\boldsymbol{J}_r = \begin{bmatrix} -\boldsymbol{I}_n & \boldsymbol{I}_n & \boldsymbol{O}_n & \ldots & \boldsymbol{O}_n \\ \boldsymbol{O}_n & -\boldsymbol{I}_n & \boldsymbol{I}_n & \ldots & \boldsymbol{O}_n \\ \vdots & \ddots & \ddots & \ddots & \\ \boldsymbol{O}_n & \ldots & \boldsymbol{O}_n & -\boldsymbol{I}_n & \boldsymbol{I}_n \end{bmatrix}, \in \mathcal{R}^{(N-1)n \times Nn}, \tag{2.7}$$

with $\boldsymbol{O}_n$ denoting the null $(n \times n)$ matrix. In the case where a rigid workpiece is cooperatively grasped by the multi-robot system, the overall DOFs of the system are reduced due to n DOFs constrained by the rigid body equations. Thus, each component $\boldsymbol{x}_{i,r}$ $(\forall i \in 1, \ldots, N-1)$ keeps its module constant.

By means of a suitable permutation matrix $\boldsymbol{P}$, the Jacobian $\boldsymbol{J}_r$ can be partitioned into its positional and orientational parts, i.e. $\boldsymbol{P}\boldsymbol{J}_r\boldsymbol{P}^{-1} = [\boldsymbol{J}_{rp}^T,\ \boldsymbol{J}_{ro}^T]^T$. Remarkably, the variables $\boldsymbol{x}_a = \left[\boldsymbol{p}_a^{\mathrm{T}}, \boldsymbol{\phi}_a^{\mathrm{T}}\right]^{\mathrm{T}}$ and $\boldsymbol{x}_r = \left[\boldsymbol{p}_r^{\mathrm{T}}, \boldsymbol{\phi}_r^{\mathrm{T}}\right]^{\mathrm{T}}$ define the cooperative task space. Due to this choice, a cooperative motion task described in terms of absolute and relative pose, $\boldsymbol{x}_a$ and $\boldsymbol{x}_r$, is fully equivalent to a barycenter and formation formulation [8].

The Moore-Penrose pseudoinverses of the Jacobian matrices $\boldsymbol{J}_a$ and $\boldsymbol{J}_r$ are given by:

$$\boldsymbol{J}_a^{\dagger} = \boldsymbol{J}_a^{\top}\left(\boldsymbol{J}_a\boldsymbol{J}_a^{\top}\right)^{-1} = \mathbf{1}_N \otimes \boldsymbol{I}_6 = \begin{bmatrix} \boldsymbol{J}_{a,1}^{\dagger} \\ \vdots \\ \boldsymbol{J}_{a,N}^{\dagger} \end{bmatrix}, \tag{2.8}$$

$$\boldsymbol{J}_r^{\dagger} = \boldsymbol{J}_r^{\mathrm{T}}\left(\boldsymbol{J}_r\boldsymbol{J}_r^{\mathrm{T}}\right)^{-1} = \begin{bmatrix} \boldsymbol{J}_{r,1}^{\dagger} \\ \vdots \\ \boldsymbol{J}_{r,N}^{\dagger} \end{bmatrix}, \tag{2.9}$$

where for each $i = 1, \ldots, N$,

$$\boldsymbol{J}_{a,i}^{\dagger} = \boldsymbol{I}_6\,, \tag{2.10}$$

$$\boldsymbol{J}_{r,i}^{\dagger} = \left[\frac{1}{N},\ \ldots,\ \frac{i-1}{N},\ -\frac{N-i}{N},\ \ldots,\ -\frac{1}{N}\right] \otimes \boldsymbol{I}_6. \tag{2.11}$$

It can be verified that the task Jacobian $\boldsymbol{J}_a$ and $\boldsymbol{J}_r$ satisfy the following *orthogonality* conditions [105]:

$$\boldsymbol{J}_a\boldsymbol{J}_r^{\mathrm{T}} = \boldsymbol{O}_{6\times 6(N-1)}, \quad \boldsymbol{J}_r\boldsymbol{J}_a^{\mathrm{T}} = \boldsymbol{O}_{6(N-1)\times 6}\,, \tag{2.12}$$

$$\boldsymbol{J}_a\boldsymbol{J}_r^{\dagger} = \boldsymbol{O}_{6\times 6(N-1)}, \quad \boldsymbol{J}_r\boldsymbol{J}_a^{\dagger} = \boldsymbol{O}_{6(N-1)\times 6}\,. \tag{2.13}$$

Thus, a trajectory defined in terms of absolute and relative motion can be fulfilled by the multi-robot system.

By resorting to the positional part of Equation (2.6), i.e., extracting the expression of $\boldsymbol{p}_r = \left[\boldsymbol{p}_{r_1}^{\mathrm{T}}, \ldots, \boldsymbol{p}_{r_{N-1}}^{\mathrm{T}}\right]^{\mathrm{T}}$, straightforward mathematical steps yield:

$$\boldsymbol{p}_r = \begin{bmatrix} \boldsymbol{p}_2 - \boldsymbol{p}_a + \boldsymbol{p}_a - \boldsymbol{p}_1 \\ \boldsymbol{p}_3 - \boldsymbol{p}_a + \boldsymbol{p}_a - \boldsymbol{p}_2 \\ \vdots \\ \boldsymbol{p}_N - \boldsymbol{p}_a + \boldsymbol{p}_a - \boldsymbol{p}_{N-1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}_2 - \boldsymbol{r}_1 \\ \boldsymbol{r}_3 - \boldsymbol{r}_2 \\ \vdots \\ \boldsymbol{r}_N - \boldsymbol{r}_{N-1} \end{bmatrix}, \tag{2.14}$$

where $\boldsymbol{r}_i = \boldsymbol{p}_i - \boldsymbol{p}_a$ represents the so-called *virtual stick* of the i-th robot [17]. Its definition belongs to the *symmetric formulation* of a dual-arm robotic cell: this vector rigidly translates the reference frame of i-th robot end-effector, $\boldsymbol{\Sigma}_i$, into the absolute frame, $\boldsymbol{\Sigma}_a$.

With similar algebraic calculations applied to (2.3), the following relations can be retrieved:

$$\sum_{i=1}^{N} \boldsymbol{p}_i - N\boldsymbol{p}_a = \sum_{i=1}^{N} (\boldsymbol{p}_i - \boldsymbol{p}_a) = \sum_{i=1}^{N} \boldsymbol{r}_i = 0. \tag{2.15}$$

Thus, (2.14) and (2.15) can be rearranged as follows:

$$\begin{bmatrix} -\boldsymbol{I}_3 & \boldsymbol{I}_3 & \boldsymbol{O}_3 & \dots & \boldsymbol{O}_3 \\ \boldsymbol{O}_3 & -\boldsymbol{I}_3 & \boldsymbol{I}_3 & \dots & \boldsymbol{O}_3 \\ \vdots & \ddots & \ddots & & \vdots \\ \boldsymbol{O}_3 & \boldsymbol{O}_3 & \dots & -\boldsymbol{I}_3 & \boldsymbol{I}_3 \\ \boldsymbol{I}_3 & \boldsymbol{I}_3 & \boldsymbol{I}_3 & \dots & \boldsymbol{I}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \\ \vdots \\ \boldsymbol{r}_{N-1} \\ \boldsymbol{r}_N \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_{rp} \\ \boldsymbol{J}_{ap} \end{bmatrix} \begin{bmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \\ \vdots \\ \boldsymbol{r}_{N-1} \\ \boldsymbol{r}_N \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}_{r_1} \\ \boldsymbol{p}_{r_2} \\ \vdots \\ \boldsymbol{p}_{r_{N-1}} \\ 0 \end{bmatrix}. \tag{2.16}$$

Remark 2.3.1. *A common and practical choice consists in setting the object frame to be coincident with the absolute one. Moreover, notice that the virtual sticks' modules are constant in the presence of an undeformable object tightly grasped by the robots.*

Similar arguments can be made considering the orientation parts of (2.3) and (2.6). Namely, extracting the collective vector $\boldsymbol{\phi}_r = \left[\boldsymbol{\phi}_{r_1}^{\mathrm{T}}, \dots, \boldsymbol{\phi}_{r_{N-1}}^{\mathrm{T}}\right]^{\mathrm{T}}$ from (2.6), it yields

$$\boldsymbol{\phi}_r = \begin{bmatrix} \boldsymbol{\phi}_2 - \boldsymbol{\phi}_a + \boldsymbol{\phi}_a - \boldsymbol{\phi}_1 \\ \boldsymbol{\phi}_3 - \boldsymbol{\phi}_a + \boldsymbol{\phi}_a - \boldsymbol{\phi}_2 \\ \vdots \\ \boldsymbol{\phi}_N - \boldsymbol{\phi}_a + \boldsymbol{\phi}_a - \boldsymbol{\phi}_{N-1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\phi}_{2,a} - \boldsymbol{\phi}_{1,a} \\ \boldsymbol{\phi}_{3,a} - \boldsymbol{\phi}_{2,a} \\ \vdots \\ \boldsymbol{\phi}_{N,a} - \boldsymbol{\phi}_{N-1,a} \end{bmatrix}, \tag{2.17}$$

where $\boldsymbol{\phi}_{i,a}$ denotes the orientation-related variable associated with the virtual stick of the i-th robot: in the Euler-angle space, $\boldsymbol{\phi}_{i,a}$ expresses the relative rotation of $\boldsymbol{\Sigma}_a$ with respect to $\boldsymbol{\Sigma}_i$. Similarly, from (2.3), it follows:

$$\sum_{i=1}^{N} \boldsymbol{\phi}_i - N\boldsymbol{\phi}_a = \sum_{i=1}^{N} (\boldsymbol{\phi}_i - \boldsymbol{\phi}_a) = \sum_{i=1}^{N} \boldsymbol{\phi}_{i,a} = 0. \tag{2.18}$$

Thus, the linearity of this representation is exploited for rearranging the previous equations into the following linear system:

$$\begin{bmatrix} -\boldsymbol{I}_3 & \boldsymbol{I}_3 & \boldsymbol{O}_3 & \dots & \boldsymbol{O}_3 \\ \boldsymbol{O}_3 & -\boldsymbol{I}_3 & \boldsymbol{I}_3 & \dots & \boldsymbol{O}_3 \\ \vdots & \ddots & \ddots & & \vdots \\ \boldsymbol{O}_3 & \boldsymbol{O}_3 & \dots & -\boldsymbol{I}_3 & \boldsymbol{I}_3 \\ \boldsymbol{I}_3 & \boldsymbol{I}_3 & \boldsymbol{I}_3 & \dots & \boldsymbol{I}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}_{1,a} \\ \boldsymbol{\phi}_{2,a} \\ \vdots \\ \boldsymbol{\phi}_{N-1,a} \\ \boldsymbol{\phi}_{N,a} \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_{ro} \\ \boldsymbol{J}_{ao} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}_{1,a} \\ \boldsymbol{\phi}_{2,a} \\ \vdots \\ \boldsymbol{\phi}_{N-1,a} \\ \boldsymbol{\phi}_{N,a} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\phi}_{r_1} \\ \boldsymbol{\phi}_{r_2} \\ \vdots \\ \boldsymbol{\phi}_{r_{N-1}} \\ 0 \end{bmatrix}. \tag{2.19}$$

As a consequence, the orientation of the i-th end-effector frame, $\boldsymbol{\phi}_i$, can be retrieved straightforwardly as

$$\boldsymbol{\phi}_i = \boldsymbol{\phi}_{i,a} - \boldsymbol{\phi}_a. \tag{2.20}$$

2.4 Multi-agent system and communication topology

According to Assumption 4, the agents have no prior knowledge of the trajectory assigned to the absolute and relative motion variables. Furthermore, fully connected communication among the agents is not available. Therefore, it is supposed that information exchange occurs over a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes (indexed by the N agents), $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of directed edges connecting the nodes. The matrix $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix defined by

$$\boldsymbol{A} = \{a_{ij}\} : \quad a_{ii} = 0, \quad a_{ij} = \begin{cases} 1 & \text{if } (j,i) \in \mathcal{E} \\ 0 & \text{otherwise,} \end{cases} \tag{2.21}$$

More in detail, $a_{ij} \neq 0$ indicates that agent j can send information to agent i, meaning that agent j belongs to the set of neighbors of agent i, i.e. $j \in \mathcal{N}_i$ where $\mathcal{N}_i = \{j \in \mathcal{V} : (j,i) \in \mathcal{E}\}$. It is assumed that the agents do not know the topology of the overall communication graph but only which are the agents they share information with, i.e their neighbors.

Moreover, the Laplacian matrix $\boldsymbol{L} \in \mathbb{R}^{N \times N}$ of the graph is given by

$$\boldsymbol{L} = \{l_{ij}\} : \quad l_{ii} = \sum_{j=1, j \neq i}^{N} a_{ij}, \quad l_{ij} = -a_{ij}, \; i \neq j \,. \tag{2.22}$$

Note that $\boldsymbol{L}$ always has at least one zero eigenvalue with the corresponding right eigenvector being $\mathbf{1}_N$. Therefore, $\text{rank}(\boldsymbol{L}) \leq N - 1$ and $\boldsymbol{L}\mathbf{1}_N = \mathbf{0}_N$, where $\mathbf{0}_N$ is the $(N \times 1)$ null vector.

A directed graph is said to be *balanced* if, for every node $i \in \mathcal{V}$, the sum of the weights of the incoming edges equals the sum of the weights of the outgoing edges, namely $\forall i \in \mathcal{V} \;\; \sum_j a_{ij} = \sum_j a_{ji}$ ([29]). Consequently, for a balanced directed graph, $\mathbf{1}_N$ is also a left eigenvector of $\boldsymbol{L}$, i.e., $\mathbf{1}_N^{\mathrm{T}} \boldsymbol{L} = \mathbf{0}_N^{\mathrm{T}}$. Moreover, a directed graph is called *strongly connected* if any two distinct nodes of the graph can be connected via a

directed path, i.e., a path that follows the direction of the edges of the graph. In the case of a strongly connected graph, $\text{rank}(\boldsymbol{L}) = N - 1$.

The set of nodes $\mathcal{V}$ of the communication graph can be partitioned in two disjoint subsets: $\mathcal{V}_a$, with cardinality V_a, and $\mathcal{V}_b$, with cardinality V_b, so that $\mathcal{V}_b = \mathcal{V} \setminus \mathcal{V}_a$. For graphs with such a partition, the requirement of strong connectivity can be relaxed by considering the weaker *subgraph connectivity* (SGC) property [34].

Property 1. *A directed graph, $\mathcal{G}(\mathcal{V}_a \cup \mathcal{V}_b, \mathcal{E})$, is said to satisfy the SGC property if, for any node $i_b \in \mathcal{V}_b$, there exists a directed path from at least one node $j_a \in \mathcal{V}_a$ to i_b.*

Let us recall Theorem 1 from [34], which builds upon the results presented in [106], and is used to establish the stability of the proposed estimation algorithm. The complete proof is provided in Appendix B of the aforementioned reference.

Theorem 1. *Let $\mathcal{G}$ be a directed graph satisfying the SGC property with Laplacian matrix $\boldsymbol{L}$. Consider the diagonal matrix $\boldsymbol{\Gamma} = \text{diag}\{\gamma_1, \ldots, \gamma_N\}$ with at least one positive diagonal entry, i.e., $\exists\ i \in \{1, \ldots, N\} : \gamma_i > 0$. Then, matrix $-\bar{\boldsymbol{L}} = -(\boldsymbol{L} + \boldsymbol{\Gamma})$ is Hurwitz.*

Moreover, by resorting the results in [10], let us introduce a property that will be used in the following for graphs satisfying the SGC property.

Property 2. *For each $i = 1, \ldots, N$, the submatrix $-\bar{\boldsymbol{L}}_i$, obtained from $-\boldsymbol{L}$ by removing its i-th row and i-th column, is Hurwitz.*

2.5 Distributed estimation of reference trajectories in coordinated motion tasks

In several application scenarios, severe communication and/or computational constraints could prevent the reference trajectories from being broadcast to (or computed by) every agent. Taking into account the coordinated motion task, presented in Section 2.3, the task variables would be expressed by the absolute and relative trajectories of the cooperative robot team, i.e., $\boldsymbol{x}_{a_d}$ $(\dot{\boldsymbol{x}}_{a_d}, \ddot{\boldsymbol{x}}_{a_d})$ and $\boldsymbol{x}_{r_d}$ $(\dot{\boldsymbol{x}}_{r_d}, \ddot{\boldsymbol{x}}_{r_d})$, where the subscript d denotes the desired trajectories to be tracked by the control law.

In such cases, it is assumed that only a subset of robots receives the reference trajectories from a *virtual leader*, which is a common approach to model situations in which only a subset of team agents is aware of some team-level variables [107]. The

connection between the virtual leader, with index $i = 0$, and the robots in the team is modeled through a set of weights γ_{0i} $(i = 1, \ldots, N)$ defined as follows:

$$\gamma_{0i} = \begin{cases} 1 & \text{if the virtual leader communicates with robot } i, \\ 0 & \text{otherwise.} \end{cases}$$

More in detail, the agents satisfying the first case, i.e., which are in the neighborhood of the virtual leader, belong to the subset of leaders, $\mathcal{V}_l$. These agents have access to information about the global tasks, defined at the team level. In turn, the agents, which do not communicate with the virtual leader, belong to the subset of followers, $\mathcal{V}_f$.

Therefore, the i-th agent can rely on a distributed estimate of the desired team variables, namely ${}^i\hat{\boldsymbol{x}}_{a_d}$ $({}^i\dot{\hat{\boldsymbol{x}}}_{a_d}, {}^i\ddot{\hat{\boldsymbol{x}}}_{a_d})$ and ${}^i\hat{\boldsymbol{x}}_{r_d}$ $({}^i\dot{\hat{\boldsymbol{x}}}_{r_d}, {}^i\ddot{\hat{\boldsymbol{x}}}_{r_d})$.

The proposed strategy is structured in two steps:

1. first, each robot estimates the desired absolute and relative trajectories, along with their time derivatives, by employing a bank of estimators[1];
2. then, it locally computes the reference motion for its end-effector in a distributed manner, based on the estimated cooperative task information.

It is worth noting that these two steps are intended to be logically consecutive, rather than temporally consecutive: at each time instant, any robot simultaneously performs both the steps. Moreover, as it will be more detailed in the following sections, this approach enables the reference trajectory estimate to be modified in accordance with other exogenous estimated signals, such as the exchanged wrenches.

2.5.1 Cooperative task space trajectories estimation

Hereafter, the notation $\boldsymbol{x}_d$ will denote either $\boldsymbol{x}_{a_d}$ or $\boldsymbol{x}_{r_d}$, since the estimation scheme is formally identical for both trajectories. Consequently, all subscripts a and r will be dropped. In the following, $\|\cdot\|$ denotes the 2-norm for both vectors and matrices and N_t is equal to n when $* = a$ and $n(N-1)$ when $* = r$. Moreover, $\boldsymbol{f}^{(k)}(t)$ represents the k-th time derivative of the function $\boldsymbol{f}(t)$, whereas, in the case of estimated variables, ${}^j\hat{\boldsymbol{f}}_k(t)$ denotes the j-th agent estimate of the k-th order derivative of the signal $\boldsymbol{f}(t)$.

[1] Although the use of the term *estimator* may be somewhat improper in this framework, it is favored over *observer*, which in turn could be misleading, as it pertains to exogenous signals, namely absolute and relative trajectories, interaction forces at the end-effector, and further geometric parameters.

The desired trajectories must satisfy either Assumption 5 or the milder Assumption 6 defined below.

Assumption 5. *There exists an integer* $p_* \geq 1$ *such that* $\boldsymbol{x}_{*,d}^{(p_*+1)}(t) = \mathbf{0}_{N_t}$, $\forall t \geq 0$.

Assumption 6. *There exist an integer* $p_* \geq 1$ *and a constant* $\bar{x}_*^{(p_*+1)} < \infty$, *such that* $\|\boldsymbol{x}_{*,d}^{(p_*+1)}(t)\| \leq \bar{x}_*^{(p_*+1)}$, $\forall t \geq 0$.

In particular, trajectories with a polynomial time law satisfy Assumption 5 with the parameter p_* depending on the order of the polynomial. Conversely, Assumption 6 mitigates the strictness of the requirement, but, as shown in the following, the achievable performance of the proposed control approach depends on the magnitude of the bounding constant $\bar{x}_*^{(p_*+1)}$. For instance, reference trajectories whose time law is a sinusoidal function belong to this case study. Generally, within an integrated objective design framework, trajectory generation can be carried out by prescribing a user-defined p_* and $\bar{x}_*^{(p_*+1)}$.

The objective is achieved through the implementation of a cascade of $p+1$ consensus-based estimator bank, inherited by [34], which enables the i-th robotic agent to compute an estimate of the derivative of order h, ${}^i\hat{\boldsymbol{x}}_{d,h}(t)$, with $h = p, p-1, \ldots, 1, 0$, as follows:

$$ {}^i\dot{\hat{\boldsymbol{x}}}_{d,h}(t) = k_o \sum_{j \in \mathcal{N}_i} \left({}^j\hat{\boldsymbol{x}}_{d,h}(t) - {}^i\hat{\boldsymbol{x}}_{d,h}(t) \right) + {}^i\hat{\boldsymbol{x}}_{d,h+1}(t) + k_o \gamma_{0i} (\boldsymbol{x}_d^{(h)}(t) - {}^i\hat{\boldsymbol{x}}_{d,h}(t)) \,, \quad (2.23) $$

where k_o is a positive gain and it is assumed that ${}^i\hat{\boldsymbol{x}}_{d,p+1}(t) = \mathbf{0}_{N_t}$ $\forall\, i \in \mathcal{N}_i \,, \forall t \geq 0$.

To simplify the notation, the temporal dependence is henceforth omitted. With simple mathematical operations, the dynamics of the estimation error, ${}^i\tilde{\boldsymbol{x}}_{d,h}$, can be retrieved:

$$ \begin{aligned} {}^i\dot{\tilde{\boldsymbol{x}}}_{d,h} = \; & \boldsymbol{x}_d^{(h+1)} - {}^i\dot{\hat{\boldsymbol{x}}}_{d,h} = \quad (2.24) \\ & = -k_o \sum_{j \in \mathcal{N}_i} \left({}^j\hat{\boldsymbol{x}}_{d,h} - {}^i\hat{\boldsymbol{x}}_{d,h} + \boldsymbol{x}_d^{(h)} - \boldsymbol{x}_d^{(h)} \right) - {}^i\hat{\boldsymbol{x}}_{d,h+1} + \boldsymbol{x}_d^{(h+1)} - k_o \gamma_{0i} (\boldsymbol{x}_d^{(h)} - {}^i\hat{\boldsymbol{x}}_{d,h}) = \\ & = -k_o \sum_{j \in \mathcal{N}_i} \left({}^i\tilde{\boldsymbol{x}}_{d,h} - {}^j\tilde{\boldsymbol{x}}_{d,h} \right) + (1 - \delta_{hp}) {}^i\tilde{\boldsymbol{x}}_{d,h+1} + \delta_{hp}\, \boldsymbol{x}_d^{(p+1)} - k_o \gamma_{0i}\, {}^i\tilde{\boldsymbol{x}}_{d,h}, \end{aligned} $$

where the Kronecker delta

$$ \delta_{hp} = \begin{cases} 1 & \text{if } h = p, \\ 0 & \text{if } h \neq p. \end{cases} $$

Then, the dynamics of the collective estimation error, $\widetilde{\boldsymbol{x}}_{d,h} = \left[{}^1\widetilde{\boldsymbol{x}}_{d,h}^{\mathrm{T}}, \quad \ldots, \quad {}^N\widetilde{\boldsymbol{x}}_{d,h}^{\mathrm{T}}\right]^{\mathrm{T}}$, is given by

$$\dot{\widetilde{\boldsymbol{x}}}_{d,h} = -\bar{\boldsymbol{L}}\,\widetilde{\boldsymbol{x}}_{d,h} + (1-\delta_{hp})\widetilde{\boldsymbol{x}}_{d,h+1} + \delta_{hp}(\mathbf{1}_N \otimes \boldsymbol{x}_d^{(p+1)}), \tag{2.25}$$

where

$$\bar{\boldsymbol{L}} = k_o\left((\boldsymbol{L} + \boldsymbol{\Gamma}_0) \otimes \boldsymbol{I}_{N_t}\right), \tag{2.26}$$

with $\boldsymbol{\Gamma}_0 = \mathrm{diag}\{\gamma_{01}, \ldots, \gamma_{0N}\}$.

Notice that the estimators in (2.23) are defined recursively, i.e., at each stage of implementation, the estimator for the h-th derivative of $\boldsymbol{x}_d$ requires the output of the estimator for the $(h+1)$-th derivative.

By exploiting the properties of the Kronecker product, any eigenvalue of the matrix $-(\boldsymbol{L} + \boldsymbol{\Gamma}_0) \otimes \boldsymbol{I}_{N_t}$ can be expressed as the product of the eigenvalues of $-(\boldsymbol{L} + \boldsymbol{\Gamma}_0)$ and $\boldsymbol{I}_{N_t}$ [104, 108], from Theorem 1 and the results in [109], the following lemma can be stated.

Lemma 1. *Given the subset of leaders, $\mathcal{V}_l$, and the subset of followers, $\mathcal{V}_f$, if the communication graph $\mathcal{G}(\mathcal{V}_l \cup \mathcal{V}_f, \mathcal{E})$ satisfies the SGC property, then the matrix $-\bar{\boldsymbol{L}}$ is Hurwitz. Moreover, given any real number $\epsilon > 0$, there exists a real number $\rho(\epsilon) > 0$ such that the transition matrix $\boldsymbol{\Phi}(t, t_0) = \exp\left(-\bar{\boldsymbol{L}}(t - t_0)\right)$ satisfies the inequality*

$$\|\boldsymbol{\Phi}(t, t_0)\| \leq \rho(\epsilon)\, e^{-\alpha(\epsilon)(t-t_0)}, \quad \forall t \geq t_0, \tag{2.27}$$

where[2] *$\alpha(\epsilon) = -(\lambda_M + \epsilon)$, $\lambda_M < 0$ is the largest eigenvalue of $-k_o(\boldsymbol{L} + \boldsymbol{\Gamma}_0)$ and t_0 is the initial time.*

The following theorems, whose proofs are reported in Appendices A1 and A2, establish the convergence properties of the h-th derivative order of the reference trajectory estimation error $\widetilde{\boldsymbol{x}}_{d,h}$ $(\forall h = 0, 1, \ldots, p)$.

Theorem 2. *If Assumption 5 holds and the communication graph $\mathcal{G}(\mathcal{V}_l \cup \mathcal{V}_f, \mathcal{E})$ satisfies the SGC property, then $\widetilde{\boldsymbol{x}}_{d,h}(t)$ converges exponentially to $\mathbf{0}_{N_t}$ as $t \to \infty$, $\forall h = 0, 1, \ldots, p$ and for any initial condition $\widetilde{\boldsymbol{x}}_{d,h}(t_0)$.*

Theorem 3. *If Assumption 6 holds and the communication graph $\mathcal{G}(\mathcal{V}_l \cup \mathcal{V}_f, \mathcal{E})$ satisfies the SGC property, then $\widetilde{\boldsymbol{x}}_{d,h}(t)$ can be norm-bounded, $\forall h = 0, 1, \ldots, p$ and $\forall t \geq 0$, as*

[2] Notice that ϵ can always be chosen small enough such that $-\alpha(\epsilon) < 0$.

follows

$$\|\widetilde{\boldsymbol{x}}_{d,h}(t)\| \leq \bar{x}_{d,0}^{(h)}(\epsilon) + \bar{\beta}^{(h)}(\epsilon)\bar{x}^{(p+1)}\,, \tag{2.28}$$

where

$$\bar{x}_{d,0}^{(h)}(\epsilon) = \rho(\epsilon)\sum_{l=h}^{p}\left(\frac{\rho(\epsilon)}{\alpha(\epsilon)}\right)^{l-h}\|\widetilde{\boldsymbol{x}}_{d,l+1}(t_0)\|\,, \tag{2.29}$$

$$\bar{\beta}^{(h)}(\epsilon) = \sqrt{N}\left(\frac{\rho(\epsilon)}{\alpha(\epsilon)}\right)^{p-h+1}, \tag{2.30}$$

and $\bar{x}^{(p+1)}$ is defined in Assumption 6, while $\rho(\epsilon)$ and $\alpha(\epsilon)$ are defined in Lemma 1.

Remark 2.5.1. *Notice that $\rho(\epsilon)$ increases for decreasing values of ϵ [109], and, thus, the bound* (2.28) *becomes more conservative. Less conservative bounds exist [109, 110], but here the goal is to show the boundedness of the estimation scheme and not to compute a tight bound. Anyway, it is worth noticing that, for a fixed value of ϵ, the bound decreases as $-\lambda_M > 0$ increases (since $\alpha = -(\lambda_M + \epsilon)$), which, in turn, can be increased by increasing the gain k_o. Assume that a large enough k_o has been chosen, such that $\rho/\alpha < 1$, then equation* (2.30) *shows that the effect of $\bar{x}^{(p+1)}$ on the upper bound on $\|\widetilde{\boldsymbol{x}}_{d,h}(t)\|$ decreases as k decreases (i.e., descending along the chain of estimators); moreover,* (2.29) *shows that the effect of the initial conditions of the estimators preceding the k-th along the chain is attenuated the more the further they are along the chain.*

Corollary 1. *Under the assumptions of Theorem 3, for $t \to \infty$ the bound in* (2.28) *becomes*

$$\|\widetilde{\boldsymbol{x}}_{d,h}(t)\| \leq \bar{\beta}^{(h)}(\epsilon)\bar{x}^{(p+1)}. \tag{2.31}$$

The proof of Corollary 1 is reported in A3.

2.5.2 Robot reference trajectories estimation

By inverting (2.3) and (2.6), and by virtue of the orthogonality conditions (2.12) and (2.13), the desired pose of the i-th robot and its time derivatives up to the p_m order, with $p_m = \min\{p_a, p_r\}$, can be computed as

$$\boldsymbol{x}_{i_d}^{(h)}(t) = \boldsymbol{J}_{a_i}^{\dagger}\,\boldsymbol{x}_{a_d}^{(h)}(t) + \boldsymbol{J}_{r_i}^{\dagger}\,\boldsymbol{x}_{r_d}^{(h)}(t)\,, \tag{2.32}$$

where the submatrices $\boldsymbol{J}_{a_i}^{\dagger}$ and $\boldsymbol{J}_{r_i}^{\dagger}$ are obtained by extracting the rows of $\boldsymbol{J}_{a}^{\dagger}$ and $\boldsymbol{J}_{r}^{\dagger}$ from $n(i-1)+1$ to $n\,i$, respectively, and $h = 0, 1, \ldots, p_m$.

However, since the signals of $\boldsymbol{x}_{a_d}^{(h)}$ and $\boldsymbol{x}_{r_d}^{(h)}$ are provided only for a subset of agents, each robot can compute only an estimate of its reference trajectory, i.e.,

$$ {}^{i}\hat{\boldsymbol{x}}_{i_{d,h}}(t) = \boldsymbol{J}_{a_i}^{\dagger}\, {}^{i}\hat{\boldsymbol{x}}_{a_{d,h}}(t) + \boldsymbol{J}_{r_i}^{\dagger}\, {}^{i}\hat{\boldsymbol{x}}_{r_{d,h}}(t)\,. \tag{2.33} $$

By subtracting (2.33) to (2.32), the estimation error of the i-th robot for the h-th order derivative of its reference trajectory, ${}^{i}\widetilde{\boldsymbol{x}}_{i_{d,h}} = \boldsymbol{x}_{i_d}^{(h)} - {}^{i}\hat{\boldsymbol{x}}_{i_{d,h}}$, is

$$ {}^{i}\widetilde{\boldsymbol{x}}_{i_{d,h}}(t) = \boldsymbol{J}_{a_i}^{\dagger}\, {}^{i}\widetilde{\boldsymbol{x}}_{a_{d,h}}(t) + \boldsymbol{J}_{r_i}^{\dagger}\, {}^{i}\widetilde{\boldsymbol{x}}_{r_{d,h}}(t)\,. \tag{2.34} $$

If Assumption 5 holds and the communication graph satisfies the SGC property, Theorem 2 ensures $\lim_{t\to\infty}\ {}^{i}\widetilde{\boldsymbol{x}}_{d,h}(t) \to \boldsymbol{0}_n$.

On the other hand, recalling (2.4) and (2.7), the norm of $\boldsymbol{J}_a$ and $\boldsymbol{J}_r$ can be expressed as $\left\|\boldsymbol{J}_{a_i}^{\dagger}\right\| = \|\boldsymbol{I}_n\| = 1$, $\left\|\boldsymbol{J}_{r_i}^{\dagger}\right\| = \bar{\varphi}_{r_i}$. Then, if both the Assumption 6 and the SGC property hold, Theorem 3 implies the following upper bound:

$$ \begin{aligned} \left\|{}^{i}\widetilde{\boldsymbol{x}}_{d,h}(t)\right\| &\leq \left\|\boldsymbol{J}_{a_i}^{\dagger}\right\| \left\|{}^{i}\widetilde{\boldsymbol{x}}_{a_{d,h}}(t)\right\| + \left\|\boldsymbol{J}_{r_i}^{\dagger}\right\| \left\|{}^{i}\widetilde{\boldsymbol{x}}_{r_{d,h}}(t)\right\| \leq \ \left\|{}^{i}\widetilde{\boldsymbol{x}}_{a_{d,h}}\right\| + \bar{\varphi}_{r_i}\ \left\|{}^{i}\widetilde{\boldsymbol{x}}_{r_{d,h}}\right\| \leq \\ &\leq \bar{x}_{a_{d,0}}^{(h)}(\epsilon) + \bar{\varphi}_{r_i}\, \bar{x}_{r_{d,0}}^{(h)}(\epsilon) + \bar{\beta}^{(h)}(\epsilon)\left(\bar{x}_{a}^{(p_a+1)} + \bar{\varphi}_{r_i}\, \bar{x}_{r}^{(p_r+1)}\right), \end{aligned} \tag{2.35} $$

where $\bar{x}_{a_d}^{(h)}$, $\bar{x}_{r_d}^{(h)}$ and $\bar{x}_{a}^{(p_a+1)}$, $\bar{x}_{r}^{(p_r+1)}$ are defined as in (2.28) and in Assumption 6, respectively.

Considering now the limit as $t \to \infty$ of (2.35), by virtue of Corollary 1, the bound on the tracking error can be re-written as

$$ \lim_{t\to\infty} \left\|{}^{i}\widetilde{\boldsymbol{x}}_{d,h}(t)\right\| \leq \bar{\beta}^{(h)}(\epsilon)\left(\bar{x}_{a}^{(p_a+1)} + \bar{\varphi}_{r_i}\, \bar{x}_{r}^{(p_r+1)}\right). \tag{2.36} $$

Once the estimate of the desired end-effector pose has been computed, by adopting a first order Closed-Loop Inverse Kinematics algorithm [111], the commanded joint velocities for the i-th robot ($i = 1, \ldots, N$), $\dot{\boldsymbol{q}}_{i_c}$, are given by

$$ \dot{\boldsymbol{q}}_{i_c} = \boldsymbol{J}_i^{\dagger}\left[{}^{i}\hat{\boldsymbol{x}}_{i_{d,1}} + \boldsymbol{K}_{q,i}\left({}^{i}\hat{\boldsymbol{x}}_{i_{d,0}} - \boldsymbol{x}_i\right)\right], \tag{2.37} $$

where $\boldsymbol{J}_i$ is the analytic Jacobian matrix, $\boldsymbol{K}_{q,i}$ is a matrix of positive gains, and $\boldsymbol{x}_i$ is the actual pose of the i-th manipulator end-effector. Taking into account the estimation

error, it yields

$$\dot{\boldsymbol{q}}_{i_c} = \boldsymbol{J}_i^{\dagger} \left[\dot{\boldsymbol{x}}_{i_d} + \boldsymbol{K}_{q,i} \left(\boldsymbol{x}_{i_d} - \boldsymbol{x}_i \right) \right] - \boldsymbol{J}_i^{\dagger} \left[{}^{i}\widetilde{\boldsymbol{x}}_{i_{d,1}} + \boldsymbol{K}_{q,i} \, {}^{i}\widetilde{\boldsymbol{x}}_{i_{d,0}} \right]. \tag{2.38}$$

By multiplying both sides of (2.38) by $\boldsymbol{J}_i$:

$$\begin{aligned} \dot{\boldsymbol{x}}_i = \boldsymbol{J}_i \, \dot{\boldsymbol{q}}_{i_c} = \boldsymbol{J}_i \boldsymbol{J}_i^{\dagger} \left[\dot{\boldsymbol{x}}_{i_d} + \boldsymbol{K}_{q,i} \left(\boldsymbol{x}_{i_d} - \boldsymbol{x}_i \right) \right] - \boldsymbol{J}_i \boldsymbol{J}_i^{\dagger} \left[{}^{i}\widetilde{\boldsymbol{x}}_{i_{d,1}} + \boldsymbol{K}_{q,i} \, {}^{i}\widetilde{\boldsymbol{x}}_{i_{d,0}} \right] = \\ = \dot{\boldsymbol{x}}_{i_d} + \boldsymbol{K}_{q,i} \left(\boldsymbol{x}_{i_d} - \boldsymbol{x}_i \right) - \left({}^{i}\widetilde{\boldsymbol{x}}_{i_{d,1}} + \boldsymbol{K}_{q,i} \, {}^{i}\widetilde{\boldsymbol{x}}_{i_{d,0}} \right) \end{aligned} \tag{2.39}$$

Thus, the dynamics of the tracking error $\widetilde{\boldsymbol{x}}_i = \boldsymbol{x}_{i_d} - \boldsymbol{x}_i$ can be straightforwardly derived as

$${}^{i}\dot{\widetilde{\boldsymbol{x}}}_i = -\boldsymbol{K}_{q,i} \, {}^{i}\widetilde{\boldsymbol{x}}_i + {}^{i}\widetilde{\boldsymbol{x}}_{i_{d,1}} + \boldsymbol{K}_{q,i} \, {}^{i}\widetilde{\boldsymbol{x}}_{i_{d,0}}. \tag{2.40}$$

The system (2.40) can be seen as a first order exponentially stable system, whose input is the perturbation ${}^{i}\widetilde{\boldsymbol{x}}_{i_{d,1}} + \boldsymbol{K}_{q,i} \, {}^{i}\widetilde{\boldsymbol{x}}_{i_{d,0}}$:

- if the trajectories assigned for the cooperative motion tasks verify Assumption 5 and the communication graph satisfies the SGC property, the perturbation can be considered as vanishing as $t \to \infty$, by virtue of Theorem 2. Thus, by exploiting the results in [112], $\lim_{t \to \infty} {}^{i}\widetilde{\boldsymbol{x}}_i \to \boldsymbol{0}_n$;
- conversely, if only Assumption 6 holds, the perturbation is non-vanishing and Theorem 3 ensures that ${}^{i}\widetilde{\boldsymbol{x}}_{i_{d,1}}$ and ${}^{i}\widetilde{\boldsymbol{x}}_{i_{d,0}}$ are bounded as in (2.35). Then, by exploiting the results in [112], ${}^{i}\dot{\widetilde{\boldsymbol{x}}}_i$ is bounded, as well.

2.5.3 Robot reference trajectories: an alternative formulation

Consider the problem of trajectory planning for a cooperative task. In the presence of an object cooperatively grasped by the robotic team, the absolute frame may coincide with the object frame. In such cases, two distinct tasks can be identified: a barycentric task, when the absolute frame coincides with the object's center of mass, and a formation task. For the trajectory of the absolute (or object) frame, a geometric primitive is defined in the operational space, along with its corresponding time law. Based on the assumptions introduced earlier, the values of the parameter p_* and the upper bound $\bar{x}_*^{p_*+1}$ are accordingly specified.

Notably, in cooperative transportation tasks involving a rigid object, the formation trajectory is fully constrained by the position and orientation trajectories assigned to the absolute frame. Due to the rigidity of the object, no relative motion between

the robots' end-effector frames is allowed. This observation motivates an alternative formulation of the relative task, expressed with respect to the absolute frame, that can also be extended to coordinated motion tasks. Since the relative motion variables depend on $\boldsymbol{p}_a$ and $\boldsymbol{\phi}_a$, computing their derivatives up to order $p_r + 1$ from the virtual leader's perspective becomes increasingly complex and cumbersome, as it would require accounting for the derivatives of the absolute variables, which are either constant or zero beyond a certain differentiation order p_a. Conversely, by expressing $\boldsymbol{p}_r$ and $\boldsymbol{\phi}_r$ with respect to the absolute frame $\boldsymbol{\Sigma}_a$, this dependency is effectively eliminated, thereby significantly simplifying the computation.

However, although this alternative formulation of the formation task with respect to the absolute frame offers a significantly more streamlined approach, it would require a revision of the equations introduced in the previous section.

With regard to the position, it is necessary to take into account a variable that allows to estimate (or compute) the end-effector position trajectory based on global kinematic variables, i.e., the virtual stick, introduced in Section 2.3. Inverting the system in (2.16), each agent can extract the rows corresponding to its own virtual stick in a distributed manner:

$$ {}^{i}\hat{\boldsymbol{r}}_i = \boldsymbol{\Gamma}_i \begin{bmatrix} \boldsymbol{J}_{rp} \\ \boldsymbol{J}_{ap} \end{bmatrix}^{\dagger} \begin{bmatrix} {}^{i}\hat{\boldsymbol{p}}^a_{r_1} \\ {}^{i}\hat{\boldsymbol{p}}^a_{r_2} \\ \vdots \\ {}^{i}\hat{\boldsymbol{p}}^a_{r_{N-1}} \\ 0 \end{bmatrix}, \tag{2.41} $$

where ${}^{i}\hat{\boldsymbol{p}}^a_r$ is expressed with reference to the local frame $\boldsymbol{\Sigma}_a$ and

$$ \boldsymbol{\Gamma}_i = \{\boldsymbol{O}_3 \quad \cdots \quad \underbrace{\boldsymbol{I}_3}_{i\text{-th robot}} \quad \cdots \quad \boldsymbol{O}_3\} \in \mathbb{R}^{3\times 3N}. $$

As a matter of fact, each agent could also retrieve an estimate of any other agent end-effector pose, leveraging on the corresponding virtual stick estimate, ${}^{i}\hat{\boldsymbol{r}}_j$, by resorting to the $\boldsymbol{\Gamma}_j$ matrix.

Subsequently, the position trajectory of the i-th end-effector can be reconstructed as:

$$ {}^{i}\hat{\boldsymbol{p}}_i = {}^{i}\hat{\boldsymbol{p}}_a + \boldsymbol{R}({}^{i}\hat{\boldsymbol{\phi}}_a){}^{i}\hat{\boldsymbol{r}}^a_i \tag{2.42} $$

that is structurally equivalent to the closed-chain constraints.

A completely analogous reasoning holds for orientation, by inverting the system in (2.19):

$$ {}^{i}\hat{\boldsymbol{\phi}}_{i,a} = \boldsymbol{\Gamma}_i \begin{bmatrix} \boldsymbol{J}_{ro} \\ \boldsymbol{J}_{ao} \end{bmatrix}^{\dagger} \begin{bmatrix} {}^{i}\hat{\boldsymbol{\phi}}_{1,a} \\ {}^{i}\hat{\boldsymbol{\phi}}_{2,a} \\ \vdots \\ {}^{i}\hat{\boldsymbol{\phi}}_{N-1,a} \\ 0 \end{bmatrix}. \tag{2.43} $$

Then, the current orientation of the end-effector frame can be obtained through

$$ {}^{i}\hat{\boldsymbol{\phi}}_{i} = {}^{i}\hat{\boldsymbol{\phi}}_{i,a} - {}^{i}\hat{\boldsymbol{\phi}}_{a}. \tag{2.44} $$

It is noticeable that both (2.42) and (2.44) hold also for coordinated motion tasks: considering (2.41), the virtual stick ${}^{i}\hat{\boldsymbol{r}}_{i}^{a}$ accounts for the relative distance between $\boldsymbol{\Sigma}_i$ and $\boldsymbol{\Sigma}_a$, as well as their relative orientation is taken into account by ${}^{i}\hat{\boldsymbol{\phi}}_{i,a}$, according to (2.43).

Adopting a velocity controller at the low-level control of the robot, only position and velocity inputs are needed to track the end-effector reference trajectory. To this aim, the derivative of (2.42) and (2.44) need to be calculated in lieu of the closed-chain constraints, in order to generalize for distributed control strategies without the presence of a rigid workpiece cooperatively grasped by the robotic team.

Taking the derivative of (2.42) and omitting the superscript i for the sake of compactness, the reference velocity of the i-th end-effector can be obtained as follows:

$$ \begin{aligned} \dot{\hat{\boldsymbol{p}}}_i &= \dot{\hat{\boldsymbol{p}}}_a + \boldsymbol{S}(\hat{\boldsymbol{\omega}}_a)\boldsymbol{R}(\hat{\boldsymbol{\phi}}_a)\hat{\boldsymbol{r}}_i^a + \boldsymbol{R}(\hat{\boldsymbol{\phi}}_a)\dot{\hat{\boldsymbol{r}}}_i^a = \\ &= \dot{\hat{\boldsymbol{p}}}_a + \boldsymbol{S}(\boldsymbol{T}(\hat{\boldsymbol{\phi}}_a)\dot{\hat{\boldsymbol{\phi}}}_a)\boldsymbol{R}(\hat{\boldsymbol{\phi}}_a)\hat{\boldsymbol{r}}_i^a + \boldsymbol{R}(\hat{\boldsymbol{\phi}}_a)\dot{\hat{\boldsymbol{r}}}_i^a, \end{aligned} \tag{2.45} $$

where $\boldsymbol{S}(\cdot)$ indicates the (3×3) skew-symmetric operator, such that $\dot{\boldsymbol{R}}(t) = \boldsymbol{S}(\boldsymbol{\omega}(t))\boldsymbol{R}(t)$, whose expression is

$$ \boldsymbol{S} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}, \tag{2.46} $$

and $\boldsymbol{\omega}_a$ denotes the angular velocity of $\boldsymbol{\Sigma}_a$. In particular, $\boldsymbol{\omega}_a = \boldsymbol{T}(\boldsymbol{\phi}_a)\dot{\boldsymbol{\phi}}_a$, with $\boldsymbol{T}(\boldsymbol{\phi}_a)$ relating the Euler angle rates $\dot{\boldsymbol{\phi}}_a$ to the angular velocity $\boldsymbol{\omega}_a$ [101].

In the presence of a rigid object, the virtual stick module is constant; thereby, $\dot{\boldsymbol{r}}_i^a = 0$ and (2.45) reduces to:

$$\dot{\hat{\boldsymbol{p}}}_i = \dot{\hat{\boldsymbol{p}}}_a + \boldsymbol{S}(\boldsymbol{T}(\hat{\boldsymbol{\phi}}_a)\dot{\hat{\boldsymbol{\phi}}}_a)\boldsymbol{R}(\hat{\phi}_a)\hat{\boldsymbol{r}}_i^a, \tag{2.47}$$

that is the equation of a generic displacement of a rigid body, according to the Mozzi–Chasles' theorem [113].

Similarly, starting from the relation between the angular velocities of $\boldsymbol{\Sigma}_a$ and $\boldsymbol{\Sigma}_i$, i.e., $\boldsymbol{\omega}_i = \boldsymbol{\omega}_a + \boldsymbol{\omega}_{i,a}$, with $\boldsymbol{\omega}_{i,a}$ denoting the relative angular velocity between the two frames, it holds:

$$\dot{\boldsymbol{\phi}}_i = \boldsymbol{T}^{-1}(\boldsymbol{\phi}_i)\left[\boldsymbol{T}(\boldsymbol{\phi}_a)\dot{\boldsymbol{\phi}}_a + \boldsymbol{T}(\boldsymbol{\phi}_{i,a})\dot{\boldsymbol{\phi}}_{i,a}\right], \tag{2.48}$$

where $\dot{\boldsymbol{\phi}}_{i,a}$ can be retrieved through the corresponding system of (2.43) for the Euler-angle derivative.

Considering the presence of a rigid body, (2.48) simplifies into

$$\dot{\boldsymbol{\phi}}_i = \boldsymbol{T}^{-1}(\boldsymbol{\phi}_i)\boldsymbol{T}(\boldsymbol{\phi}_a)\dot{\boldsymbol{\phi}}_a, \tag{2.49}$$

stating the equivalence of the angular velocities of two reference frames attached to a rigid body: $\boldsymbol{\omega}_i = \boldsymbol{\omega}_a$.

Finally, let us retrieve the corresponding expression of the reference linear and angular acceleration of the i-th end-effector. Recalling the propriety of the skew-symmetric operator $\boldsymbol{S}(\cdot)$, $\boldsymbol{S}(\boldsymbol{a})\boldsymbol{b} = -\boldsymbol{S}(\boldsymbol{b})\boldsymbol{a}$, with $\boldsymbol{a}$, $\boldsymbol{b} \in \mathcal{R}^3$, the derivative of (2.45) is:

$$\begin{aligned}
\ddot{\hat{\boldsymbol{p}}}_i =& \ddot{\hat{\boldsymbol{p}}}_a + \boldsymbol{S}(\dot{\hat{\boldsymbol{\omega}}}_a)\boldsymbol{R}(\hat{\boldsymbol{\phi}}_a)\hat{\boldsymbol{r}}_i^a + \boldsymbol{S}(\hat{\boldsymbol{\omega}}_a)\boldsymbol{S}(\hat{\boldsymbol{\omega}}_a)\boldsymbol{R}(\hat{\boldsymbol{\phi}}_a)\hat{\boldsymbol{r}}_i^a + 2\boldsymbol{S}(\hat{\boldsymbol{\omega}}_a)\boldsymbol{R}(\hat{\boldsymbol{\phi}}_a)\dot{\hat{\boldsymbol{r}}}_i^a + \boldsymbol{R}(\hat{\boldsymbol{\phi}}_a)\ddot{\hat{\boldsymbol{r}}}_i^a = \\
=& \ddot{\hat{\boldsymbol{p}}}_a - \boldsymbol{S}(\boldsymbol{R}(\hat{\boldsymbol{\phi}}_a)\hat{\boldsymbol{r}}_i^a)\dot{\hat{\boldsymbol{\omega}}}_a - \boldsymbol{S}(\hat{\boldsymbol{\omega}}_a)\boldsymbol{S}(\boldsymbol{R}(\hat{\boldsymbol{\phi}}_a)\hat{\boldsymbol{r}}_i^a)\hat{\boldsymbol{\omega}}_a - 2\boldsymbol{S}(\boldsymbol{R}(\hat{\boldsymbol{\phi}}_a)\dot{\hat{\boldsymbol{r}}}_i^a)\hat{\boldsymbol{\omega}}_a + \boldsymbol{R}(\hat{\boldsymbol{\phi}}_a)\ddot{\hat{\boldsymbol{r}}}_i^a = \\
=& \ddot{\hat{\boldsymbol{p}}}_a - \boldsymbol{S}(\boldsymbol{R}(\hat{\boldsymbol{\phi}}_a)\hat{\boldsymbol{r}}_i^a)\left[\dot{\boldsymbol{T}}(\boldsymbol{\phi}_a)\dot{\boldsymbol{\phi}}_a + \boldsymbol{T}(\boldsymbol{\phi}_a)\ddot{\boldsymbol{\phi}}_a\right] + \\
&- \boldsymbol{S}(\boldsymbol{T}(\boldsymbol{\phi}_a)\dot{\boldsymbol{\phi}}_a)\boldsymbol{S}(\boldsymbol{R}(\hat{\boldsymbol{\phi}}_a)\hat{\boldsymbol{r}}_i^a)\boldsymbol{T}(\boldsymbol{\phi}_a)\dot{\boldsymbol{\phi}}_a - 2\boldsymbol{S}(\boldsymbol{R}(\hat{\boldsymbol{\phi}}_a)\dot{\hat{\boldsymbol{r}}}_i^a)\boldsymbol{T}(\boldsymbol{\phi}_a)\dot{\boldsymbol{\phi}}_a + \boldsymbol{R}(\hat{\boldsymbol{\phi}}_a)\ddot{\hat{\boldsymbol{r}}}_i^a.
\end{aligned} \tag{2.50}$$

As concerns the orientation, calculating the time derivative of $\boldsymbol{\omega}_i = \boldsymbol{\omega}_a + \boldsymbol{\omega}_{i,a}$, it follows:

$$\ddot{\boldsymbol{\phi}}_i = \boldsymbol{T}^{-1}(\boldsymbol{\phi}_i)\left[\dot{\boldsymbol{T}}(\boldsymbol{\phi}_a)\dot{\boldsymbol{\phi}}_a + \boldsymbol{T}(\boldsymbol{\phi}_a)\ddot{\boldsymbol{\phi}}_a + \dot{\boldsymbol{T}}(\boldsymbol{\phi}_{i,a})\dot{\boldsymbol{\phi}}_{i,a} + \boldsymbol{T}(\boldsymbol{\phi}_{i,a})\ddot{\boldsymbol{\phi}}_{i,a} - \dot{\boldsymbol{T}}(\boldsymbol{\phi}_i)\dot{\boldsymbol{\phi}}_i\right]. \tag{2.51}$$

In the presence of a rigid body, (2.50) and (2.51) simplify respectively in:

$$\ddot{\hat{\boldsymbol{p}}}_i = \ddot{\hat{\boldsymbol{p}}}_a - \boldsymbol{S}(\boldsymbol{R}(\hat{\boldsymbol{\phi}}_a)\hat{\boldsymbol{r}}_i^a)\left[\dot{\boldsymbol{T}}(\boldsymbol{\phi}_a)\dot{\boldsymbol{\phi}}_a + \boldsymbol{T}(\boldsymbol{\phi}_a)\ddot{\boldsymbol{\phi}}_a\right] + \\ - \boldsymbol{S}(\boldsymbol{T}(\boldsymbol{\phi}_a)\dot{\boldsymbol{\phi}}_a)\boldsymbol{S}(\boldsymbol{R}(\hat{\boldsymbol{\phi}}_a)\hat{\boldsymbol{r}}_i^a)\boldsymbol{T}(\boldsymbol{\phi}_a)\dot{\boldsymbol{\phi}}_a \quad (2.52)$$

and

$$\ddot{\boldsymbol{\phi}}_i = \boldsymbol{T}^{-1}(\boldsymbol{\phi}_i)\left[\dot{\boldsymbol{T}}(\boldsymbol{\phi}_a)\dot{\boldsymbol{\phi}}_a + \boldsymbol{T}(\boldsymbol{\phi}_a)\ddot{\boldsymbol{\phi}}_a - \dot{\boldsymbol{T}}(\boldsymbol{\phi}_i)\dot{\boldsymbol{\phi}}_i\right]. \quad (2.53)$$

Thus, the closed-chain constraints hold in the presence of rigid object grasped by the robotic team according to the following relations (expressed with respect to the i-robot):

$$\begin{cases} \boldsymbol{p}_{i,d} = \boldsymbol{p}_{a,d} + \boldsymbol{r}_{i,d} \\ \dot{\boldsymbol{p}}_{i,d} = \dot{\boldsymbol{p}}_{a,d} - \boldsymbol{S}(\boldsymbol{r}_{i,d})\boldsymbol{T}(\boldsymbol{\phi}_{a,d})\dot{\boldsymbol{\phi}}_{a,d} \\ \ddot{\boldsymbol{p}}_{i,d} = \ddot{\boldsymbol{p}}_{a,d} - \boldsymbol{S}(\boldsymbol{r}_{i,d})\dot{\boldsymbol{T}}(\boldsymbol{\phi}_{a,d})\dot{\boldsymbol{\phi}}_{a,d} + \\ \qquad -\boldsymbol{S}(\boldsymbol{T}(\boldsymbol{\phi}_{a,d})\dot{\boldsymbol{\phi}}_{a,d})\boldsymbol{S}(\boldsymbol{r}_{i,d})\boldsymbol{T}(\boldsymbol{\phi}_{a,d})\dot{\boldsymbol{\phi}}_{a,d} + \\ \qquad -\boldsymbol{S}(\boldsymbol{r}_{i,d})\boldsymbol{T}(\boldsymbol{\phi}_{a,d})\ddot{\boldsymbol{\phi}}_{a,d} \\ \boldsymbol{\phi}_{i,d} = \boldsymbol{\phi}(\boldsymbol{R}_{a,d}\boldsymbol{R}_{i,d}^a) \\ \dot{\boldsymbol{\phi}}_{i,d} = \boldsymbol{T}^{-1}(\boldsymbol{\phi}_{i,d})\boldsymbol{T}(\boldsymbol{\phi}_{a,d})\dot{\boldsymbol{\phi}}_{a,d} \\ \ddot{\boldsymbol{\phi}}_{i,d} = \boldsymbol{T}^{-1}(\boldsymbol{\phi}_{i,d})\dot{\boldsymbol{T}}(\boldsymbol{\phi}_{a,d})\dot{\boldsymbol{\phi}}_{a,d} + \\ \qquad \boldsymbol{T}^{-1}(\boldsymbol{\phi}_{i,d})(\boldsymbol{T}(\boldsymbol{\phi}_{a,d})\ddot{\boldsymbol{\phi}}_{a,d} - \dot{\boldsymbol{T}}(\boldsymbol{\phi}_{i,d})\dot{\boldsymbol{\phi}}_{i,d}) \end{cases} \quad (2.54)$$

The proposed set of equations represents a mapping from the task-oriented representation, in terms of the absolute and relative task variables, to another representation, with reference to each robot end-effector, both in the presence of a common manipulated object and in the absence of such an object.

2.6 Distributed cooperative object transportation

Consider a cooperative object transportation task performed by a team of N manipulators.

According to (2.3), the absolute frame corresponds to the barycenter of the robot end-effector poses, $\boldsymbol{x}_i$. Consequently, $\boldsymbol{\Sigma}_a$ does not necessarily coincide with the object's center of mass. In such a case, without any loss of generality, the object frame $\boldsymbol{\Sigma}_o$ can

be placed to coincide with $\boldsymbol{\Sigma}_a$, while introducing appropriate terms to compensate for motion discrepancies and to ensure consistent force transformation.

2.6.1 Object dynamics

Let $\boldsymbol{h}_{env}$ indicate the vector of wrenches (or *generalized forces*) exerted by the object on the environment, and $\boldsymbol{h}_o$ the wrenches applied by the manipulators on the object. Then, the object dynamics can be expressed as:

$$\boldsymbol{M}_o(\boldsymbol{x}_o)\ddot{\boldsymbol{x}}_o + \boldsymbol{C}_o(\boldsymbol{x}_o, \dot{\boldsymbol{x}}_o)\dot{\boldsymbol{x}}_o + \boldsymbol{g}_o = \boldsymbol{\gamma}_o - \boldsymbol{\gamma}_{env}\,, \tag{2.55}$$

where $\boldsymbol{x}_o = [\boldsymbol{p}_o^{\mathrm{T}},\ \boldsymbol{\phi}_o^{\mathrm{T}}]^{\mathrm{T}}$ $(\dot{\boldsymbol{x}}_o, \ddot{\boldsymbol{x}}_o)$ is the pose (generalized velocity, generalized acceleration) of the reference frame attached to the object, with $\boldsymbol{p}_o$ and $\boldsymbol{\phi}_o$ denoting the position and the Euler-angle orientation. Here, $\boldsymbol{M}_o$ is the inertia matrix, $\boldsymbol{C}_o$ the Coriolis/centripetal matrix, and $\boldsymbol{g}_o$ the gravity term. The wrenches are mapped as

$$\boldsymbol{\gamma}_{env} = \boldsymbol{T}_A^{\mathrm{T}}(\boldsymbol{\phi}_o)\boldsymbol{h}_{env}, \quad \boldsymbol{\gamma}_o = \boldsymbol{T}_A^{\mathrm{T}}(\boldsymbol{\phi}_o)\boldsymbol{h}_o, \tag{2.56}$$

via the matrix

$$\boldsymbol{T}_A(\boldsymbol{\phi}_o) = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{O}_3 \\ \boldsymbol{O}_3 & \boldsymbol{T}(\boldsymbol{\phi}_o) \end{bmatrix}. \tag{2.57}$$

The wrench $\boldsymbol{h}_o$ is computed through the *grasp matrix* $\boldsymbol{G}$ [57]:

$$\boldsymbol{h}_o = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{O}_3 & \dots & \boldsymbol{I}_3 & \boldsymbol{O}_3 \\ \boldsymbol{S}(\boldsymbol{r}_1) & \boldsymbol{I}_3 & \dots & \boldsymbol{S}(\boldsymbol{r}_N) & \boldsymbol{I}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{h}_1 \\ \vdots \\ \boldsymbol{h}_N \end{bmatrix} = \boldsymbol{G}\boldsymbol{h}_e, \tag{2.58}$$

with

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{G}_1 & \boldsymbol{G}_2 & \dots & \boldsymbol{G}_N \end{bmatrix}, \quad \boldsymbol{G}_k = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{O}_3 \\ \boldsymbol{S}(\boldsymbol{r}_k) & \boldsymbol{I}_3 \end{bmatrix}, \tag{2.59}$$

where $\boldsymbol{r}_k = \boldsymbol{p}_k - \boldsymbol{p}_o$ is the vector from the object origin to the k-th end-effector. Thus, $\boldsymbol{h}_o$ represents the wrenches applied at the object center of mass, denominated as *external wrenches*, while $\boldsymbol{h}_e = [\boldsymbol{h}_1^{\mathrm{T}}\ \dots\ \boldsymbol{h}_N^{\mathrm{T}}]^{\mathrm{T}}$ collects the generalized forces exerted by each end effector.

Since $\boldsymbol{G}$ has full row rank, (2.58) admits the inverse solution [114]:

$$\boldsymbol{h}_e = \boldsymbol{G}^{\dagger}\boldsymbol{h}_o + \boldsymbol{V}\boldsymbol{h}_i = \boldsymbol{h}_E + \boldsymbol{h}_I, \tag{2.60}$$

where $\boldsymbol{G}^{\dagger}$ is a right pseudo-inverse of $\boldsymbol{G}$, $\boldsymbol{V} \in \mathbb{R}^{6N\times(6N-6)}$ is a full-column rank matrix which spans the null space of $\boldsymbol{G}$, and $\boldsymbol{h}_i \in \mathbb{R}^{6N-6}$ collects the independent wrenches that do not affect object motion, i.e., the *internal wrenches* [102]. These arise when manipulator forces lie in the null space of the grasp matrix and result from trajectory-tracking uncertainties or physical constraints. Although they do not affect object motion, they may induce undesired stresses and should be minimized.

Following [114] and [103], a possible choice of the $\boldsymbol{V}$ matrix, here adopted, is

$$\boldsymbol{V} = \begin{bmatrix} -\boldsymbol{V}_1 & \boldsymbol{O}_6 & \dots & \boldsymbol{O}_6 \\ \boldsymbol{O}_6 & -\boldsymbol{V}_2 & \dots & \boldsymbol{O}_6 \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{O}_6 & \boldsymbol{O}_6 & \dots & -\boldsymbol{V}_{N-1} \\ \boldsymbol{V}_N & \boldsymbol{V}_N & \dots & \boldsymbol{V}_N \end{bmatrix}, \quad \boldsymbol{V}_k = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{O}_3 \\ -\boldsymbol{S}(\boldsymbol{r}_k) & \boldsymbol{I}_3 \end{bmatrix}. \tag{2.61}$$

Accordingly, $\boldsymbol{h}_E$ denotes the external contribution balancing the object dynamics and the interaction with the environment,

$$\boldsymbol{h}_E = \begin{bmatrix} \boldsymbol{h}_{E_1}^{\mathrm{T}} & \dots & \boldsymbol{h}_{E_N}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} = \boldsymbol{G}^{\dagger}\boldsymbol{G}\boldsymbol{h}_e, \tag{2.62}$$

while $\boldsymbol{h}_I$ captures the robots' contribution to the internal wrench:

$$\boldsymbol{h}_I = \begin{bmatrix} \boldsymbol{h}_{I_1}^{\mathrm{T}} & \dots & \boldsymbol{h}_{I_N}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} = \boldsymbol{V}\boldsymbol{h}_i = \boldsymbol{V}\boldsymbol{V}^{\dagger}\boldsymbol{h}_e. \tag{2.63}$$

2.7 Distributed wrench estimation

In view of (2.58) and (2.63), the computation of internal and external wrenches requires knowledge of each robot's contribution exerted at its end-effector, namely the wrench collective vector, $\boldsymbol{h}_e$. In turn, this implies either the existence of a centralized unit that collects all generalized forces and computes the manipulators' control laws, or an all-to-all communication scheme among the robots. However, both solutions pose limitations in terms of scalability and robustness, as they entail an increasing computational burden with the team size.

An alternative strategy consists in interpreting each robot as the *leader* of its own wrench signal, with the remaining robots acting as *followers*. In contrast to the problem of a team of agents estimating a common global signal in a distributed manner,

addressed in Section 2.5, here each agent is in charge of estimating a set of $N-1$ exogenous signals, one for each other robot.

Thus, unlike the methods proposed in [34] and [115], this approach requires the communication graph among the robots to be strongly connected. In this framework, each agent estimates the wrenches exerted by the other $N-1$ robots by running a bank of $N-1$ consensus-based estimators, defined as

$$ {}^{i}\dot{\hat{\boldsymbol{h}}}_k(t) = k_0 \sum_{j \in \mathcal{N}_i} \left({}^{j}\hat{\boldsymbol{h}}_k(t) - {}^{i}\hat{\boldsymbol{h}}_k(t) \right), \tag{2.64} $$

where ${}^{i}\hat{\boldsymbol{h}}_k(t)$ denotes the estimate, computed by the i-th manipulator, of the wrench exerted on the object by the k-th manipulator ($k = 1, \ldots, N$), and k_0 is a positive scalar gain. Unlike the estimator in (2.24), the higher-order derivative term is not included, nor is the presence of a virtual leader, since this estimation scheme requires each manipulator (agent) to act as a leader for the generalized force signal measured at its own end-effector.

Then, omitting the time dependence for the sake of compactness, the dynamics of the estimation error, ${}^{i}\tilde{\boldsymbol{h}}_k$, can be derived straightforwardly:

$$ {}^{i}\dot{\tilde{\boldsymbol{h}}}_k = \dot{\boldsymbol{h}}_k - {}^{i}\dot{\hat{\boldsymbol{h}}}_k = \dot{\boldsymbol{h}}_k - k_0 \sum_{j \in \mathcal{N}_i} \left({}^{i}\tilde{\boldsymbol{h}}_k - {}^{j}\tilde{\boldsymbol{h}}_k \right) = \dot{\boldsymbol{h}}_k - k_0 N_i {}^{i}\tilde{\boldsymbol{h}}_k + k_0 \sum_{j \in \mathcal{N}_i} {}^{j}\tilde{\boldsymbol{h}}_k, \tag{2.65} $$

where N_i is the cardinality of the set of neighbors of the i-th robot. Let's define the $6(N-1) \times 1$ vector, $\hat{\boldsymbol{h}}_k = \left[{}^{1}\hat{\boldsymbol{h}}_k^{\mathrm{T}}, \ \ldots, \ {}^{k-1}\hat{\boldsymbol{h}}_k^{\mathrm{T}}, \ {}^{k+1}\hat{\boldsymbol{h}}_k^{\mathrm{T}}, \ \ldots, \ {}^{N}\hat{\boldsymbol{h}}_k^{\mathrm{T}} \right]^{\mathrm{T}}$, which collects the estimates of $\boldsymbol{h}_k$ provided by each robot but the k-th, and the corresponding collective estimation error, $\tilde{\boldsymbol{h}}_k$, as

$$ \tilde{\boldsymbol{h}}_k = (\mathbf{1}_{N-1} \otimes \boldsymbol{h}_k) - \hat{\boldsymbol{h}}_k. \tag{2.66} $$

The collective error dynamics $\dot{\tilde{\boldsymbol{h}}}_k$ is given by

$$ \dot{\tilde{\boldsymbol{h}}}_k = \mathbf{1}_{N-1} \otimes \dot{\boldsymbol{h}}_k + \boldsymbol{\Lambda}_k \tilde{\boldsymbol{h}}_k, \tag{2.67} $$

where $\boldsymbol{\Lambda}_k = -k_0(\bar{\boldsymbol{L}}_k \otimes \boldsymbol{I}_6)$. By virtue of Property 2, $-\bar{\boldsymbol{L}}_k$ is Hurwitz and, consequently, due to the Kronecker product properties, $\boldsymbol{\Lambda}_k$ is Hurwitz, as well. Therefore, the system in (2.67) is an exponentially stable system, with $\mathbf{1}_{N-1} \otimes \dot{\boldsymbol{h}}_k$ as input. Then, it is straightforward to prove that $\tilde{\boldsymbol{h}}_k$ is asymptotically convergent to zero in the presence of constant or slowly variant wrenches $\boldsymbol{h}_k$, while it is bounded in the presence of a bounded time derivative $\dot{\boldsymbol{h}}_k$.

Let's define as ${}^i\hat{\boldsymbol{h}}_e$ the i-th robot estimate of the end-effector wrench collective vector. According to its definition, it can be split in two contributions, distinguishing the estimation-derived one from its own wrench signal (sensed at its end-effector or, eventually, estimated starting from the force/torque measures at each joint):

$$\begin{aligned} {}^i\hat{\boldsymbol{h}}_e &= \left[{}^i\hat{\boldsymbol{h}}_1^{\mathrm{T}},\ \ldots,\ {}^i\hat{\boldsymbol{h}}_{i-1}^{\mathrm{T}},\quad \boldsymbol{h}_i^{\mathrm{T}},\quad {}^i\hat{\boldsymbol{h}}_{i+1}^{\mathrm{T}},\ \ldots,\ {}^i\hat{\boldsymbol{h}}_N^{\mathrm{T}}\right]^{\mathrm{T}} = \\ &= \left[{}^i\hat{\boldsymbol{h}}_1^{\mathrm{T}},\ \ldots,\ {}^i\hat{\boldsymbol{h}}_{i-1}^{\mathrm{T}},\quad \boldsymbol{0}_6^{\mathrm{T}},\quad {}^i\hat{\boldsymbol{h}}_{i+1}^{\mathrm{T}},\ \ldots,\ {}^i\hat{\boldsymbol{h}}_N^{\mathrm{T}}\right]^{\mathrm{T}} + \\ &+ \left[\boldsymbol{0}_6^{\mathrm{T}},\ \ldots,\ \boldsymbol{0}_6^{\mathrm{T}},\quad \boldsymbol{h}_i^{\mathrm{T}},\quad \boldsymbol{0}_6^{\mathrm{T}},\ \ldots,\ \boldsymbol{0}_6^{\mathrm{T}}\right]^{\mathrm{T}} = {}^i\hat{\boldsymbol{h}}_{j\neq i} + \bar{\boldsymbol{h}}_i, \end{aligned} \tag{2.68}$$

with $\bar{\boldsymbol{h}}_i = \left[\boldsymbol{0}_6^{\mathrm{T}},\ \ldots,\ \boldsymbol{0}_6^{\mathrm{T}},\quad \boldsymbol{h}_i^{\mathrm{T}},\quad \boldsymbol{0}_6^{\mathrm{T}},\ \ldots,\ \boldsymbol{0}_6^{\mathrm{T}}\right]^{\mathrm{T}}$.

First, the i-th robot estimate the external wrenches exerted on the object ${}^i\hat{\boldsymbol{h}}_o$, by resorting to (2.58):

$${}^i\hat{\boldsymbol{h}}_o = \boldsymbol{G}\ {}^i\hat{\boldsymbol{h}}_e. \tag{2.69}$$

Second, it can evaluate its own contribution to the internal wrench defined in (2.63) as

$${}^i\hat{\boldsymbol{h}}_{I_i} = \boldsymbol{\Gamma}_i \boldsymbol{V}\boldsymbol{V}^{\dagger}\ {}^i\hat{\boldsymbol{h}}_e, \tag{2.70}$$

by extracting the components of interest from ${}^i\hat{\boldsymbol{h}}_{I_i}$, by means of the matrix $\boldsymbol{\Gamma}_i$, defined in 2.5.3.

2.8 Distributed compliance control

As a first step, consider the goal of reducing the internal wrenches exerted on the object.

Based on the estimates computed in Section 2.7, a distributed control strategy designed to limit the internal wrenches on the object has been developed. The proposed architecture scheme is shown in Fig. 2.2.

Focusing on the i-th agent, $i = 1, \ldots, N$, the proposed distributed scheme is a two-level controller:

1. **Local compliance control**: the upper level is a local compliance control providing the reference generalized velocity and pose of the i-th end-effector as outputs. Thus, a local compliance control scheme is run by the i-th manipulator to adjust its end-effector compliance, thereby reducing its internal wrench contribution on the object.

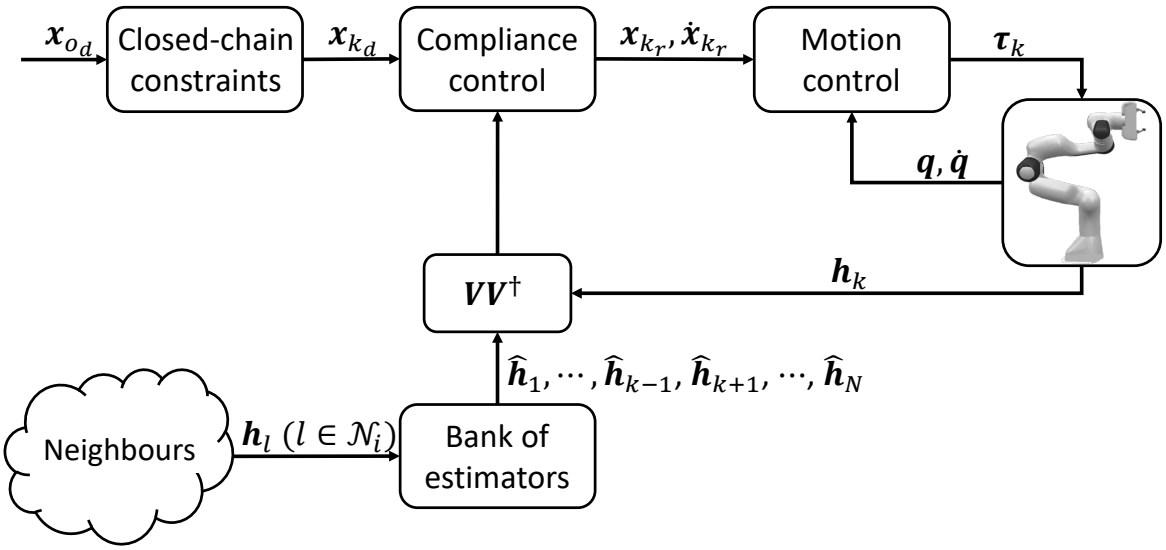


Figure 2.2 Block scheme of the overall architecture of the compliance controller.

2. **Motion control**: the lower level is a motion controller aimed at tracking the reference trajectory $(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i)$ coming from the upper level. In principle, any motion controller can be adopted at the lower level.

This framework relies on a distributed wrench estimation scheme, as the one proposed in Section 2.7: the i-th robot is assumed to be provided with its own contact wrench, $\boldsymbol{h}_k$, while the wrenches exerted on the object by the other robots, ${}^i\hat{\boldsymbol{h}}_{j\neq i}$, are estimated via a bank of consensus-based estimators (2.64). Since the estimator runs within a communication-limited network, this latter is modeled as a strongly connected directed graph. Then, the signals are rearranged into the wrench collective vector ${}^i\hat{\boldsymbol{h}}_e$, according to (2.68).

In particular, the local compliance controller receives as input the desired trajectory of the i-th end-effector provided by the closed-chain constraints (2.54) and the estimate $\hat{\boldsymbol{h}}_{I_i}$ of the i-th robot contribute to the internal wrenches (2.70). The output is a reference trajectory, in terms of end-effector pose and generalized velocity, computed as

$$\dot{\boldsymbol{x}}_{i,r} = \dot{\boldsymbol{x}}_{i,d} + \boldsymbol{K}_{D_i}^{-1}\left[\boldsymbol{K}_{P_i}(\boldsymbol{x}_{i,d} - \boldsymbol{x}_{i,r}) - \hat{\boldsymbol{h}}_{I_i}\right], \tag{2.71}$$

where $\boldsymbol{K}_{D_i}$ and $\boldsymbol{K}_{P_i}$ are positive definite matrices of gains, representing, respectively, the virtual damping and stiffness, imposed to the i-th end-effector motion.

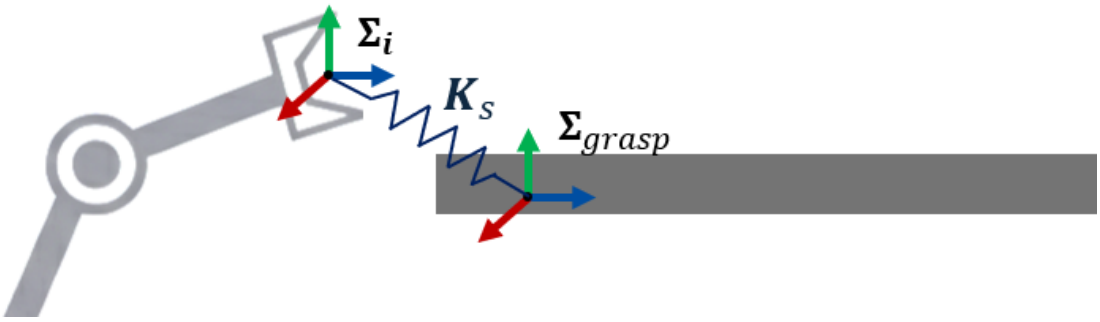


Figure 2.3 Elastic model of the contact between the i-th end-effector and the object.

At steady state, it holds

$$\lim_{t\to\infty}(\boldsymbol{x}_{i,d}(t)-\boldsymbol{x}_{i,r}(t))=\boldsymbol{x}_{i,d,\infty}-\boldsymbol{x}_{i,r,\infty}=\boldsymbol{K}_{P_i}^{-1}\hat{\boldsymbol{h}}_{I_i,\infty}, \tag{2.72}$$

as a consequence, the tuning of the gains in (2.71) is a trade-off between residual internal forces on the object due to robot i and the pose compliance.

2.8.1 Stability analysis

Let's assume that i-th agent is provided with the desired object trajectory, $\boldsymbol{x}_{o,d}, \dot{\boldsymbol{x}}_{o,d}$; this working assumption will be further removed in Section 2.10. Then, the proposed approach still leans on the local estimate of the other robots' wrenches computed by i-th agent. Thus, the stability of the overall control framework must be assessed. In the following, the stability of the proposed control scheme is focused solely on the forces (position), although analogous arguments can be adopted for the moments (orientation). It is ensured under the following assumptions:

Assumption 7. *The object planned trajectory and its time derivative, $\boldsymbol{p}_{a,d}$ and $\dot{\boldsymbol{p}}_{a,d}$, are norm bounded, i.e., there exist suitable positive constants $\bar{p}_o$ and $\dot{\bar{p}}_o$ such that $\forall t$*

$$\left\|\boldsymbol{p}_{a,d}(t)\right\|\leq\bar{p}_o \qquad \left\|\dot{\boldsymbol{p}}_{a,d}(t)\right\|\leq\dot{\bar{p}}_o \ . \tag{2.73}$$

Based on (2.54), (2.73) *implies that the desired trajectories of each end-effectors and their time derivatives are bounded as well.*

Assumption 8. *The connection between the object and i-th end-effector is modeled by introducing a grasp reference frame attached to the object linked to the i-th end-effector frame by a 6-DOFs spring system, as depicted in Fig 2.3. Thus, the force exerted by*

the i-th manipulator ($i = 1, \dots, N$) on the object is given by

$$\boldsymbol{f}_i = \boldsymbol{K}_S(\boldsymbol{p}_i - \boldsymbol{p}_{eq,i}), \tag{2.74}$$

where $\boldsymbol{p}_{eq,i}$ represents the spring equilibrium position, $\boldsymbol{p}_i$ is the actual end-effector position and $\boldsymbol{K}_S$ is a positive definite matrix representing the object stiffness.

Assumption 9. *Let us assume the perfect tracking of the low-level control, i.e., the pose $\boldsymbol{p}_i$ coincides with the reference pose $\boldsymbol{p}_{i,r}$.*

This is a reasonable assumption, due to the slow trajectories usually adopted for cooperative transportation tasks.

Therefore, the tracking position error $\boldsymbol{e}_i = \boldsymbol{p}_{i,d} - \boldsymbol{p}_i$ is characterized by the following dynamics

$$\dot{\boldsymbol{e}}_i = -\boldsymbol{K}_i \boldsymbol{e}_i + \boldsymbol{K}_{D_{i,p}}^{-1} \hat{\boldsymbol{f}}_{I_i}, \tag{2.75}$$

where the matrix $\boldsymbol{K}_i = \boldsymbol{K}_{D_{i,p}}^{-1} \boldsymbol{K}_{P_{i,p}}$ with $\boldsymbol{K}_{D_{i,p}}$ and $\boldsymbol{K}_{P_{i,p}}$ the positional parts of $\boldsymbol{K}_{D_i}$ and $\boldsymbol{K}_{P_i}$, respectively, and $\hat{\boldsymbol{f}}_{I_i}$ is the vector of the estimated contribute to the internal forces exerted by the i-th robot extracted from $\hat{\boldsymbol{h}}_{I_i}$ in (2.70).

Adopting the expression of the matrix $\boldsymbol{V}$ proposed in (2.61), let us consider the matrix $\boldsymbol{V}_p \in \mathbb{R}^{3N \times 6(N-1)}$ extracted from $\boldsymbol{V}$ selecting only the first three rows of each $\boldsymbol{V}_j$, namely

$$\boldsymbol{V}_p = \begin{bmatrix} -\boldsymbol{I}_3 & \boldsymbol{O}_3 & \boldsymbol{O}_3 & \boldsymbol{O}_3 & \dots & \boldsymbol{O}_3 & \boldsymbol{O}_3 \\ \boldsymbol{O}_3 & \boldsymbol{O}_3 & -\boldsymbol{I}_3 & \boldsymbol{O}_3 & \dots & \boldsymbol{O}_3 & \boldsymbol{O}_3 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{O}_3 & \boldsymbol{O}_3 & \boldsymbol{O}_3 & \boldsymbol{O}_3 & \dots & -\boldsymbol{I}_3 & \boldsymbol{O}_3 \\ \boldsymbol{I}_3 & \boldsymbol{O}_3 & \boldsymbol{I}_3 & \boldsymbol{O}_3 & \dots & \boldsymbol{I}_3 & \boldsymbol{O}_3 \end{bmatrix}. \tag{2.76}$$

As reported in A6, the following lemma can be proven by induction:

Lemma 2. *In virtue of the choice of $\boldsymbol{V}$, the estimated force ${}^i\hat{\boldsymbol{f}}_{I_i}$ can be expressed as*

$${}^i\hat{\boldsymbol{f}}_{I_i} = \frac{N-1}{N} \boldsymbol{f}_i - \frac{1}{N} \sum_{j \neq i} {}^i\hat{\boldsymbol{f}}_j. \tag{2.77}$$

Hence, the dynamics of the overall system for N agents has the following expression:

$$\dot{\boldsymbol{e}} = -\boldsymbol{A}_e \boldsymbol{e} + \boldsymbol{B}_e(\boldsymbol{p}_d - \boldsymbol{p}_{eq}) + \boldsymbol{C}_e \hat{\boldsymbol{f}}_e, \tag{2.78}$$

where $\boldsymbol{p}_d$ and $\boldsymbol{p}_{eq}$ are respectively the collective vectors of the end-effector desired position trajectories and the spring equilibrium positions,

$$\boldsymbol{A}_e = \left(\boldsymbol{K} - \frac{N-1}{N} \boldsymbol{K}_{D_p}^{-1} (\boldsymbol{I}_N \otimes \boldsymbol{K}_S) \right),$$

$$\boldsymbol{B}_e = \frac{N-1}{N} \boldsymbol{K}_{D_p}^{-1} (\boldsymbol{I}_N \otimes \boldsymbol{K}_S),$$

$$\boldsymbol{C}_e = \boldsymbol{K}_{D_p}^{-1} \left(\boldsymbol{V}_p \boldsymbol{V}_p^{\dagger} - \mathrm{diag}(\boldsymbol{V}_p \boldsymbol{V}_p^{\dagger}) \right),$$

$$\hat{\boldsymbol{f}}_e = \left[{}^1\hat{\boldsymbol{f}}_e^{\mathrm{T}} \quad {}^2\hat{\boldsymbol{f}}_e^{\mathrm{T}} \quad \dots \quad {}^N\hat{\boldsymbol{f}}_e^{\mathrm{T}} \right]^{\mathrm{T}},$$

with ${}^i\hat{\boldsymbol{f}}_e$ the (3×1) vector of estimated forces extracted from ${}^i\hat{\boldsymbol{h}}_e$ defined in (2.68) and $\boldsymbol{K}$, $\boldsymbol{K}_D^{-1}$ and $\boldsymbol{K}_S$ are, respectively:

$$\boldsymbol{K}_S = \mathrm{diag}\{\boldsymbol{K}_{S_1}, \dots, \boldsymbol{K}_{S_N}\},$$

$$\boldsymbol{K} = \mathrm{diag}\{\boldsymbol{K}_1, \dots, \boldsymbol{K}_N\},$$

$$\boldsymbol{K}_{D_p}^{-1} = \mathrm{diag}\{\boldsymbol{K}_{D_{1,p}}^{-1}, \dots, \boldsymbol{K}_{D_{N,p}}^{-1}\},$$

with $\mathrm{diag}\{\cdot\}$ denoting a block diagonal matrix of dimensions $3N \times 3N$.

As shown in [116], its stability properties are stated in the following theorem, whose proof is reported in Appendix A4:

Theorem 4. *If $\boldsymbol{A}_e$ is a positive definite matrix, then the collective position tracking error $\boldsymbol{e}(t)$ is bounded $\forall t$, for any initial condition $\boldsymbol{e}(0)$.*

Under the assumptions of Theorem 4, the following corollary can be stated, whose proof is reported in Appendix A5.

Corollary 2. *If the collective position tracking error $\boldsymbol{e}(t)$ is bounded, also the force estimation error $\widetilde{\boldsymbol{f}}_k$, extracted from $\widetilde{\boldsymbol{h}}_k$ in (2.66), is bounded $\forall t$, for any $k \in \{1, 2, \dots, N\}$.*

2.9 Distributed internal admittance control

An internal admittance controller has been developed to generalize the compliance control scheme proposed in Section 2.8. The desired object trajectory is supposed to be provided to each robot. This controller is made up with a second-order dynamics, comprising both forces and moments exerted on the object. Similarly to the compliance

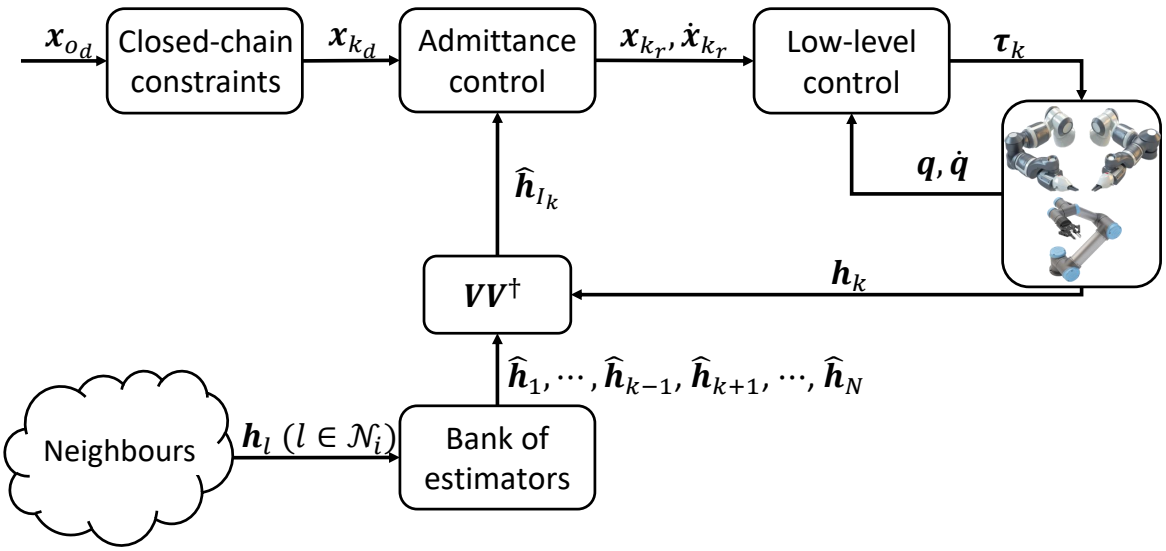


Figure 2.4 Block scheme of the overall architecture of the internal admittance controller.

controller, this framework is designed to reduce internal wrenches on the object in order to prevent plastic deformations occurring and to guarantee the integrity of the grasped object upon cooperative tasks.

The scheme is a two-level distributed controller, depicted in Fig. 2.4:

- **Internal impedance controller**: the upper level is a local admittance control fed in by the desired trajectory of the i-th end-effector coming from the closed-chain constraints (2.54) and by the estimate $\hat{\boldsymbol{h}}_{I_i}$ of the i-th robot contribute to the internal wrenches (2.63). It provides the reference generalized acceleration, velocity, and pose of the i-th end-effector as outputs.
- **Motion controller**: the lower level is a motion controller aimed at tracking such a reference trajectory. From a theoretical point of view, any motion controller can be used at the lower level.

More in detail, the output of the admittance controller is a reference trajectory, in terms of end-effector pose and generalized velocity and acceleration, computed as

$$\ddot{\boldsymbol{x}}_{i,r} = \ddot{\boldsymbol{x}}_{i,d} + \boldsymbol{M}_i^{-1}\Big[\boldsymbol{K}_{D_i}(\dot{\boldsymbol{x}}_{i,d} - \dot{\boldsymbol{x}}_{i,r}) + \boldsymbol{K}_{P_i}(\boldsymbol{x}_{i,d} - \boldsymbol{x}_{i,r}) + \boldsymbol{\Gamma}_i \boldsymbol{V}\boldsymbol{V}^\dagger \left({}^i\hat{\boldsymbol{h}}_{j\neq i} + \bar{\boldsymbol{h}}_i\right)\Big], \quad (2.79)$$

where $\boldsymbol{K}_{D_i}$, $\boldsymbol{K}_{P_i}$ and $\boldsymbol{M}_i$ are (6×6) positive definite matrices of gains, representing, respectively, a virtual damping, stiffness and inertia imposed to the i-th end-effector motion. Note that, according to (2.68), the two contributions of the wrench collective vector, $\bar{\boldsymbol{h}}_i$ and ${}^i\hat{\boldsymbol{h}}_{j\neq i}$, coming from its measurements and its bank of estimators, respectively, have been distinguished to suitably account for their different natures in the following.

2.9.1 Stability analysis

Based on the results in Section 2.8, the stability of the proposed control scheme is ensured under Assumptions 8 and 9. Unlike the previous section, wrenches are considered in lieu of forces, in the model of the contact:

$$\boldsymbol{h}_i = \boldsymbol{K}_{S_i}(\boldsymbol{x}_i - \boldsymbol{x}_{eq,i}), \tag{2.80}$$

where $\boldsymbol{x}_{eq,i}$ represents the spring equilibrium pose, $\boldsymbol{x}_i$ is the actual end-effector pose and $\boldsymbol{K}_{S_i} \in \mathbb{R}^{6\times 6}$ is a positive definite matrix representing the object stiffness.

Folding (2.80) into the dynamics in (2.79), with simple mathematical steps, the tracking pose error $\boldsymbol{e}_i = \boldsymbol{x}_{i,d} - \boldsymbol{x}_i$ is characterized by the following dynamics:

$$\boldsymbol{M}_i\ddot{\boldsymbol{e}}_i + \boldsymbol{K}_{D_i}\dot{\boldsymbol{e}}_i + (\boldsymbol{K}_{P_i} - \boldsymbol{\Gamma}_i\boldsymbol{V}\boldsymbol{V}^\dagger\,\boldsymbol{K}_{S_i})\boldsymbol{e}_i = \boldsymbol{\Gamma}_i\boldsymbol{V}\boldsymbol{V}^\dagger\,\left[{}^i\hat{\boldsymbol{h}}_{j\neq i} + \boldsymbol{K}_{S_i}(\boldsymbol{x}_{i,d} - \boldsymbol{x}_{eq,i})\right]. \tag{2.81}$$

For the sake of compactness, the following notation will be adopted: $\bar{\boldsymbol{K}}_{P_i} = \boldsymbol{\Gamma}_i\boldsymbol{V}\boldsymbol{V}^\dagger\boldsymbol{K}_{S_i}$ and the input $\boldsymbol{d}_i = \boldsymbol{\Gamma}_i\boldsymbol{V}\boldsymbol{V}^\dagger\,\left[{}^i\hat{\boldsymbol{h}}_{j\neq i} + \boldsymbol{K}_{S_i}(\boldsymbol{x}_{i,d} - \boldsymbol{x}_{eq,i})\right]$. Moreover, hereafter the subscript i will be omitted. Therefore, the tracking pose error of the generic agent has the following dynamics

$$\boldsymbol{M}\ddot{\boldsymbol{e}} + \boldsymbol{K}_D\dot{\boldsymbol{e}} + (\boldsymbol{K}_P - \bar{\boldsymbol{K}}_P)\boldsymbol{e} = \boldsymbol{d}. \tag{2.82}$$

Defining $\boldsymbol{\zeta} = \left[\boldsymbol{e}^{\mathrm{T}},\ \ \dot{\boldsymbol{e}}^{\mathrm{T}}\right]^{\mathrm{T}}$, the dynamics in (2.82) can be reformulated as a linear system

$$\dot{\boldsymbol{\zeta}} = \begin{bmatrix} \boldsymbol{O}_3 & \boldsymbol{I}_3 \\ -\boldsymbol{M}^{-1}\left(\boldsymbol{K}_P - \bar{\boldsymbol{K}}_P\right) & -\boldsymbol{M}^{-1}\boldsymbol{K}_D \end{bmatrix}\boldsymbol{\zeta} + \begin{bmatrix} 0 \\ \boldsymbol{M}^{-1}\boldsymbol{d} \end{bmatrix}, \tag{2.83}$$

whose stability properties are stated by the following theorem.

Theorem 5. *The system* (2.83) *is input-to-state stable (ISS) provided that the positive definite matrices* $\boldsymbol{M}$, $\boldsymbol{K}_D$ *and* $\boldsymbol{K}_P$ *satisfy the following conditions:*

C.1 $\lambda_m(\boldsymbol{K}_D) > \max\left\{\alpha\dfrac{\lambda_M(\boldsymbol{M})^2}{\lambda_m(\boldsymbol{M})^2} - \dfrac{1}{\alpha}\lambda_m(\boldsymbol{K}_p), \alpha\lambda_M(\boldsymbol{M})\right\},$

C.2 $\lambda_m(\boldsymbol{K}_P) > \dfrac{1}{4\alpha}\dfrac{\left\|\bar{\boldsymbol{K}}_P\right\|}{\lambda_m(\boldsymbol{K}_D) - \alpha\lambda_M(\boldsymbol{M})} + \left\|\bar{\boldsymbol{K}}_P\right\|,$

where $\lambda_M(\cdot)$ *and* $\lambda_m(\cdot)$ *denote the maximum and minimum eigenvalue of a matrix, respectively, and* α *is a positive real constant.*

The proof of the theorem is reported in Appendix A7.

2.10 Distributed external and internal admittance control

Finally, a two-stage distributed admittance control framework for cooperative manipulation with N manipulators has been developed. The method regulates internal wrenches while distributing external ones across the team. The overall architecture is illustrated in Fig. 2.5.

According to Section 2.5.3, the desired object trajectory is estimated by each robot. Subsequently, this estimate is provided to the external impedance filter together with the estimates of the external wrenches obtained from (2.69). A round-robin communication protocol is employed: each robot measures its own wrench and estimates those of its teammates. These estimates are then used to compute two compliant trajectories: one for the task object and another for the robot's motion relative to its peers, thereby reducing stress on the manipulated object.

The proposed strategy relies, in fact, on the two-stage admittance-based framework designed to minimize the internal stresses acting on the manipulated object, while ensuring bounded contact forces with the environment in a fully distributed manner. The control architecture of the i-th robot ($i = 1, \ldots, N$) is organized into the following hierarchical levels:

- **Outer external impedance loop**: it computes the object reference trajectory, $\boldsymbol{x}_{o_r}$, $\dot{\boldsymbol{x}}_{o_r}$, $\ddot{\boldsymbol{x}}_{o_r}$, while distributing the external wrench applied to the object;
- **Inner internal impedance loop**: it generates the end-effector reference trajectory of the i-th robot, $\boldsymbol{x}_{i_r}$, $\dot{\boldsymbol{x}}_{i_r}$, $\ddot{\boldsymbol{x}}_{i_r}$, while limiting its contribution to the internal wrenches;

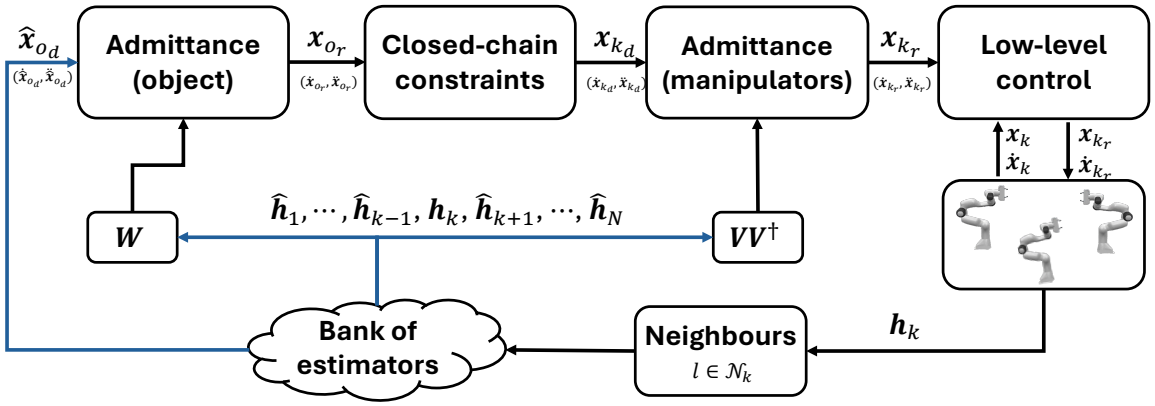


Figure 2.5 Block scheme of the overall control architecture for the k-th agent. The bank of estimators enables the distributed information flow among the robots ($k = 1, \ldots, N$). Blue arrows indicate the estimations of the object desired trajectory and the wrench collective vector performed by each agent.

- **Motion controller**: finally, a motion controller is employed to track the reference trajectory; in principle, any suitable low-level motion control strategy can be adopted.

A key feature of this framework lies in its flexibility: external and internal motion variables are decoupled, allowing the activation or deactivation of each impedance filter independently or in combination, without compromising the overall control performance.

Starting from the external admittance impedance considered in centralized approaches [57, 15],

$$(\boldsymbol{A}-\boldsymbol{I}_6)\boldsymbol{M}_o\ddot{\boldsymbol{x}}_{o_r}+\boldsymbol{D}_o\dot{\boldsymbol{x}}_{o_r}+\boldsymbol{K}_o\boldsymbol{x}_{o_r}+\boldsymbol{\gamma}_o = \boldsymbol{C}_o\dot{\boldsymbol{x}}_o+\boldsymbol{g}_o+\boldsymbol{A}\boldsymbol{M}_o\ddot{\boldsymbol{x}}_{o_d}+\boldsymbol{D}_o\dot{\boldsymbol{x}}_{o_d}+\boldsymbol{K}_o\boldsymbol{x}_{o_d}, \quad (2.84)$$

where the matrix $\boldsymbol{A}$ introduces an additional DoF in modeling the virtual inertia assigned to the system, the aim is to develop a fully distributed architecture combining both external and internal admittance control strategies.

The asymptotic stability of (2.84) is ensured provided that the eigenvalues of the positive definite diagonal matrix $\boldsymbol{A}$ are greater than or equal to 1 [57]. The object dynamics is thereby obtained by assuming the perfect tracking of the object trajectory, i.e. $\boldsymbol{x}_o = \boldsymbol{x}_{o_r}$, as

$$\boldsymbol{A}\boldsymbol{M}_o\ddot{\boldsymbol{e}}_o + \boldsymbol{D}_o\dot{\boldsymbol{e}}_o + \boldsymbol{K}_o\boldsymbol{e}_o = \boldsymbol{\gamma}_{env}. \quad (2.85)$$

In the envisaged approach, the previous assumption does not hold, since each robot computes its local estimate of the object desired trajectory, ${}^{i}\hat{\boldsymbol{x}}_{o_d}$, whilst the actual object trajectory results as the composition of all the reference trajectories, computed as:

$$(\boldsymbol{A}-\boldsymbol{I}_6)\boldsymbol{M}_o{}^{i}\ddot{\boldsymbol{x}}_{o_r}+\boldsymbol{D}_o{}^{i}\dot{\boldsymbol{x}}_{o_r}+\boldsymbol{K}_o{}^{i}\boldsymbol{x}_{o_r}+{}^{i}\hat{\boldsymbol{\gamma}}_o=\boldsymbol{C}_o\dot{\boldsymbol{x}}_o+\boldsymbol{g}_o+\boldsymbol{A}\boldsymbol{M}_o{}^{i}\ddot{\hat{\boldsymbol{x}}}_{o_d}+\boldsymbol{D}_o{}^{i}\dot{\hat{\boldsymbol{x}}}_{o_d}+\boldsymbol{K}_o{}^{i}\hat{\boldsymbol{x}}_{o_d}. \tag{2.86}$$

According to the Equation (2.97) in the following stability analysis, the object behavior can be shaped by properly tuning the positive definite matrices representing inertia ($\boldsymbol{AM}_o$), damping ($\boldsymbol{D}_o$), and stiffness ($\boldsymbol{K}_o$). The output of the external impedance filter is then mapped to the end-effector motion variables through the closed-chain constraints (2.54), and subsequently fed into the internal impedance filter. The latter processes both local kinematic and dynamic inputs: the i-th end-effector desired trajectory and the i-th robot contribution to the internal wrench, obtained from (2.70). Noticeably, this filter does not alter the object trajectory; instead, it regulates the i-th end-effector reference motion to prevent excessive internal forces and moments, via (2.79).

2.10.1 Stability analysis

For the purpose of this study, the object size and weight are supposed to be provided to the robotic team a priori, as stated in Assumption 3. However, this assumption can be further removed, as in [4]. Moreover, let us remove also the assumption of desired trajectory provided a priori to the robots, previously introduced in Sections 2.8 and 2.9. In this scenario, each robot is responsible for estimating the desired object trajectory, according to the estimation scheme aforementioned in Section 2.3, in order to calculate its end-effector trajectory via (2.54), as well as all the forces and torques exerted on the object, to implement the external and internal admittance filters through (2.58) and (2.63). Namely, by resorting to the relations proposed Section 2.5, the i-th robot retrieves its estimate of the desired trajectory of the object ${}^{i}\hat{\boldsymbol{x}}_{o_d}({}^{i}\dot{\hat{\boldsymbol{x}}}_{o_d}, {}^{i}\ddot{\hat{\boldsymbol{x}}}_{o_d})$, with a corresponding estimation error of ${}^{i}\widetilde{\boldsymbol{x}}_{o_d}({}^{i}\dot{\widetilde{\boldsymbol{x}}}_{o_d}, {}^{i}\ddot{\widetilde{\boldsymbol{x}}}_{o_d})$, via (2.23) and (2.24). Similarly, the i-th agent estimates the wrenches exerted on the object, ${}^{i}\hat{\boldsymbol{h}}_e$, with the corresponding estimation error ${}^{i}\widetilde{\boldsymbol{h}}_e$, via (2.64) and (2.66).

Moreover, the stability of the proposed control scheme is assessed if the following assumptions hold:

Assumption 10. *In the contact between the robot end-effectors and the object, no slipping phenomena occur.*

Hence, by setting the object reference frame to be coincident with the absolute one (see Remark 2.3.1), and recalling the definition of the virtual stick, the object pose can be rewritten as

$$\boldsymbol{x}_o = \frac{1}{N}\sum_{j=1}^{N} \boldsymbol{x}_j = \frac{1}{N}\sum_{j=1}^{N} ({}^j\boldsymbol{x}_{o_r} + \boldsymbol{r}_j + {}^j\widetilde{\boldsymbol{r}}_j + {}^j\boldsymbol{e}_I), \tag{2.87}$$

where ${}^j\boldsymbol{x}_{o_r}$ denotes the object reference pose computed by the j-th agent, ${}^j\widetilde{\boldsymbol{r}}_j$ represents the j-th robot error of estimating its own virtual stick, $\boldsymbol{r}_j$, via (2.41) and ${}^j\boldsymbol{e}_I = \boldsymbol{x}_{j_d} - \boldsymbol{x}_j$ is j-th end-effector pose tracking error introduced by the internal admittance filter. According to the stability analysis of the internal admittance scheme carried out in Section 2.9, ${}^j\boldsymbol{e}_I$ is also bounded.

Furthermore, by exploiting the linearity property of the time derivative, from (2.87) it yields

$$\ddot{\boldsymbol{x}}_o = \frac{1}{N}\sum_{j=1}^{N} ({}^j\ddot{\boldsymbol{x}}_{o_r} + \ddot{\boldsymbol{r}}_j + {}^j\ddot{\widetilde{\boldsymbol{r}}}_j + {}^j\ddot{\boldsymbol{e}}_I). \tag{2.88}$$

Assumption 11. *The connection between the object and the external environment is modeled by introducing a contact reference frame $\boldsymbol{\Sigma}_c$ attached to the object and linked to the environment at the contact point by a 6-DOFs spring. Thus, the wrench exerted by the external environment on the object is given by*

$$\boldsymbol{h}_{env} = \boldsymbol{K}_{env}(\boldsymbol{x}_c - \boldsymbol{x}_{env,eq}) \tag{2.89}$$

where $\boldsymbol{x}_{env,eq}$ represents the equilibrium pose of the environment, $\boldsymbol{x}_c$ is the pose of the frame $\boldsymbol{\Sigma}_c$ and $\boldsymbol{K}_{env} \in \mathbb{R}^{6\times 6}$ is a positive definite matrix representing the environment stiffness. Moreover, defining as $\boldsymbol{r}_{env}$ the virtual stick connecting the origin of $\boldsymbol{\Sigma}_c$ to $\boldsymbol{x}_o$, (2.89) *turns into*

$$\boldsymbol{h}_{env} = \boldsymbol{K}_{env}(\boldsymbol{x}_o + \boldsymbol{r}_{env} - \boldsymbol{x}_{env,eq}). \tag{2.90}$$

The i-th agent implements the external admittance equation, as in (2.86). Then, taking into account either the estimation errors ${}^i\widetilde{\boldsymbol{x}}_{o_d}$ and their time derivative, or ${}^i\widetilde{\boldsymbol{h}}_e$,

(2.86) can be rewritten as:

$$(\boldsymbol{A}-\boldsymbol{I}_6)\boldsymbol{M}_o{}^i\ddot{\boldsymbol{x}}_{o_r}+\boldsymbol{D}_o{}^i\dot{\boldsymbol{x}}_{o_r}+\boldsymbol{K}_o{}^i\boldsymbol{x}_{o_r}+\boldsymbol{\gamma}_o+{}^i\boldsymbol{\delta}_{h_o}= \\ \boldsymbol{C}_o\dot{\boldsymbol{x}}_o+\boldsymbol{g}_o+\boldsymbol{A}\boldsymbol{M}_o\ddot{\boldsymbol{x}}_{o_d}+\boldsymbol{D}_o\dot{\boldsymbol{x}}_{o_d}+\boldsymbol{K}_o\boldsymbol{x}_{o_d}+{}^i\boldsymbol{\delta}_{x_{o_d}}. \quad (2.91)$$

In particular, both trajectory and wrench estimation errors contribute to ${}^i\boldsymbol{\delta}_{h_o}$:

$${}^i\boldsymbol{\delta}_{h_o}=\boldsymbol{T}_A^{\mathrm{T}}({}^i\widetilde{\boldsymbol{\phi}}_o)\,\boldsymbol{G}({}^i\widetilde{\boldsymbol{r}}_i){}^i\widetilde{\boldsymbol{h}}_e, \quad (2.92)$$

while only kinematics estimation errors result in ${}^i\boldsymbol{\delta}_{x_{o_d}}$:

$${}^i\boldsymbol{\delta}_{x_{o_d}}=-\left(\boldsymbol{A}\boldsymbol{M}_o{}^i\ddot{\widetilde{\boldsymbol{x}}}_{o_d}+\boldsymbol{D}_o{}^i\dot{\widetilde{\boldsymbol{x}}}_{o_d}+\boldsymbol{K}_o{}^i\widetilde{\boldsymbol{x}}_{o_d}\right). \quad (2.93)$$

Remark 2.10.1. *Both ${}^i\boldsymbol{\delta}_{h_o}$ and ${}^i\boldsymbol{\delta}_{x_{o_d}}$ represent bounded inputs for the system in* (2.94)*.*

As far as (2.92) *is concerned, ${}^i\widetilde{\boldsymbol{\phi}}_o$ and ${}^i\widetilde{\boldsymbol{r}}_i$ originate from the trajectory estimator, whose estimation errors have been proven to be vanishing or, in the worst-case scenario, bounded, as shown in Section 2.5.1. In turn, ${}^i\widetilde{\boldsymbol{h}}_e$ represents the error due to the wrench estimation of the i-th agent, which is bounded, as demonstrated in Section 2.7. The same considerations apply to* (2.93)*, where only trajectory estimation errors are involved. Thus, this contribution is at most bounded.*

Then, folding (2.55) in (2.91), it follows:

$$\boldsymbol{A}\boldsymbol{M}_o(\ddot{\boldsymbol{x}}_{o_d}-{}^i\ddot{\boldsymbol{x}}_{o_r})+\boldsymbol{D}_o(\dot{\boldsymbol{x}}_{o_d}-{}^i\dot{\boldsymbol{x}}_{o_r})\boldsymbol{K}_o(\boldsymbol{x}_{o_d}-{}^i\boldsymbol{x}_{o_r})= \\ \boldsymbol{M}_o(\ddot{\boldsymbol{x}}_o-{}^i\ddot{\boldsymbol{x}}_{o_r})+\boldsymbol{\gamma}_{env}+{}^i\boldsymbol{\delta}_{h_o}-{}^i\boldsymbol{\delta}_{x_{o_d}}. \quad (2.94)$$

Finally, substituting (2.87), (2.88) and (2.90) in (2.94), it follows

$$\boldsymbol{A}\boldsymbol{M}_o(\ddot{\boldsymbol{x}}_{o_d}-{}^i\ddot{\boldsymbol{x}}_{o_r})+\boldsymbol{D}_o(\dot{\boldsymbol{x}}_{o_d}-{}^i\dot{\boldsymbol{x}}_{o_r})+\boldsymbol{K}_o(\boldsymbol{x}_{o_d}-{}^i\boldsymbol{x}_{o_r})= \\ +\boldsymbol{M}_o\left[\frac{1}{N}\sum_{j=1}^{N}({}^j\ddot{\boldsymbol{x}}_{o_r}+\ddot{\boldsymbol{r}}_j+{}^j\ddot{\widetilde{\boldsymbol{r}}}_j+{}^j\ddot{\boldsymbol{e}}_I)-{}^i\boldsymbol{x}_{o_r}\right]+ \\ \boldsymbol{K}_{env}\left[\frac{1}{N}\sum_{j=1}^{N}({}^j\boldsymbol{x}_{o_r}+\boldsymbol{r}_j+{}^j\boldsymbol{\delta}_I)-\bar{\boldsymbol{x}}_{env,eq}\right]+{}^i\boldsymbol{\delta}_E, \quad (2.95)$$

where ${}^j\boldsymbol{\delta}_I={}^j\widetilde{\boldsymbol{r}}_j+{}^j\boldsymbol{e}_I$ and ${}^i\boldsymbol{\delta}_E={}^i\boldsymbol{\delta}_{h_o}-{}^i\boldsymbol{\delta}_{x_{o_d}}$.

Remark 2.10.2. *The quantities ${}^j\ddot{\tilde{\boldsymbol{r}}}_j$, ${}^j\ddot{\boldsymbol{e}}_I$ and ${}^j\boldsymbol{\delta}_I$, as well as ${}^i\boldsymbol{\delta}_E$, represent the uncertainties resulting from the internal and external admittance filters implemented in a distributed fashion, respectively. As expected, the stability of the overall control scheme is indeed influenced by terms arising from both admittance filters, which in turn depend on the estimation errors mentioned earlier.*

Then, summing the set of N equations formulated with respect to all the manipulators, and dividing by N, it yields

$$\begin{aligned}\boldsymbol{A}\boldsymbol{M}_o(\ddot{\boldsymbol{x}}_{o_d} - \frac{1}{N}\sum_{i=1}^{N} {}^i\ddot{\boldsymbol{x}}_{o_r}) + \boldsymbol{D}_o(\dot{\boldsymbol{x}}_{o_d} - \frac{1}{N}\sum_{i=1}^{N} {}^i\dot{\boldsymbol{x}}_{o_r}) + \boldsymbol{K}_o(\boldsymbol{x}_{o_d} - \frac{1}{N}\sum_{i=1}^{N} {}^i\boldsymbol{x}_{o_r}) = \\ = \frac{1}{N}\boldsymbol{M}_o\sum_{j=1}^{N} {}^j\ddot{\boldsymbol{x}}_{o_r} + \frac{1}{N}\boldsymbol{M}_o\sum_{j=1}^{N}\left(-\ddot{\boldsymbol{r}}_j + {}^j\ddot{\tilde{\boldsymbol{r}}}_j + {}^j\ddot{\boldsymbol{e}}_I\right) - \frac{1}{N}\boldsymbol{M}_o\sum_{i=1}^{N} {}^i\ddot{\boldsymbol{x}}_{o_r} + \\ \boldsymbol{K}_{env}\left(\frac{1}{N}\sum_{j=1}^{N} {}^j\boldsymbol{x}_{o_r}\right) + \boldsymbol{\delta}_I + \boldsymbol{\delta}_E - \boldsymbol{K}_{env}\bar{\boldsymbol{x}}_{env,eq}, \quad (2.96)\end{aligned}$$

where $\boldsymbol{\delta}_E = \frac{1}{N}\sum_{i=1}^{N} {}^i\boldsymbol{\delta}_E$, and $\boldsymbol{\delta}_I = \boldsymbol{K}_{env}[\frac{1}{N}\sum_{j=1}^{N}(\boldsymbol{r}_j + {}^j\boldsymbol{\delta}_I)]$.

Consider the right-hand side of (2.96): without any loss of generality, the indices of the two sums ${}^i\ddot{\boldsymbol{x}}_{o_r}$, ${}^j\ddot{\boldsymbol{x}}_{o_r}$ can be inverted. Thus,

$$\frac{1}{N}\boldsymbol{M}_o\left[\sum_{j=1}^{N} {}^j\ddot{\boldsymbol{x}}_{o_r} - \sum_{i=1}^{N} {}^i\ddot{\boldsymbol{x}}_{o_r}\right] = 0.$$

Moreover, due to the geometric definition of the virtual stick and exploiting the linearity of the time derivative, it follows

$$\frac{1}{N}\boldsymbol{M}_o\sum_{j=1}^{N}\ddot{\boldsymbol{r}}_j = 0.$$

Therefore, the object tracking error of the multi-robot system, $\boldsymbol{e}_o = \boldsymbol{x}_{o_d} - \frac{1}{N}\sum_{i=1}^{N} {}^i\boldsymbol{x}_{o_r}$, has the following dynamics:

$$\bar{\boldsymbol{M}}_o\ddot{\boldsymbol{e}}_o + \boldsymbol{D}_o\dot{\boldsymbol{e}}_o + (\boldsymbol{K}_o + \boldsymbol{K}_{env})\,\boldsymbol{e}_o = \boldsymbol{d}, \quad (2.97)$$

where $\bar{\boldsymbol{M}}_o = \boldsymbol{A}\boldsymbol{M}_o$ and

$$\boldsymbol{d} = \boldsymbol{K}_{env}(\boldsymbol{x}_{o_d} - \bar{\boldsymbol{x}}_{env,eq}) + \boldsymbol{\delta}_I + \boldsymbol{\delta}_E + \frac{1}{N}\boldsymbol{M}_o\sum_{j=1}^{N}\left({}^j\ddot{\tilde{\boldsymbol{r}}}_j + {}^j\ddot{\boldsymbol{e}}_I\right)$$

Remark 2.10.3. *In Theorem 5, the boundedness of the extended state variables* $\xi = \left[\boldsymbol{e}_i^{\mathrm{T}},\ \dot{\boldsymbol{e}}_i^{\mathrm{T}}\right]^{\mathrm{T}}$*, where* $\boldsymbol{e}_i$ *denotes the* i*-th robot tracking error, has been proven under certain choice of the gain matrices. As a consequence, starting from* (2.79)*,*

$$\ddot{\boldsymbol{e}}_i = -\boldsymbol{M}_i \left[\boldsymbol{D}_i \dot{\boldsymbol{e}}_i + (\boldsymbol{K}_i - \boldsymbol{K}_{S_i})\, \boldsymbol{e}_i - \hat{\boldsymbol{h}}_{I_i}\right], \tag{2.98}$$

the boundedness of $\ddot{\boldsymbol{e}}_i$ *follows straightforwardly.*

Thus, the input $\frac{1}{N}\boldsymbol{M}_o \sum_{j=1}^{N} \left({}^{j}\ddot{\tilde{\boldsymbol{r}}}_j + {}^{j}\ddot{\boldsymbol{e}}_I\right)$ *is composed of:*

- *the error* ${}^{j}\ddot{\tilde{\boldsymbol{r}}}_j$*, coming from the trajectory estimator, that has been already proven to be bounded;*
- *the error* ${}^{j}\ddot{\boldsymbol{e}}_I$*, which is bounded, according to* (2.98)*.*

Consequently, the input of (2.96) *satisfies* $\|\boldsymbol{d}\| \leq \bar{d}$*, where* $\bar{d}$ *is a constant value.*

Defining $\boldsymbol{\zeta} = \left[\boldsymbol{e}_o^{\mathrm{T}},\ \ \dot{\boldsymbol{e}}_o^{\mathrm{T}}\right]^{\mathrm{T}}$, the dynamics in (2.97) can be reformulated as a linear system

$$\dot{\boldsymbol{\zeta}} = \begin{bmatrix} \boldsymbol{O}_3 & \boldsymbol{I}_3 \\ -\bar{\boldsymbol{M}}_o^{-1}(\boldsymbol{K}_o - \boldsymbol{K}_{env}) & -\bar{\boldsymbol{M}}_o^{-1}\boldsymbol{D}_o \end{bmatrix} \boldsymbol{\zeta} + \begin{bmatrix} 0 \\ \bar{\boldsymbol{M}}_o^{-1}\boldsymbol{d} \end{bmatrix}, \tag{2.99}$$

whose stability properties are stated by the following theorem (proof reported in A8).

Theorem 6. *The system* (2.99) *is input-to-state stable (ISS) provided that the positive definite matrices* $\bar{\boldsymbol{M}}_o$*,* $\boldsymbol{D}_o$ *and* $\boldsymbol{K}_o$ *satisfy the following condition:*

$$\lambda_m(\boldsymbol{D}_o) > \max\left\{\alpha\frac{\lambda_M(\bar{\boldsymbol{M}}_{\mathbf{o}})^{\mathbf{2}}}{\lambda_m(\bar{\boldsymbol{M}}_o)} - \frac{\lambda_m(\boldsymbol{K}_o)}{\alpha}, \alpha\lambda_M(\bar{\boldsymbol{M}}_o) - \frac{1}{4\alpha}\frac{\|\boldsymbol{K}_{env}\|^2}{\lambda_m(\boldsymbol{K}_o) - \|\boldsymbol{K}_{env}\|}\right\}, \tag{2.100}$$

where $\lambda_M(\cdot)$ *and* $\lambda_m(\cdot)$ *denote the maximum and minimum eigenvalue of a matrix, respectively, and* α *is a positive constant.*

Remark 2.10.4. *Although they do not give any suggestion for tuning the gain parameters in* (2.97)*, the Lyapunov conditions provide a possible interpretation of the system behavior, based on the values of* $\bar{\boldsymbol{M}}_o$*,* $\boldsymbol{D}_o$ *and* $\boldsymbol{K}_o$*:*

- *as* α *increases,* $\lambda_m(\boldsymbol{D}_o)$ *should increase;*
- *as* $\lambda_M(\bar{\boldsymbol{M}}_o)$ *increases,* $\lambda_m(\boldsymbol{D}_o)$ *should increase, according to the first argument in* (2.100)*;*

- *finally, according to the second argument in* (2.100)*, if* $\lambda_m(\boldsymbol{K}_o) > \|\boldsymbol{K}_{env}\|$*, as* $\lambda_m(\boldsymbol{K}_o)$ *increases,* $\lambda_m(\boldsymbol{D}_o)$ *should increase; on the contrary, if* $\lambda_m(\boldsymbol{K}_o) < \|\boldsymbol{K}_{env}\|$*, as* $\lambda_m(\boldsymbol{K}_o)$ *increases,* $\lambda_m(\boldsymbol{D}_o)$ *should decrease.*

Chapter 3

Simulations and experiments

3.1 Coordinated motion task

3.1.1 Simulation in CoppeliaSim

As a preliminary approach, a Franka Emika Panda dynamic simulator has been adopted to test the coordinated motion task: FrankaSim, an open-source simulator made by Inria [117], which builds on CoppeliaSim [118] and the Robot Operating System (ROS) [119]. More in detail, it integrates a library providing the Lagrangian dynamic model of the robot and is fully incorporated into the ViSP ecosystem [120]. FrankaSim is developed as part of *Visp ros*, a basket of generic ROS nodes that are based on the ViSP library and includes a C++ library of classes that allow ROS to be incorporated transparently. In FrankaSim, each robot is controlled with an inverse dynamics control, which is fed in by the desired end-effector trajectory and, in turn, computes the desired torques as output, providing them to the simulator.

A robotic cell consisting of $N = 4$ Franka Emika Panda robots [121] has been built in CoppeliaSim, as depicted in Fig. 3.1. The communication graph, representing the information flow among the agents, satisfies the subgraph connectivity property, as shown in Fig. 3.2: a simulated robot in CoppeliaSim corresponds to each agent. Specifically, the i-th agent computes its estimate of the absolute and relative desired trajectory, ${}^i\hat{\boldsymbol{x}}_a$, ${}^i\hat{\boldsymbol{x}}_r$ and their time derivative, according to (2.23). Subsequently, it retrieves its end-effector pose and velocity, ${}^i\hat{\boldsymbol{x}}_i$ and ${}^i\dot{\hat{\boldsymbol{x}}}i$, by exploiting the relations expressed in Section 2.5.3. Finally, the estimated end-effector trajectory is published on a proper ROS topic, which the low-level controller of the simulated robot subscribes to.

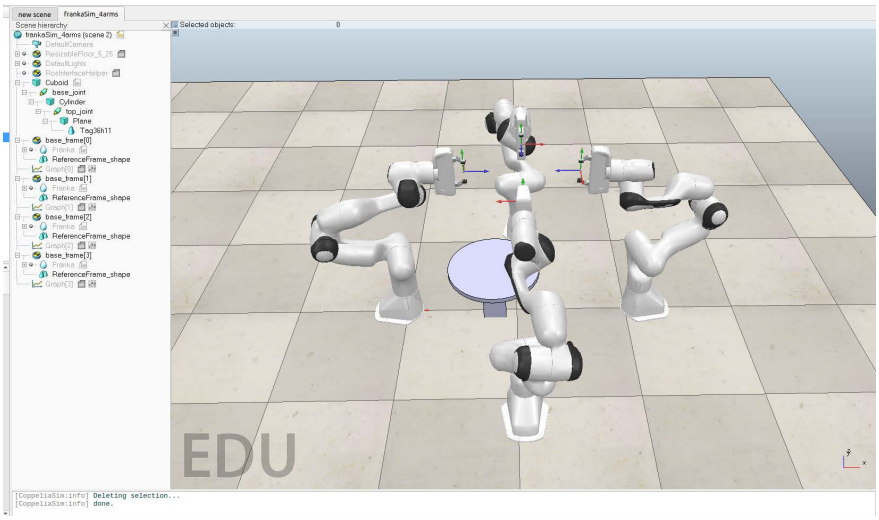


Figure 3.1 Screenshot of the simulation setup in CoppeliaSim.

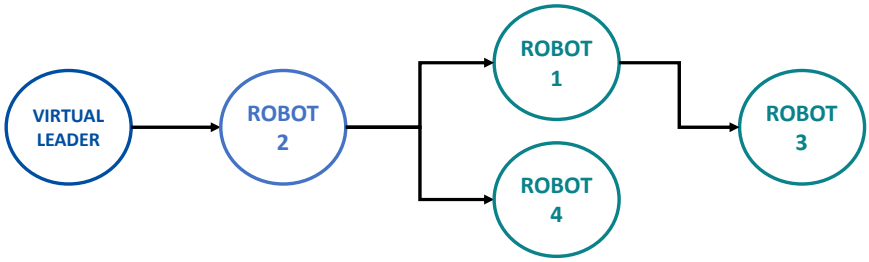


Figure 3.2 The proposed ROS system architecture: the information flow is represented by the arrows. The virtual leader node communicates with *Agent 2* node, i.e., the leader, whereas all other robots are the followers.

The assigned task consists in the execution of a planar motion planned for the absolute frame, $\boldsymbol{\Sigma}_a$, while the relative motion between the robots is expressed with respect to $\boldsymbol{\Sigma}_a$. More in detail, an arc of circumference in the $x-y$ plane of center $\boldsymbol{C}$, radius ρ and angle θ_o, and a rotation of θ_a along the z-axis are assigned to the absolute frame. The trajectory lasts 12 s, with 2 s of steady-state at its beginning to let the estimators end the transient period up and converge to the initial values. Denoting as t_i and t_f the initial and final time instant of the simulation, respectively, the time law of the positional task satisfies the Assumption 6 and can be expressed as:

$$\boldsymbol{p}_{a,d}(t) = \rho \begin{bmatrix} \cos\left(s_{a,p}(t)\right) \\ \sin\left(s_{a,p}(t)\right) \\ 0 \end{bmatrix} + \begin{bmatrix} x_C \\ y_C \\ z_C \end{bmatrix}, \tag{3.1}$$

Table 3.1 Trajectory parameters.

Parameter	Value
$\boldsymbol{C}\ [m]$	$(0.30, 0, 0.75)$
$\rho\ [m]$	0.2250
$\theta_o\ [rad]$	$\pi/6$
$\theta_a\ [rad]$	$\pi/6$
k_r	0.5

where $s_{a,p}(t) \in \{0, \theta_o\}$, while $\dot{s}_{a,p}(t_i) = \dot{s}_{a,p}(t_f) = 0$. Conversely, denoting as $\boldsymbol{\phi}_a(t_i)$ and $\boldsymbol{\phi}_a(t_f)$ the Euler angles vectors expressing the orientation of $\boldsymbol{\Sigma}_a$ at the initial and final time instants, respectively, the orientation trajectory assigned to $\boldsymbol{\Sigma}_a$ can be expressed in the Euler angles world as

$$\boldsymbol{\phi}_{a,d}(t) = \boldsymbol{\phi}_{a,d}(t_i) + s_{a,o}(t)\frac{\boldsymbol{\phi}_{a,d}(t_f) - \boldsymbol{\phi}_{a,d}(t_i)}{||\boldsymbol{\phi}_{a,d}(t_f) - \boldsymbol{\phi}_{a,d}(t_i)||}, \tag{3.2}$$

where $s_{a,o}(t) \in \{0, \ldots, ||\boldsymbol{\phi}_{a,d}(t_f) - \boldsymbol{\phi}_{a,d}(t_i)||\}$ and $\boldsymbol{\phi}_{a,d}(t_f) = \theta_a$.

Moreover, according to the formulation proposed in Section 2.5.3, $\boldsymbol{p}_r^a(t)$ accounts for the contraction or dilatation affecting the formation vector by means of a suitable coefficient k_r, with $0 < k_r < 1$ or $k_r > 1$, respectively. In turn, no relative rotation is assigned: $\boldsymbol{\phi}_r^a(t) = 0$. Thus, the relative trajectory is defined as:

$$\boldsymbol{p}_{r,d}^a(t) = \boldsymbol{p}_{r,d}^a(t_i) + s_r(t)\frac{\boldsymbol{p}_{r,d}^a(t_f) - \boldsymbol{p}_{r,d}^a(t_i)}{\left\|\boldsymbol{p}_{r,d}^a(t_f) - \boldsymbol{p}_{r,d}^a(t_i)\right\|}, \tag{3.3}$$

where $s_r(t) \in \{0, \ldots, ||\boldsymbol{p}_{r,d}^a(t_f) - \boldsymbol{p}_{r,d}^a(t_i)||\}$ and $\boldsymbol{p}_{r,d}^a(t_f) = k_r \boldsymbol{p}_{r,d}^a(t_i)$. The timing laws assigned to the abscissas $s_{a,p}(t)$, $s_{a,o}(t)$ and $s_r(t)$ are described by cubic polynomials.

Table 3.1 lists all the parameters related to the desired trajectory, which is provided to the low-level controller, namely the inverse dynamics controller integrated in FrankaSim.

Figures 3.3a and 3.3b show the position and orientation errors of the absolute trajectories estimation for the three followers, $\widetilde{\boldsymbol{p}}_{a,h}$ and $\widetilde{\boldsymbol{\phi}}_{a,h}$, with $h = 0, \ldots, p_a = 3$, respectively. The plots of $\widetilde{\boldsymbol{p}}_{a,0}$ and $\widetilde{\boldsymbol{\phi}}_{a,0}$ are restricted to the time interval $[0,\ 0.5]\,s$, focusing on the initial estimation period.

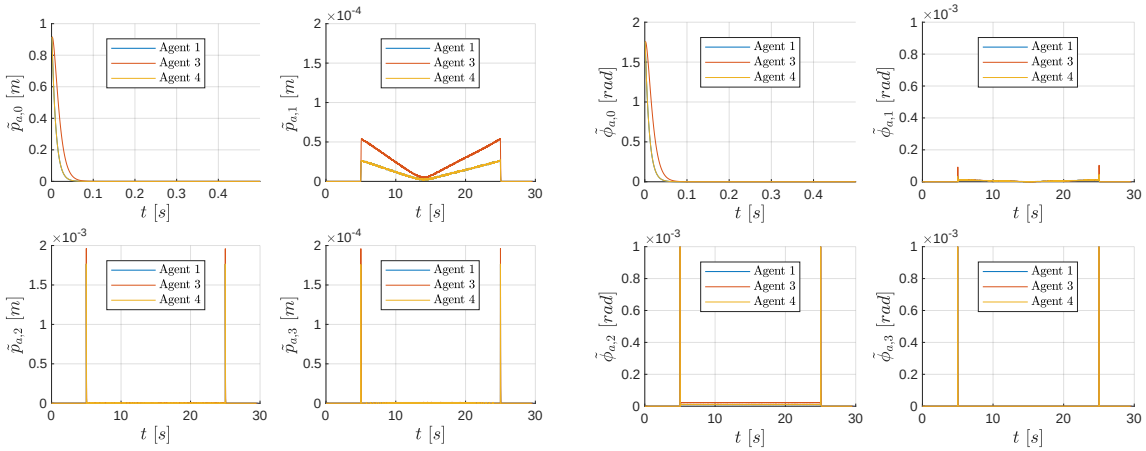
Due to (3.1), estimation position errors are bounded according to Assumption 6. As expected, it is noticeable that the estimates of *Agent 1* and *Agent 4* coincide, while

the *Agent 3* estimate exhibits a slight delay ($\sim 10^{-2}$ s), since it does not belong to the leader's neighborhood. Furthermore, the plots of $\widetilde{\boldsymbol{p}}_{a,h}$ and $\widetilde{\boldsymbol{\phi}}_{a,h}$, $h = 1, ..., 3$, show discontinuities at the starting and ending time instants of the trajectories due to the polynomials adopted.

The position errors of the formation trajectories estimation, $\widetilde{\boldsymbol{p}}^{a}_{r,h}$, for the three followers are shown in Figure 3.4a. As in the case of the absolute estimation errors, the plot of $\widetilde{\boldsymbol{p}}^{a}_{r,0}$ is restricted to the time interval $[0,\ 0.5]\, s$, focusing on the initial estimation period. As expected from (3.3), the error trends are quite similar to those observed in the case of the absolute orientation. In turn, since the relative orientation trajectory is constant, all the orientation errors are identically zero, but $\widetilde{\boldsymbol{\phi}}^{a}_{r,0}$, which converges to zero with similar dynamics of $\widetilde{\boldsymbol{p}}_{a,0}$, $\widetilde{\boldsymbol{\phi}}_{a,0}$ and $\widetilde{\boldsymbol{p}}^{a}_{r,0}$.

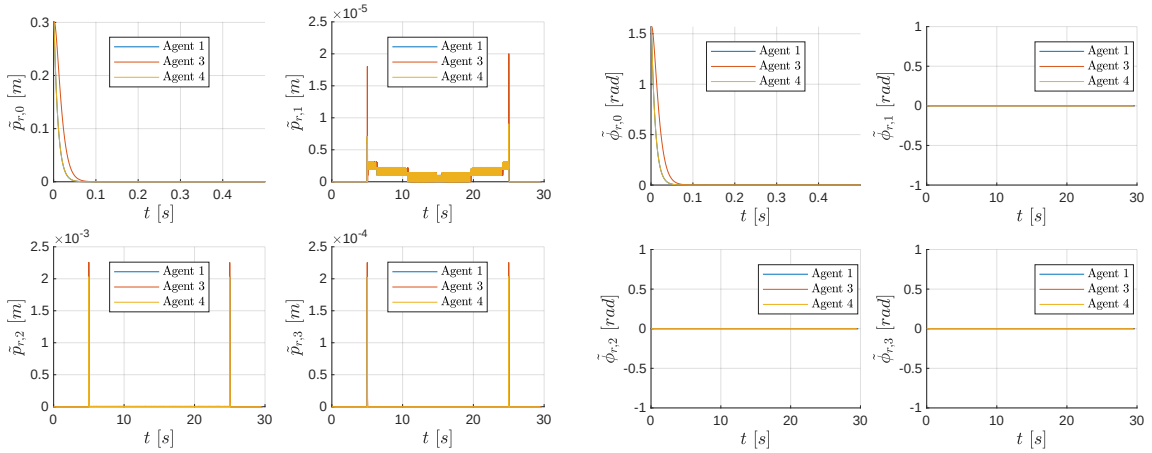
The norm of end-effectors' tracking errors in position, $||\widetilde{\boldsymbol{p}}_i||$, orientation, $||\widetilde{\boldsymbol{\phi}}_i||$ of the simulated robots are shown in Fig. 3.5, along with their time derivatives, $||\dot{\widetilde{\boldsymbol{p}}}_i||$ and $||\dot{\widetilde{\boldsymbol{\phi}}}_i||$. Since the simulated time provided by CoppeliaSim does not correspond to the actual time from ROS, the tracking errors associated with the linear velocity and Euler angle derivatives are also presented to verify the consistency of the simulated trajectory.

The signals of the actual end-effector poses are exploited to compute the absolute trajectory, whose tracking errors are shown in Figure 3.6: they are around 10^{-3} m (rad) in position (orientation), and 0.05 m/s (rad/s), as concerns the corresponding time derivatives.



(a) Followers' absolute position estimation errors. (b) Followers' absolute orientation estimation errors.

Figure 3.3 Followers' position and orientation estimation errors, $\widetilde{\boldsymbol{p}}_{a,h}$ and $\widetilde{\boldsymbol{\phi}}_{a,h}$ $(h = 0, ..., 3)$, of the absolute trajectories.



(a) Followers' relative position estimation errors. (b) Followers' relative orientation estimation errors.

Figure 3.4 Followers' position and orientation estimation errors, $\widetilde{\boldsymbol{p}}_{r,h}$ and $\widetilde{\boldsymbol{\phi}}_{r,h}$ $(h = 0, ..., 3)$, of the relative trajectories. The superscript a is omitted for compactness.

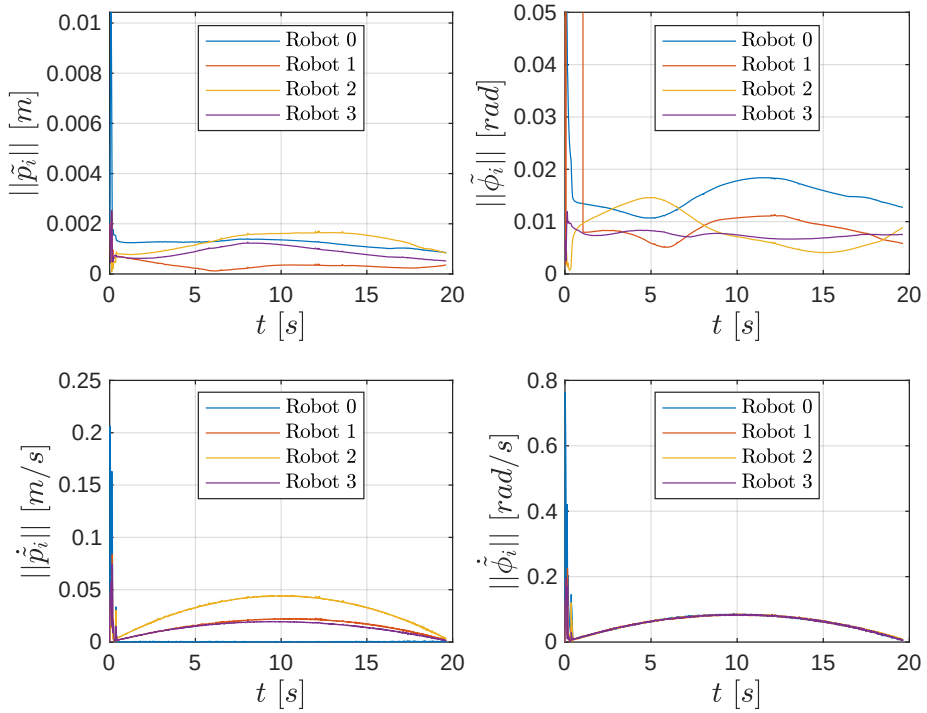


Figure 3.5 Norm of robots end-effector tracking errors (position $||\widetilde{\boldsymbol{p}_i}||$, orientation $||\widetilde{\boldsymbol{\phi}_i}||$ and their time derivatives, $||\dot{\tilde{\boldsymbol{p}}}_i||$ and $||\dot{\tilde{\boldsymbol{\phi}}}_i||$), with $i = 1, \ldots, 4$.

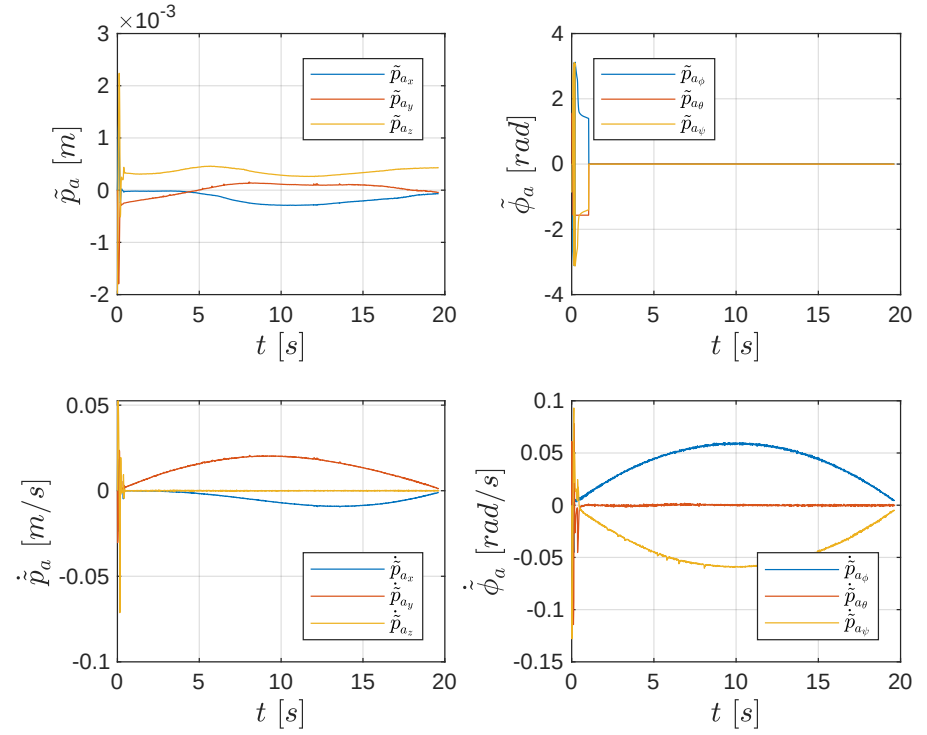


Figure 3.6 Absolute trajectory tracking error (position $\widetilde{\boldsymbol{p}}_a$, orientation $\widetilde{\boldsymbol{\phi}}_a$ and their time derivatives, $\dot{\tilde{\boldsymbol{p}}}_a$ and $\dot{\tilde{\boldsymbol{\phi}}}_a$).

3.1.2 Experiment with two real and two simulated Franka Emika Panda manipulators

Once assessed the effectiveness of the proposed strategy in simulation environments, the distributed control of coordinated motion tasks has been implemented on experimental setups.

In the first experiment, a coordinated motion task has been performed by a robotic workcell composed of both physical and virtual manipulators, as depicted in Fig. 3.7. Namely, two simulated robots have been substituted by two real Franka Emika Panda manipulators: according to the overall system architecture shown in Fig. 3.8, only *Agents 2* and *4* are associated to a virtual robot simulated in CoppeliaSim, whereas *Agents 1* and *3* communicate with the real robots. Each agent computes the reference end-effector trajectories, ${}^{i}\hat{\boldsymbol{x}}_{d,i}(t)$, but the former send them to the inverse dynamics controller of the adopted simulator, whilst the latter to the low-level controllers of the two Panda robots. These are 7-DOF anthropomorphic manipulators, featuring rotational joints, a maximum payload capacity of 3 kg, and a repeatability of 0.1 mm. Each robot is equipped with the integrated end-effector Franka Hand. Robot control is

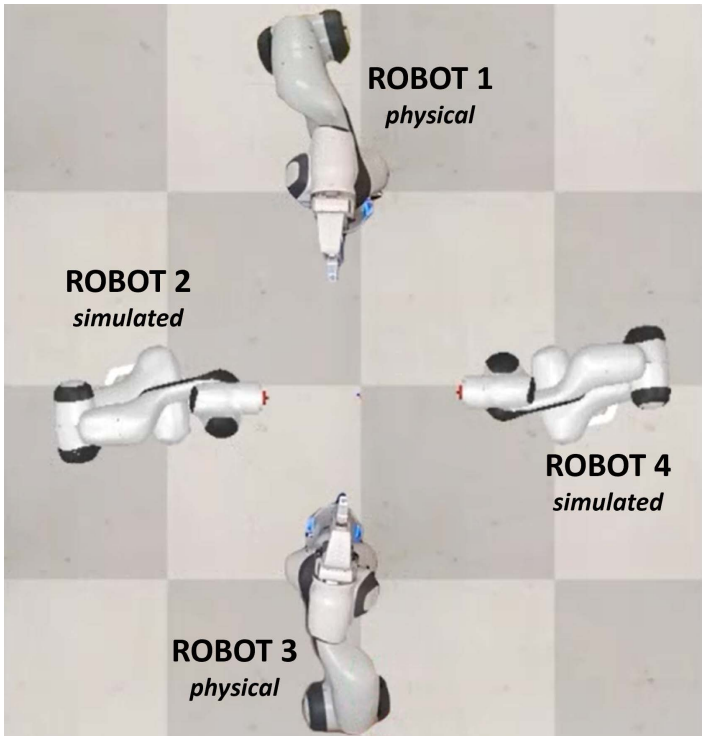


Figure 3.7 Simulation setup with two physical robots and two robots simulated in CoppeliaSim.

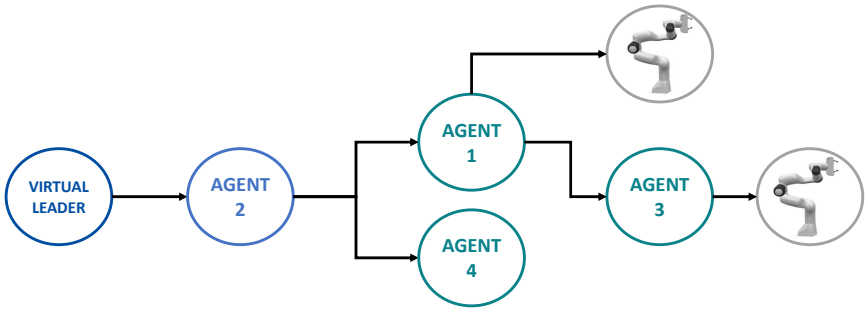


Figure 3.8 The proposed ROS system architecture: *Agents 1* and *3* communicate with two real Franka Emika Panda robots.

implemented via the open-source package *franka_ros*, which serves as a ROS wrapper for the low-level Franka Control Interface (FCI) provided through the libfranka C++ library. The FCI offers direct access to the robot's actuators and sensors and supports real-time communication at frequencies up to 1 kHz. This enables external controllers to interact directly with the robot's internal control loop by sending commands and receiving measurements with low latency [121].

Unlike Section 3.1.1, a closed-loop inverse kinematics algorithm (CLIK) [111] is adopted to track such a reference trajectory. In turn, the robot provides measured joint torques as feedback, leveraging the built-in torque sensors available at each joint, which stream data at 1 kHz via dedicated ROS nodes interfacing with the FCI. The control framework is implemented in C++, using ROS Melodic on Ubuntu 18.04.

Hence, the proposed system is composed by:

- a virtual leader node, which plans the desired trajectories for the absolute and relative motion tasks and provides them to the leader;
- a leader node and 3 follower nodes, corresponding to the 4 agents of the system. It is assumed that each agent has its ID and knows who are its neighbors. The i-th agent publishes its own estimate ${}^{i}\hat{\boldsymbol{x}}_{d,a,h}$ (${}^{i}\boldsymbol{x}_{d,a}^{(h)}$, for the leader) on a proper topic at 1 kHz frequency.
- a node, based on the *franka_ros* package, for each physical manipulator, which handles the communication with it: this node receives the desired end-effector trajectory estimate ${}^{i}\hat{\boldsymbol{x}}_{d,i}$, and solves the robot's inverse kinematics by reading the actual robot pose, through its own state. Alternatively, the agent node, either leader or follower, directly handles the communication with the simulator's low-level controller.

This setup differs from that presented in the previous section due to the addition of a ROS node that manages the communication with the robots' low-level controllers.

Moreover, the assigned trajectory is distinguished from the one proposed in Section 3.1.1 solely by the specific timing law, which is made up of an initial steady state period of 4 s, to ensure the convergence of the estimators, followed by 20 s of coordinated motion and finally 2 s of steady state. The time laws assigned to the abscissas $s_{a,p}(t)$, $s_{a,o}(t)$ and $s_r(t)$ in (3.1), (3.2) and (3.3), respectively, are quintic polynomials. Moreover, all the parameters listed in Table 3.1 are essentially the same, whilst the IK gain matrix, defined in (2.37), is $\boldsymbol{K}_q = diag\{[110, 110, 110, 80, 80, 80]\}$.

Figures 3.9a and 3.9b show the position and orientation estimation errors of the absolute trajectories for the three followers, $\widetilde{\boldsymbol{p}}_{a,h}$ and $\widetilde{\boldsymbol{\phi}}_{a,h}$, $h = 0, \ldots, p_a = 5$, respectively. The plots of $\widetilde{\boldsymbol{p}}_{a,0}$ and $\widetilde{\boldsymbol{\phi}}_{a,0}$ are restricted to the time interval $[0,\ 0.5]\, s$, focusing on the initial estimation period. Position estimation errors are bounded according to the Assumption 6, due to (3.1). As expected, it is worth noting that the estimates of *Agent 1* and *Agent 4* coincide, while the *Agent 3* estimate exhibits a slight delay ($\sim 10^{-2}$ s), since it does not belong to the leader's neighborhood. Furthermore, the plots of $\widetilde{\boldsymbol{p}}_{a,h}$, $h = 1, ..., 5$, show discontinuities at the starting and ending time instants of the trajectories due to the adoption of a polynomial time law: the continuity of the desired trajectory $\boldsymbol{p}_{d,h}$ is guarantee up to $h = 2$, while the cascaded implementation of the estimators is affected by the discontinuity of the higher derivative order, $\boldsymbol{p}_{d,p_a}$.

The position errors of the formation trajectories estimation, $\widetilde{\boldsymbol{p}}^a_{r,h}$, for the three followers are shown in Figure 3.9c, whilst the orientation errors are not shown as identical zero, except for $\widetilde{\boldsymbol{p}}^a_{r,0}$, which converges to zero. As in the case of absolute estimation errors, the plot of $\widetilde{\boldsymbol{p}}^a_{r,0}$ is restricted to the time interval $[0,\ 0.5]\, s$, focusing on the initial estimation period. As in the previous case study, the results are quite similar to the case of absolute orientation.

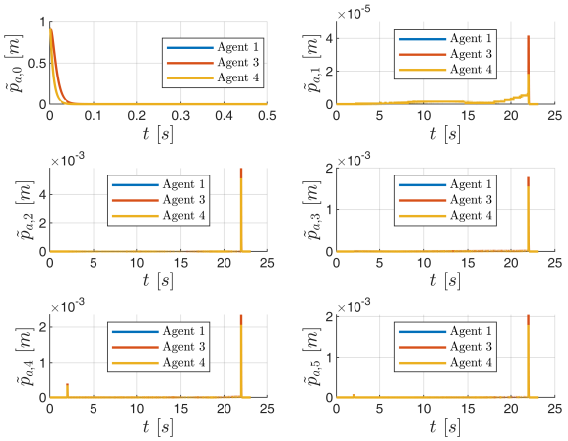
The norm of end-effectors' tracking errors in position, $||\widetilde{\boldsymbol{p}_i}||$, and orientation, $||\widetilde{\boldsymbol{\phi}_i}||$ of the simulated robots are shown in Fig. 3.10a. Regardless of the initial time errors, which can be attributed to the robots' repeatability, the position and orientation errors remain on the order of 10^{-4} m and 10^{-3} rad, respectively.

The absolute trajectory tracking errors are shown in Figure 3.10b, where plots are restricted to the time interval $[4,\ 24]\, s$, focusing on the motion period. They are the result of several error contributions:

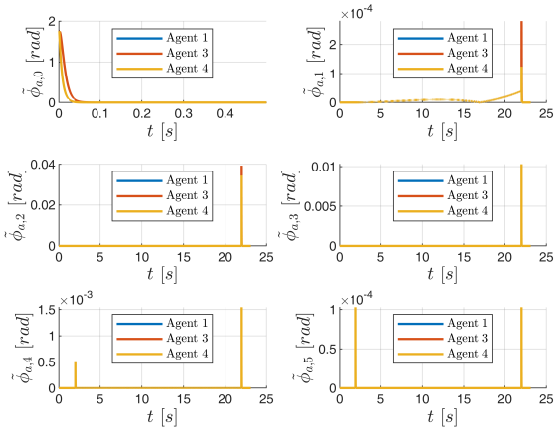
- the estimation errors $\widetilde{\boldsymbol{x}}_{a,d}$ and $\widetilde{\boldsymbol{x}}_{r,d}$ ($\sim 10^{-5}$m and rad);

- the errors introduced by the alternative computation of the desired end-effector trajectories, according to Section 2.5.3 ($\sim 10^{-6}$m and rad);
- the manipulators' repeatability ($\sim 10^{-4}$m).

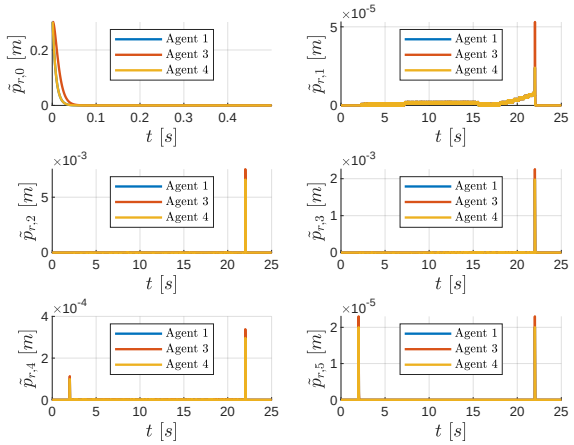
In particular, regarding the absolute position, the component x of $\widetilde{\boldsymbol{p}}_a$ is predominant with respect to the other two components, due to the fact that the assigned trajectory varies primarily along the x-axis. Similarly, an analogous argument can be made for the orientation: motion is commanded along the z-axis, so the corresponding error is higher.



(a) Followers' absolute position estimation errors.

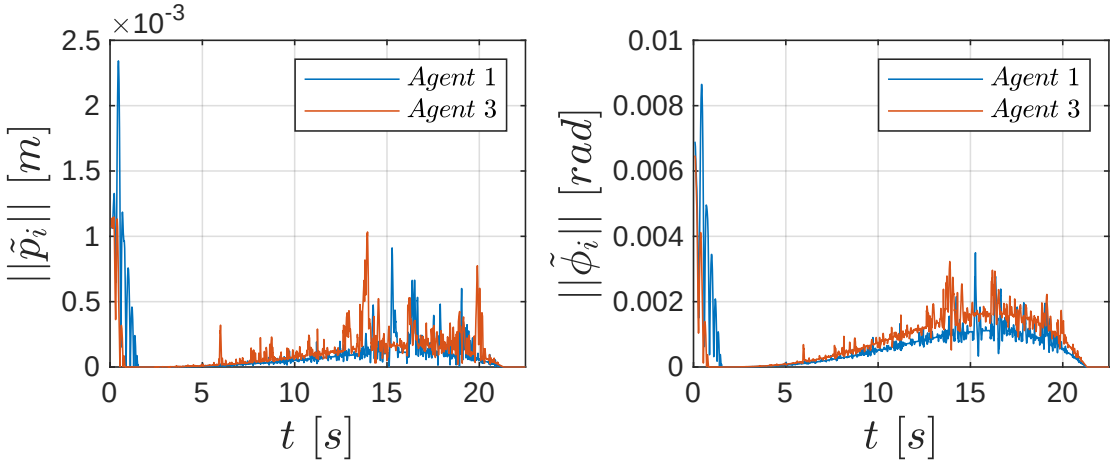


(b) Followers' absolute orientation estimation errors.

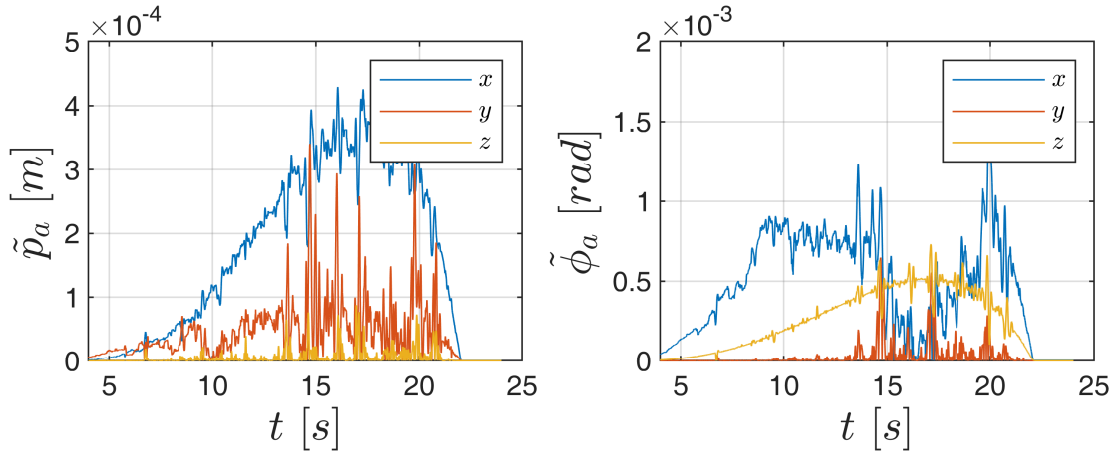


(c) Followers' relative position estimation errors.

Figure 3.9 Followers' position and orientation estimation errors, $\widetilde{\boldsymbol{p}}_{a,h}$ and $\widetilde{\boldsymbol{\phi}}_{a,h}$ ($h = 0, ..., 5$), of the absolute trajectories, along with the relative position estimation errors, $\widetilde{\boldsymbol{p}}_{r,h}$, where the superscript a is omitted for compactness.



(a) Robots end-effector tracking errors (position $||\widetilde{\boldsymbol{p}}_i||$ and orientation $||\widetilde{\boldsymbol{\phi}_i}||$), with $i = 1, \ldots, 4$.



(b) Absolute trajectory tracking error (position $||\widetilde{\boldsymbol{p}}_a||$, orientation $||\widetilde{\boldsymbol{\phi}}_a||$).

Figure 3.10 Tracking errors of robots end-effector (top) and absolute trajectory (bottom).

3.1.3 Experiment with three real Franka Emika Panda manipulators

Finally, a last test of motion coordination task has been carried out with a setup comprising three physical manipulators.

The cooperative robotic cell is composed of $N = 3$ Franka Emika Panda arms, as depicted in Fig. 3.11. One manipulator is mounted on a mobile base, which is mechanically braked for the purposes of this case study, according to Assumption 1.

As previously, a dedicated ROS node is assigned to each robot, enabling communication and data exchange according to the topology shown in Fig. 3.12: since in this experiment the number of agents is 3, a strongly connected communication graph has been adopted for establishing the data flow between the robots, where the *Agent 2* acts as the leader and the others are the followers.

Let us introduce an inertial frame $\boldsymbol{\Sigma}_w$, coincident with the base frame $\boldsymbol{\Sigma}_1$ of the manipulator mounted on the mobile base (hereafter referred to as Robot 1). The desired

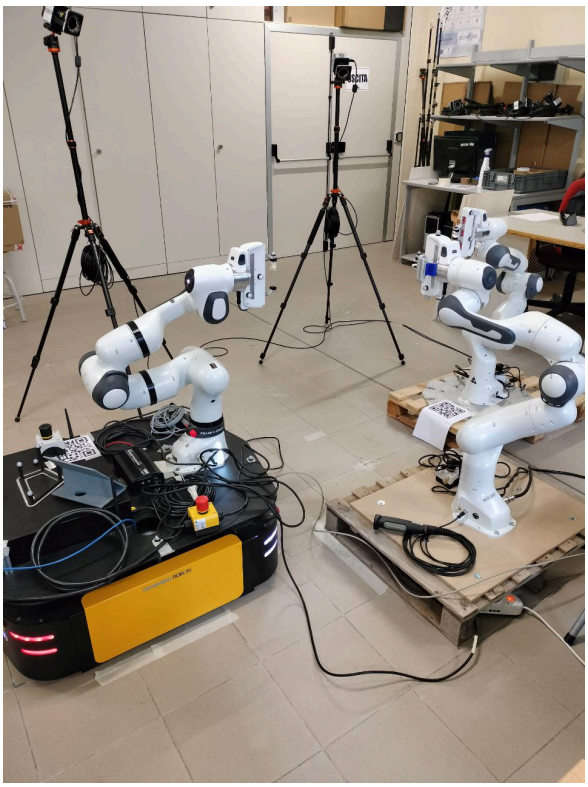


Figure 3.11 The experimental setup: the base frame of the robot mounted on the mobile base is set as the world frame $\boldsymbol{\Sigma}_w$. The Vicon motion capture system is exploited to retrieve the matrices $\boldsymbol{T}^w_{b,2}$ and $\boldsymbol{T}^w_{b,3}$, by performing a calibration procedure.

trajectory of the absolute frame consists of a translation along an arc of circumference with radius $\rho = 0.3$ m and angle $\theta_p = \pi/8$ rad, combined with a rotation about the z-axis of $\theta_o = \pi/8$ rad. The task is scheduled to be completed in 20 s: it begins with an initial steady-state phase of 2 s, followed by 15 s of motion, and ends with 3 s of steady state. The time laws are cubic polynomials.

Since the absolute and relative trajectories are formulated with respect to the world frame $\boldsymbol{\Sigma}_w$, they are then mapped into the end-effector trajectories through the relations expressed in Section 2.5.3. As regards Robots 2 and 3, these reference trajectories must be transformed into their respective base frames, $\boldsymbol{\Sigma}_{b,2}$ and $\boldsymbol{\Sigma}_{b,3}$. The transformation is achieved through the homogeneous matrices $\boldsymbol{T}^w_{b,2}$ and $\boldsymbol{T}^w_{b,3}$, which encode the pose of each robot base frame relative to $\boldsymbol{\Sigma}_w$. In particular, the matrices $\boldsymbol{T}^w_{b,1}$ and $\boldsymbol{T}^w_{b,2}$ are obtained via a calibration procedure performed with a Vicon motion capture system [122]. More in detail, six infrared cameras are adopted to measure the pose of each robot end-effector with respect to the Vicon base frame. By exploiting the robots' inverse kinematics, the relative poses of their base frames with respect to $\boldsymbol{\Sigma}_w$ can then be straightforwardly determined.

For brevity, the graphs concerning the trajectory estimation errors of *Agent 1* and 2 are not shown, since they experienced similar results of those presented in Section 3.1.1.

Fig. 3.13b shows the norm of end-effectors' tracking errors in position, $||\widetilde{\boldsymbol{p}}_i||$, and orientation, $||\widetilde{\boldsymbol{\phi}}_i||$, for $i = 1, \ldots, 3$. Since the assigned trajectories are broader than in

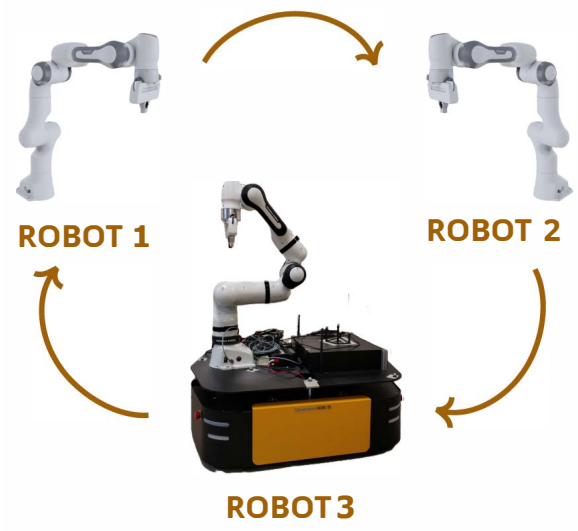
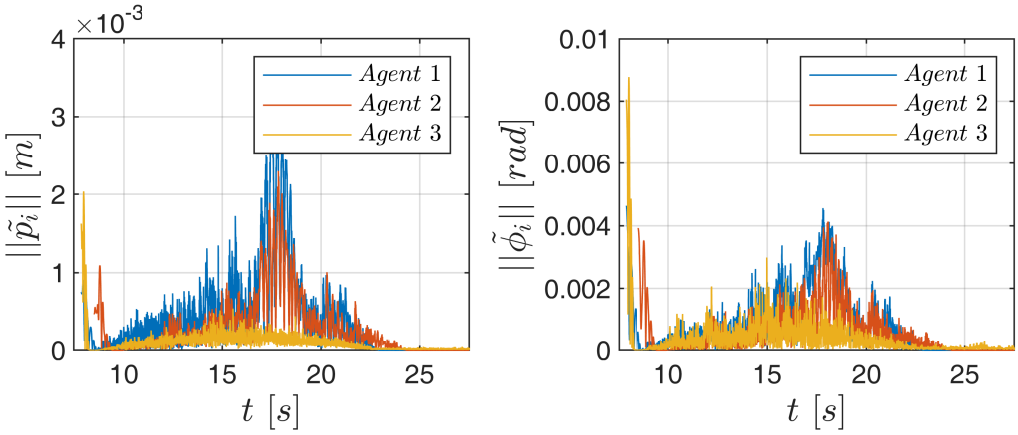


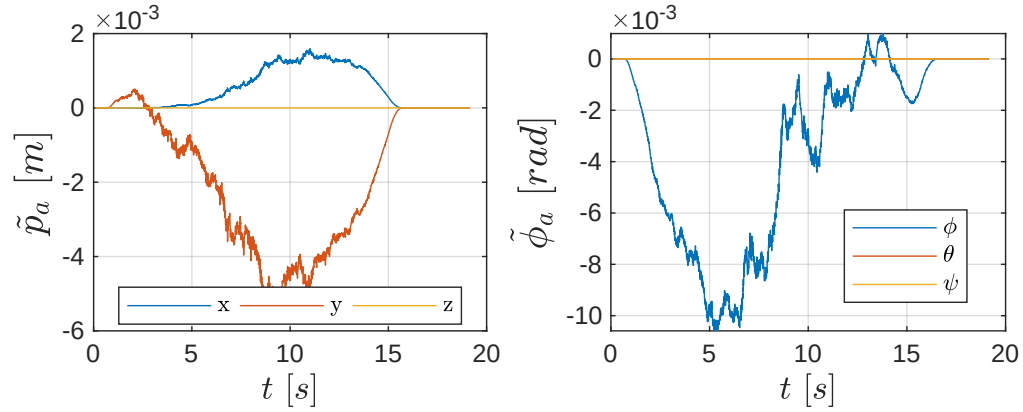
Figure 3.12 The communication graph established between the robots.

the previous case study, especially for *Robots 1* and *2*, which approach configurations close to the joint limits, their end-effectors exhibited slight oscillations upon their motion. As a result, the tracking error of these robots is onf the order of a few millimeters (10^{-3} m) in position and 10^{-3} rad, while it remains within the robot's repeatability range for *Robot 3*.

As a consequence, the absolute trajectory tracking errors experienced, shown in Figure 3.13b, belong to the same orders for both position and orientation. In particular, regarding the absolute position tracking errors, the components x and y of $\widetilde{\boldsymbol{p}}_a$ are predominant, due to the fact that the assigned trajectory varies primarily along these axes. As expected, a similar argument applies for the orientation: motion is commanded along the z-axis, so the corresponding error is higher.



(a) Robots end-effector tracking errors (position $||\widetilde{\boldsymbol{p}}_i||$ and orientation $||\widetilde{\boldsymbol{\phi}}_i||$), with $i = 1, \ldots, 3$.



(b) Absolute trajectory tracking error (position $||\widetilde{\boldsymbol{p}}_a||$, orientation $||\widetilde{\boldsymbol{\phi}}_a||$.)

Figure 3.13 Tracking errors of robots end-effector (top) and absolute trajectory (bottom).

3.2 Cooperative object transportation task

3.2.1 Simulation in CoppeliaSim with compliance control

The dynamic simulator FrankaSim has been adopted for a first test of the cooperative object transportation strategy proposed in Section 2.8, as reported in [116]. A robotic cell consisting of $N = 4$ robots, cooperatively grasping an object at their end-effectors, has been considered, as depicted in Fig. 3.14. The communication graph, representing the information flow among the robots, is modeled as the strongly connected graph shown in Fig. 3.15.

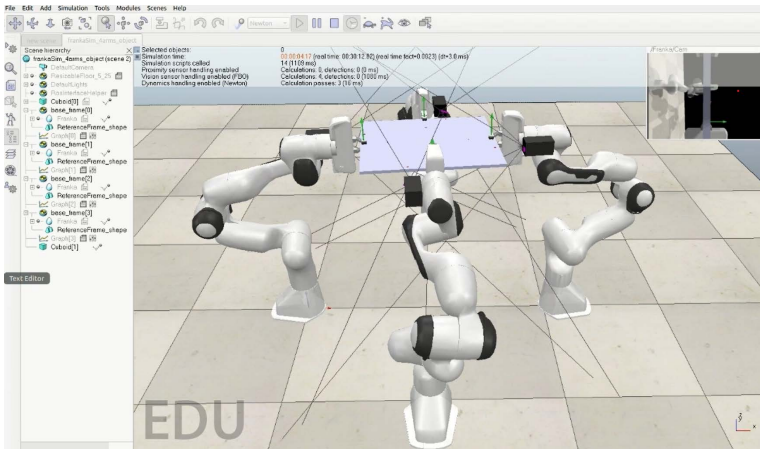


Figure 3.14 Screenshot of the simulation setup. The robotic team is cooperative grasping a wooden board.

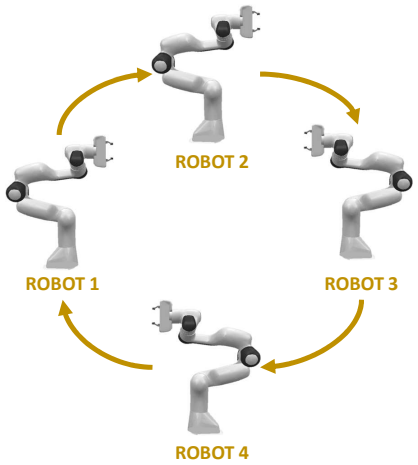


Figure 3.15 Communication graph between the agents.

Table 3.2 Control and Trajectory parameters.

Parameter	Value
k_0	200
$\boldsymbol{K}_{D_{i,p}}, \forall i$	$\text{diag}\{600, 700, 1000\}$
$\boldsymbol{K}_{P_{i,p}}, \forall i$	$\text{diag}\{1700, 1500, 1500\}$
$C\ [m]$	$(0.30, 0, 0.75)$
$\rho\ [m]$	0.2250
$\theta_o\ [rad]$	$\pi/6$

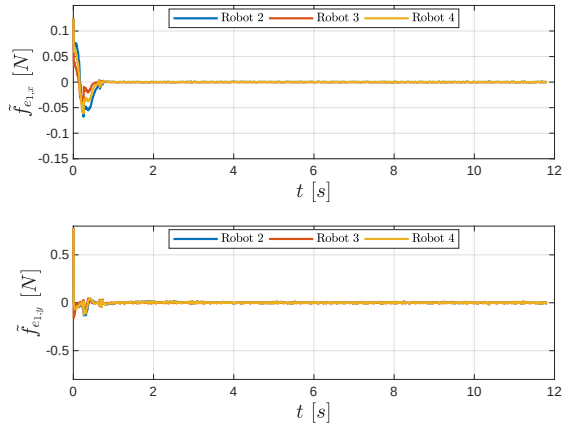
A planar motion is planned for the object, characterized by a weight of 0.5 kg and dimensions $0.475 \times 0.425 \times 0.01$ m. The object weight has been compensated in the end-effectors forces measurement. More in detail, an arc of circumference in the $x-y$ plane of center C, radius ρ and angle θ_o is assigned to the object. The timing law along the path is described by a cubic polynomials and the trajectory lasts 12 s, with 2 s of steady-state at its beginning to let the force estimators end up the transient period and converge to the initial values.

The trajectory parameters, as well as the compliance control gains, are listed in Table 3.2, while the low level controller is the inverse dynamic controller integrated in FrankaSim.

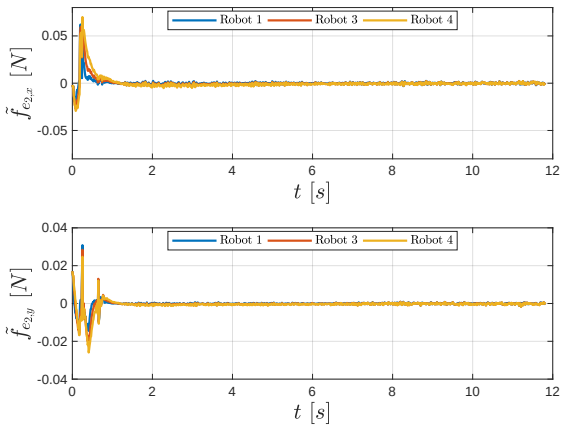
Figure 3.16 show the contact force estimation errors ${}^i\widetilde{\boldsymbol{f}}_i$ along the x and y axes, retrieved according to (2.64). The forces along the z axis are neglected since the motion is in the $x-y$ plane. According to the Corollary 2, after a transient, errors remain bounded and very close to zero during the whole experiment.

In order to show the effectiveness of the proposed estimator-controller architecture, the two-level controller in Section 2.8 is compared with a pure positional controller that does not incorporate the compliance control defined in (2.71). Figure 3.17a compares the internal forces contribution of Robot 2 on the object, $\boldsymbol{f}_{I_2}$, in both the approaches. It is noticeable that the compliance control significantly reduces the internal forces contribution. However, this reduction comes at the cost of a discrepancy between the commanded reference trajectory for each end-effector and the desired one (Fig. 3.17b). Indeed, all robots experience a negligible orientation tracking error, since no rotation is commanded and $\boldsymbol{\phi}_o$ is constant. In turn, the norm of the position tracking errors is about 10^{-2} m; we could suppose that this behavior is due to the force model adopted in (2.74).

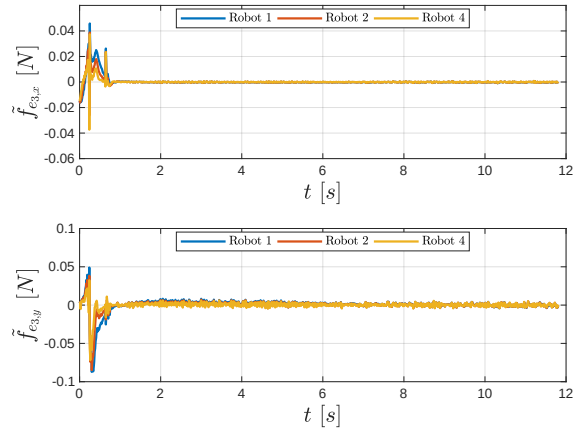
As a consequence, the object experiences similar position tracking error along the x and y axes where trajectories are commanded (Fig. 3.18).



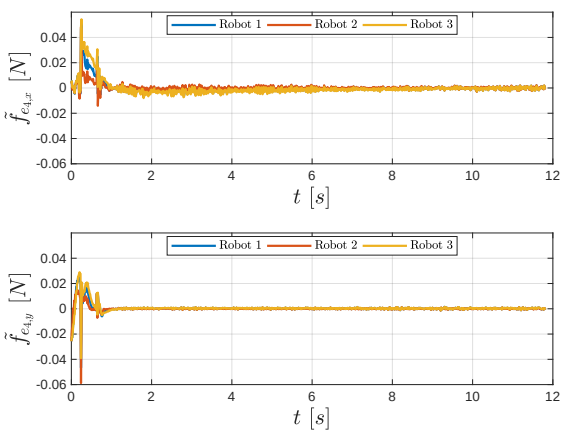
(a) Estimation error ${}^{i}\widetilde{\boldsymbol{f}}_1$ provided by Robots 2, 3, 4.



(b) Estimation error ${}^{i}\widetilde{\boldsymbol{f}}_2$ provided by Robots 1, 3, 4.

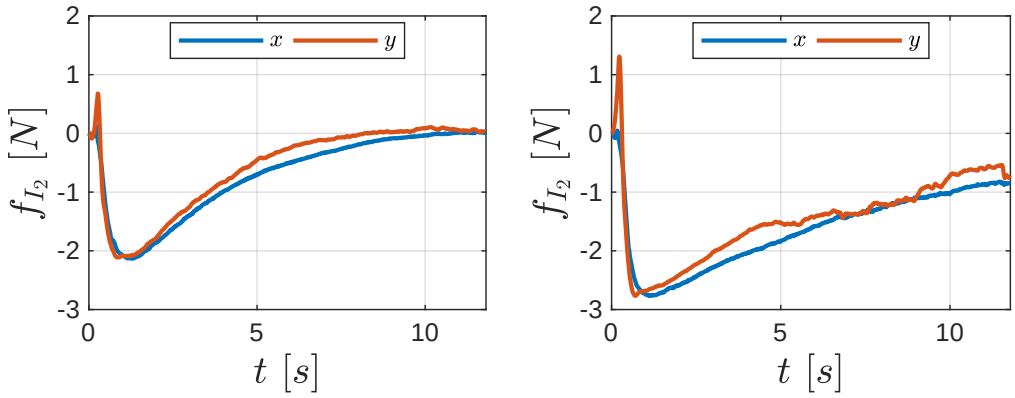


(c) Estimation error ${}^{i}\widetilde{\boldsymbol{f}}_3$ provided by Robots 1, 2, 4.

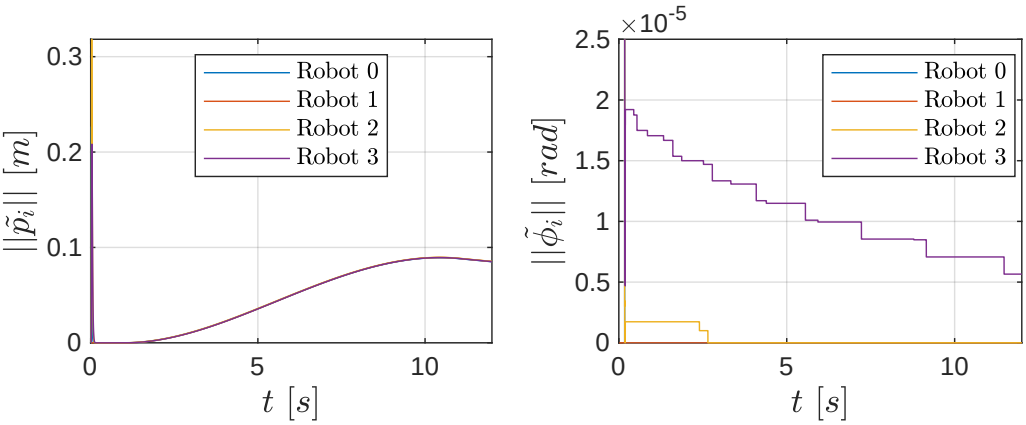


(d) Estimation error ${}^{i}\widetilde{\boldsymbol{f}}_4$ provided by Robots 1, 2, 3.

Figure 3.16 Force estimation errors ${}^{i}\widetilde{\boldsymbol{f}}_e$.



(a) Comparison of the internal forces in the $x-y$ plane with (left) and without (right) the compliance control.



(b) Robot end-effector tracking errors.

Figure 3.17 Internal force contribution for Robot 2 with and without the compliance controller activated (top); robots end-effector tracking errors with the compliance controller activated (bottom).

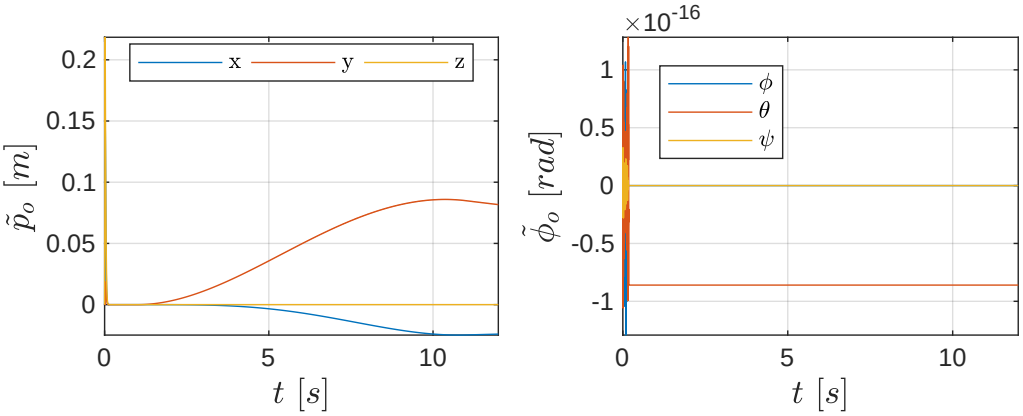


Figure 3.18 Object position and orientation tracking errors.

3.2.2 Simulation in CoppeliaSim with internal admittance control

The internal impedance controller proposed in Section 2.9 has been adopted to further investigate the object transportation task in the simulation environment of CoppeliaSim. The robotic setup and communication graph are the same as those employed in the compliance controller case study—illustrated in Figs. 3.14 and 3.15—as well as the assigned object desired trajectory. The gains of the admittance equation (2.79) have been tuned as:

- $\boldsymbol{M}_i = \text{diag}\{5,\, 5,\, 10,\, 20,\, 20,\, 20\}$,
- $\boldsymbol{K}_{D_i} = \text{diag}\{800,\, 800,\, 1000,\, 250,\, 250,\, 250\}$,
- $\boldsymbol{K}_{P_i} = \text{diag}\{2500,\, 2500,\, 3000,\, 500,\, 500,\, 500\}$,

whilst the built-in inverse dynamics controller of FrankaSim is adopted at the lower level.

Two simulations have been carried out to assess the effectiveness of the strategy proposed in Section 2.9 in the simulation environment: as previously, the two-level controller described in Section 2.9 is compared with a pure positional controller which does not enable the internal admittance control defined in (2.79).

Fig. 3.19 shows the internal forces and torques contributions of the *Robot 1* in both the simulations. In this case, the internal wrench reduction is lower than in the compliance controller case study; however, after an initial overshoot, both forces and torques converge closer to zero than in the positional controller case. Moreover, the tracking errors for the object and for each individual robot are smaller than in the compliance controller simulation, as depicted in Fig. 3.20.

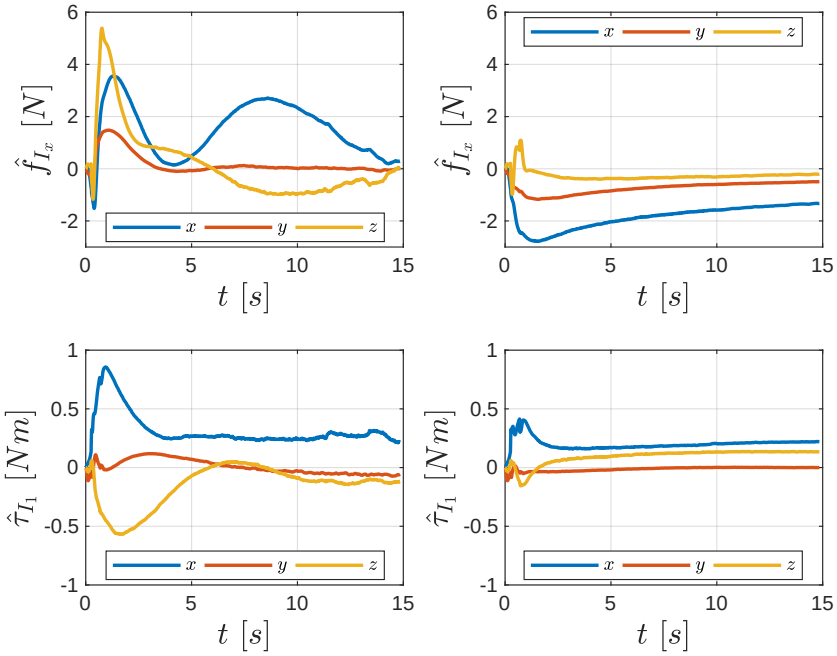
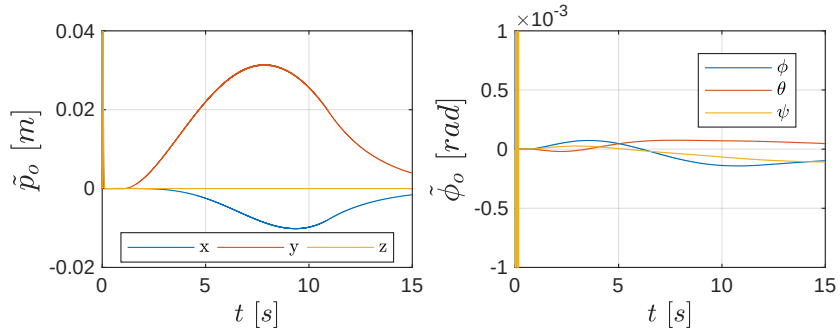
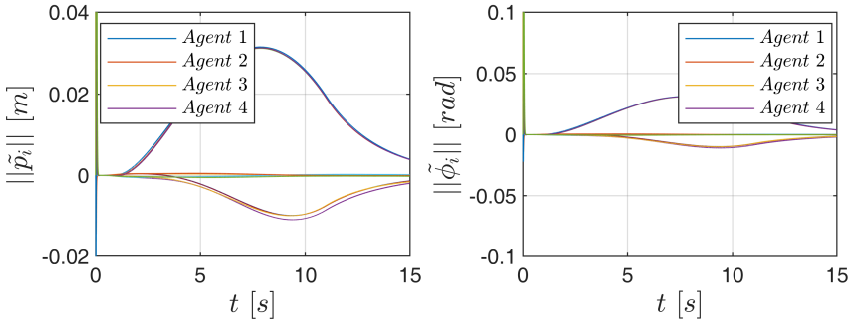


Figure 3.19 Comparison of the internal forces and torques with (left) and without (right) the compliance control.



(a) Comparison of the internal forces in the $x-y$ plane with (left) and without (right) the compliance control.



(b) Robot end-effector tracking errors.

Figure 3.20 Robots and object position and orientation tracking errors.

3.2.3 Experiment with internal admittance control

Following the assessment of the implemented distributed strategies for cooperative object transportation via simulations, experiments were conducted with various robotic setups.

Consider the cooperative robotic cell composed of $N = 3$ robotic arms shown in Fig. 3.21, which is in charge of executing a distributed cooperative transportation of a square-shaped wooden piece, whose size is $30 \times 30 \times 0.5$ cm, with $m_o = 0.250$ kg mass, as shown in [123]. Two of them form a fixed-base dual-arm system, while the third is a single-arm manipulator mounted on a mobile base that is assumed to be mechanically braked. This multi-robot system is responsible for executing an object transportation task in a distributed manner. It is assumed that each agent is provided a priori with all the necessary information regarding the desired end-effector trajectories—specifically, the positions, orientations, and their time derivatives—which serve as the input for the local admittance controller (Section 2.9).

The ABB Dual-arm YuMi is an anthropomorphic 14-DOF robot composed of two 7-DOF arms, each with a maximum payload of 500 g and a repeatability of 0.02 mm. It provides a joint velocity control interface via a ROS wrapper for the ABB Externally-Guided Motion (EGM) protocol. A Closed-Loop Inverse Kinematics (CLIK) algorithm computes joint velocity commands from desired Cartesian velocities for each arm. These commands are sent through ROS messages to a position-velocity tracking controller running at 250 Hz, with the robot state updated at the same rate. Since the YuMi does not include built-in force/torque sensors, each arm has been equipped with a Schunk Mini-40 Force/Torque sensor, mounted between the gripper and the flange. A pass-through cable ensures uninterrupted transmission of grip commands. The raw sensor data are forwarded to two ATI NETFT boxes, which perform gravity compensation and stream the resulting wrench vectors over LAN. A dedicated ROS node wraps a socket interface to receive these data and publish them to the ROS network. Although the entire process runs at over 300 Hz, subsampling is used to align the wrench data with the 250 Hz update cycle of the robot state. The tracking controller is implemented in Python, using ROS Melodic, and the system runs on Ubuntu 18.04.

The Universal Robot UR5e is a 6-DOF collaborative manipulator with a payload capacity of 5 kg. Communication is handled via the Real-Time Data Exchange (RTDE) interface, using the *ur_rtde* client library, with a maximum frequency of

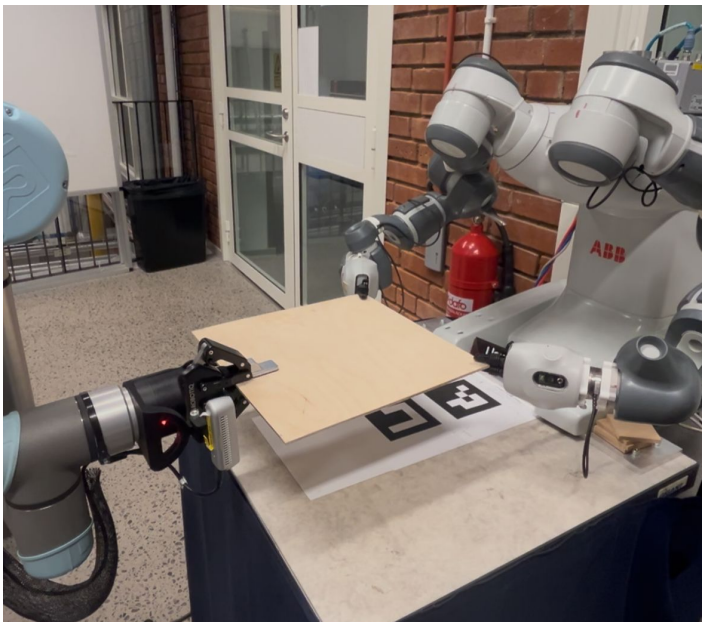


Figure 3.21 The experimental setup: the piece is grasped and transported by the robots. The Aruco marker lays on the working table and is employed for the setup calibration phase: its pose is known with reference to the world frame $\boldsymbol{\Sigma}_w$ is exploited for retrieving the UR5e robot base frame pose $\boldsymbol{T}_{ur}^{w}$.

500 Hz. Wrench measurements are obtained from the robot's built-in force-torque sensor through the same interface, and corrected for gravity to extract the net applied force. A Robotiq 2F-85 gripper is mounted on the wrist tool flange and controlled via RS485 serial communication, using a Python-based client protocol. The CLIK control law is implemented in a custom Python library that relies on Pinocchio's rigid-body algorithm framework [124]. Due to version incompatibilities with ROS Melodic, a lightweight translation layer bridges ROS messages and protobuf-based communication, introducing negligible overhead. The information flow is defined by the communication graph shown in Fig. 3.22.

At each time instant, the i-th robot receives the sensed wrench from its force-torque sensor, $\bar{\boldsymbol{h}}_i$, and the wrenches estimation of its neighbors, ${}^{i}\hat{\boldsymbol{h}}_{j\neq i}$. Then, via (2.64), the i-th robot estimates the end-effector wrench collective vector ${}^{i}\hat{\boldsymbol{h}}_e$ and publishes it on a proper ROS topic. Then, by exploiting the information related to the desired end-effector trajectory and its own contribution to the internal wrenches, obtained via (2.70), each robot computes the reference trajectory in terms of position, velocity and acceleration via (2.79). This reference trajectory is published on a suitable ROS topic that the robot's low-level controller node subscribes to. Furthermore, if the reference trajectory is published at a lower rate than the robot state updater frequency, a ROS

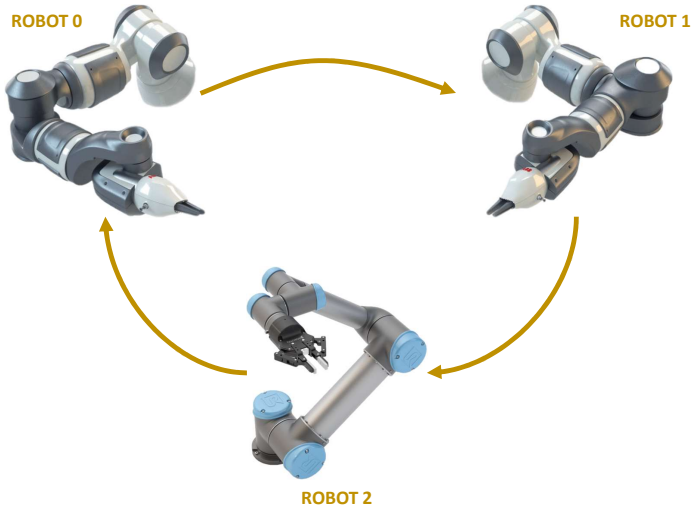


Figure 3.22 The communication graph consists of 3 nodes, each synchronizing the signal of the wrench sensed from its force-torque sensor, and the wrenches estimation of its neighbors, using the ROS *TimeSynchronizer* filter to account for delays and packet losses.

node uses a zero-order hold to deliver the trajectory inputs to each robot at a required frequency of 250 Hz.

Let us introduce an inertial frame $\boldsymbol{\Sigma}_w$ coincident with the YuMi dual-arm base frame. The desired object barycenter trajectory is designed to test the applicability of the proposed strategy and consists of a translation along an arc of circumference with radius $\rho = 0.15$ m and angle $\theta_p = \pi/6$ rad, and a rotation along the z-axis of an angle $\theta_o = \pi/6$ rad. The task has to be accomplished in 19 s, comprising of an initial steady state of 2 s, followed by 15 s of motion and 2 s of steady state. Since the admittance law (2.79) is expressed with reference to $\boldsymbol{\Sigma}_w$, the reference trajectory sent to the UR5e robot has to be transformed into the local frame, $\boldsymbol{\Sigma}_{ur}$, by means of the homogeneous transformation matrix $\boldsymbol{T}_{ur}^{w}$, expressing the pose of the UR5e base frame with respect to the world frame, obtained via an Intel RealSense camera attached to the UR5e robot and an Aruco marker [125]. The gains of the admittance equation (2.79) have been set to:

- $\boldsymbol{M}_i = \text{diag}\{2.5,\, 2.5,\, 2.5,\, 10,\, 10,\, 10\}$,
  $\boldsymbol{K}_{D_i} = \text{diag}\{25,\, 25,\, 25,\, 95,\, 95,\, 95\}$,
  $\boldsymbol{K}_{P_i} = \text{diag}\{10,\, 10,\, 10,\, 200,\, 200,\, 200\}$,

as concerns the YuMi robots, and

- $\boldsymbol{M}_i = \text{diag}\{5,\, 5,\, 5,\, 10,\, 10,\, 10\}$,
  $\boldsymbol{K}_{D_i} = \text{diag}\{25,\, 25,\, 25,\, 120,\, 120,\, 120\}$,
  $\boldsymbol{K}_{P_i} = \text{diag}\{15,\, 15,\, 15,\, 180,\, 180,\, 180\}$,

as regards the UR5e robot.

The damping gains have been tuned in order to guarantee a well-damped behavior while the stiffness matrices ensure a compliant response.

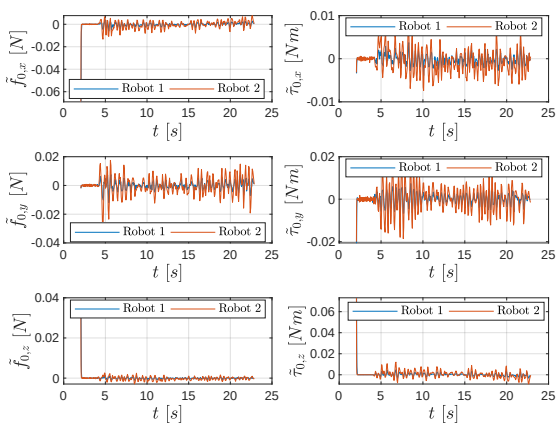
Two experiments[1] have been carried out in order to test the effectiveness of the estimator-controller architecture: in the first case study, only the low level (pure positional) controller has been adopted to control the robots cooperatively grasping the object; whilst, in the second experiment, the proposed two-level controller, incorporating the estimation scheme and the local admittance, has been exploited. In order to take into account sensor noise and non-perfect gravity compensations, a dead zone on the admittance filter input has been implemented. Thus, since any norm value of the force component below 0.5 N and any norm value torque component below 0.25 Nm will be neglected, to achieve a continuous wrench signal, the same thresholds are subtracted from the higher estimations.

Figure 3.23 shows the end-effector wrench estimation errors ${}^i\widetilde{\boldsymbol{h}}_k$ for each agent, normalized with respect to the norm of the corresponding measured wrenches $\boldsymbol{h}_k$. After a transient, errors remain bounded and lower than 2% throughout the whole experiment.

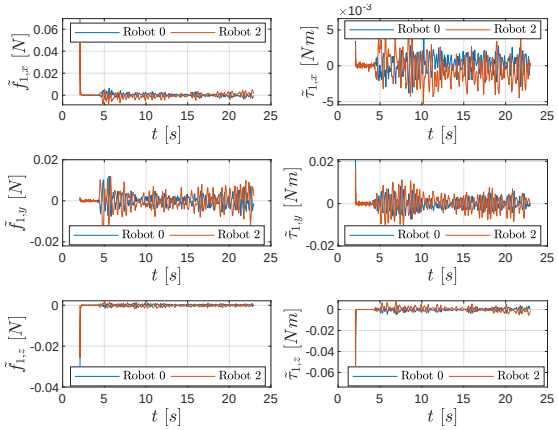
Figure 3.24 compares the internal force and torque contributions of Robot 2 on the object, $\boldsymbol{h}_{I_2}$, in both the experiments. The internal wrenches reduction experienced with the proposed approach is remarkable, as well as the boundedness of the internal forces and torques, according to Theorem 5. In the absence of the admittance control both $f_{Ix,2}$ and $\tau_{Iz,2}$ increase to dangerous levels (15N in force and 1.5Nm for the torque). Since the other robots exhibit similar behaviors, the related figures are omitted for brevity.

Conversely, this substantial decrease in internal wrenches does not result in any significant alteration in the trajectory commanded to the object, whose tracking errors are not affected by the presence of admittance controller, as depicted in Fig. 3.25. Errors are comparable with both approaches and can be mainly attributable to the

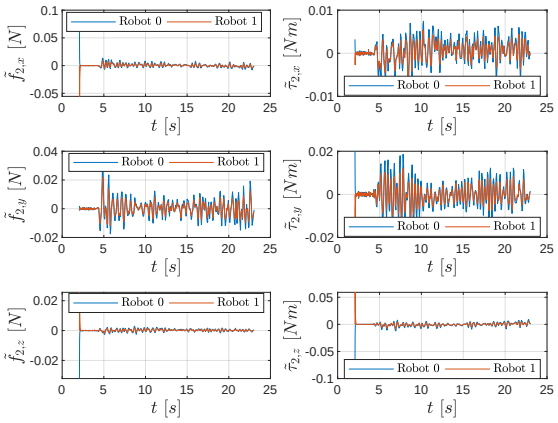
[1] A video of the experiment is at
https://www.youtube.com/watch?v=s3j1CUCVRqw



(a) Estimation error $\widetilde{\boldsymbol{h}}_0$ provided by Robot 1 and Robot 2.



(b) Estimation error $\widetilde{\boldsymbol{h}}_1$ provided by Robot 0 and Robot 2.



(c) Estimation error $\widetilde{\boldsymbol{h}}_2$ provided by Robot 0 and Robot 1.

Figure 3.23 Wrench estimation errors $\widetilde{\boldsymbol{h}}_k$.

low accuracy of the matrix $\boldsymbol{T}_{ur}^{w}$ and to the wrench model adopted in Assumption 6. Finally, low-level control tracking errors are not reported since they are negligible.

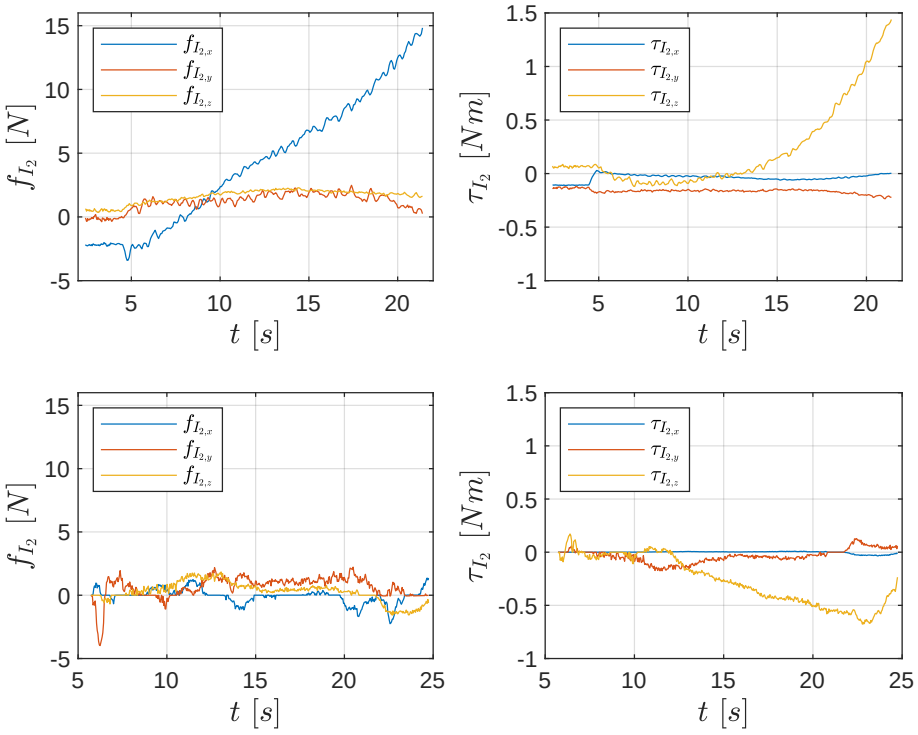
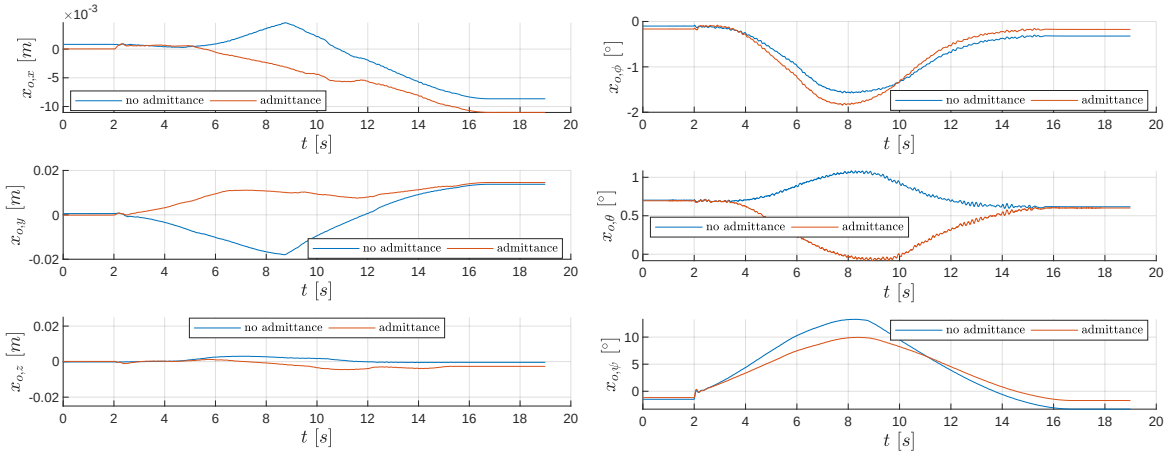


Figure 3.24 Internal force contribution (on the left side) and torque contribution (on the right side) for Robot 2, in both the case studies: pure positional controller (on top), internal admittance controller (on bottom).



(a) Object position $\boldsymbol{p}_o$ tracking errors.

(b) Object orientation $\boldsymbol{\phi}_o$ tracking errors.

Figure 3.25 Object tracking errors in the experiment with the pure positional controller (blue) and the admittance controller (orange).

3.2.4 Experiment with external and internal admittance control

Consider the cooperative robotic cell composed of $N = 3$ Franka Emika Panda arms, introduced in Section 3.1.3. Each agent is assumed to estimate the desired object trajectory, i.e., positions, orientations, and their time derivatives, according to Section 2.5.1, which serve as input to the external admittance controller. As in [126], the task under consideration consists of the distributed cooperative transportation of a rectangular wooden piece of dimensions $90 \times 45 \times 0.6$ cm and mass $m_o = 0.8$ kg, performed by the manipulators, as illustrated in Fig. 3.26.

Even though the Panda robot state can be updated at 1 kHz, the overall control scheme runs at a lower frequency due to synchronization requirements among the three ROS topics where the wrench signals are exchanged. Despite using the *Approximate*

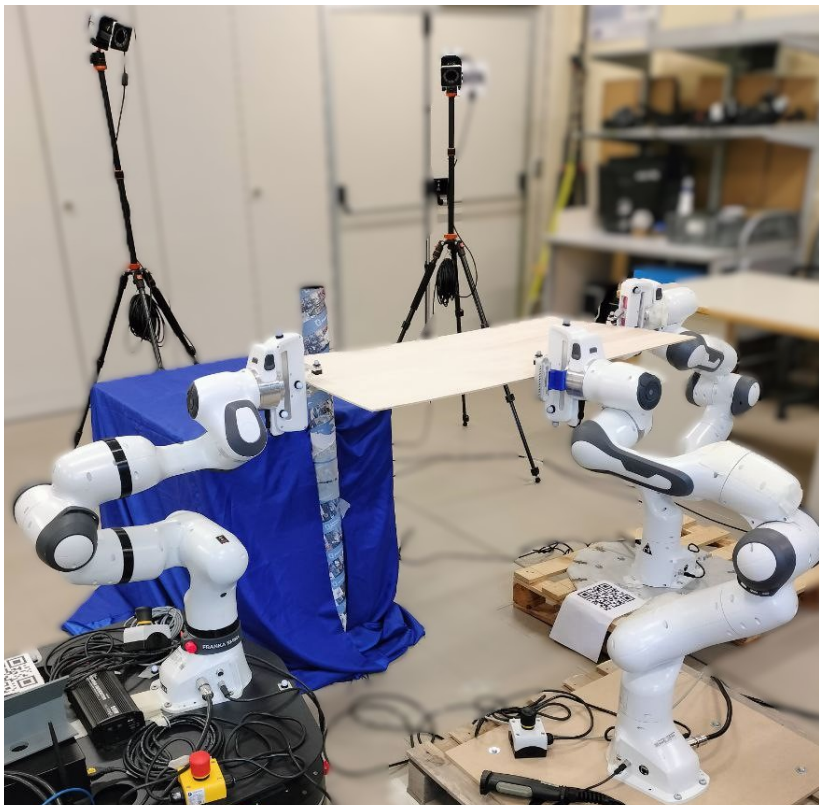


Figure 3.26 The experimental setup: the robots grasp and move the board. The base frame of the robot mounted on the mobile base is set as the world frame $\mathbf{\Sigma}_w$. A cylindrical obstacle is placed within the robot team's workspace, firmly fixed to a heavy support. The communication graph consists of 3 nodes arranged in a cycle graph, as in Fig. 3.12.

Time policy provided by the ROS message filters, this synchronization overhead imposes a lower effective control rate.

The adoption of this policy allows synchronizing two or more topics even when the time instants at which their messages are published are close to each other but not exactly the same. The behavior of the *Approximate Time* policy is shown in Fig. 3.27 [119], where 4 topics are assumed to be synchronized: time flows from left to right; each line represents a different ROS topic, and each bullet corresponds to a published message.

First, the policy identifies a pivot (red bullet) among the set of messages published on different topics, by selecting the latest message within a local temporal neighborhood. Then, it defines a candidate set of synchronized messages by discarding all newer ones belonging to the other 3 topics. Among the older messages, which form their respective queues, the best candidate is chosen as the newest one that satisfies the temporal constraints.

The attempt to synchronize more than 2 topics, each published by a different robot controller, can be affected by several issues:

- each agent may operate with a different *ROS time*, even when connected to the same *ROS_MASTER_URI*;
- different publishing frequencies: ROS does not guarantee real-time execution, meaning that robot controllers running at $\sim$ 1 kHz may produce messages at rates that are close, but not exactly identical;
- even if the publishing rates are the same, the corresponding signals may still exhibit different initial phases.

A practical approach to address the synchronization problem consists first in creating a simulated time, which eliminates the issue related to inconsistent ROS time across

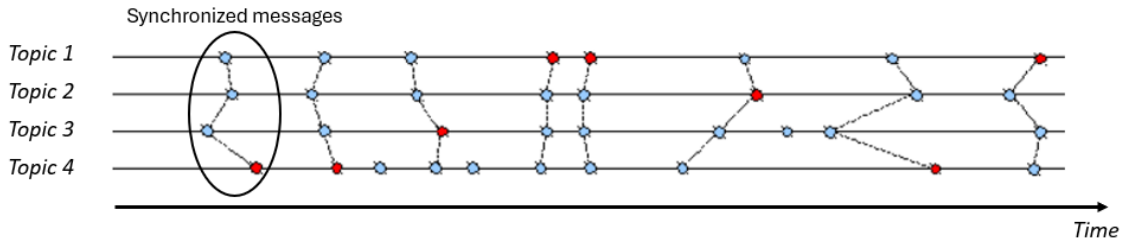


Figure 3.27 *Approximate Time* policy working method: each line that links a set of bullets across the multiple lines indicates a synchronized message.

agents; second, in adopting the Approximate Time policy provided by the ROS message filter; and finally, in reducing the publication rate of each robot controller—setting it to a lower frequency ($\sim$ 500 Hz)—in order to extend the time window within which the policy searches for messages to be synchronized.

Unlike the setup involving the ABB dual-arm Yumi robots and the Universal Robot UR5e (see Section 2.9), where the control frequency settles at approximately 250 Hz, the configuration adopted here exhibits different behavior. In the former case, the two ROS nodes associated with each arm of the ABB Yumi were essentially already synchronized, reducing the problem to the synchronization of only two messages at a time. Conversely, in the present experiment, it was observed that the control rate drops significantly and becomes unstable when attempting to synchronize more than 2 messages simultaneously.

Moreover, since the manipulators are not equipped with any force-torque sensor at their wrist, each robot uses a momentum-based observer [80] to measures its own wrench and estimates those of its teammates. A dedicated ROS node is assigned to each robot, enabling communication and data exchange according to the topology shown in Fig. 3.26.

As detailed in Section 2.10, the distributed scheme consists of three steps: a first *wrench estimation* phase, the external admittance filter, and finally the internal admittance filter.

At each time instant, the i-th robot acquires the wrench estimated by its momentum-based observer, $\boldsymbol{h}_i$, together with the wrench estimates from its neighbors, ${}^i\hat{\boldsymbol{h}}_{j\neq i}$. To account for communication delays and packet losses, these signals are synchronized using the ROS *TimeSynchronizer* filter. Then, based on (2.64), the i-th robot reconstructs the collective end-effector wrench vector, ${}^i\hat{\boldsymbol{h}}_e$, and publishes it on a dedicated ROS topic. From this information, the external generalized forces acting on the object center of mass are computed via (2.69), together with the i-th robot's contribution to the internal wrench via (2.70).

The estimated external wrenches are subsequently processed by the external admittance filter (2.91), along with the desired object trajectory. As a result, each robot computes the object reference trajectory in terms of position, velocity, and acceleration. This reference is then mapped through the closed-chain constraints, yielding the i-th end-effector trajectory. Finally, this trajectory, along with the i-th robot's internal wrench contribution, is fed into the local internal admittance filter (2.79), whose output

is published on a specific ROS topic to be subscribed by the robot's low-level controller node.

Let us introduce an inertial frame $\boldsymbol{\Sigma}_w$, coincident with the base frame $\boldsymbol{\Sigma}_1$ of the manipulator mounted on the mobile base (hereafter referred to as Robot 1). The desired trajectory of the object barycenter is designed to assess the applicability of the proposed strategy and consists of a translation along an arc of circumference with radius $\rho = 0.3$ m and angle $\theta_p = \pi/8$ rad, combined with a rotation about the z-axis of $\theta_o = \pi/8$ rad. The task is scheduled to be completed in 20 s: it begins with an initial steady-state phase of 2 s, followed by 15 s of motion, and ends with 3 s of steady state. At the initial time instant, the robotic team is already cooperatively grasping the object. During the execution, the object moves freely without environmental interactions until approximately $t = 11$ s, when it collides with a curved surface.

Since the admittance law (2.79) is formulated with respect to the world frame $\boldsymbol{\Sigma}_w$, the reference trajectories sent to Robots 2 and 3 must be transformed into their respective base frames, $\boldsymbol{\Sigma}_2$ and $\boldsymbol{\Sigma}_3$. This transformation is achieved through the homogeneous matrices $\boldsymbol{T}_2^w$ and $\boldsymbol{T}_3^w$, with similar approach of that described in Section 3.1.3.

Four experiments[2] were conducted to evaluate the effectiveness of the estimator–controller architecture presented in Section 2.10. In the first case study, the robots cooperatively grasping the object were regulated solely through the baseline low-level positional controller. In the second and third experiments, the external and the internal admittance filters were independently activated, thereby isolating their individual contributions. Finally, the fourth experiment employed the full proposed controller with both the external and internal admittance filters, demonstrating the complete capabilities of the proposed approach.

The gains of the external and internal admittance equations, (2.86) and (2.79), respectively, have been set to:

- $\boldsymbol{A} = \text{diag}\{10,\ 10,\ 15,\ 10,\ 10,\ 10\}$,
  $\boldsymbol{K}_o = \text{diag}\{15,\ 15,\ 25,\ 7.5,\ 7.5,\ 7.5\}$,
  $\boldsymbol{D}_o = \text{diag}\{75,\ 75,\ 75,\ 50,\ 50,\ 50\}$,

and

[2] A video of the experiment is at
https://web.unibas.it/automatica/multimedia.html

- $\boldsymbol{M}_i = \text{diag}\{15,\, 15,\, 15,\, 5,\, 5,\, 5\}$,
  $\boldsymbol{K}_i = \text{diag}\{45,\, 45,\, 50,\, 15,\, 15,\, 15\}$,
  $\boldsymbol{D}_i = \text{diag}\{80,\, 80,\, 80,\, 25,\, 25,\, 25\}$.

For both the object and the robots, the damping gains $\boldsymbol{D}_o$ and $\boldsymbol{D}_i$ were tuned to guarantee a well-damped response, while the stiffness matrices $\boldsymbol{K}_o$ and $\boldsymbol{K}_i$ were selected to ensure a compliant behavior. A dead-zone on both the admittance filter inputs has been applied in order to eliminate sensor noise and non-perfect gravity compensations.

Figure 3.28 shows the end-effector wrench estimation errors ${}^i\widetilde{\boldsymbol{h}}_k$ for each agent, normalized with respect to the norm of the corresponding measured wrenches $\boldsymbol{h}_k$. For compactness, only the results of the first case study are reported, as they are representative of the other cases as well. Remarkably, the impact between the object and the obstacle, which produces an impulsive increase in the applied wrench, does not compromise the estimation performance even in the absence of any admittance controller: errors remain bounded throughout the experiment and reach their maximum absolute value at the instant of collision, remaining $\leq 5\%$.

Positional Controller

In the first use case, no impedance filter is activated, and the robots directly track the desired trajectory derived from the closed-chain constraints in (2.54), neglecting any force feedback. As shown in Fig. 3.29, at the initial time instant the robots experience a positive external torque along the z-axis, which is almost completely nullified upon contact with the obstacle. This effect arises because the rotational motion around the z-axis terminates at impact, leading to a near cancellation of torques, with only a small residual sliding along the curved contact surface. The geometry of this surface allows the object to slightly continue its rotational motion around the vertical axis.

At the same time, the collision induces an increase in the external forces within the x–y plane, as expected. In particular, two peaks in the external force along the y-axis are observed. These peaks can be attributed to slipping of the object between the fingertips of one of the robots, positioned opposite the obstacle with respect to the object's barycenter. This is followed by a phase in which the external forces remain constant. Consequently, the internal force along the y-axis decreases significantly, leading to a relaxation of both the internal force and the internal torque squeezing the object.

Fig. 3.30 reports the time histories of the object's position and orientation errors. Upon contact with the obstacle, the position error exhibits an instantaneous increase before subsequently decreasing. In contrast, the curved contact surface mitigates variations in the orientation error, allowing it to remain consistently low throughout the interaction.

External admittance

The adoption of the external admittance filter considerably reduces the external torques applied to the object during the rotational phase preceding the contact with the obstacle (Fig. 3.31). After contact, the external torque signal becomes smoother and settles at a constant value during the final steady-state phase. A similar behavior is observed in the external forces: along the x-axis, they remain within the dead-zone, whereas along the y-axis an impulsive increase occurs due to the collision, followed by stabilization at a constant value. Compared to the positional control case, the external forces after the collision are significantly reduced.

As expected, internal forces and torques do not exhibit substantial deviations from those observed in the pure positional control scenario. However, after contact, their absolute values remain lower, and no impulsive drop associated with the object slipping through the robot grippers is observed. Consequently, both internal forces and torques remain nearly constant until the end of the experiment.

The reduction of the external wrenches results, as expected, in larger position errors—mainly along the y-axis, which experiences the highest external force—and orientation errors around the z-axis (Fig. 3.32). These error values remain approximately constant during the final steady-state phase.

Internal admittance

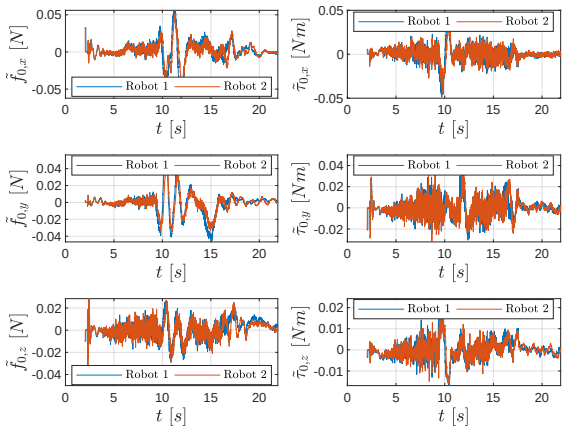
Activating the internal admittance filter significantly reduces internal forces and torques, as shown in Fig. 3.33. Before the contact with the obstacle, both the internal forces in the horizontal plane and the internal torques around the vertical axis do not exhibit the increasing trend observed in the previous case studies, but instead remain bounded. Due to the absence of the external admittance, however, the collision leads to peaks in the internal force along the y-axis and the internal torque around the z-axis, caused by slight slipping of the object between the robot grippers.

A slight coupling between internal and external torques can be noticed when analyzing the external force along the z-axis and the object errors (Fig. 3.34): no force values leave the dead-zone, unlike in the previous two cases, where such deviations are likely due to deformations of the object caused by higher internal forces and torques. Moreover, the external forces remain constant, since, although the position and orientation errors are clearly lower than in the external admittance case, they are still higher than in the positional control case. As a result, the object pushes against the environment surface, leading to high forces and torques.

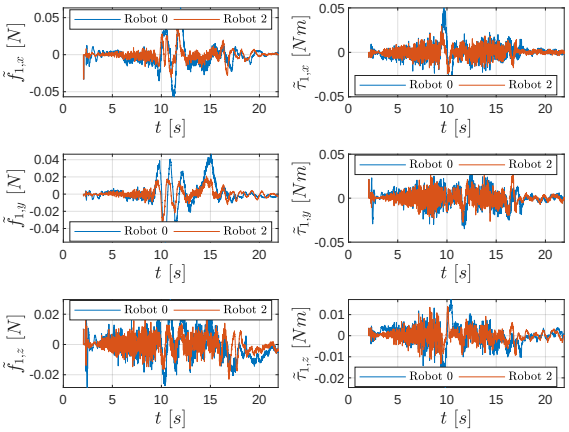
External and internal admittance

In the last case study, both the external and the internal admittance are activated. As a consequence, the external wrenches exhibit a behavior nearly identical to that of the second case study (external admittance only), as do the position and orientation errors, confirming the decoupling between the two filters (Figs. 3.35 and 3.36). Since no actual impact with the external environment occurs—and consequently no slipping of the object takes place—internal forces and torques remain bounded and very low in absolute value throughout the entire experiment.

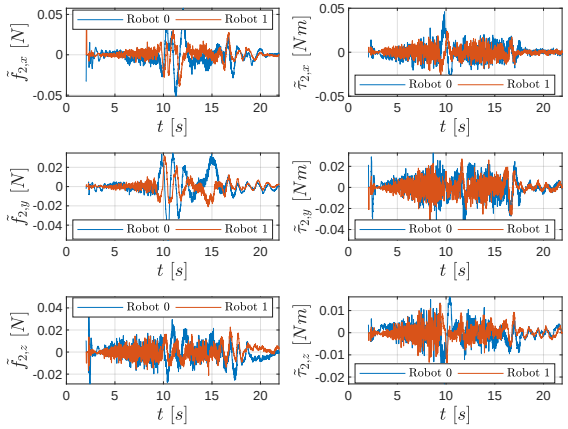
The object's flexibility leads to the presence of external torques along the z-axis and external forces along the x- and y-axes. A slight misalignment of the robots with respect to the vertical axis causes the appearance of an internal torque around the x-axis.



(a) Estimation error $\widetilde{\boldsymbol{h}}_0$ provided by Robot 1 and Robot 2.



(b) Estimation error $\widetilde{\boldsymbol{h}}_1$ provided by Robot 0 and Robot 2.



(c) Estimation error $\widetilde{\boldsymbol{h}}_2$ provided by Robot 0 and Robot 1.

Figure 3.28 Wrench estimation errors $\widetilde{\boldsymbol{h}}_k$.

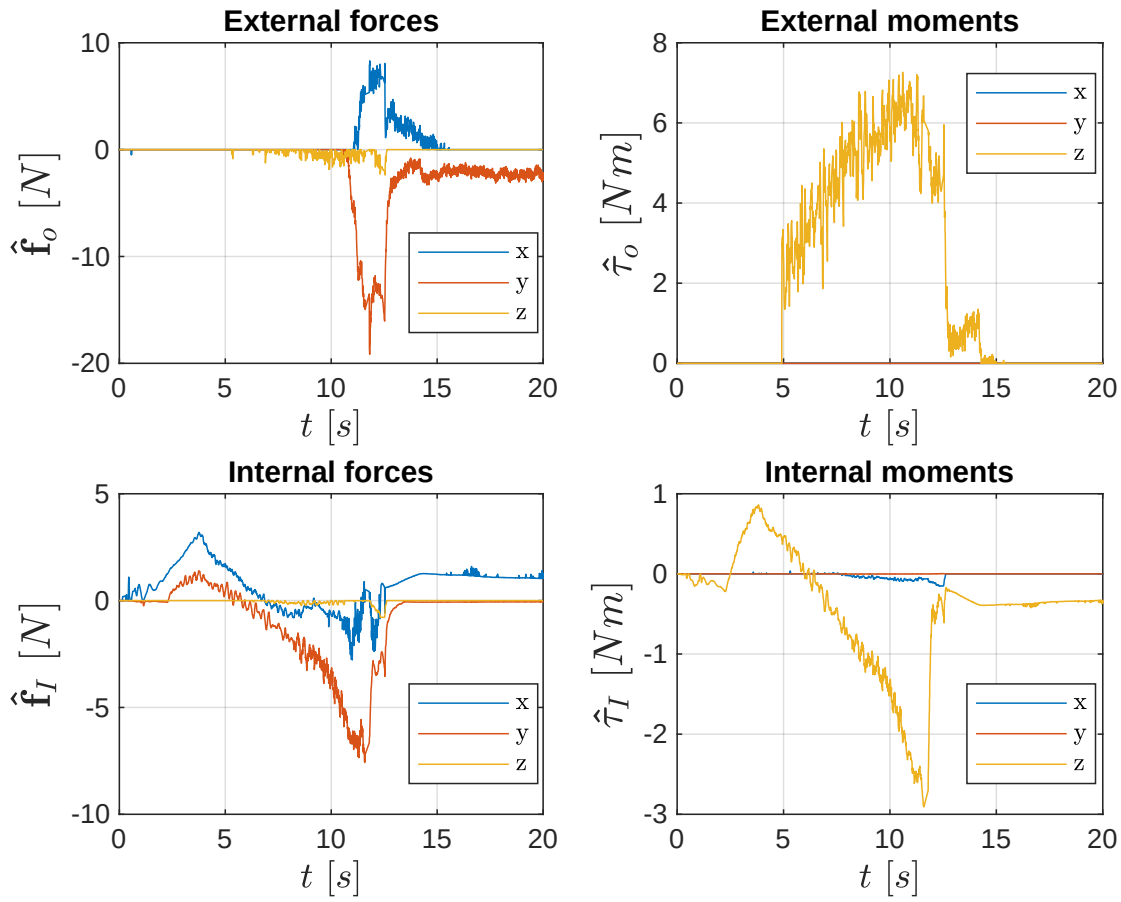


Figure 3.29 *Positional controller* case study: external wrenches (top) and internal wrenches (bottom).

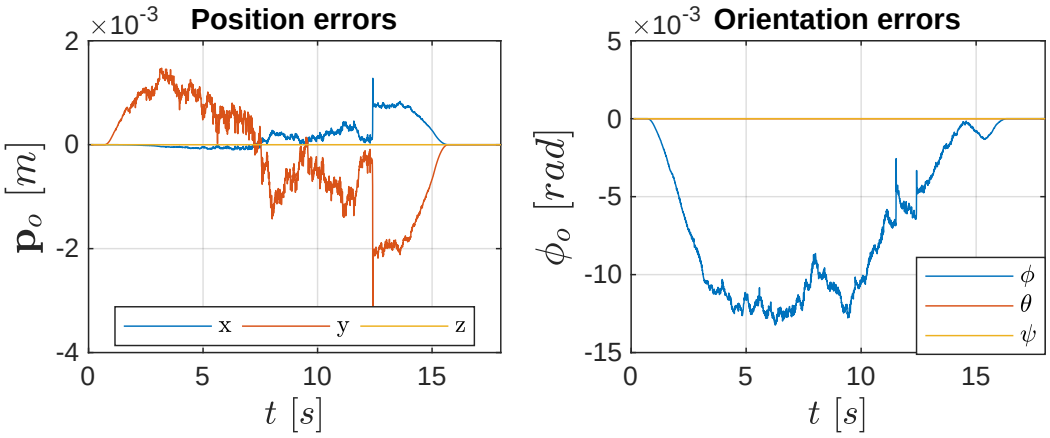


Figure 3.30 *Positional controller* case study: position errors (left) and orientation errors (right).

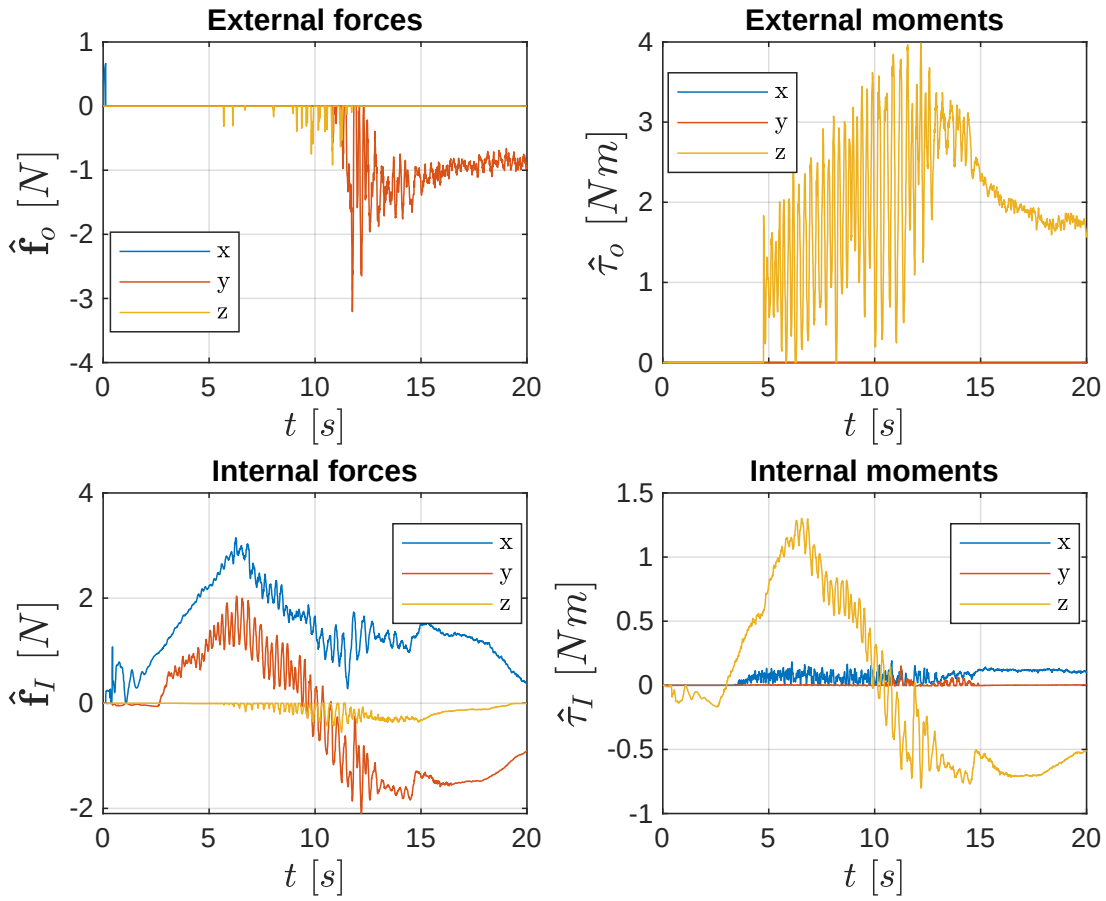


Figure 3.31 *External admittance controller* case study: external wrenches (top) and internal wrenches (bottom).

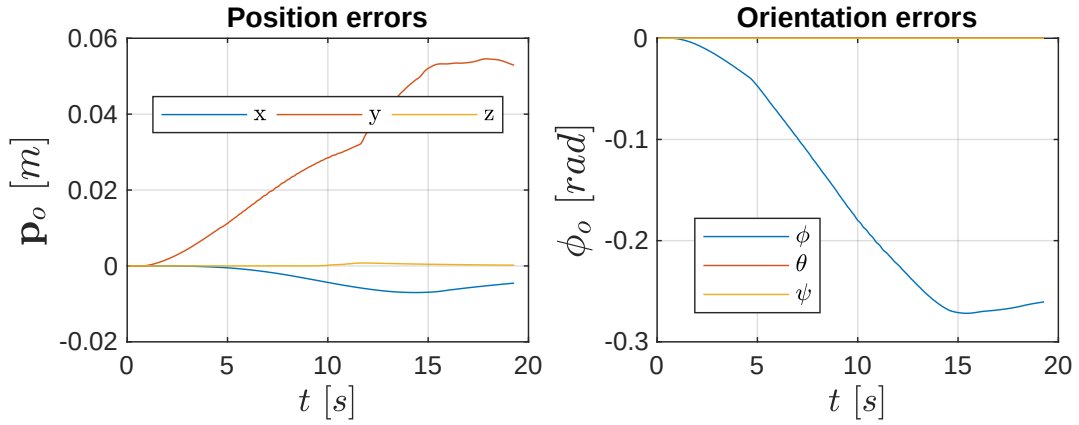


Figure 3.32 *External admittance controller* case study: position errors (left) and orientation errors (right).

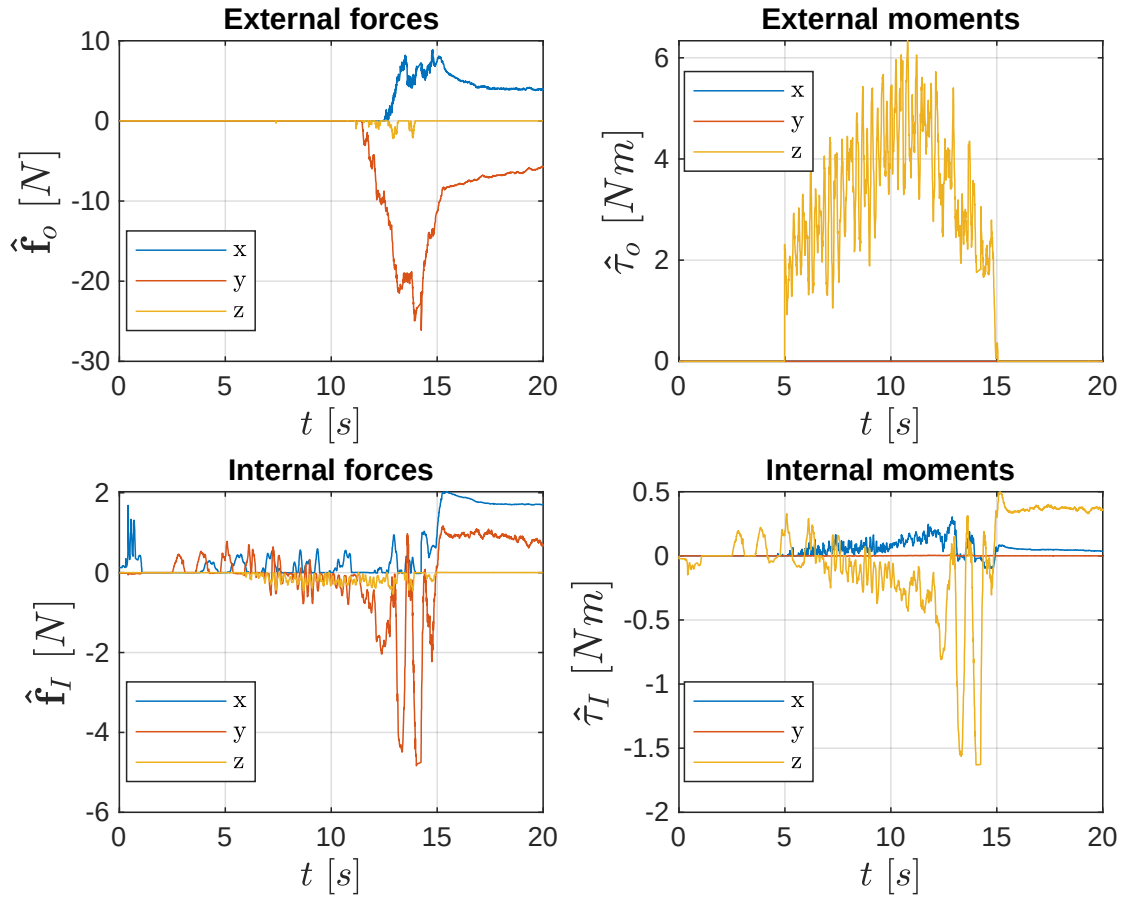


Figure 3.33 *Internal admittance controller* case study: external wrenches (top) and internal wrenches (bottom).

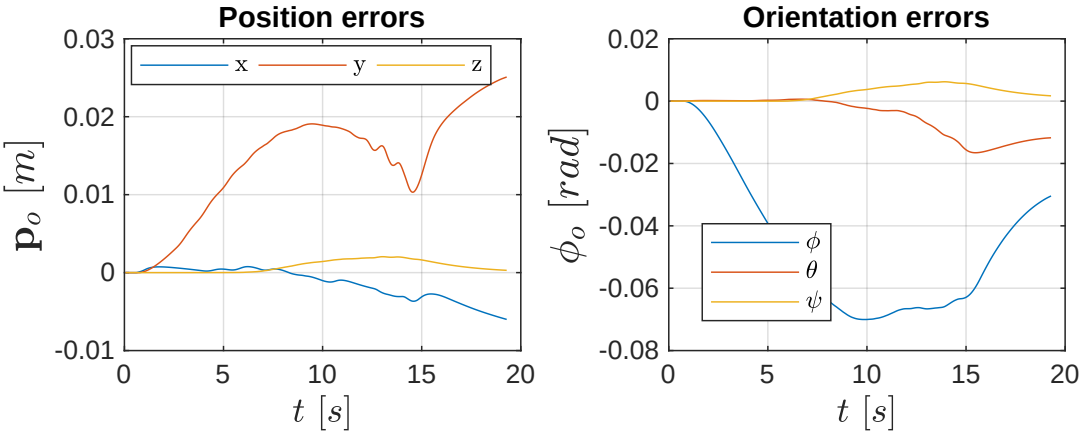


Figure 3.34 *Internal admittance controller* case study: position errors (left) and orientation errors (right).

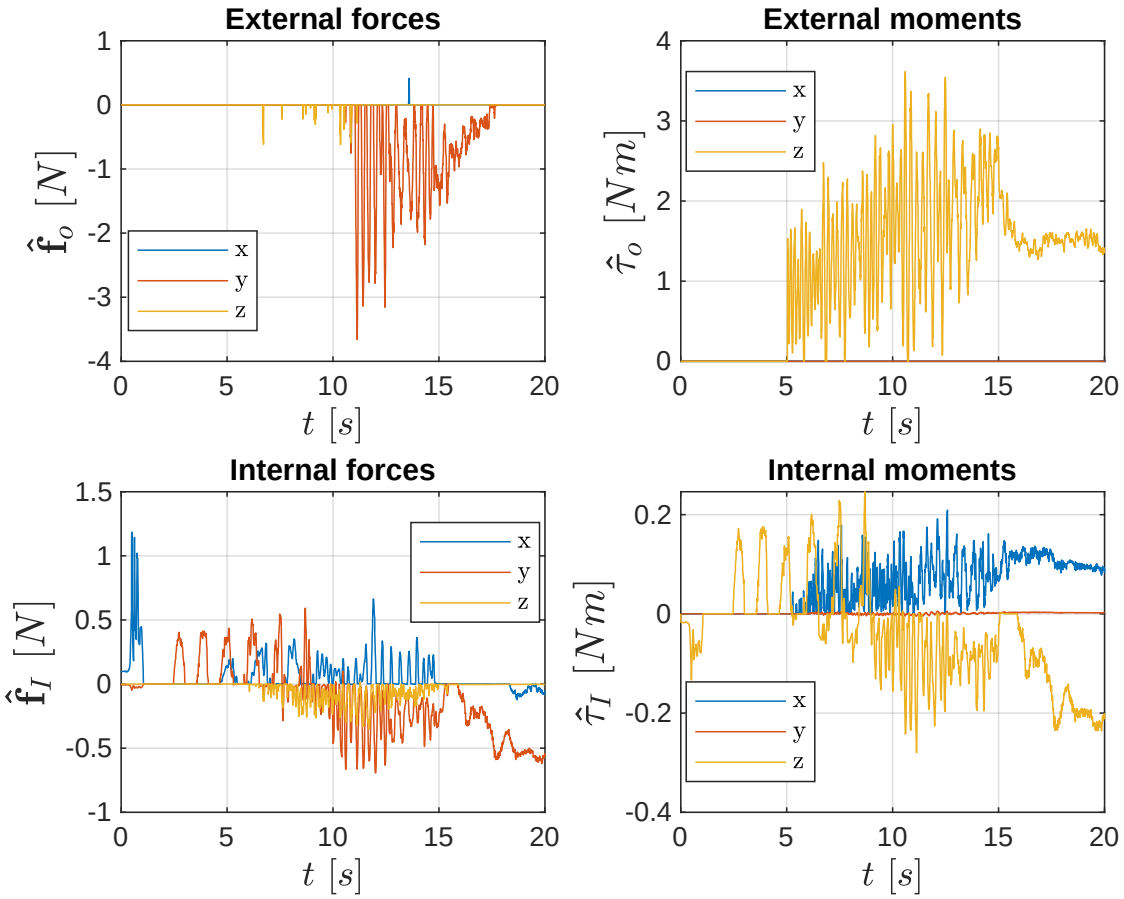


Figure 3.35 *External and internal admittance controller* case study: external wrenches (top) and internal wrenches (bottom).

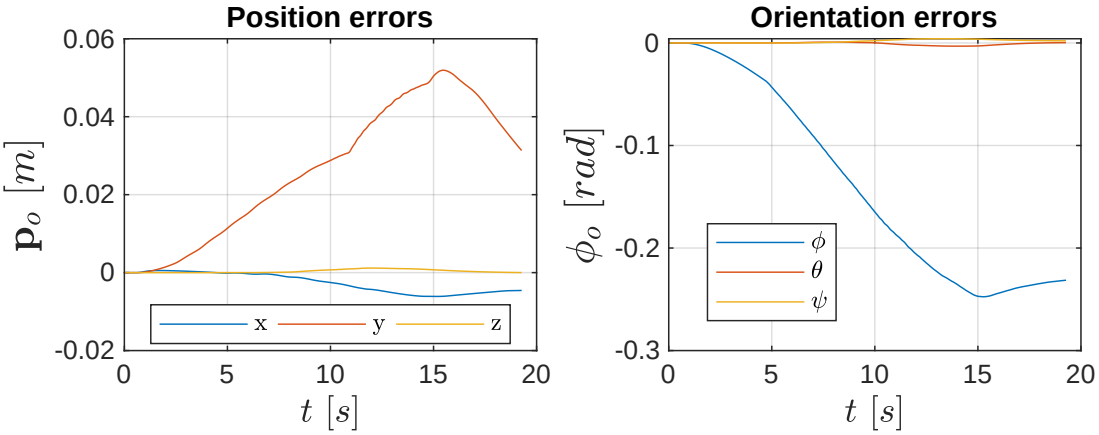


Figure 3.36 *External and internal admittance controller* case study: position errors (left) and orientation errors (right).

Chapter 4

Human-Robot Collaboration

The base concept of Industry 4.0 paradigm is the development of autonomous technologies involving robots working in close collaboration with human operators [127, 128]. Sheridan proposes a classification of the autonomy level of robots by assigning value 1 to the case of *Computer offers no assistance* and value 10 to the totally autonomous case of *Computer decides everything and acts autonomously, ignoring the human* [129]. At the intermediate levels, Human-Robot Collaboration (HRC) is involved, requiring the adoption of collaborative robots [130]. A key point of collaborative robotics is the information exchange, which can be *uni-directional* if the robot communicate with the human or vice versa, or *bi-directional* if a two-way stream is established. More in detail, the human shall be able to know the status of the robot and predict in real time its intentions and, on the other side, she/he shall give instructions and commands to the robot to proactively change and adapt the execution of its tasks.

Effective bi-directional HRC can be developed by introducing interfaces on Virtual Reality (VR) or Augmented Reality (AR) environments, which can allow human operators, without particular experience or knowledge in robotics, to easily interact with robots. Several studies showed the use of AR improves the task efficiency and the HRC [131]. To this aim, the following section presents a comprehensive overview of existing HRI platforms, focusing on the applications implemented via AR devices.

4.1 Platforms for Human-Robot Interaction

In Human-Robot Interaction systems, agency cannot be exclusively attributed to humans but it is distributed among humans and non-human resources [130]; interaction is exploited by assigning operators and robots different sub-tasks leading to a common

goal (*cooperation*), or a sequence of tasks accomplished together toward a shared goal (*collaboration*) [132, 133]. In the latter case, the envisioned goal is to make both human operator and cobot proactive, providing a bi-directional stream of exchanged information. For this purpose, hand gesture recognition could help humans to convey information and express intentions in a simple way. Relevant advantages of this solution with respect to traditional human-machines interaction, for instance mouse, keyboard and 2D display requiring a fixed operating space, are [134]: high degree of differentiation, strong flexibility and high efficiency of transmission, achievable through data gloves, ultrasound, vision or wearable devices based on electromyographic signals.

In [135], a novel Augmented Reality teleoperation system is proposed, where the 3D virtual robot model tracks dynamically the current real robot state. The system architecture is composed by a multi-DoF industrial robot and two RGB-D sensors, connected each other in the ROS framework and to the Augmented Reality (AR) interface, built with head and hand gesture trackers, by means of ROSBridge server. The system's maneuverability is enhanced thanks to this setup which guarantees interaction with the environment by hand gestures, tracked using a Leap Motion controller mounted on the Head-Mounted Display (HMD). An "interaction proxy" manages hand gestures, making task performing easier and avoiding unintentional gesture disturbances. Moreover, two control modes (coarse movement mode and fine movement mode) are proposed to prevent operational faults related to each type of movement.

In [136], the adoption of Microsoft HoloLens device allows for bi-directional communication with cobot both via visual and acoustic signals. The visual ones make the operator aware of robot movements, increasing her/his trust in collaboration, whereas the acoustic feedbacks are useful when the 3D holograms are out of sight or for adverse lighting conditions. The development tool is Unity, along with Mixed Reality Toolkit [137], while the robot is controlled through ROS. The communication exploits the User Datagram Protocol (UDP) and it is designed aiming at information exchange reduction, preventing communication overload, as well. Indeed, the operator wearing HoloLens receives information about the robot state both in terms of positions and contact forces. A light with a colour code improves the operator safety: the color is green, if the robot moves away from the operator; red, if the robot approaches the operator; yellow, if the robot is waiting for user input. On the other side, the operator sends inputs to the robot through hand gestures, in order to select the control strategy and the task to be performed. A polishing task is executed comparing 2D display and

AR systems: the AR interface provides a better feedback, allowing the user to reduce his effort during the task execution.

In [138] an AR system that outperforms the 2D display is proposed, where the AR interface reduces task completion time (about 38%), increases accuracy of collision prediction by an average of 15% and precision by 11%. In order to map the environment on real time and collocate the operator inside the AR reconstruction, the system set up consists of Microsoft HoloLens headset, an inertial measurement unit, four cameras and an infrared (IR) depth sensor. The user is immersed in a virtual space, perceiving the scene within the HoloLens device and interact with it through hand gestures (through the Unity engine) and voice commands. One-hand and two-hand gestures are exploited in order to specify the desired position and orientation of robot end-effector and a voice command instructs the robot to plan the trajectory, which is validated by the operator through AR. If the operator notices that an object collision could occur, she/he replans the trajectory with a stochastic planner.

The same approach is proposed in [139], where an AR interface supporting robot programming is presented: pick-and-place tasks are designed using sequences of way-points. In this way, 3D trajectories are generated and eventually altered by moving, adding and deleting the way-points. Furthermore, the interface allows the users to visualize the assigned path that the robot would take from two consecutive way-points, which is planned by means of MoveIt!. Robot programming through AR platforms is also considered in [74], where two trajectory generation methods are proposed: free space trajectories and surface trajectories. In the first case, the user selects the pick and place locations in the physical workspace through speech and gesture inputs that are recognized by HoloLens. A virtual scenario allows the user to dynamically modify the path in free space during the simulation or even during the execution. On the other hand, virtual paths can be defined on a surface by using head tracking, speech and gesture recognition. The robot end-effector can then move along the path while keeping its orientation normal to the surface, thanks to an impedance-controlled mode, which ensures the application of a constant normal force.

Both in [139] and in [74], the usability of proposed AR systems is evaluated and compared to classical approaches. In [139], volunteers have been instructed to program the robot to pick up and place a cube onto a platform, bringing it over walls of different heights. The task was completed using two different interfaces: a 2D monitor interface and the HoloLens AR application. In [74], participants were instructed to perform two tasks using the proposed AR platform and a kinesthetic teaching interface, i.e. a

gamepad for storing way-points: a surface contact task, consisting in the erasure of lines drawn on a flat surface by means of a marker eraser fit in the robot end-effector, and a free space task, i.e. a pick-and-place in the presence of an obstacle. In both cases, significant reductions of the teaching time and the NASA Task Load Index [140], combined with improvements of performance and usability supported the use of the AR systems for novice users.

In [141], AR has been adopted to improve the navigation of a mobile robot and visualize trajectory and spatial mapping. The setup consists in Microsoft HoloLens, as AR device, and Kuka Youbot, as mobile robot. The development platform used for HoloLens is Unity 2018, while the robot is controlled through ROS Hydro. The communication between ROS and HoloLens is ensured by the ROSBridge protocol, which follows a specific JavaScript Object Notation (JSON) notation for messages. Furthermore, the ROSSharp library was used to allow HoloLens to follow ROS publish and subscribe approach. ROSSharp is a set of open-source software libraries and tools in C# for communicating with ROS from .NET applications, in particular Unity.

Furthermore, AR can be adopted to test programmed tasks via a digital twin: real time interaction between real and virtual world allows for programming by demonstration avoiding unsafe situations.In the system developed in [142], basic and advanced functionalities about HRC are implemented:

- robot communicates its status, namely its joint positions and pose by AR interface;
- human plans a geometrical path specifying a sequence of way-points, i.e. end-effector position and orientation, by using the framework *MoveIt!*, and choose the 3D geometrical primitive which connects each goal point (e.g. an arc, a line); trajectory feasibility is then checked in joint space;
- it is possible to visualize robot workspace (WS) in AR, assisting human to assess points achievability, warning her/him in the presence of an obstacle and slowing down robot movements in case the operator shares WS with robot;
- for a correct virtual model of the robot placement, an algorithm based on point clouds processing is developed, which analyzes the scene, searches for robot-like objects and finally finds the position of the robot;
- during the trajectory generation phase, the shortest path to connect the sequence of goal points in joint and Cartesian space is automatically planned, while Unity engine API provides object collision avoidance and path scaling.

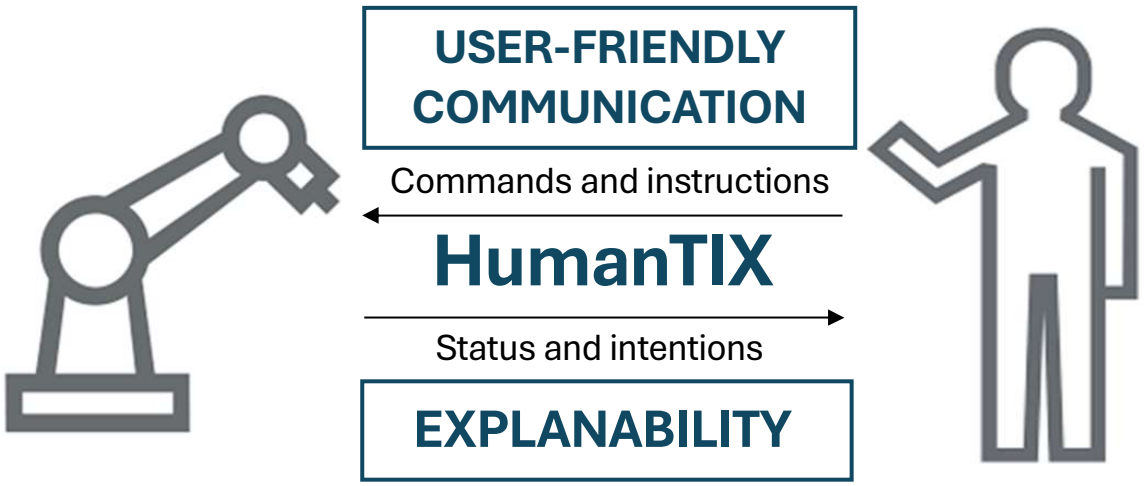


Figure 4.1 Bi-directional communication stream between Human and Robot established through HumanTIX platform.

4.2 HumanTIX: an Augmented Reality platform for Human-Robot Interaction

An AR application designed for HoloLens 2 [143], *HumanTIX*, has been developed to be employed in various research fields, specifically in the industrial one. This project is built on a platform, presented in [144], previously developed in the Automation, Robotics and Applied Electromagnetism (AREA) Laboratory of the University of Basilicata. The provided AR features have been widely broadened. Firstly, as regards to the conceptual point of view, in [144] the architecture provides only one-way communication between the robot Franka Emika Panda and the human operator. Conversely, the introduction of new features has extended the communication to a bi-directional stream: the human operator is allowed to actively interact with the robot, giving it inputs and commands. Conversely, the robot is in charge of communicating its status and intentions through the AR platform (Fig. 4.1).

The adoption of AR has many benefits for effective Human-Robot Interaction. Following the taxonomy in [73], the purposes of the proposed application can be categorized as:

1. Facilitate Programming;
2. Improve Safety;
3. Communicate Intent.

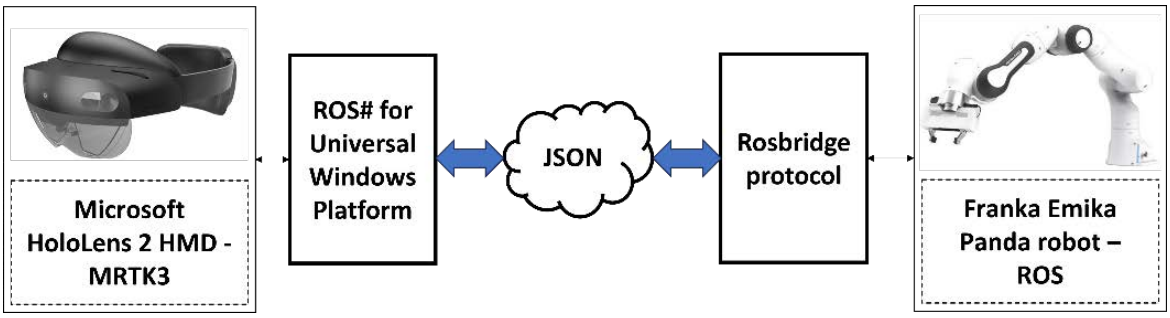


Figure 4.2 System Architecture.

The whole platform is designed for a human operator in strict contact with a robot and represents a tool for tackling industrially relevant robotic tasks that require human-robot interaction. It aims to handle real industrial needs by facilitating the robot programming. More in detail, it is designed in such a way:

- to make low-skilled operators able to program the robot for complex collaborative tasks;
- to make the operator aware of the robot intention by showing the planned trajectory in the AR environment and the robot workspace, in order to support a safe interaction in the absence of exteroceptive sensors;
- to improve the flexibility in trajectory planning in unstructured environments.

Finally, a usability test, conducted through an extensive experimental campaign involving 45 voluntary users with different levels of expertise, has assessed the effectiveness of the presented platform compared to a classical kinesthetic teaching task. The System Usability Scale survey [145] has been carried out in the experimental campaign and confirmed an improvement in the users' preference for the HoloLens platform compared to the classical programming interface. According to the results, the users' confidence in using the robot, as well as the general usability have increased. In addition, the perceived workload, computed by resorting to the NASA Task Load Index method [140], pointed out less physical demand, at the expense of a slightly higher cognitive demand, with respect to the classical programming interface.

4.3 Development Tools and Experimental Setup

The structure of the proposed platform is made up of several elements, as shown in Fig. 4.2. The main components are the aforementioned Microsoft HoloLens 2 Head-

Table 4.1 Hardware and Development Tools.

Head-Mounted Display	Microsoft HoloLens 2
Robot	Franka Emika Panda
Unity	Unity 3D v2021.3.22f1 64bit
Mixed Reality Toolkit	MRTK v.3
ROS Version	Kinetic

Mounted Display and the 7-DoF collaborative robot Franka Emika Panda. The HMD runs on the Windows Holographic operating system and has a Qualcomm Snapdragon 850 ARM64 CPU, 64 GB of internal storage, and 4 GB of RAM, along with a specialized processor for hologram visualization (HPU 2.0). An external workstation is connected to the robot via Ethernet, to control its motion. This is achieved through a ROS node from the *franka_ros* package, which enables real-time command exchange and status monitoring at a frequency of 1 kHz. The application is developed using the Unity3D environment [146], which is enhanced with the version 3 of the Microsoft Mixed Reality Toolkit (MRTK3) library [137]. The adoption of the ROS# library, a Siemens open-source initiative for Unity, facilitates data exchange between the robot and HMD [147]. The platform uses the ROSBridge protocol to convert messages from C# into the JavaScript Object Notation (JSON) format, ensuring efficient data inter-change and integration of these varied elements into a cohesive, functional system.

Finally, the Unified Robot Description Format (URDF) of the robot is inherited and then imported in Unity. It is suitably placed in the virtual environment by means of a calibration procedure, which exploits the SampleQR codes application provided by Microsoft. In Table 4.1, a comprehensive list of the adopted hardware and development tools is reported.

4.4 Design of the Augmented Reality Platform

The Mixed Reality Toolkit is exploited to handle the generation of virtual panels and buttons, such as the main menu (Fig. 4.3a). It consists of a panel with three default buttons, respectively dealing with:

- the real time supply of robot status, namely joint positions, velocities and torques, collected by the ROS system [144];

- the Human-Robot Interaction modality, which provides access to the AR feature of the platform;
- the exit from the application environment.

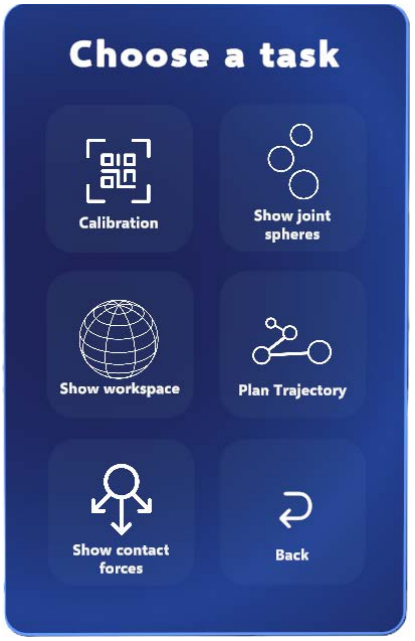
More in detail, the user is redirected to a new AR scene through the "*Human-Robot Interaction*" button on the main menu, and the Human-Robot Interaction panel is subsequently displayed, as shown in Fig. 4.3b. Specifically, this panel allows the user to overlay real-world information with data provided by the augmented reality, such as the visualization of the robot's workspace (Section 4.5.3), a qualitative assessment of the distance from joint limits (Section 4.5.2), and the forces exchanged at the end-effector.

The exploitation of these features requires the preliminary execution of a calibration procedure. Through the QR-code anchor, a correspondence between the real and virtual environments is established, guaranteeing an accurate overlap between the physical manipulator and its holographic counterpart. Consequently, only the exit button is activated before the calibration procedure, ensuring a more user-friendly application experience (see Section 4.5.1).

The final goal of this application is to ensure a simple and intuitive assignment and subsequent execution of a robot trajectory within its workspace. To this aim, once the "Plan Trajectory" button is selected, a new scene appears entirely dedicated to the Trajectory planning: the correspondent panel, shown in Fig. 4.3c, comprises 10 different buttons:

- "*New Trajectory*" button sends command to the ROS node to calculate a linear trajectory for the end-effector, reaching the target pose, starting from the actual one (see Section 4.5.4);
- "*Load Trajectory*" and "*Save Trajectory*" handle the upload of a previously saved trajectory or the recording of a novel one, respectively.
- "*Update Trajectory*" button allows the human operator to send the ROS node the way-points of the previously computed trajectory after her/his visual validation (see Section 4.5.5);
- "*Clear way-point*" button resets the position assigned to the way-points, restoring the linear trajectory, initially defined (see Section 4.5.5);

(a) Main Menu.



(b) Human-Robot Interaction panel.

(c) Trajectory planning panel.

Figure 4.3 HumanTIX virtual panels for Human-Robot Interaction.

- through the "*Simulate trajectory*" and "*Move Robot*" button, the operator commands the virtual and the real robot, respectively, to track the assigned trajectory (see Section 4.5.6);
- "*Add wall*" and "*Destroy wall*" buttons allow the user to build or delete a virtual wall splitting the robot workspace in twice: the manipulator part and the human one;
- "*Back*" to come back to the application main menu 4.7.

4.5 Augmented Reality Features

In this Section, all the features offered by the application will be detailed. They have been designed to allow low-skilled workers to plan a robot trajectory, suitably modify it, and finally verify its feasibility through simulation prior to execution on the physical robot.

4.5.1 Calibration

The pose of the robot virtual model in AR environment is assigned through a calibration procedure. More in detail, the calibration is needed to align the virtual features with the real environment. It consists in transferring a fixed frame, whose relative position and orientation with respect to the robot base frame is known, from the real to the augmented environment. A dedicated algorithm has been developed to evaluate the calibration accuracy. It is based on a quantitative evaluation of the position and orientation errors of the virtual hologram with respect to its real twin. In Fig. 4.4 two snapshots showing the two steps of the calibration procedure are reported: the robot hologram collocation phase (Fig. 4.4a) and the subsequent validation phase (Fig. 4.4b).

The calibration procedure requires the placement of two QR codes in the user's field of view in known poses with respect to the real robot base frame, denoted as $\mathbf{\Sigma}_r$. The first QR code allows to align the robot hologram in the AR environment, while the second one permits to numerically evaluate the procedure success and compute the calibration errors in position and orientation.

Denote with $\mathbf{\Sigma}_q = \{O_q, x_q y_q z_q\}$ and $\mathbf{\Sigma}_v = \{O_v, x_v y_v z_v\}$ the coordinate frames attached to the top left corner of the two QR codes, respectively. Based on the previous

assumption, the relative positions ($\boldsymbol{p}_q^r$ and $\boldsymbol{p}_v^r$) and orientations (expressed in terms of rotation matrices [101] as $\boldsymbol{R}_q^r$ and $\boldsymbol{R}_v^r$) between the QR codes' frames and the robot base frame are constant and known in the real world. The superscript r indicates that all the quantities are expressed with respect to the robot base frame.

When the two QR codes are in the user's field of view, the operator presses the "*Calibrate*" button, shown in Fig. 4.3b. At this point, HoloLens starts scanning the surrounding environment for QR codes (Fig. 4.4a), by exploiting the Unity QR code tracking algorithm. If the pursuit is successful, the QR tracking algorithm returns an estimation of the positions, $\hat{\boldsymbol{p}}_q$ and $\hat{\boldsymbol{p}}_v$, and the orientation matrices $\hat{\boldsymbol{R}}_q$ and $\hat{\boldsymbol{R}}_v$, of the QR frames, $\hat{\boldsymbol{\Sigma}}_q$ and $\hat{\boldsymbol{\Sigma}}_v$, in the HoloLens coordinate frame, $\boldsymbol{\Sigma}_0$, defined by default at each start of the application (the superscript 0 is omitted for notation compactness). Once the calibration succeeds, the application notifies the operator with an acoustic signal, and the robot hologram can then be placed within the AR environment.

Thus, the robot hologram is firstly aligned with $\hat{\boldsymbol{\Sigma}}_q$; subsequently, it is translated according to the vector $-\boldsymbol{p}_q^r$ and rotated through the rotation matrix $\boldsymbol{R}_r^q = \boldsymbol{R}_q^{r\mathrm{T}}$. For example, in the setup developed in this work, shown in Fig. 4.4a, the matrix $\boldsymbol{R}_r^q$ is given by:

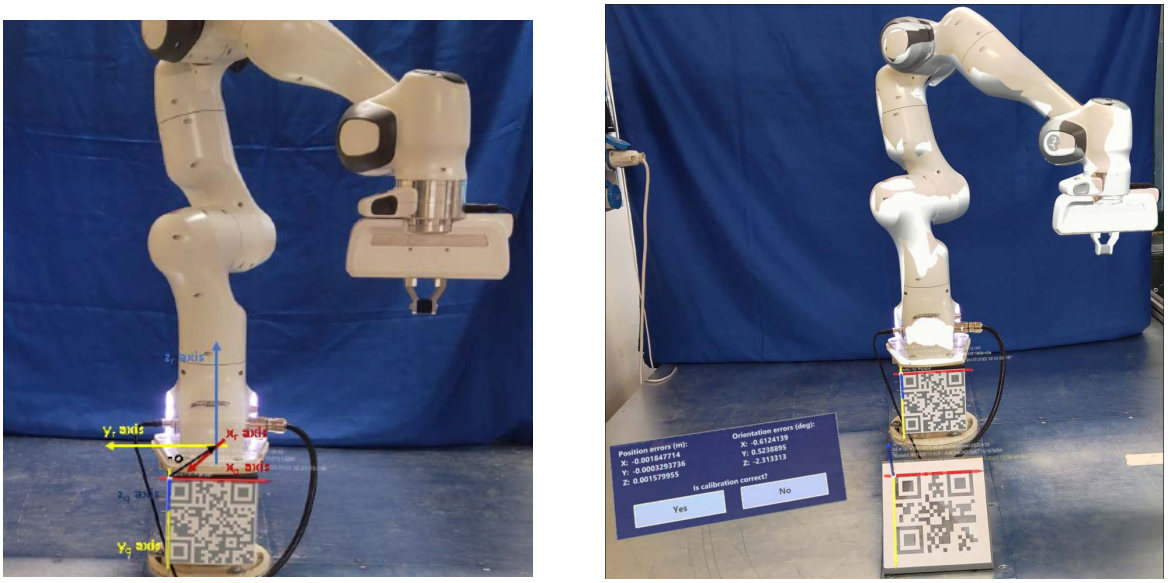
$$\boldsymbol{R}_r^q = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{bmatrix}. \tag{4.1}$$

In this way, the coordinate frame of the hologram is expected to be coincident with the real robot base frame. The QR code pose detection could be affected by the perspective distortion due to the relative height of the human operator point of view with respect to the QR frame during the calibration phase. Hence, if HoloLens perceives the QR plane as tilted with respect to the vertical plane, a further rotation is added in order to compensate this effect and make the QR code frame always aligned with the robot base.

Calibration errors assessment

Once the hologram is placed in the AR environment, the user must assess the calibration accuracy. For this reason, she/he has to evaluate the pose of the robot hologram through the calibration errors which are displayed on a panel, as shown in Fig. 4.4b. If the "*No*" option is chosen, then the procedure must be repeated.

Position errors are calculated by assessing the difference between the real and the AR reconstructed vectors connecting the origins of $\boldsymbol{\Sigma}_q$ and $\boldsymbol{\Sigma}_v$ (and the corresponding



(a) Robot hologram placement . (b) Validation of calibration procedure.

Figure 4.4 Calibration procedure.

estimates, $\hat{\boldsymbol{\Sigma}}_q$ and $\hat{\boldsymbol{\Sigma}}_v$), i.e.

$$\tilde{\boldsymbol{p}}^q = \hat{\boldsymbol{R}}_q^{\mathrm{T}}(\hat{\boldsymbol{p}}_q - \hat{\boldsymbol{p}}_v) - \boldsymbol{R}_r^q(\boldsymbol{p}_q^r - \boldsymbol{p}_v^r), \tag{4.2}$$

where $\tilde{\boldsymbol{p}}^q$ is expressed with respect to $\boldsymbol{\Sigma}_q$, and denotes the position calibration error vector.

On the other side, the orientation error, $\widetilde{\boldsymbol{R}}$, is conceived as the discrepancy between the (3×3) identity matrix $\boldsymbol{I}_3$ and the rotation matrix, obtained by

$$\widetilde{\boldsymbol{R}} = (\hat{\boldsymbol{R}}_q^{\mathrm{T}} \hat{\boldsymbol{R}}_v)^{\mathrm{T}} \boldsymbol{R}_v^q, \tag{4.3}$$

where $\boldsymbol{R}_v^q$ is the rotation matrix expressing the relative orientation between $\boldsymbol{\Sigma}_v$ and $\boldsymbol{\Sigma}_q$, measured in the real world.

In the validation panel shown in Fig. 4.4b, the x, y and z axes position errors are displayed in meters, and the rotation errors about the same axes, in degrees. Notice that the calibration procedure must be executed each time the application is started.

A test campaign has been carried out to validate the algorithm. In particular, out of a total of 25 calibration executions, the norm of average errors is 5.289 mm in position and 0.991 deg in orientation, with standard deviation 2.766 mm and 0.698 deg

for position and orientation, respectively. These data can be fitted by Gaussian distributions, whose means and standard deviations are reported in Table 4.2.

The proposed calibration procedure provides clear improvements with respect to the one mentioned in [142], whose errors are about 9 mm and 3.1 deg, achieved in an average running time of 23.1 s, much greater than the 5 s average time of the envisaged procedure.

Table 4.2 Average and Standard deviation error out of 25 calibration executions.

	Average	Standard deviation
x [mm]	−1.094	0.475
y [mm]	−4.279	0.949
z [mm]	2.910	2.555
x [deg]	0.434	0.468
y [deg]	−0.381	0.403
z [deg]	−0.805	0.326

4.5.2 Robot State

Once the robot hologram has been properly positioned in the AR environment, the virtual joint positions must be updated in real time to track the motion of the corresponding real twin. To this aim, a script fetching data from ROS by subscribing to the topic `/joint_states` has been developed. This topic provides the exchange of JointState messages, containing information about the position, $\boldsymbol{q}$, velocity, $\dot{\boldsymbol{q}}$, and torque, $\boldsymbol{\tau}$, of the joints. Based on the acquired data, the script updates the robot hologram for each frame through Unity's *Update()* trigger.

Consider the teaching-by-showing technique for robot programming: while the operator guides the robot manually along the desired motion path, the data read by joint position transducers are stored and can be played back [101]. As this action could be performed also by an operator without technical knowledge, the teaching-by-showing technique is widespread in industrial field. During the training for the task, it should be useful to inform the operator about the status of the robot joints in terms of the distance for their physical limits ($\boldsymbol{q}_{min}$ and $\boldsymbol{q}_{max}$). In detail, a sphere per virtual joint is displayed in AR environment and aligned to the relative robot joint. As can be seen in Fig. 4.5, the i-th sphere (with $i = 1, \ldots, 7$) is characterized by variable colors to inform the operator about the joint status:

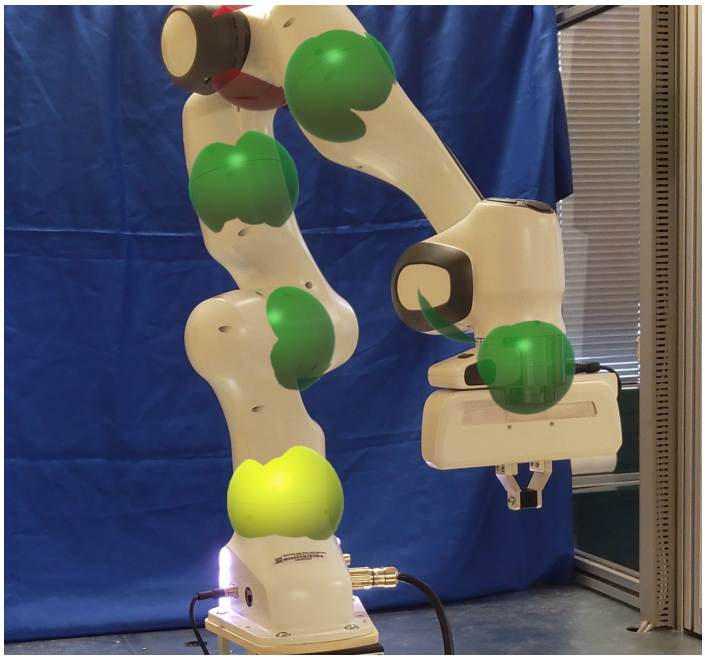


Figure 4.5 Joint spheres for the evaluation of joint limit distance.

- green, if the i-th joint position is away from the limit, i.e.,

$$q_{min_i} + 10 \text{ deg} < q_i < q_{max_i} - 10 \text{ deg}; \tag{4.4}$$

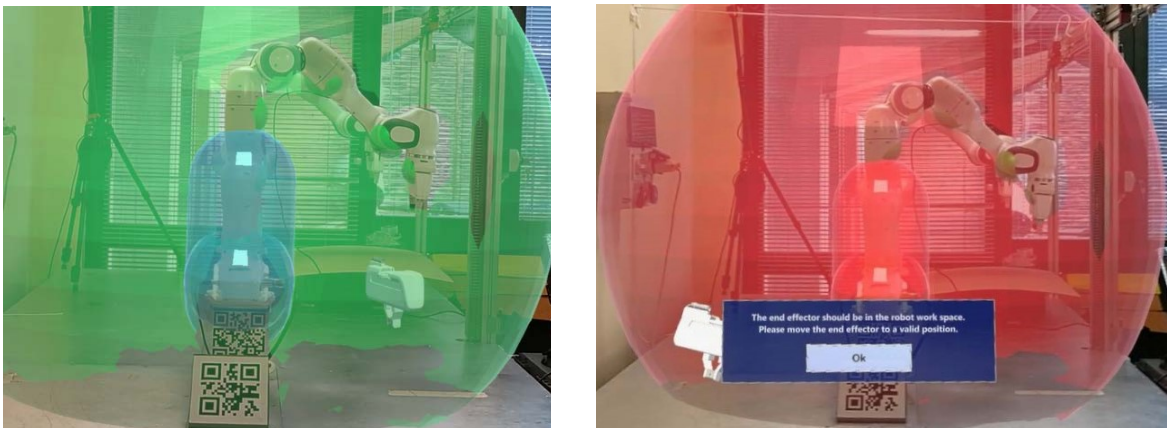
- yellow, if the i-th joint position approaches the limit, i.e.,

$$q_{min_i} + 5 \text{ deg} < q_i \leq q_{min_i} + 10 \text{ deg} \quad \text{OR} \quad q_{max_i} - 10 \text{ deg} \leq q_i < q_{max_i} - 5 \text{ deg}; \tag{4.5}$$

- red, if the i-th joint position is very close or equal to the limit

$$q_{min_i} \leq q_i \leq q_{min_i} + 5 \text{ deg} \quad \text{OR} \quad q_{max_i} - 5 \text{ deg} \leq q_i \leq q_{max_i}. \tag{4.6}$$

In other words, the color switches from green to yellow if the joint position is less than 10 deg far from the physical limit, whilst from yellow to red when it is less than 5 deg from it. The joints spheres can be activated or deactivated through the *Show/Hide Joint Spheres* button in the Human-Robot Interaction panel (Fig. 4.3b).



(a) Feasible target pose. (b) Unfeasible target pose.

Figure 4.6 Feasibility check of the end-effector target pose.

4.5.3 Robot Workspace Visualization

The adoption of AR allows to visualize the robot workspace. Starting from the information related to joints' limits in Franka Emika Panda datasheet, the reachable space has been reconstructed in a 3D model through a CAD software, obtaining an envelope surface which represents the border points reachable by the robot end-effector: in Fig. 4.6a, the blue volume has been subtracted from the reachable space (in green), for self-collision avoidance, according to the provided datasheet. Then, the 3D surface has been discretized adopting a triangle mesh: the maximum and minimum size of each element has been set as a trade-off between the accuracy and the computational burden. The user can decide whether to show the robot workspace in AR environment by selecting *Show/Hide Workspace* in the Human-Robot Interaction panel (Fig. 4.3b).

The visualization of the workspace offers a visual help to the user to locate the goal position of the end-effector. More in detail, the user can grab the end-effector hologram and move it in any position of the AR environment to assign a new trajectory to the robot (Fig. 4.6a).

Regardless the robot workspace is displayed or not, a check over trajectory feasibility is always executed. Indeed, if the operator releases the end-effector hologram outside the reachable space (Fig. 4.6b), the entire workspace is highlighted in red, warning her/him about the uncorrected position. then, if the user accepts an unfeasible end-effector pose, even in the presence of the red workspace, a panel appears advising her/him to modify the target pose.

4.5.4 Trajectory generation

Once the end-effector target pose is correctly defined within the robot workspace, the path and the corresponding time law have to be assigned. First of all, the trajectory path is computed by a proper ROS node and then sent to Unity in order to be visualized in the AR environment. When the operator presses the "*New Trajectory*" button on the Trajectory planning panel (Fig. 4.3c, HoloLens sends a message on the topic `/event_start` and a linear point-to-point motion in the operational space is planned

$$\boldsymbol{x}_d(s(t)) = \boldsymbol{x}_0 + \frac{s(t)}{\|\boldsymbol{x}_f - \boldsymbol{x}_0\|}(\boldsymbol{x}_f - \boldsymbol{x}_0). \tag{4.7}$$

The (6×1) vector $\boldsymbol{x}_d$ represents the desired value of the end-effector pose $\boldsymbol{x}$ (i.e., position and orientation):

$$\boldsymbol{x} = [p_x, \quad p_y, \quad p_z, \quad \phi, \quad \theta, \quad \psi]^{\mathrm{T}}, \tag{4.8}$$

where the position components are expressed in the robot base coordinate frame and the orientation is expressed via a triple of Euler angles (e.g., roll-pitch-yaw). Moreover, in (4.7), $\boldsymbol{x}_0$ and $\boldsymbol{x}_f$ are, respectively, the initial and final end-effector pose, and $s(t)$ is the abscissa which represents the time law of the path, planned with a fifth-order polynomial function, as

$$s(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5. \tag{4.9}$$

Finally, Fig. 4.7 shows the displayed trajectory and the end-effector hologram at the final point.

4.5.5 Trajectory validation

After the generation of the linear point-to-point trajectory, the user can suitably modify the end-effector path so as, e.g., to avoid collisions with obstacles (see Fig. 4.8a). To this aim, among the way-points visualized in the AR environment, the orange ones can be dragged by the user in order to design a new path by keeping constant the initial and final poses. By pressing the "*Update Trajectory*" button, the new positions of the orange way-points are sent to the ROS node, which computes a cubic interpolating spline, designing the new path, as shown in Fig. 4.8b. Vice versa, the user can restore

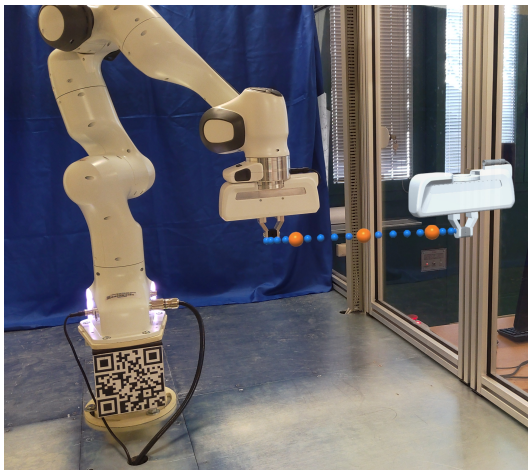
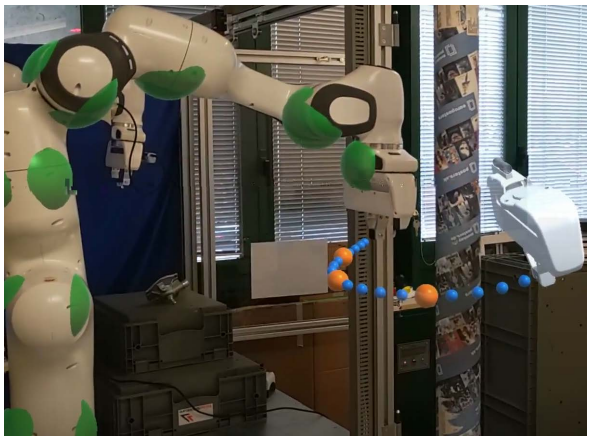


Figure 4.7 Linear point-to-point trajectory generation in AR.

(a) Possible obstacle collision.



(b) Modified trajectory around the obstacle.

Figure 4.8 Trajectory modification in presence of an obstacle.

the initial point-to-point linear path, by pressing the "*Clear waypoints*" button: then, the assigned way-points positions are discarded and all changes are lost.

The feasibility check over the way-points position is also enabled: same arguments dealing with the target end-effector pose (Section 4.5.3) hold for them.

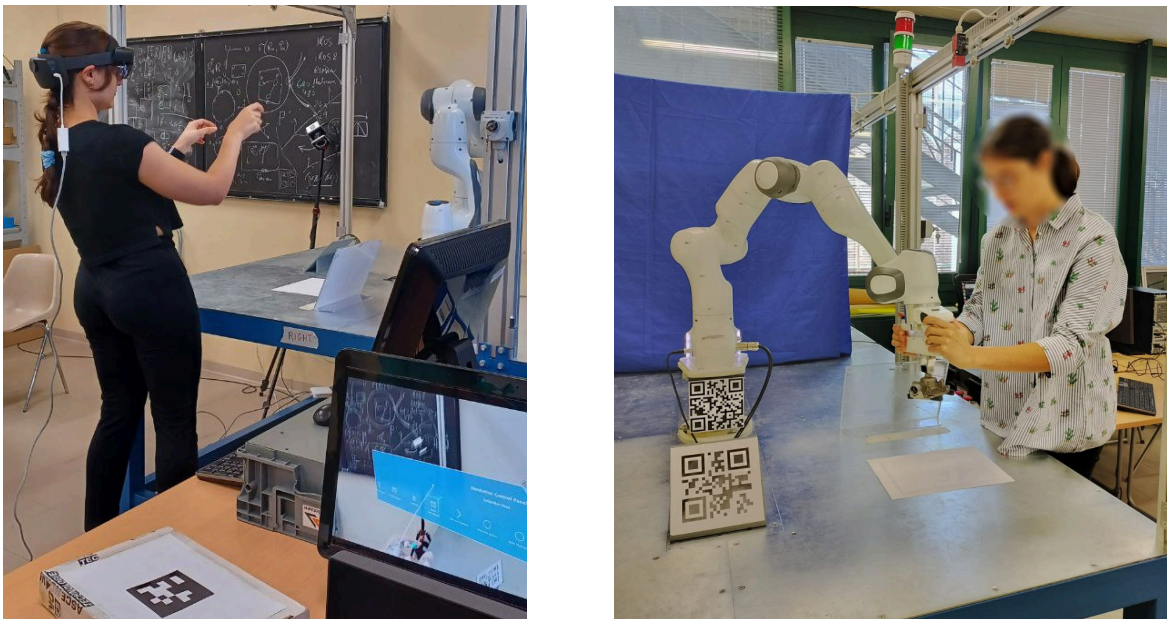
4.5.6 Trajectory simulation and execution

Once the user checks over the trajectory feasibility, a closed loop inverse kinematics (CLIK) algorithm [101] is implemented in order to compute the robot motion in terms of joint velocity:

$$\dot{\boldsymbol{q}}(t) = \boldsymbol{J}^{\dagger}(\boldsymbol{q}) \left[\dot{\boldsymbol{x}}_d(t) + \boldsymbol{K} \left(\boldsymbol{x}_d(t) - \boldsymbol{x}(t) \right) \right], \tag{4.10}$$

where $\boldsymbol{K}$ is a (6×6) matrix of positive gains. Prior to commanding the manipulator to track the desired trajectory, the operator can exploit the AR features to visualize the robot's simulated motion. The "*Simulate trajectory*" function assists the user in evaluating the planned path through the simulated behavior of the robot. Finally, when the operator presses the "*Move robot*" button, the robot executes the desired trajectory in the real environment.

4.6 Experiments

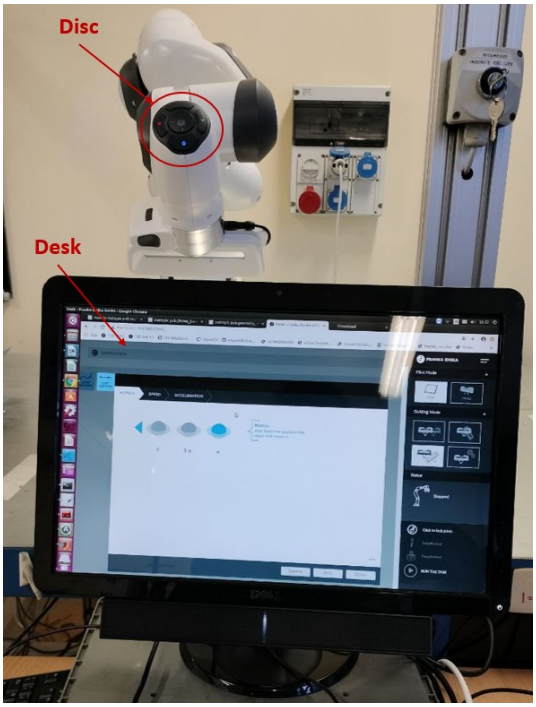


(a) Setup of the AR interface.

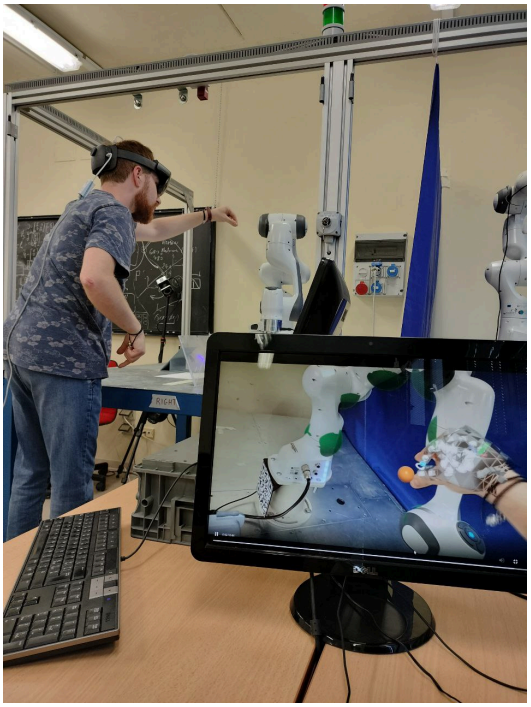
(b) Setup of the kinesthetic interface.

Figure 4.9 Setup of both the AR and the kinesthetic interface.

As reported in [148], a usability study of the AR platform involving 45 participants with different ages and expertise in robot programming and Extended Reality (XR) platforms has been conducted. Two programming methods have been proposed to the user: a kinesthetic teaching interface, provided by the Franka Emika Panda cobot, and the presented AR platform. More in detail, in the first case, users were asked to interact with the cobot in the handling guide mode by physically moving and positioning it, whilst in latter case participants wore HoloLens 2 and interacted with the AR environment to command the robot motion (see Fig. 4.9).



(a) The kinesthetic interface: the Desk is the web-Based programming interface, the Disc is used to interact with both the cobot and the Desk.



(b) The AR interface: user's point of view within HoloLens is displayed in the monitor.

Figure 4.10 The AR and the kinesthetic interfaces.

Prior to carrying out the experiments, each participant has signed a consent form and she/he has undergone a preliminary training phase on how to safely operate the two interfaces. In the kinesthetic teaching scenario, each participant has been instructed on how to physically move the robot in hand-guiding mode using the guiding buttons on the last joint of the cobot, save way-points using the cobot Disc and check their storage on the Desk (see Fig. 4.10a).

Conversely, in the AR-robotic interface, each participant has been instructed on how to interact with the Simulation Control panel, visualize the cobot workspace, assign and modify a trajectory, and, finally, move the cobot, as displayed in Fig. 4.10b.

4.6.1 Task

The task can be divided in two steps:

1. first, each participant was asked to place a blank sheet — containing the printed placement target area — on the working table, positioned on the opposite side of the pick location with respect to a vertical obstacle.

2. second, starting from the pick pose with the object (a component used in the automotive industry) already grasped, the robot had to be moved along a generated trajectory so as to release the object onto the target area, while overcoming or bypassing the obstacle.

The presence of the obstacle is intended to force the user to generate a curved trajectory to avoid collision with it. The two tasks have been executed in a fixed sequence: initially, adopting the kinesthetic interface and, then, with the AR platform, since the latter would have provided participants the information about the robot workspace, invalidating the first step of the task.

After completing the task with both the kinesthetic and the AR methods, participants filled out the "raw" NASA Task Load Index [149] and the System Usability Scale [145] surveys: regardless the accomplishment of the task, users were asked to assess the proposed methods used for locating the target area and generating the desired trajectory.

4.6.2 Surveys

The NASA Task Load Index (NASA TLX) is a tool for determining a subjective mental workload (MWL) assessment of a participant after completing a task. The overall workload index is measured after rating performance across six dimensions: mental demand, physical demand, temporal demand, effort, performance and frustration. Each subscale is accompanied by a corresponding question for clarification purposes. Participants are asked to self-rate their score on an interval scale ranging from 0 to 100, i.e. from very low to very high demand, respectively. The TLX also should employ a paired comparisons procedure: users should determinate the most effect on the workload during the task under analysis by 15 pairwise combinations between each subscale. Nevertheless, for this experimental campaign, it was preferred not to include this pairwise comparisons in order to reduce the burden on the questionnaire and to utilize the raw data, which was deemed more sensitive [149].

The System Usability Scale (SUS) is a reliable tool that can be used for global assessments of systems usability [145]. It consists of a questionnaire including ten items, employing a five-point Likert-type scale [150] ranging from "Strongly Disagree" to

"Strongly Agree", which correspond to the values 1 to 5, respectively. The statements are arranged in such a manner that the odd ones are pointing towards a greater usability of the proposed system, whilst the even statements suggest the opposite. They encompass different aspects, including the effectiveness and efficiency of the system, as well as the satisfaction of the users. The SUS total rating is then multiplied by an appropriate factor to convert it to a range of 0 to 100.

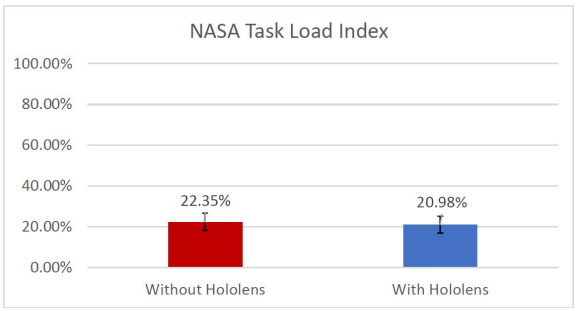
4.6.3 Analysis of the results

Paired samples t-tests has been conducted on participants' reported data obtained from the NASA TLX and the System Usability Scale. In this section, the results of these tests are reported in terms of $t(44)$ and p, which are the t-statistic value of a paired 44-samples t-test and the p-value, respectively, along with the effect size expressed in terms of Cohen's d [151]. A correction factor for samples with a cardinality of less than 50 has been taken into account in the calculation of the effect size.

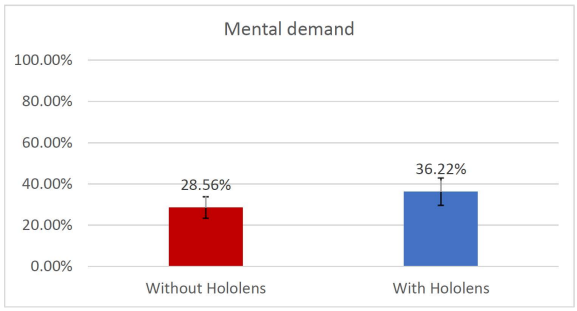
NASA TLX results

In Fig. 4.11 the main results of the NASA TLX survey are reported. Statistically significant differences have been found in the mental and physical demands between the two interfaces. The kinesthetic system appears to require less mental effort ($Mean = 28.56\%$, $Standard\ Deviation = 17.44\%$) than the use of the AR system ($M = 36.22\%$, $SD = 22.19\%$), $t(44) = 2.8350$, $p = 0.0069$, $d = 0.3690$, as depicted in Fig. 4.11b.

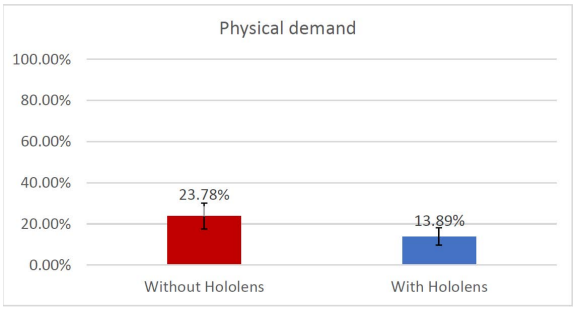
The mental workload required by the AR platform is strictly dependent on the interaction skills with holograms, as well as related to the ability to recall the specific sequence of buttons; on the other hand, the kinesthetic teaching interface provided by the cobot is extremely intuitive and user-friendly. It is not hard to imagine that a AR novice would tackle challenges when interacting with the virtual reality features. An appropriate evaluation should consider the different background of the participants in relation to AR devices. To this aim, Fig. 4.12 shows the comparison between upper-intermediate users' assessments on Mental workload (14 among 45 participants, depicted in grey) compared to those of participants with lower levels of AR expertise (31 among 45, in blue). The difference between the two samples in the case of the kinesthetic interface is almost negligible, while, in the case of the AR platform, a great discrepancy is experienced: the mental workload points out a sensible increment for the



(a) NASA TLX for kinesthetic and AR interfaces.



(b) Mental demand for kinesthetic and AR interfaces.



(c) Physical demand for kinesthetic and AR interfaces.

Figure 4.11 NASA TLX, mental demand and physical demand for kinesthetic and AR interfaces; error bars represents 95% confidence intervals (CIs).

non-expert users, whilst for the expert users it is comparable with the other interface. Thus, One could argue that the mental load requested by the AR platform can be easily reduced with a more intensive training phase on the HoloLens device, as it is rather due to the participants' inexperience than to the complexity of the proposed application.

On the other hand, as reported in Fig. 4.11c, the kinesthetic interface necessitates more physical demands ($M = 23.78\%$, $SD = 20.84\%$) than the AR interface ($M = 13.89\%$, $SD = 13.94\%$), $t(44) = 3.7025$, $p = 0.0006$, $d = 0.5358$, as evidenced by a larger effect size. This is an expected result since the kinesthetic teaching method requires users to physically move the robot to each way-point of the desired path, whereas the AR-robotic interface does not need physical interaction.

There are no other noteworthy variances in the other dimensions. Nonetheless, the adoption of the AR platform results in decreased temporal demand and effort while yielding better performance levels (85.89% without the HoloLens platform, 88.44% with the HoloLens). Although it does not represent a statistically significant difference, the

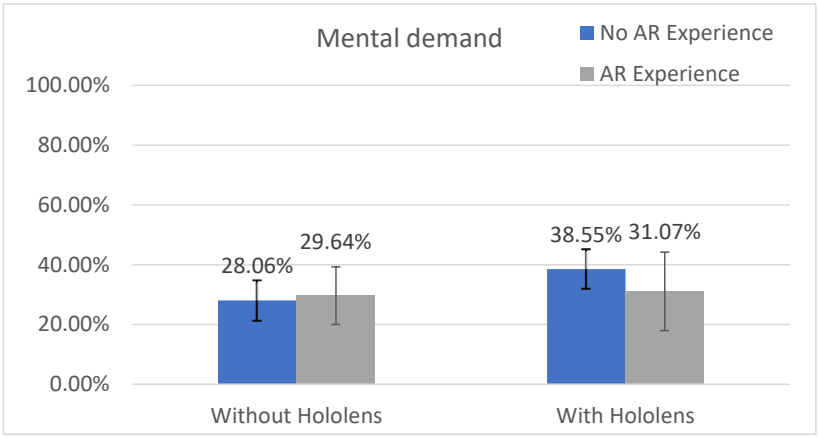


Figure 4.12 Mental demand for kinesthetic and AR interfaces, with comparison between different levels of AR expertise; error bars represents 95% CIs.

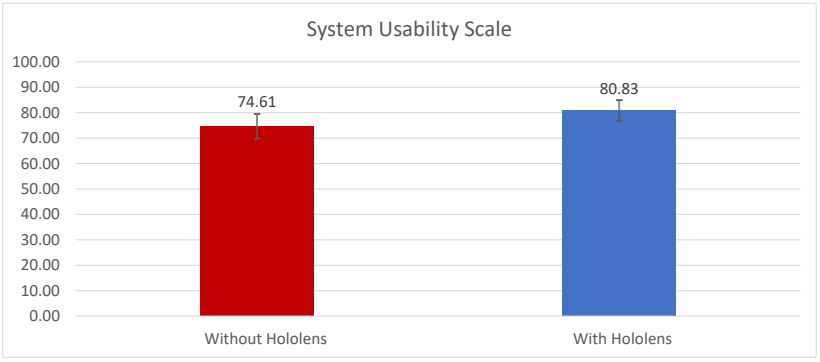


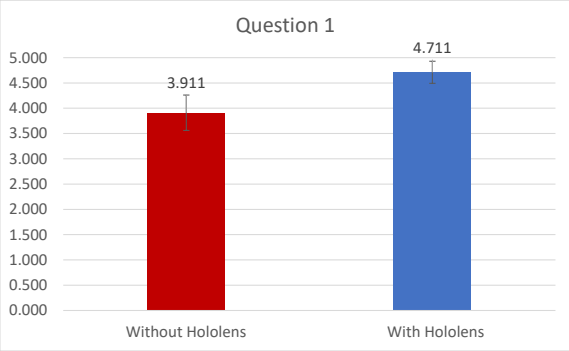
Figure 4.13 SUS scores for kinesthetic and AR interfaces; error bars represents 95% CIs.

overall NASA TLX, depicted in Fig. 4.11a is slightly decreased with the AR interface (−6.13%).

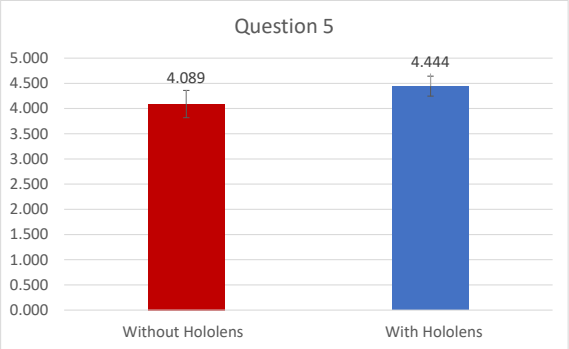
SUS results

The effectiveness of the proposed platform can be distinctly assessed through the analysis of the SUS, whose main results are reported in Figs. 4.13 and 4.14.

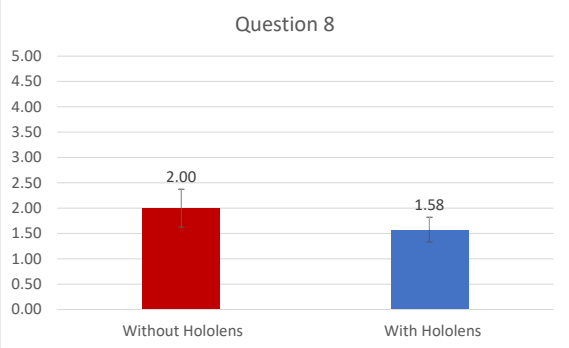
Statistically significant differences have been found in the SUS overall score between the two interfaces: the AR system ($M = 80.83\%$, $SD = 13.65\%$) surpasses the kinesthetic system ($M = 74.61\%$, $SD = 16.24\%$) by 6.22% in terms of usability, $t(44) = 2.09$, $p = 0.043$, $d = 0.3984$, as depicted in Fig. 4.13. The overall SUS score has corroborated that the Franka Emika Panda robot is designed to be a collaborative robot with a user-friendly kinesthetic teaching interface, which is much more intuitive than



(a) SUS first question scores for kinesthetic and AR interfaces.



(b) SUS fifth question scores for kinesthetic and AR interfaces.



(c) SUS eighth question scores for kinesthetic and AR interfaces.

Figure 4.14 Main statistically significant results of the SUS survey; error bars represents 95% CIs.

the industrial robots' teach pendant. Hence, it is noteworthy that the improvement in usability achieved by the proposed AR platform could be further increased in industrial scenarios.

In this regard, the results concerning the first SUS statement (*"I think that I would like to use this system frequently"*), shown in Fig. 4.14a, prove that users' preference for the HoloLens platform increases by about 16% compared to the classical programming interface ($M = 4.71$, $SD = 2.17$ for the AR system, $M = 3.91$, $SD = 1.98$ for kinesthetic interface, $t(44) = 3.769$, $p = 0.0005$), with a large effect size of $d = 0.7916$.

Furthermore, the proposed AR interface satisfies the requirements of efficiency and consistency. This is evidenced by the scores of statements 5 (*"I found the various functions in this system were well integrated"*) in Fig. 4.14b, and 8 (*"I found the system very cumbersome to use"*) in Fig. 4.14c. Indeed, statistically significant differences have been found for the statement 5 ($M = 4.44$, $SD = 0.659$ for the AR system,

Table 4.3 Comparative analysis of the proposed platform with other approaches in the literature.

	Proposed platform	[142]	[136]	[139]	[141]	[135]	[135]
Show joints' limit	Yes	Yes	Yes	No	No	No	No
Show WS	Yes	Yes	No	No	N.A.	Yes	No
Show Trajectory	Yes	Yes	No	Yes	Yes	No	Yes
Modify Trajectory	Yes	No	No	Yes	No	No	Yes
Traj. planning	Yes	Yes	No	Yes	Yes	Yes	Yes
Show forces	No	No	Yes	No	No	Yes	No
Obstacle avoid.	Yes	Yes	No	Yes	Yes	No	Yes
Show calibr. errors	Yes	Yes	No	No	No	No	No

$M = 4.089$, $SD = 0.90$ for kinesthetic interface, $t(44) = 2.701$, $p = 0.01$, $d = 0.4329$) and the statement 8 ($M = 2.00$, $SD = 1.24$ for the AR system, $M = 1.58$, $SD = 0.81$ for kinesthetic interface, $t(44) = 2.06$, $p = 0.045$, $d = 0.3863$).

Although it does not represent a statistically significant difference, the statement 9, concerning users' confidence during task execution, reveals that each participant felt safer by the HoloLens platform of about 4.4%. To this purpose, it has been experienced that visualizing the trajectory in the AR environment as a set of way-points significantly boosts users' confidence.

Moreover, according to the authors' best knowledge, an AR platform that implements exactly the same features is not available in the literature. To this aim, a detailed comparative analysis cannot be performed. However, we have conducted a qualitative comparison with similar methods (see Table 4.3) taking into account the most relevant functionalities.

4.7 Safety Developments

Human-Robot Collaboration (HRC) emerges as a pivotal aspect of Industry 5.0, aiming to integrate robots and artificial intelligence seamlessly into workplaces alongside human workers. This integration presents challenges and opportunities for redefining job roles and enhancing productivity, while ensuring safety and workers well-being. A key challenge lies in understanding how AR devices, such as the HoloLens 2, can

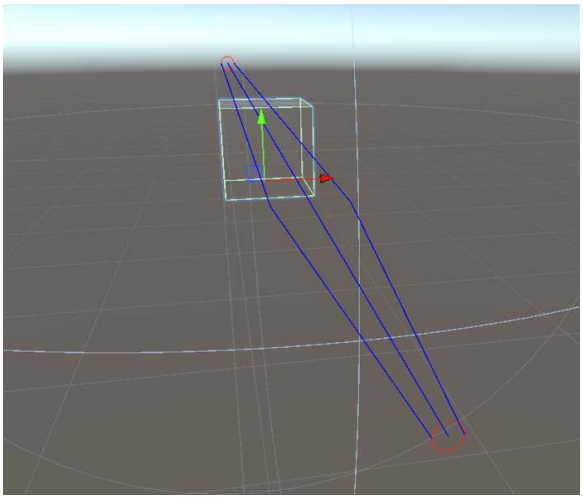
improve HRC by enhancing safety and user performance during programming and task execution, regardless of prior experience. In fact, the adoption of AR devices broadens the panorama of possible advanced safety features and allows the tracking of some new human parameters. For instance, the introduction of a virtual wall that separates the human's and the robot's workspaces could guarantee a safe execution of the operator's task. Moreover, considering the safety-rated monitored stop method, no human function is explicated except for her/his presence in the shared workspace. Additionally, no reference is provided about the operator's concentration and her/his monitoring of the robot task execution in ISO standards.

All these advanced AR features have been integrated into the HumanTIX AR platform, developed for the HoloLens 2, as described in [152]. Building upon the previously described functionalities, this enhancement enables improved safety management during human–robot collaboration. First, during the trajectory planning phase, the operator can instantly create virtual safety walls, providing greater control over the collaborative workspace. Second, the platform enhances the monitoring of the operator's attention level during robot task execution. This feature is particularly useful during trajectory programming, when the operator tests the robot task according to the safety-rated monitored stop method. By exploiting the HoloLens gaze-tracking capability, the HMD verifies whether the operator's gaze is directed towards the robot. Whenever the operator is not focused on the robot's motion, the robot trajectory is appropriately scaled down, an acoustic signal is emitted, and a visual warning is displayed to remind the operator that the robot is still in motion.

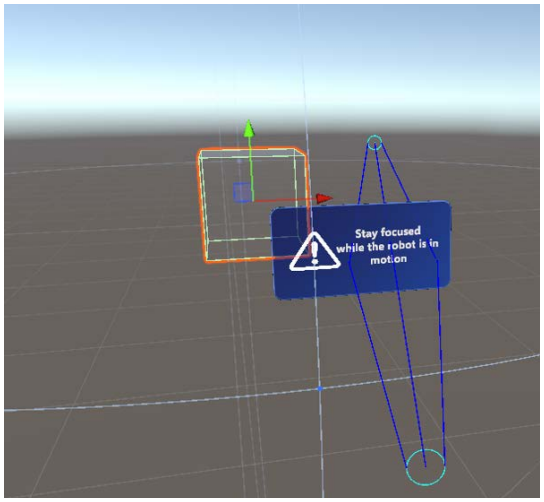
4.7.1 Eye and head tracking

AR environment

Eye tracking is a method used to monitor eye movements and determine the focus of the operator. HoloLens 2 utilizes two infrared cameras to track eye movement, as this type of light is non-invasive to the human eye during monitoring. By illuminating the eye with infrared light, visible reflections are generated in the cornea and pupil. This allows for proper illumination of the pupil and facilitates its distinction from the iris. In contrast, visible lights not only distract the subject being monitored but also do not provide suitable lighting for tracking. The proposed application utilizes gaze monitoring to assess the operator's attention towards the robot's movements. A transparent hologram cube, which follows the robot's motion, was used as a reference



(a) Unity simulation of the *OnEntered()* event. The blue lines represent the gaze ray. The red circles at the origin and end of the ray indicate that the ray intersects an object.



(b) Unity simulation of the *OnExited()* event. The blue lines represent the gaze ray. The green circles at the origin and end of the ray indicate that the ray does not intersect any object.

Figure 4.15 Unity simulations of the human gaze tracking in both cases of human focus on the object (left) or not (right).

to monitor gaze direction. The cube was equipped with the *StatefulInteractable* component, available in the MRTK3, to manage various interaction events. The event *isGazeHovered* specifically checks if the gaze ray intersects with the surface of the cube. Fig. 4.15a shows a simulation of the *OnEntered()* event, which is triggered when the operator looks at the cube. The image was captured from Unity, as the objects used for this feature are invisible during the application's execution. The figure displays both the cube and the gaze ray, represented by blue lines, with red circles at the origin and end of the ray. The red color of the circles indicates that the ray has encountered the cube hologram, indicating the operator's focus on the robot.

Fig. 4.15b shows a simulation of the *OnExited()* event, which occurs when the operator's attention shifts away from the robot. The blue circles at the origin and end of the ray indicate that the ray does not intersect with any surface. When the operator looks away from the end-effector, two warning signals are triggered: an acoustic and a visual signals. The visual signal is a panel that displays a warning not to get distracted, while the acoustic signal is an intermittent sound that imitates a safety alarm. As soon as the operator returns their attention to the robot, the acoustic signal stops, and the panel disappears.

Eye tracking is more delicate, since eye movements are faster and more abrupt, then head movements can also provide information about the operator's attention. In the proposed application, the head tracking feature of HoloLens 2 is used for the same purposes as eye tracking, using the same strategy as previously described. The operator's attention is continuously transmitted to the ROS node that man-ages the robot's movement, thereby ensuring that the robot is aware of the opera-tor's actions at all times.

Robot control

The path for the robot motion is generated through a cubic interpolating spline $\boldsymbol{\sigma}_d(s(t))$ with a time law represented by abscissa $s(t)$ planned with a third-order polynomial function, where $\boldsymbol{\sigma}_d = [\boldsymbol{\sigma}_{d,p}^{\mathrm{T}}, \boldsymbol{\sigma}_{d,o}^{\mathrm{T}}]^{\mathrm{T}}$ is the vector representing the actual end-effector position and orientation.

To ensure the safety of the operator, when her/his gaze is not directed towards the robot, it is commanded to reduce the robot's velocity up to the half of the planned one. In order to scale the desired trajectory, $\boldsymbol{\sigma}_d$, when the operator attention is not focused on the robot, a scaled time variable can be defined as

$$s(t+1) = s(t) + w(t)dt \tag{4.11}$$

where dt is the time step and $w(t)$ is a function defined as

$$w(t) = \begin{cases} 1 - \left[\left(1 - e^{-\frac{(t-t_{scal})}{\tau}}\right) p_{scal}\right] & \text{if she/he looks away from the robot,} \\ p_{scal} + \left[\left(1 - e^{-\frac{(t-t_{scal})}{\tau}}\right)(1 - p_{scal})\right] & \text{if she/he focuses on the robot} \end{cases} \tag{4.12}$$

where τ is a time constant, t_{scal} is the time instant when the operator looks away (or focuses) and $0 < p_{scal} \leq 1$ is used to scale the robot's velocity. It is worth noting that the initial value of $w(t)$ is set to 1, as it is assumed that the operator is focused on the robot when it starts the motion, since an alert panel is displayed to inform her/him that the robot is about to move.

Since $w(t) = s(t)$, the time derivative of $\boldsymbol{\sigma}_d(s(t))$ is

$$\dot{\boldsymbol{\sigma}}_d(s(t)) = \dot{s}(t)\dot{\boldsymbol{\sigma}}_d(s(t)) = w(t)\dot{\boldsymbol{\sigma}}_d(s(t)). \tag{4.13}$$

After computing the desired scaled trajectory and its first derivative, $\boldsymbol{\sigma}_d(s(t))$ and $\dot{\boldsymbol{\sigma}}_d(s(t))$, a closed loop inverse kinematics algorithm is implemented in order to compute the motion in terms of reference velocity of each robot joint

$$\dot{\boldsymbol{q}}_r(t) = \boldsymbol{J}^{\dagger}(\boldsymbol{q})(\dot{\boldsymbol{\sigma}}_d + \boldsymbol{K}(\boldsymbol{\sigma}_d - \boldsymbol{\sigma})), \tag{4.14}$$

where $\boldsymbol{K} = diag(\boldsymbol{K}_p, \boldsymbol{K}_o)$ is a (6×6) matrix of positive gains.

Given the speed at which eyes can move, it is necessary to avoid oscillations in the variable w that would result in the robot slowing down and speeding up continuously. In order to achieve this, a duration of 0.1 seconds is set to be sure of the operator's change of target before the robot is notified.

4.7.2 Virtual wall

AR environment

In the context of collaborative tasks where both the robot and the operator must perform individual operations while sharing the workspace, the operator may require greater control over the robot's movement. To this end, the proposed application provides the opportunity to create a virtual wall that limits the workspace. The virtual wall is represented by boxes in the Unity environment, which are made with translucent material so as not to obstruct the operator's view of the robot, as shown in 4.16. It is assumed that the height and width of the wall are infinite, although for the sake of visualization, the virtual object is displayed as finite. The initial position of the virtual wall is defined in a fixed location relative to the robot's base frame and arrows for manipulation are placed on it. Two types of arrows are used in order to allow the operator to manipulate the virtual wall. The arrows were equipped with the *ObjectManipulator* component, available in the MRTK3 and the red one indicates that the wall may be translated along the axis perpendicular to the wall, whereas the blue ones indicate that the wall may be rotated around the vertical parallel axis. Additionally, a label is displayed on the virtual wall, indicating the distance of the wall from the robot's base frame along the perpendicular to the wall and the angle of rotation around the vertical axis in real-time.

In order to access this feature, the operator must utilize a specific button on the menu panel that is dedicated to trajectory planning. During the interaction with the arrows, the menu panel is deactivated via the *ManipulationStarted* object event in order

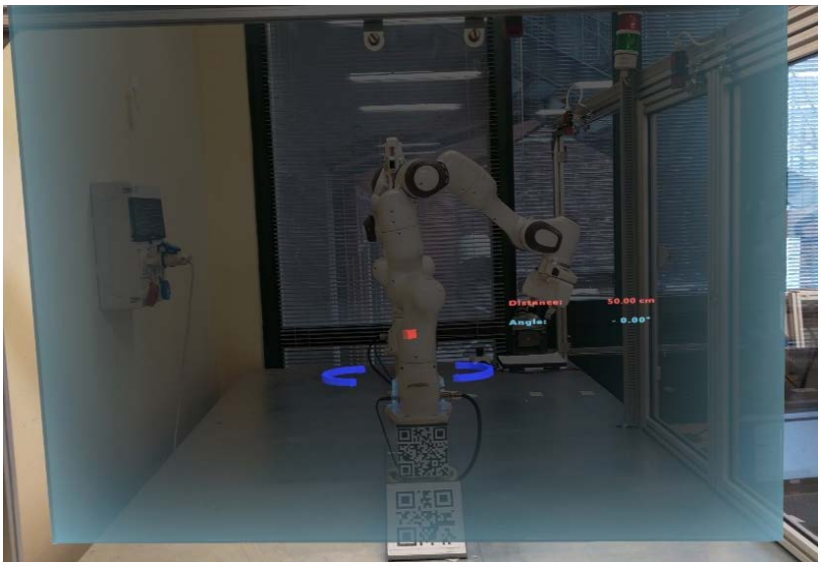


Figure 4.16 The virtual wall in the AR environment.

to permit the manipulation of the arrows without any distractions. Upon the release of the arrows, the *ManipulationEnded* object event makes the menu panel visible once more, and the information regarding the wall's position and orientation is transmitted to the ROS node responsible for managing the robot's movement.

Robot control

The message sent by the HDM consists of three non-aligned points, $\boldsymbol{p}_i$ with $i = 1, 2, 3$, from which it is possible to extract the information about the wall's position and orientation, expressed in the robot's base frame. The function used to implement the virtual wall task in the robot control algorithm is the following

$$\sigma_{wall} = \hat{\boldsymbol{n}}^{\mathrm{T}}(\boldsymbol{p}_{ee} - \boldsymbol{p}_1) \tag{4.15}$$

where $\boldsymbol{p}_{ee}$ is the actual position of the end-effector with respect to the robot's base frame, and $\hat{\boldsymbol{n}}$ is the normal unit vector of the plane representing the wall computed as [153]:

$$\hat{\boldsymbol{n}} = \frac{(\boldsymbol{p}_2 - \boldsymbol{p}_1) \times (\boldsymbol{p}_3 - \boldsymbol{p}_1)}{\|(\boldsymbol{p}_2 - \boldsymbol{p}_1) \times (\boldsymbol{p}_3 - \boldsymbol{p}_1)\|}. \tag{4.16}$$

The task Jacobian can be defined as [153]

$$\boldsymbol{J}_{wall}(\boldsymbol{q}) = \hat{\boldsymbol{n}}\boldsymbol{J}_p(\boldsymbol{q}), \tag{4.17}$$

Table 4.4 Control parameters for Safety features.

Parameter	Value
p_{scal}	0.5
$\boldsymbol{K}_p$	$diag(100, 100, 100)$
$\bar{d}_w$	0.05
τ	1.5
$\boldsymbol{K}_o$	$diag(80, 80, 80)$
k_w	1

where $\boldsymbol{J}_p(\boldsymbol{q})$ is the positional part of the Jacobian matrix of the robot, $\boldsymbol{J}(\boldsymbol{q})$, expressed in its base frame. Let $d_w = |\sigma_{wall}|$ the distance between the end-effector and the wall, the closed loop inverse kinematics algorithm can be implemented as follows:

if $d_w > \bar{d}_w$

$$\dot{\boldsymbol{q}}_r = \boldsymbol{J}^{\dagger}(\boldsymbol{q})(\dot{\boldsymbol{\sigma}}_d + \boldsymbol{K}(\boldsymbol{\sigma}_d - \boldsymbol{\sigma})) \tag{4.18}$$

if $d_w < \bar{d}_w$

$$\begin{aligned}\dot{\boldsymbol{q}}_r = -k_w \boldsymbol{J}_{wall}^{\dagger}\boldsymbol{\sigma}_{wall} + \boldsymbol{J}_o^{\dagger}(\dot{\boldsymbol{\sigma}}_{d,o} + \boldsymbol{K}_o(\boldsymbol{\sigma}_{d,o} - \boldsymbol{\sigma}_o)) + \\ + \boldsymbol{J}_p^{\dagger}\Big[(\dot{\boldsymbol{\sigma}}_{d,p} + \boldsymbol{K}_p(\boldsymbol{\sigma}_{d,p} - \boldsymbol{\sigma}_p)) - \hat{\boldsymbol{n}}^{\mathrm{T}}(\dot{\boldsymbol{\sigma}}_{d,p} + \boldsymbol{K}_p(\boldsymbol{\sigma}_{d,p} - \boldsymbol{\sigma}_p))\boldsymbol{n}\Big]\end{aligned} \tag{4.19}$$

where the explicit dependence from $\boldsymbol{q}$ has been omitted for sake of compactness and d_w is a safety threshold, k_w is a positive gain, $\boldsymbol{J}_o$ is the orientational part of the Jacobian matrix of the robot, $\boldsymbol{J}$, expressed in its base frame.

4.8 Experiments

Two experiments have been conducted in order to test the effectiveness of the approach. The experimental setup consists of a collaborative Franka Emika Panda robot and HoloLens 2 as HMD. In Table 4.4, the control parameters used in the experiments are reported.

4.8.1 Eye and head tracking

In this experiment, the operator commands the robot to execute a 3D linear trajectory in 20 s and, then, looks away from the robot during the motion. Fig. 4.17 shows

the desired (dashed lines) and the actual (solid lines) position trajectories of the end-effector. The vertical black dash-dotted lines represent the initial time instant when the operator's gaze is not directed towards the robot (5.98 s), and when the focus returns to the end-effector (17.98 s), respectively. In this time interval, the robot's velocity slows down and the motion time proportionally increases to 25.95 s. Fig. 4.18 shows the trajectory does not change in the tridimensional space. Indeed, the desired path exactly matches the desired one.

4.8.2 Virtual wall

In this experiment, the operator defines a virtual wall in order to split the collaborative area. The wall is parallel to the yz plane of the robot's base frame and is placed at a distance of 0.5 m along the x-axis. Unlike the previous experiment, the desired trajectory for the robot end-effector consists of a curved trajectory, depicted as a blue line in Fig. 4.19, that crosses the wall, in gray. On the other side, the actual path followed by the end-effector is also shown in red. As can be seen, this path does not pass through the virtual wall, allowing the operator to have a separation between her/his own workspace and the robot one.

Fig. 4.20 shows the three positional components of the desired and the actual trajectories separately. Since the wall is parallel to the yz plane, only the trajectory along the x-axis is affected by the presence of the virtual boundary. Indeed, as expected, σ_x does not overcome the value -0.5, while σ_y and σ_z remain unchanged with respect to the relative components of σ_d. In Fig. 4.21, the robot velocity along the x-axis is depicted, since for the other two components, the actual velocities coincide with the desired ones. As expected, the actual velocity is very small when the end-effector is in proximity to the virtual wall, while it matches the desired one, when the planned trajectory is at a considerable distance from the boundary.

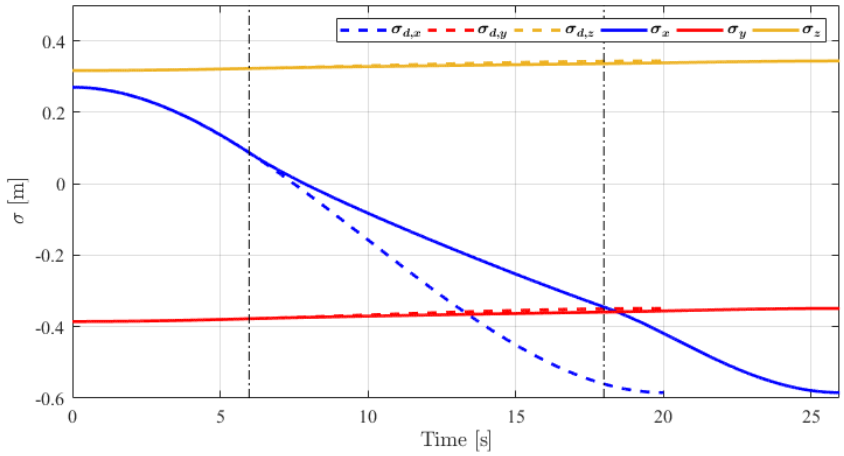


Figure 4.17 Desired (dashed lines) and the actual (solid lines) position trajectories. The vertical black lines denote when the operator looks away from the robot.

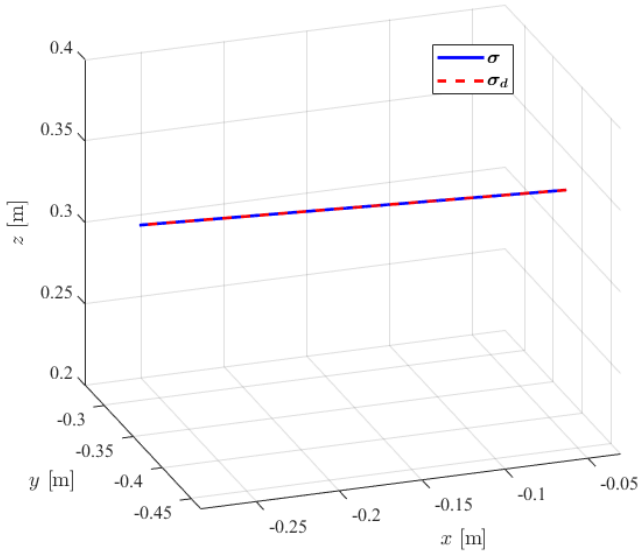


Figure 4.18 Desired (red dashed line) and actual (blue line) end-effector path.

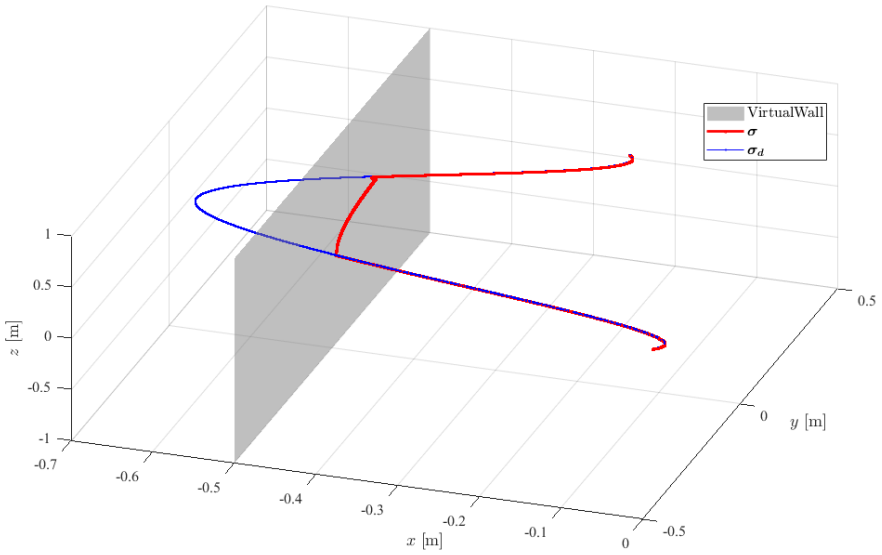


Figure 4.19 Desired (red dashed line) and actual (blue line) end-effector path.

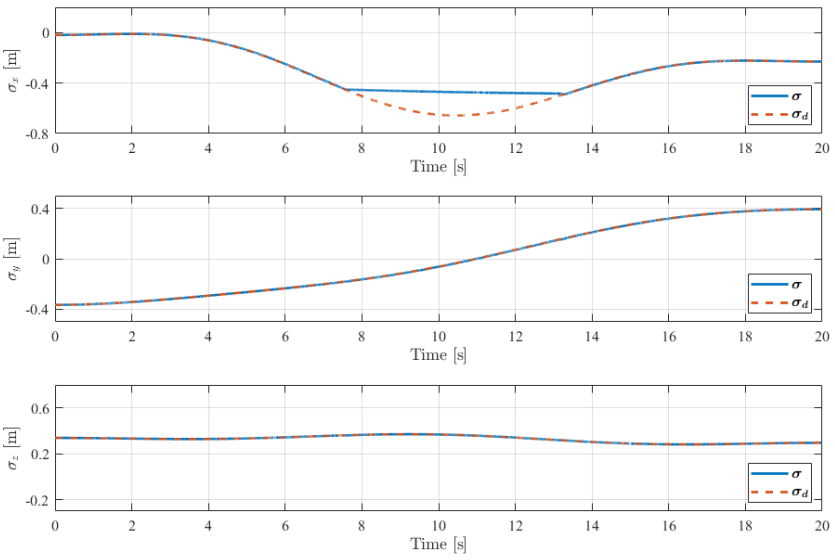


Figure 4.20 Desired (red dashed line) and actual (red line) end-effector for the three components x, y, and z.

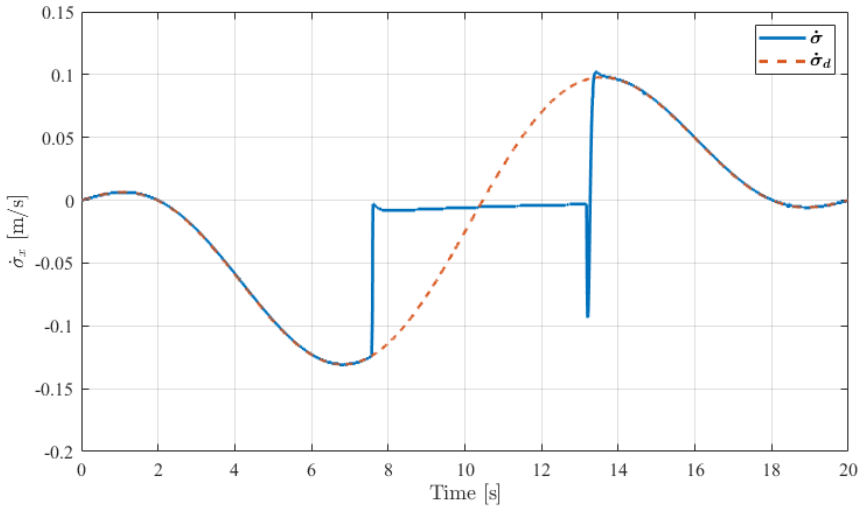


Figure 4.21 Desired (red dashed line) and actual (red line) x components of the end-effector velocity.

4.9 Human-Robot collaborative object transportation task

The control framework introduced in Section 2.10 can be extended to scenarios involving a human operator, in order to perform Human–Robot collaborative object transportation tasks. By leveraging the bi-directional communication established through the AR platform *HumanTIX*, the human operator can be seamlessly integrated as an *agent-in-the-loop* within the control architecture. On the one hand, the human operator can exploit the information provided by the Augmented Reality platform, visualized through the HoloLens 2 display, to monitor the robot's current state in real time, which shows also a feedback of the interaction forces exerted on the object during the task execution. On the other hand, the operator can plan the object trajectory and suitably modify it to minimize the forces applied on the object or for collision avoidance. Furthermore, the operator is able to proactively update the task sequence — such as grasping, pick-and-place, or motion actions — according to the task requirements. Finally, the proposed framework allows the user to switch between different control modes, ranging from a leader–follower scheme to an object admittance control. In this way, the operator can exploit her/his superior cognitive capabilities to assist the robot in complex decision-making or adaptation processes, while the manipulator assist her/him in physical demanding task, whose execution is beyond a single agent capability. A new fingertip has been 3D printed for ensuring torque transmission to the robot and avoiding object side-slipping throughout the transportation task. Figure 4.22 shows a screenshot of the collaborative object transportation task from the operator point of view, while wearing the HoloLens device (left), and an external point of view (right). The operator is provided the wrenches exerted on the object and can activate/deactivate the visualization of each component.[1]

More in detail, two interaction modalities are explored for Human-Robot collaboration:

1. In the first control mode, the human acts as the leader while the robot operates as the follower. In this configuration, the robot assists the operator in physically demanding tasks—such as object transportation—by exhibiting a compliant

[1]A video of the experiment is at
https://web.unibas.it/automatica/multimedia.html

behavior. Specifically, the operator commands the desired object trajectory by manually guiding it within the shared workspace: she/he can tackle unexpected situations and provide fine adjustments of the trajectory of the grasped object at each time instant. On the contrary, the robot provides support in lifting and moving the object through a compliant control. In this scenario, the Augmented Reality interface helps the human operator for grasping the object and provides visual feedback of the forces exerted by the robot.

According to this modality, the robot behaves as a passive agent, merely reacting to the operator's actions. It is compliant to the grasped object trajectory, according to

$$\boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{D}\dot{\boldsymbol{x}} = \boldsymbol{\gamma}, \tag{4.20}$$

where $\boldsymbol{x} = [\boldsymbol{p}^{\mathrm{T}}, \boldsymbol{\phi}^{\mathrm{T}}]^{\mathrm{T}}$ is the robot end-effector pose, with $\boldsymbol{\phi}$ denoting the Euler angles triplet, $\boldsymbol{M}$ and $\boldsymbol{D}$ are the gain matrices for the robot virtual mass and damping, respectively, and $\boldsymbol{\gamma}$ is the sensed wrench at the robot end-effector (as in (2.55)). Notice that no spring term is included in (4.20), since the desired end-effector pose is the actual pose of the manipulator in every time instant.

2. In the second control mode, the human operator still defines the object trajectory, but this information is shared with the robot through the Augmented Reality interface. As in the previous case, the AR platform assists the operator during the initial grasping phase and provides real-time visualization of the robot state and applied wrenches.

   In addition, the operator can assign the desired object trajectory and transmit it to the robotic agent. After performing the manipulator calibration, the AR device identifies the initial pose of the object through a QR code placed on it. Consequently, the virtual object reference frame appears within the AR scene, thereby the user can translate and rotate it to assign the target pose of the object.

   Based on this information, a path is generated in the task space, which can be further modified by the operator, as described in Section 4.5.5. The robot, in turn, exploits the geometric relations presented in Section 2.5.3 to compute the corresponding desired end-effector trajectory $(\boldsymbol{x}_d, \dot{\boldsymbol{x}}_d, \ddot{\boldsymbol{x}}_d)$.

   Throughout the task execution, an admittance control scheme is employed to reduce the wrenches exerted on the object. The robot computes the human wrench contribution on the object, along with the forces arising from an eventual

Figure 4.22 A screenshot of the video showing the collaborative object transportation, available at https://web.unibas.it/automatica/multimedia.html.

contact with the external environment, both concurring to end-effector sensed wrench, $\boldsymbol{\gamma} = \boldsymbol{T}_A^{\mathrm{T}}(\boldsymbol{\phi})\boldsymbol{h}$, according to

$$\boldsymbol{M}\ddot{\boldsymbol{e}} + \boldsymbol{D}\dot{\boldsymbol{e}} + \boldsymbol{K}\boldsymbol{e} = \boldsymbol{\gamma}, \tag{4.21}$$

where $\boldsymbol{e} = \boldsymbol{x}_d - \boldsymbol{x}_r$, with $\boldsymbol{x}_r$ denoting the end-effector reference pose. In this way, the manipulator acts as an active agent, since it is in charge of limiting both external and internal forces acting on the object, by means of an admittance controller framework analogous to that proposed in Section 2.10.

Finally, the human operator can switch between the two proposed collaborative control methods through the AR interface, enabling intuitive and flexible task supervision and execution.

Figure 4.23 shows the end-effector trajectory and the corresponding wrench (moving average filtered) exerted on the object for a collaborative transportation task executed through the first interaction mode. In this experiment, the object motion is constrained in the xy plane. As expected, the robot complies the sensed forces and torques, following the object commanded trajectory.

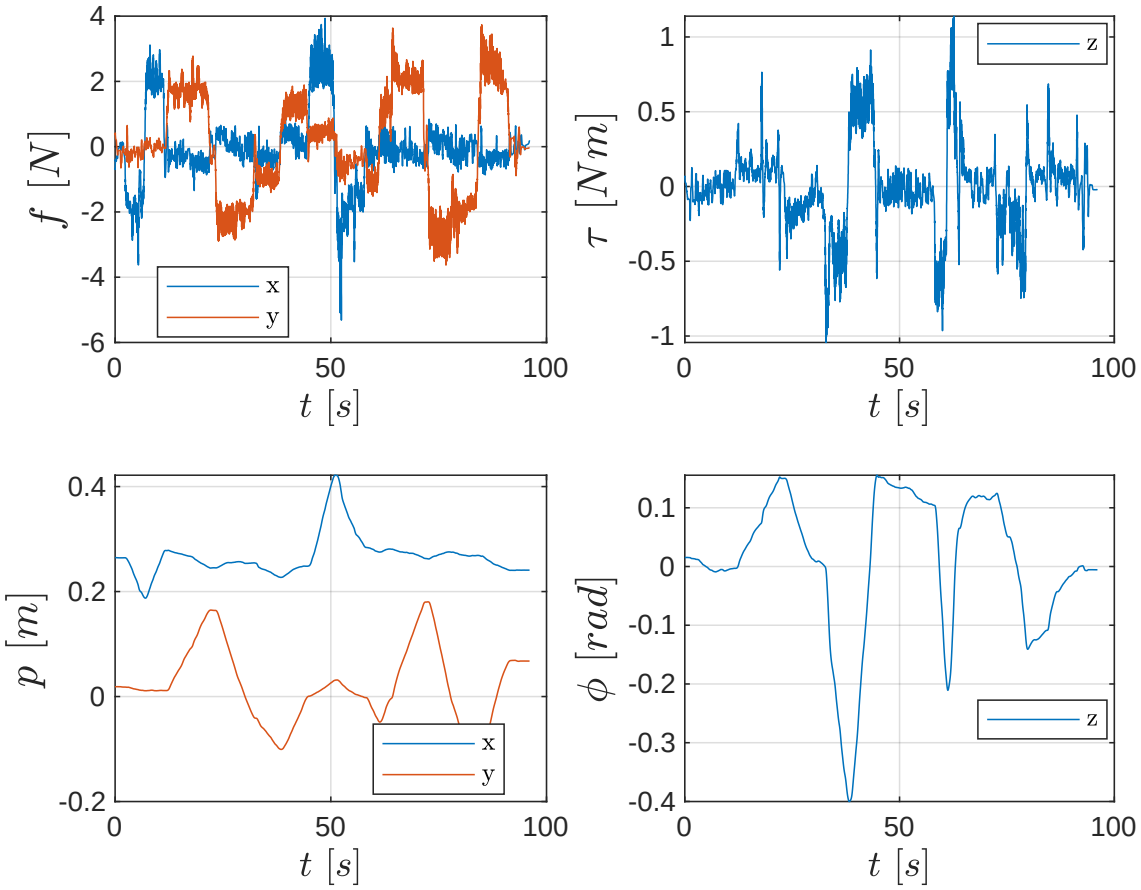


Figure 4.23 Robot trajectory (bottom) and corresponding filtered sensed wrench (top).

Chapter 5

Conclusions and future work

This thesis is focused on the development of novel distributed control strategies for cooperative multi-robot systems and collaborative multi-agent systems involving both humans and robots. The research objectives were twofold. The primary focus was the distributed cooperative transportation of rigid objects executed by a team of robotic manipulators. Building on distributed estimator–controller architectures, control frameworks have been designed to enable multiple robots to coordinate their motions, manipulate a common object and regulate interaction forces using only local information and communication with their neighbors. The second research line concerned Human–Robot Collaboration, aiming to let heterogeneous multi-agent systems perform an object transportation task, in a distributed manner, by integrating the human operator into the control loop through an intuitive Augmented Reality interface.

The first contribution of this work deals with the motion coordination of multi-robot systems and consists in devising a novel formulation for coordinating the motion tasks in a distributed fashion. This aim is achieved by resorting to the task-oriented formulation, which describes the motion in terms of absolute and relative variables, expressed in terms of the poses of the robot end-effector reference frames. Consensus-based estimators allow each agent to compute an estimate of the desired absolute and relative trajectories (up to a given derivative order) based solely on local information exchange among neighboring agents described by a given communication graph. Each robot then locally computes its desired end-effector trajectory by exploiting the estimated absolute and relative motion variables. A key strength of this formulation is that it remains valid even when the robots rigidly grasp a common object, i.e., when the relative task remains constant over time.

The second contribution concerns the design of distributed control strategies to reduce the wrenches exerted on the object. Building on the internal–external wrench decomposition introduced in [57], the objective is to simultaneously regulate internal forces, preventing plastic deformation of the object, and appropriately distribute external forces arising from environmental interactions. A novel consensus-based wrench estimator was proposed, in which each agent estimates the collective wrench vector using only its own measurements and information received from its neighbors. On this basis, it computes an estimate of both internal and external wrenches and implements controllers to keep them bounded. Several control laws have been developed, including a compliance controller and an internal admittance controller for internal wrench regulation, as well as a combined internal–external admittance control framework to also mitigate external wrenches.

Finally, the stability and convergence properties of all distributed controllers are formally established. Rigorous Lyapunov-based analyses are presented, with detailed mathematical proofs, ensuring the boundedness of the collective tracking errors either of the robot—in the case of the internal admittance— or the object— for the combined external-internal framework.

Several tests have been conducted in both simulated and real environments to assess the effectiveness of the proposed control frameworks. The internal admittance controller significantly reduces the internal forces and torques acting on the object; however, in the absence of external admittance, collisions with obstacles still generate pronounced internal wrench peaks and high external forces. Conversely, when both admittance filters are activated, external forces and torques after contact are remarkably reduced, object slipping is fully eliminated and internal wrenches remain consistently low throughout the entire task.

A further objective of this thesis was the extension of the multi-robot framework to include a human operator, integrating the human in the control loop to enable collaborative Human–Robot object transportation. The main contribution consists in the development of new functionalities for HumanTIX, an Augmented Reality platform designed for Microsoft HoloLens 2 that enables a bi-directional information flow between humans and robots, while ensuring safety and usability even for non-expert users. This bi-directional communication operates both from Human to Robot, allowing the operator to intuitively generate and modify robot trajectories in 3D space through virtual panels and hand gestures, and from Robot to Human, with the AR interface providing enhanced perception of the robot state, its workspace and the

end-effector exchanged forces and torques in real time. The usability of HumanTIX has been assessed through a usability test, where the platform has been compared with a traditional kinesthetic interface: results show that the AR application substantially reduces the operator's physical workload, although it may introduce a slight increase in cognitive effort for novice users.

The platform also incorporates safety mechanisms, such as virtual walls that separate human and robot workspaces and head/gaze tracking to monitor user attention and ensure safe collaboration. Through the AR interface, the operator can assign and adapt object trajectories during execution—for instance, to reduce applied forces or avoid collisions—and dynamically adjust the sequence of subtasks, such as grasping, pick-and-place or motion actions. Moreover, the framework supports switching between different control modes, ranging from leader–follower schemes to object admittance control. This allows the operator to exploit their superior cognitive abilities for high-level decision-making and adaptation, while the robot provides assistance in physically demanding tasks that exceed the capabilities of a single agent.

5.1 Future work

Regarding the cooperative transportation of rigid objects with multi-robot systems, future developments will extend the experimental campaign to scenarios involving a multi-robot system comprising mobile manipulators. This will enable the execution of wider and more complex trajectories, bringing the proposed framework closer to real industrial applications. From a theoretical perspective, a first line of investigation will focus on removing two key assumptions: the knowledge of the object's mass and geometry and the prior knowledge of the number of robots composing the work cell. In this way, the control and estimation framework will be generalized, allowing it to adapt to more heterogeneous and dynamically varying teams.

Another direction arises from the difficulties encountered in synchronizing signals across multiple robots. A possible improvement will be the introduction of an event-triggered communication scheme, activated only when specific thresholds, such as tracking error bounds, are exceeded. This will significantly reduce the communication load while preserving the required performance level.

Concerning Human–Robot collaborative transportation, several extensions will be pursued. The system will be scaled to enable collaboration between one human operator and multiple robots, possibly mobile, allowing for more flexible and realistic

cooperative manipulation tasks. This evolution will require the development of reliable environment-interaction detection modules to distinguish unintended collisions with external objects from intentional physical interactions with the human partner.

Another line of future work will involve moving toward shared autonomy rather than purely shared control centered on the human. By incorporating human intention recognition, the robot's transition between active and passive behaviors will be managed more smoothly, enabling more natural collaboration. This shift will also support the implementation of adaptive admittance control strategies, with stiffness matrices varying according to context, user intention and task demands, ultimately leading to smoother and more intuitive cooperative transportation.

Finally, an extensive user study campaign will be conducted to thoroughly evaluate the usability, transparency and overall effectiveness of the proposed control architecture in real collaborative scenarios. The resulting insights will guide further refinements and support the deployment of the system in practical applications.

References

[1] F. Basile, F. Caccavale, P. Chiacchio, J. Coppola, and C. Curatella, "Task-oriented motion planning for multi-arm robotic systems," *Robotics and Computer-Integrated Manufacturing*, vol. 28, pp. 569–582, 2012.

[2] M. Uchiyama and T. Yamashita, "Adaptive load sharing for hybrid controlled two cooperative manipulators," in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pp. 986–991 vol.2, 1991.

[3] J. Alonso-Mora, S. Baker, and D. Rus, "Multi-robot formation control and object transport in dynamic environments via constrained optimization," *The International Journal of Robotics Research*, vol. 36, no. 9, pp. 1000–1021, 2017.

[4] F. Pierri, M. Nigro, G. Muscio, and F. Caccavale, "Cooperative manipulation of an unknown object via omnidirectional unmanned aerial vehicles," *Journal of Intelligent & Robotic Systems*, vol. 100, no. 3, pp. 1635–1649, 2020.

[5] G. Antonelli, F. Caccavale, C. Sansone, and L. Villani, "Fault diagnosis for AUVs using support vector machines," in *Proceedings 2004 IEEE International Conference on Robotics and Automation*, (New Orleans, LA), pp. 4486–4491, April 2004.

[6] G. Antonelli, *Fault diagnosis and tolerance for mechatronic systems. Recent advances*, ch. A survey of fault detection/tolerance strategies for AUVs and ROVs, pp. 109–127. Heidelberg, D: F. Caccavale, L. Villani, (Eds.), Springer Tracts in Advanced Robotics, Springer-Verlag, 2002.

[7] J. Thunberg, W. Song, E. Montijano, Y. Hong, and X. Hu, "Distributed attitude synchronization control of multi-agent systems with switching topologies," *Automatica*, vol. 50, no. 3, pp. 832–840, 2014.

[8] G. Antonelli, F. Arrichiello, F. Caccavale, and A. Marino, "Decentralized time-varying formation control for multi-robot systems," *The International Journal of Robotics Research*, vol. 33, pp. 1029–1043,, 2014.

[9] F. Basile, F. Caccavale, P. Chiacchio, J. Coppola, and A. Marino, "A decentralized kinematic control architecture for collaborative and cooperative multi-arm systems," *Mechatronics*, vol. 23, no. 8, pp. 1100–1112, 2013.

[10] F. Caccavale and F. Pierri, "An approach to distributed estimation of time-varying signals by multi-agent systems," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 3556–3561, IEEE, 2022.

[11] K. A. Demir, G. Döven, and B. Sezen, "Industry 5.0 and human-robot coworking," *Procedia Computer Science*, vol. 158, pp. 688–695, 2019. 3rd WORLD CONFERENCE ON TECHNOLOGY, INNOVATION AND ENTREPRENEURSHIP"INDUSTRY 4.0 FOCUSED INNOVATION, TECHNOLOGY, ENTREPRENEURSHIP AND MANUFACTURE" June 21-23, 2019.

[12] K. I. Kim and Y. F. Zheng, "Two strategies of position and force control for two industrial robots handling a single object," *Robotics and Autonomous Systems*, vol. 5, no. 4, pp. 395–403, 1989.

[13] R. Setchi, M. B. Dehkordi, and J. S. Khan, "Explainable robotics in human-robot interactions," *Procedia Computer Science*, vol. 176, pp. 3057–3066, 2020. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 24th International Conference KES2020.

[14] D. Almeida and Y. Karayiannidis, "Asymmetric Dual-Arm Task Execution Using an Extended Relative Jacobian," in *Robotics Research* (T. Asfour, E. Yoshida, J. Park, H. Christensen, and O. Khatib, eds.), vol. 20, pp. 18–34, Cham: Springer International Publishing, 2022. Series Title: Springer Proceedings in Advanced Robotics.

[15] F. Caccavale, G. Giglio, G. Muscio, and F. Pierri, "Cooperative impedance control for multiple uavs with a robotic arm," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2366–2371, IEEE, 2015.

[16] X. An, C. Wu, Y. Lin, M. Lin, T. Yoshinaga, and Y. Ji, "Multi-robot systems and cooperative object transport: Communications, platforms, and challenges," *IEEE Open Journal of the Computer Society*, vol. 4, pp. 23–36, 2023.

[17] M. Uchiyama and P. Dauchez, "A symmetric hybrid position/force control scheme for the coordination of two robots," in *Proc. of IEEE Int.Conf. on Robotics and Automation*, pp. 350–356, 1988.

[18] P. Chiacchio, S. Chiaverini, and B. Siciliano, "Direct and inverse kinematics for coordinated motion tasks of a two-manipulator system," *Journal of Dynamic Systems, Measurement, and Control*, vol. 118, pp. 691–697, 12 1996.

[19] F. Huzaefa and Y.-C. Liu, "Centralized control architecture for cooperative object transportation using multiple omnidirectional agvs," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6526–6532, IEEE, 2019.

[20] H. A. Park and C. G. Lee, "Dual-arm coordinated-motion task specification and performance evaluation," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 929–936, 2016.

[21] P. Arcara and C. Melchiorri, "Control schemes for teleoperation with time delay: A comparative study," *Robotics and Autonomous Systems*, vol. 38, no. 1, pp. 49–64, 2002.

[22] S. Zampieri, "Trends in networked control systems," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 2886–2894, 2008. 17th IFAC World Congress.

[23] A. Bahr, J. J. Leonard, and M. F. Fallon, "Cooperative localization for autonomous underwater vehicles," *The International Journal of Robotics Research*, vol. 28, no. 6, pp. 714–728, 2009.

[24] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 693–703, 2007.

[25] C. K. Verginis, A. Nikou, and D. V. Dimarogonas, "Communication-based decentralized cooperative object transportation using nonlinear model predictive control," in *2018 European Control Conference (ECC)*, pp. 733–738, 2018.

[26] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control systems magazine*, vol. 27, no. 2, pp. 71–82, 2007.

[27] R. Saber and R. Murray, "Consensus protocols for networks of dynamic agents," in *Proceedings of the 2003 American Control Conference, 2003.*, vol. 2, pp. 951–956, 2003.

[28] D. Spanos, R. Olfati-Saber, and R. Murray, "Dynamic consensus on mobile networks," in *IFAC World Congress*, 2005.

[29] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[30] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson, "Distributed event-triggered control for multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1291–1297, 2012.

[31] P. B. g. Dohmann and S. Hirche, "Distributed control for cooperative manipulation with event-triggered communication," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1038–1052, 2020.

[32] E. Simetti and G. Casalino, "Manipulation and transportation with cooperative underwater vehicle manipulator systems," *IEEE Journal of Oceanic Engineering*, vol. 42, no. 4, pp. 782–799, 2017.

[33] Y. Ren, S. Sosnowski, and S. Hirche, "Fully distributed cooperation for networked uncertain mobile manipulators," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 984–1003, 2020.

[34] F. Caccavale and F. Pierri, "Distributed source-user estimation over directed graphs," *Automatica*, vol. 155, p. 111116, 2023.

[35] M. Wada and R. Torii, "Cooperative transportation of a single object by omnidirectional robots using potential method," in *2013 16th International Conference on Advanced Robotics (ICAR)*, pp. 1–6, 2013.

[36] Z. Wang, M. Admadabadi, E. Nakano, and T. Takahashi, "A multiple robot system for cooperative object transportation with various requirements on task performing," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 2, pp. 1226–1233 vol.2, 1999.

[37] A. Tagliabue, M. Kamel, S. Verling, R. Siegwart, and J. Nieto, "Collaborative transportation using mavs via passive force control," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5766–5773, IEEE, 2017.

[38] A. Tsiamis, C. K. Verginis, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Cooperative manipulation exploiting only implicit communication," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 864–869, 2015.

[39] A. Heins and A. P. Schoellig, "Keep it upright: Model predictive control for nonprehensile object transportation with obstacle avoidance on a mobile manipulator," *IEEE Robotics and Automation Letters*, 2023.

[40] J. Markdahl, Y. Karayiannidis, X. Hu, and D. Kragic, "Distributed cooperative object attitude manipulation," in *2012 IEEE International Conference on Robotics and Automation*, pp. 2960–2965, IEEE, 2012.

[41] A. Yufka, O. Parlaktuna, and M. Ozkan, "Formation-based cooperative transportation by a group of non-holonomic mobile robots," in *2010 IEEE International Conference on Systems, Man and Cybernetics*, pp. 3300–3307, 2010.

[42] X. Chen and Y. Li, "Cooperative transportation by multiple mobile manipulators using adaptive nn control," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pp. 4193–4200, 2006.

[43] L. Zhang, Y. Sun, A. Barth, and O. Ma, "Decentralized control of multi-robot system in cooperative object transportation using deep reinforcement learning," *IEEE Access*, vol. 8, pp. 184109–184119, 2020.

[44] D. Zeng, Y. Wang, Y. Jiang, H. Tan, Z. Miao, and Y. Feng, "Distributed neural adaptive impedance control for cooperative manipulation with unknown objects," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 7, pp. 12363–12376, 2025.

[45] Y. Li, L. Wang, K. Liu, W. He, Y. Yin, and R. Johansson, "Distributed neural-network-based cooperation control for teleoperation of multiple mobile manipulators under round-robin protocol," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 8, pp. 4841–4855, 2023.

[46] N. Hogan, "Impedance control: An approach to manipulation: Parts i-iii," *Journal of dynamic systems, measurement, and control*, vol. 107, no. 1, pp. 1–24, 1985.

[47] R. G. Bonitz and T. C. Hsia, "Internal force-based impedance control for cooperating manipulators," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 1, pp. 78–89, 1996.

[48] C. Ott, R. Mukherjee, and Y. Nakamura, "Unified impedance and admittance control," in *2010 IEEE international conference on robotics and automation*, pp. 554–561, IEEE, 2010.

[49] F. Caccavale, C. Natale, B. Siciliano, and L. Villani, "Experiments of spatial impedance control," in *Experimental Robotics V* (A. Casals and A. T. de Almeida, eds.), (Berlin, Heidelberg), pp. 91–104, Springer Berlin Heidelberg, 1998.

[50] Z. Li, C. Yang, C.-Y. Su, S. Deng, F. Sun, and W. Zhang, "Decentralized fuzzy control of multiple cooperating robotic manipulators with impedance interaction," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 4, pp. 1044–1056, 2014.

[51] S. Heshmati-Alamdari, C. P. Bechlioulis, G. C. Karras, and K. J. Kyriakopoulos, "Decentralized impedance control for cooperative manipulation of multiple underwater vehicle manipulator systems under lean communication," in *2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV)*, pp. 1–6, IEEE, 2018.

[52] M. Pokornỳ, J. Nowaková, and T. Dočekal, "Adaptive ascent control of a collaborative object transportation system using two quadrotors," *Sensors*, vol. 22, no. 8, p. 2923, 2022.

[53] C. K. Verginis, M. Mastellaro, and D. V. Dimarogonas, "Robust cooperative manipulation without force/torque measurements: Control design and experiments," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 3, pp. 713–729, 2020.

[54] S. Erhart and S. Hirche, "Adaptive force/velocity control for multi-robot cooperative manipulation under uncertain kinematic parameters," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 307–314, 2013.

[55] F. Huzaefa and Y.-C. Liu, "Force distribution and estimation for cooperative transportation control on multiple unmanned ground vehicles," *IEEE Transactions on Cybernetics*, vol. 53, no. 2, pp. 1335–1347, 2021.

[56] M. Logothetis, G. C. Karras, K. Alevizos, C. K. Verginis, P. Roque, K. Roditakis, A. Makris, S. Garcıa, P. Schillinger, A. Di Fava, *et al.*, "A decentralized framework for efficient cooperation of heterogeneous robotic agents," *IEEE Robot Autom Mag*, vol. 28, pp. 75–87, 2021.

[57] F. Caccavale, P. Chiacchio, A. Marino, and L. Villani, "Six-dof impedance control of dual-arm cooperative manipulators," *IEEE/ASME Transactions On Mechatronics*, vol. 13, no. 5, pp. 576–586, 2008.

[58] A. Marino, "Distributed adaptive control of networked cooperative mobile manipulators," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 5, pp. 1646–1660, 2017.

[59] A. Marino and F. Pierri, "A two stage approach for distributed cooperative manipulation of an unknown object without explicit communication and unknown number of robots," *Robotics and Autonomous Systems*, vol. 103, pp. 122–133, 2018.

[60] S. Erhart and S. Hirche, "Internal force analysis and load distribution for cooperative multi-robot manipulation," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1238–1243, 2015.

[61] C. K. Verginis, D. Zelazo, and D. V. Dimarogonas, "Cooperative manipulation via internal force regulation: A rigidity theory perspective," *IEEE Transactions on Control of Network Systems*, vol. 10, no. 3, pp. 1222–1233, 2023.

[62] D. Romero, J. Stahre, T. Wuest, O. Noran, P. Bernus, Å. Fast-Berglund, and D. Gorecky, "Towards an operator 4.0 typology: a human-centric perspective on the fourth industrial revolution technologies," in *proceedings of the international conference on computers and industrial engineering (CIE46), Tianjin, China*, pp. 29–31, 2016.

[63] J. Alves, T. M. Lima, and P. D. Gaspar, "Is industry 5.0 a human-centred approach? a systematic review," *Processes*, vol. 11, no. 1, 2023.

[64] D.-G. f. R. European Commission and M.-D. N. L. P. A. Innovation, Breque, *Industry 5.0 – Towards a sustainable, human-centric and resilient European industry*. Publications Office of the European Union, 2021.

[65] Y. Lu, H. Zheng, S. Chand, W. Xia, Z. Liu, X. Xu, L. Wang, Z. Qin, and J. Bao, "Outlook on human-centric manufacturing towards industry 5.0," *Journal of Manufacturing Systems*, vol. 62, pp. 612–627, 2022.

[66] C. N. Mavridis, K. Alevizos, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Human-robot collaboration based on robust motion intention estimation with prescribed performance," in *2018 European Control Conference (ECC)*, pp. 249–254, 2018.

[67] A. M. Zanchettin, N. M. Ceriani, P. Rocco, H. Ding, and B. Matthias, "Safety in human-robot collaborative manufacturing environments: Metrics and control," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 882–893, 2016.

[68] "Robots and robotic devices — Collaborative robots," standard, International Organization for Standardization, Feb. 2016.

[69] "Robots and robotic devices — Safety requirements for industrial robots Part 2: Robot systems and integration," standard, International Organization for Standardization, July 2011.

[70] "Robotics — Safety requirements Part 2: Industrial robot applications and robot cells," standard, International Organization for Standardization, Feb. 2025.

[71] B. Gladysz, T. anh Tran, D. Romero, T. van Erp, J. Abonyi, and T. Ruppert, "Current development on the operator 4.0 and transition towards the operator 5.0: A systematic literature review in light of industry 5.0," *Journal of Manufacturing Systems*, vol. 70, pp. 160–185, 2023.

[72] M. Bdiwi, M. Pfeifer, and A. Sterzing, "A new strategy for ensuring human safety during various levels of interaction with industrial robots," *CIRP Annals*, vol. 66, no. 1, pp. 453–456, 2017.

[73] R. Suzuki, A. Karim, T. Xia, H. Hedayati, and N. Marquardt, "Augmented reality and robotics: A survey and taxonomy for ar-enhanced human-robot interaction and robotic interfaces," in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, (New York, NY, USA), Association for Computing Machinery, 2022.

[74] C. P. Quintero, S. Li, M. K. Pan, W. P. Chan, H. Machiel Van der Loos, and E. Croft, "Robot programming through augmented trajectories in augmented reality," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1838–1844, 2018.

[75] S. H. Choi, K.-B. Park, D. H. Roh, J. Y. Lee, M. Mohammed, Y. Ghasemi, and H. Jeong, "An integrated mixed reality system for safety-aware human-robot collaboration using deep learning and digital twin generation," *Robotics and Computer-Integrated Manufacturing*, vol. 73, p. 102258, 2022.

[76] A. Hietanen, R. Pieters, M. Lanz, J. Latokartano, and J.-K. Kämäräinen, "Ar-based interaction for human-robot collaborative manufacturing," *Robotics and Computer-Integrated Manufacturing*, vol. 63, p. 101891, 2020.

[77] K. Merckaert, B. Convens, C. ju Wu, A. Roncone, M. M. Nicotra, and B. Vanderborght, "Real-time motion control of robotic manipulators for safe human–robot coexistence," *Robotics and Computer-Integrated Manufacturing*, vol. 73, p. 102223, 2022.

[78] A. De Santis, B. Siciliano, A. De Luca, and A. Bicchi, "An atlas of physical human–robot interaction," *Mechanism and Machine Theory*, vol. 43, no. 3, pp. 253–270, 2008.

[79] A. Bicchi, M. Bavaro, G. Boccadamo, D. De Carli, R. Filippini, G. Grioli, M. Piccigallo, A. Rosi, R. Schiavi, S. Sen, and G. Tonietti, "Physical human-robot interaction: Dependability, safety, and performance," in *2008 10th IEEE International Workshop on Advanced Motion Control*, pp. 9–14, 2008.

[80] A. de Luca and R. Mattone, "Sensorless robot collision detection and hybrid force/motion control," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 999–1004, 2005.

[81] S. Haddadin, A. De Luca, and A. Albu-Schäffer, "Robot collisions: A survey on detection, isolation, and identification," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1292–1312, 2017.

[82] A. Kouris, F. Dimeas, and N. Aspragathos, "A frequency domain approach for contact type distinction in human–robot collaboration," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 720–727, 2018.

[83] M. Lippi and A. Marino, "Enabling physical human-robot collaboration through contact classification and reaction," in *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 1196–1203, 2020.

[84] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human-robot collaboration," *Autonomous Robots*, vol. 42, 06 2018.

[85] T. Chakraborti, S. Sreedharan, A. Kulkarni, and S. Kambhampati, "Projection-aware task planning and execution for human-in-the-loop operation of robots in a mixed-reality workspace," in *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4476–4482, 2018.

[86] M. Selvaggio, M. Cognetti, S. Nikolaidis, S. Ivaldi, and B. Siciliano, "Autonomy in physical human-robot interaction: A brief survey," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7989–7996, 2021.

[87] A. Mörtl, M. Lawitzky, A. Kucukyilmaz, M. Sezgin, C. Basdogan, and S. Hirche, "The role of roles: Physical cooperation between humans and robots," *The International Journal of Robotics Research*, vol. 31, no. 13, pp. 1656–1674, 2012.

[88] Y. Pan, Z. Li, T. Shi, and C. Wen, "Composite learning variable impedance robot control with stability and passivity guarantees," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 119–126, 2024.

[89] M. Mujica, M. Crespo, M. Benoussaad, S. Junco, and J.-Y. Fourquet, "Robust variable admittance control for human–robot co-manipulation of objects with unknown load," *Robotics and Computer-Integrated Manufacturing*, vol. 79, p. 102408, 2023.

[90] A. Sidiropoulos, Y. Karayiannidis, and Z. Doulgeri, "Human-robot collaborative object transfer using human motion prediction based on cartesian pose dynamic movement primitives," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3758–3764, 2021.

[91] E. Yang and C. Mavrogiannis, "Implicit communication in human-robot collaborative transport," 2025.

[92] H. Arai, T. Takubo, Y. Hayashibara, and K. Tanie, "Human-robot cooperative manipulation using a virtual nonholonomic constraint," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 4, pp. 4063–4069 vol.4, 2000.

[93] M. L. Elwin, B. Strong, R. A. Freeman, and K. M. Lynch, "Human-multirobot collaborative mobile manipulation: The omnid mocobots," *IEEE Robotics and Automation Letters*, vol. 8, p. 376–383, Jan. 2023.

[94] Y. Hirata, H. Song, Z. Wang, and K. Kosuge, "Control of passive object handling robot with free joint for reducing human assistive force," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1154–1159, 2007.

[95] K. Kosuge and Y. Hirata, "Human-robot interaction," *2004 IEEE International Conference on Robotics and Biomimetics*, pp. 8–11, 2004.

[96] J. Chen, M. Gauci, W. Li, A. Kolling, and R. Groß, "Occlusion-based cooperative transport with a swarm of miniature mobile robots," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 307–321, 2015.

[97] D. Sirintuna, A. Giammarino, and A. Ajoudani, "An object deformation-agnostic framework for human–robot collaborative transportation," *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 2, pp. 1986–1999, 2024.

[98] G. Nicola, E. Villagrossi, and N. Pedrocchi, "Co-manipulation of soft-materials estimating deformation from depth images," *Robotics and Computer-Integrated Manufacturing*, vol. 85, p. 102630, 2024.

[99] G. Cortigiani, M. Malvezzi, D. Prattichizzo, and M. Pozzi, "Human-robot collaborative cable-suspended manipulation with contact distinction," *IEEE Robotics and Automation Letters*, vol. 10, no. 1, pp. 740–747, 2025.

[100] G. Li, X. Liu, and G. Loianno, "Human-aware physical human–robot collaborative transportation and manipulation with multiple aerial robots," *IEEE Transactions on Robotics*, vol. 41, pp. 762–781, 2025.

[101] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics – Modelling, Planning and Control*. London, UK: Springer, 2009.

[102] F. Caccavale and M. Uchiyama, *Cooperative Manipulators*, pp. 701–718. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[103] F. Pierri and A. Franchi, *Cooperative Manipulation*. Germany: Springer, living edition ed., 2023.

[104] J. Brewer, "Kronecker products and matrix calculus in system theory," *IEEE Transactions on Circuits and Systems*, vol. 25, no. 9, pp. 772–781, 1978.

[105] G. Antonelli, "Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems," *IEEE Transactions on Robotics*, vol. 25, pp. 985–994, October 2009.

[106] F. Bullo, *Lectures on network systems*, vol. 1. Kindle Direct Publishing Seattle, DC, USA, 2020.

[107] W. Ren and N. Sorensen, "Distributed coordination architecture for multi-robot formation control," *Robotics and Autonomous Systems*, vol. 56, pp. 24–333,, 2008.

[108] R. Horn and C. Johnson, *Matrix analysis*. Cambridge Univ Pr, 1990.

[109] B. Kågström, "Bounds and perturbation bounds for the matrix exponential," *BIT Numerical Mathematics*, vol. 17, no. 1, pp. 39—57, 1977.

[110] C. V. Loan, "The sensitivity of the matrix exponential," *SIAM Journal on Numerical Analysis*, vol. 14, no. 6, pp. 971–981, 1977.

[111] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Verlag, 2009.

[112] H. Khalil, *Nonlinear Systems*. Upper Saddle River, New Jersey: Prentice-Hall, 2nd ed., 1996.

[113] M. Ceccarelli, "Screw axis defined by giulio mozzi in 1763 and early studies on helicoidal motion," *Mechanism and Machine Theory*, vol. 35, no. 6, pp. 761–770, 2000.

[114] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano, "Global task space manipulability ellipsoids for multiple-arm systems," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 5, pp. 678–685, 1991.

[115] M. Sileo, Y. Karayiannidis, F. Pierri, and F. Caccavale, "Decentralized leader-follower control for centroid and formation tracking," in *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 3349–3354, IEEE, 2023.

[116] G. Carriero, M. Sileo, Y. Karayiannidis, F. Pierri, and F. Caccavale, "Decentralized control of internal wrenches in a multi-manipulators object transportation task," in *2025 23rd European Control Conference (ECC)*, pp. 705–710, IEEE, 2025.

[117] A. A. Oliva, F. Spindler, P. R. Giordano, and F. Chaumette, "Frankasim: A dynamic simulator for the franka emika robot with visual-servoing enabled capabilities," in *2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 868–874, IEEE, 2022.

[118] E. Rohmer, S. P. N. Singh, and M. Freese, "Coppeliasim (formerly v-rep): a versatile and scalable robot simulation framework," in *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2013. www.coppeliarobotics.com.

[119] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, p. 5, Kobe, Japan, 2009.

[120] E. Marchand, F. Spindler, and F. Chaumette, "Visp for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robotics and Automation Magazine*, vol. 12, pp. 40–52, December 2005.

[121] S. Haddadin, S. Parusel, L. Johannsmeier, S. Golz, S. Gabl, F. Walch, M. Sabaghian, C. Jähne, L. Hausperger, and S. Haddadin, "The franka emika robot: A reference platform for robotics research and education," *IEEE Robotics and Automation Magazine*, vol. 29, no. 2, pp. 46–64, 2022.

[122] P. Merriaux, Y. Dupuis, R. Boutteau, P. Vasseur, and X. Savatier, "A study of vicon system positioning performance," *Sensors*, vol. 17, no. 7, 2017.

[123] G. Carriero, M. Sileo, S. Fregnan, M. Guberina, F. Pierri, F. Caccavale, and Y. Karayiannidis, "Decentralized admittance control for a multi-manipulator system: implementation and analysis," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4574–4580, 2025.

[124] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse, and N. Mansard, "The pinocchio c++ library – a fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," in *IEEE International Symposium on System Integrations (SII)*, 2019.

[125] A. Babinec, L. Jurišica, P. Hubinskỳ, and F. Duchoň, "Visual localization of mobile robot using artificial markers," *Procedia Engineering*, vol. 96, pp. 1–9, 2014.

[126] G. Carriero, M. Sileo, S. Fregnan, F. Pierri, F. Caccavale, and Y. Karayiannidis, "Decentralized admittance control for a multi–manipulator system: theory and experiments," 2026. Submitted to *2026 IEEE International Conference on Robotics and Automation (ICRA)*.

[127] R. Salvato, G. Marra, P. Scardamaglia, G. Di Gironimo, D. Marzullo, and R. Mozzillo, "Design and integration of automation systems with manual operation: small and medium enterprises issues," in *Design Tools and Methods in Industrial Engineering II: Proceedings of the Second International Conference on Design Tools and Methods in Industrial Engineering, ADM 2021, September 9–10, 2021, Rome, Italy*, pp. 298–307, Springer, 2022.

[128] P. Iaccarino, S. Inserra, P. Cerreta, and R. Mozzillo, "Determinant assembly approach for flat-shaped airframe components," *The International Journal of Advanced Manufacturing Technology*, vol. 108, no. 7-8, pp. 2433–2443, 2020.

[129] T. Sheridan, "Humans and automation: System design and research issues," 01 2002.

[130] A. Weiss, A.-K. Wortmeier, and B. Kubicek, "Cobots in industry 4.0: A roadmap for future practice studies on human–robot collaboration," *IEEE Transactions on Human-Machine Systems*, vol. 51, no. 4, pp. 335–345, 2021.

[131] S. A. Green, M. Billinghurst, X. Chen, and J. G. Chase, "Human-robot collaboration: A literature review and augmented reality approach in design," *International journal of advanced robotic systems*, vol. 5, no. 1, p. 1, 2008.

[132] T. Li, S. Yu, B. Sun, Y. Li, X. Wang, Y. Pan, C. Song, Y. Ren, Z. Zhang, K. T. Grattan, *et al.*, "Bioinspired claw-engaged and biolubricated swimming microrobots creating active retention in blood vessels," *Science Advances*, vol. 9, no. 18, p. eadg4501, 2023.

[133] H. Wang, S. Yu, J. Liao, X. Qing, D. Sun, F. Ji, W. Song, L. Wang, and T. Li, "A robot platform for highly efficient pollutant purification," *Frontiers in Bioengineering and Biotechnology*, vol. 10, p. 903219, 2022.

[134] L. Guo, Z. Lu, and L. Yao, "Human-machine interaction sensing technology based on hand gesture recognition: A review," *IEEE Transactions on Human-Machine Systems*, vol. 51, no. 4, pp. 300–309, 2021.

[135] D. Sun, A. Kiselev, Q. Liao, T. Stoyanov, and A. Loutfi, "A new mixed-reality-based teleoperation system for telepresence and maneuverability enhancement," *IEEE Transactions on Human-Machine Systems*, vol. 50, no. 1, pp. 55–67, 2020.

[136] A. De Franco, E. Lamon, P. Balatti, E. De Momi, and A. Ajoudani, "An intuitive augmented reality interface for task scheduling, monitoring, and work performance improvement in human-robot collaboration," in *2019 IEEE International Work Conference on Bioinspired Intelligence (IWOBI)*, pp. 75–80, IEEE, 2019.

[137] Unity Technologies, "Mixed reality toolkit 3," 2023. Mixed Reality Toolkit.

[138] E. Rosen, D. Whitney, E. Phillips, G. Chien, J. Tompkin, G. Konidaris, and S. Tellex, "Communicating and controlling robot arm motion intent through mixed-reality head-mounted displays," *The International Journal of Robotics Research*, vol. 38, no. 12-13, pp. 1513–1526, 2019.

[139] S. Y. Gadre, E. Rosen, G. Chien, E. Phillips, S. Tellex, and G. Konidaris, "End-user robot programming using mixed reality," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 2707–2713, 2019.

[140] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," in *Advances in psychology*, vol. 52, pp. 139–183, Elsevier, 1988.

[141] L. Kästner and J. Lambrecht, "Augmented-reality-based visualization of navigation data of mobile robots on the microsoft hololens-possibilities and limitations," in *2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, pp. 344–349, IEEE, 2019.

[142] M. Ostanin, S. Mikhel, A. Evlampiev, V. Skvortsova, and A. Klimchik, "Human-robot interaction for robotic manipulator programming in mixed reality," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2805–2811, IEEE, 2020.

[143] Microsoft, "Hololens 2.," 2019. https://www.microsoft.com/en-us/hololens/hardware [Accessed: 2025-10-29].

[144] N. Calzone, M. Sileo, R. Mozzillo, F. Pierri, and F. Caccavale, "Mixed reality platform supporting human-robot interaction," in *Advances on Mechanics, Design Engineering and Manufacturing IV: Proceedings of the International Joint Conference on Mechanics, Design Engineering & Advanced Manufacturing, JCM 2022, June 1-3, 2022, Ischia, Italy*, pp. 1172–1182, Springer, 2022.

[145] J. Brooke, "Sus: a "quick and dirty'usability," *Usability evaluation in industry*, vol. 189, no. 3, pp. 189–194, 1996.

[146] Unity Technologies, "Unity," 2023. Game development platform.

[147] Siemens, "GitHub. ROS#.." https://github.com/siemens/ros-sharp, 2012.

[148] G. Carriero, N. Calzone, M. Sileo, F. Pierri, F. Caccavale, and R. Mozzillo, "Human-robot collaboration: An augmented reality toolkit for bi-directional interaction," *Applied Sciences*, vol. 13, no. 20, 2023.

[149] S. G. Hart, "Nasa-task load index (nasa-tlx); 20 years later," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 50, no. 9, pp. 904–908, 2006.

[150] G. M. Sullivan and A. R. Artino Jr, "Analyzing and interpreting data from likert-type scales," *Journal of graduate medical education*, vol. 5, no. 4, pp. 541–542, 2013.

[151] J. Cohen, *Statistical power analysis for the behavioral sciences*. Academic press, 2013.

[152] M. Sileo, G. Carriero, C. Brancato, N. Calzone, F. Pierri, F. Caccavale, and R. Mozzillo, "Safety features for humantix: An augmented reality platform to enhance human–robot collaboration," in *Advances on Mechanics, Design Engineering and Manufacturing V* (C. Manchado del Val, R. Miralbes Buil, G. Peris Fajarnés, M. Moncho Santonja, C. Rizzi, and L. Roucoules, eds.), (Cham), pp. 557–572, Springer Nature Switzerland, 2025.

[153] P. Di Lillo, F. Pierri, G. Antonelli, F. Caccavale, and A. Ollero, "A framework for set-based kinematic control of multi-robot systems," *Control Engineering Practice*, vol. 106, p. 104669, 2021.

[154] H. Khalil, *Nonlinear Systems*. Upper Saddle River, New Jersey: Prentice-Hall, 3rd ed., 2002.

Appendices

A1 - Proof of Theorem 2

Under Assumption 5, for $h = p$, the estimation error dynamics (**??**) becomes

$$\dot{\widetilde{\boldsymbol{x}}}_{d,p} = -\bar{\boldsymbol{L}}\, \widetilde{\boldsymbol{x}}_{d,p}\,. \tag{1}$$

In turn, for $h = p-1, \ldots, 1, 0$, the estimation error dynamics (**??**) is given by

$$\dot{\widetilde{\boldsymbol{x}}}_{d,h} = -\bar{\boldsymbol{L}}\, \widetilde{\boldsymbol{x}}_{d,h} + \widetilde{\boldsymbol{x}}_{d,h+1}\,. \tag{2}$$

Thus, the overall estimation error dynamics can be expressed as the following linear time-invariant system with a block-triangular structure:

$$\begin{bmatrix} \dot{\widetilde{\boldsymbol{x}}}_{d,p} \\ \dot{\widetilde{\boldsymbol{x}}}_{d,p-1} \\ \dot{\widetilde{\boldsymbol{x}}}_{d,p-2} \\ \vdots \\ \dot{\widetilde{\boldsymbol{x}}}_{d,0} \end{bmatrix} = \begin{bmatrix} -\bar{\boldsymbol{L}} & \boldsymbol{O}_{N_t} & \boldsymbol{O}_{N_t} & \dots & \boldsymbol{O}_{N_t} \\ \boldsymbol{I}_{N_t} & -\bar{\boldsymbol{L}} & \boldsymbol{O}_{N_t} & \dots & \boldsymbol{O}_{N_t} \\ \boldsymbol{O}_{N_t} & \boldsymbol{I}_{N_t} & -\bar{\boldsymbol{L}} & \dots & \boldsymbol{O}_{N_t} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ \boldsymbol{O}_{N_t} & \boldsymbol{O}_{N_t} & \boldsymbol{O}_{N_t} & \dots & -\bar{\boldsymbol{L}} \end{bmatrix} \begin{bmatrix} \widetilde{\boldsymbol{x}}_{d,p} \\ \widetilde{\boldsymbol{x}}_{d,p-1} \\ \widetilde{\boldsymbol{x}}_{d,p-2} \\ \vdots \\ \widetilde{\boldsymbol{x}}_{d,0} \end{bmatrix}. \tag{3}$$

Since the eigenvalues of the matrix in (3) coincide with those of the diagonal blocks and $-\bar{\boldsymbol{L}}$ is Hurwitz, the system in (3) is exponentially stable.

A2 - Proof of Theorem 3

For $h = p$, the estimation error dynamics (**??**) is given by the linear time-invariant system in (4) with the bounded input $\mathbf{1}_N \otimes \boldsymbol{x}_d^{(p+1)}$

$$\dot{\widetilde{\boldsymbol{x}}}_{d,p} = -\bar{\boldsymbol{L}}\, \widetilde{\boldsymbol{x}}_{d,p} + (\mathbf{1}_N \otimes \boldsymbol{x}_d^{(p+1)}). \tag{4}$$

The expression of the estimation error at the generic time instant can be computed from the Lagrange's formula for linear systems:

$$\widetilde{\boldsymbol{x}}_{d,p}(t) = \boldsymbol{\Phi}(t, t_0)\widetilde{\boldsymbol{x}}_{d,p}(t_0) + \int_{t_0}^{t} \boldsymbol{\Phi}(t, \tau) \left(\mathbf{1}_N \otimes \boldsymbol{x}^{(p+1)}\right) d\tau. \tag{5}$$

Leveraging (2.27) from Lemma 1 and Assumption 6, $\widetilde{\boldsymbol{x}}_{d,p}(t)$ can be bounded as follows (the argument ϵ of ρ and α is dropped for the sake of compactness)

$$\begin{aligned}\|\widetilde{\boldsymbol{x}}_{d,p}(t)\| &\leq \rho e^{-\alpha(t-t_0)}\,\|\widetilde{\boldsymbol{x}}_{d,p}(t_0)\| + \int_{t_0}^{t} \rho e^{-\alpha(t-\tau)}\left\|\mathbf{1}_N \otimes \boldsymbol{x}^{(p+1)}\right\| d\tau \leq \\ &\leq \rho e^{-\alpha(t-t_0)}\,\|\widetilde{\boldsymbol{x}}_{d,p}(t_0)\| + \rho\sqrt{N}\,\bar{x}^{(p+1)} \int_{t_0}^{t} e^{-\alpha(t-\tau)} d\tau = \\ &= \rho\, e^{-\alpha(t-t_0)}\,\|\widetilde{\boldsymbol{x}}_{d,p}(t_0)\| + \rho\frac{\sqrt{N}\,\bar{x}^{(p+1)}}{\alpha}\left(1 - e^{-\alpha(t-t_0)}\right) \leq \\ &\leq \rho\left(\|\widetilde{\boldsymbol{x}}_{d,p}(t_0)\| + \frac{\sqrt{N}\,\bar{x}^{(p+1)}}{\alpha}\right) = \bar{x}_d^{(p)}.\end{aligned} \tag{6}$$

For $h = p-1, \ldots, 1, 0$, the estimation error dynamics (**??**) is described by the linear and time-invariant system with the bounded input $\widetilde{\boldsymbol{x}}_{d,h+1}$

$$\dot{\widetilde{\boldsymbol{x}}}_{d,h} = -\bar{\boldsymbol{L}}\,\widetilde{\boldsymbol{x}}_{d,h} + \widetilde{\boldsymbol{x}}_{d,h+1}\,.$$

By applying the same arguments, a bound for $\widetilde{\boldsymbol{x}}_{d,p-1}(t)$ can be established:

$$\begin{aligned}\|\widetilde{\boldsymbol{x}}_{d,p-1}(t)\| &\leq \rho e^{-\alpha(t-t_0)}\,\|\widetilde{\boldsymbol{x}}_{d,p-1}(t_0)\| + \int_{t_0}^{t} \rho e^{-\alpha(t-\tau)}\,\|\widetilde{\boldsymbol{x}}_{d,p}(\tau)\|\, d\tau \leq \\ &\leq \rho e^{-\alpha(t-t_0)}\,\|\widetilde{\boldsymbol{x}}_{d,p-1}(t_0)\| + \frac{\rho}{\alpha}\,\|\widetilde{\boldsymbol{x}}_{d,p}(t)\|\left(1 - e^{-\alpha(t-t_0)}\right)\end{aligned} \tag{7}$$

and substituting the bound of (6),

$$\|\widetilde{\boldsymbol{x}}_{d,p-1}(t)\| \leq\ \rho\,\|\widetilde{\boldsymbol{x}}_{d,p-1}(t_0)\| + \frac{\rho}{\alpha}\left(\rho\,\|\widetilde{\boldsymbol{x}}_{d,p}(t_0)\| + \frac{\rho}{\alpha}\sqrt{N}\bar{x}^{(p+1)}\right).$$

Thus, starting from (8), a bound for $\widetilde{\boldsymbol{x}}_{d,h}(t)$, for $h = p, p-1, \ldots, 1, 0$, can be recursively determined as

$$\|\widetilde{\boldsymbol{x}}_{d,h}(t)\| \leq \rho\sum_{l=h}^{p}\left(\frac{\rho}{\alpha}\right)^{l-h}\|\widetilde{\boldsymbol{x}}_{d,l+1}(t_0)\| + \left(\frac{\rho}{\alpha}\right)^{p-h+1}\sqrt{N}\bar{x}^{(p+1)} = \bar{x}_{d,0}^{(h)} + \bar{\beta}^{(h)}\bar{x}^{(p+1)}. \tag{8}$$

A3 - Proof of Corollary 1

By following the same arguments used in the proof of Theorem 3, the first term on the right hand member of the chain of inequalities in (6)-(7), $\rho e^{-\alpha(t-t_0)}\,\|\widetilde{\boldsymbol{x}}_{d,p}(t_0)\|$,

exponentially converges to zero for $t \to \infty$. In fact, for $h = p, p-1, \ldots, 0$, it yields

$$lim_{t\to\infty}\rho e^{-\alpha(t-t_0)} \left\|\widetilde{\boldsymbol{x}}_{d,h}(t_0)\right\| \longrightarrow 0. \tag{9}$$

A4 - Proof of Theorem 4

The tracking error dynamics (2.78) is described by a linear and time-invariant system with $\boldsymbol{p}_d$, $\boldsymbol{p}_{eq}$ and $\hat{\boldsymbol{f}}_e$ as inputs. Due to Assumption 7, $\boldsymbol{p}_d$ is bounded, similarly, due to the elastic model of contact, $\boldsymbol{p}_{eq}$ is bounded as well. Since the system is modeled as a network of integrator agents with balanced strongly connected digraph characterized by a fixed topology, the linear consensus protocol in (2.64) globally asymptotically solves the average-consensus problem, according to Theorem 4 in [29]. Moreover, by resorting to the properties of the estimation scheme in [29], also the estimated internal forces vector $\hat{\boldsymbol{f}}_I$ is bounded.

Consider the homogeneous system associated to (2.78), it is exponentially convergent to zero, provided that the matrix $\boldsymbol{A}_e$ is positive definite. If the matrices $\boldsymbol{K}$, $\boldsymbol{K}_{D_p}$ and $\boldsymbol{K}_S$ are diagonal, the positive definiteness can be guaranteed if the condition

$$k_{D_{p,ii}} k_{ii} > \frac{N-1}{N} k_{S_{ii}},\ \forall i \in \{1, \ldots, 3N\}, \tag{10}$$

holds. Therefore, it follows that, due to the boundedness of inputs in (2.78), the collective position tracking error is bounded as well.

A5 - Proof of Corollary 2

Since $||\boldsymbol{e}|| = ||\boldsymbol{p}_d - \boldsymbol{p}||$ is bounded, also its time derivative $||\dot{\boldsymbol{e}}|| = ||\dot{\boldsymbol{p}}_d - \dot{\boldsymbol{p}}||$ is bounded given (2.75) and thus, due to the Assumption 7, $\dot{\boldsymbol{p}}$ is bounded as $||\dot{\boldsymbol{p}}|| \leq \dot{\bar{p}}$, where $\dot{\bar{p}}$ is a finite scalar value. Hence, by differentiating (2.74)

$$||\dot{\boldsymbol{f}}|| = \|\boldsymbol{K}_S\|\, ||\dot{\boldsymbol{x}}|| \leq \lambda_M(\boldsymbol{K}_S)\dot{\bar{p}}, \tag{11}$$

where $\lambda_M(\boldsymbol{K}_S)$ is the maximum eigenvalue of $\boldsymbol{K}_S$. Considering (2.67), since $\boldsymbol{\Lambda}_k$ is Hurwitz, it follows straightforwardly that $\widetilde{\boldsymbol{f}}_k$ is bounded, that proves the statement.

A6 - Proof of Lemma 2

Case study with 3 robots

Denoting as $\boldsymbol{p} = \left[\boldsymbol{p}_1^{\mathrm{T}},\, \boldsymbol{p}_2^{\mathrm{T}},\, \boldsymbol{p}_3^{\mathrm{T}}\right]^{\mathrm{T}}$ and as $\boldsymbol{p}_d = \left[\boldsymbol{p}_{d_1}^{\mathrm{T}},\, \boldsymbol{p}_{d_2}^{\mathrm{T}},\, \boldsymbol{p}_{d_3}^{\mathrm{T}}\right]^{\mathrm{T}}$ the collective actual and desired end-effector position vector, respectively, the i-th agent dynamics can be written as

$$\dot{\boldsymbol{p}}_i = \dot{\boldsymbol{p}}_{d,i} - \boldsymbol{K}_i(\boldsymbol{p}_i - \boldsymbol{p}_{d,i}) + \boldsymbol{K}_D^{-1}\hat{\boldsymbol{f}}_{I,i}. \tag{12}$$

In virtue of (2.76), the vector collecting the estimates of the internal force contributions ${}^i\hat{\boldsymbol{f}}_I$ can be expressed as:

$$\hat{\boldsymbol{f}}_I = \begin{bmatrix}\hat{\boldsymbol{f}}_{I_1}\\ \hat{\boldsymbol{f}}_{I_2}\\ \hat{\boldsymbol{f}}_{I_3}\end{bmatrix} = \boldsymbol{V}_p\boldsymbol{V}_p^{\dagger}\begin{bmatrix}\hat{\boldsymbol{f}}_1\\ \hat{\boldsymbol{f}}_2\\ \hat{\boldsymbol{f}}_3\end{bmatrix} = \begin{bmatrix}\frac{2}{3}\boldsymbol{f}_1 + \frac{1}{3}\hat{\boldsymbol{f}}_2 + \frac{1}{3}\hat{\boldsymbol{f}}_3\\ \frac{1}{3}\hat{\boldsymbol{f}}_1 + \frac{2}{3}\boldsymbol{f}_2 - \frac{1}{3}\hat{\boldsymbol{f}}_3\\ \frac{1}{3}\hat{\boldsymbol{f}}_1 - \frac{1}{3}\hat{\boldsymbol{f}}_2 + \frac{2}{3}\boldsymbol{f}_3\end{bmatrix} \tag{13}$$

Folding (2.74) in (13) and substituting it in (12), it follows:

$$\begin{aligned}\dot{\boldsymbol{p}}_1 - \dot{\boldsymbol{p}}_{d,1} + \boldsymbol{K}_1(\boldsymbol{p}_1 - \boldsymbol{p}_{d,1}) - \boldsymbol{K}_D^{-1}\left(\frac{2}{3}\boldsymbol{f}_1 + \frac{1}{3}\hat{\boldsymbol{f}}_2 + \frac{1}{3}\hat{\boldsymbol{f}}_3\right) = 0\\ \dot{\boldsymbol{e}}_1 + \left(\boldsymbol{K}_1 - \frac{2}{3}\boldsymbol{K}_D^{-1}\boldsymbol{K}_{S_1}\right)\boldsymbol{e}_1 - \frac{2}{3}\boldsymbol{K}_D^{-1}\boldsymbol{K}_{S_1}(\boldsymbol{p}_{d,1} - \boldsymbol{p}_{eq,1}) - \frac{1}{3}\boldsymbol{K}_D^{-1}\left(\hat{\boldsymbol{f}}_2 + \hat{\boldsymbol{f}}_3\right) = 0,\end{aligned} \tag{14}$$

where $\boldsymbol{e}_i = \boldsymbol{p}_i - \boldsymbol{p}_{d_i}$ stands for the i-th robot tracking error.

With similar arguments, the dynamics of $\boldsymbol{e}_2$ and $\boldsymbol{e}_3$ can be expressed as follows:

$$\dot{\boldsymbol{e}}_2 + \left(\boldsymbol{K}_2 - \frac{2}{3}\boldsymbol{K}_D^{-1}\boldsymbol{K}_{S_2}\right)\boldsymbol{e}_2 - \frac{2}{3}\boldsymbol{K}_D^{-1}\boldsymbol{K}_{S_2}(\boldsymbol{p}_{d,2} - \boldsymbol{p}_{eq,2}) - \frac{1}{3}\boldsymbol{K}_D^{-1}\left(\hat{\boldsymbol{f}}_1 - \hat{\boldsymbol{f}}_3\right) = 0 \tag{15}$$

$$\dot{\boldsymbol{e}}_3 + \left(\boldsymbol{K}_3 - \frac{2}{3}\boldsymbol{K}_D^{-1}\boldsymbol{K}_{S_3}\right)\boldsymbol{e}_3 - \frac{2}{3}\boldsymbol{K}_D^{-1}\boldsymbol{K}_{S_3}(\boldsymbol{p}_{d,3} - \boldsymbol{p}_{eq,3}) - \frac{1}{3}\boldsymbol{K}_D^{-1}\left(\hat{\boldsymbol{f}}_1 - \hat{\boldsymbol{f}}_2\right) = 0 \tag{16}$$

Then, the overall system dynamics can be rearranged in the matrix formulation as:

$$\dot{\boldsymbol{e}} + \left(\boldsymbol{K} - \frac{2}{3}\bar{\boldsymbol{K}}_D^{-1}\boldsymbol{K}_S\right)\boldsymbol{e} = \frac{2}{3}\bar{\boldsymbol{K}}_D^{-1}\boldsymbol{K}_S(\boldsymbol{p}_d - \boldsymbol{p}_{eq}) + \bar{\boldsymbol{K}}_D^{-1}\left(\boldsymbol{V}_p\boldsymbol{V}_p^{\dagger} - diag(\boldsymbol{V}_p\boldsymbol{V}_p^{\dagger})\right)\hat{\boldsymbol{f}}_I, \tag{17}$$

where $\boldsymbol{e} = \left[\boldsymbol{e}_1^{\mathrm{T}},\, \boldsymbol{e}_2^{\mathrm{T}},\, \boldsymbol{e}_3^{\mathrm{T}}\right]^{\mathrm{T}}$ is the collective position tracking error vector and the matrices $\boldsymbol{K}$, $\bar{\boldsymbol{K}}_D^{-1}$ and $\bar{\boldsymbol{K}}_S$ are block-diagonal matrices, representing the controller gain, the

virtual damping and the virtual stiffness of the system, respectively:

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}_1 & \boldsymbol{O}_3 & \boldsymbol{O}_3 \\ \boldsymbol{O}_3 & \boldsymbol{K}_2 & \boldsymbol{O}_3 \\ \boldsymbol{O}_3 & \boldsymbol{O}_3 & \boldsymbol{K}_3 \end{bmatrix}, \quad \bar{\boldsymbol{K}}_D^{-1} = \begin{bmatrix} \boldsymbol{K}_{D,1}^{-1} & \boldsymbol{O}_3 & \boldsymbol{O}_3 \\ \boldsymbol{O}_3 & \boldsymbol{K}_{D,2}^{-1} & \boldsymbol{O}_3 \\ \boldsymbol{O}_3 & \boldsymbol{O}_3 & \boldsymbol{K}_{D,3}^{-1} \end{bmatrix}, \quad \bar{\boldsymbol{K}}_S = \begin{bmatrix} \boldsymbol{K}_{S,1} & \boldsymbol{O}_3 & \boldsymbol{O}_3 \\ \boldsymbol{O}_3 & \boldsymbol{K}_{S,2} & \boldsymbol{O}_3 \\ \boldsymbol{O}_3 & \boldsymbol{O}_3 & \boldsymbol{K}_{S,3} \end{bmatrix} \tag{18}$$

Case study with 4 **robots**

Consider a team of $N = 4$ manipulators grasping an object. The internal forces estimation vector $\hat{\boldsymbol{f}}_I^i$ is defined as following:

$$\hat{\boldsymbol{f}}_I = \begin{bmatrix} \hat{\boldsymbol{f}}_{I_1} \\ \hat{\boldsymbol{f}}_{I_2} \\ \hat{\boldsymbol{f}}_{I_3} \\ \hat{\boldsymbol{f}}_{I_4} \end{bmatrix} = \boldsymbol{V}\boldsymbol{V}^\dagger \begin{bmatrix} \hat{\boldsymbol{f}}_1 \\ \hat{\boldsymbol{f}}_2 \\ \hat{\boldsymbol{f}}_3 \\ \hat{\boldsymbol{f}}_4 \end{bmatrix} = \begin{bmatrix} \frac{3}{4}\boldsymbol{f}_1 + \frac{1}{4}\hat{\boldsymbol{f}}_2 + \frac{1}{4}\hat{\boldsymbol{f}}_3 + \frac{1}{4}\hat{\boldsymbol{f}}_4 \\ \frac{1}{4}\hat{\boldsymbol{f}}_1 + \frac{3}{4}\boldsymbol{f}_2 - \frac{1}{4}\hat{\boldsymbol{f}}_3 - \frac{1}{4}\hat{\boldsymbol{f}}_4 \\ \frac{1}{4}\hat{\boldsymbol{f}}_1 - \frac{1}{4}\hat{\boldsymbol{f}}_2 + \frac{3}{4}\boldsymbol{f}_3 - \frac{1}{4}\hat{\boldsymbol{f}}_4 \\ \frac{1}{4}\hat{\boldsymbol{f}}_1 - \frac{1}{4}\hat{\boldsymbol{f}}_2 - \frac{1}{4}\hat{\boldsymbol{f}}_3 + \frac{3}{4}\boldsymbol{f}_4 \end{bmatrix} \tag{19}$$

Adopting the same considerations previously made, the overall system dynamics is

$$\begin{aligned}
\dot{\boldsymbol{e}}_1 + \left(\boldsymbol{K}_1 - \frac{3}{4}\boldsymbol{K}_D^{-1}\boldsymbol{K}_{S_1}\right)\boldsymbol{e}_1 &= \frac{3}{4}\boldsymbol{K}_D^{-1}\boldsymbol{K}_{S_1}(\boldsymbol{p}_{d,1} - \boldsymbol{p}_{eq,1}) + \frac{1}{4}\boldsymbol{K}_D^{-1}\left(\hat{\boldsymbol{f}}_2 + \hat{\boldsymbol{f}}_3 + \hat{\boldsymbol{f}}_4\right) \\
\dot{\boldsymbol{e}}_2 + \left(\boldsymbol{K}_2 - \frac{3}{4}\boldsymbol{K}_D^{-1}\boldsymbol{K}_{S_2}\right)\boldsymbol{e}_2 &= \frac{3}{4}\boldsymbol{K}_D^{-1}\boldsymbol{K}_{S_2}(\boldsymbol{p}_{d,2} - \boldsymbol{p}_{eq,2}) + \frac{1}{4}\boldsymbol{K}_D^{-1}\left(\hat{\boldsymbol{f}}_1 - \hat{\boldsymbol{f}}_3 - \hat{\boldsymbol{f}}_4\right) \\
\dot{\boldsymbol{e}}_3 + \left(\boldsymbol{K}_3 - \frac{3}{4}\boldsymbol{K}_D^{-1}\boldsymbol{K}_{S_3}\right)\boldsymbol{e}_3 &= \frac{3}{4}\boldsymbol{K}_D^{-1}\boldsymbol{K}_{S_3}(\boldsymbol{p}_{d,3} - \boldsymbol{p}_{eq,3}) + \frac{1}{4}\boldsymbol{K}_D^{-1}\left(\hat{\boldsymbol{f}}_1 - \hat{\boldsymbol{f}}_2 - \hat{\boldsymbol{f}}_4\right) \\
\dot{\boldsymbol{e}}_4 + \left(\boldsymbol{K}_4 - \frac{3}{4}\boldsymbol{K}_D^{-1}\boldsymbol{K}_{S_4}\right)\boldsymbol{e}_4 &= \frac{3}{4}\boldsymbol{K}_D^{-1}\boldsymbol{K}_{S_4}(\boldsymbol{p}_{d,4} - \boldsymbol{p}_{eq,4}) + \frac{1}{4}\boldsymbol{K}_D^{-1}\left(\hat{\boldsymbol{f}}_1 - \hat{\boldsymbol{f}}_2 - \hat{\boldsymbol{f}}_3\right),
\end{aligned} \tag{20}$$

whose matrix formulation is:

$$\dot{\boldsymbol{e}} + \left(\boldsymbol{K} - \frac{3}{4}\bar{\boldsymbol{K}}_D^{-1}\boldsymbol{K}_S\right)\boldsymbol{e} = \frac{3}{4}\bar{\boldsymbol{K}}_D^{-1}\boldsymbol{K}_S(\boldsymbol{p}_d - \boldsymbol{p}_{eq}) + \bar{\boldsymbol{K}}_D^{-1}\left(\boldsymbol{V}_p\boldsymbol{V}_p^\dagger - diag(\boldsymbol{V}_p\boldsymbol{V}_p^\dagger)\right)\hat{\boldsymbol{f}}_I, \tag{21}$$

with $\boldsymbol{K}$, $\bar{\boldsymbol{K}}_D^{-1}$ and $\bar{\boldsymbol{K}}_S$ defined in a similar way of (18).

A7 - Proof of Theorem 5

Let $V(\boldsymbol{e}, \dot{\boldsymbol{e}})$ be a candidate Lyapunov function

$$V = \frac{1}{2}\dot{\boldsymbol{e}}^{\mathrm{T}}\boldsymbol{M}\dot{\boldsymbol{e}} + \alpha \boldsymbol{e}^{\mathrm{T}}\boldsymbol{M}\dot{\boldsymbol{e}} + \frac{1}{2}\alpha \boldsymbol{e}^{\mathrm{T}}\boldsymbol{K}_D\boldsymbol{e} + \frac{1}{2}\boldsymbol{e}^{\mathrm{T}}\boldsymbol{K}_P\boldsymbol{e}. \tag{22}$$

Firstly, let us prove that exist two class $\mathcal{K}_\infty$ functions $\beta_1(\boldsymbol{\zeta})$ and $\beta_2(\boldsymbol{\zeta})$[1], such that

$$\beta_1(\boldsymbol{\zeta}) \leq V(\boldsymbol{\zeta}) \leq \beta_2(\boldsymbol{\zeta}).$$

Let us derive the lower bound β_1 as:

$$\begin{aligned} V \geq \frac{1}{2}\lambda_m(\boldsymbol{M}) \left\|\dot{\boldsymbol{e}}\right\|^2 - \alpha\lambda_M(\boldsymbol{M}) \left\|\dot{\boldsymbol{e}}\right\| \left\|\boldsymbol{e}\right\| + \frac{1}{2}\alpha\lambda_m(\boldsymbol{K}_D) \left\|\boldsymbol{e}\right\|^2 + \frac{1}{2}\lambda_m(\boldsymbol{K}_P) \left\|\boldsymbol{e}\right\|^2 = \\ = \begin{bmatrix} \left\|\boldsymbol{e}\right\| & \left\|\dot{\boldsymbol{e}}\right\| \end{bmatrix} \boldsymbol{\Omega} \begin{bmatrix} \left\|\boldsymbol{e}\right\| \\ \left\|\dot{\boldsymbol{e}}\right\| \end{bmatrix} = \beta_1(\boldsymbol{\zeta}), \end{aligned} \tag{24}$$

where the matrix

$$\boldsymbol{\Omega} = \begin{bmatrix} \frac{1}{2}\left[\alpha\lambda_m(\boldsymbol{K}_D) + \lambda_m(\boldsymbol{K}_P)\right] & -\frac{1}{2}\alpha\lambda_M(\boldsymbol{M}) \\ -\frac{1}{2}\alpha\lambda_M(\boldsymbol{M}) & \frac{1}{2}\alpha\lambda_m(\boldsymbol{M}) \end{bmatrix}$$

is positive definite if the following condition holds:

$$\lambda_m(\boldsymbol{K}_D) > \alpha\frac{\lambda_M(\boldsymbol{M})^2}{\lambda_m(\boldsymbol{M})^2} - \frac{1}{\alpha}\lambda_m(\boldsymbol{K}_p). \tag{25}$$

The bound function β_2 can be straightforwardly derived as

$$\beta_2(\boldsymbol{\zeta}) = \frac{1}{2}\left\|\boldsymbol{\zeta}\right\|^2 \lambda_M(\boldsymbol{M}) + \alpha\left\|\boldsymbol{\zeta}\right\|^2 \lambda_M(\boldsymbol{M}) + \frac{1}{2}\left\|\boldsymbol{\zeta}\right\|^2 \lambda_M(\boldsymbol{K}_D) + \frac{1}{2}\left\|\boldsymbol{\zeta}\right\|^2 \lambda_M(\boldsymbol{K}_P).$$

[1] According to the Theorem 4.19 of [154],

$$\left\|\boldsymbol{\zeta}(t)\right\| \leq \beta(\left\|\boldsymbol{\zeta}(t_0)\right\|, t - t_0) + \gamma(\sup_{\tau \geq t_0} \left\|u(\tau)\right\|), \forall t \geq t_0, \tag{23}$$

where $\beta(\cdot)$ depends on the initial condition $\left\|\boldsymbol{\zeta}(t_0)\right\|$, while γ is a composition of the functions $\beta_1(\boldsymbol{\zeta})$ and $\beta_2(\boldsymbol{\zeta})$ and u represents the input of the system.

Secondly, by taking the time derivative of $V(\boldsymbol{e}, \dot{\boldsymbol{e}})$, it yields

$$\dot{V} = \dot{\boldsymbol{e}}^{\mathrm{T}} \boldsymbol{M} \ddot{\boldsymbol{e}} + \alpha \boldsymbol{e}^{\mathrm{T}} \boldsymbol{M} \ddot{\boldsymbol{e}} + \alpha \dot{\boldsymbol{e}}^{\mathrm{T}} \boldsymbol{M} \dot{\boldsymbol{e}} + \alpha \boldsymbol{e}^{\mathrm{T}} \boldsymbol{K}_D \dot{\boldsymbol{e}} + \boldsymbol{e}^{\mathrm{T}} \boldsymbol{K}_P \dot{\boldsymbol{e}}. \tag{26}$$

Then, by substituting the dynamics in (2.82),

$$\begin{aligned} \dot{V} = \dot{\boldsymbol{e}}^{\mathrm{T}} \left[\boldsymbol{d} - \boldsymbol{K}_D \dot{\boldsymbol{e}} - (\boldsymbol{K}_P - \bar{\boldsymbol{K}}_P) \boldsymbol{e} \right] + \alpha \boldsymbol{e}^{\mathrm{T}} \left[\boldsymbol{d} - \boldsymbol{K}_D \dot{\boldsymbol{e}} - (\boldsymbol{K}_P - \bar{\boldsymbol{K}}_P) \boldsymbol{e} \right] + \\ + \alpha \dot{\boldsymbol{e}}^{\mathrm{T}} \boldsymbol{M} \dot{\boldsymbol{e}} + \alpha \boldsymbol{e}^{\mathrm{T}} \boldsymbol{K}_D \dot{\boldsymbol{e}} + \boldsymbol{e}^{\mathrm{T}} \boldsymbol{K}_P \dot{\boldsymbol{e}}, \end{aligned} \tag{27}$$

which can be upper-bounded as follows

$$\begin{aligned} \dot{V} \leq \|\dot{\boldsymbol{e}}\|^2 \left(\alpha \lambda_M(\boldsymbol{M}) - \lambda_m(\boldsymbol{K}_D) \right) + \left\| \bar{\boldsymbol{K}}_p \right\| \|\boldsymbol{e}\| \|\dot{\boldsymbol{e}}\| + \\ + \alpha \left(\left\| \bar{\boldsymbol{K}}_p \right\| - \lambda_m(\boldsymbol{K}_P) \right) \|\boldsymbol{e}\|^2 + \|\dot{\boldsymbol{e}} + \alpha \boldsymbol{e}\| \|\boldsymbol{d}\|. \end{aligned} \tag{28}$$

This latter upper bound can be reformulated in a more compact form

$$\dot{V} \leq - \begin{bmatrix} \|\boldsymbol{e}\| & \|\dot{\boldsymbol{e}}\| \end{bmatrix} \boldsymbol{\Sigma} \begin{bmatrix} \|\boldsymbol{e}\| \\ \|\dot{\boldsymbol{e}}\| \end{bmatrix} + \left[\begin{bmatrix} \|\boldsymbol{e}\| & \|\dot{\boldsymbol{e}}\| \end{bmatrix} \begin{bmatrix} \alpha \\ 1 \end{bmatrix} \right] \|\boldsymbol{d}\|, \tag{29}$$

where the matrix

$$\boldsymbol{\Sigma} = \begin{bmatrix} \alpha \left(\lambda_m(\boldsymbol{K}_P) - \left\| \bar{\boldsymbol{K}}_P \right\| \right) & -\frac{1}{2} \left\| \bar{\boldsymbol{K}}_P \right\| \\ -\frac{1}{2} \left\| \bar{\boldsymbol{K}}_P \right\| & \lambda_m(\boldsymbol{K}_D) - \alpha \lambda_M(\boldsymbol{M}) \end{bmatrix},$$

is positive definite if the conditions **C.1** and **C.2** hold. Then, the inequality (28) can be further rewritten as

$$\dot{V} \leq -\lambda_m(\boldsymbol{\Sigma}) \|\boldsymbol{\zeta}\|^2 + \bar{\alpha} \|\boldsymbol{\zeta}\| \|\boldsymbol{d}\|, \tag{30}$$

with $\bar{\alpha} = \sqrt{\alpha^2 + 1}$. The bound on the right-hand term of (30) is negative definite $\forall \|\boldsymbol{\zeta}\| \geq \dfrac{\bar{\alpha} \|\boldsymbol{d}\|}{\lambda_m(\boldsymbol{\Sigma})}$. Finally, according to the Theorem 4.19 in [154], conditions (24) and (30) prove that the system (2.83) is ISS.

A8 - Proof of Theorem 6

Let $V(\boldsymbol{e}_o, \dot{\boldsymbol{e}}_o)$ be a candidate Lyapunov function

$$V = \frac{1}{2}\dot{\boldsymbol{e}}_o^{\mathrm{T}}\bar{\boldsymbol{M}}_o\dot{\boldsymbol{e}}_o + \alpha\boldsymbol{e}_o^{\mathrm{T}}\bar{\boldsymbol{M}}_o\dot{\boldsymbol{e}}_o + \frac{1}{2}\alpha\boldsymbol{e}_o^{\mathrm{T}}\boldsymbol{D}_o\boldsymbol{e}_o + \frac{1}{2}\boldsymbol{e}_o^{\mathrm{T}}\boldsymbol{K}_o\boldsymbol{e}_o. \tag{31}$$

Firstly, let us prove that there exist two class $\mathcal{K}_\infty$ functions $\beta_1(\boldsymbol{\zeta})$ and $\beta_2(\boldsymbol{\zeta})$, such that

$$\beta_1(\boldsymbol{\zeta}) \leq V(\boldsymbol{\zeta}) \leq \beta_2(\boldsymbol{\zeta}). \tag{32}$$

Let us derive the lower bound β_1 as:

$$V \geq \frac{1}{2}\lambda_m(\bar{\boldsymbol{M}}_o)\left\|\dot{\boldsymbol{e}}_o\right\|^2 - \alpha\lambda_M(\bar{\boldsymbol{M}}_o)\left\|\dot{\boldsymbol{e}}_o\right\|\left\|\boldsymbol{e}_o\right\| + \frac{1}{2}\alpha\lambda_m(\boldsymbol{D}_o)\left\|\boldsymbol{e}_o\right\|^2 + \frac{1}{2}\lambda_m(\boldsymbol{K}_o)\left\|\boldsymbol{e}_o\right\|^2 = \tag{33}$$

$$= \begin{bmatrix}\left\|\boldsymbol{e}_o\right\| & \left\|\dot{\boldsymbol{e}}_o\right\|\end{bmatrix}\boldsymbol{\Omega}\begin{bmatrix}\left\|\boldsymbol{e}_o\right\| \\ \left\|\dot{\boldsymbol{e}}_o\right\|\end{bmatrix} = \beta_1(\boldsymbol{\zeta}),$$

where the matrix

$$\boldsymbol{\Omega} = \begin{bmatrix}\frac{1}{2}\left[\alpha\lambda_m(\boldsymbol{D}_o) + \lambda_m(\boldsymbol{K}_o)\right] & -\frac{1}{2}\alpha\lambda_M(\bar{\boldsymbol{M}}_o) \\ -\frac{1}{2}\alpha\lambda_M(\bar{\boldsymbol{M}}_o) & \frac{1}{2}\alpha\lambda_m(\bar{\boldsymbol{M}}_o)\end{bmatrix}$$

is positive definite if the following condition is satisfied:

$$\lambda_m(\boldsymbol{D}_o) > \alpha\frac{\lambda_M(\bar{\boldsymbol{M}}_o)^2}{\lambda_m(\bar{\boldsymbol{M}}_o)} - \frac{\lambda_m(\boldsymbol{K}_o)}{\alpha}. \tag{34}$$

The bound function β_2 can be straightforwardly derived as

$$\beta_2(\boldsymbol{\zeta}) = \frac{1}{2}\lambda_M(\bar{\boldsymbol{M}}_o)\left\|\boldsymbol{\zeta}\right\|^2 + \alpha\lambda_M(\bar{\boldsymbol{M}}_o)\left\|\boldsymbol{\zeta}\right\|^2 + \frac{1}{2}\alpha\lambda_M(\boldsymbol{D}_o)\left\|\boldsymbol{\zeta}\right\|^2 + \frac{1}{2}\lambda_M(\boldsymbol{K}_o)\left\|\boldsymbol{\zeta}\right\|^2.$$

By taking the time derivative of $V(\boldsymbol{e}_o, \dot{\boldsymbol{e}}_o)$, it yields

$$\dot{V} = \left(\dot{\boldsymbol{e}}_o^{\mathrm{T}} + \alpha\boldsymbol{e}_o^{\mathrm{T}}\right)\bar{\boldsymbol{M}}_o\ddot{\boldsymbol{e}}_o + \alpha\dot{\boldsymbol{e}}_o^{\mathrm{T}}\bar{\boldsymbol{M}}_o\dot{\boldsymbol{e}}_o + \alpha\boldsymbol{e}_o^{\mathrm{T}}\boldsymbol{D}_o\dot{\boldsymbol{e}}_o + \boldsymbol{e}_o^{\mathrm{T}}\boldsymbol{K}_o\dot{\boldsymbol{e}}_o. \tag{35}$$

Then, substituting the dynamics in (2.96),

$$\dot{V} = \left(\dot{\boldsymbol{e}}_o^{\mathrm{T}} + \alpha \boldsymbol{e}_o^{\mathrm{T}}\right) \boldsymbol{d} - \dot{\boldsymbol{e}}_o^{\mathrm{T}} \boldsymbol{D}_o \dot{\boldsymbol{e}}_o - \dot{\boldsymbol{e}}_o^{\mathrm{T}} \boldsymbol{K}_{env} \boldsymbol{e}_o - \alpha \boldsymbol{e}_o^{\mathrm{T}} \left(\boldsymbol{K}_o + \boldsymbol{K}_{env}\right) \boldsymbol{e}_o + \alpha \dot{\boldsymbol{e}}_o^{\mathrm{T}} \bar{\boldsymbol{M}}_o \dot{\boldsymbol{e}}_o, \quad (36)$$

which can be upper-bounded as follows

$$\dot{V} \leq \left[\alpha \lambda_M(\bar{\boldsymbol{M}}_o) - \lambda_m(\boldsymbol{D}_o)\right] \|\dot{\boldsymbol{e}}_o\|^2 + \|\boldsymbol{K}_{env}\| \, \|\boldsymbol{e}_o\| \, \|\dot{\boldsymbol{e}}_o\| + \\ + \alpha \left[\|\boldsymbol{K}_{env}\| - \lambda_m(\boldsymbol{K}_o)\right] \|\boldsymbol{e}_o\|^2 + \|\dot{\boldsymbol{e}}_o + \alpha \boldsymbol{e}_o\| \, \bar{d}, \quad (37)$$

and reformulated in a more compact form

$$\dot{V} \leq - \begin{bmatrix} \|\boldsymbol{e}_o\| & \|\dot{\boldsymbol{e}}_o\| \end{bmatrix} \boldsymbol{\Sigma} \begin{bmatrix} \|\boldsymbol{e}_o\| \\ \|\dot{\boldsymbol{e}}_o\| \end{bmatrix} + \begin{bmatrix} \|\boldsymbol{e}_o\| & \|\dot{\boldsymbol{e}}\| \end{bmatrix} \begin{bmatrix} \alpha \\ 1 \end{bmatrix} \bar{d}, \quad (38)$$

where the matrix

$$\boldsymbol{\Sigma} = \begin{bmatrix} \alpha \left(\lambda_m(\boldsymbol{K}_o) - \left\|\bar{\boldsymbol{K}}_{env}\right\|\right) & -\frac{1}{2} \left\|\bar{\boldsymbol{K}}_{env}\right\| \\ -\frac{1}{2} \left\|\bar{\boldsymbol{K}}_{env}\right\| & \lambda_m(\boldsymbol{D}_o) - \alpha \lambda_M(\bar{\boldsymbol{M}}_o) \end{bmatrix}$$

is positive definite if the following condition holds:

$$\lambda_m(\boldsymbol{D}_o) > \alpha \lambda_M(\bar{\boldsymbol{M}}_o) - \frac{1}{4\alpha} \frac{\|\boldsymbol{K}_{env}\|^2}{\lambda_m(\boldsymbol{K}_o) - \|\boldsymbol{K}_{env}\|}.$$

The inequality (37) can be further rewritten as

$$\dot{V} \leq -\lambda_m(\boldsymbol{\Sigma}) \|\boldsymbol{\zeta}\|^2 + \bar{\alpha} \|\boldsymbol{\zeta}\| \, \bar{d}, \quad (39)$$

with $\bar{\alpha} = \sqrt{\alpha^2 + 1}$. The bound on the right-hand term of (39) is negative definite $\forall \|\boldsymbol{\zeta}\| \geq \dfrac{\bar{\alpha} \bar{d}}{\lambda_m(\boldsymbol{\Sigma})}$.

Finally, according to the Theorem 4.19 in [154], conditions (32) and (39) prove that the system (2.99) is ISS.