Università degli Studi di Salerno

Dottorato di Ricerca in Informatica e Ingegneria dell'Informazione
Ciclo 33 – a.a 2019/2020

TESI DI DOTTORATO / PH.D. THESIS

Human multi-robot interaction: from workspace sharing to physical collaboration

MARTINA LIPPI

SUPERVISOR: **PROF. PASQUALE CHIACCHIO**

PHD PROGRAM DIRECTOR: **PROF. PASQUALE CHIACCHIO**

Dipartimento di Ingegneria dell'Informazione ed Elettrica
e Matematica Applicata
Dipartimento di Informatica

Contents

- Ai miei nonni,
per il loro esempio continuo e l'amore incondizionato -

Chapter 1

Introduction

1.1 Overview and motivation

A drastic change in the vision of future industries is taking place nowadays with the strategic initiative of Industry 4.0 [1]: from traditional mass production based on assembly lines (low variety of high-volume products) to mass customization based on product personalization (high variety of low-volume products). The concept of *smart factories* is the driving factor of Industry 4.0 which envisages factories as integrated systems where robots, sensors, devices and people are "intelligent" entities constantly interconnected along the whole value chain, cooperating with each other and sharing real-time data in order to rapidly adapt to new or personalized requests and predict failures. However, although devices and machines are equipped with artificial intelligence, the human reasoning and adaptation skills are still far superior to it. It inevitably follows that the role of human operators is crucial in this transformation in order to fulfill the need for flexibility of the production environments. Rather than being replaced by devices, they play the central role of "strategic decision-makers and flexible problem solvers" [1] in smart factories, leading to a human-centric concept of industry [2].

In this scenario, removing the barriers between humans and robots, enabling their collaboration, as represented in Figure 1.1

(image from [3]), is one of the fundamental steps to realize the required agile production. More in detail, these two entities exhibit

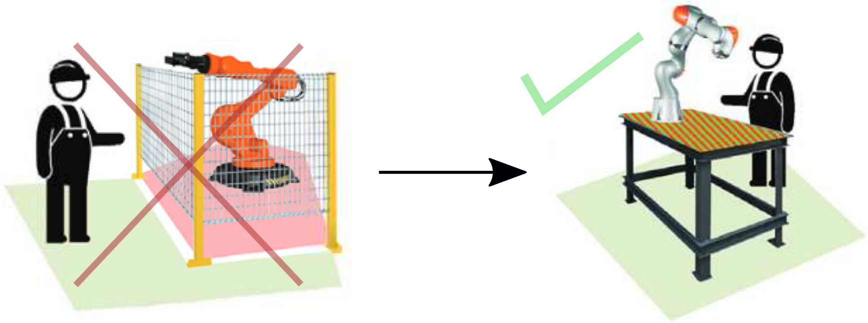
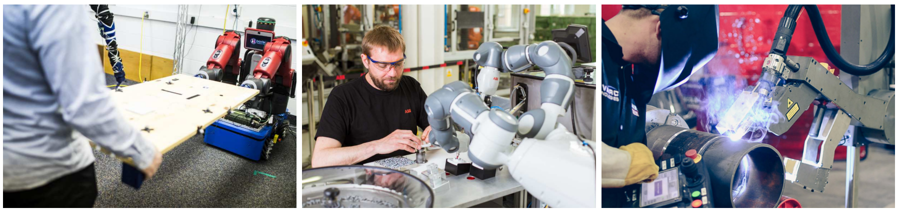


Figure 1.1: Industrial transformation taking place according to the principles of Industry 4.0: from caged (on the left) to collaborative (on the right) robots.

complementary skills whose fusion can potentially lead to increase the production efficiency, quality and flexibility [4, 5]: on the one hand, humans have higher abilities for environment interpretation and decision-making process, are characterized by greater manipulation skills and are more flexible in the sense of simplicity to be re-tasked; on the other hand, robots are faster, stronger, more precise and better suited to repetitive and/or heavy tasks than humans. In addition, the physical abilities as well as the robustness to faults of the robotic component are significantly enhanced when *multiple* cooperative robots are introduced into industrial setups instead of individual robots, enabling the execution of tasks that would otherwise not be possible.

It is evident that the advantages arising from the combination of human and robotic skills are not limited to industrial settings but they have a dramatic impact also in numerous other scenarios where robots can be used to assist people and to reduce their stress and fatigue, as for example, in service robotics [6] and search-and-rescue scenarios [7]. A wide range of applications can thus benefit from Human-Robot Collaboration (HRC), only some of which are:

- transport of loads, where one or more robots exert most of the required effort and the human “guides” them, i.e., im-



a) b) c)

Figure 1.2: Examples of HRC: cooperative transport of loads (a), assembly (b) and welding (c).

poses the object trajectory, through unstructured environments, as depicted in Figure 1.2.a (image from [8]);

- assembly, where the robots exploit their built-in high accuracy for the assembly of components and the human exploits his/her major flexibility for the insertion of these components in more variable and complex objects, as represented in Figure 1.2.b (image from [9]);
- packing and palletizing, where the robots are in charge of the "logistic task", i.e., of collecting and carrying loads, while the human is in charge of the last refining and packing operations;
- welding, where the human operator executes the positioning task and the robot the welding one, as shown in the Figure 1.2.c (image by Novarc Technologies[1]), thus allowing to perform welding on different objects while preserving robot accuracy;
- any complex and unstructured task where the presence of human teammates allows to deal with such complexity and non-structurality by guiding or correcting the robots' behavior.

Note that also in the case the collaboration only includes the coexistence of human and robots in the same workspace while per-

[1] `https://www.novarctech.com/`

forming independent tasks, immediate advantages follow in terms of productivity, since there is no more need to turn off robots when humans enter their workspace (i.e., they can still accomplish the predefined tasks as long as it is compatible with the human safety requirements), reconfigurability of the work-cell and reduction of costs for protective barriers and footprint area; the latter has a drastic impact especially in the case of mobile robots for which current solutions are restricted to predefined paths (e.g., tracks on the floor) and humans are not allowed to enter the navigation zone. Furthermore, note that HRC is also encouraged by recent standards that provide for regularization in industrial settings. Examples are ISO 10218-1 [10], ISO 10218-2 [11] and the technical specification document ISO/TS 15066 [12]. The latter specifies the guidelines for four types of human-robot collaboration, that are:

1. Safety-rated monitored stop, i.e., robots are required to stop when humans enter the working area;

2. Hand guiding, i.e., robots can only move through direct human input;

3. Speed and separation monitoring, i.e., robots are required to keep a minimum safety distance from operators;

4. Power and force limiting, i.e., robots are required to guarantee limited power and interaction force in the case of physical collaboration with the human.

Finally, a further form of human-robot teamwork emerges when humans, rather than working side-by-side in the same work space of the robots, provide examples to the latter for the accomplishment of a given task and the robotic system learns from these examples. This form of collaboration is typically referred to as *learning from demonstration* [13]. The strength of this collaboration lies in the fact that it enables to bypass the explicit planning of the robots motion according to the system state, by imitating the human behavior. This feature is particularly relevant when the system state cannot be easily determined analytically, such as for the manipulation of deformable objects [14], causing the planning to achieve desired configurations to be extremely challenging.

Motivated by the above scenarios, the present thesis aims to investigate human multi-robot interaction from multiple perspectives. Indeed, even if, as highlighted so far, there are innumerable advantages in the use of multiple collaborative robots and standards exist that regularize their use also in industrial settings, the actual integration of such robots into real systems is far from trivial and several aspects need to be taken into account.

In particular, for an effective collaboration, it is first and foremost necessary to devise methodologies that guarantee the *safety* of the person at all times, despite his/her highly dynamic and unpredictable behavior and regardless of the robotic task. According to [15], human safety is hereby referred to the physical safety, i.e., no unwanted contact between humans and robots and no human injury must occur. To this aim, as an example, strategies for human avoidance or for limitation of exchange forces might need to be designed depending on the scenario. Note that the requirement for human safety also implies the necessity of robots' *reliability*, i.e., in order not to compromise the safety of the person, not only strategies to ensure it must be defined, but their correct execution also needs to be guaranteed despite possible faults of the robotic system. It follows that robot control must exhibit properties of (i) convergence to desired and safe states (when no faults occur) as well as (ii) robustness to faults which need to be detected, isolated and accommodated.

Next, it is necessary to define proper control algorithms which respond to the task-dependent desired human-robot collaboration, e.g., it may be desired that the human operator is physically assisted by robots or that he/she can intervene in any moment to correct their behavior.

Finally, in the case of human demonstrations, it is necessary to endow the robots with the ability to generalize from the provided examples in order to successfully accomplish the desired task.

The objective of the thesis is to cover all the aforementioned aspects starting from the reliability of the multi-robot system and the human safety in shared environments up to scenarios of physical interaction and learning from demonstrations. In the following, possible scenarios of human-robot interaction are defined and it is detailed which of them the thesis work contributes to.

1.1.1 Human-robot interaction scenarios

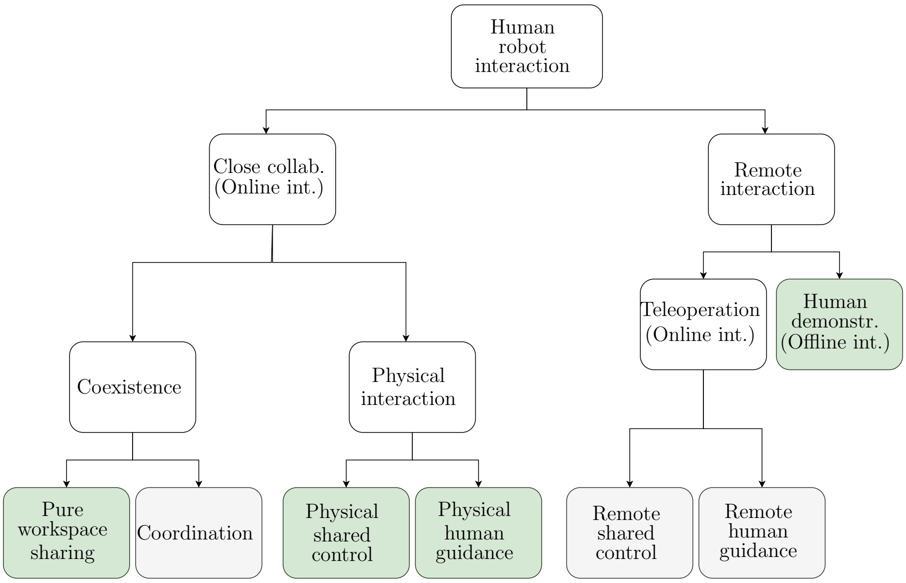


Figure 1.3: Overview of possible HRI scenarios. The blocks addressed in the thesis are highlighted in green.

Figure 1.3 summarizes a possible taxonomy for Human Robot Interaction (HRI) scenarios and highlights in green the ones addressed in the thesis. At the first level, the interaction is divided into *close* collaboration and *remote* interaction depending on whether humans and robots work close to each other or not, respectively. Moving down through the branch of close collaboration, it is assumed that the interaction always takes place *online*, i.e., humans and robots operate simultaneously, as the human presence would otherwise become invisible to robots, and a subdivision is made that regards the presence of physical contact, i.e., of an exchange of forces, between robots and human operators. More specifically, *coexistence* refers to the case of operation in the same environment but without exchange of forces while *physical interaction* refers to the case in which such exchange occurs, thus implying the need to control it in order to prevent harm to the person. Note that this exchange of forces can occur directly or through the manipulation of a common object.

In the case of coexistence two possible scenarios are then prospected that depend on how much the tasks of robots and humans are intertwined: *pure workspace sharing* when they carry out independent tasks and only share the same workspace and *coordination* when they closely collaborate to pursue the task, distributing, for example, the operations necessary for the processing of interest. For pure workspace sharing scenarios, the HRI strategies translate into avoidance strategies which allow the robots to replan their motion in real-time in order to prevent unsafe contacts with the personnel, e.g., [16, 17]. For coordination scenarios, the HRI strategies translate into task allocation and scheduling strategies in which the humans play the role of partially controllable and non deterministic agents, e.g., [18]. Note that a low-level safety layer also needs to be always included in order to prevent robots from causing harm to humans during coordination.

In the case of physical interaction, a sub-division is made on the level of robots' autonomy. In detail, in a *physical human guidance* scenario a leader-follower paradigm is adopted in which the human acts as leader and the robot only plays a follower role aimed at minimizing the leader effort. Therefore, the respective control strategies aim to impose a highly compliant behavior to the robots which enables the human to drive them. On the contrary, the *physical shared control* scenario [19] finds inspiration from human-human interaction scenarios where the two entities act as peers. In the shared control, the robot autonomy is thus preserved to a certain extent and a more equal distribution of roles is defined for the robotic and human counterparts.

Stepping back to the remote interaction scenario, a distinction is made on whether the interaction is online or not: in the online case, a *teleoperation* scenario [20] is traditionally envisaged where humans use ad-hoc interfaces to remotely and actively control the robotic component, also referred to as teleoperator, and/or intervene during its activity; in the offline case, the human plays instead the role of provider of *demonstrations* to achieve a desired task. Starting from these examples, the robots then need to learn how to autonomously carry out the task. Focusing on the teleoperation, a multitude of control devices and architectures exist, as reported in the survey [21], which may involve or not the force feedback to the human operator. However, regardless of the technological details, it is possible to classify the teleoperation approaches into two main macro-categories depending on the level of robots' autonomy: remote human guidance and shared control. In particular, in the former case, the human operator fully determines the robots behavior, e.g., speed or position, by providing the inputs to their actuators (through the specific interface). Only a low-level control loop is then generally present on the teleoperator side to track the human reference. In contrast, in the latter case, the robotic component always preserves a level of autonomy [22] in order, for example, to avoid obstacles in the environment and to generally carry out part of the desired task.

Despite the high practical relevance of teleoperation and coordination scenarios, the thesis will not dwell on them but will focus, as emphasized in green in Figure 1.3, on pure workspace sharing as well as physical interaction and offline human demonstrations scenarios. Note that although other possible taxonomies exist in literature, as for example [23, 24], the one in Figure 1.3 has been proposed to the aim of outlining the scenarios of interest in the thesis, ignoring any details not relevant for it.

1.1.2 Open questions

The remarkable potentialities of HRI, briefly outlined in Section 1.1, have obviously led to a strong interest of the scientific community in recent years towards this topic. However, further progress still needs to be made before we can achieve a concrete and systematic integration of these technologies in concrete environments. Moreover, the current state-of-the-art is lacking in analyzing cases of human *multi*-robot interaction although this can be necessary for achieving tasks requiring high physical capabilities or have demanding time or robustness requirements.

The thesis thus aims to provide progress steps towards the answer to the following questions:

1. *How to combine the collaboration with the human and the coordination among different robots? How do these aspects influence and limit each other?*

   In both physical and non-physical collaboration scenarios, the control strategy should address human-robot interaction aspects, e.g., human safety or physical assistance to the operator, but also manage the cooperation of the different robots fulfilling possible constraints that they may introduce, such as closed kinematic chain constraints in the case of cooperative manipulation tasks.

2. *How to distinguish if a contact is intentional or accidental regardless of the robotic task in case the human is allowed to work in proximity of the robots?*

   When physical interaction is allowed, accidental contact may also occur beyond intentional contact. The robots must therefore be able to detect and recognize it whatever the task being performed, which may also require exchange of forces with the environment.

3. *How to ensure a reliable low-level control system which can be distributed over different robots?*

   In order to ensure human safety, the robotic system must first of all be reliable meaning that it must be able to detect any possible fault. Guaranteeing this property becomes more challenging in the case of distributed architecture, i.e., in case there is no central control unit that monitors the state of all robots and coordinates them.

4. *How to carry out complex manipulation tasks and produce understandable visual plans from visual demonstrations?*

   In the case of complex manipulation tasks, traditional planning, requiring analytical modeling of the system, can be very challenging. The ability to learn from visual examples, provided by the person, allows to overcome this issue but requires defining methodologies to correctly reproduce the demonstrations and generalize from them.

Note that questions 2 and 4 do not specifically focus on multi-robot systems and, indeed, the respective solutions in the following address single robot scenarios. However, such solutions lay the foundations for tackling the same problems in multi-robot settings and can be generalized to the latter.

The remainder of the thesis is structured as follows. First, preliminaries for multi-robot control along with methodologies for the system reliability in terms of robustness to faults will be introduced in Chapter 2, addressing question 3, then scenarios of coexistence with pure workspace sharing will be addressed in Chapter 3 by focusing on safety issues and addressing question 1 for the non-physical collaboration case, next both shared control and human guidance scenarios during physical collaboration (question 1) will be covered in Chapter 4, where also the problem of distinguishing whether a contact is intentional or not will be considered (question 2), finally the problem of performing visual planning from demonstrations (question 4) will be tackled in Chapter 5 while conclusions and future work will be discussed in Chapter 6.

Chapter 2

Multi-robot systems and their robustness

This chapter aims to introduce the fundamentals for the control of multi-robot systems as well as to present a methodology for ensuring their robustness to faults. Indeed, in order to deploy a robotic system in the vicinity of human operator, it is necessary to guarantee that it is inherently safe and that, in case of faults, they are promptly detected and no damage to the personnel is made.

A robot is herein assumed to be a *manipulator*, i.e., a mechanical structure composed of rigid links interconnected by rotational or prismatic joints, that can be mounted on a mobile base. The robots are assumed to be *cooperative*, i.e., they all participate in the pursuit of a common task.

Two main architectures, as exemplified in Figure 2.1, can be possibly adopted for controlling the multi-robot system [25] which differ on the distribution of the decision making process: (*i*) centralized (on the left in the figure) and (*ii*) distributed (on the right). In the former case, a central control unit (depicted as a grey box) exists which communicates with all the robots (depicted as blue circles) of the system and establishes the actions to undertake in order to carry out the desired task, in the latter case the decision-making process is distributed among the robots and each robot autonomously defines its actions in order to accomplish the common task, only relying on *local* information retrieved from on-board sensors, e.g., encoders or vision systems, or explicitly exchanged with neighbouring robots.

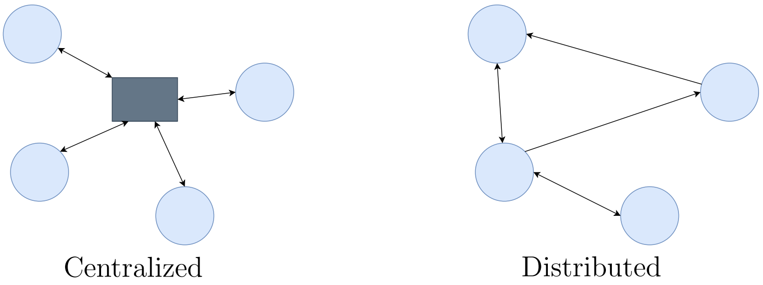


Figure 2.1: Example of centralized architecture (left), where a central control unit depicted as a grey box is present and communicates with all robots, and distributed architecture (right), where no need of central unit exists.

Centralized approaches are typically highly effective in terms of performance since they rely on a *global* knowledge of the system but the need of a central unit and/or communication among all robots of the system might be undesirable when robots, which can also be mobile, are employed in unstructured environments. Indeed, in this case, *physical* limitations and scalability issues may arise in having all the robots communicate with a single control unit. In addition, reliability problems may occur if the operational status of the central unit is compromised. For these reasons, in the last decade, many efforts have been devoted to the design of distributed solutions which can significantly improve the overall system in terms of flexibility, scalability and robustness to faults [25, 26] with respect to centralized ones. However, the lack of global information generally leads to reduced performance and more complex control design when these solutions are adopted. Therefore, the research community, and the present thesis, aim to devise distributed solutions that achieve the same performance as the centralized counterpart [27] while preserving their major flexibility and robustness.

In addition, note that, even if one of the main advantages of distributed robotic systems is identified in the increased robustness to failures of teammates, such a robustness is only *potential* and, on the contrary, if a failure is not properly handled, it can jeopardize the entire mission and the human safety. Thus, the thesis also aims to investigate distributed Fault Detection and Isolation (FDI) strategies which enable robustness to faults.

In the following, the notation adopted in the thesis is first reported. Then, the modeling of a single robot, from the kinematics and dynamics point of views, and the modeling of the information exchange between different robots are defined. Subsequently, a general formulation for the definition of cooperative tasks is introduced and finally a possible FDI strategy, introduced in [28] and [29], is presented.

2.1 Table of symbols

The main variables and notation adopted in the thesis are briefly reported in Table 2.1.

2.2 Robot modeling

The representation of the generic i th robot given by a manipulator which can be mounted on a mobile base is reported in Figure 2.2. Its overall model consists of a kinematic model and a dynamic model.

As far as the kinematic model is concerned, it defines the relationship between the joint configuration and the position and orientation of robot end effector, i.e., it expresses the relationship between the joint space and the operational space or Cartesian space. Let $\boldsymbol{q}_i \in \mathbb{R}^{n_i}$ be the joint position and $\boldsymbol{x}_i = \left[\boldsymbol{p}_i^T, \, \boldsymbol{\phi}_i^T\right]^T \in \mathbb{R}^p$ be the end effector configuration of the i th manipulator expressing position $\boldsymbol{p}_i$ and orientation $\boldsymbol{\phi}_i$ of the attached frame $\Sigma_{r,i}$ with respect to the world frame Σ_w. The robot kinematics is expressed

Variable	Description
γ, Γ	Scalar variables are denoted by lowercase or uppercase letters and in italic
$\boldsymbol{\gamma}$	Vectors are denoted by lowercase letters and boldface
$\boldsymbol{\Gamma}$	Matrices are denoted by uppercase letters and boldface
$\boldsymbol{v}_i$	Generic variable $\boldsymbol{v}$ relative to the i th robot
$\boldsymbol{v}$	Stacked vector collecting the variables $\boldsymbol{v}_i$ for all the robots $\boldsymbol{v} = \begin{bmatrix}\boldsymbol{v}_1^T & \boldsymbol{v}_2^T & \dots & \boldsymbol{v}_N^T\end{bmatrix}^T$
${}^{i}\hat{(\cdot)}$	Estimation of a *global* variable made by robot i, e.g., ${}^{i}\hat{\boldsymbol{x}}$ is the estimation made by robot i of the stacked vector $\boldsymbol{x}$
$\hat{(\cdot)}$	Estimation of a *local* variable, e.g., $\hat{(\boldsymbol{x}_i)}$ would be the estimation made by robot i of $\boldsymbol{x}_i$
N	Number of robots
$\boldsymbol{q}_i \in \mathbb{R}^{n_i}$	Joint configuration of the i th robot
$\boldsymbol{x}_i \in \mathbb{R}^{p}$	End effector configuration of the i th robot
$\boldsymbol{\pi}_i \in \mathbb{R}^{n_{\pi_i}}$	Dynamic parameters of the i th robot
$\boldsymbol{\sigma} \in \mathbb{R}^{m}$	Cooperative task function
$\mathcal{G}(\mathcal{E}, \mathcal{V})$	Communication graph with set of nodes $\mathcal{V}$ and set of edges $\mathcal{E}$
$\mathbf{0}_m(\mathbf{1}_m)$	Column vector in $\mathbb{R}^m$ with all zero (one) elements
$\boldsymbol{O}_{n\times m}$	Matrix in $\mathbb{R}^{n\times m}$ of all elements equal to 0
$\boldsymbol{O}_m\,(\boldsymbol{I}_m)$	Null (identity) matrix in $\mathbb{R}^{m\times m}$
$\Vert\cdot\Vert$	2-norm of $(\cdot)$

Table 2.1: Table of symbols

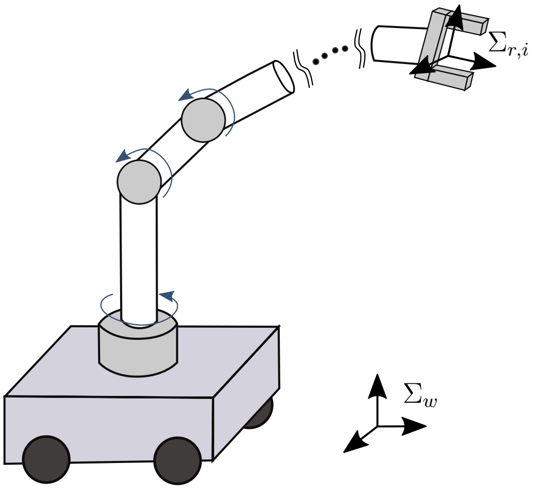


Figure 2.2: Representation of the generic i th robot given by a manipulator which can be mounted on a mobile base. The respective end effector reference frame $\Sigma_{r,i}$ and the world frame Σ_w are reported.

by

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

where $k_i : R^{n_i} \to R^p$ is the direct kinematics function of the i th robot. When the number of Degrees Of Freedom (DOFs) n_i is greater than the number of variables to be controlled, generally coincident with the operational space dimension p, the manipulator is defined redundant and the additional DOFs can be used to meet secondary objectives [30]. By differentiating (2.1), the differential kinematics is obtained

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

where $\boldsymbol{J}_i(\boldsymbol{q}_i) \in \mathbb{R}^{p \times n_i}$ is the manipulator Jacobian matrix for which the following assumption is made.

Assumption 2.1. *The matrix $\boldsymbol{J}_i$, $\forall i$, has always full rank along the robot trajectory, i.e., the robots are away from kinematic singularities.*

This is a common assumption when operating in the robots task space and can be mitigated by resorting to well-known techniques such as damped least squares methods [31]. By differentiating (2.2), the second order kinematic relationship is finally obtained

$$\ddot{\boldsymbol{x}}_i = \boldsymbol{J}_i(\boldsymbol{q}_i)\ddot{\boldsymbol{q}}_i + \dot{\boldsymbol{J}}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i)\dot{\boldsymbol{q}}_i. \tag{2.3}$$

For the purpose of the overall description of the work-cell, the variable $n = \sum_{i=1}^{N} n_i$ and the following collective vectors are introduced

$$\begin{aligned} \boldsymbol{x} &= \begin{bmatrix} \boldsymbol{x}_1^T, & \boldsymbol{x}_2^T, & \ldots, & \boldsymbol{x}_N^T \end{bmatrix}^T \in \mathbb{R}^{Np} \\ \boldsymbol{q} &= \begin{bmatrix} \boldsymbol{q}_1^T, & \boldsymbol{q}_2^T, & \ldots, & \boldsymbol{q}_N^T \end{bmatrix}^T \in \mathbb{R}^{n} \\ \boldsymbol{J}(\boldsymbol{q}) &= \mathrm{diag}\{\boldsymbol{J}_1(\boldsymbol{q}_1), \ldots, \boldsymbol{J}_N(\boldsymbol{q}_N)\} \in \mathbb{R}^{Np\times n} \end{aligned} \tag{2.4}$$

for which it holds

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}},\ \ddot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}. \tag{2.5}$$

Regarding the robot dynamic model, i.e., the motion equations based on forces and torques on robot, the Euler-Lagrange formulation is leveraged. The joint space dynamics is thus defined as

$$\boldsymbol{M}_i(\boldsymbol{q}_i)\ddot{\boldsymbol{q}}_i + \boldsymbol{C}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i)\dot{\boldsymbol{q}}_i + \boldsymbol{F}_i\dot{\boldsymbol{q}}_i + \boldsymbol{g}_i(\boldsymbol{q}_i) = \boldsymbol{\tau}_i - \boldsymbol{J}_i(\boldsymbol{q}_i)^T\boldsymbol{h}_i \tag{2.6}$$

where $\boldsymbol{\tau}_i \in \mathbb{R}^{n_i}$ is the joint torque vector, $\boldsymbol{M}_i(\boldsymbol{q}_i) \in \mathbb{R}^{n_i\times n_i}$ is the symmetric positive definite inertia matrix, $\boldsymbol{C}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i) \in \mathbb{R}^{n_i\times n_i}$ is the centrifugal and Coriolis terms matrix, $\boldsymbol{F}_i \in \mathbb{R}^{n_i\times n_i}$ is the matrix modeling viscous friction, $\boldsymbol{g}_i(\boldsymbol{q}_i) \in \mathbb{R}^{n_i}$ is the vector of gravity terms and $\boldsymbol{h}_i \in \mathbb{R}^{p}$ is the vector of interaction wrenches between the robot end effector and the environment. By introducing the auxiliary variable $\boldsymbol{n}_i \in \mathbb{R}^{n_i}$

$$\boldsymbol{n}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i) = \boldsymbol{C}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i)\dot{\boldsymbol{q}}_i + \boldsymbol{F}_i\dot{\boldsymbol{q}}_i + \boldsymbol{g}_i(\boldsymbol{q}_i) \tag{2.7}$$

eq. (2.6) can be rewritten as

$$\boldsymbol{M}_i(\boldsymbol{q}_i)\ddot{\boldsymbol{q}}_i + \boldsymbol{n}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i) = \boldsymbol{\tau}_i - \boldsymbol{J}_i(\boldsymbol{q}_i)^T\boldsymbol{h}_i. \tag{2.8}$$

Moreover, the model in (2.6) can be expressed in linear form with respect to the constant vector of the dynamic parameters of the robot denoted by $\boldsymbol{\pi}_i \in \mathbb{R}^{n_{\pi_i}}$, i.e., the following property holds [30].

Property 2.1. *Model in* (2.6) *can be reformulated as*

$$\boldsymbol{M}_i(\boldsymbol{q}_i)\ddot{\boldsymbol{q}}_i + \boldsymbol{C}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i)\dot{\boldsymbol{q}}_i + \boldsymbol{F}_i\dot{\boldsymbol{q}}_i + \boldsymbol{g}_i(\boldsymbol{q}_i) = \boldsymbol{Y}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i, \ddot{\boldsymbol{q}}_i)\boldsymbol{\pi}_i \quad (2.9)$$

with $\boldsymbol{Y}_i \in \mathbb{R}^{p \times n_{\pi_i}}$ the known regressor matrix.

By leveraging the results in [32], the joint torque vector can be selected as $\boldsymbol{\tau}_i = \boldsymbol{J}_i(\boldsymbol{q}_i)^T\boldsymbol{u}_i$ and, based on (2.6), the following dynamics in the operational space can be derived

$$\bar{\boldsymbol{M}}_i(\boldsymbol{x}_i)\ddot{\boldsymbol{x}}_i + \bar{\boldsymbol{C}}_i(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i)\dot{\boldsymbol{x}}_i + \bar{\boldsymbol{\eta}}_i(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i) = \boldsymbol{u}_i - \boldsymbol{h}_i \quad (2.10)$$

where it holds

$$\begin{aligned} \bar{\boldsymbol{M}}_i &= (\boldsymbol{J}_i\boldsymbol{M}_i^{-1}\boldsymbol{J}_i^T)^{-1}, & \bar{\boldsymbol{\eta}}_i &= \bar{\boldsymbol{F}}_i(\boldsymbol{x}_i)\dot{\boldsymbol{x}}_i + \bar{\boldsymbol{g}}(\boldsymbol{x}_i) \\ \bar{\boldsymbol{F}}_i &= \bar{\boldsymbol{M}}_i\boldsymbol{J}_i\boldsymbol{M}_i^{-1}\boldsymbol{F}_i, & \bar{\boldsymbol{g}}_i &= \bar{\boldsymbol{M}}_i\boldsymbol{J}_i\boldsymbol{M}_i^{-1}\boldsymbol{g}_i \end{aligned}$$

whereas $\bar{\boldsymbol{C}}_i$ is the Coriolis and centrifugal matrix for which it is supposed that Christoffel symbols of the first kind associated with the matrix $\bar{\boldsymbol{M}}_i$ are adopted [33]. In view of Assumption 2.1, the following properties hold [33, 34].

Property 2.2. *Matrix $\bar{\boldsymbol{M}}_i$ is symmetric and positive definite.*

Property 2.3. *Matrix $\dot{\bar{\boldsymbol{M}}}_i - 2\bar{\boldsymbol{C}}_i$ is skew-symmetric, i.e., $\boldsymbol{v}^T(\dot{\bar{\boldsymbol{M}}}_i - 2\bar{\boldsymbol{C}}_i)\boldsymbol{v} = 0$, $\forall\, \boldsymbol{v} \in \mathbb{R}^p$.*

Property 2.4. *Model in* (2.6) *can be expressed as*

$$\bar{\boldsymbol{M}}_i\ddot{\boldsymbol{x}}_i + \bar{\boldsymbol{C}}_i\dot{\boldsymbol{x}}_i + \bar{\boldsymbol{\eta}}_i = \bar{\boldsymbol{Y}}(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i, \ddot{\boldsymbol{x}}_i)\boldsymbol{\pi}_i \quad (2.11)$$

with $\bar{\boldsymbol{Y}}(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i, \ddot{\boldsymbol{x}}_i) = \bar{\boldsymbol{M}}_i\boldsymbol{J}_i\boldsymbol{M}_i^{-1}\boldsymbol{Y}(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i, \ddot{\boldsymbol{q}}_i)$.

Note that generally the exact dynamic model is *unknown* and only an estimate of the model in (2.10) is available:

$$\begin{aligned}\hat{\bar{\boldsymbol{M}}}_i\ddot{\boldsymbol{x}}_i + \hat{\bar{\boldsymbol{C}}}_i\dot{\boldsymbol{x}}_i + \hat{\bar{\boldsymbol{\eta}}}_i &= \bar{\boldsymbol{Y}}_i(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i, \ddot{\boldsymbol{x}}_i)\hat{\boldsymbol{\pi}}_i \\ &= \boldsymbol{u}_i - \boldsymbol{h}_i - \bar{\boldsymbol{Y}}_i(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i, \ddot{\boldsymbol{x}}_i)\tilde{\boldsymbol{\pi}}_i\end{aligned} \tag{2.12}$$

where $\tilde{\boldsymbol{\pi}}_i = \boldsymbol{\pi}_i - \hat{\boldsymbol{\pi}}_i$ and the symbol $\hat{\cdot}$ denotes the estimate of the corresponding quantity.

2.3 Information exchange modeling

As stated at the beginning of the chapter, it is of interest to devise distributed solutions for multi-robot control. As common [35], the information exchange between robots is described through a connectivity graph $\mathcal{G}(\mathcal{E}, \mathcal{V})$ characterized by the set $\mathcal{V}$ of nodes representing the robots, and the set $\mathcal{E} = \mathcal{V} \times \mathcal{V}$ of edges connecting the nodes and representing the communication links, i.e., robot i can send information to robot j when it holds $(i, j) \in \mathcal{E}$. Some notions about graph theory are thus recalled here.

The graph topology can be described via the $(N \times N)$ adjacency matrix $\boldsymbol{A}$

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

whose element a_{ij} is equal to 1 if the j th node can send information to i th node, i.e., $(j, i) \in \mathcal{E}$, and 0 otherwise; such matrix is symmetric in case of undirected graphs, i.e., in case all the communication links among the robots are bi-directional.
The i th robot receives information only from its neighbors $\mathcal{N}_i = \{j \in \mathcal{V} : \ (j, i) \in \mathcal{E}\}$. The cardinality of $\mathcal{N}_i$ is the in-degree of node i, i.e., $d_i = |\mathcal{N}_i| = \sum_{j=1}^{N} a_{ij}$. Moreover, the cardinality of the set of nodes receiving information from node i represents the out-degree of node i, i.e., $D_i = \sum_{k=1}^{N} a_{ki}$.

The communication topology can be also characterized by the

$(N \times N)$ Laplacian matrix [35, 36] defined as

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

The Laplacian matrix is often preferred in the field of multi-agent systems due to the following property.

Property 2.5. *All eigenvalues of $\boldsymbol{L}$ have real part equal to or greater than zero. Moreover, $\boldsymbol{L}$ exhibits at least a zero eigenvalue with corresponding right eigenvector the $N \times 1$ vector of all ones $\mathbf{1}_N$. Hence, rank$(\boldsymbol{L}) \leq N - 1$, with rank$(\boldsymbol{L}) = N - 1$ if and only if the graph is strongly connected, i.e., if any two distinct nodes of the graph can be connected via a directed path, and $\boldsymbol{L}\mathbf{1}_N = \mathbf{0}_N$, where $\mathbf{0}_N$ is the $(N \times 1)$ null vector.*

The following assumption about the communication graph is made in the thesis.

Assumption 2.2. *The graph topology is assumed fixed, i.e., there are no communication links that can appear or disappear over the time.*

2.4 Cooperative task formulation

The cooperative task is specified by means of a task function $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\boldsymbol{x}) \in \mathbb{R}^m$ which depends on the collective vector $\boldsymbol{x}$ as follows

$$\boldsymbol{\sigma} = \boldsymbol{J}_\sigma \boldsymbol{x}, \quad \dot{\boldsymbol{\sigma}} = \boldsymbol{J}_\sigma \dot{\boldsymbol{x}}, \quad \ddot{\boldsymbol{\sigma}} = \boldsymbol{J}_\sigma \ddot{\boldsymbol{x}} \tag{2.13}$$

with $\boldsymbol{J}_\sigma \in \mathbb{R}^{m \times Np}$ the constant task Jacobian matrix which, as discussed in [37], can be used to formalize a variety of multi-robot tasks of general interest.

As a possible and flexible choice, the task can be expressed by means of a proper set of absolute-relative variables. In particular, the absolute variables define the position and orientation of the centroid of the end effector configurations, i.e.,

$$\boldsymbol{\sigma}_1 = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i = \boldsymbol{J}_{\sigma_1}\boldsymbol{x} \tag{2.14}$$

with $\boldsymbol{J}_{\sigma_1} = \frac{1}{N}\boldsymbol{1}_N^T \otimes \boldsymbol{I}_p \in \mathbb{R}^{p\times Np}$.

Regarding the relative motion, it can be described as below

$$\boldsymbol{\sigma}_2 = [(\boldsymbol{x}_N - \boldsymbol{x}_{N-1})^T \ldots (\boldsymbol{x}_2 - \boldsymbol{x}_1)^T]^T = \boldsymbol{J}_{\sigma_2}\boldsymbol{x} \tag{2.15}$$

with

$$\boldsymbol{J}_{\sigma_2} = \begin{bmatrix} -\boldsymbol{I}_p & \boldsymbol{I}_p & \boldsymbol{O}_p & \ldots & \boldsymbol{O}_p \\ \boldsymbol{O}_p & -\boldsymbol{I}_p & \boldsymbol{I}_p & \ldots & \boldsymbol{O}_p \\ \vdots & & \ddots & & \vdots \\ \boldsymbol{O}_p & \ldots & \boldsymbol{O}_p & -\boldsymbol{I}_p & \boldsymbol{I}_p \end{bmatrix} \in \mathbb{R}^{(N-1)p\times Np} \tag{2.16}$$

Thus, a possible choice for the task function in (2.13) is

$$\boldsymbol{\sigma} = \begin{bmatrix} \boldsymbol{\sigma}_1 \\ \boldsymbol{\sigma}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_{\sigma_1} \\ \boldsymbol{J}_{\sigma_2} \end{bmatrix}\boldsymbol{x} = \boldsymbol{J}_{\sigma}\boldsymbol{x} \tag{2.17}$$

with $\boldsymbol{J}_{\sigma} \in \mathbb{R}^{Np\times Np}$ and $m = Np$.

Moreover, according to the taxonomy presented in [37], it is generally possible to identify the following entities (or a subset of them) in a generic multi-robot work-cell:

- *positioners*, that are the elements in charge of moving the workpieces and placing and/or holding them in specific configurations;
- *workers*, that are the elements responsible for executing the work on the workpieces;
- *watchers*, that are elements equipped with sensors allowing to monitor the work-cell.

In particular, workers are generally robotic manipulators, while positioners could be specialized devices, such as conveyors, or robotic manipulators as well. Concerning the watchers, they are generally robots equipped with external sensors, such as vision devices, and their motion is planned according to the cooperative task as well as to the motion of workers and positioners.

2.5 Robustness to faults

A failure to detect faults in a robotic system can lead to the latter behaving improperly and unexpectedly, endangering the safety of people in the same workspace and the surrounding environment in general. It follows that, for the robotic system to be reliable, it must be equipped with the ability to recognize and respond to faults in a timely manner. More specifically, in the case of single-robot systems, the property of robustness to failures requires that the robot itself must be able to identify any faults of *on board* sensors or actuators. In the case of multi-robot systems, this identification capability must be extended to the entire team, i.e., it must be possible to recognize any faulty robot in the system. This allows, on the one hand, not to compromise cooperative tasks that require the participation of multiple robots in order to be completed and, on the other hand, to potentially exploit the intrinsic redundancy of multi-robot systems in such a way to accomplish the assigned tasks also in the case of failure of one or more units. However, it is clear that the fault detection strategy strictly depends on the architecture of the multi-robot system: in the centralized case, the central unit knows the state of each robot in the system (i.e., has access to the *global* state of the system) and on the basis of the latter it can identify possible malfunctions, while in the more challenging distributed case, each robot only knows its own state and that of neighboring robots and must, on the basis of this *local* knowledge, establish possible failures in the overall team. Note that each robot needs to be able to detect and identify any possible fault in the team even if no direct communication exists with the malfunctioning robot.

The properties of reliability and robustness to faults are thus of primary importance for a robotic system and have driven the interest of the scientific community towards these issues for years.

Starting from solutions for single units systems and then for multi-robot systems with centralized architectures [38, 39, 40], the research is now mainly focused on the case of distributed architectures where each robot is required to recognize a fault of a teammate exclusively on the basis of *local* information. In this regard, observer-based approaches are traditionally adopted which involve the definition of residual signals, obtained by comparing measured and estimated quantities, that are then monitored to recognize and isolate faults. According to this approach, a FDI strategy for non linear uncertain systems is designed in [41] which also relies on adaptive thresholds to detect faults, while large scale systems composed of non linear agents are analyzed in [42], where a decomposition into smaller overlapping subsystems is proposed. As a different approach, a bank of unknown input observers is adopted in [43] with second-order linear time-invariant systems, where the existence of these observers is provided for two classes of distributed control laws. Heterogeneous linear agents with possible simultaneous faults are then considered in [44] but the designed strategy only allows each robot to detect its own fault or the one of a direct neighbor. Analogously, the same limitation is present in [45, 46].

Although much effort has been devoted to distributed FDI schemes, few contributions deal with multi-manipulator systems which are characterized by non-linear continuous-time dynamics. In this regard, the work in [47] leverages H_∞ theory and focuses on multi Euler-Lagrange systems involved in a synchronized set-point regulation problem. A leader tracking task with possible communication link and actuator faults is instead discussed in [48] where a fault-tolerant distributed control protocol is proposed which does not rely on a FDI scheme. Finally, a distributed formation control problem with multiple Euler-Lagrange systems is addressed in [49] where a fault diagnosis strategy is devised which requires each robot to broadcast an alarm signal in the case of fault.

Motivated by the above reasons and differently from the existing solutions, in this thesis a general distributed framework for multiple mobile-manipulator systems, presented in [28, 29], is investigated in order to allow each robot to detect and isolate possible faults of any other robot in the team without the need of direct communication with it. The solution builds on the schemes presented in [50] and [51], where the first focuses on continuous-time single integrator dynamic systems, whereas the latter deals with discrete-time linear systems.

Consider a fault affecting the joint actuators of the i th robot, $\boldsymbol{f}_i \in \mathbb{R}^{n_i}$, and thus, in the presence of the fault, (2.8) becomes

$$\boldsymbol{M}_i(\boldsymbol{q}_i)\ddot{\boldsymbol{q}}_i + \boldsymbol{n}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i) + \boldsymbol{d}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i) = \boldsymbol{\tau}_i - \boldsymbol{J}_i^T(\boldsymbol{q}_i)\boldsymbol{h}_i + \boldsymbol{f}_i \quad (2.18)$$

where $\boldsymbol{d}_i \in \mathbb{R}^{n_i}$ is an additional vector collecting the modeling uncertainties and disturbances (e.g., low-velocity friction, motor electromagnetic disturbances, noise).

Based on (2.18) and (2.10), the equation of motion of the end effector in the Cartesian space can be derived, i.e.,

$$\bar{\boldsymbol{M}}_i(\boldsymbol{q}_i)\ddot{\boldsymbol{x}}_i + \bar{\boldsymbol{n}}_i(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i) + \bar{\boldsymbol{d}}_i(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i) = \boldsymbol{u}_i - \boldsymbol{h}_i + \boldsymbol{J}_{M,i}^T\boldsymbol{f}_i \quad (2.19)$$

with

$$\bar{\boldsymbol{n}}_i = \boldsymbol{J}_{M,i}^T\boldsymbol{n}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i) - \bar{\boldsymbol{M}}_i\dot{\boldsymbol{J}}_i(\boldsymbol{q})\dot{\boldsymbol{q}}_i, \quad \bar{\boldsymbol{d}}_i = \boldsymbol{J}_{M,i}^T\boldsymbol{d}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i),$$

$\boldsymbol{J}_{M,i}$ a dynamically consistent generalized inverse of $\boldsymbol{J}_i$ given by

$$\boldsymbol{J}_{M,i} = \boldsymbol{M}_i^{-1}(\boldsymbol{q}_i)\boldsymbol{J}_i^T(\boldsymbol{q}_i)\bar{\boldsymbol{M}}_i.$$

The following assumption is made in the following.

Assumption 2.3. *For each robot, the vector of modeling uncertainties and disturbances in the operational space, $\bar{\boldsymbol{d}}_i$, is norm-bounded by a positive scalar $\bar{d}$, i.e., it holds $\|\bar{\boldsymbol{d}}_i\| \leq \bar{d} \quad \forall i = 1, \ldots, N$.*

Remark 2.1. *It is worth noticing that the fault $\boldsymbol{f}_i$ does not affect the end effector dynamics if it belongs to the null space of the* *matrix* $\boldsymbol{J}_{M,i}^T$. *Thus, such faults cannot be detected with the above formulation since they affect the internal dynamics of a given manipulator and cannot be visible to the other ones through the observation of the dynamics at the end effector.*

By considering the extended state vector $\boldsymbol{z}_i = \begin{bmatrix}\boldsymbol{x}_i^T & \dot{\boldsymbol{x}}_i^T\end{bmatrix}^T \in \mathbb{R}^{2p}$, the end effector equation of motion (2.19) can be written in matrix form as follows

$$\dot{\boldsymbol{z}}_i = \boldsymbol{A}\boldsymbol{z}_i + \boldsymbol{B}_i(\boldsymbol{z}_i)(\boldsymbol{u}_i - \boldsymbol{h}_i + \boldsymbol{J}_{M,i}^T\boldsymbol{f}_i - \bar{\boldsymbol{n}}_i(\boldsymbol{z}_i) - \bar{\boldsymbol{d}}_i(\boldsymbol{z}_i)) \qquad (2.20)$$

with

$$\boldsymbol{A} = \begin{bmatrix}\boldsymbol{O}_p & \boldsymbol{I}_p \\ \boldsymbol{O}_p & \boldsymbol{O}_p\end{bmatrix} \qquad \boldsymbol{B}_i(\boldsymbol{z}_i) = \begin{bmatrix}\boldsymbol{O}_p \\ \bar{\boldsymbol{M}}_i^{-1}(\boldsymbol{z}_i)\end{bmatrix}.$$

Formally, the following is the problem that is aimed to be solved for guaranteeing robustness to faults.

Problem 2.1. *Consider a team of N mobile manipulators for which a cooperative task is assigned. Assume a central control unit is not present and robot i has a fault at a certain time t_f, i.e., $\|\boldsymbol{f}_i(t_f)\| > 0$. The objective is to enable each robot in the team i) to detect that a fault has occurred and ii) to identify which robot is in fault.*

2.5.1 Observer-controller scheme

Assume that the robots in the work-cell are in charge of achieving a global task depending on the overall state of the cell

$$\boldsymbol{z} = \begin{bmatrix}\boldsymbol{z}_1^T & \boldsymbol{z}_2^T & \dots & \boldsymbol{z}_N^T\end{bmatrix}^T \in \mathbb{R}^{2pN}.$$

The observer-controller scheme proposed for the solution of Problem 2.1 considers that each robot firstly estimates the overall state of the cell through an observer that is based on the information exchange with the neighboring robots; then, the state estimate is used to calculate the input that allows to achieve a global tasks as in a centralized configuration. The overall distributed control architecture of the i th robot, combining both the observer-controller and the fault detection schemes, is reported in Figure 2.3.

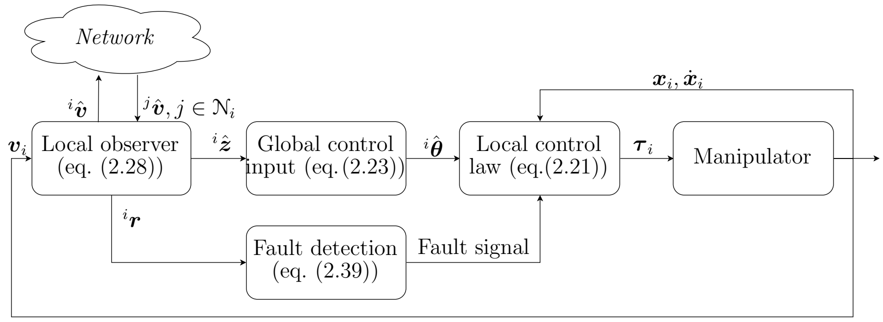

Network
${}^i\hat{\boldsymbol{v}}$
${}^j\hat{\boldsymbol{v}}, j \in \mathcal{N}_i$
$\boldsymbol{v}_i$
Local observer (eq. (2.28))
${}^i\hat{\boldsymbol{z}}$
Global control input (eq. (2.23))
${}^i\hat{\boldsymbol{\theta}}$
Local control law (eq.(2.21))
$\boldsymbol{\tau}_i$
Manipulator
$\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i$
${}^i\boldsymbol{r}$
Fault detection (eq. (2.39))
Fault signal

Figure 2.3: Distributed control architecture of the i th robot.

It is considered that a local fault signal is generated by robot i in case a fault is detected in such a way that it can be exploited to determine the local control input, e.g., the robot may stop when a fault of a teammate is detected.

For each robot, assume that the following control law is adopted

$$\boldsymbol{u}_i = \bar{\boldsymbol{M}}_i(\boldsymbol{z}_i)\left(\boldsymbol{\theta}_i(\boldsymbol{z}) + \bar{\boldsymbol{n}}_i(\boldsymbol{z}_i) + \boldsymbol{h}_i\right) \tag{2.21}$$

where $\boldsymbol{\theta}_i$ is an auxiliary input depending on the overall state $\boldsymbol{z}$ which is not available in a distributed setting. In case of a centralized system, in which a central unit, capable of computing the control input for each robot, is present, the collective auxiliary input

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\theta}_1^T & \boldsymbol{\theta}_2^T & \dots & \boldsymbol{\theta}_N^T \end{bmatrix} \in \mathbb{R}^{Np}$$

for achieving a global task would assume the following form

$$\boldsymbol{\theta} = \boldsymbol{K}\boldsymbol{z} + \boldsymbol{\theta}_f \tag{2.22}$$

where $\boldsymbol{K} \in \mathbb{R}^{Np \times 2Np}$ is a constant gain matrix and $\boldsymbol{\theta}_f \in \mathbb{R}^{Np}$ represents a feed-forward term [52].

In the absence of the central unit, the computation of the control law (2.22) requires that the i th robot estimates the overall cell state in order to use such an estimate, denoted as ${}^i\hat{\boldsymbol{z}}$, in lieu of $\boldsymbol{z}$ (see e.g., [53]). More specifically, based on the estimate ${}^{i}\hat{\boldsymbol{z}}$, the ith robot computes an estimate ${}^{i}\hat{\boldsymbol{\theta}}_g$ of the collective control input (2.22), i.e.,

$$ {}^{i}\hat{\boldsymbol{\theta}}_g = \boldsymbol{K}\,{}^{i}\hat{\boldsymbol{z}} + \boldsymbol{\theta}_f. \tag{2.23}$$

Moreover, in the decentralized solution, a local stabilizing term must be added in order to ensure the convergence of the state estimation to the actual value. Therefore, the effective auxiliary control input is obtained by considering the following selection matrix

$$\boldsymbol{\Gamma}_i = \{\boldsymbol{O}_p \quad \cdots \quad \underbrace{\boldsymbol{I}_p}_{i \text{ th node}} \quad \cdots \quad \boldsymbol{O}_p\} \in \mathbb{R}^{p\times Np} \tag{2.24}$$

such that

$$\boldsymbol{\theta}_i = \boldsymbol{\theta}_{g,i} + \boldsymbol{\theta}_{s,i} = \boldsymbol{\Gamma}_i {}^{i}\hat{\boldsymbol{\theta}}_g + \boldsymbol{K}_s \boldsymbol{z}_i \tag{2.25}$$

where $\boldsymbol{K}_s \in \mathbb{R}^{p\times 2p}$ is a constant gain matrix. Based on (2.21) and (2.25), (2.20) can be rearranged as

$$\begin{aligned}\dot{\boldsymbol{z}}_i &= \boldsymbol{A}\boldsymbol{z}_i + \boldsymbol{C}(\boldsymbol{\theta}_{g,i} + \boldsymbol{\theta}_{s,i}) + \boldsymbol{B}_i(-\bar{\boldsymbol{d}}_i(\boldsymbol{z}_i) + \boldsymbol{J}_{M,i}^T \boldsymbol{f}_i)\\ &= \boldsymbol{F}\boldsymbol{z}_i + \boldsymbol{C}\boldsymbol{\theta}_{g,i} + \boldsymbol{B}_i(-\bar{\boldsymbol{d}}_i(\boldsymbol{z}_i) + \boldsymbol{J}_{M,i}^T \boldsymbol{f}_i)\end{aligned} \tag{2.26}$$

with $\boldsymbol{C} = \begin{bmatrix}\boldsymbol{O}_p & \boldsymbol{I}_p\end{bmatrix}^T$ and $\boldsymbol{F} = \boldsymbol{A} + \boldsymbol{C}\boldsymbol{K}_s$, which leads to the following dynamics of the collective state

$$\dot{\boldsymbol{z}} = (\boldsymbol{I}_N \otimes \boldsymbol{F})\boldsymbol{z} + (\boldsymbol{I}_N \otimes \boldsymbol{C})\boldsymbol{\theta}_g + \bar{\boldsymbol{B}}(\boldsymbol{f}_M - \bar{\boldsymbol{d}}) = \bar{\boldsymbol{F}}\boldsymbol{z} + \bar{\boldsymbol{C}}\boldsymbol{\theta}_g + \bar{\boldsymbol{B}}(\boldsymbol{\phi}_M - \bar{\boldsymbol{d}}) \tag{2.27}$$

where the symbol ⊗ denotes the Kronecker product and

$$\boldsymbol{\theta}_g = \begin{bmatrix}\boldsymbol{\theta}_{g,1}\\ \vdots \\ \boldsymbol{\theta}_{g,N}\end{bmatrix}, \bar{\boldsymbol{B}} = \begin{bmatrix}\boldsymbol{B}_1(\boldsymbol{z}_1)\\ \vdots \\ \boldsymbol{B}_N(\boldsymbol{z}_N)\end{bmatrix}, \boldsymbol{f}_M = \begin{bmatrix}\boldsymbol{J}_{M,1}^T\boldsymbol{f}_1\\ \vdots \\ \boldsymbol{J}_{M,N}^T\boldsymbol{f}_N\end{bmatrix}, \bar{\boldsymbol{d}} = \begin{bmatrix}\bar{\boldsymbol{d}}_1\\ \vdots \\ \bar{\boldsymbol{d}}_N\end{bmatrix}.$$

2.5.1.1 Collective state estimation

In order to estimate the collective state, $\boldsymbol{z}$, each robot runs an observer requiring only local information provided by robot sensors and information received from neighbor robots. The same observer system is also exploited for the FDI strategy, without increasing the information exchange burden.

The observer of robot i has the following dynamics

$$
{}^i\dot{\hat{\boldsymbol{z}}} = k_o\left(\sum_{j\in\mathcal{N}_i}\left({}^j\hat{\boldsymbol{v}} - {}^i\hat{\boldsymbol{v}}\right) + \boldsymbol{\Pi}_i\left(\boldsymbol{v} - {}^i\hat{\boldsymbol{v}}\right)\right) + \bar{\boldsymbol{C}}\,{}^i\hat{\boldsymbol{\theta}}_g + \bar{\boldsymbol{F}}\,{}^i\hat{\boldsymbol{z}} \tag{2.28}
$$

where ${}^i\hat{\boldsymbol{z}}$ is the estimate of the collective state made by robot i and $\boldsymbol{\Pi}_i \in \mathbb{R}^{2Np\times 2Np}$ is a selection matrix defined as

$$
\boldsymbol{\Pi}_i = \mathrm{diag}\{\boldsymbol{O}_{2p}, \quad \cdots, \quad \underbrace{\boldsymbol{I}_{2p}}_{i\,\text{th robot}} \quad ,\cdots \quad \boldsymbol{O}_{2p}\} \tag{2.29}
$$

which selects the components of $\boldsymbol{v}$ and ${}^i\hat{\boldsymbol{v}}$ referred to robot i. The variable $\boldsymbol{v}$ is an auxiliary state defined as

$$
\boldsymbol{v} = \boldsymbol{z} - \int_{t_0}^{t}\left(\bar{\boldsymbol{F}}\boldsymbol{z} + \bar{\boldsymbol{C}}\boldsymbol{\theta}_g\right)d\sigma \tag{2.30}
$$

where t_0 is the initial time instant. Its estimate ${}^i\hat{\boldsymbol{v}}$ in (2.28) is then obtained as

$$
{}^i\hat{\boldsymbol{v}} = {}^i\hat{\boldsymbol{z}} - \int_{t_0}^{t}\left(\bar{\boldsymbol{F}}\,{}^i\hat{\boldsymbol{z}} + \bar{\boldsymbol{C}}\,{}^i\hat{\boldsymbol{\theta}}_g\right)d\sigma \tag{2.31}
$$

which depends only on local information available to robot i. It is worth noticing that each observer is updated using only the estimates ${}^j\hat{\boldsymbol{v}}$ received from direct neighbors, which is, thus, the only information exchanged among neighbors.

The collective estimation dynamics is

$$
\dot{\hat{\boldsymbol{z}}}^\star = -k_o\boldsymbol{L}^\star\hat{\boldsymbol{v}}^\star + k_o\boldsymbol{\Pi}^\star\tilde{\boldsymbol{v}}^\star + \boldsymbol{I}_N\otimes\bar{\boldsymbol{C}}\hat{\boldsymbol{\theta}}_g^\star + \boldsymbol{I}_N\otimes\bar{\boldsymbol{F}}\hat{\boldsymbol{z}}^\star \tag{2.32}
$$

where

$$
\begin{aligned}
\boldsymbol{L}^\star &= \boldsymbol{L}\otimes\boldsymbol{I}_{2Np},\ \boldsymbol{\Pi}^\star = \mathrm{diag}\{\boldsymbol{\Pi}_1,\ldots,\boldsymbol{\Pi}_N\} \in \mathbb{R}^{2Np^2\times 2Np^2},\\
\hat{\boldsymbol{z}}^\star &= \begin{bmatrix}{}^1\hat{\boldsymbol{z}}^T & \ldots & {}^N\hat{\boldsymbol{z}}^T\end{bmatrix}^T,\ \hat{\boldsymbol{v}}^\star = \begin{bmatrix}{}^1\hat{\boldsymbol{v}}^T & \ldots & {}^N\hat{\boldsymbol{v}}^T\end{bmatrix}^T \in \mathbb{R}^{2Np^2},
\end{aligned}
$$

$$\hat{\boldsymbol{\theta}}_g^\star = \begin{bmatrix} {}^1\hat{\boldsymbol{\theta}}_g^T & \dots & {}^N\hat{\boldsymbol{\theta}}_g^T \end{bmatrix}^T \in \mathbb{R}^{Np^2}, \; \tilde{\boldsymbol{v}}^\star = \mathbf{1}_N \otimes \boldsymbol{v} - \hat{\boldsymbol{v}}^\star.$$

Based on the properties of the Lagrangian matrix and by using the same arguments adopted in [50], the following lemma can be proven.

Lemma 2.1. *In the presence of a strongly connected directed communication graph and in the absence of faults and model uncertainties (i.e., $\|\boldsymbol{f}_M\| = 0$ and $\|\bar{\boldsymbol{d}}\| = 0$), the term $\tilde{\boldsymbol{v}}^\star$ is globally exponentially convergent to zero, given the observer* (2.28) *and for any $k_o > 0$.*

Proof. The proof is provided in the Appendix A.1. ■
In the presence of bounded uncertainties $\bar{\boldsymbol{d}}$ and on the basis on the results in [54], it can be easily proven that the system (2.32) is globally uniformly ultimately bounded (see [50]).

Based on Lemma 2.1, under the same assumptions, the following theorem states the convergence properties of the collective state estimation error $\tilde{\boldsymbol{z}}^\star = \mathbf{1}_N \otimes \boldsymbol{z} - \hat{\boldsymbol{z}}^\star$.

Theorem 2.2. *Consider a strongly connected directed communication graph and the absence of faults and model uncertainties (i.e., $\|\boldsymbol{f}_M\| = 0$ and $\|\bar{\boldsymbol{d}}\| = 0$). There exists a nonsingular permutation matrix $\boldsymbol{P} \in \mathbb{R}^{N^2 \times N^2}$ such that*

$$(\boldsymbol{P} \otimes \boldsymbol{I}_p)(\boldsymbol{I}_{Np^2} - \boldsymbol{\Pi}_u^\star) = \left[\begin{array}{c} \mathbf{0}_{Np \times Np^2} \\ \hline \boldsymbol{S} \end{array}\right]$$

where $\boldsymbol{S} \in \mathbb{R}^{Np(N-1) \times Np^2}$ is a full row rank matrix and $\boldsymbol{\Pi}_u^\star \in \mathbb{R}^{Np^2 \times Np^2}$ is defined as $\boldsymbol{\Pi}_u^\star = \mathbf{1}_N \otimes \begin{bmatrix} \boldsymbol{\Pi}_{u_1} \dots \boldsymbol{\Pi}_{u_N} \end{bmatrix}$, with $\boldsymbol{\Pi}_{u_l} \in \mathbb{R}^{Np \times Np}$ a matrix that nullifies all the elements of a Np-dimensional vector but the l th block of dimension p. Given the observer (2.28) *for any $k_o > 0$ and by selecting $\boldsymbol{K}_s$ and $\boldsymbol{K}$ in* (2.25) *in such a way that $\boldsymbol{F} = \boldsymbol{A} + \boldsymbol{C}\boldsymbol{K}_s$ is a Hurwitz matrix and the subsystem $\{\boldsymbol{I}_{N(N-1)} \otimes \boldsymbol{F}, (\boldsymbol{I}_{N(N-1)} \otimes \boldsymbol{C})\boldsymbol{S}\}$ is stabilized, the term $\tilde{\boldsymbol{z}}^\star$ is convergent to zero.*

Proof. The proof is provided in the Appendix A.2. ■
Finally, by resorting to the same arguments used for $\tilde{\boldsymbol{v}}^\star$, in the

presence of bounded uncertainties $\bar{\boldsymbol{d}}$, it can be proven that the system $\tilde{\boldsymbol{z}}^\star$ is globally uniformly ultimately bounded [54].

2.5.2 Fault diagnosis and isolation scheme

In order to detect and isolate the presence of a faulty robot, the collective state observer devised in Section 2.5.1.1 is exploited for defining a set of residual vectors which allow to monitor the health condition of each robot. In detail, the following residual vector [50] for the i th robot is considered

$$
{}^{i}\boldsymbol{r} = \sum_{j \in \mathcal{N}_i} \left({}^{j}\hat{\boldsymbol{v}} - {}^{i}\hat{\boldsymbol{v}} \right) + \boldsymbol{\Pi}_i (\boldsymbol{v} - \hat{\boldsymbol{v}}_i) \tag{2.33}
$$

that can be rearranged as a stacked vector ${}^{i}\boldsymbol{r} = \begin{bmatrix} {}^{i}\boldsymbol{r}_1^T & \dots & {}^{i}\boldsymbol{r}_N^T \end{bmatrix}^T \in \mathbb{R}^{2Np}$, where each component ${}^{i}\boldsymbol{r}_k \in \mathbb{R}^{2p}$ represents the residual computed by robot i relative to robot k, and that allows the robot i to monitor the healthy state of robot k. By introducing the collective residual vector $\boldsymbol{r}^\star \in \mathbb{R}^{2Np^2}$

$$
\boldsymbol{r}^\star = \begin{bmatrix} {}^{1}\boldsymbol{r}^T & \dots & {}^{N}\boldsymbol{r}^T \end{bmatrix}^T \in \mathbb{R}^{2Np^2} \tag{2.34}
$$

eq. (2.33) leads to

$$
\boldsymbol{r}^\star = \tilde{\boldsymbol{L}}^\star \tilde{\boldsymbol{v}}^\star. \tag{2.35}
$$

It is possible to select the vector collecting all the residuals relative to the k th robot, i.e., $\boldsymbol{r}_k^\star = \begin{bmatrix} {}^{1}\boldsymbol{r}_k^{\star T}, \dots, {}^{N}\boldsymbol{r}_k^{\star T} \end{bmatrix}^T \in \mathbb{R}^{2Np}$ as

$$
\boldsymbol{r}_k^\star = \mathrm{diag}\left\{ \boldsymbol{\Gamma}_k, \boldsymbol{\Gamma}_k, \dots, \boldsymbol{\Gamma}_k \right\} \boldsymbol{r}^\star = \boldsymbol{\Gamma}_k^\star \tilde{\boldsymbol{L}}^\star \tilde{\boldsymbol{v}}^\star = \tilde{\boldsymbol{L}}_k^\star \, \tilde{\boldsymbol{v}}_k^\star \tag{2.36}
$$

where $\tilde{\boldsymbol{L}}_k^\star = \boldsymbol{L} \otimes \boldsymbol{I}_{2p} + \boldsymbol{\Pi}_k$ and the vector $\tilde{\boldsymbol{v}}_k^\star \begin{bmatrix} {}^{1}\tilde{\boldsymbol{v}}_k^T, \dots, {}^{N}\tilde{\boldsymbol{v}}_k^T \end{bmatrix}^T \in \mathbb{R}^{2Np}$ collects the estimation errors ${}^{i}\tilde{\boldsymbol{v}}_k$ of the observers.

Note that, in case of strongly connected communication graph, the matrix $-\tilde{\boldsymbol{L}}^\star$ is Hurwitz [55], thus the following property holds true:

Property 2.6. *The matrix* $-\tilde{\boldsymbol{L}}_k^\star$ *is Hurwitz, thus the dynamics in* (A.19) *is asymptotically stable and there exist two positive con-*

stants κ, $\lambda > 0$ such that [56]

$$\left\| \mathrm{e}^{-k_o \tilde{\boldsymbol{L}}_k^\star t} \right\| \leq \kappa \mathrm{e}^{-\lambda t}, \ \ \forall t. \tag{2.37}$$

The following theorems can be proved regarding the residual vectors in case of absence or presence of faults.

Theorem 2.3. *In the absence of faults, the residual ${}^i\boldsymbol{r}_k$, i.e., the residual computed by robot i and relative to robot k, is norm-bounded by the following time-varying threshold ${}^i\mu_k(t)$*

$${}^i\mu_k(t) = (\sqrt{2p}d_i + {}^i\delta_k) \left(\|\tilde{\boldsymbol{v}}_k^\star(0)\| \, \kappa \mathrm{e}^{-\lambda t} + \frac{\kappa \sqrt{N} \, \bar{d}}{\lambda \, \varepsilon_m(\bar{\boldsymbol{M}}_k)}(1 - \mathrm{e}^{-\lambda t}) \right). \tag{2.38}$$

where ${}^i\delta_k = 1$ if $i = k$ and ${}^i\delta_k = 0$ otherwise, $\varepsilon_m(\bar{\boldsymbol{M}}_k)$ is the minimum eigenvalue of $\bar{\boldsymbol{M}}_k$.

Proof. The proof is provided in the Appendix A.3.

■

Consider a fault $\boldsymbol{f}_k$ affecting the kth robot. The following theorem can be stated.

Theorem 2.4. *A fault occurring on the k th robot at time $t_f > t_0$, affects only the residuals components ${}^i\boldsymbol{r}_k$ ($\forall i = 1, 2, \ldots, N$) and not the residual components ${}^i\boldsymbol{r}_j$ ($\forall i, j = 1, 2, \ldots, N$ and $j \neq k$).*

Proof. The proof is provided in the Appendix sec:proof-thm-res-fault.

■

On the basis of Theorems 2.3 and 2.4, the following strategy can be implemented. When the following condition is verified, a fault on robot k is detected by robot i

$$\begin{cases} \exists\, t > t_f : & \|{}^i\boldsymbol{r}_k(t)\| > {}^i\mu_k(t) \\ \forall l \in (1, 2, \ldots, N), l \neq k, \forall t > t_0, & \|{}^i\boldsymbol{r}_l(t)\| \leq {}^i\mu_l(t). \end{cases} \tag{2.39}$$

Note that, in the presence of nonzero initial observer estimation errors and model uncertainty, the residuals are different from zero even in the absence of faults. For this reason, in order to avoid the occurrence of false alarms, and by virtue of Theorem 2.3, the decision about the occurrence of a fault is made only when a residual exceeds the adaptive thresholds computed in (2.38). Furthermore, it is worth remarking that Theorem 2.4 ensures that only the residual ${}^{i}\boldsymbol{r}_k$ is affected by the fault $\boldsymbol{f}_k$ while the other residuals computed by robot i are insensitive to it. The condition (2.39) ensures that all the robots of the cell can detect a fault even if there is no direct communication with the faulty robot. Algorithm 1 summarizes the steps of the fault detection and isolation strategy.

Algorithm 1 FDI strategy

```
Protocol: All robots repeat indefinitely the following operations,
here reported for robot $i$:
Gather the neighbors auxiliary variables ${}^{j}\hat{\boldsymbol{v}}$, $j \in \mathcal{N}_i$
Execute local observer according to eq. (2.28)
Update local auxiliary variable ${}^{i}\boldsymbol{v}$ according to eq. (2.31)
Compute the residual vector ${}^{i}\boldsymbol{r}$ according to eq. (2.33)
Compute the time-varying thresholds ${}^{i}\mu_k(t)$, $\forall k \in \mathcal{V}$ according
  to eq. (2.38)
if $\|{}^{i}\boldsymbol{r}_k(t)\| > {}^{i}\mu_k(t)$ for any $k \in \mathcal{V}$ then
    Activate procedure for handling fault on robot $k$
end if
```

2.5.3 Simulation results

The FDI strategy has been validated both in simulation [28] and in real-world [29], as shown in the following section. Simulations are performed with Matlab[1] environment and CoppeliaSim[2] sim-

[1] http://www.mathworks.com

[2] http://www.coppeliarobotics.com

ulator. A video is available at the following link[3]. A team of four worker Kinova Movo mobile robots ($N = 4$) deployed in a factory is considered as represented in Figure 2.4. Each robot is equipped with a 7 DOFs Kinova ultra lightweight Gen2 Jaco arm, an omnidirectional base (3 DOFs) and variable height torso (1 DOF); thus, it holds $n_i = 11$, $i = 1, 2, 3, 4$.

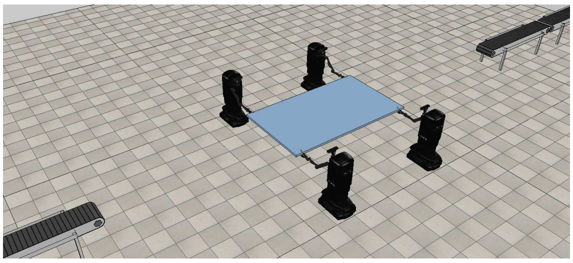


Figure 2.4: Simulation scenario composed of 4 mobile robots performing a cooperative transportation task.

A cooperative object transportation task between two conveyor belts is performed. The absolute-relative formulation for the task function $\boldsymbol{\sigma}$ in Section 2.4 is exploited and a desired trajectory $\boldsymbol{\sigma}_d(t)$ ($\dot{\boldsymbol{\sigma}}_d(t)$,$\ddot{\boldsymbol{\sigma}}_d(t)$) is assigned. By resorting to a second-order closed loop inverse kinematics law, the central control law in (2.22) for the cooperative task space trajectory tracking is given by

$$\boldsymbol{\theta} = \boldsymbol{J}_{\sigma}^{-1} \left[\ddot{\boldsymbol{\sigma}}_d + \boldsymbol{K}_d(\dot{\boldsymbol{\sigma}}_d - \dot{\boldsymbol{\sigma}}(\dot{\boldsymbol{x}})) + \boldsymbol{K}_p(\boldsymbol{\sigma}_d - \boldsymbol{\sigma}(\boldsymbol{x})) \right] \tag{2.40}$$

with $\boldsymbol{K}_d, \boldsymbol{K}_p \in \mathbb{R}^{Np \times Np}$ positive definite gain matrices, for which it is straightforward to show that the tracking error $\boldsymbol{\sigma}_d - \boldsymbol{\sigma}$ asymptotically converges to the origin. In accordance to the devised strategy, by replacing $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ in (2.40) with the respective estimates, each robot i can compute the estimated global input ${}^{i}\hat{\boldsymbol{\theta}}_g$ as in (2.23) and, in turn, can derive the local auxiliary input $\hat{\boldsymbol{\theta}}_i$ as in (2.25).

For the simulation case study, the following gains are selected: $\boldsymbol{K}_d = 20\boldsymbol{I}_{24}$ and $\boldsymbol{K}_p = 100\boldsymbol{I}_{24}$ in (2.40) and $k_o = 30$ in (2.28).

[3] https://www.youtube.com/watch?v=pW2nu8RqNO8

Moreover, an initial error $\|{}^{i}\tilde{\boldsymbol{z}}(0)\| > 0$ is considered $\forall i = 1,..,4$ and a model uncertainty and disturbance signal $\bar{\boldsymbol{d}}_i$ in (2.19) with bounded norm $\bar{d} = 0.12$ is simulated $\forall i = 1,..,4$. Finally, with regard to the communication graph, the following edge set is assumed $\mathcal{E} = \{(1,2), (2,3), (3,4), (4,1)\}$, thus resulting in a directed strongly connected graph.
The task is divided in the following steps: *(i)* starting from the initial configuration (Figure 2.5.a) the mobile robot team reaches the left conveyor belt where the object to transport is placed; *(ii)* the object is grasped (Figure 2.5.b) and moved towards the right conveyor belt where it is released (Figure 2.5.c). Finally, while the team is still approaching the picking conveyor belt, a fault occurs to robot 1 at time $t = 14$ s (Figure 2.5.d). A damage of the mobile base is simulated which leads to the stop of the respective 3 joints in 1 s. Furthermore, when a robot detects the fault of any other robot in the team, the policy of stopping its motion is adopted so as to minimize possible damages that would occur in case of tight connection.

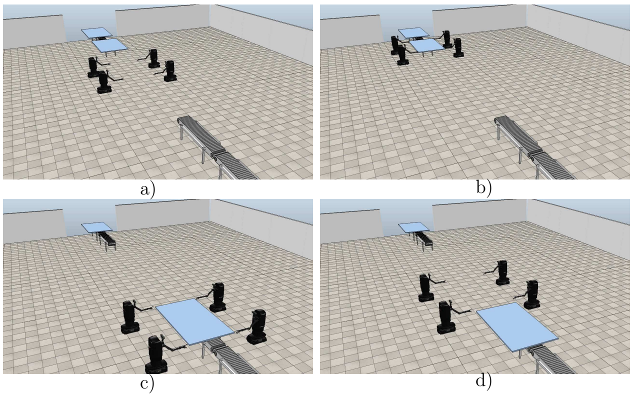


Figure 2.5: Simulation snapshots. In detail, a) the starting configuration, b) the grasping phase, c) the deposit phase and d) the fault occurrence.

Figures 2.6 and 2.7 summarize the numerical results. In particular, Figure 2.6 shows the norm of the task tracking error ($\tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}_d - \boldsymbol{\sigma}(\boldsymbol{x})$), of its derivative ($\dot{\tilde{\boldsymbol{\sigma}}} = \dot{\boldsymbol{\sigma}}_d - \dot{\boldsymbol{\sigma}}(\boldsymbol{x})$) and, finally, of the state estimation error ($\tilde{\boldsymbol{z}}^{\star}$), and it shows that the errors grow in correspondence of the fault occurrence at $t = 14\,\text{s}$. Figure 2.7 reports the norm of the residuals computed by the different robots and relative to the faulty robot 1 ($\|{}^i\boldsymbol{r}_1\|$, $\forall\, i$ in solid lines) together with the respective adaptive thresholds ($\|{}^i\mu_1\|$, $\forall\, i$ in dashed lines) and computed according to (2.38); without loss of generality, it has been considered $\|\tilde{\boldsymbol{y}}_k^\star(0)\| = 0$. The figure makes evident that all the robots are able to detect the fault, i.e. the condition $\|{}^i\boldsymbol{r}_1\| \geq {}^i\mu_1$ is verified $\forall i$ after the fault occurrence, even if there is no direct communication with the faulty robot.

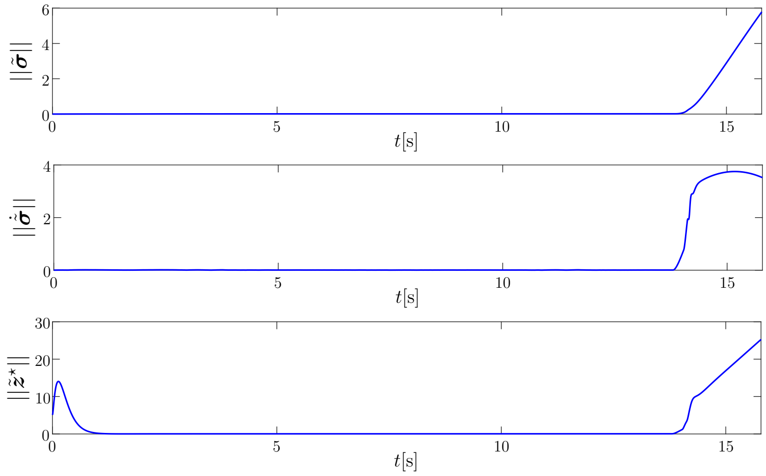


Figure 2.6: Simulation case study. Evolution of the norm of the task error $||\tilde{\boldsymbol{\sigma}}||$ (top), the norm of its derivative $||\dot{\tilde{\boldsymbol{\sigma}}}||$ (middle), and the norm of the state estimation error $||\tilde{\boldsymbol{z}}^\star||$ (bottom).

2.5.4 Experimental validation

This section presents the experimental results on a real multi-manipulator setup. A video of the experiment is available at the link[4]. As represented in Figure 2.8, a heterogeneous team of three robots ($N = 3$) is considered which is composed of: 1) a fixed-base

[4] https://www.youtube.com/watch?v=1ZdyH6B9tNE

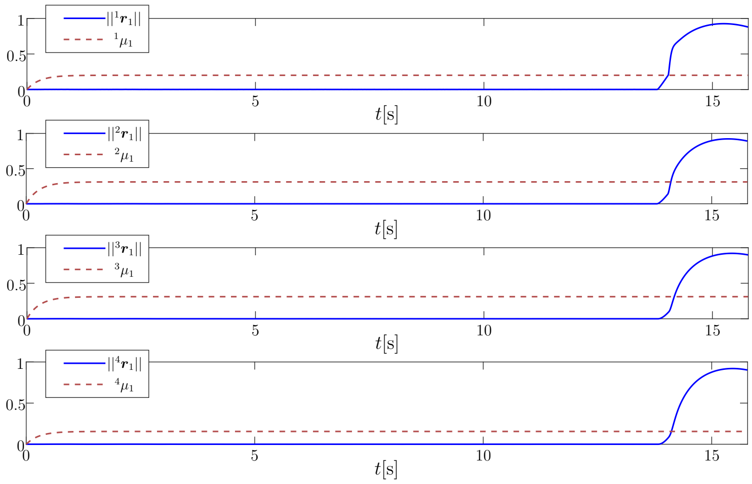


Figure 2.7: Simulation case study. Evolution of the norm of residual components relative to robot 1 (faulty robot) computed by the 4 robots (solid line) compared with the respective adaptive threshold (dashed line).

7 DOFs Kinova ultra lightweight Gen2 Jaco arm, for which it holds $n_1 = 7$; 2) a Kinova Movo mobile robot equipped with a Kinova Gen2 Jaco arm as in the simulations, for which it holds $n_2 = 11$; 3) a Kinova Movo mobile robot consisting of an omnidirectional base (3 DOFs), a variable height torso (1 DOF) and a 2-link robot arm (2 DOFs), for which it holds $n_3 = 6$. The Kinova Movo robots

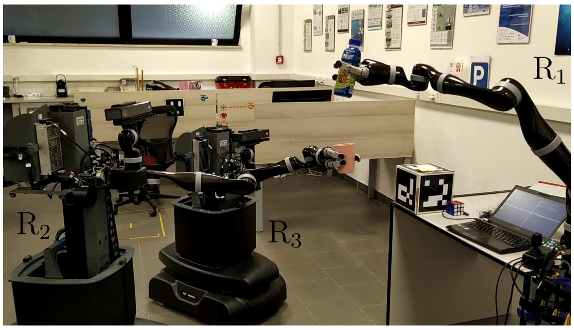


Figure 2.8: Experimetal setup composed of three heterogeneous robots.

are also equipped with RGB-D sensors, in particular Microsoft

Kinect v2, which allow to possibly monitor the scene; this sensor is considered as end effector of the latter robot. By resorting to the taxonomy in Section 2.4, the aformentioned robots will be referred to as worker, positioner and watcher, respectively.

Concerning the hardware specifications, Movo mobile robots are provided with two dedicated Intel NUC Kits NUC5i7RYH with Intel Core i7-5557U processor and 16 GB RAM, whereas the worker is controlled by a standard PC with Intel Core i7-5500U processor and 8 GB RAM; moreover, wi-fi modules TP-Link TL-WN821N are used for each robot which enable the intra-robot communication through a local network set on a TP-Link TD-W8960N router. Finally, the software architecture relies on ROS middleware and ArUco markers [57] have been introduced to initially localize mobile robots in the environment.

A cooperative service task is considered where the worker is in charge of pouring the contents of a bottle into a glass, while the positioner is in charge of holding that glass; finally, the watcher may be exploited to provide a different point of view for scene monitoring. As in the simulation case study, the desired trajectory $\boldsymbol{\sigma}_d(t)$ ($\dot{\boldsymbol{\sigma}}_d(t)$,$\ddot{\boldsymbol{\sigma}}_d(t)$) is defined in terms of absolute-relative coordinates and the control law in (2.40) is considered.

The following set of gains is used: $\boldsymbol{K}_p = \boldsymbol{I}_{18}$ and $\boldsymbol{K}_d = 2\boldsymbol{I}_{18}$ in (2.40) and $k_o = 5$ in (2.28). Concerning the communication graph, the directed strongly connected graph composed of the following set of edges $\mathcal{E} = \{(1,2),(2,3),(3,1)\}$ has been adopted. Starting from the initial configuration shown in Figure 2.9.a, the desired task consists of the following steps:

1. the positioner and the watcher move closer to the fixed-base manipulator;

2. the positioner hangs out the glass as in Figure 2.9.b;

3. the fixed-base manipulator pours the contents of the bottle into the glass as in Figure 2.9.c;

4. the mobile base manipulator delivers the glass while the fixed-base manipulator returns in a configuration with the

bottle vertical.

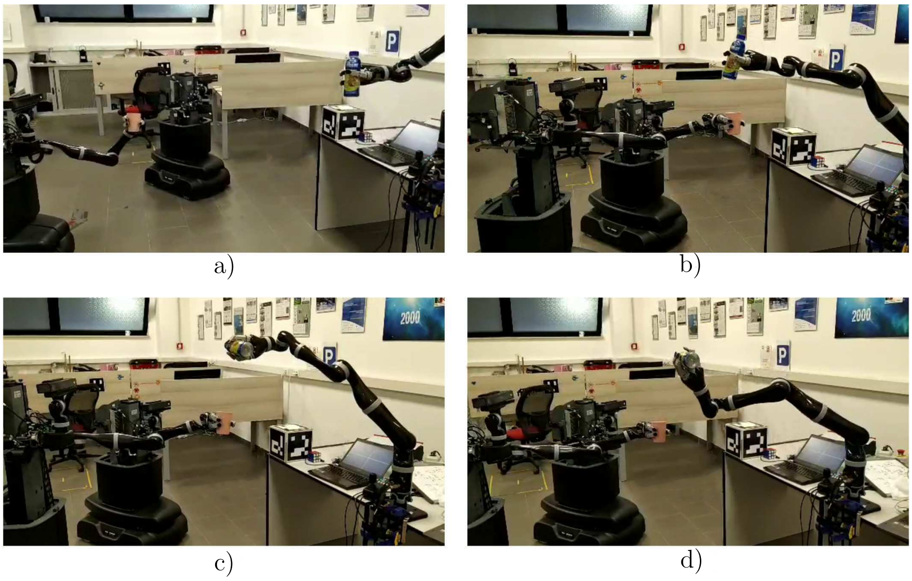


Figure 2.9: Snapshots of the experiment. In detail, a) represents the starting configuration, b) shows the positioner hanging out the glass, c) is the pouring phase and d) represents the fault occurrence.

A fault occurs on the fixed-base manipulator (with index $i = 1$) during last phase. In particular, the following fault term is introduced at time $t \approx 35$ s

$$\boldsymbol{f}_1 = \boldsymbol{J}_{M,1}^{\dagger}\boldsymbol{\psi} \qquad \text{with} \quad \boldsymbol{\psi} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^T \text{m/s}^2 \tag{2.41}$$

which induces a downwards motion of the manipulator's end effector as shown in Figure 2.9.d.

As for the simulation case study, it is considered that, when a fault is detected, a shutdown procedure is activated.

Figure 2.10 reports the norm of the residuals computed by all the robots and associated with the faulty robot 1 (solid lines), i.e., $\|{}^i\boldsymbol{r}_1\|$, $\forall\, i$, compared with the respective adaptive thresholds (dashed lines), i.e., $\|{}^i\underline{\mu}_1\|$ $\forall\, i$, which are computed according to (2.38) by considering $\bar{\boldsymbol{d}} = 0.1$ and, without loss of generality,

$\|\tilde{\boldsymbol{v}}_k^\star(0)\| = 0$. Hence, the figure shows that all the robots are able to detect the occurrence of the fault without the need for direct communication with the faulty robot, that is all the robots verify the condition $\|{}^i\boldsymbol{r}_1\| \geq {}^i\mu_1 \; \forall i$ after the fault occurrence. In addition, the norms of the residuals associated with the healthy robot 2 (solid lines) are reported in Figure 2.11. The respective adaptive thresholds are also reported (dashed lines) which are shown to always be greater than the residual signals; this verifies that the fault of one robot does not affect residuals of the other teammates. Analogous norms are obtained for the residuals of robot 3 which are not reported here for the sake of brevity.

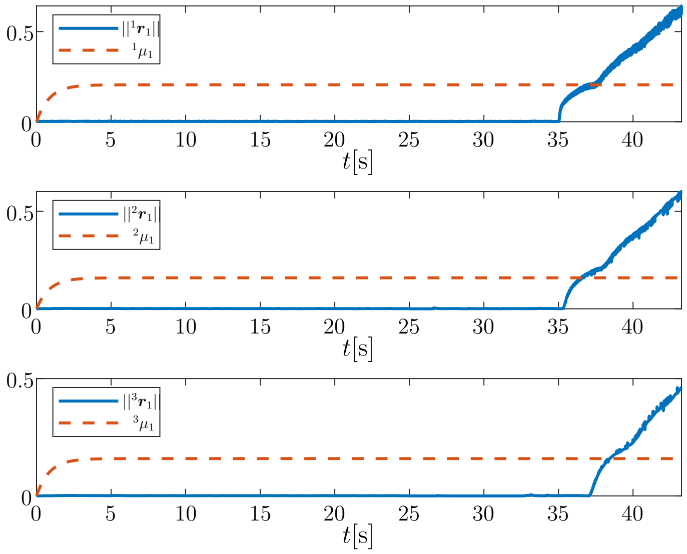


Figure 2.10: Evolution of the norm of residual components relative to robot 1 (faulty robot) computed by the 3 robots (solid lines) compared with the respective adaptive thresholds (dashed lines).

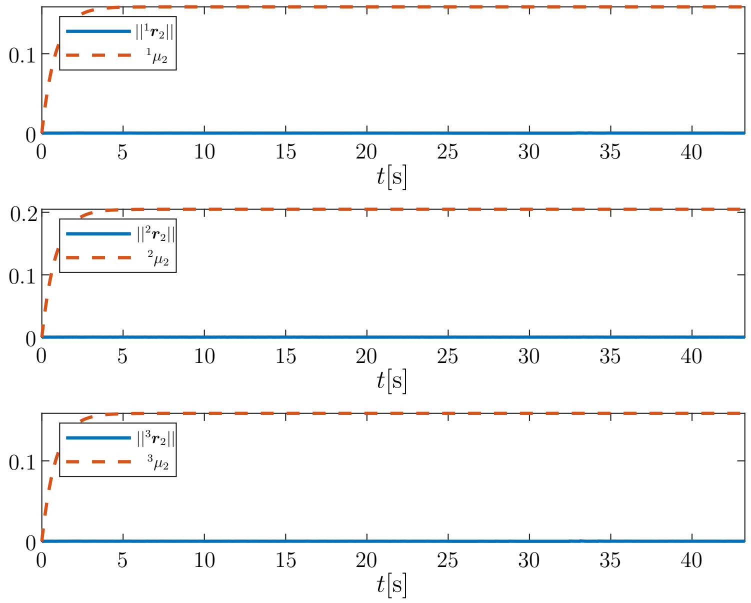


Figure 2.11: Evolution of the norm of residual components relative to robot 2 (healthy robot) computed by the 3 robots (solid lines) compared with the respective adaptive thresholds (dashed lines).

Chapter 3

Human multi-robot workspace sharing

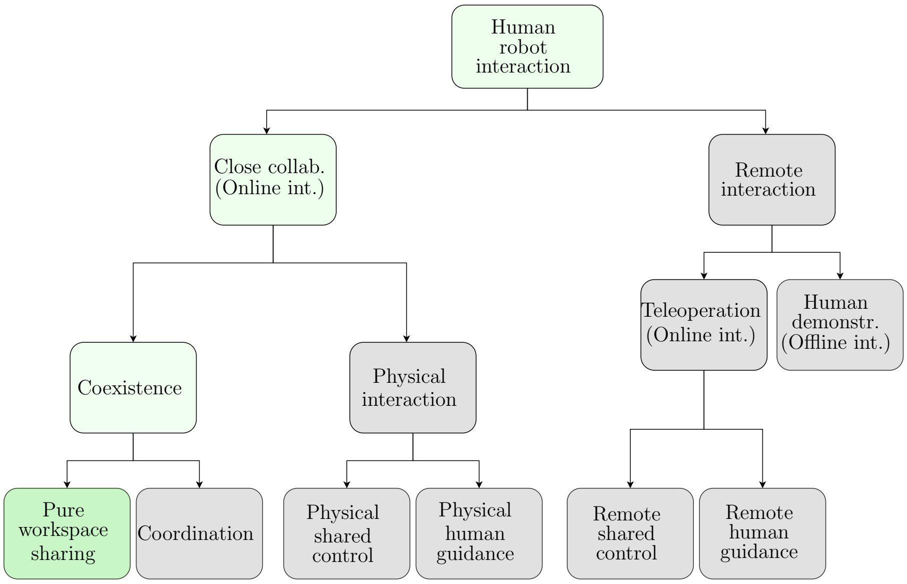


Figure 3.1: Taxonomy for HRI scenarios where the scenario considered in this chapter is highlighted in green.

In this chapter, a human multi-robot coexistence scenario is considered in which *pure workspace sharing* occurs, highlighted in green in the taxonomy Figure 3.1. In this scenario, depicted in Figure 3.2, humans and robots work side-by-side on independent tasks. As discussed in the Introduction, this implies the need to guarantee the primary requirement of human safety at all times and, thus, to devise real-time replanning strategies that allow to dynamically adapt the robots' behavior depending on the human one to prevent unsafe contacts. Indeed, since in this scenario independent tasks are carried out by humans and robots, the human safety requirement translates into that of *human avoidance*.

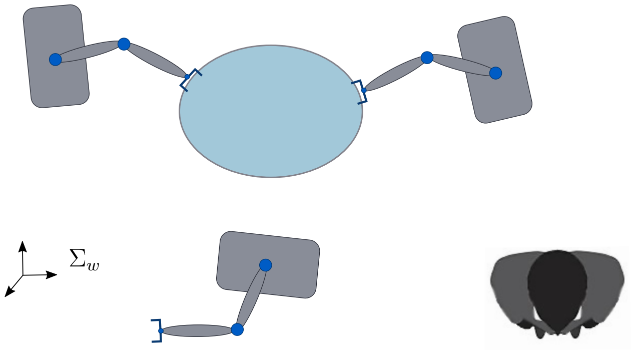


Figure 3.2: Representation of the system in a human multi-robot workspace sharing setup. The system is composed of multiple collaborative robots and human operator. The world reference frame Σ_w is reported.

Obviously, depending on the constraints of the specific work context, different strategies, based on various methodologies and rationales, can be developed to ensure human avoidance. In particular, in the case the robots are not strictly required to follow a planned path to carry out their task, gradient-based techniques can be adopted to adjust the trajectory and drive the manipulators away from the human operators reducing the probability of collision. In contrast, when the robots' task requires the motion along a specific path to be accomplished, the replanning strategy should avoid altering the path to ensure human safety, unless strictly necessary. In this case, in order not to violate the *path preservation constraint*, trajectory scaling approaches can be leveraged that only modulate the robots' velocity along the planned path without modifying the latter. Note that this requirement of path preserving is rather common in industrial settings where it is generally recommended to follow the desired path without deviating from it in order to accomplish the robotic tasks, such as in welding or assembly tasks.

In addition, when multiple manipulators are involved in the system and work in presence of human operators, the robotic strategy also needs to handle any possible constraint that the multi-robot system itself introduces. For example, in the case of cooperative manipulation tasks, the robots cannot move independently from each other when performing actions for human avoidance, but their motions are coupled and kinematically constrained. This coordination is even more challenging when a distributed architecture is required that allows, as discussed in the Chapter 2, to improve flexibility and reconfigurability of the work environment at the expense of higher control complexity, due to the lack of global knowledge of the system.

The focus of this Chapter is thus to present a general architecture, incrementally built in [58], [59] and [16], for performing cooperative tasks with multiple robots working in environments shared with human operators. In detail, a cumulative *safety index* is defined which takes into account the relative motion considering both position and velocity between the human operator and each point of each manipulator in the team. Then, a control strategy based on trajectory scaling is presented to modulate the velocity of the cooperative task preventing the safety index from falling below a time-varying minimum threshold; however, in the case velocity modulation is not sufficient to guarantee the minimum safety index, an emergency procedure is envisaged which relaxes the robots cooperative task. The proposed strategy is finally extended to the decentralized case according to a leader-follower paradigm and is validated both in a simulation and a real-world setup.

The rest of the Chapter is structured as follows: the main methodologies for safe workspace sharing in the state-of-the-art are summarized in Section 3.1, the safety index formulation is devised in Section 3.2 and, building on this, a solution for multi-robot trajectory scaling, also characterized by a distributed implementation, is presented in Section 3.3, finally, simulation and experimental results to validate the proposed approach are reported in Sections 3.4 and 3.5, respectively.

3.1 Literature review

In case the robotic task does not require to strictly follow a planned path to be realized, evasive actions that move the manipulators in the opposite direction from the human operators can be generally adopted. In this regard, a first possible approach consists in defining a measure which quantifies the level of human danger and then in carrying out appropriate evasive actions to minimize it. This idea is, for example, pursued in [60] where the concept of *Danger Field* is introduced. More specifically, first, an index is defined which assesses the level of human danger with respect to a point-mass robot on the basis of both relative human-robot distance and point-mass robot velocity; then, this index is integrated along the overall structure of the manipulator and is adopted to define joint space velocity commands used to generate internal motions (in the case of redundant robots) and to give a reactive evasive behavior to the robot. In this context, on the basis of the concept of danger field introduced in [61] for a point-mass robot, the study in [62] considers the different factors that affect the impact force during a potential collision as a measure of danger: human-point distance, human-robot relative velocity and robot inertia. Then, this index is evaluated for each *critical point* which is the closest point to the operator for each link, and is exploited in the definition of a virtual force according to the impedance paradigm which drives the robot away from the operator. The work in [63] focuses, instead, on the way the perception of the environment is performed; indeed, a method based on depth data is defined to fast compute the distances between the manipulator and possible obstacles in the workspace. Information regarding these distances and obstacles velocities are then exploited to generate repulsive velocity commands in the Cartesian space according to the artificial potential field theory [64].

Concerning the case of tasks with path preservation constraints, the study in [17] proposes a scaling procedure where only velocity reduction along the nominal path is allowed. In detail, a constrained optimization problem is solved to minimize the slowing down of the trajectory, while guaranteeing a condition on the minimum distance between each point of the robot and the human operator. However, the proposed solution does not account for the human-robot relative motion, but only considers the robot velocity in the direction of the human operator; more specifically, only the maximum value of the human operator velocity can be included in a clearance parameter but no measure of his/her actual velocity in the direction of the robot is explicitly considered. The work in [65] proposes a trajectory scaling method which relies on the separation distance between the operator and the robot without considering where their motion is directed: if the separation distance is above a safe value, no velocity reduction is applied, if it is less than a dangerous value, zero velocity is imposed or, otherwise, a velocity reduction factor inversely proportional to the distance is considered. The study in [66] presents an approach based on dynamical systems which combines a velocity reduction action with a path reshaping one; more specifically, the former is based on the human-robot distance and is higher as the distance decreases, while the latter is such as to reduce the velocity in the plane normal to the human surface and to project the motion into the one tangent to the surface, thus avoiding the surface collision. A trajectory scaling approach for Dynamical Movement Primitives (DMPs), successfully applied in many learning from demonstration contexts, is then provided in [67] where velocity constraints are taken into account.

In addition, in order for the robot to proactively react to the human behavior, a prediction of his/her intentional motion [68] can be involved in the avoidance strategy as proposed in [69] where the human workspace occupancy is predicted and exploited by a trajectory planner to minimize the penetration cost in it. Finally, a hybrid solution that addresses both workspace sharing and physical interaction can be found in [70], where a reactive controller is proposed that responds to both virtual and real contact forces.

Note that, although great effort has been dedicated by the research community to the topic of human-robot safe workspace sharing, none of the mentioned contributions deal with multiple manipulator systems working in presence of human operators for which, as stated above, additional constraints need to be taken into account. In the following sections, a formulation and a solution for the latter scenario is provided.

3.2 Assessment of human safety

In order to devise a strategy for enabling safe human multi-robot workspace sharing, an index is first introduced to asses the human safety with respect to the team of robots. More specifically, as shown below, it is built in an incremental way starting from individual points on the robotic structures and gradually extending it to consider all the robots in the team and all the points associated with human operators. Given the overall human safety index, it is therefore possible to formulate the problem of safe coexistence as a problem of maintaining the safety index above a certain minimum threshold.
The definition of a safety index for a single robot point and a single human point is first considered. By drawing on the idea of danger field in [60], the following scalar index is defined that quantifies the level of human safety with respect to a generic point P of a robot structure and a point P_o of the human operator

$$f(\boldsymbol{p}, \dot{\boldsymbol{p}}, \boldsymbol{p}_o, \dot{\boldsymbol{p}}_o) = \alpha_1(d) + \alpha_2(d, \dot{d}) \tag{3.1}$$

where $\boldsymbol{p}\,(\dot{\boldsymbol{p}}) \in \mathbb{R}^3$ denotes the position (velocity) of the robot point P, $\boldsymbol{p}_o\,(\dot{\boldsymbol{p}}_o) \in \mathbb{R}^3$ represents the position (velocity) of the human point P_o and d the distance between these two points, (i.e., $d = \|\boldsymbol{p} - \boldsymbol{p}_o\|$).

Functions α_1 and α_2 are generic scalar functions which meet the following properties.

Property 1. $\alpha_1(d)$ is a continuous monotonically increasing Lipschitz function with respect to d and with Lipschitz constant k_L;

Property 2. $\alpha_2(d, \dot{d})$ is a continuous monotonically increasing function with respect to $\dot{d}$ and such that:

(a) $\lim\limits_{\dot{d} \to +\infty} \alpha_2(d, \dot{d}) = k$, $\forall d$ with $k \in \mathbb{R}^+$;

(b) $\frac{\partial \alpha_2(d,\dot{d})}{\partial \dot{d}} \neq 0$ $\forall d$ and $\forall \dot{d} \neq \infty$.

In detail, Property 1 basically states that the level of human safety increases with the distance d, whereas Property 2 states that function α_2 is such that the safety index increases with $\dot{d}$ with a slope that, if required, might be modulated by the distance d. Moreover, Property 2(a) implies that there exists an asymptotic bound k for $\dot{d} \to +\infty$ so as to prevent the safety index to reach a too high value for high values of $\dot{d}$ with arbitrarily small values of the distance d; this feature makes the distance d the highest priority parameter affecting the level of safety. Finally, Property 2(b) states that, in practical conditions of $\dot{d} \neq \infty$, function f is sensitive to variation of $\dot{d}$, i.e. by changing $\dot{d}$ the value of f can be modified. Multiple relevant points associated with the human operator are now considered, for example the end points of the links composing the human skeleton [71]. By following the idea presented in [60], the cumulative safety index associated with the generic manipulator i is obtained by integrating the function in (3.1) along the robot structure and, then, by evaluating it for each human point. To this aim, the measure of the human safety with respect to the l th link of the i th robot is computed by integrating f along the volume $\mathcal{V}_l$ of link l, that is

$$F_{i,l} = \int_{\mathcal{V}_l} f(\boldsymbol{p}, \dot{\boldsymbol{p}}, \boldsymbol{p}_o, \dot{\boldsymbol{p}}_o)\, dV. \tag{3.2}$$

However, in order to make the computation of (3.2) affordable in practical applications, each link is simplified as a segment starting at $\boldsymbol{p}^0_{i,l}$ and ending at $\boldsymbol{p}^1_{i,l}$ such that (3.2) becomes

$$\begin{cases} F_{i,l} = \displaystyle\int_0^1 f(\boldsymbol{p}^s_{i,l}, \dot{\boldsymbol{p}}^s_{i,l}, \boldsymbol{p}_o, \dot{\boldsymbol{p}}_o) ds \\ \boldsymbol{p}^s_{i,l} = \boldsymbol{p}^0_{i,l} + (\boldsymbol{p}^1_{i,l} - \boldsymbol{p}^0_{i,l}) s \\ \dot{\boldsymbol{p}}^s_{i,l} = \dot{\boldsymbol{p}}^0_{i,l} + (\dot{\boldsymbol{p}}^1_{i,l} - \dot{\boldsymbol{p}}^0_{i,l}) s \end{cases} \tag{3.3}$$

where $s \in [0,\, 1]$ is the segment parameter. At this point, let n_o denote the total number of human points and $\boldsymbol{p}_{o,j}$ the position of the j th one. The safety index associated with the j th human point with respect to the i th manipulator can be easily defined from (3.3) as

$$\bar{F}_{i,j} = \sum_{l=1}^{n_i} F_{i,l}(\boldsymbol{p}^0_{i,l}, \boldsymbol{p}^1_{i,l}, \dot{\boldsymbol{p}}^0_{i,l}, \dot{\boldsymbol{p}}^1_{i,l}, \boldsymbol{p}_{o,j}, \dot{\boldsymbol{p}}_{o,j}) \tag{3.4}$$

being n_i the number of links of the i th robot. By considering all the human points, the cumulative safety index associated with robot i can be derived

$$\bar{F}_i = \frac{1}{n_o} \sum_{j=1}^{n_o} \bar{F}_{i,j}(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i, \boldsymbol{p}_{o,j}, \dot{\boldsymbol{p}}_{o,j}) \tag{3.5}$$

which is finally extended to the team of robots as follows

$$\bar{F} = \sum_{i=1}^{N} \bar{F}_i, \quad \dot{\bar{F}} = \sum_{i=1}^{N} \dot{\bar{F}}_i. \tag{3.6}$$

In the following, the cumulative safety index $\bar{F}$ will be also referred to as cumulative safety field.

Remark 3.1. *Note that the approach is not limited to the single operator case, but the multi-human case is straightforward to tackle by considering that the n_o points in* (3.5) *might belong to different human operators.*

The main problem addressed for the safe human multi-robot workspace sharing is the following.

Problem 3.1. *Consider a multi-robot system composed of N mobile manipulators performing a cooperative task expressed by means of a task function $\boldsymbol{\sigma}(\boldsymbol{x})$ defined as in* (2.13) *and for which a desired trajectory $\boldsymbol{\sigma}_n(t)$ is assigned. Moreover, assume that a time-varying minimum function $\bar{F}_{min}(t)$ for the safety index $\bar{F}(t)$ in* (3.6) *is assigned. The objective is to properly scale the task trajectory $\boldsymbol{\sigma}_n(t)$ in order to generate a reference trajectory $\boldsymbol{\sigma}_r(t)$ which ensures the safety condition*

$$\bar{F}(t) \geq \bar{F}_{min}(t), \ \ \forall t \tag{3.7}$$

to be always satisfied.

The problem formulation requires the definition of the time-varying threshold $\bar{F}_{min}(t)$. As proved in Appendix B, a possible choice is to ensure a minimum safety distance between the human operator and any robot of the team, i.e., such that the condition $\bar{F} \geq \bar{F}_{min}$ also guarantees that the distance $d^* \geq d_{min}$ being d^* defined as

$$d^* = \min_{\forall i, l, s, j} \|\boldsymbol{p}^s_{i,l} - \boldsymbol{p}_{o,j}\| \tag{3.8}$$

and d_{min} a positive constant (see Appendix B for further details). In addition, a further possibility consists in experimentally calibrating the threshold by appropriately measuring the *experienced* minimum safety level; to this aim, as instance, the human stress may be evaluated from the analysis of the heart rate variability, i.e., the variation over time of the interval between consecutive heart beats, or any other combination of biometric parameters [72].

3.3 Trajectory scaling approach

In this section, a solution for Problem 3.1 is devised. The proposed approach is schematically summarized in Figure 3.3 and the basic idea is that, as long as the safety index $\bar{F}(t)$ is greater than the allowed minimum value $\bar{F}_{min}(t)$, the nominal trajectory is tracked, whereas if the minimum value is reached, a scaling procedure is applied in order to modulate the trajectory velocity without deviating from the nominal path. Then, if the safety condition is restored ($\bar{F}(t) > \bar{F}_{min}(t)$), the nominal trajectory is recovered whereas, if this modulation is not sufficient to guarantee the safety condition in (3.7), any emergency procedure is activated, as it will be shown in Sections 3.4 and 3.5.

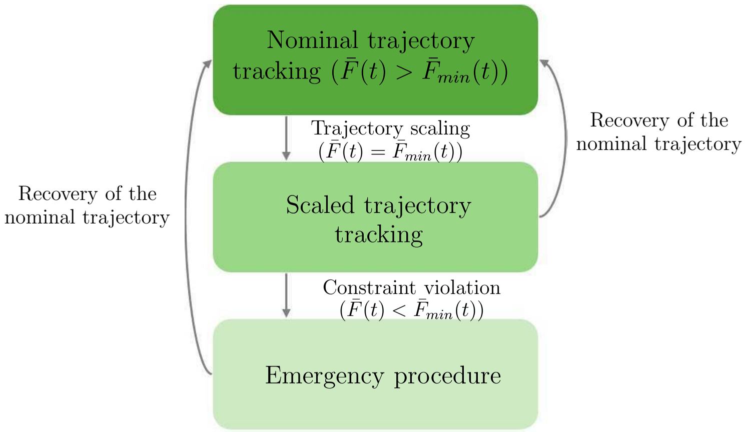

Nominal trajectory tracking ($\bar{F}(t) > \bar{F}_{min}(t)$)
Trajectory scaling ($\bar{F}(t) = \bar{F}_{min}(t)$)
Recovery of the nominal trajectory
Scaled trajectory tracking
Recovery of the nominal trajectory
Constraint violation ($\bar{F}(t) < \bar{F}_{min}(t)$)
Emergency procedure

Figure 3.3: Overview of the strategy adopted for the human multi-robot avoidance; transition conditions are detailed in Section 3.3.2.

More in detail, let $\boldsymbol{\sigma}_n(t)$ ($\dot{\boldsymbol{\sigma}}_n(t)$, $\ddot{\boldsymbol{\sigma}}_n(t)$) be the nominal trajectory corresponding to the task variables in (2.13), the scaling procedure is based on the introduction of a monotonically increasing scalar function of time $c(t)$

$$c : [t_0, t_f] \in \mathbb{R} \rightarrow [t_0, t_f] \in \mathbb{R} \tag{3.9}$$

being t_0 and t_f the starting and final time instants of the nominal trajectory, respectively, and on the definition of a reference trajectory obtained as the parametrization of the nominal one $\boldsymbol{\sigma}_n(t)$

with respect to the time parameter $c(t)$, that is

$$\begin{cases} \boldsymbol{\sigma}_r(t) = \boldsymbol{\sigma}_n(c(t)) \\ \dot{\boldsymbol{\sigma}}_r(t) = \dfrac{\partial \boldsymbol{\sigma}_n(c(t))}{\partial c}\dot{c}(t) \\ \ddot{\boldsymbol{\sigma}}_r(t) = \dfrac{\partial^2 \boldsymbol{\sigma}_n(c(t))}{\partial c^2}\dot{c}^2(t) + \dfrac{\partial \boldsymbol{\sigma}_n(c(t))}{\partial c}\ddot{c}(t). \end{cases} \tag{3.10}$$

Note that the reference trajectory is then the one actually tracked by the robots. The result of this approach is that the path of the nominal task trajectory is preserved while its time dependence can be modified; this means that, by modulating the scaling parameter $c(t)$, the nominal trajectory $\boldsymbol{\sigma}_n(t)$ may be online scaled along the nominal path depending on the relative motion of the human operator and the team of robots, namely on the safety index $\bar{F}(t)$. Furthermore, by virtue of the definition in (3.9) and since $c(t)$ is monotonically increasing, the following conditions on the scaling parameters must always be met

$$\dot{c}(t) \geq 0 \tag{3.11}$$

$$c(t) \leq t_f \tag{3.12}$$

where (3.11) implies that no reverse motion is allowed along the path, while (3.12) means that, in the case of non-periodic trajectories, the ending point must not be overcome.
Note that the presented solution assumes that either robots are able to detect and locate the human operator or an external vision system, as in the experimental setup in Section 3.5, makes this information available to robots, resorting to approaches like [73] for example.

The rest of this section is structured as follows. First, the low level input of each robot to track the cooperative reference trajectory $\boldsymbol{\sigma}_r(t)$ is introduced and, then, a centralized solution to Problem 3.1, based on the strategy in Figure 3.3, is proposed which is finally extended to the decentralized framework.

3.3.1 Robot low-level control

The focus is here on defining the robot low-level control to track a cooperative reference trajectory $\boldsymbol{\sigma}_r(t)$ obtained as in (3.10). Let $\boldsymbol{q}_{r,i}(t) \in \mathbb{R}^{n_i}$ ($\dot{\boldsymbol{q}}_{r,i}(t)$, $\ddot{\boldsymbol{q}}_{r,i}(t)$) be the reference joint position (velocity, acceleration) of robot i, the following assumption is made in the rest of the chapter.

Assumption 3.1. *Each robot is equipped with an inner motion control loop which guarantees tracking of a reference joint trajectory, i.e.,* $\boldsymbol{q}_{r,i} \approx \boldsymbol{q}_i$ *(*$\dot{\boldsymbol{q}}_{r,i} \approx \dot{\boldsymbol{q}}_i$, $\ddot{\boldsymbol{q}}_{r,i} \approx \ddot{\boldsymbol{q}}_i$*).*

This assumption is realistic when dealing with commercial platforms and makes the devised solution suitable also for off-the-shelf robotic platforms for which the input $\boldsymbol{\tau}_i$ in (2.6) is typically not directly accessible and only joint or end effector references are allowed.

Based on the second order kinematic equation in (2.3), the following virtual model is considered

$$\ddot{\boldsymbol{x}}_i = \boldsymbol{J}_i(\boldsymbol{q}_i)\boldsymbol{y}_i + \dot{\boldsymbol{J}}(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i)\dot{\boldsymbol{q}}_i \tag{3.13}$$

where $\boldsymbol{y}_i = \ddot{\boldsymbol{q}}_i$ is the input of the assumed virtual model to be designed so as to track the cooperative reference trajectory $\boldsymbol{\sigma}_r(t)$. As in (2.40), the input $\boldsymbol{y}_i$ is designed by adopting a standard closed loop inverse kinematic law [74] as follows

$$\boldsymbol{y}_i = \boldsymbol{J}_i^{\dagger}\left[\boldsymbol{\Gamma}_i\boldsymbol{J}_{\sigma}^{\dagger}\left(\ddot{\boldsymbol{\sigma}}_r + k_{\sigma,d}\dot{\tilde{\boldsymbol{\sigma}}} + k_{\sigma,p}\tilde{\boldsymbol{\sigma}}\right) - \dot{\boldsymbol{J}}_i\dot{\boldsymbol{q}}_i\right] + \ddot{\boldsymbol{q}}_{n,i} \tag{3.14}$$

where $\tilde{\boldsymbol{\sigma}}(t) = (\boldsymbol{\sigma}_r(t) - \boldsymbol{\sigma}(\boldsymbol{x}(t))) \in \mathbb{R}^m$ is the task tracking error, $\ddot{\boldsymbol{q}}_{n,i} \in \mathbb{R}^{n_i}$ is an arbitrary vector of joint accelerations such that $\boldsymbol{J}_i(\boldsymbol{q}_i)\ddot{\boldsymbol{q}}_{n,i} = \boldsymbol{0}_p$ which might be exploited to locally increase the safety field, $k_{\sigma,d}$, $k_{\sigma,p}$ are positive gains and $\boldsymbol{\Gamma}_i \in \mathbb{R}^{p\times Np}$ is a selection matrix defined as in (2.24). By recalling the kinematic equation (2.3) and the task formulation (2.13), it easily follows that

$$\boldsymbol{J}_{\sigma}(\boldsymbol{J}\boldsymbol{y} + \dot{\boldsymbol{J}}\dot{\boldsymbol{q}}) = \boldsymbol{J}_{\sigma}\ddot{\boldsymbol{x}} = \ddot{\boldsymbol{\sigma}} = \ddot{\boldsymbol{\sigma}}_r + k_{\sigma,d}\dot{\tilde{\boldsymbol{\sigma}}} + k_{\sigma,p}\tilde{\boldsymbol{\sigma}}$$

with $\boldsymbol{y} = \left[\boldsymbol{y}_1^T, \ldots, \boldsymbol{y}_N^T\right]^T \in \mathbb{R}^n$, which finally leads to the following exponentially stable linear second order dynamics

$$\ddot{\tilde{\boldsymbol{\sigma}}} + k_{\sigma,d}\dot{\tilde{\boldsymbol{\sigma}}} + k_{\sigma,p}\tilde{\boldsymbol{\sigma}} = \mathbf{0}_m.$$

3.3.2 Centralized solution

In this section, a centralized solution to Problem 3.1 is provided which is based on the strategy in Figure 3.3. The team of robots is thus coordinated by a central unit having access to all the information of the system and the objective is to determine how the scaling procedure is performed in this case, i.e., how to compute the scaling parameter $c(t)$. For this purpose, by considering the virtual input $\boldsymbol{y}_i$ in (3.14), the time derivative of the safety field associated with robot i in (3.5) can be shown to be linear with respect to the scaling parameter $\ddot{c}(t)$ and this property will be, then, exploited to implement the scaling procedure. In detail, the following lemma holds true.

Lemma 3.1. *The derivative of the cumulative safety function* (3.5) *associated with the i th robot is linear with respect to the scaling parameter $\ddot{c}(t)$, i.e., it holds*

$$\dot{\bar{F}}_i(t) = \mu_{1,i}(t)\,\ddot{c}(t) + \mu_{2,i}(t) \tag{3.15}$$

where $\mu_{1,i}(t)$, $\mu_{2,i}(t)$ are scalar functions whose expressions are provided in the proof.

Proof. In order to prove the result, the time derivative of (3.1) is first considered and the linear dependence of $\dot{f}(t)$ with respect to the scaling parameter $\ddot{c}(t)$ is shown. Then, the result is extended to $\dot{\bar{F}}_i$. Based on (3.3), $\dot{f}$ can be computed as follows

$$\dot{f} = \left(\frac{\partial\alpha_1(d_{i,l}^s)}{\partial d_{i,l}^s} + \frac{\partial\alpha_2(d_{i,l}^s, \dot{d}_{i,l}^s)}{\partial d_{i,l}^s}\right)\dot{d}_{i,l}^s + \frac{\partial\alpha_2(d_{i,l}^s, \dot{d}_{i,l}^s)}{\partial \dot{d}_{i,l}^s}\ddot{d}_{i,l}^s \tag{3.16}$$

being $d_{i,l}^s = \|\boldsymbol{p}_{i,l}^s - \boldsymbol{p}_o\|$, i.e., the distance between the point $\boldsymbol{p}_{i,l}^s$ and the human operator $\boldsymbol{p}_o$. By differentiating the distance term, one obtains

$$\dot{d}^s_{i,l} = \frac{(\boldsymbol{p}^s_{i,l} - \boldsymbol{p}_o)^T(\dot{\boldsymbol{p}}^s_{i,l} - \dot{\boldsymbol{p}}_o)}{d^s_{i,l}} \tag{3.17}$$

with $d^s_{i,l} \neq 0$. Therefore, by further differentiating (3.17) and by introducing the following coefficients

$$\begin{cases} \boldsymbol{\beta}_1 = \dfrac{\boldsymbol{p}^s_{i,l} - \boldsymbol{p}_o}{d^s_{i,l}} & d^s_{i,l} \neq 0 \\ \beta_2 = -\boldsymbol{\beta}_1^T \ddot{\boldsymbol{p}}_o + \dfrac{\|\dot{\boldsymbol{p}}^s_{i,l} - \dot{\boldsymbol{p}}_o\|^2}{d^s_{i,l}} - \dfrac{[\boldsymbol{\beta}_1^T(\dot{\boldsymbol{p}}^s_{i,l} - \dot{\boldsymbol{p}}_o)]^2}{d^s_{i,l}} & d^s_{i,l} \neq 0 \end{cases}$$

$\ddot{d}^s_{i,l}$ can be expressed as

$$\ddot{d}^s_{i,l} = \boldsymbol{\beta}_1^T \ddot{\boldsymbol{p}}^s_{i,l} + \beta_2 \tag{3.18}$$

which is linear in the acceleration vector $\ddot{\boldsymbol{p}}^s_{i,l}$. Now, the relation between the linear acceleration of the point $\boldsymbol{p}^s_{i,l}$ and the joints variables of the same robot i is considered, that is

$$\ddot{\boldsymbol{p}}^s_{i,l} = \boldsymbol{J}^s_{i,l}(\boldsymbol{q}_i)\ddot{\boldsymbol{q}}_i + \dot{\boldsymbol{J}}^s_{i,l}(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i)\dot{\boldsymbol{q}}_i \tag{3.19}$$

where $\boldsymbol{J}^s_{i,l} \in \mathbb{R}^{3\times n_i}$ is the positional Jacobian matrix associated with $\boldsymbol{p}^s_{i,l}$. By taking into account the virtual input in (3.14), (3.19) can be rewritten as

$$\ddot{\boldsymbol{p}}^s_{i,l} = \boldsymbol{J}^s_{i,l}\boldsymbol{J}^\dagger_i \left[\boldsymbol{\Gamma}_i \boldsymbol{J}^\dagger_\sigma \left(\ddot{\boldsymbol{\sigma}}_r + k_{\sigma,d}\dot{\tilde{\boldsymbol{\sigma}}} + k_{\sigma,p}\tilde{\boldsymbol{\sigma}}\right) - \dot{\boldsymbol{J}}_i\dot{\boldsymbol{q}}_i\right] + \boldsymbol{J}^s_{i,l}\ddot{\boldsymbol{q}}_{n,i} + \dot{\boldsymbol{J}}^s_{i,l}\dot{\boldsymbol{q}}_i \tag{3.20}$$

which, in view of the reference trajectory in (3.10), leads to

$$\ddot{\boldsymbol{p}}^s_{i,l} = \boldsymbol{\gamma}_1 \ddot{c} + \boldsymbol{\gamma}_2 \tag{3.21}$$

that is linear with respect to $\ddot{c}$ and where the coefficients $\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2 \in$

$\mathbb{R}^3$ are defined as follows

$$\begin{cases} \boldsymbol{\gamma}_1 = \boldsymbol{J}^s_{i,l} \boldsymbol{J}_i^\dagger \boldsymbol{\Gamma}_i \boldsymbol{J}_\sigma^\dagger \dfrac{\partial \boldsymbol{\sigma}_r}{\partial c} \\ \boldsymbol{\gamma}_2 = \boldsymbol{J}^s_{i,l} \left[\boldsymbol{J}_i^\dagger \boldsymbol{\Gamma}_i \boldsymbol{J}_\sigma^\dagger \Big(\dfrac{\partial^2 \boldsymbol{\sigma}_r}{\partial c^2} \dot{c}^2 + k_{\sigma,d} \dot{\tilde{\boldsymbol{\sigma}}} + k_{\sigma,p} \tilde{\boldsymbol{\sigma}} \Big) - \boldsymbol{J}_i^\dagger \dot{\boldsymbol{J}}_i \dot{\boldsymbol{q}}_i \right] \\ \quad + \boldsymbol{J}^s_{i,l} \ddot{\boldsymbol{q}}_{n,i} + \dot{\boldsymbol{J}}^s_{i,l} \dot{\boldsymbol{q}}_i. \end{cases}$$

By replacing (3.18) and (3.21) in (3.16), the derivative of the point safety index f can be reformulated as

$$\dot{f} = \lambda_1 \ddot{c} + \lambda_2 \tag{3.22}$$

where the expressions of $\lambda_1 \in \mathbb{R}$ and $\lambda_2 \in \mathbb{R}$ are

$$\begin{cases} \lambda_1 = (\boldsymbol{\beta}_1^T \boldsymbol{\gamma}_1) \dfrac{\partial \alpha_2}{\partial \dot{d}^s_{i,l}} \\ \lambda_2 = (\boldsymbol{\beta}_1^T \boldsymbol{\gamma}_2 + \beta_2) \dfrac{\partial \alpha_2}{\partial \dot{d}^s_{i,l}} + \left(\dfrac{\partial \alpha_1}{\partial d^s_{i,l}} + \dfrac{\partial \alpha_2}{\partial d^s_{i,l}} \right) \dot{d}^s_{i,l}. \end{cases}$$

Therefore, in view of the integration defined in Section 3.2, by extending the point safety index to the entire structure of the i th manipulator and by considering all the human relevant points, it finally holds

$$\dot{\bar{F}}_i = \mu_{1,i} \, \ddot{c} + \mu_{2,i}$$

with $\mu_{1,i} \in \mathbb{R}$ and $\mu_{2,i} \in \mathbb{R}$ defined as

$$\begin{cases} \mu_{1,i} = \dfrac{1}{n_o} \displaystyle\sum_{j=1}^{n_o} \sum_{l=1}^{n_i} \int_0^1 \lambda_1 (\boldsymbol{p}^s_{i,l}, \dot{\boldsymbol{p}}^s_{i,l}, \boldsymbol{p}_{o,j}, \dot{\boldsymbol{p}}_{o,j}, \boldsymbol{q}_i, \dot{\boldsymbol{q}}_i, c) \; ds \\ \mu_{2,i} = \dfrac{1}{n_o} \displaystyle\sum_{j=1}^{n_o} \sum_{l=1}^{n_i} \int_0^1 \lambda_2 (\boldsymbol{p}^s_{i,l}, \dot{\boldsymbol{p}}^s_{i,l}, \boldsymbol{p}_{o,j}, \dot{\boldsymbol{p}}_{o,j}, \ddot{\boldsymbol{p}}_{o,j}, \boldsymbol{q}_i, \dot{\boldsymbol{q}}_i, \ddot{\boldsymbol{q}}_{n,i}, c, \dot{c}) \; ds. \end{cases} \tag{3.23}$$

where the dependencies of λ_1 and λ_2 on their parameters are now made explicit for the sake of completeness. This completes the proof. ■

By virtue of Lemma 3.1, it follows that the time derivative of the cumulative safety index in (3.6), associated with the entire team, is linear with respect to $\ddot{c}(t)$, i.e., it holds

$$\dot{\bar{F}}(t) = \mu_1(t)\,\ddot{c}(t) + \mu_2(t) \tag{3.24}$$

with $\mu_1(t) = \sum_{i=1}^{N} \mu_{1,i}(t)$ and $\mu_2 = \sum_{i=1}^{N} \mu_{2,i}(t)$.

The scaling procedure proposed to solve Problem 3.1 can be now presented. Assume that the initial safety index meets the safety condition in (3.7), i.e., $\bar{F}(t_0) > \bar{F}_{min}(t_0)$, and let $\dot{\bar{F}}_{min}(t)$ be the time derivative of the minimum safety index. The idea behind the scaling procedure is that, once the safety index reaches the minimum value $\bar{F}_{min}(t)$, the parameter $c(t)$ is computed so as to guarantee that the condition $\bar{F}(t) = \bar{F}_{min}(t)$ holds. By considering the time derivative of the safety index, this implies that the condition $\dot{\bar{F}}(t) \geq \dot{\bar{F}}_{min}(t)$ has to be fulfilled when $\bar{F}(t) = \bar{F}_{min}(t)$. Thus, by leveraging the linear expression of $\bar{F}(t)$ with respect to $\ddot{c}(t)$ in (3.24), the lower ($\ddot{c}_{min}(t)$) and upper ($\ddot{c}_{max}(t)$) bounds on the scaling parameter $\ddot{c}(t)$ to meet the safety condition condition can be derived as follows

$$\ddot{c}_{max}(t) = \begin{cases} \frac{\dot{\bar{F}}_{min}(t) - \mu_2(t)}{\mu_1(t)}, & \mu_1(t) < 0 \wedge \bar{F}(t) = \bar{F}_{min}(t) \\ +\infty, & \text{otherwise} \end{cases} \tag{3.25}$$

and

$$\ddot{c}_{min}(t) = \begin{cases} \frac{\dot{\bar{F}}_{min}(t) - \mu_2(t)}{\mu_1(t)}, & \mu_1(t) > 0 \wedge \bar{F}(t) = \bar{F}_{min}(t) \\ -\infty, & \text{otherwise.} \end{cases} \tag{3.26}$$

The computed bounds are exploited to compute the scaling parameters at each time instant. More specifically, two update laws are proposed, that are

$$\begin{cases} \ddot{c}(t) = k_{c,d}\,(1 - \dot{c}(t)) \\ \ddot{c}(t) = \text{sat}\,(\ddot{c}(t), \ddot{c}_{min}(t), \ddot{c}_{max}(t)) \end{cases} \tag{3.27}$$

and

$$\begin{cases} \ddot{c}(t) = k_{c,d}\,(1 - \dot{c}(t)) + k_{c,p}\,(t - c(t)) \\ \ddot{c}(t) = \text{sat}\,(\ddot{c}(t), \ddot{c}_{min}(t), \ddot{c}_{max}(t)) \end{cases} \tag{3.28}$$

where $k_{c,p}$ and $k_{c,d}$ are positive gains that regulate the velocity of the trajectory recovery after the scaling phase and once the safety index is above the minimum threshold $\bar{F}_{min}$, $\text{sat}(x, \underline{x}, \overline{x})$ is any $\mathcal{C}^1$ saturation function saturating x in the range $[\underline{x}, \overline{x}]$. The initial conditions are set to $c(t_0) = t_0$, $\dot{c}(t_0) = 1$ and $\ddot{c}(t_0) = 0$.
The rationale behind (3.27) is that the first equation continuously brings $\dot{c}(t)$ and $\ddot{c}(t)$ to 1 and 0 respectively, while the second equation limits $\ddot{c}(t)$ in the range $[\ddot{c}_{min}, \ddot{c}_{max}]$. Thus, in nominal conditions and after a scaling phase, when no scaling is needed anymore to meet human safety requirements (and so infinite bounds on $\ddot{c}$ are obtained in (3.25) and (3.26), i.e., $\ddot{c}_{min} = -\infty$, $\ddot{c}_{max} = +\infty$), the scaling parameters tend to

$$\dot{c}(t) = 1, \quad \ddot{c}(t) = 0$$

which means that scaling parameter $c(t)$ evolves with the same variation as the time t and the nominal velocity is recovered with a rate depending on $k_{c,d}$, i.e., $\dot{\boldsymbol{\sigma}}_r(t) = \dot{\boldsymbol{\sigma}}_n(c(t))$. On the contrary, (3.28) implies that in nominal conditions the scaling parameters tend to

$$c(t) = t, \quad \dot{c}(t) = 1, \quad \ddot{c}(t) = 0$$

which means that both the nominal velocity and the nominal position are recovered with a rate depending on $k_{c,p}$ and $k_{c,d}$, i.e., $\boldsymbol{\sigma}_r(t) = \boldsymbol{\sigma}_n(c(t)) = \boldsymbol{\sigma}_n(t)$ and $\dot{\boldsymbol{\sigma}}_r(t) = \dot{\boldsymbol{\sigma}}_n(c(t)) = \dot{\boldsymbol{\sigma}}_n(t)$.
In conclusion, depending on whether or not the cooperative task requires both the nominal velocity and position to be restored af-

ter the scaling phase, the update law in (3.28) or (3.27) is selected, respectively.

Note that, as already stated at the beginning of the section, the scaling procedure does not generally ensure that the safety condition in (3.7) is always met since no scaling is allowed when it leads to violation of constraints in (3.11)-(3.12). As a consequence, an emergency procedure needs to be foreseen until the safety condition is restored as shown in Figure 3.3.

Finally, it is worth noticing that the scaling procedure might also be extended to take into account kinematic velocity, acceleration and jerk constraints. Indeed, as in [75] and [76], such constraints can be formulated in terms of acceleration constraints which are linear with respect to the scaling parameter $\ddot{c}(t)$. Thus, additional bounds for $\ddot{c}(t)$ can be derived as in the above and combined with (3.25) and (3.26) so as to deal with both kinematic and safety constraints where, clearly, the former have higher priority due to mechanical limits.

3.3.3 Decentralized extension

Centralized approaches as the one presented above are effective in terms of performance but, as shown in Figure 2.1, they require a central control unit and/or complete communication among all robots. However, as discussed in Chapter 2, limitations in physical realization and scalability of the system may arise from this requirement. For this reason, a decentralized extension of the devised methodology is developed in this section and the following problem, extending Problem 3.1, is formulated to the purpose.

Problem 3.2. *Consider a multi-robot system composed of N mobile manipulators performing a cooperative task expressed by means of a task function $\boldsymbol{\sigma}(\boldsymbol{x})$ defined as* (2.13) *and for which a desired trajectory $\boldsymbol{\sigma}_n(t)$ is assigned. Assume also that a central unit is not present and that each robot has access only to information coming from on-board sensors and from the set $\mathcal{N}_i$ of neighbors robots. Moreover, assume that a time-varying minimum function $\bar{F}_{min}(t)$ for the safety index $\bar{F}(t)$ in* (3.6) *is assigned. The objective is* *to properly scale the task trajectory $\boldsymbol{\sigma}_n(t)$ in order to generate a reference trajectory $\boldsymbol{\sigma}_r(t)$ which ensures the safety condition*

$$\bar{F}(t) \geq \bar{F}_{min}(t), \ \ \forall t \tag{3.29}$$

is always satisfied and so as to obtain results as close as possible to those of the centralized case.

In order to solve the above problem, the approach presented in Section 3.3.2 is extended to a distributed setting by integrating proper decentralized observers to counteract the lack of global information. A leader-follower paradigm is adopted where a leader robot performs the scaling procedure by computing the scaling parameters $c(t)$, $\dot{c}(t)$, $\ddot{c}(t)$, while followers estimate them to determine the reference trajectory $\boldsymbol{\sigma}_r(t)$. The distributed extension is detailed in the following.

The following assumptions are made in the chapter.

Assumption 3.2. *The human position, velocity and acceleration, namely the vectors $\boldsymbol{p}_o(t)$, $\dot{\boldsymbol{p}}_o(t)$, $\ddot{\boldsymbol{p}}_o(t)$, are known by each robot at each time. For this purpose, either each robot may be equipped with an adequate sensor system or appropriate distributed observers can be introduced to track this information which may be known only to some of the robot in the team [77].*

Assumption 3.3. *The communication graph $\mathcal{G}$ (introduced in Section 2.3) is undirected, i.e., all communication links are bidirectional.*

Assumption 3.4. *The nominal cooperative task trajectory $\boldsymbol{\sigma}_n(t)$ ($\dot{\boldsymbol{\sigma}}_n(t)$, $\ddot{\boldsymbol{\sigma}}_n(t)$) is known by each robot.*

This assumption is not restrictive given the considered framework; however, it can be easily overcome at the expense of transferring more data between the robots, i.e., in the case the nominal trajectory is known only by a non-empty subset of robots, the remaining ones can estimate this trajectory by means of an observer of dimension $2m$, as proposed for example in [78].

At this point, observe that, as already detailed in Section 3.3.2, in order for a leader robot to perform the scaling procedure and compute the scaling parameter $c(t)$, it is required to estimate both the cumulative safety index in (3.6) and the coefficients of its derivative in (3.24). Moreover, by looking at the virtual input of the i th robot in (3.14), it also follows that each robot requires the knowledge of the reference task trajectory $\boldsymbol{\sigma}_r(t)$ (and of $\dot{\boldsymbol{\sigma}}_r(t)$ and $\ddot{\boldsymbol{\sigma}}_r(t)$) as well as of the collective state of the multi-robot system $\boldsymbol{x}$ and its derivative $\dot{\boldsymbol{x}}$ in order to compute $\boldsymbol{\sigma}(\boldsymbol{x})$ and its derivative, respectively, according to (2.13). Thus, by appropriately estimating these quantities, the input $\boldsymbol{y}_i$ in (3.14) is adapted to the decentralized framework as follows

$$\begin{aligned}\boldsymbol{y}_i =& \boldsymbol{J}_i^{\dagger}\boldsymbol{\Gamma}_i\boldsymbol{J}_{\sigma}^{\dagger}\left({}^i\ddot{\hat{\boldsymbol{\sigma}}}_r + k_{\sigma,d}\left({}^i\dot{\hat{\boldsymbol{\sigma}}}_r - {}^i\dot{\hat{\boldsymbol{\sigma}}}\right) + k_{\sigma,p}\left({}^i\hat{\boldsymbol{\sigma}}_r - {}^i\hat{\boldsymbol{\sigma}}\right)\right)\\ &- \boldsymbol{J}_i^{\dagger}\dot{\boldsymbol{J}}_i\dot{\boldsymbol{q}}_i + \ddot{\boldsymbol{q}}_{n,i}\end{aligned} \tag{3.30}$$

where ${}^i\hat{\boldsymbol{\sigma}}_r$,${}^i\dot{\hat{\boldsymbol{\sigma}}}_r$, ${}^i\ddot{\hat{\boldsymbol{\sigma}}}_r$ are now the estimates of $\boldsymbol{\sigma}_r$ and of its first and second derivatives made by robot i, which by virtue of (3.10), are computed as

$$\begin{cases}{}^i\hat{\boldsymbol{\sigma}}_r(t) = \boldsymbol{\sigma}_n({}^i\hat{c}(t))\\ {}^i\dot{\hat{\boldsymbol{\sigma}}}_r(t) = \dfrac{\partial\boldsymbol{\sigma}_n({}^i\hat{c}(t))}{\partial\,{}^i\hat{c}}\,{}^i\dot{\hat{c}}(t)\\ {}^i\ddot{\hat{\boldsymbol{\sigma}}}_r(t) = \dfrac{\partial^2\boldsymbol{\sigma}_n({}^i\hat{c}(t))}{\partial\,{}^i\hat{c}^2}{}^i\dot{\hat{c}}^2(t) + \dfrac{\partial\boldsymbol{\sigma}_n({}^i\hat{c}(t))}{\partial\,{}^i\hat{c}}\,{}^i\ddot{\hat{c}}(t)\end{cases} \tag{3.31}$$

being ${}^i\hat{c}$, ${}^i\dot{\hat{c}}$, ${}^i\ddot{\hat{c}}$ the estimates made by robot i of the scaling parameters c, $\dot{c}$, $\ddot{c}$ computed by the leader robot, and ${}^i\hat{\boldsymbol{\sigma}}$, ${}^i\dot{\hat{\boldsymbol{\sigma}}}$ are the estimates of the task function $\boldsymbol{\sigma}$, $\dot{\boldsymbol{\sigma}}$ in (2.13) computed as

$${}^i\hat{\boldsymbol{\sigma}} = \boldsymbol{J}_{\sigma}{}^i\hat{\boldsymbol{x}}, \quad {}^i\dot{\hat{\boldsymbol{\sigma}}} = \boldsymbol{J}_{\sigma}{}^i\dot{\hat{\boldsymbol{x}}} \tag{3.32}$$

being ${}^i\hat{\boldsymbol{x}}$ and ${}^i\dot{\hat{\boldsymbol{x}}}$ the estimates made by robot i of the collective vectors $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$, respectively.
Summarizing, once the leader robot has computed the scaling parameter $c(t)$ and its first two derivatives, it is necessary for the

solution to Problem 3.2 to work, that they are estimated by follower robots in order to determine the reference trajectory according to (3.31) and that $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ are estimated by all robots so as to compute (3.32) and, finally, the virtual input in (3.30). In the following, the distributed estimators of all required quantities are detailed.

Distributed estimation of $\bar{F}$, μ_1, μ_2, $\nabla\bar{F}$

The problem of estimating and tracking the cumulative safety index $\bar{F}(t)$ in (3.6) and the coefficients $\mu_1(t)$ and $\mu_2(t)$ in (3.24) is first considered. Note that the latter are required to perform the scaling procedure by the leader robot. To this aim, the following stacked vector is introduced

$$\boldsymbol{\eta}_i(t) = \begin{bmatrix}\bar{F}_i(t) & \mu_{1,i}(t) & \mu_{2,i}(t) & \nabla\bar{F}_i^T(t)\end{bmatrix}^T \in \mathbb{R}^6$$

with $i = 1, 2, \ldots, N$, and the cumulative variable $\boldsymbol{\eta}(t) = \sum_{i=1}^N \boldsymbol{\eta}_i(t)$, where $\nabla\bar{F}_i(t) \in \mathbb{R}^3$ represents the cumulative gradient of the safety index $\bar{F}_i$ with respect to the positions of the human points defined as $\nabla\bar{F}_i(t) = \frac{1}{n_o}\sum_{j=1}^{n_o} \nabla_{p_{o,j}}\bar{F}_i(t)$. In addition, the cumulative gradient of the safety index $\bar{F}$ is defined as $\nabla\bar{F} = \sum_{i=1}^N \nabla\bar{F}_i(t)$, which is introduced in order to make it possible its exploitation in the case of redundant robots as shown in Section 3.4. The objective is to define a distributed algorithm such that each robot tracks in finite time $\boldsymbol{\eta}(t)$, i.e., such that there exists a time T_η for which it holds

$$\|{}^i\hat{\boldsymbol{\eta}}(t) - \boldsymbol{\eta}(t)\| = 0, \ \forall t \geq T_\eta, \ \forall\, i \tag{3.33}$$

being ${}^i\hat{\boldsymbol{\eta}}(t) \in \mathbb{R}^3$ the estimate of $\boldsymbol{\eta}(t)$ made by the ith robot. By leveraging the approach in [79] working for undirected graphs, this problem can be solved by having each robot running the following update law

$$\begin{cases} {}^{i}\dot{\hat{\boldsymbol{\nu}}}(t) = k_{\eta} \displaystyle\sum_{j \in \mathcal{N}_i} \operatorname{sign}\left({}^{j}\hat{\boldsymbol{\xi}}(t) - {}^{i}\hat{\boldsymbol{\xi}}(t) \right) \\ {}^{i}\hat{\boldsymbol{\xi}}(t) = {}^{i}\hat{\boldsymbol{\nu}}(t) + \boldsymbol{\eta}_i(t) \end{cases} \tag{3.34}$$

where ${}^{i}\hat{\boldsymbol{\nu}}(t) \in \mathbb{R}^3$ is an internal state, k_η is a positive constant and sign$(\cdot)$ is the component-wise signum function. By defining ${}^{i}\hat{\boldsymbol{\eta}}(t) = N\,{}^{i}\hat{\boldsymbol{\xi}}(t)$, it can be proved [79] that ${}^{i}\hat{\boldsymbol{\eta}}(t)$ converges to $\boldsymbol{\eta}(t)$ in finite time and, then, (3.33) holds true. Therefore, the update law in (3.34) allows the leader to track $\boldsymbol{\eta}(t)$ and consequently to execute the scaling procedure in (3.27) or (3.28).

Note that the observer in (3.34) requires the same quantity $\boldsymbol{\eta}(t)$ to be estimated by the follower robots despite it being not strictly necessary. However, this quantity can be exploited to let each agent (not only the leader one) monitor the safety field, and/or, as stated before, to properly handle the possible kinematic redundancy in order to increase the cumulative safety index. Finally, the related computational load is totally affordable for the system at hand being $\boldsymbol{\eta}(t) \in \mathbb{R}^6$.

Distributed estimation of $c(t)$, $\dot{c}(t)$ and $\ddot{c}(t)$ by the follower robots

At this point, as explained before, the problem of allowing follower robots to track the scaling parameters determined by the leader is consider. Let $\boldsymbol{\varsigma}(t)$ denote the stacked vector of the scaling parameters, that is

$$\boldsymbol{\varsigma}(t) = \begin{bmatrix} \hat{c}(t) & \dot{\hat{c}}(t) & \ddot{\hat{c}}(t) \end{bmatrix}^T \in \mathbb{R}^3$$

and ${}^{i}\hat{\boldsymbol{\varsigma}}(t)$ denote the respective estimation made by the follower robot i. The objective is to define a distributed leader tracking algorithm such that there exists a time T_c for which it holds

$$\|{}^{i}\hat{\boldsymbol{\varsigma}}(t) - \boldsymbol{\varsigma}(t)\| = 0, \ \forall t \geq T_c, \forall\, i. \tag{3.35}$$

To this aim, the solution in [80] is exploited and the following distributed observer is adopted by the follower robots

$$\begin{cases} {}^{i}\dot{\hat{\boldsymbol{\varsigma}}}(t) = \boldsymbol{A}_{\varsigma}{}^{i}\hat{\boldsymbol{\varsigma}}(t) - k_{\varsigma,1}\boldsymbol{B}_{\varsigma}\boldsymbol{B}_{\varsigma}^{T}\boldsymbol{P}_{\varsigma}^{-1}\,{}^{i}\hat{\boldsymbol{\rho}}(t) - k_{\varsigma,2}\,\mathrm{sign}\left(\boldsymbol{P}_{\varsigma}^{-1}\,{}^{i}\hat{\boldsymbol{\rho}}(t)\right) \\ {}^{i}\hat{\boldsymbol{\rho}}(t) = \displaystyle\sum_{j\in\mathcal{N}_i,\, j\neq l}\left({}^{i}\hat{\boldsymbol{\varsigma}}(t) - {}^{j}\hat{\boldsymbol{\varsigma}}(t)\right) + b_i\left({}^{i}\hat{\boldsymbol{\varsigma}}(t) - \boldsymbol{\varsigma}(t)\right) \end{cases} \tag{3.36}$$

where l is the index associated with the leader robot, i belongs to the set of the follower robots, b_i is 1 if the leader belongs to $\mathcal{N}_i$ ($l \in \mathcal{N}_i$) and is 0 otherwise, ${}^{i}\hat{\boldsymbol{\rho}} \in \mathbb{R}^3$ is an auxiliary vector, $k_{\varsigma,1}$ and $k_{\varsigma,2}$ are positive gains, $\boldsymbol{A}_{\varsigma} \in \mathbb{R}^{3\times3}$ and $\boldsymbol{B}_{\varsigma} \in \mathbb{R}^3$ are matrices selected as

$$\boldsymbol{A}_{\varsigma} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{B}_{\varsigma} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$$

and $\boldsymbol{P}_{\varsigma} \in \mathbb{R}^{3\times3}$ is a positive definite matrix. It is worth reminding that, since the communication graph is undirected and connected, there always exists a path from the leader to any follower. By following the same reasoning as in [80], it can be proved that the distributed observer in (3.36) fulfills the convergence condition in (3.35) under the connectedness condition of the undirected communication graph and a proper selection of the gains $k_{\varsigma,1}$ and $k_{\varsigma,2}$. Thus, by exploiting the components of ${}^{i}\hat{\boldsymbol{\varsigma}}(t)$, the ith robot can compute ${}^{i}\hat{\boldsymbol{\sigma}}_r(t)$, ${}^{i}\dot{\hat{\boldsymbol{\sigma}}}_r(t)$ and ${}^{i}\ddot{\hat{\boldsymbol{\sigma}}}_r(t)$ according to (3.31).

Distributed estimation of $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$

Finally, the finite-time observer that allows each robot to estimate the overall multi-robot state needed for (3.32) is introduced. For this purpose, let $\boldsymbol{\chi} \in \mathbb{R}^{2Np}$ denote the stacked vector of the collective end effector configurations and velocities, i.e.,

$$\boldsymbol{\chi} = \begin{bmatrix} \boldsymbol{x}_1^T & \dot{\boldsymbol{x}}_1^T & \dots & \boldsymbol{x}_N^T & \dot{\boldsymbol{x}}_N^T \end{bmatrix}^T$$

and $\boldsymbol{S}_p \in \mathbb{R}^{Np\times 2Np}$ and $\boldsymbol{S}_v \in \mathbb{R}^{Np\times 2Np}$ the selection matrices of the positional and velocity components in $\boldsymbol{\chi}$, respectively, that is

$$\boldsymbol{S}_p = \boldsymbol{I}_N \otimes \begin{bmatrix}\boldsymbol{I}_p & \boldsymbol{O}_p\end{bmatrix}, \quad \boldsymbol{S}_v = \boldsymbol{I}_N \otimes \begin{bmatrix}\boldsymbol{O}_p & \boldsymbol{I}_p\end{bmatrix}$$

such that

$$\boldsymbol{S}_p\boldsymbol{\chi} = \begin{bmatrix}\boldsymbol{x}_1^T & \dots & \boldsymbol{x}_N^T\end{bmatrix}^T, \boldsymbol{S}_v\boldsymbol{\chi} \ = \begin{bmatrix}\dot{\boldsymbol{x}}_1^T & \dots & \dot{\boldsymbol{x}}_N^T\end{bmatrix}^T.$$

The aim is to design a distributed observer which allows each robot to track in finite-time the collective state $\boldsymbol{\chi}$, i.e., such that there exists a finite time T_χ for which it holds

$$\|{}^i\hat{\boldsymbol{\chi}}(t) - \boldsymbol{\chi}(t)\| = 0, \ \forall t \geq T_\chi \tag{3.37}$$

being ${}^i\hat{\boldsymbol{\chi}} \in \mathbb{R}^{2Np}$ the estimation of $\boldsymbol{\chi}$ made by robot i. By resorting to the approach in [81], the following update law is selected

$$\begin{cases} {}^i\dot{\hat{\boldsymbol{\chi}}} = \boldsymbol{A}_\chi {}^i\hat{\boldsymbol{\chi}} + k_{\chi,1}\,\boldsymbol{G}_\chi {}^i\hat{\boldsymbol{\zeta}} + k_{\chi,2} \text{ sign}\left(\boldsymbol{G}_\chi {}^i\hat{\boldsymbol{\zeta}}\right) + \boldsymbol{B}_\chi {}^i\hat{\boldsymbol{u}}_\chi \\ {}^i\hat{\boldsymbol{\zeta}} {=} \boldsymbol{\Pi}_i(\boldsymbol{\chi} - {}^i\hat{\boldsymbol{\chi}}) + \sum\limits_{j\in\mathcal{N}_i}\left({}^j\hat{\boldsymbol{\chi}} - {}^i\hat{\boldsymbol{\chi}}\right) \end{cases} \tag{3.38}$$

where ${}^i\hat{\boldsymbol{\zeta}} \in \mathbb{R}^{2Np}$ is an internal state, $k_{\chi,1}$ and $k_{\chi,2}$ are positive gains, $\boldsymbol{G}_\chi \in \mathbb{R}^{2Np\times 2Np}$ is a positive definite matrix, $\boldsymbol{A}_\chi \in \mathbb{R}^{2Np\times 2Np}$ and $\boldsymbol{B}_\chi \in \mathbb{R}^{2Np\times Np}$ are selected as follows

$$\boldsymbol{A}_\chi = \boldsymbol{I}_N \otimes \begin{bmatrix}\boldsymbol{O}_p & \boldsymbol{I}_p \\ \boldsymbol{O}_p & \boldsymbol{O}_p\end{bmatrix}, \quad \boldsymbol{B}_\chi = \boldsymbol{I}_N \otimes \begin{bmatrix}\boldsymbol{O}_p \\ \boldsymbol{I}_p\end{bmatrix}$$

${}^i\hat{\boldsymbol{u}}_\chi \in \mathbb{R}^{Np}$ is the observer input defined as

$${}^i\hat{\boldsymbol{u}}_\chi = \boldsymbol{J}_\sigma^\dagger\left({}^i\ddot{\hat{\boldsymbol{\sigma}}}_r + k_{\sigma,d}\left({}^i\dot{\hat{\boldsymbol{\sigma}}}_r - {}^i\dot{\hat{\boldsymbol{\sigma}}}\right) + k_{\sigma,p}\left({}^i\hat{\boldsymbol{\sigma}}_r - {}^i\hat{\boldsymbol{\sigma}}\right)\right) \tag{3.39}$$

where, by taking into account (3.32), it holds

$$\begin{cases} {}^{i}\hat{\boldsymbol{\sigma}} = \boldsymbol{J}_{\sigma}\boldsymbol{S}_{v}{}^{i}\hat{\boldsymbol{\chi}} \\ {}^{i}\dot{\hat{\boldsymbol{\sigma}}} = \boldsymbol{J}_{\sigma}\boldsymbol{S}_{v}{}^{i}\dot{\hat{\boldsymbol{\chi}}}. \end{cases} \tag{3.40}$$

It is worth remarking that, despite the presence of the collective vector $\boldsymbol{\chi}$, the observer system in (3.38) exclusively depends on local information; indeed, the matrix $\boldsymbol{\Pi}_i$ selects only the components of $\boldsymbol{\chi}$ related to the i th robot, i.e.,

$$\boldsymbol{\Pi}_i\boldsymbol{\chi} = \begin{bmatrix}\boldsymbol{0}_{2p}^T & \cdots & \underbrace{\boldsymbol{x}_i^T \quad \dot{\boldsymbol{x}}_i^T}_{i \text{ th robot}} & \cdots & \boldsymbol{0}_{2p}^T\end{bmatrix}^T.$$

By leveraging the approach in [81] it can be shown that the update law in (3.38) ensures that (3.37) holds provided that the gains $k_{\chi,1}$ and $k_{\chi,2}$ satisfy the conditions detailed in [81].

The reader is referred to [79], [80] and [81] for further details concerning the respective observers.

In conclusion, the main components of the devised decentralized solution for each robot i are reported in Figure 3.4, where different block colors are adopted for leader (in blue) and follower (in block) if different actions are performed from them. It makes evident that the information flow of the i th follower robot differs from that of the leader in the definition of the reference trajectory. More specifically, no execution of the avoidance strategy is performed but the scaling parameters $\boldsymbol{\varsigma}(t)$ are estimated through the observer system in (3.36) and are then used in (3.32) to compute the reference trajectory.

3.4 Simulation results

In this section, numerical simulations are provided to corroborate the devised approach. In detail, first the cooperative multi-robot task and the safety index generically introduced in (2.13) and (3.1), respectively, are defined and, then, two case studies are discussed

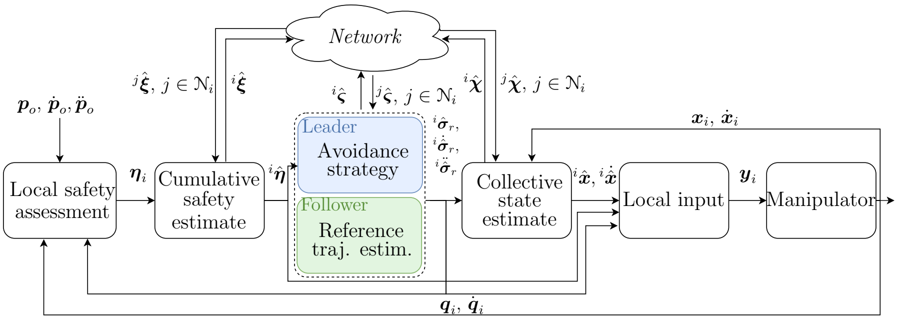

Network
${}^{j}\hat{\boldsymbol{\xi}},\ j \in \mathcal{N}_i$
${}^{i}\hat{\boldsymbol{\xi}}$
${}^{i}\hat{\varsigma}$
${}^{j}\hat{\varsigma},\ j \in \mathcal{N}_i$
${}^{i}\hat{\boldsymbol{\chi}}$
${}^{j}\hat{\boldsymbol{\chi}},\ j \in \mathcal{N}_i$
$\boldsymbol{p}_o, \dot{\boldsymbol{p}}_o, \ddot{\boldsymbol{p}}_o$
Leader
Avoidance strategy
Follower
Reference traj. estim.
${}^{i}\hat{\boldsymbol{\sigma}}_r,$
${}^{i}\dot{\hat{\boldsymbol{\sigma}}}_r,$
${}^{i}\ddot{\hat{\boldsymbol{\sigma}}}_r$
$\boldsymbol{x}_i,\ \dot{\boldsymbol{x}}_i$
Local safety assessment
$\boldsymbol{\eta}_i$
Cumulative safety estimate
${}^{i}\hat{\boldsymbol{\eta}}$
Collective state estimate
${}^{i}\hat{\boldsymbol{x}},\ {}^{i}\dot{\hat{\boldsymbol{x}}}$
Local input
$\boldsymbol{y}_i$
Manipulator
$\boldsymbol{q}_i,\ \dot{\boldsymbol{q}}_i$

Figure 3.4: Decentralized solution. Control scheme associated with robot i. When different actions are performed by leader and followers, the respective blocks are denoted with blue and green colors, while the other ones are in common.

which concern the centralized and decentralized solutions to Problems 3.1 and 3.2, respectively. A website[1] is provided to collect simulation and experimental results.

3.4.1 Simulation setup

A setup composed of $N = 3$ Comau Smart SiX manipulators (6 DOFs) mounted on mobile bases (2 DOFs) is considered (see Figure 3.5) for a cooperative load transportation task performed in presence of human operators. Therefore, it holds $n_i = 8$ and $p = 6$ in (2.1). As in Section 2.5.3, Matlab environment and CoppeliaSim simulator are used for the simulation setup.
The cooperative task function $\boldsymbol{\sigma}(\boldsymbol{x})$ is expressed in terms of the absolute-relative formulation defined in Section 2.4. In the case of cooperative load transportation task, the absolute variables are suitable for describing the position and the orientation of the grasped object, while the relative variables for specifying how the end effectors are placed around the object in order to grasp it. Furthermore, in order to perform the proposed human multi-robot avoidance strategy, the following expressions of functions α_1 and

[1] `https://m-lippi.github.io/safe-hmri/`

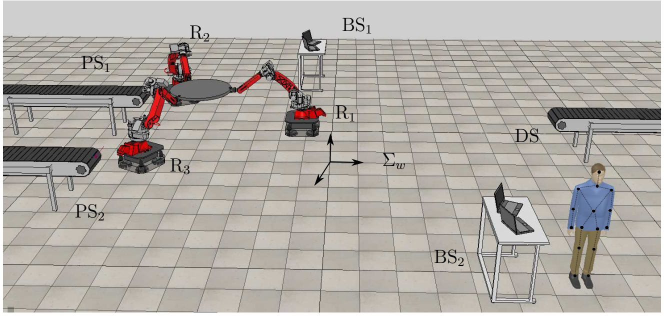


Figure 3.5: Simulation setup composed of 3 cooperative robots (R$_i$, $i = 1, 2, 3$), stations for picking (PS$_i$, $i = 1, 2$) and depositing (DS) loads and base stations for human operators (BS$_i$, $i = 1, 2$); the relevant points considered for the human operator and the world reference frame Σ_w are also reported.

α_2 that comply with Properties 1 and 2 in (3.1) are adopted

$$\begin{cases} \alpha_1(d) = k_1 d \\ \alpha_2(\dot{d}) = k_2 \tanh(\dot{d}) \end{cases} \tag{3.41}$$

with $k_1, k_2 \in \mathbb{R}^+$, leading the safety index f to the form

$$f(\boldsymbol{p}, \dot{\boldsymbol{p}}, \boldsymbol{p}_o, \dot{\boldsymbol{p}}_o) = k_1 d + k_2 \tanh(\dot{d}). \tag{3.42}$$

The rationale behind the choice in (3.41) is to consider a term which linearly grows with the distance and a term that is increasing with the distance derivative but contributes positively to the safety index if the distance is growing, i.e., $\dot{d} > 0$, and negatively if the distance is decreasing, i.e., $\dot{d} < 0$. Concerning the human relevant points, the following points, as highlighted in Figure 3.5 and as common in human skeleton tracking approaches [71], are considered: head, neck, torso, left and right shoulder, elbow, hand, hip, knee and foot, which leads to $n_o = 15$. The safety index in (3.42) is, then, extended to each human point and to the whole structure of each manipulator, resulting in the computation of $\bar{F}_i$ according to (3.3), (3.4) and (3.5). The computation time to obtain $\bar{F}_i$ is discussed in Section 3.4.5.

Now, by considering that robots are redundant (as it holds $n_i > p$, $i = 1, 2, 3$), the devised strategy might also envisage the exploitation of this redundancy to maximize the level of human safety. For this purpose, the joint acceleration vector $\ddot{\boldsymbol{q}}_{n,i}$ in (3.14) of each robot is designed as follows

$$\ddot{\boldsymbol{q}}_{n,i} = k_{n,1}(\boldsymbol{I}_{n_i} - \boldsymbol{J}_i^{\dagger}\boldsymbol{J}_i)(\dot{\boldsymbol{q}}_{0,i} - \dot{\boldsymbol{q}}_i) \tag{3.43}$$

being $k_{n,1}$ a positive constant and $\dot{\boldsymbol{q}}_{0,i} \in \mathbb{R}^{n_i}$ the vector of joint velocities for achieving arbitrary secondary tasks. A possible choice of $\dot{\boldsymbol{q}}_{0,i}$ leverages the procedure defined in [60] that maps the cumulative gradient $\nabla\bar{F}$ into desired velocity of the robot links extreme points and, then, converts them into joint velocities as

$$\dot{\boldsymbol{q}}_{0,i} = -k_{n,2}\left(\sum_{l=1}^{n_i} {\boldsymbol{J}_{i,l}^{1}}^{T}\nabla\bar{F} + {\boldsymbol{J}_{i,1}^{0}}^{T}\nabla\bar{F}\right) \tag{3.44}$$

with $k_{n,2}$ a positive constant. With regards to the trajectory scaling procedure, the update law in (3.27) is selected which allows to recover the nominal velocity when scaling is no longer necessary, i.e., it is considered that this task does not require to restore the position after the scaling phase. Concerning the emergency procedure envisaged in Section 3.3.2 and highlighted in the scheme in Figure 3.3, a monitored stop procedure is considered. More specifically, let t_s be the time when the emergency procedure is activated, such a procedure is based on bringing and keeping the scaling parameters to the following values: $c(t) = c(t_s)$, $\dot{c}(t) = 0$ and $\ddot{c}(t) = 0$, implying that a zero velocity reference trajectory is imposed when this procedure is active. Moreover, zero null space velocities are set in order to prevent internal motions ($\dot{\boldsymbol{q}}_{0,i} = \boldsymbol{0}_{n_i}$). In the considered scenario, the motion of the human operator is characterized by two phases: in the first phase, the human moves from the configuration in Figure 3.6.a to the base station BS_1 as in Figure 3.6.b while, in the second phase, he crosses the work-cell to reach the second base station BS_2 as shown in Figure 3.6.c. At the same time, the robot team cooperatively performs two load transportation tasks: the first one from the top-left conveyor belt PS_1 to the right one DS (see Figure 3.6.a) and the second one from the bottom-left conveyor belt PS_2 to DS (see Figure 3.6.c). In particular during the first transportation, the human operator is stationary at first base station (Figure 3.6.b), whereas during the second transportation the human operator is on the path of the robot team (Figure 3.6.c). A video of the simulation is available in the respective section of the website[1].

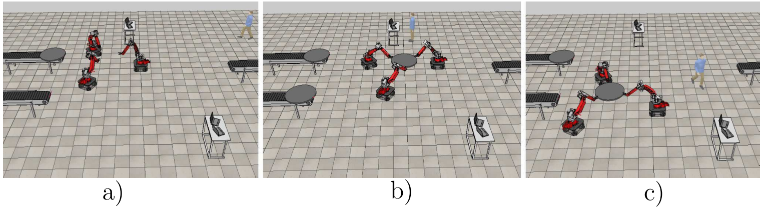


Figure 3.6: Snapshots of the key phases of the simulation: the initial configuration of the work-cell is in Figure (a); the first and the second multi-robot cooperative transport motions are in Figures (b) and (c), respectively.

3.4.2 First case study: centralized solution

Results relative to the centralized architecture described in Section 3.3.2 are here reported. Gains are selected as $k_{\sigma,d} = 20$, $k_{\sigma,p} = 100$, $k_{c,d} = 4.5$, $k_{n,1} = 1$, $k_{n,2} = 2$ in (3.14), (3.27), (3.43) and (3.44), respectively. Moreover, $\bar{F}_{min} = 105$ ($\dot{\bar{F}}_{min} = 0$) is assumed in Problem 3.1, which is defined by requiring $d^* \geq 0.1$ m in (3.8) and by following the computations in the Appendix B for the robots and the safety index at hand.
The combined motion of the human operator and the robots outlined above leads the avoidance strategy to mainly modify the nominal cooperative trajectory in two phases:

1. a scaling phase, indicated with S_c in Figures 3.7, 3.8 and 3.9, which occurs during the first cooperative transport, when the robots pass next to the base station where the human is standing. In this case, the trajectory is slowed down in order to comply with the minimum safety index;

2. an emergency phase, indicated with E_m in the figures, which occurs during the second cooperative transport, when the person crosses the nominal path of the robots. In this case, the velocity modulation is not sufficient to ensure the minimum safety index, thus a monitored stop of the robots is performed as detailed in Section 3.4.1.

The occurrence of these phases depend on how the scaling parameter $c(t)$ and its derivatives vary over time in response to the human multi-robot coexistence. In detail, as shown in Figure 3.7, where the evolution of the scaling parameters (blue lines) with respect to their nominal values (green lines) is reported, the scaling phase S_c is characterized by a decrease of the coefficient $\dot{c}(t)$ from its nominal value (namely, 1), thus implying a slowing down of the nominal trajectory; then, in accordance to the update law in (3.27), the nominal velocity is always restored after a scaling phase ($\dot{c}(t) = \dot{t} = 1$). Note that no nominal position is restored as it is assumed, as an example, that the latter is not necessary for the task at hand and the velocity recovery is sufficient. During the emergency phase E_m, no representation of these parameters is provided in Figure 3.7 as the task of the robots is completely aborted ($\dot{c}(t)$ reaches the origin); however, it can be noticed that the emergency phase starts since the scaling procedure leads to violate the constraint on the reverse motion along the path, that is $\dot{c}(t) < 0$. Then, when the safety conditions are restored ($\bar{F} > \bar{F}_{min}$) due to the increasing of the distance of the operator from the robots, the nominal cooperative trajectory is restored as well.

Such phases are also evident in Figure 3.8, where the evolution over time of the cumulative safety index $\bar{F}$ (blue line) with respect to the minimum value $\bar{F}_{min}$ (red line) is shown. In detail, the scaling phase starts when the safety index reaches the minimum value and is mainly characterized by the saturation of this index to this minimum value. Concerning the emergency phase, it determines the interruption of the robots task and their consequent stop; this means that, once the robots are stationary, the safety index can fall below the minimum only if the person continues to approach the robots. It is worth remarking that this situation has been

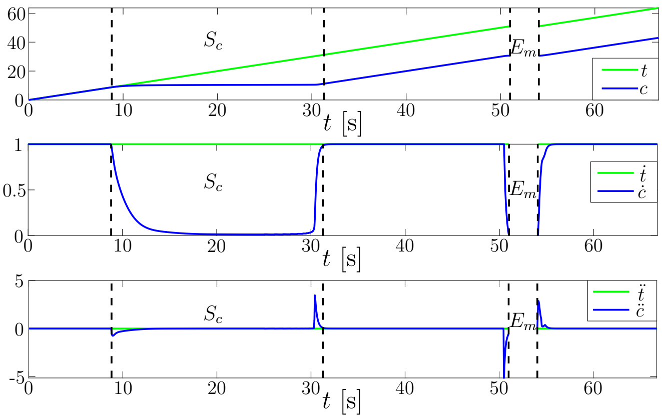


Figure 3.7: First case study. Evolution of the scaling parameters $c(t)$, $\dot{c}(t)$, $\ddot{c}(t)$ (in blue) compared with their nominal values (in green); scaling and emergency phases are highlighted with S_c and E_m, respectively. No plots are provided during the latter phase since the task is interrupted.

stressed in the simulation for the sake of completeness but that the values of $\bar{F}$ below $\bar{F}_{min}$ are only related to the human motion towards the inactive robots, situation that does not endanger the human operator.

Finally, Figure 3.9 shows the reference task trajectory $\boldsymbol{\sigma}_r(t)$ (in blue), which is cooperatively tracked by the robots, compared with the nominal one $\boldsymbol{\sigma}_n(t)$ (in green); more specifically, the positional components of the absolute task variables $\boldsymbol{\sigma}_1$ are shown, which highlight how the devised strategy modifies the nominal behavior. In particular, the scaling phase determines the slowing down of the trajectory and, after this, in accordance to the update law in (3.27), no recovery of the introduced shift between the nominal and reference positional trajectories is performed and the latter evolves with the same variation as t; concerning the emergency phase, the monitored stop procedure generates a constant trajectory that is equal to the one at the stopping time $t \approx 51$ s.

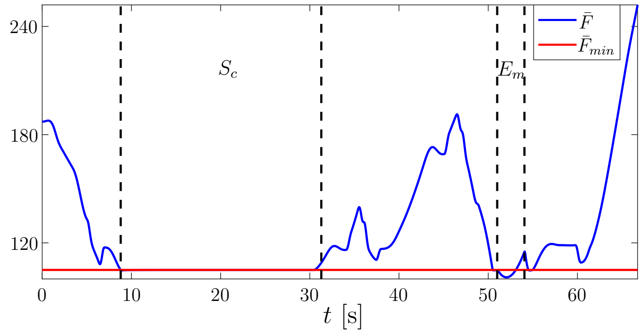


Figure 3.8: First case study. Evolution of the cumulative safety index (in blue) with respect to its minimum allowed value (in red); scaling and emergency phases are marked with S_c and E_m, respectively.

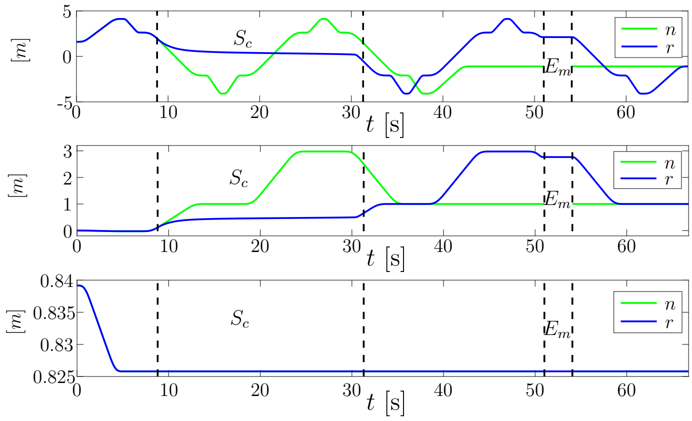


Figure 3.9: First case study. Evolution of the nominal (n, in green) and reference (r, in blue) trajectories; in detail, the positional components of the team centroid $\boldsymbol{\sigma}_1$ are shown (namely, $\boldsymbol{\sigma}_{1,x}$, $\boldsymbol{\sigma}_{1,y}$, $\boldsymbol{\sigma}_{1,z}$); scaling and emergency phases are marked with S_c and E_m, respectively. When the emergency procedure is active (E_m), the nominal trajectory is not shown since its tracking is interrupted.

3.4.3 Second case study: decentralized solution

In this section, it is shown how the same results obtained above might be obtained within a decentralized solution as described in Section 3.3.3 which proves the feasibility of the devised decentralized strategy. To this aim, the same setup as the one described in Section 3.4.1 is considered and the corresponding results are compared with those presented in Section 3.4.2. Therefore, the minimum safety index $\bar{F}_{min}$ and the gains involved in the virtual inputs as well as in the scaling update law are selected as in Section 3.4.2. Concerning the gains of the estimators in (3.34), (3.36) and (3.38), they are selected as $k_\eta = 10$, $k_{\varsigma,1} = 20$, $k_{\varsigma,2} = 10$, $k_{\chi,1} = 5$, $k_{\chi,2} = 5$,

$$\boldsymbol{P}_\varsigma = \begin{bmatrix} 0.09 & -0.05 & 0.01 \\ -0.05 & 0.08 & -0.08 \\ 0.01 & -0.08 & 0.31 \end{bmatrix}, \quad \boldsymbol{G}_\chi \quad = 2\,\boldsymbol{I}_{2Np}.$$

Furthermore, in order for each robot to exploit its redundancy for increasing the safety index $\bar{F}(t)$, the observer system in (3.34) is exploited. More in detail, the last three components of ${}^i\hat{\boldsymbol{\eta}}$ represent the estimation of the cumulative gradient of the safety index, namely $\nabla\bar{F}$, required to compute the null space joint acceleration vector $\ddot{\boldsymbol{q}}_{n,i}$ according to (3.43) and (3.44).

With regard to the communication graph, bi-directional links between robot 1 and 2 and between robot 2 and 3 are assumed. Finally, without loss of generality, the robot 1 is set as team leader ($l = 1$).

Simulation results are summarized in Figure 3.10 which compares the evolution of the safety index $\bar{F}$ obtained through the decentralized architecture (in blue and denoted by $\bar{F}_d$) with the one obtained with the centralized approach (in green and denoted by $\bar{F}_c$); the minimum allowed value $\bar{F}_{min} = 105$ is also shown (in red). Hence, it is evident that the two plots are almost equal and, therefore, the same considerations as before regarding the avoidance strategy hold. Note that, even in the case the distributed solution is adopted in a real-world scenario, the employment of finite-time observers guarantees that only during the initial transient phases (i.e., before the estimation errors reach the origin), an erroneous solution may be obtained which may not fulfill the human safety condition. The convergence time mainly depends on initial estimation error (if the estimation error is zero at the initial time instant, then it is zero at all times), and it decreases by increasing the respective gains that are $k_{\eta}, k_{\varsigma,1}, k_{\varsigma,2}, k_{\chi,1}, k_{\varsigma,2}$ in (3.34), (3.36), and (3.38), and increases with the number of robots and the diameter of the communication network. Since the number of mobile manipulators is always limited in applications that make sense in practical industrial scenarios, the network topology is never an issue, while control gains can be made arbitrary large compatibly with a digital implementation. In addition, in order to be as conservative as possible, the leader agent could initialize the estimation of $\bar{F}$ at a value which is equal or close to the threshold $\bar{F}_{min}$ in order to start with or be close to a scaling phase; while the estimation of the scaling coefficient c is never an issue since it is initialized by all robots at the same right value (i.e., t_0) and then, the estimation error is always zero. As a further conservative solution, a waiting time with duration equal to the maximum convergence time can be introduced at the start-up of the system to guarantee that all the quantities are properly tracked afterwards. The maximum convergence time can be computed assuming that the robots initial workspace is limited in a certain region, as realistic in industrial settings with collaborative robots. Moreover, the dependence of the settling time from the initial conditions might be further released if *fixed*-time observers are considered, which might be object of future work.

3.4.4 Comparison with other approaches

In this section, the proposed approach is validated against other possible approaches to the same problem tackled in this chapter. By leveraging and extending the fluency metrics in [15], the following metrics are considered to quantitatively and compactly

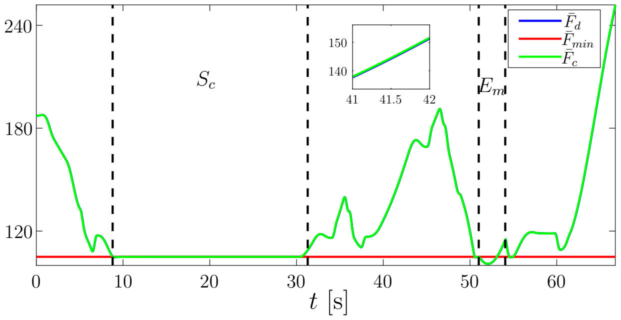


Figure 3.10: Second case study. Evolution of the cumulative safety index $\bar{F}_d$ (in blue) in the decentralized case with respect to its minimum allowed value $\bar{F}_{min}$ (in red) and the cumulative safety index obtained with the centralized strategy $\bar{F}_c$ (in green). A zoom of the evolution of these indexes in the interval [41, 42]s is provided. Scaling and emergency phases are indicated with S_c and E_m, respectively.

compare the proposed method with baseline solutions:

(M1) Total execution time, that is the time to complete the entire nominal path;

(M2) Robots idle time, that is the amount of time in which the robots task is interrupted;

(M3) Time of concurrent motion, that is the amount of time in which both robots and human are moving simultaneously normalized with respect to the human motion duration;

(M4) Path constraint satisfaction, that is the amount of time in which the nominal path is followed;

(M5) Average safety index.

In particular, metric (M4) reflects the constraint to preserve as much as possible the nominal path, while metric (M5) condenses information about the human-robots relative motion. In addition, the robots are considered to be in idle state when either the path is abandoned or no motion is made along it.

Table 3.1: Comparison with baselines (best results in bold)

Method	(M1) [s]	(M2) [%]	(M3) [%]	(M4) [%]	(M5)
ES baseline	88.1	50	28.3	100	169
EA baseline	90.8	52.3	35.3	47.7	**180.7**
Proposed	**66.8**	**4.6**	**100**	**100**	127.4

Two baseline solutions are thus introduced against which the proposed method is compared. First, the Emergency Stop (ES) baseline is considered which performs a monitored stop of the robots along the nominal path when the safety index $\bar{F}$ in (3.6) reaches the minimum safety $\bar{F}_{min}$ and recovers the task when $\bar{F} > \bar{F}_{min} + \Delta\bar{F}$, with $\Delta\bar{F} > 0$. Second, the Evasive Action (EA) baseline is analyzed which resorts to the approach in [59, Section 4.2] and consists in relaxing the constraint on the nominal path maintenance. To this aim, the absolute position of the object is modified according to an impedance-based dynamics whose virtual force depends on the gradient $\nabla\bar{F}(t)$ of the cumulative safety index and on its value $\bar{F}(t)$. This allows to respect the kinematic constraint introduced by the tight connection which requires the relative variables $\boldsymbol{\sigma}_2$ to be constant, while allowing possible variation of the absolute ones $\boldsymbol{\sigma}_1$. The reader can refer to [59] for further details which are here omitted for the sake of brevity.

Table 3.1 summarizes the comparison between the proposed method and the two baselines ($\Delta\bar{F} = 10$ is set for ES) in terms of metrics (M1)-(M5) in the same simulation scenario of Section 3.4.1. In particular, it shows that the proposed approach outperforms the baselines with respect to all metrics except the average safety index (M5). This is motivated by the fact that the proposed approach does not require to abort the task when the safety index reaches the minimum value, but rather tries to continue the task by modulating the velocity along the path while ensuring the safety index does not fall below $\bar{F}_{min}$. In contrast, the EA baseline achieves best average safety index as it allows to modify the path to increase human safety differently from the other methods. Concerning the other metrics, the strategy not to interrupt the task allows to significantly reduce the total execution

Table 3.2: Real-time computation analysis

# points	*Computation time*[s]	*Integration error*
50	$2.27 \cdot 10^{-5} \pm 6.95 \cdot 10^{-9}$	$4.2 \cdot 10^{-4} \pm 4.12 \cdot 10^{-5}$
100	$3.01 \cdot 10^{-5} \pm 1.36 \cdot 10^{-8}$	$2.05 \cdot 10^{-4} \pm 2.01 \cdot 10^{-5}$
500	$8.84 \cdot 10^{-5} \pm 6.65 \cdot 10^{-8}$	$3.23 \cdot 10^{-5} \pm 3.17 \cdot 10^{-6}$
1000	$1.61 \cdot 10^{-4} \pm 1.33 \cdot 10^{-7}$	$1.08 \cdot 10^{-5} \pm 1.06 \cdot 10^{-6}$
1500	$2.34 \cdot 10^{-4} \pm 1.99 \cdot 10^{-7}$	$3.59 \cdot 10^{-6} \pm 3.53 \cdot 10^{-7}$

time (M1) and the robots idle time (M2) as well as to maximize the human-robots concurrent motion time (M3) (100%) compared to the baselines. A small percentage of inactivity is recorded in the proposed method simulation as the emergency stop procedure is activated when the human crosses the robots nominal path. Finally, the proposed method as well as the ES baseline ensure by construction that the path constraint (M4) is satisfied.

3.4.5 Computational issues

As discussed in [60], a trade-off exists between accuracy and speed of computation of the cumulative safety index $\bar{F}$, especially when non-integrable functions α_1 and α_2 are used in (3.1). Therefore, real-time computation issues are investigated in this section, and the analysis is carried out on a PC with standard hardware consisting of Intel Core i7-2.4 GHz processor and 8 GB RAM. The case of single Comau SmartSiX and single representative point for the human is considered. Forward Euler integration is adopted for computing the safety index and each link is discretized with a number of points ranging in the set $\mathcal{P} = \{2, 3, \ldots, 2000\}$. For each discretization parameter in $\mathcal{P}$, $5 \cdot 10^5$ computations of the safety index, randomly varying human and robot configurations, are performed to compute mean values and standard deviations. Table 3.2 summarizes results on how the number of points for numerical integration affects the computation time as well as the relative integration errors by assuming as baseline the safety value obtained with $\max\{\mathcal{P}\}$ points, i.e., 2000 points (by considering more points, the value of the field does not significantly change).

More specifically, although the computation times increase with the number of points, they are shown to be fully compatible with real-time execution on standard PCs also considering a high number of points. This is in view of the fact that typical robot sample time is greater than or equal to 10^{-3} s (as reported for example in Section 3.5). Concerning the relative errors, by choosing about 350 points in set $\mathcal{P}$, a mean value lower than 0.005% (with standard deviation lower than $5.3 \cdot 10^{-6}$) is measured with a computational time less than 10^{-4} s. By virtue of these results, it follows that the computational load is totally affordable still preserving high accuracy in the computation. Numerical integration with 400 points (having mean computation time $6.8 \cdot 10^{-5}$ s) was considered for real-time experiments in the following.

Finally, note that, when the distributed architecture is adopted, each robot only computes its own contribution $\bar{F}_i$ leading to an *inherently* parallel and distributed computation, i.e., the computational burden for $\bar{F}$ does not increase with the number of robots.

3.5 Experiments

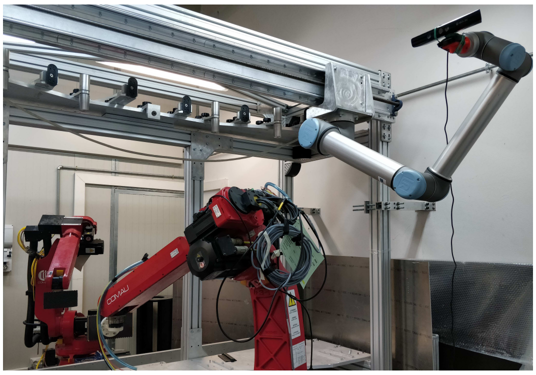


Figure 3.11: Experimental case study. Picture of the work-cell composed of two Comau SmartSiX robots (red robots) and one UR10 robot (ceiling mounted and gray robot) equipped with a Kinect sensor at its end effector.

The proposed solution has been validated on the setup shown in Figure 3.11. The work-cell is composed of two ($N = 2$) Comau Smart SiX robots (6 DOFs) mounted on a common sliding track (1 DOF), that have the role of workers, and one UR10 from Universal Robots (6 DOFs) equipped with a RGB-D Microsoft Kinect sensor for scene analysis, that has the role of watcher. Due to the limited dimension of the work-cell (see Figure 3.11), the redundant degrees of freedom introduced by the sliding track are not exploited in the experiment (i.e., $n_1 = n_2 = 6$ for both the Comau robots). Concerning the control of the Comau robots, the *open* architecture of the respective controllers, namely the C4G controllers, is used. This architecture allows to integrate a standard PC featuring a Linux real-time operating system with the robot control unit. In the considered work-cell, the two workers are connected via Ethernet to the same PC and exchange data at a rate of 2 ms; the PC is then in charge of acquiring the data from robot controllers (e.g., joint positions, velocity and currents), executing the human multi-robot avoidance strategy and finally setting the joint references back to the C4G controllers. It is worth highlighting that, since only two robots are present (so the hypothesis of connected communication graph would also make the graph complete), the centralized approach in Section 3.3.2 has been considered. Concerning the UR10, it is controlled by a different PC, equipped with ROS middleware and standard Linux operating system, which is in charge of controlling the robot motion at a rate of 8 ms and performing image analysis computations to monitor the work-cell with respect to the human presence. More specifically, the human skeleton is tracked by using the OpenNI tracker package [2] which allows to retrieve the configurations of the human head, neck, torso, left and right shoulder, elbow, hand, hip, knee and foot. Without loss of generality, only one point associated with the human operator is considered ($n_o = 1$) and it is selected to coincide with that of the right hand; this choice allows to better understand the robots behavior in the video experiment. The watcher robot is then moved in such a way as to continuously inspect the robots

[2] `http://wiki.ros.org/openni_tracker`

working area and, once an operator is detected, in such a way as to maintain a favorable distance and viewing angle from him/her. The human hand position is, then, transferred to the PC controlling the Comau robots in order to execute the avoidance strategy. Finally, human velocity and acceleration are obtained by filtering the position data and by applying finite difference. In this regard, different solutions, also possibly relying on Inertial Measurement Unit systems [82], may be included, but are not examined hereby being out of the scope. Finally, both PCs have the characteristics listed in Section 3.4.5. The experiment consists in performing a cooperative transportation of a box by the workers, as shown in Figure 3.12, while a human operator can move in their proximity.

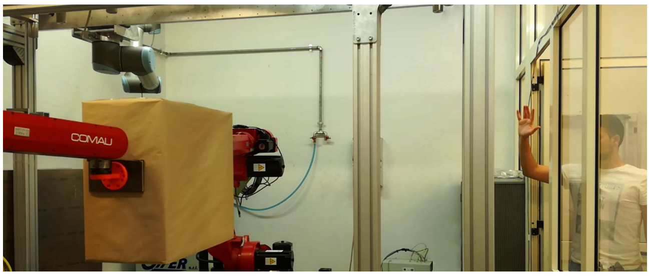


Figure 3.12: Experimental case study. Representation of the cooperative transportation task by the Comau robots. A human operator (on the right) moves nearby and his right hand is selected as representative point.

The same absolute-relative formulation as for the simulation case study is here adopted to define the workers' task and the nominal trajectory is designed so as to perform a periodic circular motion with radius 0.15 m in the vertical plane normal to the sliding track. A video of the experiment is available in the respective section of the website[1].

Concerning the human multi-robot avoidance strategy defined in Section 3.3.2, details about the safety index, the scaling update law and the emergency procedure are now provided. As regards the trajectory scaling, differently from the simulation case studies, the update law in (3.28) is adopted which implies that both position and velocity are recovered after a scaling phase. Finally, concerning the emergency procedure in Figure 3.3, the approach of the EA baseline in Section 3.4.4 is adopted.

Parameters in (3.14) and (3.28) are selected as follows $k_{\sigma,d} = 20$, $k_{\sigma,p} = 100$, $k_{c,d} = 1$, $k_{c,p} = 1$, whereas a constant minimum value $\bar{F}_{min} = 24$ is considered in Problem 3.1, which is tuned on the basis of the *experimented* human level of safety.

Figures 3.13-3.15 report the experimental results. According to the scheme in Figure 3.3, depending on the relative motion of the cooperative robots and the human operator, scaling phases, indicated with S_c, or emergency phases during which the task is aborted, denoted by E_m, occur.

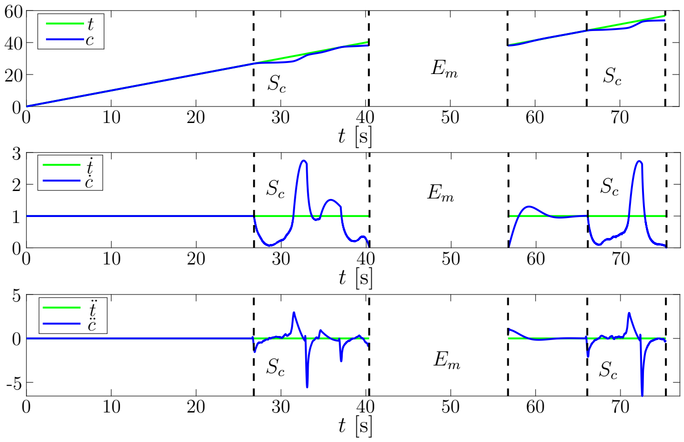


Figure 3.13: Experimental case study. Evolution of the scaling parameters (in blue) with respect to the nominal ones (in green); scaling and emergency phases are highlighted with S_c and E_m, respectively. When the emergency procedure is active (E_m), no plots are provided since the task is aborted.

Figure 3.13 shows how the scaling parameters $c(t)$, $\dot{c}(t)$, $\ddot{c}(t)$ (in blue) vary over time in relation to the nominal values t, $\dot{t} = 1$, $\ddot{t} = 0$ (in green), respectively. In detail, when the first scaling phase S_c starts at time $t \approx 27$ s, a slowing down of the trajectory occurs ($\dot{c}(t) < 1$) and a value close to zero is reached at time $t \approx 28$ s; then, both the nominal velocity and position are restored at time $t \approx 37$ s, i.e., it holds $c(t) = t$ and $\dot{c}(t) = 1$; after this, the velocity starts again to be reduced until time $t \approx 41$ s when the constraint on reverse motion is violated, i.e., $\dot{c}(t)$ reaches a negative value, causing the activation of the emergency procedure. When the latter is active, no evolution of the scaling parameters is provided since the robots task is aborted; then, since it ends with zero velocity (and position equals to that at the aborting time $t \approx 41$ s), the nominal values of the scaling parameters are recovered. A scaling phase also occurs between $t \approx 57$ s and $t \approx 76$ s, where the trajectory is first decelerated, then accelerated and finally again decelerated along the nominal path. In particular, at time $t \approx 75$ s the coefficient $\dot{c}(t)$ reaches a negative value and the task is aborted via a proper emergency procedure. It is worth noticing that, as expected from the update law in (3.28) and differently from simulation results, the scaling procedure always restores both position and velocity to their nominal values.

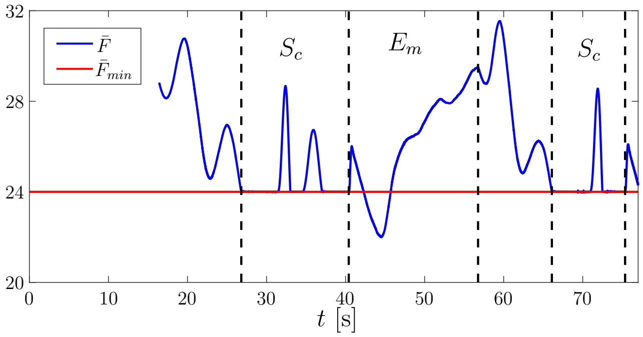


Figure 3.14: Experimental case study. Evolution of the cumulative safety index (in blue) with respect to its minimum allowed value (in red). At time $t \approx 16$ s the human operator enters the work-cell; scaling and emergency phases are marked with S_c and E_m, respectively.

Starting from the time the human enters the work-cell (at $t \approx 16$ s), the evolution of the cumulative safety index $\bar{F}(t)$ (in blue) with respect to its minimum value $\bar{F}_{min}$ (in red) is reported in Figure 3.14. Analogously to the simulative results in Section 3.4, when the scaling procedure is active, the safety index is mostly saturated at the minimum value; whereas when the emergency procedure is active, an increase of the safety index is registered at its beginning due to the relaxation of the constraint on the nominal path maintenance. The threshold is, then, overcome at time $t \approx 42$ s when the robots stop since they reach the boundary of the cooperative workspace and the human operator keeps moving towards them; however, as already stressed in the description of the simulation results, this situation does not endanger the operator because of robots zero velocity. Finally, Figure 3.15 shows the reference trajectory (in blue), obtained as output of the avoidance strategy reported in Figure 3.3, with respect to the nominal one (in green). In particular, the positional part of the absolute task variable $\boldsymbol{\sigma}_1$ (the orientation part and $\boldsymbol{\sigma}_2$ are constant over time) is reported which makes evident the action of the avoidance strategy: during the scaling phase the reference trajectory is mainly slowed down with respect to the nominal one, whereas during the emergency phase, starting from the position at time $t \approx 41$ s, it is modified according to an impedance based dynamics and then, when the safety index allows it, the same position of time $t \approx 41$ s is restored.

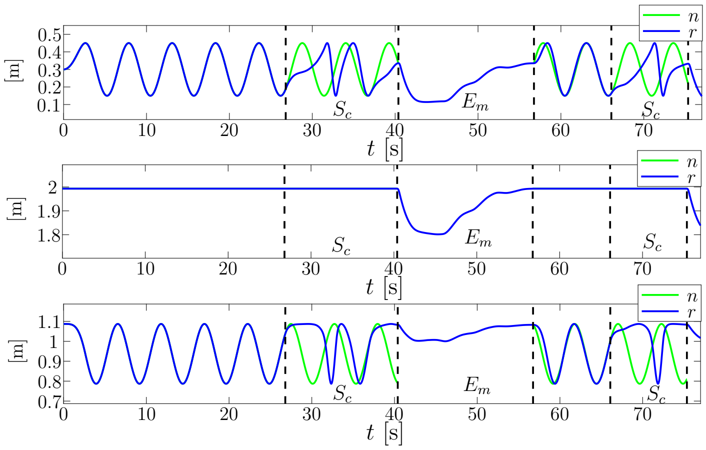


Figure 3.15: Experimental case study. Evolution of the nominal (n, in green) and reference (r, in blue) trajectories of the team centroid position (namely $\boldsymbol{\sigma}_{1,x}$, $\boldsymbol{\sigma}_{1,y}$, $\boldsymbol{\sigma}_{1,z}$); scaling and emergency phases are marked with S_c and E_m, respectively. When the emergency procedure is active (E_m), the nominal trajectory is not shown since its tracking is interrupted.

Chapter 4

Human multi-robot physical collaboration

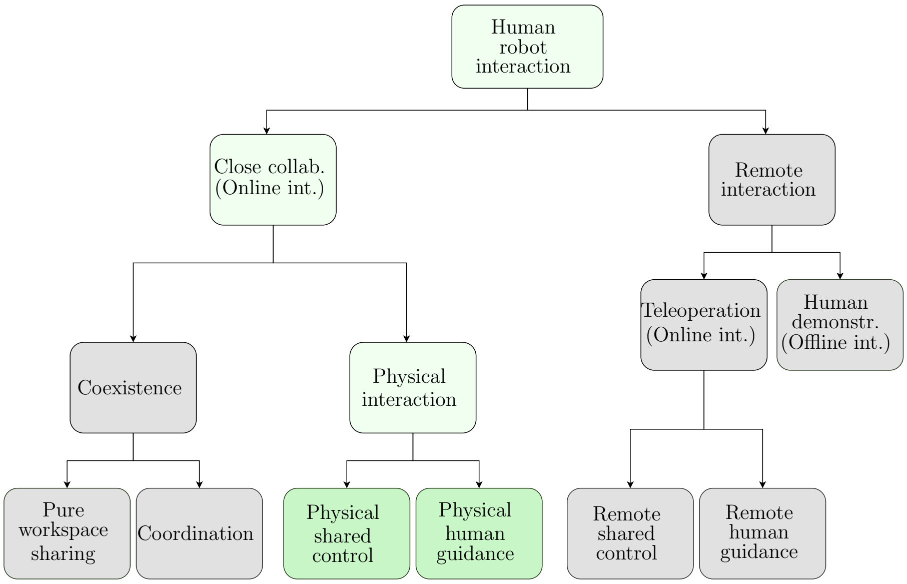


Figure 4.1: Taxonomy for HRI scenarios where the ones considered in this chapter are highlighted in green.

This chapter investigates the scenario of physical collaboration between robots and human operators where intentional exchange of wrenches between them is required to cooperatively carry out assigned tasks. As highlighted in green in the taxonomy in Figure 4.1, two main types of physical interaction are addressed which differ on the human-robot reciprocal role: *(i)* shared control and *(ii)* human guidance, also referred to as human assistance. More specifically, in the former scenario the robot autonomy is preserved to a certain extent and, as it happens in the case of *human-human* interaction, equal roles are attributed to both robotic and human counterparts. In contrast, the latter scenario foresees that the robot is aimed at assisting the human operator regardless of its tasks, i.e., it relies on a leader-follower paradigm in which the human acts as leader, and the robot only plays a follower role aimed at minimizing the leader effort. As discussed in [19] and [83], the type of interaction to be realized in a given application depends on the nature of the application itself and whether it requires the robot to be fully driven by the person or not.

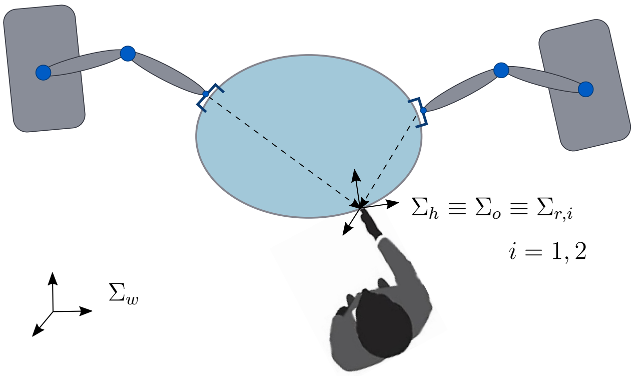


Figure 4.2: Representation of the system in a human multi-robot physical interaction setup. The system is composed of multiple robots co-manipulating a rigid object with a human operator. The reference frames Σ_w, Σ_h, Σ_o and $\Sigma_{r,i}$ are reported.

Moreover, as highlighted throughout the previous chapters, also in this case the collaboration can benefit from the presence of *multiple* robots, which improve the overall system performance in terms of payload and robustness with respect to the single robot case. Therefore, a scenario of physical human multi-manipulator interaction is considered where the human and the robots co-manipulate the same object, as depicted in Figure 4.2. In this context, the following additional issues need to be addressed with respect to the case of single robot single human interaction:

1. since the human does not directly interact with one of the robots but with the co-manipulated object, the human interaction wrench needs to be estimated for pursuing any interaction strategy. The problem is particularly challenging when no force sensors are adopted to measure the human wrench and in absence of a central control unit;

2. internal wrenches are required to be regulated during cooperative manipulation in order not to damage the object and/or the manipulators. In this regard, the computations of internal and the human-robot wrenches are coupled;

3. at low level, specific actions need to be included to synchronize the tracking errors, counteract model uncertainties and reduce internal wrenches in transient phase.

Despite its potentiality, few studies are available in literature in regard to this scenario which this thesis aims to investigate.

Furthermore, when physical human-robot interaction is allowed, a further aspect needs to be taken into account. In particular, both intentional and *accidental* contacts between human and robot might occur due to the complexity of tasks and environment, the uncertainty of human behavior, and the typical lack of awareness of each other actions. The two cases obviously require the robot to adopt different reaction strategies: compliant behaviors in the case of intentional contact, as described in the following sections of the chapter, and avoidance actions in the case of accidental contact, as discussed in Chapter 3. Therefore, in order to undertake a suitable reaction, it is necessary to endow the robot with the ability of detecting the occurrence of a contact and, then, of recognizing its type [84].

This chapter first reports the main relevant related work for physical interaction (Section 4.1) and presents the system modeling in case of co-manipulation of the same object (Section 4.2). Then, strategies for human multi-robot shared control (Section 4.3) and assistance (Section 4.4) addressing the issues 1-3 are presented. Finally, the problem of detecting and classifying contacts (Section 4.5) is tackled.

4.1 Literature review

4.1.1 Human-robot physical interaction

Whatever the specific interaction (shared control or human assistance), the primary requirement in a physical interaction scenario is to carry out a careful control of the interaction forces so as not to cause damage to the human operator. In this regard, admittance or impedance controllers [85] have laid the foundations as they confer a compliant behavior to the robot, which is generally rigid from a mechanical point of view. This is achieved by imposing a mass-spring-damper dynamics to the robot end effector or, in general, to its structure without the need of integrating ad-hoc mechanical components such as in [86]. Starting from these controllers, several strategies have been then developed in order to improve the interaction to a certain extent. As a common approach, variable admittance controllers have been proposed in literature in which admittance parameters, i.e., mass-spring-damper coefficients, are dynamically adapted on the basis of some aspects of the interaction. Stability issues arising in the case the human operator stiffens his/her arm are addressed, for instance, in [87, 88, 89]. More specifically, an instability index is proposed in [87] in which the high-frequency oscillations of the human force are interpreted as a measure of an unstable behavior of the coupled system and is exploited for varying the damping and inertia parameters. Differently, energy tanks theory is leveraged in [88] to guarantee passivity of the overall system at all times, whereas an online estimate of the human stiffness is proposed in [89] to adapt the admittance damping accordingly.

Furthermore, depending on the interaction to be accomplished, specific strategies can be included. For example, the estimation of human intentions in assistance tasks can be exploited to enable the robot not only to passively comply to the interaction forces but also to proactively participate to the motion. In this case, the robot reference trajectory is adjusted on the basis of the estimate of the human intended motion in order to minimize his/her effort. This approach is pursued in [90] where Neural Networks (NNs) are adopted for estimating the human desired trajectory, which is then used as reference trajectory in the spring term of an admittance controller. Alternatively, authors of [91] build on the assumption that human point-to-point motion obey to a minimum jerk trajectory and online estimate the parameters of the latter. Human prediction uncertainty is then explicitly taken into account in [92], where human behavior is modeled as a Gaussian Process and a risk-sensitive optimal control problem is formalized. Finally, the approach presented in [93] generates the admittance parameters as the output of a neural network trained to minimize the interaction forces. Differently from the above methods and as detailed in Section 4.4, the human assistance problem is *formally* stated and solved in the following and a multi-robot setup is considered.

Concerning the shared control paradigm, it arises in teleoperation scenarios [83] where the human operator typically provides control inputs via haptic interface, while the robotic system preserves autonomous behaviors, as instance, for collision avoidance [94]. More recently, this paradigm has also been successfully explored in the context of physical human-robot interaction. A solution based on game theory is presented in [95] where human and robot are assumed to optimize the same cost function. In the proposed approach, the human and robot roles are continuously adapted on the basis of the human exerted force: the higher the force, the higher the influence of the human intended motion. The work in [96] proposes to achieve the same behavior by deforming the robot trajectory in dependence of the exerted human forces and formulates a constrained optimization problem to the purpose. A heuristic agreement index is designed in [97] so that, if the forces of robot and human agree, the robot acts as leader, otherwise as follower. Finally, a solution based on data-driven stochastic modeling is devised in [92] where a risk sensitive optimization problem is defined to tackle the uncertainty in the human behavior model. Note that all the aforementioned works but [97] refer to the interaction with a single robot and heuristics are generally employed. As detailed in the following, the solution devised in Section 4.3 aims to overcome the limitations of existing approaches by defining a framework for multi-robot systems based on a novel Linear Quadratic Tracking problem taking into account human and robot intentions.

4.1.2 Contact detection and classification

For the purpose of detecting and recognizing the nature of human contact, the dynamical model of the robot and joint torque sensors, as in [98], or motor currents as in [99] are generally exploited. Other approaches, like the one in [100], leverage the different frequency characteristics of accidental and intentional contacts to achieve classification in the frequency domain. Additional solutions might also leverage proper artificial skins for robots as in [101, 102].

Classical approaches, like the ones cited above, requires thresholds to be manually tuned in order to take into account sensor noise and different type of contacts, which results in poor flexibility and robustness of the overall system. For this reason, data-driven approaches have been devised in recent years for contact detection and classification, due to their flexibility and capability of handling the non-linearity and variety of the human-robot contact. In [103] and [104], NNs are used to detect sole accidental collisions on the basis of data coming from joint torque sensors. The same objective is achieved in [105] by using a deep learning approach which requires the tuning of a moving time window. The study in [106] uses Convolutional Neural Networks (CNNs) to detect also intentional contacts on the basis of joint velocity and external torque data in a moving time window, but the approach limited to the upper and lower parts of the robot. A hybrid solution for classification is instead presented in [107] where a thresholding method is considered to detect contact, while Support-Vector Machines are adopted to recognize the type of interaction. Methods based on thresholding and neural networks are compared in [108] to detect and localize the contact and a frequency-based neural network is defined to distinguish contact with hard or soft environment. Finally, a control low combining tracking performance and compliant behavior is presented in [109] where possible external impacts are explicitly taken into account. However, no distinction is made on the type of contact, whether intentional or not.

Note that the above approaches consider that the only source of interaction of the robot with the environment is given by the human operator, thus ignoring the case in which the robot does not work in free space but needs to interact with the environment to perform its task. Moreover, no possible constraints on the robot motion are taken into account. The solution proposed in Section 4.5 tries to overcome these limitations by defining a general framework capable of dealing with robotic tasks that involve interaction with the environment as well as with robotic constraints that can be task-dependent.

4.2 Physical interaction system modeling

This section provides the model of the human arm end-point as well as of the co-manipulated object. Consider a system composed of N serial-chain mobile manipulators which tightly grasp a rigid object and a human operator that co-manipulates the same object as in Figure 4.2. In the following, the term *human interaction* is referred to the case in which the human operator exerts, through his/her hand, forces on the object that is tightly co-manipulated by the multi-robot system. In this way, the human is able to modify the object motion according to his/her desired motion. In the figure, the following reference frames are defined:

- Σ_w is the world reference frame;
- Σ_o is the object reference frame;
- $\Sigma_{r,i}$ is the reference frame attached to the end effector of the i th robot;
- Σ_h is the reference frame attached to the human arm end point.

Without loss of generality, the following assumption is considered in the rest of the chapter.

Assumption 4.1. *Robot kinematics and dynamics in the operational space are referred to the human arm end point frame Σ_h (i.e., $\Sigma_{r,i} \equiv \Sigma_h$, $\forall\, i$). Analogously, the object dynamics is referred to the same frame Σ_h (i.e., $\Sigma_o \equiv \Sigma_h$).*

The above assumption is not unrealistic in many practical scenarios since the geometry of the object is known or can be estimated beforehand.

4.2.1 Human arm end point modeling

The model assumed for the human arm end point, which will be referred to as human model, is

$$-\boldsymbol{D}_h \dot{\boldsymbol{p}}_h + \boldsymbol{K}_h(\boldsymbol{p}_{h,d} - \boldsymbol{p}_h) = \boldsymbol{f}_h \tag{4.1}$$

where $\boldsymbol{f}_h \in \mathbb{R}^m$ is the vector of forces exerted by the human, $\boldsymbol{p}_h \in \mathbb{R}^m$ is the human arm end point position, which will be referred to as human position, with respect to Σ_w, $\boldsymbol{p}_{h,d} \in \mathbb{R}^m$ is the human desired position and $\boldsymbol{D}_h \in \mathbb{R}^{m \times m}$ and $\boldsymbol{K}_h \in \mathbb{R}^{m \times m}$ are the matrices that regulate the human damping and stiffness actions, respectively. As stated, for example in [110] and [111], these matrices are generally time-varying and their variation depends on the activation of the human arm muscles. The model in (4.1) is commonly assumed as representative for the human arm end point, e.g. in [89, 90, 112], and its validation can be found, for instance, in [110, 113] with the inertial term shown to be negligible compared to damping and stiffness. Moreover, by following the assumptions in [89, 112, 114], the decoupling of the impedance parameters in the different directions is assumed, i.e., $\boldsymbol{D}_h = \mathrm{diag}(\boldsymbol{d}_h)$ and $\boldsymbol{K}_h = \mathrm{diag}(\boldsymbol{k}_h)$ with $\boldsymbol{d}_h \in \mathbb{R}^m$ and $\boldsymbol{k}_h \in \mathbb{R}^m$. Then, the model in (4.1) can be rewritten as

$$\boldsymbol{Y}_h(\boldsymbol{p}_h, \dot{\boldsymbol{p}}_h)\boldsymbol{\pi}_h = \boldsymbol{f}_h \tag{4.2}$$

with

$$\begin{aligned} \boldsymbol{Y}_h &= [-\mathrm{diag}(\dot{\boldsymbol{p}}_h) \quad -\mathrm{diag}(\boldsymbol{p}_h) \quad \boldsymbol{I}_m] \in \mathbb{R}^{m\times 3m} \\ \boldsymbol{\pi}_h &= [\boldsymbol{d}_h^T \quad \boldsymbol{k}_h^T \quad \boldsymbol{p}_{h,d}^T\boldsymbol{K}_h]^T \in \mathbb{R}^{3m}. \end{aligned} \tag{4.3}$$

It is worth remarking that the human parameters in $\boldsymbol{\pi}_h$ are *unknown* and might be time-varying.

Finally, the human wrench vector is defined as $\boldsymbol{h}_h = \boldsymbol{G}_h\boldsymbol{f}_h \in \mathbb{R}^p$ with $\boldsymbol{G}_h = [\boldsymbol{I}_m \quad \boldsymbol{O}_m]^T \in \mathbb{R}^{p\times m}$.

4.2.2 Object modeling

Considering the scenario in Figure 4.2, the dynamics of the rigid object can be derived as follows

$$\boldsymbol{M}_o\ddot{\boldsymbol{x}}_o + \boldsymbol{C}_o(\boldsymbol{x}_o, \dot{\boldsymbol{x}}_o)\dot{\boldsymbol{x}}_o + \boldsymbol{g}_o = \sum\nolimits_{i=1}^{N} \boldsymbol{G}_i\boldsymbol{h}_i + \boldsymbol{h}_h \tag{4.4}$$

where $\boldsymbol{x}_o = \begin{bmatrix}\boldsymbol{p}_o^T & \boldsymbol{\phi}_o^T\end{bmatrix}^T \in \mathbb{R}^p$ represents the configuration in terms of position $\boldsymbol{p}_o$ and orientation $\boldsymbol{\phi}_o$ of the object reference frame Σ_o with respect to Σ_w, $\boldsymbol{M}_o \in \mathbb{R}^{p\times p}$ is the inertia matrix, $\boldsymbol{C} \in \mathbb{R}^{p\times p}$ is the matrix of centrifugal and Coriolis terms, $\boldsymbol{g}_o \in \mathbb{R}^p$ is the vector of the gravity terms, $\boldsymbol{G}_i \in \mathbb{R}^{p\times p}$ is the grasping matrix associated with the ith robot for which, in light of Assumption 4.1, it holds $\boldsymbol{G}_i = \boldsymbol{I}_p$, $\forall i$.

By denoting with $\boldsymbol{h}$ the stacked vector of the interaction wrenches and with $\boldsymbol{G}$ the collective grasping matrix, i.e.,

$$\boldsymbol{h} = [\boldsymbol{h}_1^T \ \dots \ \boldsymbol{h}_N^T]^T \in \mathbb{R}^{Np}, \ \boldsymbol{G} = [\boldsymbol{G}_1 \ \dots \ \boldsymbol{G}_N] \in \mathbb{R}^{p\times Np}$$

the interaction wrenches can be divided into external $\boldsymbol{h}_e \in \mathbb{R}^{Np}$ and internal $\boldsymbol{h}_{int} \in \mathbb{R}^{Np}$ components [78], i.e., those that contribute and do not contribute, respectively, to the object motion for which it holds

$$\boldsymbol{h} = \boldsymbol{h}_e + \boldsymbol{h}_{int} = \boldsymbol{G}^{\dagger}\boldsymbol{G}\boldsymbol{h} + (\boldsymbol{I}_{Np} - \boldsymbol{G}^{\dagger}\boldsymbol{G})\boldsymbol{h}. \tag{4.5}$$

Finally, the following assumption is considered in the chapter.

Assumption 4.2. *The kinematic and dynamic parameters of the manipulated object model are known.*

This hypothesis can be easily overcome by resorting to *ad-hoc* techniques that allow to distributively estimate these parameters by properly interacting with the object like in [115, 116].

4.3 Shared-control scenario

The first human multi-robot physical interaction scenario that is analyzed is the one of shared control, in which, as shown in Figure 4.2, humans and robots physically interact through the comanipulation of a rigid object and peer roles are preserved between these entities. This interaction, as previously discussed, resembles the one that occur in human-human physical interactions. In order to realize this shared control, this section presents a framework, based on [117], which resorts to an optimization formulation and leverages the human model to online estimate its parameters.

More specifically, based on the above modeling, the following problem is formally addressed.

Problem 4.1. *Consider a multi-robot system composed of N manipulators rigidly grasping an object which a human operator is interacting with. Assume that a central unit is not available and that a robots' desired trajectory $\boldsymbol{x}_{r,d} \in \mathbb{R}^p$ (with derivative $\dot{\boldsymbol{x}}_{r,d} \in \mathbb{R}^p$) is assigned to the object as well as a desired internal wrench $\boldsymbol{h}_{int}^d \in \mathbb{R}^{Np}$ is given. The objective is to design the robot control input $\boldsymbol{u}_i$ ($i = 1, 2, \ldots, N$) in* (2.10) *such that the manipulated* *object behaves according to the following dynamics*

$$\boldsymbol{M}_v \ddot{\boldsymbol{x}}_v = \boldsymbol{u}_v \tag{4.6}$$

where $\boldsymbol{M}_v \in \mathbb{R}^{p\times p}$ *is the virtual inertia matrix,* $\boldsymbol{x}_v = \begin{bmatrix}\boldsymbol{p}_v^T & \boldsymbol{\phi}_v^T\end{bmatrix}^T \in \mathbb{R}^p$ *is the configuration of the virtual object and* $\boldsymbol{u}_v \in \mathbb{R}^p$ *represents the virtual input to be defined so as to optimize the following cost function*

$$\begin{aligned} J = \frac{1}{2}\int_{t_0}^{+\infty} \Big(& \underbrace{(\bar{\boldsymbol{x}}_v - \bar{\boldsymbol{x}}_{r,d})^T \boldsymbol{Q}_{r,d} (\bar{\boldsymbol{x}}_v - \bar{\boldsymbol{x}}_{r,d})}_{i} + \underbrace{\boldsymbol{u}_v^T \boldsymbol{R}_v \boldsymbol{u}_v}_{ii} \\ & + \underbrace{(\bar{\boldsymbol{x}}_v - \bar{\boldsymbol{x}}_{h,d})^T \boldsymbol{Q}_{h,d} (\bar{\boldsymbol{x}}_v - \bar{\boldsymbol{x}}_{h,d})}_{iii} + \underbrace{\boldsymbol{f}_h^T \boldsymbol{R}_h \boldsymbol{f}_h}_{iv} \Big) dt \end{aligned} \tag{4.7}$$

where $\bar{\boldsymbol{x}}_v = \begin{bmatrix}\boldsymbol{x}_v^T & \dot{\boldsymbol{x}}_v^T\end{bmatrix}^T \in \mathbb{R}^{2p}$ *is the aggregate vector of state,* $\bar{\boldsymbol{x}}_{r,d} = \begin{bmatrix}\boldsymbol{x}_{r,d}^T & \dot{\boldsymbol{x}}_{r,d}^T\end{bmatrix}^T \in \mathbb{R}^{2p}$ *and* $\bar{\boldsymbol{x}}_{h,d} = \begin{bmatrix}\boldsymbol{p}_{h,d}^T & \boldsymbol{0}_{2p-m}^T\end{bmatrix}^T \in \mathbb{R}^{2p}$ *are the aggregate vectors of the robots and human desired trajectory, respectively,* $\boldsymbol{R}_h \in \mathbb{R}^{m\times m}$ *and* $\boldsymbol{R}_v \in \mathbb{R}^{p\times p}$ *are symmetric positive definite matrices,* $\boldsymbol{Q}_{r,d} \in \mathbb{R}^{2p\times 2p}$ *is a symmetric positive semi-definite matrix,* $\boldsymbol{Q}_{h,d} \in \mathbb{R}^{2p\times 2p}$ *is defined as*

$$\boldsymbol{Q}_{h,d} = \boldsymbol{S}^T \boldsymbol{W}_{h,d} \boldsymbol{S}, \quad \boldsymbol{S} = \begin{bmatrix} \boldsymbol{I}_m & \boldsymbol{O}_{m\times 2p-m} \\ \boldsymbol{O}_{m\times 2p-m} & \boldsymbol{I}_m \end{bmatrix} \tag{4.8}$$

being $\boldsymbol{W}_{h,d} \in \mathbb{R}^{2m\times 2m}$ *a symmetric positive semi-definite weighting matrix.*
In addition, internal wrench regulation is required, i.e., $\boldsymbol{h}_{int} \to \boldsymbol{h}_{int}^d$.

The rationale behind (4.7) is to continuously combine the robots desired trajectory for the object (i) with the human desired one (iii) and, similarly, to x the effort of the robots (ii) and of the human (iv) on the basis of the respective weighting matrices. This means that by increasing $\boldsymbol{Q}_{h,d}$ and $\boldsymbol{R}_h$ with respect to $\boldsymbol{Q}_{r,d}$ the human intention overtakes the robots one (and vice-versa) according to an optimal formulation. In addition, the proposed approach is also suitable for an assistance task, whose aim is to minimize the human effort, by setting $\boldsymbol{Q}_{r,d} = \boldsymbol{O}_{2p}$.

In the above formulation, the human intention in terms of object orientation is not taken into account which means that robots have full control on it. Like in [118], this is motivated by the fact that the human is in charge of modifying the object motion while robots autonomously and precisely control the orientation.

In what follows, first a centralized solution to the Problem 4.1 is provided in Section 4.3.1; then, building on this, a distributed solution to the same problem is presented in Section 4.3.2.

4.3.1 Linear Quadratic Tracking approach

A two-layer architecture depicted in Figure 4.3 is devised in the case of *centralized* solution. In detail, the high-level is in charge of defining the object reference trajectory $\boldsymbol{x}_v$ as solution of the optimal problem in (4.7), on the basis of human intentions, and the low-level of defining the local control inputs $\boldsymbol{u}_i$ in (2.10) to actually track the trajectory while achieving wrench regulation.

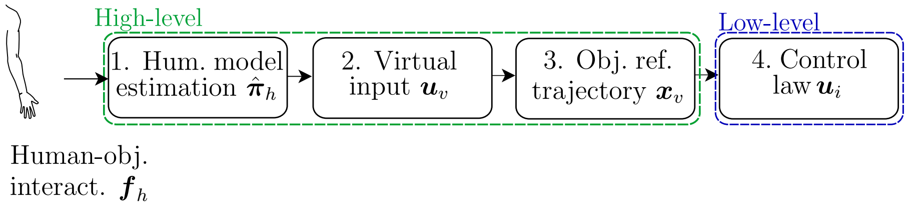

High-level
Low-level
1. Hum. model estimation $\hat{\boldsymbol{\pi}}_h$
2. Virtual input $\boldsymbol{u}_v$
3. Obj. ref. trajectory $\boldsymbol{x}_v$
4. Control law $\boldsymbol{u}_i$
Human-obj. interact. $\boldsymbol{f}_h$

Figure 4.3: Two-layer architecture for human multi-robot shared control.

Concerning the high-level, the formulation in (4.6) requires the virtual input $\boldsymbol{u}_v$ to be designed (block 2) so as to optimize (4.7). By replacing the human model (4.1) in the cost function (4.7), the latter can be rewritten as

$$J = \frac{1}{2}\int_{t_0}^{+\infty} \Big((\bar{\boldsymbol{x}}_v - \bar{\boldsymbol{x}}_{r,d})^T \boldsymbol{Q}_{r,d} (\bar{\boldsymbol{x}}_v - \bar{\boldsymbol{x}}_{r,d}) + \boldsymbol{u}_v^T \boldsymbol{R}_v \boldsymbol{u}_v + (\bar{\boldsymbol{x}}_v - \bar{\boldsymbol{x}}_{h,d})^T \bar{\boldsymbol{Q}}_h (\bar{\boldsymbol{x}}_v - \bar{\boldsymbol{x}}_{h,d}) \Big) dt \tag{4.9}$$

with $\bar{\boldsymbol{Q}}_h = \boldsymbol{S}^T \bar{\boldsymbol{W}}_{h,d} \boldsymbol{S}$ and

$$\bar{\boldsymbol{W}}_{h,d} = \boldsymbol{W}_{h,d} + \begin{bmatrix} \boldsymbol{K}_h^T \boldsymbol{R}_h \boldsymbol{K}_h & \boldsymbol{K}_h^T \boldsymbol{R}_h \boldsymbol{D}_h \\ \boldsymbol{D}_h^T \boldsymbol{R}_h \boldsymbol{K}_h & \boldsymbol{D}_h^T \boldsymbol{R}_h \boldsymbol{D}_h \end{bmatrix}. \tag{4.10}$$

It is worth noticing that, according to (4.10), the weight associated with the human desired motion dynamically adapts on the basis of the human impedance parameters $\boldsymbol{K}_h$ and $\boldsymbol{D}_h$, e.g., when the human stiffens the arm (i.e., $\boldsymbol{K}_h$ increases) the corresponding quantities in $\bar{\boldsymbol{Q}}_h$ *automatically* increase as well. This allows for a dynamic roles adaptation between human and robots. However, as shown in the simulation results in Section 4.5.4, the mutual roles also depend on the value of the weight matrix $\boldsymbol{Q}_{r,d}$, i.e., the higher it is, the more the tracking error of the robots is being minimized and they assume a leading role. A similar reasoning also applies in regard to the magnitude of the robot tracking error itself, i.e., assuming all other parameters are equal, the higher it is, the more the resulting motion will follow the robots desired trajectory.
The formulation in (4.9)-(4.10) shows the dependence of the objective function on the *unknown* human parameters $\boldsymbol{K}_h, \boldsymbol{D}_h, \boldsymbol{p}_{h,d}$, that is on $\boldsymbol{\pi}_h$ in (4.3). Therefore, an online estimation of these parameters is needed by exploiting the knowledge of human force and motion, i.e., $\boldsymbol{f}_h$, $\boldsymbol{p}_h$, $\dot{\boldsymbol{p}}_h$ (block 1 in Figure 4.3), and is then used to find $\boldsymbol{u}_v$ which minimizes (4.9). Afterwards, the virtual object trajectory $\boldsymbol{x}_v$ is computed according to the dynamics in (4.6) (block 3) and is adopted by the robots as actual reference trajectory for the object. In particular, a standard control law for trajectory tracking with internal force regulation as in [119] can be leveraged at this point for the low-level *centralized* solution (block 4) and details are here omitted since a distributed solution to this problem is presented in next sections.

The presented method introduces the following contributions with respect to the state of the art:

- a novel shared control strategy in the framework of optimal control is presented. The role of the robots is dynamically adapted depending on the estimation of the human arm parameters and takes into account the uncertainty of the latter;

- the case of multi-manipulator systems is tackled for the first time. This case poses additionally issues with respect to the single manipulator case, due to the necessity to handle internal wrenches in addition to external one in order to no affect the human-object interaction dynamics. Moreover, a sensorless solution is adopted in which both the interaction with the human and internal wrenches are estimated by resorting to a momentum based approach;
- the solution presented is completely decentralized since the local control strategy only relies on information locally available.

Finally, note that the approach can be applied to any cooperative manipulation task which involves the specification of an object trajectory. This includes, by resorting to the list of applications identified in the survey in [5], applications such as handling, assembly and welding in any of which the human operator may wish to intervene during the robots' activity to correct the task or to participate in it and, to this end, the proposed shared control strategy can be applied. In the following, the individual blocks in Figure 4.3 are detailed.

Human parameters estimation

A Recursive Least Square method with forgetting factor is proposed to estimate the human arm parameter $\boldsymbol{\pi}_h$. Consider that the measurements are acquired at each kT, with $k \in \mathbb{N}$ the discrete time index and $T \in \mathbb{R}$ the sampling time, and denote with $(\cdot)_k$ the corresponding quantity at time kT. The estimation error $\boldsymbol{e}_{h,k} \in \mathbb{R}^m$ is defined as

$$\boldsymbol{e}_{h,k} = \boldsymbol{f}_{h,k} - \boldsymbol{Y}_{h,k}\hat{\boldsymbol{\pi}}_{h,k\text{-}1} \tag{4.11}$$

where $\boldsymbol{Y}_{h,k}$ is the human regressor defined in (4.2) and $\hat{\boldsymbol{\pi}}_{h,k} \in \mathbb{R}^{3m}$ is the estimate of the unknown human parameters.

Note that the computation of (4.11) requires the knowledge of $\boldsymbol{f}_h$ which, in the centralized case, can be retrieved from the collective wrench $\boldsymbol{h}$ and the knowledge of the object dynamics in (4.4) as in [98].

By leveraging [120], the parameters estimate is updated as

$$\hat{\boldsymbol{\pi}}_{h,k} = \hat{\boldsymbol{\pi}}_{h,k-1} + \boldsymbol{L}_k \boldsymbol{e}_{h,k} \tag{4.12}$$

where the matrices $\boldsymbol{L}_k \in \mathbb{R}^{3m \times m}$ and $\boldsymbol{P}_k \in \mathbb{R}^{3m \times 3m}$ are

$$\begin{aligned} \boldsymbol{L}_k &= \boldsymbol{P}_{k-1} \boldsymbol{Y}_{h,k}^T (\lambda \boldsymbol{I}_m + \boldsymbol{Y}_{h,k} \boldsymbol{P}_{k-1} \boldsymbol{Y}_{h,k}^T)^{-1} \\ \boldsymbol{P}_k &= \frac{1}{\lambda} (\boldsymbol{I}_{3m} - \boldsymbol{L}_k \boldsymbol{Y}_{h,k}) \boldsymbol{P}_{k-1} \end{aligned} \tag{4.13}$$

being $\lambda \in (0,\, 1]$ the forgetting factor which regulates the amount of data to be forgotten at each estimate update: the lower the forgetting factor, the lower the weight associated with past inputs. This implies that the lower the forgetting factor, the more the estimation is able to track changes in the parameters but the more the misadjustment and the possible instability [120]. Implementation details of (4.12) and (4.13) are not addressed here but a detailed discussion can be found in [121].
However, in order for the estimate $\hat{\boldsymbol{\pi}}_h$ in (4.12) to converge to the real values $\boldsymbol{\pi}_h$, the estimator input variables, i.e., $\boldsymbol{p}_h$ and $\dot{\boldsymbol{p}}_h$, have to satisfy the persistence of excitation condition for which the approximation proposed in [122] is here considered, that is the following condition should be verified

$$\|\boldsymbol{P}_{k-1} \boldsymbol{Y}_{h,k}\| > \alpha, \quad \forall k \text{ with } \alpha \in \mathbb{R}^+. \tag{4.14}$$

Moreover, by leveraging the results in [112], a finite-time interval with predetermined duration Δ, during which the input signals are excited, allows the parameters to be estimated with a certain tolerance. For further details, the reader is referred to [112] and [122]. Note that, as also pointed out in [123], the condition of persistent excitation may generally not be met by human motion. To mitigate this issue, ad-hoc motions of very short du-

ration [112], unnoticeable to the human, can be induced by the robotic component on the co-manipulated object. Moreover, even if the persistent excitation condition is not achieved, the proposed solution does not harm the human operator since, as described below, it discards the human estimates if they are not sufficiently reliable and a safe by construction interaction is enforced.
Based on the above considerations, a confidence index $\mathcal{I}_c$ is devised to establish the reliability of the human parameters estimate. The basic idea is that the latter is assumed reliable if there exists a time interval for which the system is "sufficiently" excited and as long as the estimation error is below a certain threshold, which leads to the following index

$$\mathcal{I}_c(\hat{\boldsymbol{\pi}}_{h,k}) = \begin{cases} 1, \text{if } \|\boldsymbol{e}_{h,k}\| < \bar{e} \wedge \exists\, k_1, k_2 \in \mathbb{N} : T|k_2 - k_1| \geq \Delta \\ \qquad\qquad\qquad \wedge\ \forall\, k \in [k_1,\, k_2] \text{ eq. (4.14) holds} \\ 0, \text{otherwise} \end{cases} \tag{4.15}$$

with $\bar{e} \in \mathbb{R}$ a positive constant representing the maximum allowed estimation error. Finally, it is worth remarking that any other technique than RLS, as the approach in [124], might be used in block 1 of Figure 4.3.

Virtual input $\boldsymbol{u}_v$

The confidence index $\mathcal{I}_c$ specifies whether the human parameter estimation is reliable or not (as in the initial transient phase). Therefore, the latter case needs to be explicitly handled in order to avoid undesired interaction behavior.

In detail, the virtual input is chosen on the basis of the value of the confidence index, as reported in the Finite State Machine (FSM) in Figure 4.4: as long as the estimation is not reliable ($\mathcal{I}_c = 0$), an admittance model is used which simply makes the object compliant towards the human force; then, when the estimation is assessed as reliable ($\mathcal{I}_c = 1$), it is used to solve a Linear Quadratic Tracking (LQT) problem.
Admittance model

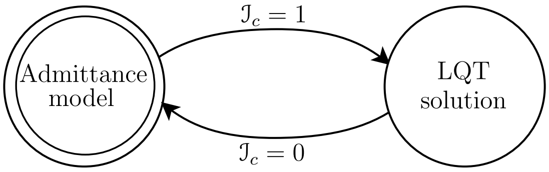


Figure 4.4: FSM for virtual input definition.

When $\mathcal{I}_c(\hat{\boldsymbol{\pi}}_{h,k}) = 0$, the estimate $\hat{\boldsymbol{\pi}}_{h,k}$ is discarded and full control of the system is given to the human operator (instead of minimizing (4.9)) by setting $\boldsymbol{u}_v$ in (4.6) as

$$\boldsymbol{u}_v = -\boldsymbol{D}_v \dot{\boldsymbol{x}}_v + \boldsymbol{\gamma}(\boldsymbol{h}_h)$$

which leads to the following virtual model

$$\boldsymbol{M}_v \ddot{\boldsymbol{x}}_v + \boldsymbol{D}_v \dot{\boldsymbol{x}}_v = \boldsymbol{\gamma}(\boldsymbol{h}_h) \tag{4.16}$$

where $\boldsymbol{D}_v \in \mathbb{R}^{p \times p}$ represents a damping matrix and $\boldsymbol{\gamma}(\cdot) : \mathbb{R}^p \to \mathbb{R}^p$ is a function of the human wrench possibly chosen as a deadzone function with threshold $t_h \in \mathbb{R}^+$, that is

$$\boldsymbol{\gamma}(\boldsymbol{h}_h) = \begin{cases} \boldsymbol{K}_\gamma \big(\boldsymbol{h}_h - t_h \dfrac{\boldsymbol{h}_h}{\|\boldsymbol{h}_h\|}\big) & \text{if } \|\boldsymbol{h}_h\| > t_h \\ \boldsymbol{0}_p & \text{if } \|\boldsymbol{h}_h\| \leq t_h \end{cases} \tag{4.17}$$

with $\boldsymbol{K}_\gamma \in \mathbb{R}^{p \times p}$ a positive definite matrix. Dynamics in (4.16) allows the human to move the object with the robots executing no autonomous tasks.

LQT solution
When $\mathcal{I}_c(\hat{\boldsymbol{\pi}}_{h,k}) = 1$, the human estimate is exploited to minimize the cost in (4.9). To this aim, the following lemma holds true.

Lemma 4.1. *Consider a time-invariant linear system*

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \tag{4.18}$$

with $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{u} \in \mathbb{R}^m$ the state and the input of the system, *and* $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ *and* $\boldsymbol{B} \in \mathbb{R}^{n\times m}$ *the dynamic and input matrices, respectively. Consider the following infinite-horizon cost function*

$$J=\frac{1}{2}\int_{t_0}^{\infty}\Big((\boldsymbol{x}-\boldsymbol{r}_1)^T\boldsymbol{Q}_1(\boldsymbol{x}-\boldsymbol{r}_1)+(\boldsymbol{x}-\boldsymbol{r}_2)^T\boldsymbol{Q}_2(\boldsymbol{x}-\boldsymbol{r}_2)+\boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u}\Big)dt \tag{4.19}$$

with $\boldsymbol{r}_1 \in \mathbb{R}^n$ *and* $\boldsymbol{r}_2 \in \mathbb{R}^n$ *reference signals,* $\boldsymbol{Q}_1 \in \mathbb{R}^{n\times n}$ *and* $\boldsymbol{Q}_2 \in \mathbb{R}^{n\times n}$ *symmetric semi-definite positive matrices and* $\boldsymbol{R} \in \mathbb{R}^{m\times m}$ *a symmetric definite positive matrix. If the pair* $(\boldsymbol{A}, \boldsymbol{B})$ *is reachable and the pair* $(\boldsymbol{A}, [\boldsymbol{Q}_1\ \boldsymbol{Q}_2]^T)$ *is observable, then the optimal control input minimizing* (4.19) *is*

$$\boldsymbol{u} = -\boldsymbol{K}\boldsymbol{x} + \boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{v} \tag{4.20}$$

where $\boldsymbol{K} \in \mathbb{R}^{m\times n}$ *is a gain matrix defined as*

$$\boldsymbol{K} = \boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{T} \tag{4.21}$$

with $\boldsymbol{T} \in \mathbb{R}^{n\times n}$ *solution of the following equation*

$$\boldsymbol{T}\boldsymbol{A} + \boldsymbol{A}^T\boldsymbol{T} - \boldsymbol{T}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{T} + \boldsymbol{Q}_1 + \boldsymbol{Q}_2 = \boldsymbol{O}_n$$

and $\boldsymbol{v} \in \mathbb{R}^n$ *is an auxiliary signal that evolves according to*

$$-\dot{\boldsymbol{v}} = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})^T\boldsymbol{v} + \boldsymbol{Q}_1\boldsymbol{r}_1 + \boldsymbol{Q}_2\boldsymbol{r}_2. \tag{4.22}$$

Proof. The result follows by considering an aggregate weighting matrix $\bar{\boldsymbol{Q}} = [\boldsymbol{Q}_1\ \boldsymbol{Q}_2]^T \in \mathbb{R}^{2n\times n}$ with the aggregate reference signal $\bar{\boldsymbol{r}} = [\boldsymbol{r}_1^T\ \boldsymbol{r}_2^T]^T \in \mathbb{R}^{2n}$ and by extending the reasoning presented in [125] for the LQT problem. ■

By recalling (4.6), the state vector $\bar{\boldsymbol{x}}_v$ evolves according to

$$\dot{\bar{\boldsymbol{x}}}_v = \begin{bmatrix}\boldsymbol{O}_p & \boldsymbol{I}_p\\ \boldsymbol{O}_p & \boldsymbol{O}_p\end{bmatrix}\bar{\boldsymbol{x}}_v + \begin{bmatrix}\boldsymbol{O}_p\\ \boldsymbol{M}_v^{-1}\end{bmatrix}\boldsymbol{u}_v. \tag{4.23}$$

By noticing that (4.9) and (4.23) are in the form of (4.19) and (4.18), respectively, and that reachability and observability conditions are fulfilled, Lemma 4.1 applies. The optimal virtual input $\boldsymbol{u}_v$ for (4.9) can be thus defined according to (4.20), by considering the estimated human parameters $\hat{\boldsymbol{\pi}}_h$ in place of the respective values in the cost function (4.9).

Finally, recall that, given the virtual input $\boldsymbol{u}_v$, the object reference trajectory $\boldsymbol{x}_v$ is computed according to eq. (4.6) (block 3).

4.3.2 Distributed extension

Based on the centralized solution in Section 4.3.1, the *distributed* control framework to solve Problem 4.1 is now presented. The basic idea is to provide a decentralized implementation of the centralized architecture in Figure 4.3 by estimating the needed *global* information, and without changing the human-object interaction behavior regulated by (4.9). As shown in the following, since each robot has only access to the local wrench $\boldsymbol{h}_i$ and the human operator interacts with the object co-manipulated by *all* the robots, the main global quantities to be estimated are (i) the measure of the force $\boldsymbol{f}_h$ exerted by the human operator and (ii) the contribution of each robot to the internal wrenches $\boldsymbol{h}_{int}$ in (4.5). Moreover, concerning the human-robot interaction strategy, a leader-follower approach is proposed in which the leader, based on the estimation of $\boldsymbol{f}_h$, executes the high-level strategy of the architecture in Figure 4.3 and computes the optimal object reference trajectory $\boldsymbol{x}_v$, $\dot{\boldsymbol{x}}_v$, as in the centralized counterpart, while the followers estimate this trajectory via decentralized observers. These quantities feed a low-level adaptive control law which allows each robot to track the reference trajectory while regulating the internal wrenches. The resulting architecture is shown in Figure 4.5 and details are in the following.

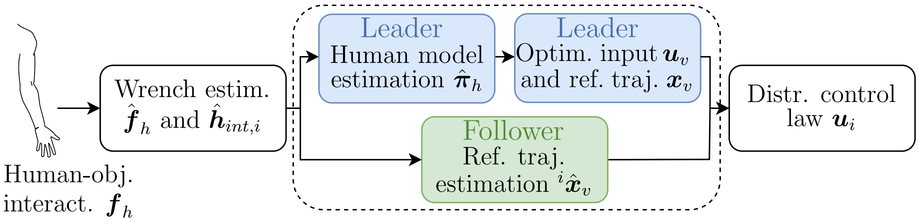

Human-obj. interact. $\boldsymbol{f}_h$
Wrench estim. $\hat{\boldsymbol{f}}_h$ and $\hat{\boldsymbol{h}}_{int,i}$
Leader
Human model estimation $\hat{\boldsymbol{\pi}}_h$
Leader
Optim. input $\boldsymbol{u}_v$ and ref. traj. $\boldsymbol{x}_v$
Follower
Ref. traj. estimation ${}^i\hat{\boldsymbol{x}}_v$
Distr. control law $\boldsymbol{u}_i$

Figure 4.5: Distributed architecture implemented by robot i for shared control. The dashed box denotes that, depending on the role of the robot, different blocks are executed: blue blocks on the top in the case of leader, green blocks on the bottom in the case of follower.

Distributed human and internal wrench estimation

Let $\boldsymbol{h}_{int,i}$ be the i th sub-vector of $\boldsymbol{h}_{int}$ (i.e., the contribution of the i th robot to the internal wrenches $\boldsymbol{h}_{int}$). The objective is to define a local observer in order for the i th robot to estimate $\boldsymbol{h}_{int,i}$ as well as the human force $\boldsymbol{f}_h$ (or equivalently $\boldsymbol{h}_h$). In view of (4.5) and by considering the selection matrix $\boldsymbol{\Gamma}_i$ introduced in (2.24)

$$\boldsymbol{\Gamma}_i = \begin{bmatrix} \boldsymbol{O}_p \dots & \underbrace{\boldsymbol{I}_p}_{robot\ i} & \dots & \boldsymbol{O}_p \end{bmatrix} \in \mathbb{R}^{p \times Np}$$

the vector $\boldsymbol{h}_{int,i}$ can be computed as

$$\begin{aligned} \boldsymbol{h}_{int,i} &= \boldsymbol{\Gamma}_i \boldsymbol{h}_{int} = \boldsymbol{h}_i - \boldsymbol{\Gamma}_i \boldsymbol{G}^\dagger \boldsymbol{G} \boldsymbol{h} = \boldsymbol{h}_i - \boldsymbol{\Gamma}_i \boldsymbol{G}^\dagger (\boldsymbol{G}_i \boldsymbol{h}_i + \sum_{j \neq i} \boldsymbol{G}_j \boldsymbol{h}_j) \\ &= \underbrace{(\boldsymbol{I}_p - \boldsymbol{\Gamma}_i \boldsymbol{G}^\dagger \boldsymbol{G}_i) \boldsymbol{h}_i}_{known} \underbrace{- \boldsymbol{\Gamma}_i \boldsymbol{G}^\dagger \sum_{j=1, j \neq i}^{N} \boldsymbol{G}_j \boldsymbol{h}_j}_{unknown} \end{aligned} \tag{4.24}$$

which is composed of a first local known term and a second unknown term depending on the wrenches exerted by the other robots. Thus, by considering the i th robot, the object model in (4.4) can be reformulated as

$$\boldsymbol{M}_o \ddot{\boldsymbol{x}}_o = \boldsymbol{G}_i \boldsymbol{h}_i + \sum_{j=1, j \neq i}^{N} \boldsymbol{G}_j \boldsymbol{h}_j + \boldsymbol{h}_h - \boldsymbol{C}_o \dot{\boldsymbol{x}}_o - \boldsymbol{g}_o. \tag{4.25}$$

Based on the approach in [98], the following residual vector $\boldsymbol{\theta}_i(t) \in \mathbb{R}^p$ is introduced

$$\boldsymbol{\theta}_i(t) = \boldsymbol{K}_\theta \int_{t_0}^{t} (\boldsymbol{\alpha} - \boldsymbol{G}_i \boldsymbol{h}_i - \boldsymbol{\theta}_i) d\tau + \boldsymbol{K}_\theta \boldsymbol{m}(t) \tag{4.26}$$

with $\boldsymbol{K}_\theta \in \mathbb{R}^{p \times p}$ a constant diagonal positive definite matrix, $\boldsymbol{m}(t) = \boldsymbol{M}_o \dot{\boldsymbol{x}}_o$ the generalized momentum of the object and $\boldsymbol{\alpha} = \boldsymbol{g}_o - \frac{1}{2} \dot{\boldsymbol{x}}_o^T \frac{\partial \boldsymbol{M}_o}{\partial \boldsymbol{x}_o} \dot{\boldsymbol{x}}_o$. From [98], it follows

$$\dot{\boldsymbol{\theta}}_i(t) = -\boldsymbol{K}_\theta \boldsymbol{\theta}_i(t) + \boldsymbol{K}_\theta \Big(\textstyle\sum_{j=1, j \neq i}^{N} \boldsymbol{G}_j \boldsymbol{h}_j + \boldsymbol{h}_h \Big) \tag{4.27}$$

which represents a low-pass filter whose bandwidth depends on the gain matrix $\boldsymbol{K}_\theta \in \mathbb{R}^{p \times p}$ and which leads to

$$\boldsymbol{\theta}_i(t) \approx \sum_{j=1, j \neq i}^{N} \boldsymbol{G}_j \boldsymbol{h}_j(t) + \boldsymbol{h}_h(t), \quad \forall t. \tag{4.28}$$

Thus, by replacing (4.28) in (4.24), it holds

$$\boldsymbol{h}_{int,i} = \boldsymbol{\Gamma}_i (\boldsymbol{I}_{Np} - \boldsymbol{G}^\dagger \boldsymbol{G}) \boldsymbol{h} \approx (\boldsymbol{I}_p - \boldsymbol{\Gamma}_i \boldsymbol{G}^\dagger \boldsymbol{G}_i) \boldsymbol{h}_i - \boldsymbol{\Gamma}_i \boldsymbol{G}^\dagger (\boldsymbol{\theta}_i - \boldsymbol{h}_h) \tag{4.29}$$

Remark 4.1. *By virtue of* (4.28)*, the approximation error made in* (4.29) *about the computation of* $\boldsymbol{h}_{int,i}$ *is*

$$\boldsymbol{h}_{int,i} - \big((\boldsymbol{I}_p - \boldsymbol{\Gamma}_i \boldsymbol{G}^\dagger \boldsymbol{G}_i) \boldsymbol{h}_i - \boldsymbol{\Gamma}_i \boldsymbol{G}^\dagger (\boldsymbol{\theta}_i - \boldsymbol{h}_h) \big) = \boldsymbol{\Gamma}_i \boldsymbol{G}^\dagger \big(\boldsymbol{\theta}_i - \boldsymbol{h}_h - \sum_{j \neq i} \boldsymbol{G}_j \boldsymbol{h}_j \big) \tag{4.30}$$

that is negligible only when the filter input (namely, $\sum_{j \neq i} \boldsymbol{G}_j \boldsymbol{h}_j + \boldsymbol{h}_h$*) has a bandwidth much smaller than the cut-off frequency of the filter. Therefore, as made in [98] and citing works, a practical choice is to set* $\boldsymbol{K}_\theta$ *as high as possible subject to the potential digital implementation of the filter itself.*

Equation (4.29) makes evident that $\boldsymbol{h}_{int,i}$ can not be computed without the knowledge of $\boldsymbol{h}_h$ which is unknown as well. In this regard, the following auxiliary signal $\boldsymbol{\xi}_i \in \mathbb{R}^p$ is defined

$$\boldsymbol{\xi}_i = (\boldsymbol{I}_p - \boldsymbol{\Gamma}_i \boldsymbol{G}^\dagger \boldsymbol{G}_i)\boldsymbol{h}_i - \boldsymbol{\Gamma}_i \boldsymbol{G}^\dagger \boldsymbol{\theta}_i \tag{4.31}$$

which, in light of (4.29) and by computing the term $\boldsymbol{\Gamma}_i \boldsymbol{G}^\dagger$, can be rewritten as

$$\boldsymbol{\xi}_i \approx \boldsymbol{h}_{int,i} - \boldsymbol{\Gamma}_i \boldsymbol{G}^\dagger \boldsymbol{h}_h = \boldsymbol{h}_{int,i} - \frac{1}{N}\boldsymbol{h}_h. \tag{4.32}$$

Note that neither $\boldsymbol{h}_{int,i}$ nor $\boldsymbol{h}_h$ are known in (4.32), but an approximation of $\boldsymbol{h}_{int,i} - \frac{1}{N}\boldsymbol{h}_h$ is provided by the right-hand side of (4.31), i.e., by the auxiliary variable $\boldsymbol{\xi}_i$. By resorting to the following lemma, the ith robot can compute the estimate ${}^i\hat{\boldsymbol{h}}_h \in \mathbb{R}^p$ of the human wrench $\boldsymbol{h}_h$ as well as the estimate $\hat{\boldsymbol{h}}_{int,i}$ of its contribution to the internal wrench.

Lemma 4.2. *Let each robot run the following observer*

$$\begin{cases} \dot{\boldsymbol{z}}_i = \gamma \sum\limits_{j \in \mathcal{N}_i} \mathrm{sign}({}^j\hat{\boldsymbol{h}}_h - {}^i\hat{\boldsymbol{h}}_h) \\ {}^i\hat{\boldsymbol{h}}_h = \boldsymbol{z}_i - N\boldsymbol{\xi}_i \\ \hat{\boldsymbol{h}}_{int,i} = \boldsymbol{\xi}_i + \dfrac{1}{N}{}^i\hat{\boldsymbol{h}}_h \end{cases} \tag{4.33}$$

where $\gamma \in \mathbb{R}$ is a positive constant, $\boldsymbol{z}_i \in \mathbb{R}^p$ is an auxiliary state, and sign$(\cdot)$ *is the component-wise signum function. Then,*

- *${}^i\hat{\boldsymbol{h}}_h$ approaches $\boldsymbol{h}_h$ in finite time T_h, $\forall i = 1, ..., N$. Equivalently, by defining the estimation error ${}^i\tilde{\boldsymbol{h}}_h = \boldsymbol{h}_h - {}^i\hat{\boldsymbol{h}}_h$, it approaches the origin in finite time T_h.*

- *$\hat{\boldsymbol{h}}_{int,i}$ approaches $\boldsymbol{h}_{int,i}$ in finite time T_h, $\forall i = 1, ..., N$. Equivalently, by defining the estimation error $\tilde{\boldsymbol{h}}_{int,i} = \boldsymbol{h}_{int,i} - \hat{\boldsymbol{h}}_{int,i}$, it approaches the origin approaches the origin in finite time.*

Proof. By leveraging the same reasoning as in [79], it can be proved that the estimate ${}^i\hat{\boldsymbol{h}}_h$ converges to the average of

$-N\boldsymbol{\xi}_i$ in finite-time T_h. Thus, by virtue of (4.32) and by recalling that, by construction, it holds $\sum_{i=1}^{N} \boldsymbol{h}_{int,i} = \boldsymbol{0}_p$, it follows ${}^{i}\hat{\boldsymbol{h}}_h \to -\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\xi}_i \approx \boldsymbol{h}_h$ or, equivalently, ${}^{i}\tilde{\boldsymbol{h}}_h \to \boldsymbol{0}_p$. Therefore, based on (4.32) and (4.33), the ith robot can also estimate its contribution to the internal wrenches $\boldsymbol{h}_{int,i}$ as follows

$$\hat{\boldsymbol{h}}_{int,i} = \boldsymbol{\xi}_i + \frac{1}{N}{}^{i}\hat{\boldsymbol{h}}_h. \tag{4.34}$$

Since ${}^{i}\tilde{\boldsymbol{h}}_h \to 0$, from (4.32) it also holds

$$\hat{\boldsymbol{h}}_{int,i} \to \boldsymbol{\xi}_i + \frac{1}{N}\boldsymbol{h}_h = \boldsymbol{h}_{int,i}. \tag{4.35}$$

This completes the proof. ■

Finally, as consequence of the lemma above, the wrench contribution of robot i to the object motion can be estimated as

$$\boldsymbol{h}_i - \hat{\boldsymbol{h}}_{int,i} = \boldsymbol{h}_i - \boldsymbol{\xi}_i - \frac{1}{N}{}^{i}\hat{\boldsymbol{h}}_h = \boldsymbol{\Gamma}_i\boldsymbol{G}^{\dagger}\boldsymbol{G}_i\boldsymbol{h}_i + \boldsymbol{\Gamma}_i\boldsymbol{G}^{\dagger}\boldsymbol{\theta}_i - \frac{1}{N}{}^{i}\hat{\boldsymbol{h}}_h \tag{4.36}$$

which converges to the actual value as well.

Generation and estimation of the object trajectory $\boldsymbol{x}_v$

Depending on the role of the ith robot, different approaches are pursued.

Leader robot: The leader robot is in charge of estimating the human model according to (4.12) based on the estimate made as in (4.33) and defining the virtual input $\boldsymbol{u}_v$ (Section 4.3.1), from which the object reference trajectory is then derived as in (4.23).

Follower robot: In order for the followers to estimate the reference trajectory, the solution in [80] is exploited. Let $\boldsymbol{\chi}_v$ be the stacked vector of the desired trajectory, that is $\boldsymbol{\chi}_v = \left[\boldsymbol{x}_v{}^T\ \dot{\boldsymbol{x}}_v{}^T\ \ddot{\boldsymbol{x}}_v{}^T\right]^T \in \mathbb{R}^{3p}$ and ${}^{i}\hat{\boldsymbol{\chi}}_v$ the respective estimate made by the follower robot i. The following distributed observer is adopted by the followers

$$\begin{cases} {}^{i}\dot{\hat{\boldsymbol{\chi}}}_v = \boldsymbol{A}_{\chi}{}^{i}\hat{\boldsymbol{\chi}}_v - \mu_{v,1}\boldsymbol{B}_{\chi}\boldsymbol{B}_{\chi}{}^{T}\boldsymbol{P}_{\chi}{}^{-1}\,{}^{i}\hat{\boldsymbol{\nu}}_v - \mu_{v,2}\mathrm{sign}\big(\boldsymbol{P}_{\chi}^{-1}\,{}^{i}\hat{\boldsymbol{\nu}}_v\big) \\ {}^{i}\hat{\boldsymbol{\nu}}_v = \sum_{j\in\mathcal{N}_i,\, j\neq l}\left({}^{i}\hat{\boldsymbol{\chi}}_v - {}^{j}\hat{\boldsymbol{\chi}}_v\right) + b_i\left({}^{i}\hat{\boldsymbol{\chi}}_v - \boldsymbol{\chi}_v\right) \end{cases} \tag{4.37}$$

where l is the index associated with the leader robot, i belongs to the set of the follower robots, b_i is 1 if the leader belongs to $\mathcal{N}_i$ and 0 otherwise, $\mu_{v,1}, \mu_{v,2} \in \mathbb{R}$ are positive gains, $\boldsymbol{A}_{\chi} \in \mathbb{R}^{3p\times 3p}$ and $\boldsymbol{B}_{\chi} \in \mathbb{R}^{3p}$ are matrices selected as

$$\boldsymbol{A}_{\chi} = \begin{bmatrix} \boldsymbol{O}_p & \boldsymbol{I}_p & \boldsymbol{O}_p \\ \boldsymbol{O}_p & \boldsymbol{O}_p & \boldsymbol{I}_p \\ \boldsymbol{O}_p & \boldsymbol{O}_p & \boldsymbol{O}_p \end{bmatrix}, \quad \boldsymbol{B}_{\chi} = \begin{bmatrix} \boldsymbol{O}_p & \boldsymbol{O}_p & \boldsymbol{I}_p \end{bmatrix}^T$$

and $\boldsymbol{P}_{\chi} \in \mathbb{R}^{3p\times 3p}$ is a positive definite matrix. By following the same reasoning as in [80], it can be proved that, under a proper selection of gains, the observer in (4.37) guarantees the finite-time leader tracking, i.e., it holds in finite-time T_{χ}

$$\|{}^{i}\hat{\boldsymbol{\chi}}_v(t) - \boldsymbol{\chi}_v(t)\| = 0\,\forall i \neq l,\ t \geq T_{\chi}. \tag{4.38}$$

Distributed low-level control

The control input $\boldsymbol{u}_i$ in (2.10) to cooperatively track the object reference trajectory and control the internal stresses, in the hypothesis of uncertain dynamics (2.12), is here devised. Let $\boldsymbol{e}_{int,i}$ denote the internal wrench error vector of robot i, i.e., $\boldsymbol{e}_{int,i} = \boldsymbol{h}^{d}_{int,i} - \boldsymbol{h}_{int,i} \in \mathbb{R}^p$, being $\boldsymbol{h}^{d}_{int,i} = \boldsymbol{\Gamma}_i\boldsymbol{h}^{d}_{int}$, and let $\Delta\boldsymbol{u}_{f,i} \in \mathbb{R}^p$ denote the following integral error

$$\Delta\boldsymbol{u}_{f,i}(t) = k_f\int_{t_0}^{t}\boldsymbol{e}_{int,i}\,d\tau \tag{4.39}$$

with $k_f \in \mathbb{R}$ a positive scalar regulating the internal wrench control, and the error vector $\boldsymbol{e}_{x,i} \in \mathbb{R}^p$

$$\boldsymbol{e}_{x,i}(t) = ({}^i\hat{\boldsymbol{x}}_v - \boldsymbol{x}_i) + k_c \int_{t_0}^{t} \sum\nolimits_{j \in \mathcal{N}_i} (\boldsymbol{x}_j - \boldsymbol{x}_i) d\tau \tag{4.40}$$

with $k_c \in \mathbb{R}$ a positive constant, which is composed of a first term for the tracking of the object trajectory and a second synchronization term aimed at limiting the internal forces during the transient phases or due to unmodeled dynamics. Finally, the following auxiliary variables are introduced

$$\begin{cases} \boldsymbol{\zeta}_i = {}^i\dot{\hat{\boldsymbol{x}}}_v + k_c \sum\nolimits_{j \in \mathcal{N}_i} (\boldsymbol{x}_j - \boldsymbol{x}_i) + k_p \boldsymbol{e}_{x,i} \quad \in \mathbb{R}^p \\ \tilde{\boldsymbol{\zeta}}_i = \boldsymbol{\zeta}_i - \dot{\boldsymbol{x}}_i = \dot{\boldsymbol{e}}_{x,i} + k_p \boldsymbol{e}_{x,i} \quad \in \mathbb{R}^p \\ \boldsymbol{\rho}_i = \boldsymbol{\zeta}_i + \Delta \boldsymbol{u}_{f,i} \quad \in \mathbb{R}^p \\ \boldsymbol{s}_i = \tilde{\boldsymbol{\zeta}}_i + \Delta \boldsymbol{u}_{f,i} \quad \in \mathbb{R}^p \end{cases} \tag{4.41}$$

with $k_p \in \mathbb{R}$ a positive constant and $\hat{\Delta} \boldsymbol{u}_{f,i}$ the estimate of $\Delta \boldsymbol{u}_{f,i}(t)$ made by robot i as

$$\hat{\Delta} \boldsymbol{u}_{f,i}(t) = k_f \int_{t_0}^{t} \hat{\boldsymbol{e}}_{int,i} \, d\tau \tag{4.42}$$

where $\hat{\boldsymbol{e}}_{int,i} = \boldsymbol{h}^d_{int,i} - \hat{\boldsymbol{h}}_{int,i}$ which takes into account that the internal wrench $\boldsymbol{h}_{int,i}$ is not known but only locally estimated as in (4.35). By extending the approach in [78], the following control law is proposed

$$\begin{aligned} \boldsymbol{u}_i &= \hat{\bar{\boldsymbol{M}}}_i \dot{\boldsymbol{\rho}}_i + \hat{\bar{\boldsymbol{C}}}_i \boldsymbol{\rho}_i + \hat{\bar{\boldsymbol{\eta}}}_i + \boldsymbol{K}_s \boldsymbol{s}_i + \Delta \boldsymbol{u}_i \\ &= \bar{\boldsymbol{Y}}_i(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i, \boldsymbol{\rho}_i, \dot{\boldsymbol{\rho}}_i) \hat{\boldsymbol{\pi}}_i + \boldsymbol{K}_s \boldsymbol{s}_i + \Delta \boldsymbol{u}_i \end{aligned} \tag{4.43}$$

being $\boldsymbol{K}_s \in \mathbb{R}^{p\times p}$ a positive definite matrix and $\Delta\boldsymbol{u}_i \in \mathbb{R}^p$ selected as

$$\begin{aligned}\Delta\boldsymbol{u}_i =& \kappa_i(t)\boldsymbol{s}_i + \boldsymbol{h}^d_{int,i} + \hat{\Delta}\boldsymbol{u}_{f,i} + \boldsymbol{\Gamma}_i\boldsymbol{G}^\dagger\boldsymbol{G}_i\boldsymbol{h}_i + \boldsymbol{\Gamma}_i\boldsymbol{G}^\dagger\boldsymbol{\theta}_i - \frac{1}{N}{}^i\hat{\boldsymbol{h}}_h \\ =& \underbrace{\kappa_i(t)\boldsymbol{s}_i}_{a} + \underbrace{\boldsymbol{h}^d_{int,i} + \hat{\Delta}\boldsymbol{u}_{f,i}}_{b} + \underbrace{\boldsymbol{\Gamma}_i\boldsymbol{G}^\dagger(\boldsymbol{G}_i\boldsymbol{h}_i + \boldsymbol{\theta}_i) - \frac{1}{N}\boldsymbol{h}_h + \frac{1}{N}{}^i\tilde{\boldsymbol{h}}_h}_{c}\end{aligned} \tag{4.44}$$

with $\kappa_i(t) \in \mathbb{R}$ a positive time-varying gain, and where

- *a*) is a robust term to guarantee convergence of internal forces in the transient phase and despite model uncertainties;
- *b*) is a force feed-forward and integral error contribution;
- *c*) represents, based on (4.36), the compensation of the contribution to the external generalized forces made by robot i on the object.

Moreover, the dynamic parameters of robot i are updated as

$$\dot{\hat{\boldsymbol{\pi}}}_i = \boldsymbol{K}_\pi^{-1}\bar{\boldsymbol{Y}}_i^T(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i, \boldsymbol{\rho}_i, \dot{\boldsymbol{\rho}}_i)^T\boldsymbol{s}_i \tag{4.45}$$

with $\boldsymbol{K}_\pi \in \mathbb{R}^{n_{\pi_i}\times n_{\pi_i}}$ a positive definite matrix.
The analysis of control law $\boldsymbol{u}_i$ is provided in the theorem.

Theorem 4.3. *Consider N robots with dynamics in* (2.10) *for which an estimate is known as in* (2.12)*. Consider the observer in* (4.37)*, the control law in* (4.43) *and the parameters update law in* (4.45)*. Then, it asymptotically holds* $\boldsymbol{x}_i \rightarrow \boldsymbol{x}_v$ *and* $\boldsymbol{e}_{int,i} \rightarrow \boldsymbol{0}_p$ $\forall i = 1, ..., N$.

Proof. The proof is provided in Appendix C.1. ■

4.3.3 Simulation validation

In this section, results are provided to validate the proposed approach. As in Section 3.4.1, Matlab environment is adopted to

Table 4.1: Simulation parameters for physical shared control.

Parameter	Value	Parameter	Value
$\boldsymbol{M}_v$	$\mathrm{diag}\{2\boldsymbol{I}_3\mathrm{kg}, \boldsymbol{I}_3\mathrm{kgm}^2\}$	$\bar{e}$	0.5
$\boldsymbol{D}_v$	$\mathrm{diag}\{10\boldsymbol{I}_3\mathrm{Ns/m}, 8\boldsymbol{I}_3\mathrm{Nms}\}$	T	0.001s
$\boldsymbol{K}_\gamma$	$0.3\boldsymbol{I}_6$	Δ	0.2s
t_h	1	$\boldsymbol{K}_\theta$	$800\boldsymbol{I}_6$
$\boldsymbol{R}_h$	$0.5\boldsymbol{I}_3$	γ	300
$\boldsymbol{R}_v$	$0.1\boldsymbol{I}_6$	$\mu_{v,1}$	30
$\boldsymbol{Q}_{r,d}$	$10\,\mathrm{diag}\{250\boldsymbol{I}_3, \boldsymbol{I}_3, 250\boldsymbol{I}_3, \boldsymbol{I}_3\}$	k_f	0.5
$\boldsymbol{W}_{h,d}$	$\boldsymbol{I}_6$	$\mu_{v,2}$	25
λ	0.998	k_c	1
$\boldsymbol{P}_0$	$10000\boldsymbol{I}_9$	k_p	1
α	0.1	$\boldsymbol{K}_s$	1

corroborate the theoretical findings in which both modelling and control are implemented. Moreover, CoppeliaSim simulator is adopted to visualize the robots and human motion as in Figure 4.6; the respective environment is provided at the link[1]. In detail, the setup shown in the figure is composed of $N = 3$ Comau Smart SiX manipulators (6 DOFs) mounted on mobile bases (2 DOFs) and they rigidly grasp a cylindrical object. Robots are redundant and it holds $p = 6$ for the robots' operational space dimension and $m = 3$ for the human model in eq. (4.1). moreover, the reduced regressor matrix $\boldsymbol{Y}_i \in \mathbb{R}^{6\times 56}$ ($n_{\pi_i} = 56$) in (2.10) is considered and a uniformly distributed 5% uncertainty is assumed concerning the robots' dynamic parameters $\boldsymbol{\pi}_i$, $i = 1, 2, 3$. For the sake of the implementation on a real setup, an initial estimation of the robot dynamic model might be obtained by resorting to one of the approaches mentioned in [126]. Furthermore, it is also required the robots agree on a common reference frame in which the object trajectory and the estimation of global variables are specified and/or computed. However, this requirement is common to almost all distributed algorithms and, in the case of mobile manipulators,

[1] `http://webuser.unicas.it/lai/robotica/papers/JINTVREPscene.ttt`

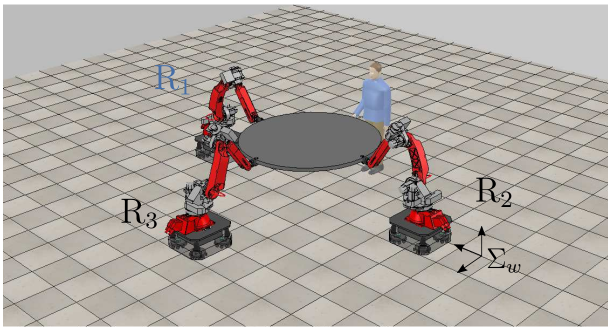


Figure 4.6: Simulation setup composed of 3 cooperative robots (R$_i$, $i = 1, 2, 3$) rigidly grasping an object with which a human operator physically interacts; the leader robot (R_1) is highlighted with blue color and the world reference frame Σ_w is reported.

implies the robots to be able to localize in a common frame; this can be obtained, for example, by using passive markers to be detected by on-board vision sensors. The latter can also be used to estimate the point in which the object is grasped by the human operator if such an information is not known beforehand.

Concerning the communication graph, bi-directional communication links are considered between robots 1 and 2 and between robots 2 and 3, while robot 1, without loss of generality, is set as leader. Finally, the manipulated object is represented by a cylinder with mass $m_o = 5$ kg, radius $r_o = 1$ m and height $h_o = 0.05$ m. In order to simulate the overall system dynamics, the Udwadia-Kalaba equation of motion for constrained systems is adopted which, as done in [127], allows to compute the constraint wrenches $\boldsymbol{h}$ arising from the robots rigid grasping constraint. In addition, uncertainty on both the manipulated object model and initial human parameters is introduced. In particular, for each robot, a uniformly distributed 3% of error is generated regarding the object dynamic model, while, concerning the initial human parameters, an uncertainty with normal distribution $\mathcal{N}(0, 0.3\pi_{h,i})\,\forall i \in \{1, ..., 9\}$, with $\pi_{h,i}$ the ith element of vector $\boldsymbol{\pi}_h$ in (4.2), is introduced.

The objective of the simulation is to achieve the shared control of the object motion where, as shown in Figure 4.7, the robots task is to perform a circular trajectory in xy-plane with radius $r_c = 1.5$ m, center $\boldsymbol{c} = [0 \quad 0 \quad 0.94]^T$m and period $T_c = 3.8$ s, while keeping the orientation constant, whereas the human desired configuration coincides with the circumference center, i.e., $\boldsymbol{p}_{h,d} \equiv \boldsymbol{c}$. Note that, as assumed for example in [128], a constant human desired motion is typically considered for human reaching motions. The human operator force is modeled according to eq. (4.1) where stiffness $\boldsymbol{K}_h$ and damping $\boldsymbol{D}_h$ matrices are increased during the simulation time (as discussed below and shown in Figure 4.8 with solid lines) in order to dynamically modify the object trajectory and make the human leading the motion. Finally, zero internal wrenches are required, i.e., the stacked vector of the desired internal wrenches $\boldsymbol{h}_{int}^d = [\boldsymbol{h}_{int,1}^{d\ T} \ \boldsymbol{h}_{int,2}^{d\ T} \ \boldsymbol{h}_{int,3}^{d\ T}]^T \in \mathbb{R}^{18}$ is $\boldsymbol{h}_{int}^d = \boldsymbol{0}_{18}$.

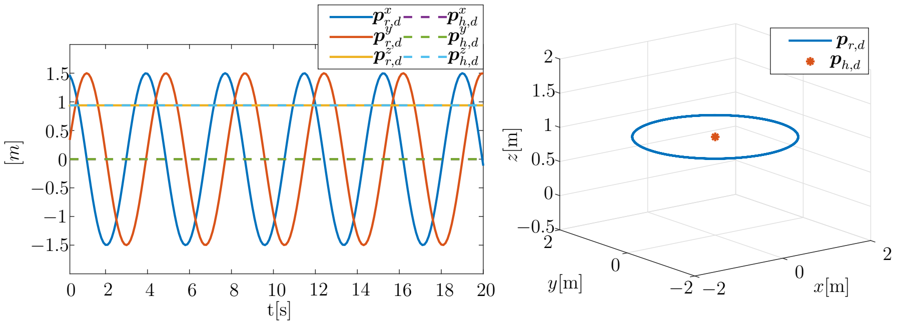


Figure 4.7: Representation in 2D (left) and 3D (right) of the robots' desired position (solid lines on the left and blue line on the right) and the human desired one (dashed lines on the left and red star on the right).

A summary video of the simulation results is available at the link[2], while simulation parameters are reported in Table 4.1. Among them and concerning the human parameters estimation discussed in Section 4.3.1, the initial covariance matrix $\boldsymbol{P}_0$ is set to a high value to denote a high uncertainty on the initial human parameters, while the values of T and Δ are selected as in [112]. In addition, the values of the weighting matrix $\boldsymbol{Q}_{r,d}$ are defined

[2] `https://youtu.be/eQwBT74F1Po`

so as to make the contribution associated with the robots desired trajectory comparable with respect to the one associated to the human operator. Finally, the rationale behind the gain matrix $\boldsymbol{K}_\theta$ is, as also stressed in Remark 4.1, to set it to a higher value than the low-level control law gains in order to estimate the respective quantity sufficiently fast. The same reasoning applies for the choice of the gain γ.

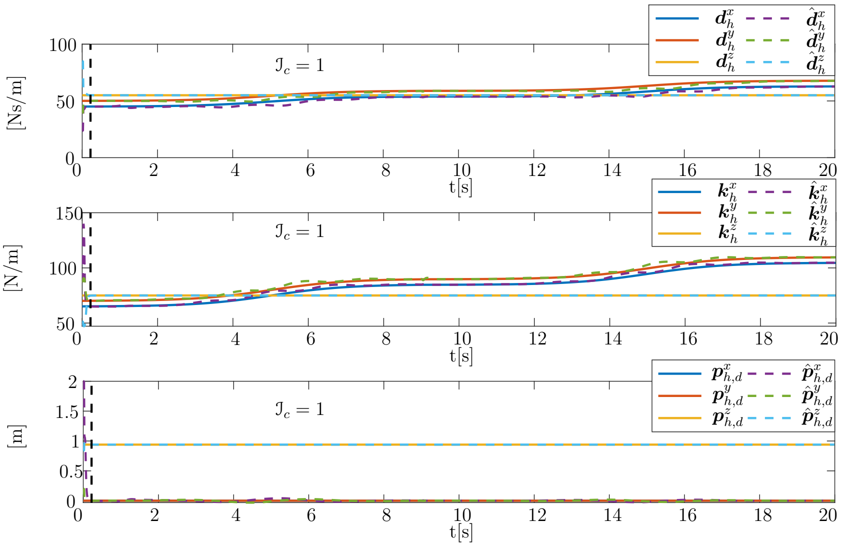


Figure 4.8: Evolution of the human parameters $\boldsymbol{d}_h$, $\boldsymbol{k}_h$ and $\boldsymbol{p}_{h,d}$ (from the top, solid lines), respectively, along x, y and z directions and the respective estimates (dashed lines); the time instant $t_c \approx 0.22$ s in which the confidence index $\mathcal{I}_c$ in (4.15) becomes 1 is highlighted by the black vertical line.

Simulation results are reported in Figures 4.8-4.11. More specifically, Figure 4.8 shows the evolution of the human parameters (solid lines) along x, y and z directions as well as the respective estimates (dashed lines) computed by the leader robot. In particular, the human stiffness $\boldsymbol{K}_h$ (middle plot) increases along x and y during the time intervals $[2, 8]$ s and $[12, 18]$ s in order to increase the leading action of the human. Similarly, an increase of the human damping (top plot) is simulated during the same intervals as a stabilizing and slowing down action. In addition, the time instant $t_c \approx 0.22$s in which the estimate is assessed reliable according to the confidence index in (4.15) is highlighted (vertical black line).

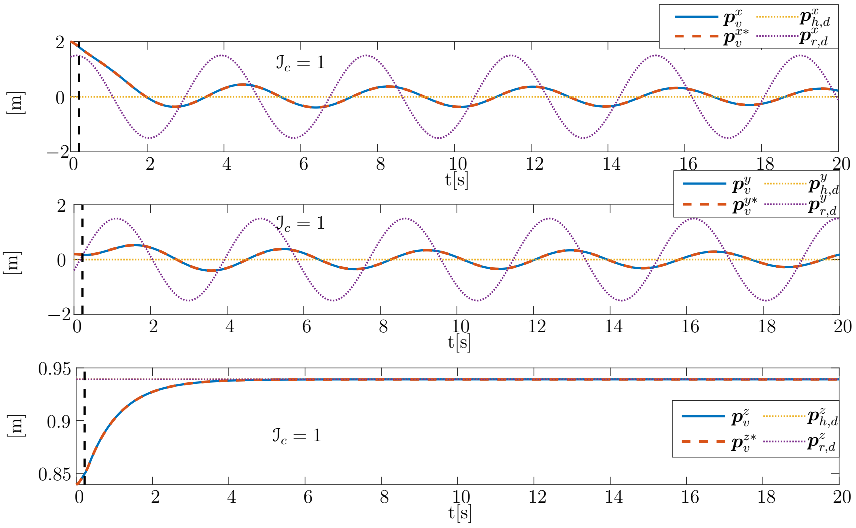


Figure 4.9: Evolution of the position of the virtual model (blue solid lines) along x, y and z directions compared with the ideal one obtained with the real human parameters $\boldsymbol{\pi}_h$ (red dashed lines); for the sake of completeness, human (yellow dotted lines) and robot (purple dotted lines) desired positions and time instant when it holds $\mathcal{I}_c = 1$ (black line) are also shown.

Figure 4.9 reports the position components $\boldsymbol{p}_v$ of the desired object trajectory (solid bold lines) resulting from the devised strategy (orientation variables are constant). In particular, as long as the estimate is assessed as not reliable ($t < t_c$), the desired object motion is generated according to the admittance model in (4.16) and is driven towards the desired position of the human; then, starting from time t_c, the optimal policy in (4.9) for the shared control is adopted which leads to perform a circular trajectory with time-varying radius ($< r_c$) and time-varying period ($< T_c$). In detail, the trajectory dynamically adapts according to the human behavior: the higher the human stiffness, the more the motion amplitudes reduce, being the human desired position coincident with the center of the trajectory, i.e., $\boldsymbol{p}_{h,d} = [0 \ 0 \ 0.94]^T$ m, as also reported in the figure (yellow dotted lines); moreover, the higher the human damping, the slower the resulting motion is. For the sake of comparison, the robots desired position $\boldsymbol{p}_{r,d}$ is also shown in the figure (purple dotted lines). In addition, in order to show the effect of human arm parameter estimation on the overall behavior of the system, the object trajectory obtained in the ideal case of zero human parameter estimation error, i.e., with $\boldsymbol{\pi}_h = \hat{\boldsymbol{\pi}}_h$, is reported (dashed line, denoted with $\boldsymbol{p}_v^*$) and is shown to be equivalent to the one derived with the estimated parameters in Figure 4.8.

Figure 4.10 (top plot) shows the error $\|{}^l\hat{\boldsymbol{h}}_h - \boldsymbol{h}_h\|$ between the real human wrench $\boldsymbol{h}_h$ and the one estimated according to Lemma 4.2. The figure shows that the estimated wrench converges to the real one and in finite time according to the theory. Therefore, each agent is adopting a consistent estimate for computing the local control strategy. In the same way, the bottom plot shows that the internal wrench $\boldsymbol{h}_{int}$ converges to the desired value $\boldsymbol{h}_{int}^d$ since the error $\|\boldsymbol{h}_{int} - \boldsymbol{h}_{int}^d\|$ converges to the origin and required in Problem 1. A zoom of the evolution of both quantities up to 0.05 s is provided in the plots.

The purpose of Figure 4.11 is to show the performance of the two-layer architecture presented in Figure 4.3. As explained in previous sections, the high-level is in charge of defining the object reference trajectory $\boldsymbol{x}_v$ which is the solution of the optimal problem defined in (4.7). This trajectory represents the desired trajectory of the object $\boldsymbol{x}_o$ to be tracked by the robots. Therefore, Figure 4.3 shows that the position $\|\boldsymbol{x}_v - \boldsymbol{x}_o\|$ (top) and velocity $\|\dot{\boldsymbol{x}}_v - \dot{\boldsymbol{x}}_o\|$ (bottom) tracking errors converge to the origin, which means that the real object trajectory converges to the desired one.

Finally, the effects of some main parameters of the framework have been analyzed. More specifically, the influence of the weight matrices $\boldsymbol{Q}_{r,d}$ and $\boldsymbol{R}_v$ in the cost function (4.7) as well as the inertia matrix $\boldsymbol{M}_v$ of the virtual model (4.6) has been considered.

No plot of the internal wrench is provided in the following analysis since it is not affected by the considered matrices and for the sake of space.

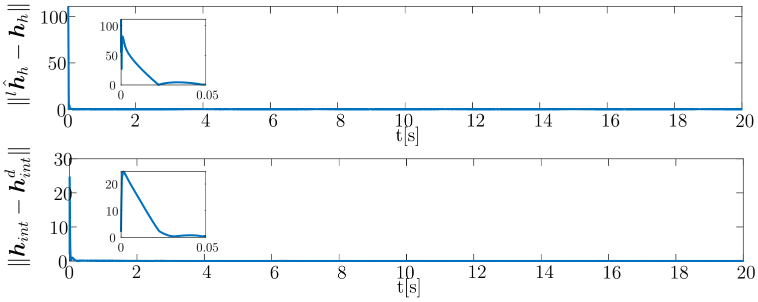


Figure 4.10: Evolution of the human forces estimation error (top) and of the internal forces error (bottom).

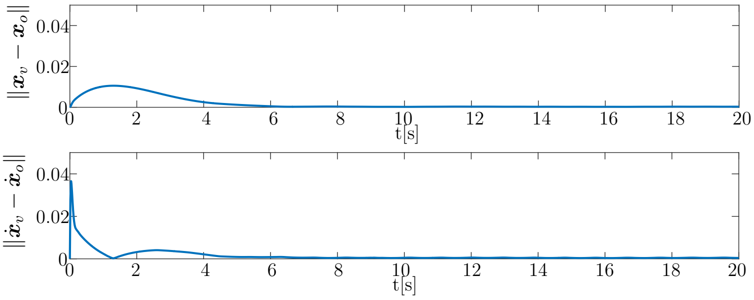


Figure 4.11: Evolution of trajectory tracking error $\|\boldsymbol{x}_v - \boldsymbol{x}_o\|$ (top) and its derivative $\|\dot{\boldsymbol{x}}_v - \dot{\boldsymbol{x}}_o\|$ (bottom).

Concerning the weight matrices $\boldsymbol{Q}_{r,d}$, it regulates how much relevance is given to tracking the robots trajectory and its effect has been investigated in regard to the resulting shared motion and human force. In particular, by expressing $\boldsymbol{Q}_{r,d}$ as $\boldsymbol{Q}_{r,d} = \text{diag}\{Q_{r,p}\boldsymbol{I}_3, Q_{r,o}\boldsymbol{I}_3, Q_{r,p}\boldsymbol{I}_3, Q_{r,o}\boldsymbol{I}_3\}$, Figures 4.12 and 4.13 report the object motion and human force, respectively, obtained by varying $Q_{r,p}$ in the set $\{10, 500, 2500, 10000\}$. For the sake of comparison, the human desired motion (dotted orange line) and the robots desired one (dotted gray line) are also shown in Figure 4.12. The figures make evident that, as the weight $Q_{r,p}$ decreases, the resulting object trajectory approaches the human desired configuration $\boldsymbol{p}_{h,d}$, and, accordingly, the human effort decreases since the person lead the control. On the contrary, the higher the value, the more the resulting trajectory gets closer to the robots desired circular motion, thus leading to a major human effort. It follows that the role of human and robots in the shared control is driven both by the tuning of the static weighting matrices in the cost function, such as $\boldsymbol{Q}_{r,d}$, and, dynamically, by the time-varying parameters of the human arm. No analysis of the weight matrices $\boldsymbol{Q}_{h,d}$ and $\boldsymbol{R}_h$ has been reported since their effect is dual to the one of $\boldsymbol{Q}_{r,d}$, i.e., the higher norm the more the human desired trajectory is pursued with respect to the robot one and the human force minimized.

As far as the weight $\boldsymbol{R}_v$ is concerned, it regulates how much relevance is given to the minimization of the robots effort $\boldsymbol{u}_i$. Figures 4.14 and 4.15 report the norm of the overall robot efforts and the resulting object motion, respectively, when varying R_v. More specifically, by denoting with $\boldsymbol{u} = [\boldsymbol{u}_1^T \ \boldsymbol{u}_2^T \ \cdots \boldsymbol{u}_N^T]^T \in \mathbb{R}^{Np}$ the stacked vector of the robots inputs, Figure 4.14 shows how by increasing the weight R_v, the robots control effort, i.e., $\|\boldsymbol{u}\|$, reduces. This is achieved, as reported in Figures 4.15, by slowing down the trajectory and by decreasing the amplitude of the oscillations. It thus follows that the higher R_v the more the control effort is reduced at the expense of lower tracking performance of the desired robot trajectory. As for the previous parameters, the human desired motion (dotted orange line) and the robots desired one (dotted gray line) are also shown in Figure 4.15 for the sake of completeness. Note that the same trajectory is recorded for all parameters for $t < t_c$ since the weight matrices are only used in the cost function and do not affect the admittance controller in eq. (4.16) that is used when the human estimation is not reliable.

Regarding the inertia matrix, let us formulate it as $\boldsymbol{M}_v = \text{diag}\{M_{v,p}\boldsymbol{I}_3, M_{v,o}\boldsymbol{I}_3\}$. The following set of values matrix has been tested for analyzing its influence: $M_{v,p} = \{2, 5, 10, 50, 500\}$. Figures 4.16 and 4.17 report the resulting object motion and the norm of the overall robots' effort, respectively, when varying $M_{v,p}$. In particular, the latter influences the bandwidth of the virtual dynamics implying that the higher $M_{v,p}$ the slower the system's response will be. This behavior is evident from the figures where a more dampened motion (see Figure 4.16), with reduced acceleration, is obtained as $M_{v,p}$ grows. This also implies that a lower robots control effort is required when increasing $M_{v,p}$.

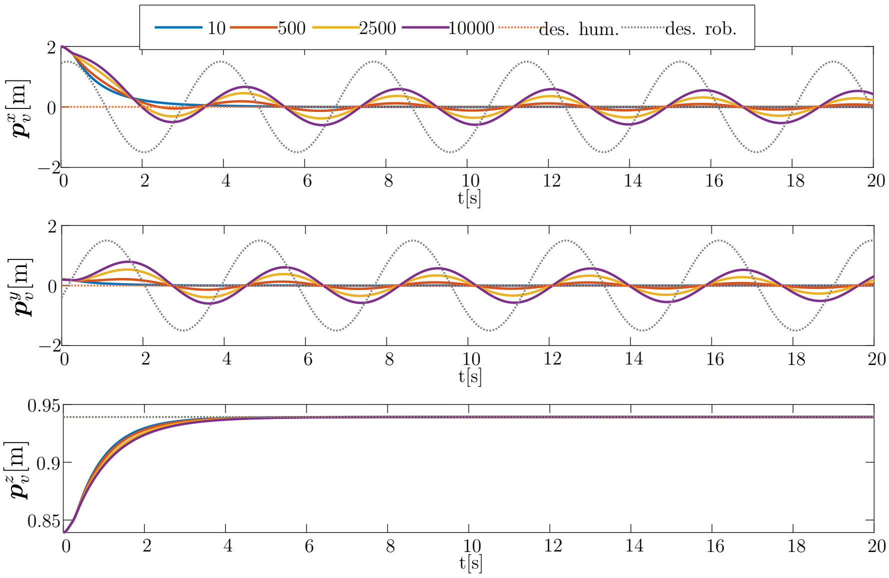


Figure 4.12: Evolution of the object motion (bold lines) along x, y and z directions by varying the robots weight $Q_{r,p} \in \{10, 500, 2500, 10000\}$ in the cost function. The human desired motion (dotted orange line) and the robots desired one (dotted gray line) are also shown.

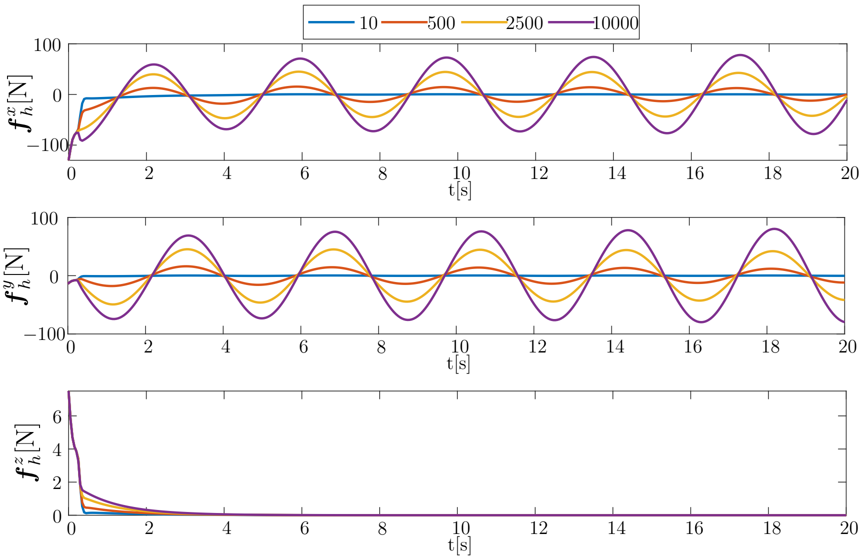


Figure 4.13: Evolution of the human force along x, y and z directions by varying the robots weight $Q_{r,p} \in \{10, 500, 2500, 10000\}$ in the cost function.

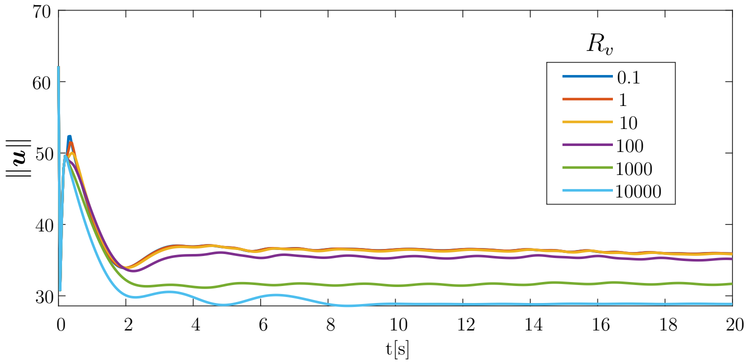


Figure 4.14: Evolution of the norm of the overall robots control input by varying the robots effort weight $R_v \in \{0.1, 1, 10, 100, 1000, 10000\}$ in the cost function.

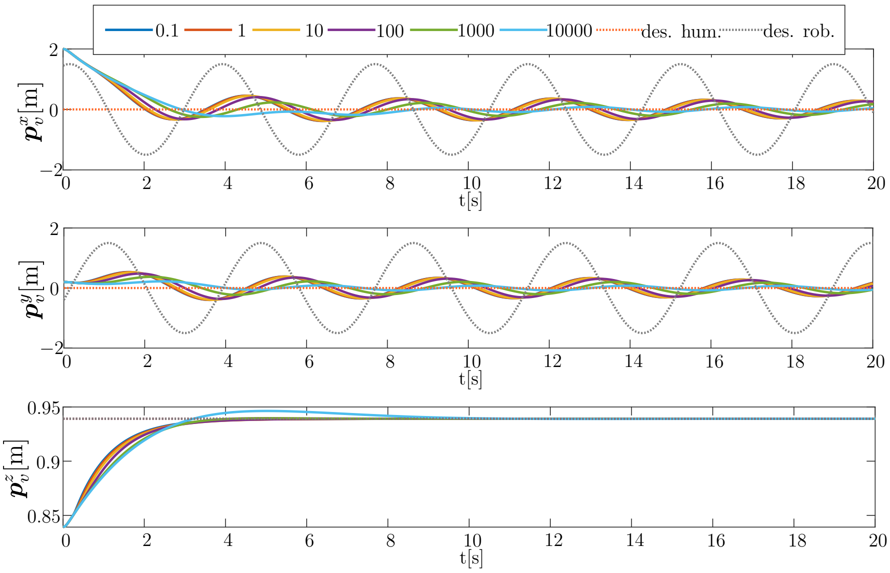


Figure 4.15: Evolution of the object motion along x, y and z directions by varying the robots effort weight $R_v \in \{0.1, 1, 10, 100, 1000, 10000\}$ in the cost function. The human desired motion (dotted orange line) and the robots desired one (dotted gray line) are also shown.

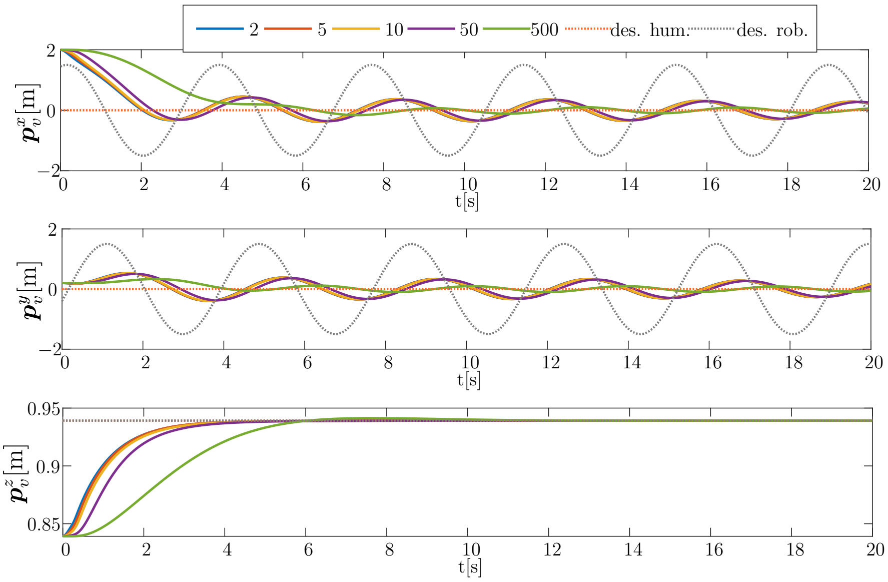


Figure 4.16: Evolution of the object motion along x, y and z directions by varying the virtual inertia $M_{v,p}$. The human desired motion (dotted orange line) and the robots desired one (dotted gray line) are also shown.

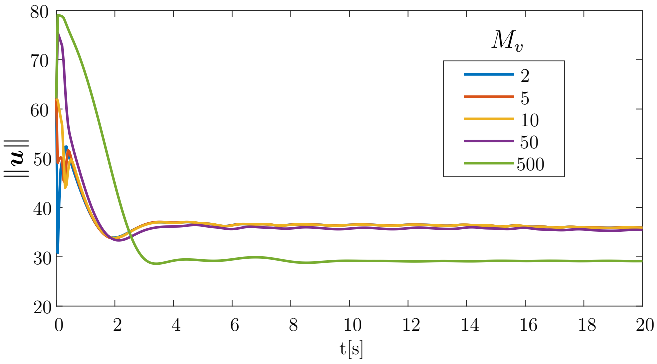


Figure 4.17: Evolution of the norm of the robots control input by varying the virtual inertia $M_{v,p}$.

4.4 Human assistance scenario

As pointed out in Chapter 1 and in the introductory part of this chapter, the human-robot physical interaction can also be aimed at fully assisting the human operator, instead of collaboratively sharing control as in the previous section. For this reason, a human multi-robot assistance scenario is considered in this section in which multiple robots co-manipulate a rigid object and its motion is guided by a human operator. The basic idea is thus that the latter provides a reference trajectory for the object while the former are responsible for producing the actual effort to transport it. In the following, based on the work presented in [129], a distributed framework is proposed to the purpose, which formulates a human wrench regulation problem with unknown and time-varying parameters.

More specifically, the main problem addressed in this section is formally stated as follows.

Problem 4.2. *Consider a system composed of N robots rigidly grasping an object which a human operator is interacting with. Assume that a central control unit is not available and a desired human wrench $\boldsymbol{h}_{h,d} \in \mathbb{R}^p$ as well as a desired internal wrench $\boldsymbol{h}_d^{int} \in \mathbb{R}^{\mu p}$ are assigned. The objective is to cooperatively enforce the following virtual dynamics to the object*

$$\boldsymbol{M}_v \ddot{\boldsymbol{x}}_v + \boldsymbol{D}_v \dot{\boldsymbol{x}}_v + \boldsymbol{K}_v(\boldsymbol{x}_v - \boldsymbol{x}_r) = \boldsymbol{e}_h \tag{4.46}$$

where $\boldsymbol{M}_v$, $\boldsymbol{D}_v$, $\boldsymbol{K}_v \in \mathbb{R}^{p\times p}$ are the virtual inertia, damping and stiffness diagonal matrices, respectively, $\boldsymbol{x}_v = \begin{bmatrix}\boldsymbol{p}_v^T & \boldsymbol{\phi}_v^T\end{bmatrix}^T \in \mathbb{R}^p$ ($\dot{\boldsymbol{x}}_v$, $\ddot{\boldsymbol{x}}_v$) is the configuration (velocity, acceleration) of the virtual object, $\boldsymbol{e}_h = \boldsymbol{h}_h - \boldsymbol{h}_{h,d} \in \mathbb{R}^p$ is the wrench regulation error and $\boldsymbol{x}_r \in \mathbb{R}^p$ is an additional trajectory which must be designed so as to achieve human wrench regulation, i.e., $\boldsymbol{e}_h \to \boldsymbol{0}_p$. Finally, regulation of internal wrenches is required, i.e., $\boldsymbol{h}^{int} \to \boldsymbol{h}_d^{int}$.

The basic idea is, thus, to cooperatively impose a compliant behavior to the object through an admittance model and, at the same time, to regulate this model in order to achieve a desired human wrench. In particular, a reasonable choice for the latter is a zero wrench, i.e., $\boldsymbol{h}_{h,d} = \mathbf{0}_p$, which allows to minimize the human effort. Therefore, two sub-issues are required to be solved: i) the definition of the trajectory $\boldsymbol{x}_r$, referred to as *regulation trajectory*, allowing the human wrench regulation and ii) the definition of a distributed framework to make the object behave according to the virtual dynamics in (4.46) while regulating the internal wrenches $\boldsymbol{h}^{int}$ to $\boldsymbol{h}_d^{int}$.

In the rest of the section, the damping term in the human model in eq. (4.1) is neglected. As stressed in [89], this is motivated by the fact that the stiffness range is typically greater than those associated with inertia and damping terms and the stiffness term is dominant with low velocities. Therefore, the following time-varying spring model is assumed for the human arm end-point

$$\boldsymbol{K}_h(t)\,(\boldsymbol{x}_{h,d}(t) - \boldsymbol{x}_h(t)) = \boldsymbol{h}_h(t) \tag{4.47}$$

with $m = p$ in (4.1). Note that both human parameters $\boldsymbol{K}_h$ and $\boldsymbol{x}_{h,d}$ are assumed to be *unknown* and can generally be time-varying with unknown derivatives. Moreover, as discussed in Section 4.2.1, $\boldsymbol{K}_h$ is a diagonal matrix which is expressed as $\boldsymbol{K}_h = \mathrm{diag}(\boldsymbol{k}_h)$. Based for example on the measures in [110], it is reasonable to assume the human parameters and their derivatives to be bounded with *unknown* bounds, thus leading to the following assumption.

Assumption 4.3. *The human stiffness and desired configuration are bounded in the sense that for each component k_h^i of the diagonal of $\boldsymbol{K}_h$ it holds $\underline{k}_h \leq k_h^i \leq \bar{k}_h$ and for each component $x_{h,d}^i$ of $\boldsymbol{x}_{h,d}$ it holds $\underline{x}_{h,d} \leq x_{h,d}^i \leq \bar{x}_{h,d}$ with $i = 1, .., p$ and $\underline{k}_h$, $\bar{k}_h$, $\underline{x}_{h,d}$, $\bar{x}_{h,d} \in \mathbb{R}$ unknown positive constants; in the same way, the first and second time derivatives of $\boldsymbol{K}_h$ and $\boldsymbol{x}_{h,d}$ are bounded with unknown bounds.*

4.4.1 Human wrench regulation approach

The modular distributed architecture realized for the shared control scenario in Section 4.3.2 is here leveraged to solve Problem 4.2, by properly modifying the components dealing with the trajectory definition (by the leader) according to the problem at hand. The resulting architecture, based on the leader-follower paradigm, is reported in Figure 4.18. More specifically, each robot runs the distributed observer in eq. (4.33) to estimate the human wrench and compute its contribution to the internal wrench as in eq. (4.34). Then, the leader robot computes the regulation trajectory $\boldsymbol{x}_r$ and, based on this, the object reference trajectory $\boldsymbol{x}_v$ according to eq. (4.46), while the followers estimate the latter. Finally, each robot runs its local control law in eq. (4.43) which allows to track the object reference trajectory as well as to regulate the internal wrench.

In the following, the proposed method to compute the regulation trajectory $\boldsymbol{x}_r$ that achieves wrench regulation is detailed. Differently from the works reported in Section 4.1, the proposed approach provides a *formal* analysis, instead of heuristics, to derive the control action for regulating the human wrench. Furthermore, as for the human multi-robot shared control strategy in Section 4.3, a multi-robot scenario, even with distributed architecture, is considered for the first time.

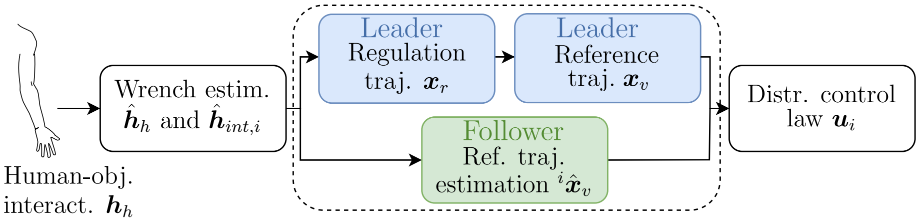

Leader
Regulation traj. $\boldsymbol{x}_r$
Leader
Reference traj. $\boldsymbol{x}_v$
Wrench estim. $\hat{\boldsymbol{h}}_h$ and $\hat{\boldsymbol{h}}_{int,i}$
Distr. control law $\boldsymbol{u}_i$
Follower
Ref. traj. estimation ${}^i\hat{\boldsymbol{x}}_v$
Human-obj. interact. $\boldsymbol{h}_h$

Figure 4.18: Distributed architecture implemented by robot i for human assistance. The dashed box denotes that, depending on the role of the robot, different blocks are executed: blue blocks on the top in the case of leader, green blocks on the bottom in the case of follower.

Regulation trajectory definition

By resorting to force tracking theory in impedance control as in [130], the aim is here, as stated above, to design the regulation trajectory $\boldsymbol{x}_r$ in (4.46) such that a desired human wrench is achieved. By virtue of the human model in (4.47), the following lemma holds.

Lemma 4.4. *The virtual dynamics in* (4.46) *can be reformulated in terms of the wrench error* $\boldsymbol{e}_h$ *as follows*

$$\begin{aligned}\ddot{\boldsymbol{e}}_h + \bar{\boldsymbol{\Omega}}_1\dot{\boldsymbol{e}}_h + \bar{\boldsymbol{\Omega}}_2\boldsymbol{e}_h = &- \boldsymbol{\Omega}_1(t)\dot{\boldsymbol{e}}_h - \boldsymbol{\Omega}_2(t)\boldsymbol{e}_h + \boldsymbol{\omega}_3(t) + \bar{\boldsymbol{\omega}}_3 \\ &- \boldsymbol{M}_v^{-1}\boldsymbol{K}_h\boldsymbol{u}_h\end{aligned} \tag{4.48}$$

where $\bar{\boldsymbol{\Omega}}_1, \bar{\boldsymbol{\Omega}}_2 \in \mathbb{R}^{p\times p}$ *and* $\bar{\boldsymbol{\omega}}_3 \in \mathbb{R}^p$ *are known constant matrices,* $\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2 \in \mathbb{R}^{p\times p}$ *and* $\boldsymbol{\omega}_3 \in \mathbb{R}^p$ *are unknown time-varying matrices and* $\boldsymbol{u}_h = \boldsymbol{K}_v\boldsymbol{x}_r \in \mathbb{R}^p$ *represents the virtual input to be designed.*

Proof. The proof is in the Appendix C.2 which also provides the expressions of $\bar{\boldsymbol{\Omega}}_1\, \bar{\boldsymbol{\Omega}}_2$, $\bar{\boldsymbol{\omega}}_3$ and $\boldsymbol{\Omega}_1\, \boldsymbol{\Omega}_2$, $\boldsymbol{\omega}_3$. ■

Let $\boldsymbol{\xi} = \begin{bmatrix}\boldsymbol{e}_h^T & \dot{\boldsymbol{e}}_h^T\end{bmatrix}^T \in \mathbb{R}^{2p}$ be the aggregate wrench regulation error vector whose dynamics, based on (4.48), is given by

$$\dot{\boldsymbol{\xi}} = \boldsymbol{H}\boldsymbol{\xi} + \boldsymbol{D}(\boldsymbol{\eta} - \boldsymbol{M}_v^{-1}\boldsymbol{K}_h\boldsymbol{u}_h) \tag{4.49}$$

with

$$\begin{aligned}&\boldsymbol{H} = \begin{bmatrix}\boldsymbol{O}_p & \boldsymbol{I}_p \\ -\bar{\boldsymbol{\Omega}}_2 & -\bar{\boldsymbol{\Omega}}_1\end{bmatrix} \in \mathbb{R}^{2p\times 2p}, \quad \boldsymbol{D} = \begin{bmatrix}\boldsymbol{O}_p \\ \boldsymbol{I}_p\end{bmatrix} \in \mathbb{R}^{2p\times p} \\ &\boldsymbol{\eta} = -\boldsymbol{\Omega}_1(t)\dot{\boldsymbol{e}}_h - \boldsymbol{\Omega}_2(t)\boldsymbol{e}_h + \boldsymbol{\omega}_3(t) + \bar{\boldsymbol{\omega}}_3 \in \mathbb{R}^p.\end{aligned} \tag{4.50}$$

In addition, let $\boldsymbol{z} \in \mathbb{R}^p$ be an auxiliary variable defined as $\boldsymbol{z} = \boldsymbol{D}^T\boldsymbol{Q}\boldsymbol{\xi}$ where $\boldsymbol{Q} \in \mathbb{R}^{2p\times 2p}$ is a a symmetric positive definite matrix such that the following equality holds

$$\boldsymbol{H}^T\boldsymbol{Q} + \boldsymbol{Q}\boldsymbol{H} = -\boldsymbol{P} \tag{4.51}$$

with $\boldsymbol{P}$ a symmetric positive definite matrix.

The following input $\boldsymbol{u}_h$, related through $\boldsymbol{K}_v$ to $\boldsymbol{x}_r$ in (4.46), is proposed to achieve human force regulation

$$\boldsymbol{u}_h(t) = \boldsymbol{M}_v\big(\alpha(t)\|\dot{\boldsymbol{e}}_h(t)\| + \beta(t)\|\boldsymbol{e}_h(t)\| + \gamma(t)\big)\,\mathrm{sign}(\boldsymbol{z}(t)) \tag{4.52}$$

where sign$(\cdot)$ is the component-wise signum function and $\alpha,\ \beta,\ \gamma \in \mathbb{R}$ are time-varying gains updated as follows

$$\dot{\alpha} = \|\dot{\boldsymbol{e}}_h\|\|\boldsymbol{z}\|, \quad \dot{\beta} \ = \|\boldsymbol{e}_h\|\|\boldsymbol{z}\|, \quad \dot{\gamma} = \|\boldsymbol{z}\|. \tag{4.53}$$

The following lemma holds true.

Lemma 4.5. *Consider the dynamics in* (4.49) *and the virtual input in* (4.52) *with time-varying gains in* (4.53) *and assume* $\boldsymbol{K}_h$ *is diagonal and conditions in Assumption 4.3 are verified. Then, the wrench regulation error* $\boldsymbol{e}_h$ *asymptotically converges to the origin and the time-varying gains converge to some finite steady state values.*

Proof. In order to prove the lemma, the following Lyapunov function candidate is analyzed

$$V = \frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{Q}\boldsymbol{\xi} + \frac{1}{2\sigma_\alpha}(\alpha-\bar{\alpha})^2 + \frac{1}{2\sigma_\beta}(\beta-\bar{\beta})^2 + \frac{1}{2\sigma_\gamma}(\gamma-\bar{\gamma})^2 \tag{4.54}$$

with $\sigma_\alpha,\ \sigma_\beta,\ \sigma_\gamma, \bar{\alpha},\ \bar{\beta}, \bar{\gamma} \in \mathbb{R}$ positive constants and $\boldsymbol{Q}$ defined according to (4.51). In view of the dynamics in (4.49), the time derivative of V is

$$\begin{aligned}\dot{V} = &-\frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{P}\boldsymbol{\xi} + \boldsymbol{z}^T(\boldsymbol{\eta} - \boldsymbol{M}_v^{-1}\boldsymbol{K}_h\boldsymbol{u}_h) + \frac{1}{\sigma_\alpha}(\alpha-\bar{\alpha})\dot{\alpha} \\ &+ \frac{1}{\sigma_\beta}(\beta-\bar{\beta})\dot{\beta} + \frac{1}{\sigma_\gamma}(\gamma-\bar{\gamma})\dot{\gamma}.\end{aligned} \tag{4.55}$$

By considering the input law in (4.52) with gains update law in (4.53) and by recalling that $\boldsymbol{K}_h$ and $\boldsymbol{M}_v$ are diagonal matrices, (4.55) can be rewritten as

$$\begin{aligned}\dot{V} = &-\frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{P}\boldsymbol{\xi} + \boldsymbol{z}^T\big(\boldsymbol{\eta} - \boldsymbol{K}_h(\alpha\|\dot{\boldsymbol{e}}_h\| + \beta\|\boldsymbol{e}_h\| + \gamma)\,\mathrm{sign}(\boldsymbol{z})\big) \\ &+ \frac{1}{\sigma_\alpha}(\alpha-\bar{\alpha})\|\dot{\boldsymbol{e}}_h\|\|\boldsymbol{z}\| + \frac{1}{\sigma_\beta}(\beta-\bar{\beta})\|\boldsymbol{e}_h\|\|\boldsymbol{z}\| + \frac{1}{\sigma_\gamma}(\gamma-\bar{\gamma})\|\boldsymbol{z}\|\end{aligned} \tag{4.56}$$

where, as in the above, $\|\cdot\|$ represents the 2-norm of $(\cdot)$. Note that, as stated in Section 4.2.1, the diagonality of the stiffness matrix is commonly assumed and validated in the literature [89, 112, 114]. By virtue of the diagonality of $\boldsymbol{K}_h$ and by denoting with z^i the ith component of $\boldsymbol{z}$, the term $\boldsymbol{z}^T\boldsymbol{K}_h\,\mathrm{sign}(\boldsymbol{z})$ can be rewritten as

$$\begin{aligned} \boldsymbol{z}^T\boldsymbol{K}_h\,\mathrm{sign}(\boldsymbol{z}) &= \sum_{i=1}^{p} k_h^i|z^i| \\ &\geq \underline{k}_h \sum_{i=1}^{p} |z^i| \\ &= \underline{k}_h\|\boldsymbol{z}\|_1 \end{aligned} \tag{4.57}$$

where the last inequality follows from the lower bound on the human stiffness in Assumption 4.3 and $\|\boldsymbol{z}\|_1$ is the 1-norm of $\boldsymbol{z}$, for which it holds $\|\boldsymbol{z}\|_1 \geq \|\boldsymbol{z}\|$. Equation (4.56) thus becomes

$$\begin{aligned} \dot{V} \leq &-\frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{P}\boldsymbol{\xi} + \|\boldsymbol{z}\|\big(\|\boldsymbol{\eta}\| - \underline{k}_h(\alpha\|\dot{\boldsymbol{e}}_h\| + \beta\|\boldsymbol{e}_h\| + \gamma)\big) \\ &+\frac{1}{\sigma_\alpha}(\alpha-\bar{\alpha})\|\dot{\boldsymbol{e}}_h\|\|\boldsymbol{z}\| + \frac{1}{\sigma_\beta}(\beta-\bar{\beta})\|\boldsymbol{e}_h\|\|\boldsymbol{z}\| + \frac{1}{\sigma_\gamma}(\gamma-\bar{\gamma})\|\boldsymbol{z}\|. \end{aligned} \tag{4.58}$$

At this point, in light of Assumption 4.3, the boundedness of the time-varying matrices $\boldsymbol{\Omega}_i$ $i = 1, 2$ and the vector $\boldsymbol{\omega}_3$ is verified, i.e., it holds $\|\boldsymbol{\Omega}_i(t)\| \leq \Omega_i\,\forall t$, $i = 1, 2$ and $\|\boldsymbol{\omega}_i(t)\| \leq \Omega_3\,\forall t$ with $\Omega_i \in \mathbb{R}, i = 1, 2, 3$ positive constants, which implies that the following holds true

$$\|\boldsymbol{\eta}\| \leq \Omega_1\|\dot{\boldsymbol{e}}_h\| + \Omega_2\|\boldsymbol{e}_h\| + \Omega_3 + \|\bar{\boldsymbol{\omega}}_3\|. \tag{4.59}$$

Equation (4.58) can be thus rewritten as

$$\begin{aligned}\dot{V} \leq & -\frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{P}\boldsymbol{\xi} + \|\boldsymbol{z}\|\big(\Omega_1\|\dot{\boldsymbol{e}}_h\| + \Omega_2\|\boldsymbol{e}_h\| + \Omega_3 + \|\bar{\boldsymbol{\omega}}_3\| \\ & - \underline{k}_h(\alpha\|\dot{\boldsymbol{e}}_h\| + \beta\|\boldsymbol{e}_h\| + \gamma)\big) + \frac{1}{\sigma_\alpha}(\alpha - \bar{\alpha})\|\dot{\boldsymbol{e}}_h\|\|\boldsymbol{z}\| \\ & + \frac{1}{\sigma_\beta}(\beta - \bar{\beta})\|\boldsymbol{e}_h\|\|\boldsymbol{z}\| + \frac{1}{\sigma_\gamma}(\gamma - \bar{\gamma})\|\boldsymbol{z}\| \\ = & -\frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{P}\boldsymbol{\xi} + \|\boldsymbol{z}\|\|\dot{\boldsymbol{e}}_h\|\Big(\Omega_1 - \frac{\bar{\alpha}}{\sigma_\alpha}\Big) + \|\boldsymbol{z}\|\|\dot{\boldsymbol{e}}_h\|\alpha\Big(\frac{1}{\sigma_\alpha} - \underline{k}_h\Big) \\ & + \|\boldsymbol{z}\|\|\boldsymbol{e}_h\|\Big(\Omega_2 - \frac{\bar{\beta}}{\sigma_\beta}\Big) + \|\boldsymbol{z}\|\|\boldsymbol{e}_h\|\beta\Big(\frac{1}{\sigma_\beta} - \underline{k}_h\Big) \\ & + \|\boldsymbol{z}\|\Big(\Omega_3 + \|\bar{\boldsymbol{\omega}}_3\| - \frac{\bar{\gamma}}{\sigma_\gamma}\Big) + \|\boldsymbol{z}\|\gamma\Big(\frac{1}{\sigma_\gamma} - \underline{k}_h\Big).\end{aligned} \tag{4.60}$$

Therefore, by selecting $\sigma_i > \underline{k}_h$ $i = \alpha, \beta, \gamma$ and $\bar{\alpha}$, $\bar{\beta}$, $\bar{\gamma}$ sufficiently high such that $\bar{\alpha} \geq \sigma_\alpha \Omega_1$, $\bar{\beta} \geq \sigma_\beta \Omega_2$ and $\bar{\gamma} \geq \sigma_\gamma(\Omega_3 + \|\bar{\boldsymbol{\omega}}_3\|)$, one obtains

$$\dot{V} \leq -\frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{P}\boldsymbol{\xi}$$

from which it follows that $\dot{V}$ is negative semi-definite and, then, that V is bounded as well as $\boldsymbol{\xi}$, α, β and γ. By applying La-Salle Yoshizawa corollary [131], it follows

$$\boldsymbol{\xi}^T\boldsymbol{P}\boldsymbol{\xi} \to 0$$

thus implying $\boldsymbol{\xi} \to \boldsymbol{0}_{2p}$. Finally, by noticing from (4.53) that the gains are monotonically increasing, their boundedness also implies the convergence to some steady state values. This completes the proof. ■

Formally, due to the discontinuous input in (4.52), the solution to (4.48) should be intended in the Filippov's sense and a non-smooth analysis should be carried out to prove the lemma. This formalism is here omitted but the same results would follow.

4.4.2 Simulation results

In this section, the proposed solution to Problem 4.2 is validated through simulation results. The considered setup is depicted in Figure 4.19. According to the multiple arms formulation in [129], three dual-arm Kinova Movo mobile robots ($N = 3$, $p = 6$) are considered that cooperatively grasp a rigid object for an assisted transportation task.

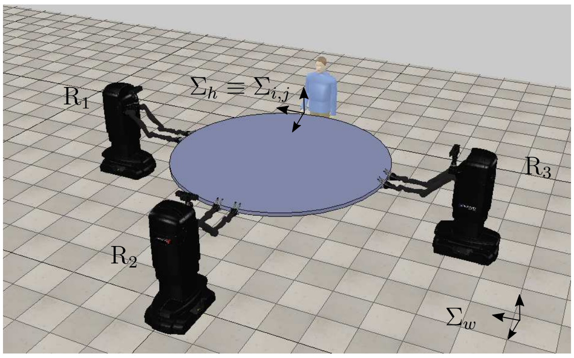


Figure 4.19: Simulation setup composed of 3 cooperative robots (R$_i$, $i = 1, 2, 3$) and a human operator; Σ_w is the world reference frame, Σ_h is the human reference frame and $\Sigma_{i,j}$ is the reference frame of the jth arm belonging to the ith robot with $i = 1, 2, 3$ and $j = 1, 2$.

With regards to the robots' model, the reduced regressor matrix $\bar{\boldsymbol{Y}}_i \in \mathbb{R}^{6\times126}$ ($n_{\pi_i} = 126$) in (2.10) has been considered and a 3% uncertainty has been assumed for the dynamic parameters $\boldsymbol{\pi}_i$, $i = 1, 2, 3$. Bi-directional communication links have been supposed between robots 1 and 2 and between robots 2 and 3. Finally, robot 1 has been selected as the leader robot. With regard to the human arm end-point model, a time-varying human desired configuration $\boldsymbol{x}_{h,d}$ has been considered, which is shown in Figure 4.20, with constant stiffness matrix $\boldsymbol{K}_h = 10\boldsymbol{I}_p$. More in detail, the positional components of $\boldsymbol{x}_{h,d}$ are such that a continuous transition from $\boldsymbol{p}_{h,d} = \begin{bmatrix}-3.1 & 0.7 & 0.8\end{bmatrix}^T$ m to $\boldsymbol{p}_{h,d} = \begin{bmatrix}-3.1 & 1.2 & 0.8\end{bmatrix}^T$ m occurs in the time interval [7.5 8.5] s, thus simulating the human intention to accomplish different reaching motions; whereas the orientation components are kept constant and equal to the initial configuration. Concerning the object model, a cylindrical object with mass $m_o = 5$ kg, radius $r_o = 2.6$ m, height $h_o = 0.05$ m, starting position $\boldsymbol{p}_o(0) = \begin{bmatrix}-2.6 & 0 & 0.8\end{bmatrix}^T$ m and with orientation as in the Figure 4.19 has been considered.

The aim of the simulation is to cooperatively transport the load while minimizing the human effort, i.e., $\boldsymbol{h}_{h,d} = \boldsymbol{0}_p$ in (4.46), and the internal stresses, i.e., $\boldsymbol{h}_d^{int} = \boldsymbol{0}_{\mu p}$. The following set of parameters has been used: $\boldsymbol{M}_v = \mathrm{diag}\{5\boldsymbol{I}_3\mathrm{kg},\ \boldsymbol{I}_3\mathrm{kgm}^2\}$, $\boldsymbol{D}_v = \mathrm{diag}\{6\boldsymbol{I}_3\mathrm{Ns/m},\ 3\boldsymbol{I}_3\mathrm{Nms}\}$ and, finally, $\boldsymbol{K}_v = \mathrm{diag}\{10\boldsymbol{I}_3\mathrm{N/m},\ 3\boldsymbol{I}_3\mathrm{Nm}\}$ in (4.46), $\alpha(0) = \beta(0) = \gamma(0) = 5$ in (4.52), $\boldsymbol{K}_\theta = 800\boldsymbol{I}_p$, $\kappa_{v,1} = 30$, $\kappa_{v,2} = 25$, $k_f = 0.5$, $k_c = 1$, $\boldsymbol{K}_s = 10\boldsymbol{I}_{12}$ and $\boldsymbol{K}_\pi = 0.1\boldsymbol{I}_{126}$ in (4.27), (4.37), (4.43) and (4.45), respectively.

Numerical results are presented in Figures 4.20-4.23 and in the attached video[3]. In particular, Figure 4.20 reports the evolution of both the human desired position and the virtual model position (4.46), which is then used as reference trajectory for the low-level control.

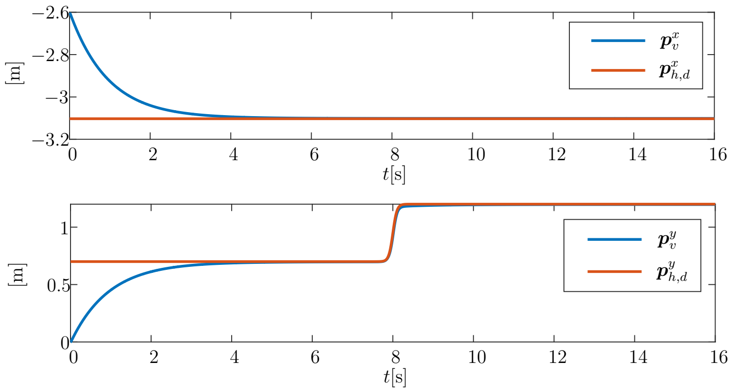


Figure 4.20: Evolution of the position of the virtual model $\boldsymbol{p}_v$ compared to the human desired position $\boldsymbol{p}_{h,d}$; the notation $\boldsymbol{p}_c^a$ denotes the component of $\boldsymbol{p}_c$ along the axis a.

In addition, the human wrench regulation error is reported in Figure 4.21 and is shown to be convergent to the origin; an increase

[3] http://webuser.unicas.it/lai/robotica/video/pHMRI_SMC2019.mp4

is recorded in correspondence of the change in human desired trajectory at $t \approx 8$ s which is then recovered.

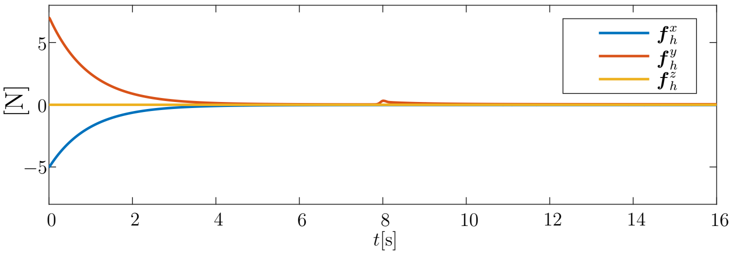


Figure 4.21: Evolution of the human force $\boldsymbol{f}_h$ where $\boldsymbol{f}_h^a$ represents the human force along the direction a.

The time-varying gains $\alpha(t)$, $\beta(t)$, $\gamma(t)$, updated as in (4.53), are shown in Figure 4.22. In detail, the figure makes evident that the gains monotonically increase with the wrench regulation error and, then, assess to steady state values. At time $t \approx 8$ s, gains adapt to the change in human desired trajectory.

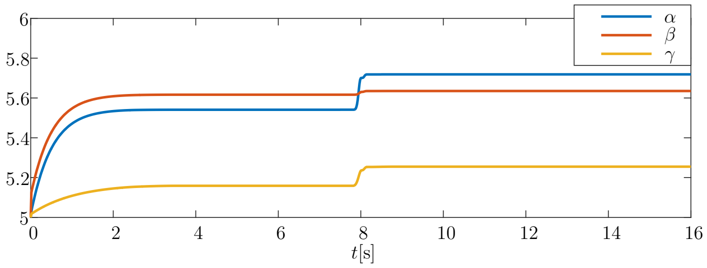


Figure 4.22: Evolution of the adaptive gains α, β, γ according to (4.53).

Finally, concerning the low-level control, Figure 4.23 shows the virtual dynamics tracking error defined as $\tilde{\boldsymbol{x}}_o = \boldsymbol{x}_v - \boldsymbol{x}_o$ and the internal forces error, that is $\tilde{\boldsymbol{h}}^{int} = \begin{bmatrix}\tilde{\boldsymbol{h}}_1^{int} & \tilde{\boldsymbol{h}}_2^{int} & \tilde{\boldsymbol{h}}_3^{int}\end{bmatrix} \in \mathbb{R}^{32}$; more specifically, they are both convergent to the origin.

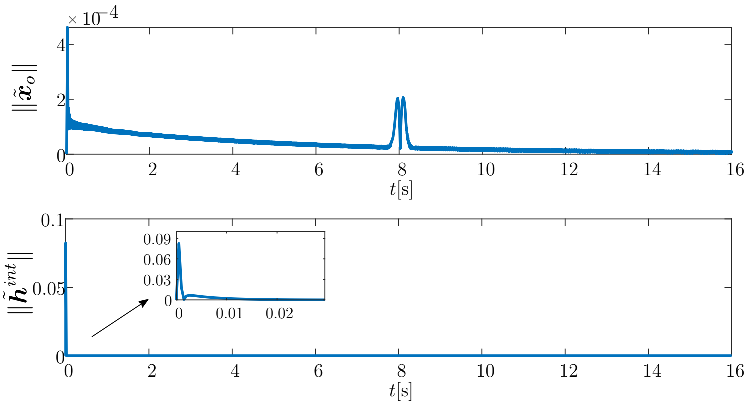


Figure 4.23: Evolution of the virtual dynamics tracking error (top) and the internal forces regulation error (bottom).

4.5 Contact classification: accidental or intentional

The strategies presented above allow to properly handle the exchange of forces between the human operator and the robots on the basis of the desired interaction, either shared control or human assistance. To this aim, the assumption is made that this exchange of forces only occurs *voluntarily*. However, in realistic human-robot physical interaction scenarios, it cannot be guaranteed that the interaction is always intentional but, as shown in Figure 4.24, the problem arises that also *accidental* contacts might occur. This implies the need to integrate strategies to detect and classify the human contact despite the robot task, i.e., contact detection and classification must be performed even when the robot task requires explicit contact with the environment. Based on this, proper reaction strategies can be then carried out: in case of intentional contact, one of the approaches previously presented in this chapter may be leveraged, while, in case of accidental contact, an avoidance strategy should be performed.

In light of the above, an approach is presented in the following which, based on the works in [132] and [133], allows to detect human contact despite the robotic task, recognize its nature and react accordingly. The respective components as well as the experimental validation are provided in the section. Note that a preliminary scenario with single manipulator is considered in this chapter. The extension to multi-robot setups will be subject of future work. In these setups, the main challenges that need to be addressed are: *i)* in the case the robots are tightly connected by co-manipulating the same object, the human wrench needs to be identified despite possible internal wrenches, *ii)* the reaction strategy needs to also take into account the robots' cooperative task, i.e., if the robots are closely cooperating and the human operator interacts only with one of them, the reaction behaviors of the individual robots cannot be independent.

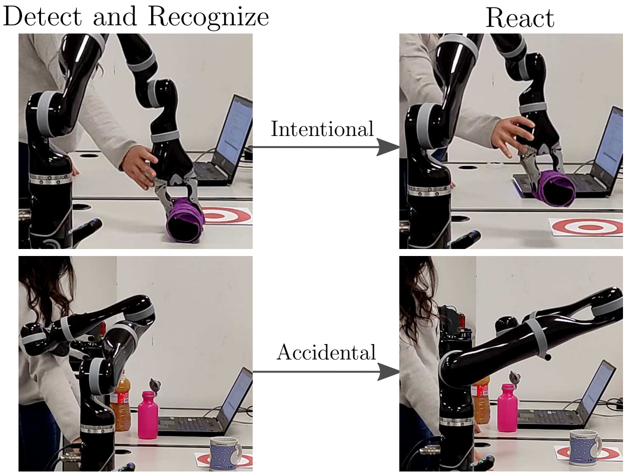


Figure 4.24: Framework example: the human contact is detected and classified (on the left) and the system reacts according to its nature (on the right), i.e., intentional (top row) or accidental (bottom row).

4.5.1 Preliminaries for contact classification

Robot model

Let $\boldsymbol{q}$ be the joint vector of the single robot involved in the system. The human operator directly exerts forces on the robot and no co-manipulated object exists (differently from Sections 4.3 and 4.4).

The model in eq. (2.6) is reformulated to explicitly express the torques induced by the interaction with the environment and the ones induced by the interaction with the human operator:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{\mathcal{T}} + \boldsymbol{\tau}_h \tag{4.61}$$

where $\boldsymbol{\tau}_{\mathcal{T}} = \boldsymbol{J}(\boldsymbol{q})^T \boldsymbol{h}_{\mathcal{T}}$ is the torque vector induced by the interaction $\boldsymbol{h}_{\mathcal{T}} \in \mathbb{R}^6$ with the environment to carry out a given task $\mathcal{T}$, while $\boldsymbol{\tau}_h = \boldsymbol{J}_P(\boldsymbol{q})^T \boldsymbol{h}_h$ is the torque vector induced by the human wrench $\boldsymbol{h}_h \in \mathbb{R}^6$ exerted at a generic point P along the robot structure, with $\boldsymbol{J}_P(\boldsymbol{q}) \in \mathbb{R}^{6\times n}$ the robot Jacobian matrix at this point.

As in Assumption 3.1, the robot is assumed to be able to track a joint space reference trajectory through an inner motion control loop. By virtue of this, the robot model can be simplified to

$$\ddot{\boldsymbol{q}} = \boldsymbol{y} \tag{4.62}$$

where $\boldsymbol{y} \in \mathbb{R}^n$ is a virtual control input to be designed. Finally, the well-known second order kinematic relationship in eq. (3.13) is generalized at a generic point P on the robot structure as follows

$$\ddot{\boldsymbol{x}}_P = \boldsymbol{J}_P(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_P(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \boldsymbol{J}_P(\boldsymbol{q})\boldsymbol{y} + \dot{\boldsymbol{J}}_P(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} \tag{4.63}$$

where $\boldsymbol{x}_P = \left[\boldsymbol{p}_P^T \; \boldsymbol{\varphi}_P^T\right]^T \in \mathbb{R}^6$ is the configuration of a frame centered in P with position $\boldsymbol{p}_P$ and orientation $\boldsymbol{\varphi}_P$. In the following, and as in the previous sections, the subscript P is omitted when the kinematics is referred to the robot end effector.

Human torque and wrench estimation

In order to detect and recognize the type of contact, the estimation of the human torque and wrench is needed. To this aim, the same momentum-based observer introduced in eq. (4.26) is adopted, which leads to the following residual vector $\boldsymbol{\theta}(t)$

$$\boldsymbol{\theta}(t) \approx (\boldsymbol{\tau}_{\mathcal{T}}(t) + \boldsymbol{\tau}_h(t)). \tag{4.64}$$

Assume that an estimate $\hat{\boldsymbol{\tau}}_{\mathcal{T}}$ of $\boldsymbol{\tau}_{\mathcal{T}}$ is available (details in Section 4.5.2) and that, in the case of human interaction, the contact point P along the robot structure is estimated (details in Section 4.5.2). From (4.64), an estimate $\hat{\boldsymbol{\tau}}_h$ of $\boldsymbol{\tau}_h$ is obtained as

$$\hat{\boldsymbol{\tau}}_h(t) = \boldsymbol{\theta}(t) - \hat{\boldsymbol{\tau}}_{\mathcal{T}}(t) \tag{4.65}$$

based on which the components of $\boldsymbol{h}_h(t)$ not belonging to the null space of $\boldsymbol{J}_P^T$ can be retrieved [134] as

$$\hat{\boldsymbol{h}}_h(t) = \left(\boldsymbol{J}_P\left(\boldsymbol{q}(t)\right)^T\right)^{\dagger} \hat{\boldsymbol{\tau}}_h(t). \tag{4.66}$$

Problem formulation and solution overview

Formally, the following problem is addressed.

Problem 4.3. *Consider the robot dynamics in* (4.62) *and a task represented by a desired end effector trajectory* $\boldsymbol{x}_d(t)$ $(\dot{\boldsymbol{x}}_d(t), \ddot{\boldsymbol{x}}_d(t))$ *which can possibly involve interaction* $\boldsymbol{h}_{\mathcal{T}}$ *with the environment. Assume that a human operator can intentionally or accidentally physically interact with any point of the robot structure, i.e.,* $\|\boldsymbol{\tau}_h\| > 0$. *The aim is to design a strategy enabling the robot to detect and classify possible contacts with the human and to undertake proper reaction behaviors, while complying with possible robot constraints.*

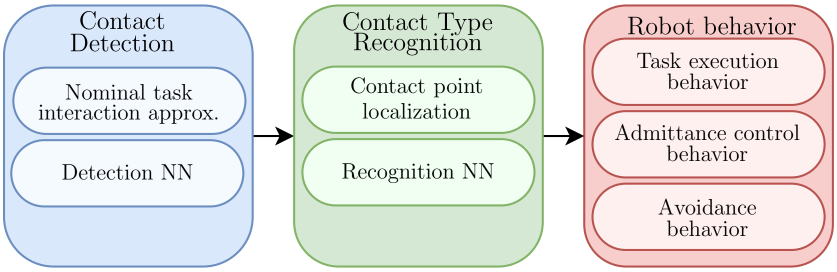


Figure 4.25: Overview of the proposed approach to detect human contact (in blue), recognize its type (in green) and react accordingly (in red).

As shown in Figure 4.25, a modular architecture composed of three blocks is proposed to solve the problem above. In detail, a *detection module* is in charge of assessing whether or not a contact with a human operator is taking place by considering that the robot might be interacting with the environment. Next, a *recognition module* establishes the nature of the human interaction when contact is detected. To this aim, the contact point along the robot structure is localized, and the estimated wrench exerted by the human is evaluated according to (4.66) to evaluate whether or not the contact is intentional. Finally, a *behavior module* determines the robot virtual input $\boldsymbol{y}$ in (4.62). In detail, the robot is endowed with three basic behaviors, namely task execution, admittance control and avoidance behaviors, that are selected considering the type of contact (if any) and the current task.

Differently from the existing approaches mentioned in Section 4.1, the proposed solution introduces the following *novel* contributions: *(i)* the human-robot contact is detected and classified even in the case the robot is in contact with the environment; this is achieved without adopting additional sensors and by resorting to Gaussian Mixture Models (GMMs) for modeling the contact required by the robot task; *(ii)* a comprehensive control strategy is devised to handle both type of contacts while taking into account the contact point along the robot structure, the current task and the robot constraints.

4.5.2 Contact detection and recognition

This section details the components related to the detection and recognition modules of the proposed solution.

Nominal task interaction approximation

In order to assess whether or not a contact with a human operator is occurring, an approximation $\hat{\boldsymbol{\tau}}_{\mathcal{T}}$ (blue block of Figure 4.25) of the expected external torque $\boldsymbol{\tau}_{\mathcal{T}}$, arising from the interaction with the environment for task $\mathcal{T}$, is defined. Indeed, based on this estimation and on the estimated overall external torque in (4.64), the estimated torques induced by the *human* interaction $\hat{\boldsymbol{\tau}}_h$ can be derived.

Gaussian Mixture Models [135] are considered to approximate the nominal wrench profile. In this way, a *rough* definition of the expected nominal profile can be derived on the basis of *few* demonstrations of the nominal interaction with the environment, as shown in the experiments in Section 4.5.4.

Let $\boldsymbol{\zeta}_{\mathcal{T},t} = [t \ \boldsymbol{h}_{\mathcal{T},n}(t)^T]^T \in \mathbb{R}^7$ be the extended vector of the nominal wrench of task $\mathcal{T}$ at time t. This can be probabilistically modeled as a mixture of $K_{\mathcal{T}}$ Gaussian distributions as follows

$$p(\boldsymbol{\zeta}_{\mathcal{T},t}) = \sum\nolimits_{k=1}^{K_{\mathcal{T}}} \pi_{\mathcal{T},k} \, \mathcal{N}(\boldsymbol{\zeta}_{\mathcal{T},t}|\boldsymbol{\mu}_{\mathcal{T},k}, \boldsymbol{\Sigma}_{\mathcal{T},k}) \tag{4.67}$$

where $\mathcal{N}(\boldsymbol{\zeta}_{\mathcal{T},t}|\boldsymbol{\mu}_{\mathcal{T},k}, \boldsymbol{\Sigma}_{\mathcal{T},k})$ is the Gaussian distribution with mean $\boldsymbol{\mu}_{\mathcal{T},k} \in \mathbb{R}^7$ and covariance $\boldsymbol{\Sigma}_{\mathcal{T},k} \in \mathbb{R}^{7\times 7}$ and $\pi_{\mathcal{T},k}$ is the k th mixture coefficient such that $\pi_{\mathcal{T},k} \in [0, 1]$ and $\sum_{k=1}^{K_{\mathcal{T}}} \pi_{\mathcal{T},k} = 1$.

The Expectation-Maximization (EM) algorithm [136] is then leveraged to estimate the GMM parameters $\boldsymbol{\mu}_{\mathcal{T},k}$, $\boldsymbol{\Sigma}_{\mathcal{T},k}$ and $\pi_{\mathcal{T},k}$ $\forall k \in \{1, \dots, K_{\mathcal{T}}\}$ based on demonstrations of $\boldsymbol{\zeta}_{\mathcal{T},t} \, \forall t$. To gather the latter, the task $\mathcal{T}$ is executed few times in the absence of human wrench, and the residual vector $\boldsymbol{\theta}(t)$ in (4.26) is recorded; the latter coincides with the estimate of $\boldsymbol{\tau}_{\mathcal{T}}(t)$ when $\boldsymbol{\tau}_h = \mathbf{0}_n$, being $\mathbf{0}_n$ the null vector of n elements. Each datapoint $\boldsymbol{h}_{\mathcal{T},n}(t)$ in the demonstrations of $\boldsymbol{\zeta}_{\mathcal{T},t}$ is then obtained as

$$\boldsymbol{h}_{\mathcal{T},n}(t) = \left(\boldsymbol{J}\left(\boldsymbol{q}(t)\right)^T\right)^{\dagger} \boldsymbol{\theta}(t). \tag{4.68}$$

With no loss of generality, it is considered that the *expected* physical interaction with the environment occurs at the end effector of the robot since its task is assigned in the Cartesian space. The choice of approximating the expected wrench $\boldsymbol{h}_{\mathcal{T},n}$ instead of the expected external torque $\boldsymbol{\tau}_{\mathcal{T}}$ enables the proposed methodology to be independent from the joint space configuration while executing the task.

Finally, Gaussian Mixture Regression, as done for instance in [137], is used to get the probabilistic model of $\boldsymbol{h}_{\mathcal{T},n}(t)$ at each time t, with mean $\hat{\boldsymbol{h}}_{\mathcal{T},n}(t) = \mathbb{E}(\boldsymbol{h}_{\mathcal{T},n}(t))$. The estimate $\hat{\boldsymbol{\tau}}_{\mathcal{T}}(t)$ in eq. (4.65) is then achieved, i.e., $\hat{\boldsymbol{\tau}}_{\mathcal{T}}(t) = \boldsymbol{J}(\boldsymbol{q}(t))^T \hat{\boldsymbol{h}}_{\mathcal{T},n}(t)$.

Detection and recognition classifiers

As mentioned above, the estimated torques $\hat{\boldsymbol{\tau}}_{\mathcal{T}}$ are exploited to compute the human torques $\hat{\boldsymbol{\tau}}_h$. Based on the latter, two Recurrent Neural Networks (RNNs) are trained to determine the presence of a human contact and its type. In detail, the first network, referred to as *detection NN*, outputs the information on whether or not human contact is occurring, while the second network, referred to as *recognition NN*, outputs the information on whether the contact is intentional or accidental. In light of the binary classification problems, the two networks are trained to minimize the log loss. The same architecture is devised for both the networks and it comprises: *(i)* a recurrent Long Short Term Memory layer, that enables to learn long-term dependencies between data samples in the time series, *(ii)* a fully connect layer, *(iii)* a logistic activation layer that outputs the prediction.

The training dataset of the detection NN consists of labeled time series representing the norm of the estimated human torque, i.e., $\|\hat{\boldsymbol{\tau}}_h(t)\|$ in eq. (4.65), while estimated human wrenches are considered for the recognition NN, i.e., the dataset consists of the datapoints $\|\hat{\boldsymbol{h}}_h(t)\|$ in eq. (4.66). The use of norm values allows to take into account the intensity and variation of the human interaction ignoring the directions in which the interaction occurs. The datasets were collected by voluntarily and accidentally interacting with different points along the robot structure during free space motion (i.e., with $\|\boldsymbol{\tau}_{\mathcal{T}}\| = 0$) and recording the labeled respective quantities. Note that the datasets only contain data acquired with no environment interaction; however, the proposed approach is able to tackle the case of $\|\boldsymbol{\tau}_{\mathcal{T}}\| > 0$ thanks to the GMM modeling and without wrist force/torque sensors. In this way, an arbitrary number of tasks can be performed that require interaction with the environment *without* the need of re-training the networks and adapting their datasets. Finally, note that the generalization capabilities of the networks allows to tackle, without manually tuned thresholds, noisy measures, model uncertainties in (4.61) and task interaction estimation inaccuracies which generally leads to not accurate human torque estimation, i.e., it generally holds $||\hat{\boldsymbol{\tau}}_h|| > 0$ even if no human-robot contact is happening.

Contact point localization

In case of human contact, a procedure is needed to localize the contact point P on the robot structure. This information is then necessary to estimate the human wrench in (4.66) used by the recognition classifier and the behavior module. Different techniques can be used to the purpose relying on proper perception devices such as IMU sensors [138] or vision systems. The latter case is considered in this thesis and in the experiments reported in Section 4.5.4. In particular, as in Section 3.2, the human can be represented as a set of relevant points $\boldsymbol{p}_{o,j}$ whose position over time is provided by a vision-based human skeleton tracking algorithm. Based on the points $\boldsymbol{p}_{o,j}$ and by approximating each robot link l as a segment with starting point $\boldsymbol{p}_{l,0}$ and ending $\boldsymbol{p}_{l,1}$ (as in Section 3.2), the contact point P can be defined as

$$P = ||\boldsymbol{p}_{l^*,0} + r^*(\boldsymbol{p}_{l^*,1} - \boldsymbol{p}_{l^*,0}) - \boldsymbol{p}_{o,j^*}|| \tag{4.69}$$

where l^*,r^* and j^* are obtained as solution of the following optimization problem

$$\begin{aligned}(l^*, r^*, j^*) = \arg\min_{l,r,j} \quad & ||\boldsymbol{p}_{l,0} + r(\boldsymbol{p}_{l,1} - \boldsymbol{p}_{l,0}) - \boldsymbol{p}_{o,j}||. \\ s.t. \quad & r \in \big[0, 1\big]\end{aligned} \tag{4.70}$$

which implies that the contact point is defined by considering the point on the robot structure (identified by the link l^* and the curvilinear abscissa r^*) which is closest to any of the human points $\boldsymbol{p}_{o,j}, \forall j$.

4.5.3 Robot behaviors

Three basic robot behaviors (red block of Figure 4.25) are envisaged which are activated depending on the type (if any) of interaction. In the following, the individual behaviors are first introduced and, then, their activation activation rationale as well as the handling of possible robot and task constraints are presented.
The following behaviors are identified.

1) Task execution behavior

The task execution behavior allows the robot to perform its desired task which, as stated in Problem 4.3, is given by the desired trajectory at the effector $\boldsymbol{x}_d(t)$. The closed loop inverse kinematic law is leveraged to define the respective virtual input $\boldsymbol{y}$ in eq. (4.62):

$$\boldsymbol{y}_n = \ddot{\boldsymbol{q}}_{\mathcal{T}_n} + \ddot{\boldsymbol{q}}_N, \quad \ddot{\boldsymbol{q}}_{\mathcal{T}_n} = \boldsymbol{J}^{\dagger}\Big(\ddot{\boldsymbol{x}}_d + \boldsymbol{K}_d\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_p\tilde{\boldsymbol{x}} - \dot{\boldsymbol{J}}\dot{\boldsymbol{q}}\Big) \tag{4.71}$$

where $\tilde{\boldsymbol{x}}(t) = \boldsymbol{x}_d(t) - \boldsymbol{x}(t) \in \mathbb{R}^6$ is the task tracking error, $\boldsymbol{K}_d \in \mathbb{R}^{6\times6}$, $\boldsymbol{K}_p \in \mathbb{R}^{6\times6}$ are positive definite matrices, and $\ddot{\boldsymbol{q}}_N \in \mathbb{R}^n$ is an arbitrary vector of joint accelerations which can be used for secondary tasks.

When this behavior is active, the human operator is also allowed to reconfigure the *internal* structure of the robot without altering the robot end effector task. In particular, it is considered that, in order to express this intention, the human intentionally interacts with the point P of the robot structure (not coinciding with the end effector) that he/she wants to reconfigure. To this aim, the following joint accelerations $\ddot{\boldsymbol{q}}_N$ [139] in eq. (4.71) are defined:

$$\ddot{\boldsymbol{q}}_N = \big(\boldsymbol{J}_P(\boldsymbol{I}_n - \bar{\boldsymbol{J}}^{\dagger}\bar{\boldsymbol{J}})\big)^{\dagger}\left(\boldsymbol{M}_d^{-1}\left(-\boldsymbol{D}_d\dot{\boldsymbol{x}}_P + \hat{\boldsymbol{h}}_h\right) - \dot{\boldsymbol{J}}_P\dot{\boldsymbol{q}} - \boldsymbol{J}_P\ddot{\boldsymbol{q}}_{\mathcal{T}_n}\right)$$

where $\boldsymbol{M}_d, \boldsymbol{D}_d \in \mathbb{R}^{6\times6}$ are positive definite matrix gains, and $\bar{\boldsymbol{J}}$ coincides with the positional Jacobian matrix at the end effector, if the task orientation can be relaxed, and with the full Jacobian matrix $\boldsymbol{J}$ (in case of redundant robots), otherwise. The formulation in (4.72) makes the robot compliant at the contact point P, enabling the internal reconfiguration by the human, while preserving the respective components of the desired task.

2) Admittance control behavior

This behavior allows the human to voluntary modify the robot task, i.e., when the human intentionally interacts with the robot end effector, it is assumed that he/she wants to intervene in the robot task and correct its execution. To this aim, a simple admittance controller is realized which allows the human to guide the robot motion according to a mass-damper model. In particular, the following formulation for the input $\boldsymbol{y}$ in eq. (4.62) is used:

$$\boldsymbol{y}_a = \ddot{\boldsymbol{q}}_{\mathcal{T}_a} + \ddot{\boldsymbol{q}}_N, \ \ddot{\boldsymbol{q}}_{\mathcal{T}_a} = \boldsymbol{J}^{\dagger}\left(\boldsymbol{M}_d^{-1}\left(-\boldsymbol{D}_d\dot{\boldsymbol{x}} + \hat{\boldsymbol{h}}_h\right) - \dot{\boldsymbol{J}}\dot{\boldsymbol{q}}\right) \tag{4.72}$$

where, as above, $\ddot{\boldsymbol{q}}_N \in \mathbb{R}^n$ is an arbitrary vector of joint accelerations, while $\boldsymbol{M}_d, \boldsymbol{D}_d \in \mathbb{R}^{6\times 6}$ are the positive definite desired inertia and damping, respectively. Note that also the methods presented in Sections 4.3 and 4.4 can be adopted for this behavior. By replacing (4.72) in (4.63), the closed loop model, analogous to (4.16), is derived:

$$\boldsymbol{M}_d\,\ddot{\boldsymbol{x}} + \boldsymbol{D}_d\,\dot{\boldsymbol{x}} = \hat{\boldsymbol{h}}_h. \tag{4.73}$$

3) Avoidance behavior

An avoidance behavior is activated when an accidental collision occurs. To this aim, the concept of safety field $F(t)$ introduced in Section 3.2 is considered to assess the level of human safety with respect to the robot in a comprehensive manner, i.e., the entire human body and robot structure are taken into account. The objective of the avoidance behavior is to recover a safety condition $F \geq F_d$ after the occurrence of the accidental contact, with F_d a positive threshold to be tuned. For this purpose, an evasive action is realized, since the velocity modulation is not suitable *after* the collision occurrence. More specifically, the following virtual input $\boldsymbol{y}_o$ is defined

$$\boldsymbol{y}_o = \ddot{\boldsymbol{q}}_{\mathcal{T}_o} + \ddot{\boldsymbol{q}}_N, \;\; \ddot{\boldsymbol{q}}_{\mathcal{T}_o} = \boldsymbol{J}_F^{\dagger}\Big(k_d\dot{\Delta}F + k_p\Delta F - \dot{\boldsymbol{J}}_F\dot{\boldsymbol{q}}\Big) \tag{4.74}$$

where $\boldsymbol{J}_F \in \mathbb{R}^{1\times n}$ is the safety field Jacobian matrix such that $\dot{F} = (\partial F/\partial \boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{J}_F\dot{\boldsymbol{q}}$, $\Delta F = \min(0, F_d - F)$, k_p, k_d are positive gains, and $\ddot{\boldsymbol{q}}_N \in \mathbb{R}^n$ is an additional acceleration vector. The rationale behind (4.74) is that, as long as $F < F_d$, the robot is driven away from the human operator in order to increase his/her safety and restore the minimum index F_d. Then, when $F \geq F_d$, no action is carried out anymore to increase the human safety. In detail, by replacing (4.74) in (4.62), the safety field dynamics is

$$\ddot{\Delta}F + k_d\dot{\Delta}F + k_p\Delta F = 0 \tag{4.75}$$

implying the asymptotic convergence of ΔF to the origin.

The activation of the appropriate behavior on the basis of the human-robot interaction state is regulated by the Finite State Machine (FSM) in Figure 4.26. More specifically, the robot task is carried out according to (4.71)-(4.72) as long as no human interaction is detected, or if a force is voluntarily applied along the robot structure to internally reconfigure it. Then, in the case an intentional contact is recognized at the end effector, the admittance behavior in (4.72) is selected which enables the human to adjust the robot task. The robot persists in this state until the human exerts wrenches at the contact point. Finally, if an accidental contact is recognized, the robot selects the avoidance behavior to increase the human safety field F. The robot persists in this state as long as a safety condition is not restored, i.e., as long as $F < F_{min}$, with $F_{min} > F_d$ a scalar threshold.

In order to handle possible constraints, depending on the task itself and on the nature of the interaction (see Section 4.5.4 for

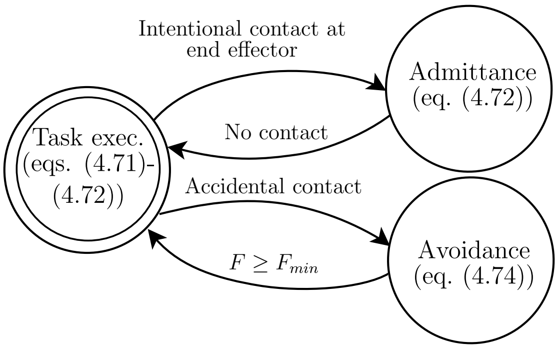


Figure 4.26: FSM for robot's behaviors activation.

examples), the system in (4.62) is rewritten in the form

$$\dot{\boldsymbol{\xi}}_q = \boldsymbol{f}(\boldsymbol{\xi}_q) + \boldsymbol{g}(\boldsymbol{\xi}_q)(\boldsymbol{y}) = \begin{bmatrix} \boldsymbol{O}_n & \boldsymbol{I}_n \\ \boldsymbol{O}_n & \boldsymbol{O}_n \end{bmatrix} \boldsymbol{\xi}_q + \begin{bmatrix} \boldsymbol{O}_n \\ \boldsymbol{I}_n \end{bmatrix} \boldsymbol{y} \tag{4.76}$$

with $\boldsymbol{\xi}_q = [\boldsymbol{q}^T \;\; \dot{\boldsymbol{q}}^T]^T \in \mathbb{R}^{2n}$, and $\boldsymbol{O}_m$ ($\boldsymbol{I}_m$) the $m \times m$ null (identity) matrix. The i th constraint is expressed as

$$\boldsymbol{\phi}_i(\boldsymbol{\xi}_x(\boldsymbol{\xi}_q)) \geq 0 \tag{4.77}$$

where $\boldsymbol{\xi}_x = [\boldsymbol{x}_P^T \;\; \dot{\boldsymbol{x}}_P^T]^T \in \mathbb{R}^{12}$ and $\boldsymbol{\phi}_i(\cdot)$ is a continuous scalar function. In order to have these constraints satisfied, the Control Barrier Function (CBF) approach [140] is adopted. According to the latter, the following sets are defined

$$\begin{aligned} \mathcal{C} &= \{\boldsymbol{\xi}_x \in \mathbb{R}^{12} : \boldsymbol{\phi}_i(\boldsymbol{\xi}_x) \geq \boldsymbol{0}\} \\ \partial\mathcal{C} &= \{\boldsymbol{\xi}_x \in \mathbb{R}^{12} : \boldsymbol{\phi}_i(\boldsymbol{\xi}_x) = \boldsymbol{0}\} \\ \mathrm{Int}(\mathcal{C}) &= \{\boldsymbol{\xi}_x \in \mathbb{R}^{12} : \boldsymbol{\phi}_i(\boldsymbol{\xi}_x) > \boldsymbol{0}\} \end{aligned} \tag{4.78}$$

which implies that the state $\boldsymbol{\xi}_x$ is required to belong to the set $\mathcal{C}$ in order to satisfy constraints. By following the steps in [140], it can be proved that the control input $\boldsymbol{y}^\star$ which achieves the task function belonging to the set $\mathcal{C}$ can be computed as the solution of the following Quadratic Program problem

$$\begin{aligned} \boldsymbol{y}^{\star} &= \arg\min_{\boldsymbol{y}} \frac{1}{2} \left(\boldsymbol{y} - \boldsymbol{y}_{(\cdot)}\right)^{T} \boldsymbol{Q} \left(\boldsymbol{y} - \boldsymbol{y}_{(\cdot)}\right) \\ s.t. \quad & L_f\phi_i + L_g\phi_i \boldsymbol{y} \geq -\gamma(\phi_i(\boldsymbol{\xi})),\ \forall i \end{aligned} \tag{4.79}$$

where $\boldsymbol{y}_{(\cdot)}$ is the desired input computed according to eqs. (4.71), (4.72) or (4.74) in dependence of the FSM state, $\boldsymbol{Q} \in \mathbb{R}^{n \times n}$ is a positive definite matrix, $\gamma(\cdot)$ is an extended $\mathcal{K}_{\infty}$ class function and $L_f\phi_i$, $L_g\phi_i$ are the Lie derivatives of ϕ_i with respect to f and g, respectively. Here, the input $\boldsymbol{y}_{(\cdot)}$ is obtained in accordance to the current state of the system in the FSM (Figure 4.26), i.e., according to (4.71)-(4.72) in the task execution state, to (4.72) in the admittance behavior state and to (4.74) in the human avoidance state. The constraints $\boldsymbol{\phi}_i$ are defined according to the specific task to execute and setup, as shown in the following section, and the respective Lie derivatives are computed. The resulting $\boldsymbol{y}^{*}$ is then provided as input to the virtual model in (4.62).

4.5.4 Experimental validation

The experimental validation of the approach is now provided. The entire experiment execution can be also visualized in the video at the provided link[4].

System architecture and robot tasks

The system setup, illustrated in Figure 4.27, is composed of a Kinova Jaco2 robotic arm with 7 DOFs ($n = 7$), previously used to validate the FDI strategy in Section 2.5, and a Microsoft Kinect One RGB-D sensor.

A scheme of the overall system architecture is shown in Fig. 4.28. In detail, all the algorithms related to the contact classification and to the Jaco2 control run on Matlab 2020 and exchange information with Robot Operating System (ROS). The

[4] `https://youtu.be/cf4ecX-_gHY`

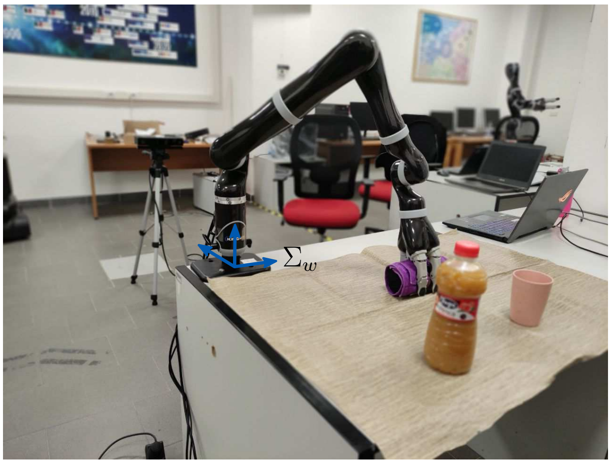


Figure 4.27: Experimental setup composed of a Kinova Jaco2 and a Microsoft Kinect One RGB-D sensor. The world frame Σ_w is also shown in blue.

latter is also used by the localization component which uses the RGB-D sensor for identifying the world frame Σ_w using markers and ArUco libraries [141] and for tracking the human skeleton through a proper algorithm[5]. In detail, the following relevant points are considered for the human: torso, left and right hand, elbow, shoulder.

A domestic collaborative scenario is considered where the following tasks can be performed:

1. *Object pick and place*
   The robot can manipulate bottles and mugs in the scene. However, to prevent the liquid from spilling out of the objects, the following orientation constraints are considered:

$$\begin{aligned} \phi^l_{o,i} &= \varphi_i - (\varphi_{d,i} - \Delta\varphi_i) \geq 0 \\ \phi^u_{o,i} &= (\varphi_{d,i} + \Delta\varphi_i) - \varphi_i \geq 0 \end{aligned}, \qquad i = 1, 2, 3 \tag{4.80}$$

   which are in the same form as in (4.77), being $\boldsymbol{\varphi}_d \in \mathbb{R}^3$ a desired orientation, $\Delta\varphi_i$ a positive tolerance and $(\cdot)_i$ the

[5] https://github.com/mcgi5sr2/kinect2_tracker

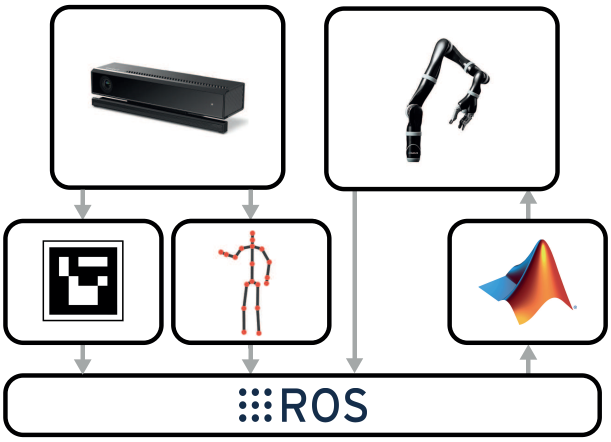


Figure 4.28: Experimental system architecture. A Microsoft Kinect One is used to detect a marker to define the common frame and track the human skeleton movements. The information is sent to ROS, which also communicate with the classification and control algorithms running on MATLAB 2020. The latter generates desired joint velocities for the robot Kinova Jaco2.

ith component of the respective vector. The above constraints have a relative degree equal to 2 which are handled as in [142].

2. *Pouring task*
   This task foresees the robot holding a mug and the human filling it. The nominal wrench profile due to the filling is approximated with $K_{\mathcal{T}} = 5$ Gaussians and by using 5 randomly selected demonstrations. The GMM parameters for the EM algorithm are initialized by resorting to a K-means clustering algorithm. In order to validate the GMM approximation, 40 additional test samples $\hat{\boldsymbol{h}}_h(t)$ have been considered, and the average approximation error of the quantity $\|\hat{\boldsymbol{h}}_h(t) - \hat{\boldsymbol{h}}_{\mathcal{T},n}(t)\|$, $\forall t$ has been computed. The latter resulted equal to 0.5, confirming the effectiveness of the approximation. Also in this case, the end effector orientation is constrained as in (4.80).

3. *Table cleaning*
   The robot executes a periodical motion on the table surface

Training user

	Detection NN				Recognition NN		
	nc	wc			ic	ac	
nc	4697	373	92.6%	ic	3979	135	96.7%
wc	139	2930	95.5%	ac	555	1826	76.7%
	97.1%	88.7%	93.7%		87.8%	93.1%	89.3%

Novel users

	nc	wc			ic	ac	
nc	6120	456	93.1%	ic	21041	2647	88.8%
wc	2836	19745	87.4%	ac	3669	7016	65.7%
	68.3%	97.7%	88.7%		81.2%	72.6%	81.6%

Table 4.2: Confusion matrices of the RNNs with data acquired by the user involved in the training dataset (top part) and with different users not involved in the training dataset (bottom part).

holding a cylindrical sponge (see Figure 4.24), in such a way to perform a cleaning task. The force exerted on the table is approximated through GMMs by using 5 randomly selected demonstrations with, as before, $K_{\mathcal{J}} = 5$ and K-means algorithm for initialization. The average approximation error is also evaluated, which in this case is equal to 1 considering 100 additional test samples.

Further constraints in the CBF framework are introduced in all the tasks to limit the robot workspace and avoid collisions with the table, i.e., $\phi_t = p_z - h_t \geq 0$ in (4.77) being h_t the height of the table. Note that the human is always allowed to physically interact with the robot end effector, changing its configuration, or with the robot structure, changing its internal joint configuration (without affecting the end effector position). In this way, the user can help the robot to accomplish the task or to avoid possible collisions with obstacles not detected by the vision system.

Classification results

Classification results of the two RNNs (with 100 hidden units) are provided in the following. In particular, the detection NN determines if a datapoint in the time series belongs to the class *no contact* (*nc*) or *with contact* (*wc*), while the recognition network establishes if, in the case of contact, a datapoint belong to the class *intentional contact* (*ic*) or *accidental contact* (*ac*).

The training dataset of the detection (recognition) NN consists of 43487 (15647) time samples, representing 869.7 s (312.9 s) with time step $T = 0.02$ s, equally distributed in their respective two classes. Concerning the test set, the accuracy was evaluated both with the user that was involved in the collection of the training dataset, referred to as *training user*, and with other four users *not* involved in the training process, referred to as *novel users*. Table 4.2 reports the confusion matrices of the classifiers for the training user (top part) and the novel users (bottom part). It can be observed that all the RNNs achieve good overall accuracy which is, for the training user, $\approx 94\%$ for the detection NN and $\approx 90\%$ for the recognition NN. The lower accuracy of the latter is explained by the intrinsic greater complexity of the problem of recognizing the type of contact rather than identifying *any* possible contact. Remarkably, the results also confirm the generalization capabilities of the chosen NNs which allow classification also with novel users. In detail, an overall accuracy equal to $\approx 90\%$ and $\approx 82\%$ is obtained for the detection and recognition classifications, respectively, implying a decrease of performance of only $\approx 4\%$ and $\approx 8\%$, compared to the training user. Note that the results in Table 4.2 are obtained with a sample-by-sample evaluation; however, a certain delay for detecting and classifying a contact always occurs as the networks obviously need some samples before being able to correctly classify. This explains the classification inaccuracies in Table 4.2 but, as demonstrated in the following case studies, it does not undermine the human-robot collaboration.

Concerning possible misclassifications, in the case they occur in the recognition network, two possibilities exist: *i)* an intentional contact is recognized as accidental, and *ii)* an accidental contact is recognized as intentional. The former does not compromise human safety as it only leads the robot to move unnecessarily away from the person. The latter instead may be critical. However, the admittance behavior leads the robot to become compliant towards the human force, thus reducing anyhow possible human injuries compared to the rigid case. In the case of misclassification in the detection network, which barely occurs, no particular issues exist when a false positive is detected (i.e., the network detects a contact that is not actually taking place) but critical safety issue may arise when a false negative (i.e., the network does not detect a contact) occurs. However, if the force magnitude increases due to continuous contact with the person, it is very likely that this contact, albeit with a certain delay, is then recognized by the network, allowing robot to react appropriately.

Case studies and experimental results

Two case studies are analyzed for the experimental validation which involve different tasks and human interactions. In both case studies, the following parameters has been used: $\boldsymbol{K}_\theta = 50\boldsymbol{I}_7$ in eq. (4.26), $\boldsymbol{K}_d = 5\boldsymbol{I}_6$, $\boldsymbol{K}_p = 6\boldsymbol{I}_6$ in (4.71), $\boldsymbol{M}_d = 5\boldsymbol{I}_6$, $\boldsymbol{D}_d = 100\boldsymbol{I}_6$ in (4.72) and (4.72), $k_d = 5$, $k_p = 6$ in (4.74), $\boldsymbol{Q} = \boldsymbol{I}_7$ in (4.79). Moreover, according to the computations in Appendix B, it is set $F_d = 10$ in (4.74) to ensure a minimum distance ≈ 0.4 m between every point of the robot and the human operator, and $F_{min} = 11$ in the FSM.

Case study 1

In this case study, the robot executes three different tasks, while the human interacts with it. At the beginning, the robot takes the cylindrical sponge and starts a cleaning task, during which the human intentionally interacts with its end effector to change its configuration. Once the cleaning task is completed, the robot starts a bottle pick and place operation. During this task execution, the human voluntarily interacts along the robot structure to modify the joint configuration while preserving the end effector position. Finally, a pick operation of a mug and a pouring task are performed. During each task, the respective constraints and possible expected environment interaction are taken into account.

Figure 4.29 summarizes the results of this case study. In detail, proceeding from the top to the bottom, it reports the robot end effector trajectory $\boldsymbol{x}$, the nominal interaction wrench $\hat{\boldsymbol{h}}_{\mathcal{T}}$, the norm of estimated human torque $\|\hat{\boldsymbol{\tau}}_h\|$ and the classification output of the RNNs (in blue) compared to the Ground Truth (GT, in green). Initially, the robot autonomously performs the cleaning task (highlighted with red boxes in the plots) and executes the desired periodic motion (first plot) which generates interaction wrenches with the table. The use of the nominal wrench profile (second plot) modeled via GMM allows to estimate the human torque (third plot) and to properly recognize that, in the initial phase, only interaction with the environment is occurring while no human contact is present (last plot). Once the human intentionally interacts with the end effector at $t \approx 21$ s, it is recognized by the RNNs (with a delay < 1 s) and the admittance behavior is activated. This makes the robot end effector compliant towards the human wrench and its trajectory is modified as shown in the first plot. At the end of the interaction, the execution of the desired tasks is restored and the pick and place operations are carried out (highlighted with green boxes) from $t \approx 27$ s to $t \approx 100$ s. In this phase, an intentional contact on link 4 of the robot is recognized via the localization component at $t \approx 70$ s, which leads the robot to change its joint configuration according to (4.71) and (4.72) while preserving its end effector position, as shown in the top plot. Orientation variables are also limited according to (4.80) with $\boldsymbol{\varphi}_d = [-1.4,\, 1.6,\, -2.9]^T$ and $\Delta\varphi_i = 0.2\,\forall i$. Finally, the pouring task (blue box in the plots) is executed starting from $t \approx 100$ s during which the respective GMM model is exploited and no contact with the human is detected.

Case study 2

This case study aims to prove the effectiveness of the solution also with accidental contacts. It is structured as follows: the human first intentionally interacts with the robot end effector during a

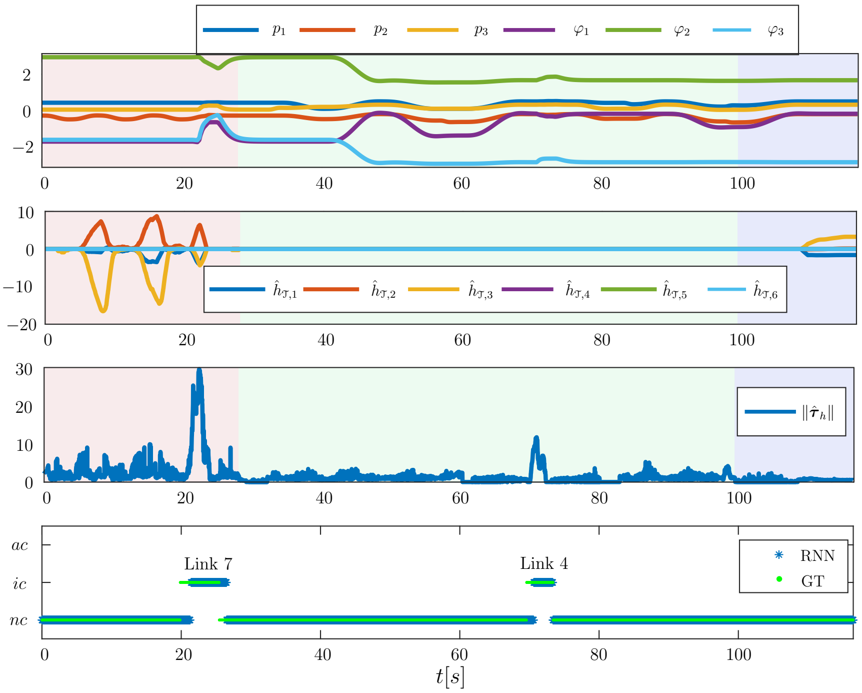


Figure 4.29: Case study 1. From the top: end effector trajectory, nominal task interaction wrench, norm of the estimated human torques and classification results compared to ground truth. Cleaning, pick and place and pouring phases are denoted in red, green and blue.

pick and place task, and then an accidental collision happens.

Figure 4.30 shows, from the top, the robot end effector trajectory $\boldsymbol{x}$, the norm of the estimated human torques $\|\hat{\boldsymbol{\tau}}_h\|$ and the classification results (in blue), compared to ground truth values (in green). Based on the estimated human wrench (second plot) and on the localization component, an intentional contact at the end effector is recognized by the RNNs (third plot) at $t \approx 9$ s; then, the admittance behavior is activated, leading to a modification of the robot trajectory. The robot task execution is then restored when the interaction terminates and an accidental contact is detected on link 7 at $t \approx 33$ s. This activates the avoidance behavior which drives the robot away from the human operator in order to increase the safety (see also video at the link [4]).

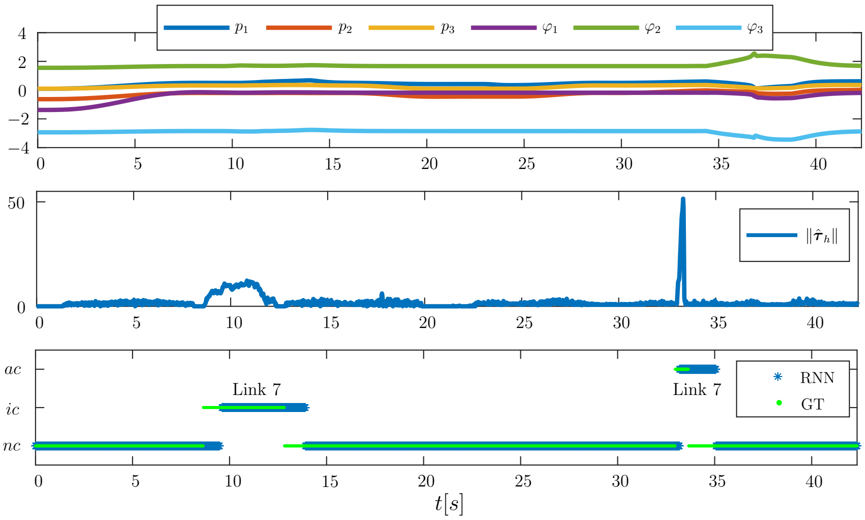


Figure 4.30: Case study 2. From the top: end effector trajectory, estimated human torques and classification results compared to ground truth.

The experiments thus validate the approach which allows an effective human-robot collaboration where both intentional and accidental contacts can happen.

Chapter 5

Planning through human demonstrations

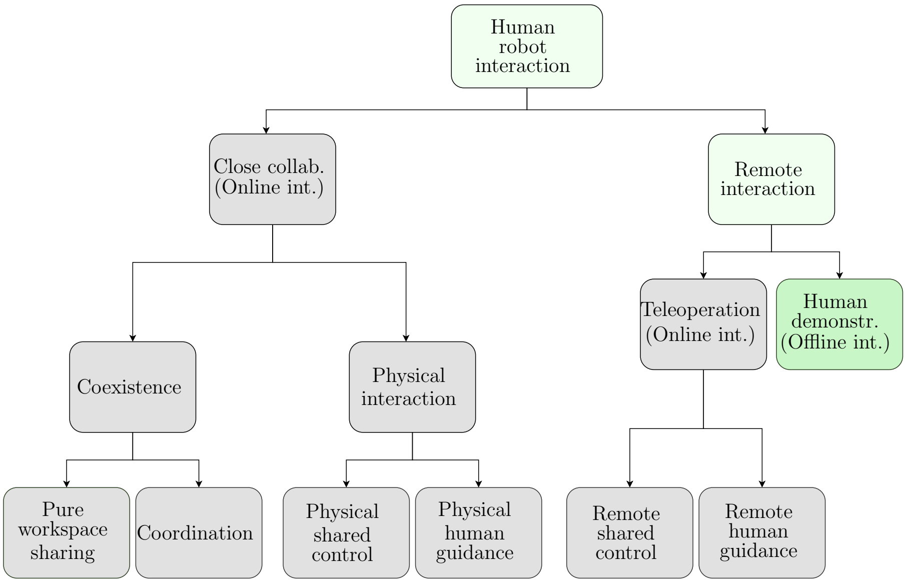


Figure 5.1: Taxonomy for HRI scenarios where the one considered in this chapter is highlighted in green.

After having addressed the themes of human-robot workspace sharing in Chapter 3 and physical interaction in Chapter 4, this chapter represents the final step towards the maximum benefit of human-robot collaboration, that is learning from human skills. In particular, as highlighted in green in the taxonomy in Figure 5.1, it is here considered that the human operator provides examples to the robot about how to perform a given task, rather than collaborating side-by-side with it. These examples are provided in the form of high-dimensional raw observations, such as images or video sequences. This is especially useful in complex scenarios where explicit analytical modeling of states is challenging, causing difficulties in planning and control processes. This is the case, for example, of tasks which involve manipulation of *highly deformable* objects, which is currently an active research field [143]. A first strategy to mitigate this representation problem to a certain extent is to use sampling-based planning algorithms [144], which randomly sample the state space and avoid representing it explicitly. However, these approaches become intractable [145] when dealing with higher-dimensional spaces and more complex systems.

For this reason, data-driven low-dimensional latent space representations for planning are receiving increasing attention as they allow to implicitly represent complex state spaces enabling an automatic extraction of lower-dimensional state representations [146]. Some of the most common approaches to learning compact representations in an unsupervised fashion are latent variable models such as Variational Autoencoders (VAEs) [147, 148] or encoder-decoder based Generative Adversarial Networks (GANs) [149, 150]. These models can learn low-dimensional state representations directly from images instead of a separate perception module. In this way, images can be used as input for planning algorithms to generate *"visual action plans"* [151, 152] as depicted in Figure 5.2.

Latent state representations, however, are not guaranteed to capture the *global structure* and dynamics of the system, i.e., to encode all the possible system states and respective feasible transitions. Furthermore, not all points in the latent space necessarily correspond to physically *valid* states of the system, which makes it hard to plan by naively interpolating between start and goal states

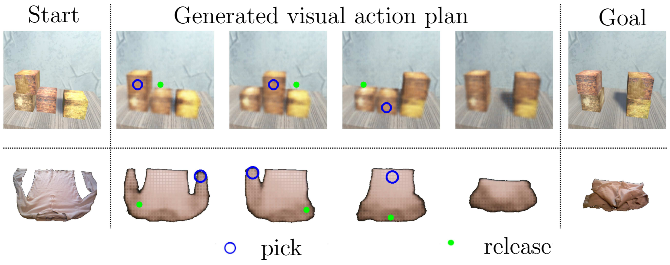


Figure 5.2: Examples of visual action plans for a stacking (top) and a folding (bottom) task.

as shown in the chapter. In addition, the transitions between the generated states might not be valid.

For this reason, a method is presented in the following, based on the works [153] and [154], which builds a graph-based structure in the latent space to plan a manipulation sequence given a start and goal image. This graph is derived on the basis of weak supervision that the human provides: a small number of actions between observation pairs is recorded, and the observations are marked as "same" or "different", depending on whether or not an action is needed to bring the system from one state to the successor one. The approach is validated on a T-shirt folding task requiring manipulation of deformable objects. Note that the same approach can also be used to generate plans that involve the human as an executive agent. In this case, the human role would be two-fold: on the hand he/she would provide examples to the system, on the other hand he/she would also be involved in the execution of the plan. This scenario is subject of on-going work. Finally, in regard to the manipulation of deformable objects, the lack of standardized procedures to evaluate and compare different methodologies was noticed and a benchmark was proposed to the purpose in [155]. The latter is reported in Appendix D for the sake of completeness.

The remainder of the chapter is organized as follows. First, the related work is presented in Section 5.1. Next, the problem of visual action planning is formally stated in Section 5.2 and an overview of its solution is provided, whose respective three modules are detailed in Sections 5.3,5.4 and 5.4. Finally, the effectiveness of the method is shown with two versions of a simulated box stacking task in Section 5.5 and with a real-world folding task in Section 5.6.

5.1 Related work

In task planning, it is common to assume that the state representation is given to the planner as input, and the shape and the poses of objects are known. As an example, robot and grasping knowledge is exploited in [156] to accomplish sequential manipulation tasks, while all possible distributions over the robot state space, namely the belief space, are employed in [157] to tackle partially observable control problems. In addition, sampling-based planning approaches, such as Rapidly exploring Random Tree (RRT) or Rapidly exploring Dense Tree (RDT) [144], can be adopted to probe the state space when an explicit representation of it is difficult to achieve. However, all the above mentioned approaches are generally prone to fail when high-dimensional states, such as images, are involved in the system [158]. For this reason, as mentioned above, the research community is investigating data-driven low-dimensional latent space representations for planning purposes. However, despite the dimensionality reduction, the system's global structure and dynamics may be not properly represented in the latent space. One way to address these shortcomings is to restrict the exploration of the latent space via imitation learning, as presented in [159], where a latent space Universal Planning Network (UPN) that embeds differentiable planning policies is proposed and the process is learned in an end-to-end fashion from imitation learning. The authors then perform gradient descent to find optimal trajectories.

A more common solution to mitigate these challenges is to collect a very large amount of training data that densely covers the state space and allows to infer dynamically valid transitions between states. Following this approach the authors in [151] propose a framework for *global search* in the latent space based on three components: *i)* a latent state representation, *ii)* a network that approximates the latent space dynamics, and *iii)* a collision checking network. Motion planning is then performed directly in the latent space by an RRT-based algorithm. Similarly, a Deep Planning Network is proposed in [160] to perform continuous control tasks where a transition model, an observation model and a reward model in the latent space are learned and are then exploited to maximize an expected reward function. Following the trend of self-supervised learning, the manipulation of a deformable rope from an initial start state to a desired goal state is investigated in [161]. In particular, it builds upon [152], where 500 hours worth of data collection are used to learn the rope inverse dynamics and then produce an *understendable* visual foresight plan for the intermediate steps to deform the rope using a Context Conditional Causal InfoGAN (C^3IGAN). In the proposed framework, neither full imitation nor large amount of data are required for planning but rather a weak supervision is exploited which is given by *pairs* of images and demonstrated actions connecting them, and then learn feasible transitions between states from this *partial* data.

5.2 Problem statement and overview

The goal of visual action planning, also referred to as *"visual planning and acting"* in [161], can be formulated as follows: given start and goal images, generate a path as a sequence of images representing intermediate states and compute dynamically valid actions between them.

Let $\mathcal{I}$ be the space of all possible observations of the system's states represented as images with fixed resolution and let $\mathcal{U}$ be the set of possible control inputs or actions. Formally, the following problem is addressed.

Problem 5.1. *Consider that the images $I_{start}, I_{goal} \in \mathcal{I}$ capturing the start and goal states of the system, respectively, are assigned. The objective is to define a* visual action plan *consisting*

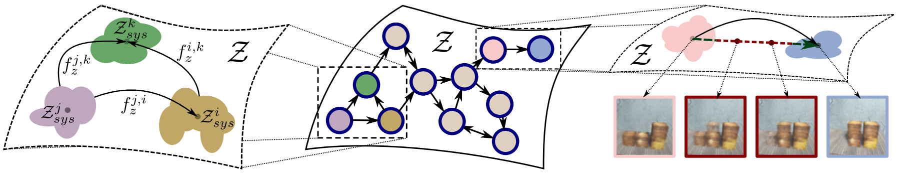


Figure 5.3: Illustrative representation of the latent space $\mathcal{Z}$. In the middle, possible transitions (arrows) between covered regions (sketched with circles) are shown. On the left, details of the covered regions with different shapes and representative points are provided. On the right, observations from a box stacking tasks are shown. In detail, the ones obtained from covered regions (marked in pink and blue) contain meaningful states of the task, while the ones generated from not covered regions (marked in red) show fading boxes that do not represent possible states of the system.

of a visual plan *represented as a sequence of images* $P_I = \{I_{start} = I_0, I_1..., I_N = I_{goal}\}$ *and an* action plan *represented as a sequence of actions* $P_u = \{u_0, u_1, ..., u_{N-1}\}$ *where* $u_n \in \mathcal{U}$ *generates a transition between consecutive states contained in the observations* I_n *and* I_{n+1} *for each* $n \in \{0, ..., N-1\}$.

To retrieve the underlying states represented in the observations as well as to reduce the complexity of the problem, $\mathcal{I}$ is mapped into a lower-dimensional latent space $\mathcal{Z}$ such that each observation $I_n \in \mathcal{I}$ is encoded as a point $z_n \in \mathcal{Z}$ extracting the state of the system captured in the image I_n. Such map is defined *latent mapping* and is denoted by $\xi : \mathcal{I} \to \mathcal{Z}$. In order to generate visual plans, it is additionally assumed the existence of a mapping $\omega : \mathcal{Z} \to \mathcal{I}$ called *observation generator*.

Let $\mathcal{T}_I = \{I_1, ..., I_M\} \subset \mathcal{I}$ be a finite set of input observations inducing a set of *covered states* $\mathcal{T}_z = \{z_1, ..., z_M\} \subset \mathcal{Z}$, i.e., $\mathcal{T}_z = \xi(\mathcal{T}_I)$. In order to identify a set of unique covered states, the following assumption on $\mathcal{T}_z$ is made.

Assumption 5.1. *Let* $z \in \mathcal{T}_z$ *be a covered state. Then, there exists* $\varepsilon_z > 0$ *such that any other state* z' *in the* ε_z-*neighborhood* $N_{\varepsilon_z}(z)$ *of* z *can be considered as the same underlying state.*

This allows both generating a valid visual action plan and taking into account the uncertainty induced by imprecisions in action execution. Let

$$\mathcal{Z}_{sys} = \bigcup_{z \in \mathcal{T}_z} N_{\varepsilon_z}(z) \subset \mathcal{Z} \tag{5.1}$$

be the union of ε_z-neighborhoods of the covered states $z \in \mathcal{T}_z$. Using $\mathcal{Z}_{sys}$, a visual plan can be computed in the latent space using a *latent plan* $P_z = \{z_{start} = z_0, z_1, ..., z_N = z_{goal}\}$, where $z_n \in \mathcal{Z}_{sys}$, which is then decoded with the observation generator ω into a sequence of images.

To obtain a valid visual plan, the structure of the space $\mathcal{Z}_{sys}$ is studied which is general not path-connected. As shown in Figure 5.3 on the right, linear interpolation between two states z_1 and z_2 in $\mathcal{Z}_{sys}$ may result in a path containing points from $\mathcal{Z} - \mathcal{Z}_{sys}$ that do not correspond to covered states of the system and are therefore not guaranteed to be meaningful. To formalize this, the following equivalence relation in $\mathcal{Z}_{sys}$ is defined

$$z \sim z' \iff z \text{ and } z' \text{ are path-connected in } \mathcal{Z}_{sys}, \tag{5.2}$$

which induces a partition of the space $\mathcal{Z}_{sys}$ into m equivalence classes $[z_1], \ldots, [z_m]$. Each equivalence class $[z_i]$ represents a path-connected component of $\mathcal{Z}_{sys}$

$$\mathcal{Z}^i_{sys} = \bigcup_{z \in [z_i]} N_{\varepsilon_z}(z) \subset \mathcal{Z}_{sys} \tag{5.3}$$

called *covered region.* To connect the covered regions, the following set of transitions between them is defined:

Definition 5.1. *A* transition function $f_z^{i,j} : \mathcal{Z}^i_{sys} \times \mathcal{U} \to \mathcal{Z}^j_{sys}$ *maps any point* $z \in \mathcal{Z}^i_{sys}$ *to an equivalence class representative* $z^j_{sys} \in \mathcal{Z}^j_{sys}$*, where* $i, j \in \{1, 2, ..., m\}$ *and* $i \neq j$*.*

Equivalence relation (5.2) and Assumption 5.1 imply that two distinct observations I_1 and I_2 which are mapped into the same covered region $\mathcal{Z}^i_{sys}$ contain the same underlying state of the system, and can be represented by the same equivalence class representative z^i_{sys}. Given a set of covered regions $\mathcal{Z}^i_{sys}$ in $\mathcal{Z}_{sys}$ and a set of transition functions connecting them, the global transitions of $\mathcal{Z}_{sys}$ can be approximated as shown in Figure 5.3 on the left. To this end, a Latent Space Roadmap is defined (see Figure 5.3 in the middle):

Definition 5.2. *A* Latent Space Roadmap *is a directed graph* LSR $= (\mathcal{V}_{\text{LSR}}, \mathcal{E}_{\text{LSR}})$ *where each vertex* $v_i \in \mathcal{V}_{\text{LSR}} \subset \mathcal{Z}_{sys}$ *for* $i \in \{1, 2, ..., m\}$ *is an equivalence class representative of the covered region* $\mathcal{Z}^i_{sys} \subset \mathcal{Z}_{sys}$*, and an edge* $e_{i,j} = (v_i, v_j) \in \mathcal{E}_{\text{LSR}}$ *represents a transition function* $f_z^{i,j}$ *between the corresponding covered regions* $\mathcal{Z}^i_{sys}$ *and* $\mathcal{Z}^j_{sys}$ *for* $i \neq j$*. Moreover, weakly connected components of an* LSR *are called* graph-connected components.

In the following, the structure of the training dataset as well as an overview of the approach to solve Problem 5.1 are presented.

5.2.1 Training dataset and system overview

The proposed framework considers a training dataset $\mathcal{T}_I$ consisting of tuples of the form (I_1, I_2, ρ) where $I_1 \subset \mathcal{I}$ is an image representing a system state, $I_2 \subset \mathcal{I}$ is an image of a successor state, and ρ is a variable representing the action that took place between the two observations. Here, an action is considered to be a *single* transformation that produces any consecutive state represented in I_2 different from the start state in I_1, i.e., ρ cannot be a composition of several transformations. On the contrary, no action is performed if images I_1 and I_2 are observations of the same state, i.e., if $\xi(I_1) \sim \xi(I_2)$ with respect to the equivalence relation (5.2). The variable $\rho = (a, u)$ consists of a binary variable $a \in \{0, 1\}$ indicating whether or not an action occurred as well as a variable u containing the task-dependent action-specific information which can be used to infer the transition functions $f_z^{i,j}$. A tuple $(I_1, I_2, \rho = (1, u))$ is called *action pair* and a tuple $(I_1, I_2, \rho = (0, u))$ is called *no-action pair.* For instance, Figure 5.4 shows an example of an action pair (top row) and a no-action pair (bottom row) for the folding task. When the specifics of an action u are not needed, they are omitted from the tuple notation and simply write (I_1, I_2, a). By abuse of notation, an observation I

contained in any of the training tuples is sometimes referred to as $I \in \mathcal{T}_I$. Finally, the encoded training dataset is denoted by $\mathcal{T}_z$ and consists of latent tuples (z_1, z_2, ρ) obtained from the input tuples $(I_1, I_2, \rho) \in \mathcal{T}_I$ by encoding the inputs I_1 and I_2 into the latent space $\mathcal{Z}_{sys}$ with the latent mapping ξ. The obtained states $z_1, z_2 \in \mathcal{Z}_{sys}$ are called *covered states*. Note that the dataset $\mathcal{T}_I$ is not required to contain all possible action pairs of the system but only a subset of them that sufficiently cover the dynamics, making the proposed approach data efficient as shown in Section 5.5.

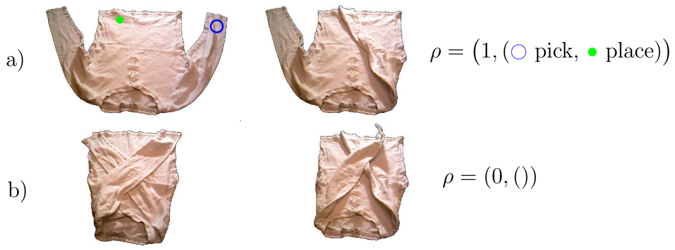


Figure 5.4: Example of action (a) and no-action (b) pairs for the folding task.

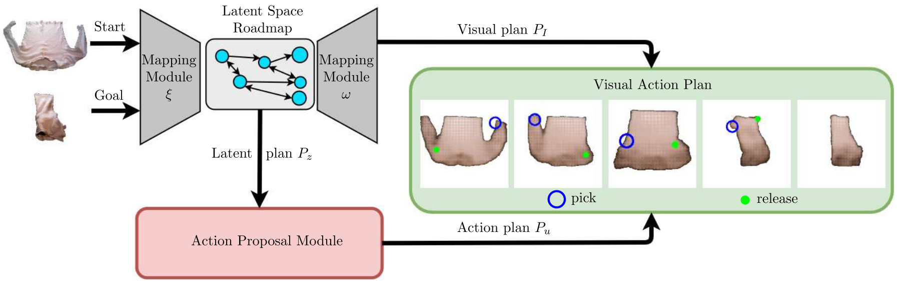


Figure 5.5: Overview of the method. Start and goal images (on the left) are mapped to the latent space $\mathcal{Z}$ by the latent mapping ξ. A latent plan is then found with the Latent Space Roadmap (cyan circles and arrows) and is decoded to a visual plan using the observation generator ω. The Action Proposal Module (red) generates the respective action plan. The final result is a *visual action plan* (green) from start to goal state.

The proposed framework consists of three components, shown in Figure 5.5. First, a *Mapping Module (MM)* is used to both extract a low-dimensional representation of a state contained in a given observation, through the mapping $\xi : \mathcal{I} \to \mathcal{Z}$, and to generate an observation from a given latent state, through the mapping $\omega : \mathcal{Z} \to \mathcal{I}$. Note that ideally the mapping ξ should perfectly extract the underlying state of the system, i.e., different observations containing the same state should be mapped into exact same latent point. However, ξ is generally only an *approximation* of the unknown true latent embedding, implying that different observations containing the same state could be mapped to different latent points. Therefore, a *Latent Space Roadmap (LSR)* is built in the low dimensional latent space to perform planning. In particular, it is a graph-based structure identifying the latent points belonging to the same underlying state and approximating the system dynamics. This enables finding the latent plans P_z between the extracted states $z_{start} = \xi(I_{start})$ and $z_{goal} = \xi(I_{goal})$. For the sake of interpretability, latent plans P_z are decoded into visual plans P_I by the observation generator ω. The visual plan is finally complemented with the action plan produced by the *Action Proposal Module (APM)*. It takes a pair (z_i, z_{i+1}) of consecutive states from the latent plan P_z found by the LSR as inputs, and proposes an action u_i to achieve the desired transition $f_z^{i,i+1}(z_i, u_i) = z_{i+1}$. Applying APM to all consecutive pairs of states in P_z results in an action plan P_u.

Remark 5.1. *The proposed method is able to generate a sequence of actions $\{u_0, \ldots, u_{N-1}\}$ to reach a goal state in I_N from a given start state represented by I_0, even though the tuples in the input dataset $\mathcal{T}_I$ only contain single actions u.*

5.3 Mapping module

The mappings $\xi : \mathcal{I} \to \mathcal{Z}$ and $\omega : \mathcal{Z} \to \mathcal{I}$ as well as the low-dimensional space $\mathcal{Z}$ can be realized using any encoder-decoder based algorithms, for example VAEs, AEs or GANs combined with an encoder network. The primary goal of MM is to find the best possible approximation ξ such that the structure of the extracted states in the latent space $\mathcal{Z}$ resembles the one corresponding to the unknown underlying system. The secondary goal of MM is to learn an observation generator ω which enables visual interpretability of the latent plans. The action information in the binary variable a contained in the training tuples (I_1, I_2, a) is leveraged to improve the quality of the latent space structure. To this aim, a contrastive loss term [162], called *action* term, is introduced which can be easily added to the loss function of any algorithm used to model the MM:

$$\mathcal{L}_{action}(I_1, I_2) = \begin{cases} \max(0, d_m - ||z_1 - z_2||_p) & \text{if } a = 1 \\ ||z_1 - z_2||_p & \text{if } a = 0 \end{cases} \tag{5.4}$$

where $z_1, z_2 \subset \mathcal{Z}_{sys}$ are the latent encodings of the input observations $I_1, I_2 \subset \mathcal{I}_I$, respectively, d_m is a hyperparameter, and the subscript $p \in \{1, 2, \infty\}$ denotes the metric L_p. The action term $\mathcal{L}_{action}$ naturally imposes the formulation of the covered regions $\mathcal{Z}_{sys}^i$ in the latent space: it encodes identical states contained in the no-action pairs close by and it encourages different states to be encoded in separate parts of the latent space via the hyperparameter d_m. As experimentally shown in Section 5.5.1, the choice of d_m has a substantial impact on the latent space structure. In this framework, it is *dynamically* increased during the training of the MM until the separation of action and no-action pairs is achieved. In detail, starting from 0 at the beginning of the training, d_m is increased by Δd_m every k epochs as long as the maximum distance between no-action pairs is larger then the minimum distance between action pairs. The effect of dynamically increasing d_m is shown in Figure 5.6 where it is visualized the distance $||z_1 - z_2||_1$ between the latent encodings of every action training pair (in blue) and no-action training pair (in green) obtained at various epochs during training on a box stacking task. It can be clearly seen that the parameter d_m is increased as long as there is an intersection between action and no-action pairs.

A VAE [147, 148] is used such that its latent space represents the space $\mathcal{Z}$, while the encoder and decoder networks realize the mappings ξ and ω, respectively. Let $I \subset \mathcal{I}_I$ be an input image, and let z denote the unobserved latent variable with prior distribution $p(z)$. The VAE model consists of encoder and decoder neural networks that are jointly optimized to represent the parameters of the approximate posterior distribution $q(z|I)$ and the likelihood function $p(I|z)$, respectively. In particular, VAE is trained to minimize

$$\mathcal{L}_{vae}(I) = E_{z\sim q(z|I)}[\log p(I|z)] + \beta \cdot D_{KL}\left(q(z|I)||p(z)\right) \tag{5.5}$$

with respect to the parameters of the encoder and decoder neural networks. The first term influences the quality of the reconstructed samples, while the second term, called KL divergence term, regulates the structure of the latent space. The trade-off between better reconstructions or a more structured latent space is controlled by the parameter β [163, 164]. The action term (5.4) is easily added to the VAE training objective (5.5) as follows:

$$\mathcal{L}(I_1, I_2) = \frac{1}{2}(\mathcal{L}_{vae}(I_1) + \mathcal{L}_{vae}(I_2)) + \gamma \cdot \mathcal{L}_{action}(I_1, I_2) \tag{5.6}$$

where $I_1, I_2 \subset \mathcal{T}_I$ and the parameter γ controls the influence of the distances among the latent encodings on the structure of the latent space. Note that the same procedure applies for integrating the action term into any other framework for modeling the MM.

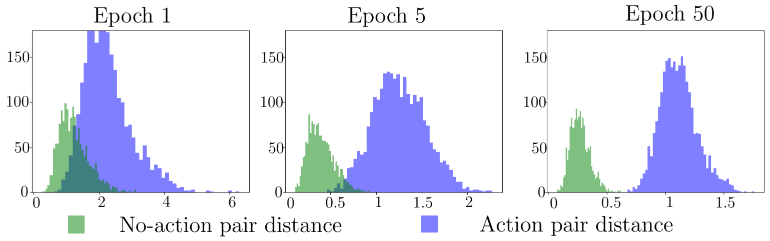


Figure 5.6: An example showing histograms of distances $||z_1 - z_2||_1$ for the latent action (in blue) and no-action pairs (in green) obtained at epochs 1, 5 and 50 during the training of VAE on the hard box stacking task (more details in Section 5.5). The figure shows the separation of the action and no-action distances induced by dynamically increasing the minimum distance d_m in the action term (5.4).

5.4 Latent space roadmap

The LSR, defined in Definition 5.2, is a graph built in the latent space $\mathcal{Z}$ obtained from the MM. The basic idea is that each node in the roadmap is associated with a covered region $\mathcal{Z}_{sys}^{i}$ and two nodes are connected by an edge if there exists an action pair $(I_1, I_2, \rho = (1, u_1))$ in the training dataset $\mathcal{T}_I$ such that the transition $f_z^{1,2}(z_1, u_1) = z_2$ is achieved in $\mathcal{Z}_{sys}$.

The LSR building procedure is summarized in Algorithm 2 and discussed in the following. It relies on a clustering algorithm that builds the LSR using the encoded training data $\mathcal{T}_z$ and a specified metric L_p as inputs. The additional input parameter τ is inherited from the clustering algorithm and is then automatically determined, as explained in the following.

LSR building

Algorithm 2 consists of three phases. In Phase 1 (lines $1.1-1.5$), a *reference* graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, induced by $\mathcal{T}_z$, is built. It is visualized in the left part of Figure 5.7. Its set of vertices $\mathcal{V}$ is the set of all the latent states in $\mathcal{T}_z$, while edges exists only among the latent action pairs. It serves as a look-up graph to preserve the edges that later induce the transition functions $f_z^{i,j}$.

In Phase 2, the covered regions $\mathcal{Z}_{sys}^{i} \subset \mathcal{Z}_{sys}$ are identified. This is achieved by first clustering the training samples and then retrieving the covered regions from these clusters. To this aim, agglomerative clustering [165] on the encoded dataset $\mathcal{T}_z$ (line 2.1) is performed. Agglomerative clustering is a hierarchical clustering scheme that starts from single nodes of the dataset and merges the closest nodes, according to a dissimilarity measure, step-by-step until only one node remains. It results in a *stepwise dendrogram* M, depicted in the middle part of Figure 5.7, which is a tree structure visualizing the arrangement of data points in clusters with respect to the level of dissimilarity between them. The *unweighted average* distance between points in each cluster, a method also referred to as UPGMA [166], is used to measure inter-cluster dissimilarity.

Algorithm 2 LSR building

```
Require: Dataset $\mathcal{T}_z$, metric $L_p$, clustering threshold $\tau$
Phase 1
init graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}) := (\{\}, \{\})$
for each $(z_1, z_2, a) \in \mathcal{T}_z$ do
    $\mathcal{V} \leftarrow$ create nodes $z_1, z_2$
    if $a = 1$ then
        $\mathcal{E} \leftarrow$ create edge $(z_1, z_2)$
    end if
end for
Phase 2
$M \leftarrow$ Average-Agglomerative-Clustering($\mathcal{T}_z, L_p$) [165]
$\mathcal{W} \leftarrow$ get-Disjoint–Clusters($M, \tau$)
$\mathcal{Z}_{sys} \leftarrow \{\}$
for each $\mathcal{W}^i \in \mathcal{W}$ do
    $\varepsilon^i \leftarrow$ get-Cluster-Epsilon($\mathcal{W}^i$)
    $\mathcal{Z}^i_{sys} := \cup_{w \in \mathcal{W}^i} N_{\varepsilon^i}(w)$
    $\mathcal{Z}_{sys} := \mathcal{Z}_{sys} \cup \{\mathcal{Z}^i_{sys}\}$
end for
Phase 3
init graph LSR $= (\mathcal{V}_{\text{LSR}}, \mathcal{E}_{\text{LSR}}) := (\{\}, \{\})$
for each $\mathcal{Z}^i_{sys} \in \mathcal{Z}_{sys}$ do
    $w^i := \frac{1}{|\mathcal{W}^i|} \sum_{w \in \mathcal{W}^i} w$
    $z^i_{sys} := \text{argmin}_{z \in \mathcal{Z}^i_{sys}} ||z - w^i||_p$
    $\mathcal{V}_{\text{LSR}} \leftarrow$ create node $z^i_{sys}$
end for
for each edge $e = (v_1, v_2) \in \mathcal{E}$ do
    find $\mathcal{Z}^i_{sys}, \mathcal{Z}^j_{sys}$ containing $v_1, v_2$, respectively
    $\mathcal{E}_{\text{LSR}} \leftarrow$ create edge $(z^i_{sys}, z^j_{sys})$
end for
return LSR
```

Next, the dissimilarity value τ, referred to as *clustering threshold*, induces the set of disjoint clusters $\mathcal{W}$, also called *flat* or *partitional* clusters [167, 168], from the stepwise dendrogram M [165] (line 2.2). Points in each cluster $\mathcal{W}^i$ are then assigned a uniform ϵ^i (line 2.5), i.e., the neighborhood size from Assumption 5.1 of each point $z \in \mathcal{W}^i$ is $\epsilon_z = \epsilon^i$. The definition of the ϵ^i value is discussed at the end of this phase. The union of the ϵ^i-neighborhoods of the points in $\mathcal{W}^i$ then forms the covered region $\mathcal{Z}^i_{sys}$ (line 2.6). Illustrative examples of covered regions obtained from different values of τ are visualized on the right side of Figure 5.7 using various colors. The optimization of τ is discussed in the rest of the section. The result of this phase is the set of the identified covered regions $\mathcal{Z}_{sys} = \{\mathcal{Z}^i_{sys}\}$ (line 2.7).
The value ϵ^i is approximated as follows

$$\epsilon^i = \mu^i + \sigma^i \tag{5.7}$$

where μ^i and σ^i are the mean and the standard deviation of the distances $\|z^i_j - z^i_k\|_p$ among all the training pairs $(z^i_j, z^i_k) \in \mathcal{T}_z$ belonging to the i th cluster. The approximation in (5.7) allows to take into account the cluster density such that denser clusters get lower ϵ^i. The approximation (5.7) is validated in Sections 5.5 and 5.6 where the covered regions identified by the LSR are analyzed.

In Phase 3, the LSR $= (\mathcal{V}_{\text{LSR}}, \mathcal{E}_{\text{LSR}})$ is built. First, the mean value w^i of all the points in each cluster $\mathcal{W}^i$ is computed (line 3.3). As the mean itself might not be contained in the corresponding path-connected component, the equivalence class representative $z^i_{sys} \in \mathcal{Z}^i_{sys}$ that is the closest is found (line 3.4). The found representative then defines a node $v_i \in \mathcal{V}_{\text{LSR}}$ representing the covered region $\mathcal{Z}^i_{sys}$ (line 3.5). Lastly, the set of edges $\mathcal{E}$ in the reference graph built in Phase 1 is used to infer the transitions $f^{i,j}_z$ between the covered regions identified in Phase 2. An edge is created in the LSR if there exists an edge in $\mathcal{E}$ between two vertices in $\mathcal{V}$ that were allocated to different covered regions (lines $3.6 - 3.8$). The right side of Figure 5.7 shows the final LSRs, obtained with different values of the clustering threshold τ.

Note that, as in the case of the VAE (Section 5.3), no action-specific information u is used in Algorithm 2 but solely the binary variable a indicating the occurrence of an action.

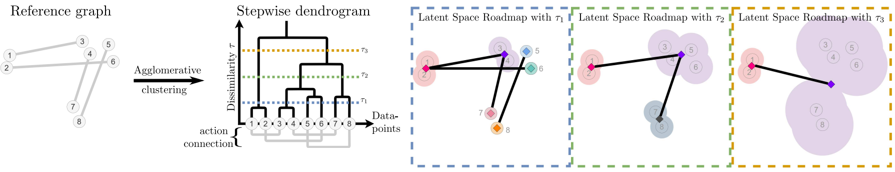


Figure 5.7: Illustrative example visualizing the LSR building steps and the effect of the clustering threshold τ. On the left is the reference graph built in Phase 1 of Algorithm 2. Middle part visualizes a dendrogram M obtained from the clustering algorithm in Phase 2. On the right, three examples of LSRs are shown together with the covered regions (marked with various colors) corresponding to different thresholds τ (with $\tau_1 < \tau_2 < \tau_3$) chosen from M.

Optimization of LSR clustering threshold τ

As illustrated in Figure 5.7, the number of vertices and edges in LSR$_{\tau_i}$ changes with the choice of the clustering threshold τ_i in Phase 2 of Algorithm 2. Moreover, the resulting LSRs can have different number of *graph*-connected components. For example, LSR$_{\tau_1}$ in Figure 5.7 has 2 graph-connected components, while LSR$_{\tau_2}$ and LSR$_{\tau_3}$ have only a single one. Ideally, one wants to obtain a graph that exhibits both good connectivity which best approximates the true underlying dynamics of the system, and has a limited number of graph-connected component. Intuitively, high number of edges increases the possibility to find latent paths from start to goal state. At the same time, this possibility is decreased when the graph is fragmented into several isolated components, which is why the maximum number of graph-connected components is also limited.

While the clusters themselves cannot be analyzed, the information captured by the LSR that correlates with the performance of the task can be evaluated, i.e., a graph can be assessed by the number of edges and graph-connected components it exhibits as discussed above. This induces an objective which is used to optimize the value of the clustering threshold τ. This is formulated as

$$\psi(\tau, c_{\max}) = \begin{cases} |\mathcal{E}_{\mathrm{LSR}_\tau}| & \text{if } c_{\mathrm{LSR}_\tau} \leq c_{\max}, \\ -\infty & \text{otherwise,} \end{cases} \tag{5.8}$$

where $|\mathcal{E}_{\mathrm{LSR}_\tau}|$ is the cardinality of the set $\mathcal{E}_{\mathrm{LSR}_\tau}$, c_{LSR_τ} represents the number of graph-connected components of the graph LSR_τ induced by τ, and the hyperparameter $c_{\max}$ represents the upper bound on the number of graph-connected components. The optimal τ in a given interval $[\tau_{\min}, \tau_{\max}]$ can be found by any scalar optimization method. Brent's optimization method [169] is used which maximizes the objective (5.8)

$$\max_{\tau_{\min} \leq \tau \leq \tau_{\max}} \psi(\tau, c_{\max}). \tag{5.9}$$

This optimization procedure is summarized in Algorithm 3. It takes as an input the encoded training data $\mathcal{T}_z$, the metric L_p, the search interval where the clustering parameter τ is to be optimized, and the upper bound $c_{\max}$ to compute the optimization objective in (5.8). After initialization of the parameter τ (line 1), for example, by considering the average value of its range, the Brent's optimization loop is performed (lines 2-5). Firstly, the LSR with the current τ is built according to Algorithm 2 (line 3). Secondly, the optimization objective (5.8) is computed on the obtained LSR_τ (line 4). Thirdly, the parameter τ as well as the bounds $\tau_{\min}$ and $\tau_{\max}$ are updated according to [169] (line 5). The optimization loop is performed until the convergence is reached, i.e., until $|\tau_{\max} - \tau_{\min}|$ is small enough according to [169]. Lastly, the optimal τ^* (line 6) is selected for the final LSR_{τ^*}.

Note that even though Algorithm 3 still needs the selection of the hyperparameter $c_{\max}$, it is shown in Section 5.5.2 that the parameter is rather robust to the choice of this parameter.

Visual plan generation

In order to find a plan from start to goal observation, these two are encoded by ξ into the VAE's latent space $\mathcal{Z}$ and their closest

Algorithm 3 LSR input optimization

```
Require: Dataset $\mathcal{T}_z$, metric $L_p$, search interval $[\tau_{\min}, \tau_{\max}]$, $c_{\max}$
$\tau \leftarrow$ init($\tau_{\min}$, $\tau_{\max}$)
while $|\tau_{\max} - \tau_{\min}|$ not small enough do
    LSR$_\tau \leftarrow$ LSR-building($\mathcal{T}_z, L_p, \tau$)    [Algorithm 2]
    $\psi \leftarrow$ Evaluate(LSR$_\tau$)    [Eq. (5.8)]
    $\tau, \tau_{\min}, \tau_{\max} \leftarrow$ Brent-update($\psi$)    [169]
end while
$\tau^* \leftarrow \tau$
return LSR$_{\tau^*}$
```

nodes in the LSR are found. Next, all shortest paths [170] in the LSR between the identified nodes are retrieved. Finally, the equivalence class representatives of the nodes comprising each of the found shortest path compose the respective latent plan P_z, which is then decoded into the visual plan P_I using ω.

Action proposal module

The final component of the framework is the Action Proposal Module (APM) which is used to complement a latent plan, produced by the LSR, with an action plan that can be executed by a suitable framework. The action plan P_u corresponding to a latent plan P_z produced by the LSR is generated sequentially: given two distinct consecutive latent states (z_i, z_{i+1}) from P_z, APM predicts an action u_i that achieves the transition $f^{i,i+1}(z_i, u_i) = z_{i+1}$. Such functionality can be realized by any method that is suitable to model the action specifics of the task at hand.

The action specifics are modeled with a neural network called Action Proposal Network (APN). A diamond-shaped multi layer perceptron is designed to the purpose and trained in a supervised fashion on the latent *action* pairs. In detail, the training dataset $\overline{\mathcal{T}_z}$ for the APN is derived from $\mathcal{T}_I$ but preprocessed with the VAE encoder (representing the latent mapping ξ). Each training *action* pair $(I_1, I_2, \rho = (1, u)) \in \mathcal{T}_I$ is encoded into $\mathcal{Z}$ and the parameters μ_i, σ_i of the approximate posterior distributions $q(z|I_i) = N(\mu_i, \sigma_i)$ are obtained, for $i = 1, 2$. Novel points $z_1^s \sim q(z|I_1)$ and $z_2^s \sim q(z|I_2)$ are then sampled for $s \in \{0, 1, \ldots, S\}$. This results in $S + 1$ tuples (μ_1, μ_2, ρ) and $(z_1^s, z_2^s, \rho), 0 \leq s \leq S$, where $\rho = (1, u)$ was omitted from the notation for simplicity. The set of all such low-dimensional tuples then forms the APN training dataset $\overline{\mathcal{T}_z}$. It is worth remarking the two-fold benefit of this preprocessing step: not only does it reduce the dimensionality of the APN training data but also enables enlarging it with novel points by factor $S + 1$. Note that the latter procedure is not possible with non-probabilistic realizations of ξ.

5.5 Box stacking simulations

The proposed method is validated on two different versions of a simulated box stacking task shown in Figure 5.8. The setup in the top row is referred to as *normal stacking* task, denoted by ns, while the one in the bottom row is referred to as *hard stacking* task, denoted by hs. In particular, in the latter the task of retrieving the underlying state of the system had been made harder with respect to the former. This was achieved by using more similar box textures, which makes it harder to separate the underlying states, as well as different lighting conditions, which make observations containing the same underlying states look more dissimilar.

Both setups were developed with the Unity engine [171] and are composed of four boxes with different textures that can be stacked in a 3×3 grid (dotted lines in Figure 5.8). A grid cell can be occupied by only one box at a time which can be moved according to the *stacking rules*: i) it can be picked only if there is no other box on top of it, and ii) it can be released only on the ground or on top of another box inside the 3×3 grid. The action-specific information u, shown in Figure 5.8, is a pair $u = (p, r)$ of pick p and release r coordinates in the grid modeled by the row and column indices, i.e., $p = (p_r, p_c)$ with $p_r, p_c \in \{0, 1, 2\}$, and equivalently for

$r = (r_r, r_c)$. In both versions of the dataset, the position of each box in a grid cell was generated by introducing $\sim 17\%$ noise along x and y axes, which was applied when generating both action and no-action pairs. The resulting images have dimension $256 \times 256 \times 3$.

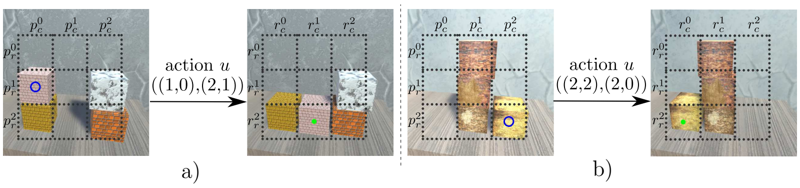


Figure 5.8: Example of actions $u = (p, r)$ in the normal (a - left) and hard (b - right) box stacking tasks. The blue circle shows the picking location p, and the green one the release position r.

The designed tasks contain exactly 288 different grid configurations, i.e., the specification of which box, if any, is contained in each cell. These 288 grid configurations represent the underlying states in this task. Note that the exact number of underlying states is in general not known.
Given a pair of states and the ground truth stacking rules, it is possible to analytically determine whether or not an action is allowed between them. In addition, the grid configuration associated with an image (i.e., its underlying state) in the produced visual plan P_I can be determined using classifiers[1]. These were trained on the decoded images and achieved accuracy greater than 99.5% on a holdout dataset composed of 750 samples for both versions of the stacking task. This allows to automatically evaluate the structure of the latent space $\mathcal{Z}_{sys}$, the quality of visual plans P_I generated by the LSR and MM, and the quality of action plans P_u predicted by the APN. In particular, the simulations aim to investigate:

1. **MM** the impact of the action term (5.4) in the augmented loss function (5.6) on the structure of the latent space and the influence of the respective parameters (latent space dimension, minimum distance, etc.) on the LSR performance

[1] `https://github.com/visual-action-planning/lsr-code`

2. **LSR** the performance of the LSR and the influence of the action term (5.4) and the maximum number of components on it as well as the quality of the covered regions approximation provided by the LSR.

3. **APM** the performance of the APN model.

Implementation details

For VAEs, each model is annotated by VAE$_{ld}$-*task*-*d* where *ld* denotes the dimension of the latent space, *task* denotes the version of the task and is either *ns* or *hs* for the normal and hard stacking tasks, respectively, and *d* indicates whether or not the model was trained with the action loss term (5.4). Here, it is used $d = b$ to denote a *baseline* VAE trained with the original VAE objective (5.5), and $d = L_p$ to denote an *action* VAE trained with the loss function (5.6) including the action term (5.4) using metric L_p for $p \in \{1, 2, \infty\}$. All VAE models use a ResNet architecture [172] for the encoder and decoder networks. They are trained for 500 epochs on a training dataset $\mathcal{T}_I$, composed of 2500 tuples (65% action pairs and 35% no-action pairs). Note that the dataset size is much smaller compared to the number of all possible combinations of system states, given by $(n+1)n/2$ for a system with n different states. This results in 41616 possible combinations as $n = 288$ in the stacking task, while only 2500/41616 pairs are considered, which corresponds to 6% of all possible state pairs. For each combination of parameters *ld*, *task*, and *d*, five VAEs were trained by initializing different random seeds. Same seeds were also used to create training and validations splits of the training dataset. The weight β from (5.5) and (5.6) was gradually increased from 0 to 2 over 400 epochs following a scheduling procedure[1], while γ was fixed to 100. In this way, the models were encouraged to first learn to reconstruct the input images and then to gradually structure the latent space. The minimum distance d_m was dynamically increased every fifth epoch starting from 0 using $\Delta d_m = 0.1$ as described in Section 5.3.

For LSR, the notation LSR-L_p is used to denote a graph built using the metric L_p in Algorithm 2. The parameters $\tau_{\min}$ and $\tau_{\max}$ in the LSR optimization (5.9) were set to 0 and 3, respectively. Unless otherwise specified, $ld = 12$ and used L_1 metric are fixed for both tasks. Moreover, the number of graph-components $c_{\max}$ in the optimization of the clustering threshold (5.8) was set to 1 for ns and 20 for hs. These choices are explained in detail in the following sections. Given an LSR, its performance is evaluated by measuring the quality of the visual plans found between 1000 randomly selected start and goal observations from an unseen test dataset containing 2500 images. In the evaluation of the planning performance, the following quantities are considered: *i)* percentage of cases when all shortest paths from start to goal observations are correct, denoted as *% All*, *ii)* percentage of cases when at least one of the proposed paths is correct, denoted as *% Any*, and *iii)* percentage of correct single transitions in the paths, denoted as *% Trans*. The % Any score in *ii)* is referred to as *partial scoring*, while the combination of scores *i)-iii)* as *full scoring*. Moreover, mean and standard deviation values are reported over the 5 different random seeds used to train the VAEs.

For APNs, the notation APN$_{ld}$-*task-d*, analogous to the VAEs, is used. The APN models are trained for 500 epochs on the training dataset $\overline{\mathcal{T}_z}$ obtained following the procedure described in Sec 5.4 using $S = 1$. The validation split, corresponding to 15% of $\overline{\mathcal{T}_z}$, is used to extract the best performing models that were used in the evaluation. Similarly as for LSR, mean and standard deviation values are reported. These are obtained considering the 5 different random seeds used in the VAE training.

The complete implementation details for all the models and respective hyperparameters can be found in the configuration files in the code repository[1].

5.5.1 MM analysis

In this section, the positive effect of the action term (5.4) on the structure of the latent space is demonstrated thus answering the questions listed in point 1). In detail, first, the influence of the dynamic minimum distance d_m on the LSR performance is investigated. Then, the structure of the latent space is studied by analyzing the distance between the encodings of the 288 states. Lastly, the influence of the latent dimension on the LSR performance is examined.

Influence of dynamic d_m

A key parameter in the action term (5.4) is the minimum distance d_m encouraged among the action pairs. The approach proposed in Section 5.3 is validated using the hard box stacking task. In detail, the dynamical increase of d_m (see Figure 5.6), which reaches $d_m = 2.3$ at the end of the training, is compared with (i) the choice in [153] which uses d_m fixed to the average action distance among all the training action pairs measured in the corresponding baseline VAE resulting in $d_m = 11.6$, and (ii) the case where d_m is fixed to a constant value $d_m = 100$ significantly higher than the one obtained in (i).
The performance of the LSR, evaluated using full scoring, with respect to the choice of d_m is summarized in Table 5.1. The results show that the dynamic selection of d_m significantly outperforms the other methods for all the scores. This approach not only eliminates the need for training the baseline VAEs as in [153] but also reaches a value of d_m that obtains a better separation of covered regions $\mathcal{Z}^i_{sys}$ without compromising the optimization of the reconstruction and KL terms.

	d_m	% All	% Any	% Trans.
Dynamic (Prop.)	-	**90.9 ± 3.5**	**92.1 ± 2.9**	**95.8 ± 1.3**
Baseline [153]	11.6	48.7 ± 40.7	50.7 ± 41.5	72.8 ± 26.0
High	100	0.0	0.0	14.6 ± 2.0

Table 5.1: Comparison of the LSR performance when using different methods for selecting d_m. Top row is the proposed approach where d_m is determined dynamically, middle row is the approach used in [153] where d_m is calculated from the latent action pairs in the baseline VAE, while in the bottom row d_m is set a high value. Results are obtained using VAE$_{12}$-hs-L_1 and LSR-L_1.

Separation of the states

The effect of the action loss (5.4) on the structure of the latent space is investigated by analyzing the separation of the latent points $z \in \mathcal{T}_z$ corresponding to different underlying states of the system. Recall that images in $\mathcal{T}_I$ containing the same state will look different because of the introduced positioning noise in both tasks, and different lightning conditions in the case of hs.

Let $\bar{z}_s$ be the *centroid* for state s defined as the mean point of the training latent samples $\{z_{s,i}\}_i \subset \mathcal{T}_z$ associated with the state s. Let $d_{intra}(z_{s,i}, \bar{z}_s)$ be the *intra-state* distance defined as the distance between the latent sample i associated with the state s, namely $z_{s,i}$, and the respective centroid $\bar{z}_s$. Similarly, let $d_{inter}(\bar{z}_s, \bar{z}_p)$ denote the *inter-state* distance between the centroids $\bar{z}_s$ and $\bar{z}_p$ of states s and p, respectively. In the following analysis, the models VAE$_{12}$-ns-L_1 and VAE$_{12}$-ns-L_1 are considered for the normal and hard stacking tasks, respectively.

Figure 5.9 reports the mean values (bold points) and the standard deviations (thin lines) of the inter- (in blue) and intra-state (in orange) distances for each state $s \in \{1, ..., 288\}$ in the normal stacking task when using the baseline model VAE$_{12}$-ns-b (top) and the action model VAE$_{12}$-ns-L_1 (bottom). In case of the baseline VAE, similar intra-state and inter-state distances are obtained. This implies that samples of different states are encoded close together in the latent space which can raise ambiguities when planning. On the contrary, when using VAE$_{12}$-ns-L_1, the inter- and intra-state distances approach the values 5 and 0, respectively. These values are imposed with the action term (5.4) as the minimum distance d_m reaches 2.6. Therefore, even when there exists no direct link between two samples of different states, and thus the action term for the pair is never activated, the VAE is able to encode them such that the desired distances in the latent space are respected.

Similar conclusions also hold in the hard stacking task for which the inter- (in blue) and intra-state (in orange) distances are depicted in Figure 5.10. Top row again shows the distances calcu-

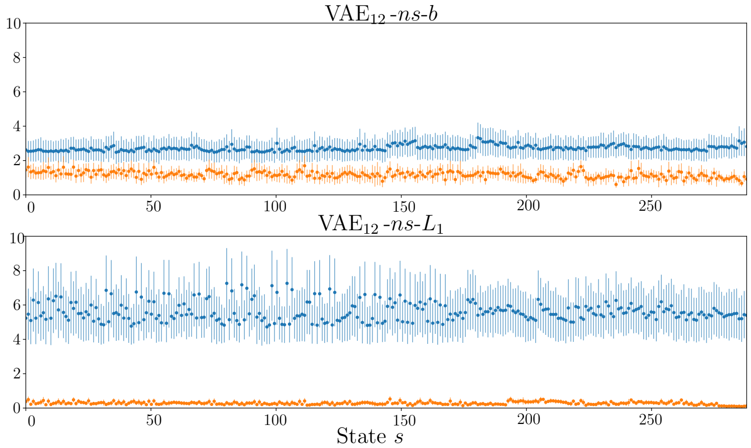


Figure 5.9: Mean values (bold points) and standard deviations (thin lines) of inter- (blue) and intra- (orange) state distances for each state calculated using a VAE trained with (bottom) and without (top) action term. Results are evaluated using VAE$_{12}$-ns-L_1 model on the normal stacking task.

lated with the baseline model VAE$_{12}$-hs-b, while the ones shown in the bottom are obtained with the action model VAE$_{12}$-hs-L_1. Note that lower inter-state distance is obtained than in case of ns because the minimum distance d_m reaches only 2.3 in hs.

Finally, the difference between the minimum inter-state distance and the maximum intra-state distance is analyzed for each state. The higher the value the better separation of states in the latent space since samples of the same state are in this case closer to each other than samples of different states. When the latent states are obtained using the baseline VAE$_{12}$-ns-b, a non-negative distance is obtained for 0/288 states with an average value of ≈ -1.2. This implies that only weak separation occurs in the latent space for samples of different states. On the other hand, when calculated on points encoded with VAE$_{12}$-ns-L_1, the difference becomes non-negative for 284/288 states and its mean value increases to ≈ 0.55, thus achieving almost perfect separation. In the hard stacking task, it is similarly obtained that VAE$_{12}$-hs-b reaches an average difference of -5.86 (being non-negative for 0/288 states), while the action model VAE$_{12}$-hs-L_1 reduces the average difference to -0.04 (being non-negative for 121/288 states). These results val-

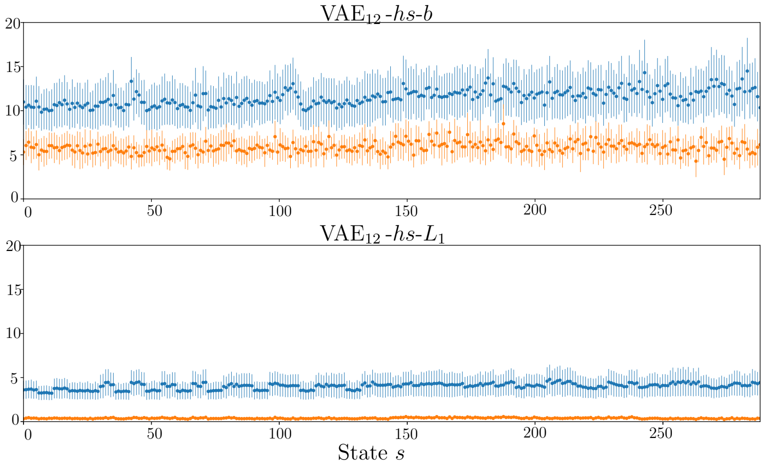


Figure 5.10: Mean values (bold points) and standard deviations (thin lines) of inter- (blue) and intra- (orange) state distances for each state calculated using a VAE trained with (bottom) and without (top) action term. Results are evaluated using VAE$_{12}$-*hs*-L_1 model on the hard stacking task.

idate that the action term (5.4) and the dynamic setting of d_m contribute to a better structured latent space $\mathcal{Z}_{sys}$ as well as confirm the higher level of complexity of the hard task compared to the normal one.

Latent space dimension

The choice of the latent space dimension is not a trivial task, since it should be as low as possible while still encoding all the relevant features. Table 5.2 reports the partial scoring of the LSR on both normal and hard stacking tasks using VAE models with various latent dimensions. The results demonstrate an evident drop in the performance when the latent dimension is too small, such as $ld = 4$. As ld increases, gradual improvements in the performance are observed and a satisfactory level is achieved using $ld \geq 6$ for ns, and $ld \geq 12$ for hs. Therefore, hs requires more dimensions in order to capture all the relevant and necessary features. This result not only demonstrates the complexity of each task version but also justifies the choice $ld = 12$ in the rest of experiments.

ld	ns [%]	hs [%]
4	7.9 ± 2.2	8.8 ± 7.9
6	99.96 ± 0.08	56.2 ± 23.1
8	99.96 ± 0.08	62.7 ± 18.7
12	100.0 ± 0.0	92.1 ± 2.9
16	100.0 ± 0.0	95.9 ± 1.4
32	97.5 ± 4.33	96.4 ± 0.4
64	99.8 ± 0.5	95.3 ± 0.9

Table 5.2: Comparison of the LSR performance when using different latent dimensions for the normal (left) and hard (right) box stacking tasks.

5.5.2 LSR analysis

In this section, the LSR performance is analyzed in dependence on whether or not the action term is included in the VAE loss function. Secondly, the influence of the upper bound on the number of connected components $c_{\max}$ (used in (5.8)) is investigated. Finally, the covered regions determined by the LSR are studied.

LSR performance

The baseline models VAE$_{12}$-b and the action VAE$_{12}$-L_1, trained with the action term (5.4) using metric L_1, are considered. Table 5.3 reports the LSR full scoring on the normal (top part) and hard (bottom part) box stacking tasks. The results show deteriorated LSR performance when using baselines VAE$_{12}$-b compared to the action VAEs regardless the version of the task. This indicates that VAEs-b are not able to separate states in $\mathcal{Z}_{sys}$. It can be again concluded that the action term (5.4) needs to be included in the VAE loss function (5.6) in order to obtain distinct covered regions $\mathcal{Z}^i_{sys}$. In addition, the results confirm the different level of difficulty of the two tasks as indicated by the drop in the LSR performance on hs compared to ns using the action VAE-L_1. In summary, this experiment validates the effectiveness of the LSR for performing normal and hard box stacking as well as confirms the need to integrate the action term in the VAE loss function.

Task	Model	% All	% Any	% Trans.
ns	VAE-b+ LSR-L_1	2.5 ± 0.5	4.1 ± 1.0	59.7 ± 4.9
	VAE-L_1+ LSR-L_1	**100.0 ± 0**	**100.0 ± 0**	**100.0 ± 0**
hs	VAE-b+ LSR-L_1	0.2 ± 0.1	0.2 ± 0.1	38.0 ± 2.0
	VAE-L_1+ LSR-L_1	**90.9 ± 3.5**	**92.1 ± 2.9**	**95.8 ± 1.3**

Table 5.3: The LSR performance for the normal (top part) and hard (bottom part) box stacking tasks using the action VAE-L_1 trained with the loss function (5.6) including the action loss (5.4), and baseline VAE-b train with the original objective given in 5.5 (best results in bold).

Influence of the maximum number of connected components

The optimization method described in Algorithm. 3 requires setting the upper bound on the number of graph-connected components c_{max} of the LSR. Table 5.4 shows how different upper bounds influence the LSR performance on both normal and hard stacking tasks. The LSR is evaluated on partial scoring using VAE$_{12}$ -ns-L_1 and VAE$_{12}$ -hs-L_1.

c_{max}	ns [%]	hs [%]
1	**100.0 ± 0.0**	65.3 ± 24.6
5	99.5 ± 0.4	88.6 ± 5.4
10	99.0 ± 0.3	91.5 ± 3.8
20	97.5 ± 0.5	**92.1 ± 2.9**
50	91.3 ± 1.1	88.2 ± 2.0
100	80.0 ± 1.4	77.9 ± 2.1

Table 5.4: Performance of the LSR on normal ns and hard stacking task hs for different c_{max}. Best results shown in bold.

The results are shown to be rather robust with respect to the c_{max} value. For both task versions, the performance drops for a very high c_{max}, such as $c_{\text{max}} = 100$, while in the hard stacking task an additional drop for a very low c_{max}, such as $c_{\text{max}} = 1$, is observed. This behavior can be explained by the fact that the lower the c_{max} the more the system is sensitive to outliers, while the higher the c_{max} the greater the possibility that the obtained graph is disconnected which potentially compromises its planning capabilities. For example, in the hard stacking task, outliers arise from different lightning conditions. In contrast, no outliers exists in the normal stacking task, which is why a single connected component is sufficient for the LSR to perform perfectly. For all further evaluation, $c_{max} = 1$ and $c_{max} = 20$ are set for the normal and hard box stacking tasks, respectively. This experiment validates the robustness of the approach with respect to the parameter c_{max} as well as justifies the choices $c_{max} = 1$ and $c_{max} = 20$ for ns and hs, respectively, in the rest of experiments.

Covered regions using LSR

In order to show that the LSR captures the structure of the system, it is used to determine if various novel observations are recognized as covered or not. First, it is checked if observations corresponding to true underlying states of the system, that have not been seen during training, are properly recognized as covered. Then, it is checked if observations taken from standard datasets, such as CIFAR10 [173] and 3D Shapes [174], are marked as uncovered since they correspond to out-of-distribution observations. The covered regions $\mathcal{Z}^i_{sys}$ are computed using the epsilon approximation in (5.7).

Table 5.5 reports the results of the classification of covered states for both normal (top block) and hard (bottom block) box stacking tasks produced using VAE$_{12}$-ns-L_1 and VAE$_{12}$-hs-L_1, respectively. In detail, 5000 novel observations are considered that correspond to covered states of the normal and hard box stacking tasks, and the percentage of those classified as covered is reported. The higher this percentage the better the classification. The table shows that the LSR almost perfectly recognizes all the covered states for the normal task with the average recognition equal to 99.5%, while for the hard stacking task, it properly recognizes on average 4694/5000 samples (93.9%). Regarding the out-of-distribution observations, 5000 images are sampled from each of the 3D Shapes and CIFAR-10 datasets, and the percentage of those recognized as covered is evaluated. In this case, the lower the percentage the better the classification. As shown in Table 5.5, almost all the samples in both datasets are correctly recognized as not covered with all the models. In particular, the classification with CIFAR-10 only fails with an average percentage lower than 0.03%, while perfect recognition is obtained with 3D Shapes.

It can be thus concluded that LSR successfully captures the global structure of the system as it correctly classifies the observations representing the possible states of the system as covered, and out-of-distribution observations as not covered.

	Boxes [%]	CIFAR-10 [%]	3D Sh. [%]
ns	99.5 ± 0.3	0.008 ± 0.011	0 ± 0
hs	93.9 ± 0.7	0.028 ± 0.044	0 ± 0

Table 5.5: Classification of covered states for the normal and hard box stacking models when considering as inputs novel boxes images (left column) as well as images from CIFAR-10 and 3D Shapes datasets (middle and right columns).

5.5.3 APM analysis

The accuracy of action predictions obtained by APN-L_1 is evaluated on an unseen test set consisting of 1611 and 1590 action pairs for the normal and hard stacking task, respectively. In particuar, it is assessed in terms of percentage of the correct proposals for picking and releasing, as well as percentage of pairs where both pick and release proposals are correct. For both task versions, all the models achieve 99% or higher accuracy evaluated on 5 different random seeds determining the training and validation sets[1]. This is because the box stacking task results in an 18-class classification problem for action prediction which is simple enough to be learned from any of the VAEs.

5.6 Folding experiments

In this section, the proposed approach is validated on a real world experiment involving manipulation of deformable objects, namely folding of a T-shirt. As opposed to the box stacking task, the true underlying states are in this case unknown and it is therefore not possible to define an automatic verification of the correctness of a given visual action plan. The folding task setup, depicted in Figure 5.11, is composed of a Rethink Robotics Baxter robot equipped with a Primesense RGB-D camera mounted on its torso. The execution videos of all the performed experiments and respective visual action plans can be found on the project website[2]. For this task, a dataset $\mathcal{T}_I$ containing 1283 training tuples was collected. Each tuple consists of two images of size $256 \times 256 \times 3$, and action specific information $u = (p, r, h)$ where $p = (p_r, p_c)$ are the picking coordinates, $r = (r_r, r_c)$ the releasing coordinates and h picking height. An example of an action and a no-action pair is shown in Figure 5.4. The values $p_r, p_c, r_r, r_c \in \{0, \ldots, 255\}$ correspond to image coordinates, while $h \in \{0, 1\}$ is either the height of the table or a value measured from the RGB-D camera to pick up only the top layer of the shirt. Note that the latter is a challenging task in its own [175] and leads to decreased performance when it is necessary to perform it, as shown in the following. The dataset $\mathcal{T}_I$ was collected by a human operator manually selecting pick and release points on images showing a given T-shirt configuration, and recording the corresponding action and following configuration. No-action pairs, representing $\approx 37\%$ of training tuples in $\mathcal{T}_I$, were generated by slightly perturbing the cloth appearance.

Finally, a re-planning step after each action completion is introduced as shown in Figure 5.12. This accounts for potential execution uncertainties, such as inaccuracies in grasping or in the positioning phases of pick-and-place operations, which lead to observations different from the ones planned in P_I. Note that after each action execution, the current observation of the cloth is considered as a new start observation, and a new visual action plan

[2] `https://visual-action-planning.github.io/lsr-v2/`

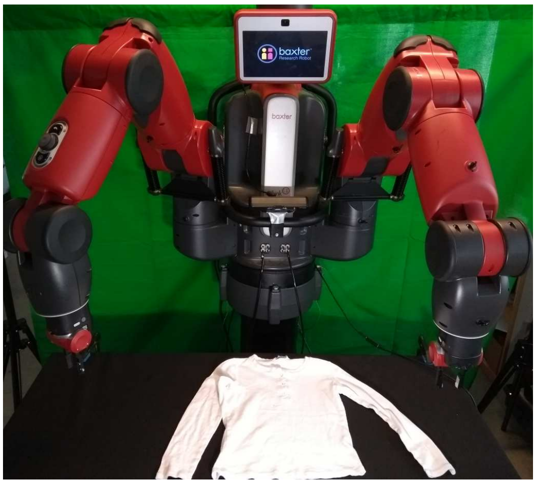


Figure 5.11: Experimental setup composed of a Baxter Robot and an RGB-D camera.

is produced until the goal observation is reached or the task is terminated. Such re-planning setup is used for all folding experiments. As the goal configuration does not allude to how the sleeves should be folded, the LSR suggests multiple latent plans. A subset of the corresponding visual action plans is shown on the left of Figure 5.12. If multiple plans are generated, a human operator selects one to execute. After the first execution, the ambiguity arising from the sleeve folding is removed. The re-planning therefore generates a single plan, shown in the right, that leads from start to goal state.

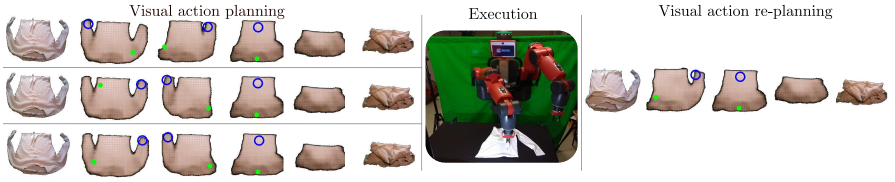


Figure 5.12: Execution of the folding task with re-planning. On the left, a set of initial visual action plans reaching the goal state is proposed. After the first execution, only one viable visual action plan remains.

Implementation details

Similarly to Section 5.5, the notation VAE_{ld}-f-d is used to denote a VAE with ld-dimensional latent space, where f stands for the folding task and d indicates whether or not the model was trained with the action loss (5.4). In particular, the case $d = b$ denotes *baseline* VAEs which are trained with the original training objective (5.5), while $d = L_p$ denotes *action* VAEs trained with the objective (5.6) containing the action term (5.4) using metric L_p for $p \in \{1, 2, \infty\}$. The VAEs are modeled with the same ResNet architecture and same hyperparameters β, γ and d_m as in the box stacking task introduced in Section 5.5 but increase the latent space dimension to $ld = 16$. For the LSR, the same notation as in Section 5.5.2 is adopted, where LSR-L_p denotes a graph obtained by using metric L_p in Algorithm 2. The upper bound $c_{\max}$ on the maximum number of graph-connected components in (5.8) is to 5, and the search interval boundaries $\tau_{\min}$ and $\tau_{\max}$ in Algorithm 3 are set to 0 and 3.5, respectively. The performance of the APMs and the evaluation of the system is based on the VAE_{16}-f-L_1 realization of the MM. The experiments are thus performed using APN_{16}-f-L_1 which is trained on latent action pairs $\overline{\mathcal{T}_z}$ extracted by the latent mapping ξ of VAE_{16}-f-L_1. Five models are trained for 500 epochs using different random seeds as in case of VAEs. Finally, 15% of the training dataset is used as a validation split used to extract the best performing model for the evaluation.

Covered regions using LSR

As in the box stacking tasks, the covered regions identified by the LSR are analyzed. To this aim, the model VAE_{16}-f-L_1 is used and the following inputs are considered: 224 novel observations, that correspond to possible states of the system and that are not used during training, and 5000 images from each of the 3D Shapes and CIFAR-10 datasets, which represent out-of-distribution samples that are not resembling the training data. The LSR achieves good recognition performance even in the folding task. More specifically, on average 213/224 samples representing the true underlying states of the system are correctly recognized as covered, resulting in $95 \pm 2.4\%$ accuracy averaged over the 5 different random seeds. For 3D Shapes dataset, 0/5000 samples are recognized as covered, while only 20/5000 samples from the CIFAR-10 dataset are on average wrongly recognized as covered. This analysis thus confirms the effectiveness of the LSR in capturing the covered regions of the latent space. It also shows the greater complexity of learning the latent mapping on real world observations, representing states of deformable objects, than on the simulated observations.

System performance

The proposed method is here experimentally validated and compared with the preliminary framework [153] on which it builds, as well as it is additionally employed on a more challenging fold that involves picking a layer of the cloth on top of another layer. In particular, the following quantities are evaluated: *(i)* the *system success rate*, i.e., a folding is considered successful if the system is able to actually fold the T-shirt into the desired goal configuration, *(ii)* the percentage of successful *transitions of the system*, i.e., a transition is considered successful if the respective folding step is executed correctly, *(iii-iv)* the quality of the generated visual plans P_I and action plans P_u, i.e., a visual (action) plan successful if all the intermediate states (actions) are correct. This evaluation is done by a human for a given fold on the very first generated visual action plan.

Concerning the comparison with [153], five type of folds are carried out and each fold is repeated five times using framework S-OUR, consisting of VAE$_{16}$-f-L_1, LSR-L_1 and APN$_{16}$-f-L_1, and is compared with work [153] obtained using S-L_1, S-L_2 and S-L_∞. The results are shown in Table 5.6, while all execution videos, including the respective visual action plans, are available on the website[2]. It can be noticed that S-PROP outperforms the systems from [153] with a notable 96% system performance, only missing a single folding step which results in a transition performance of 99%. As for S-L_1 [153], S-PROP also achieves optimal

Method	Syst.	Trans.	P_I	Pu
Fold 1 to 5 - comparison to [153]				
S-PROP	**96%**	**99%**	**100%**	**100%**
S-L_1 [153]	80%	90%	100%	100%
S-L_2 [153]	40%	77%	60%	60%
S-L_∞ [153]	24%	44%	56%	36%
Fold layer				
S-PROP	50%	83%	100%	100%

Table 5.6: Results (best in bold) for executing visual action plans on 5 folding tasks (each repeated 5 times) shown in the top. The bottom row shows the results on the fold requiring to pick the top layer of the garment (repeated 10 times).

performance when scoring the initial visual plans P_I as well as the initial action plans P_u.

Concerning the additional fold, it is repeated 10 times and the final results are reported in Table 5.6 (bottom row). It can be observed that the system has no trouble planning the folding but fails to pick up the top layer of the T-shirt in half of the cases during the plan execution. This is due to the imprecision of the Baxter and the difficulty of picking up layered clothing. The generated action plan, however, correctly identify the layer fold as a fold where it had to pick the top layer. Therefore, methods that are specialized in performing a layered cloth picking could be integrated into the proposed system.

Finally, the APN performance is also evaluated through the mean squared error (MSE) between the predicted and the ground truth action specifics on picking and releasing as well as the total model error. In detail, it achieves 82.6 ± 22.9, 29.3 ± 2.2, 270.6 ± 158.2, 71.8 ± 15.0, 0.0 ± 0.0, 454.3 ± 153.8 for the picking and release coordinates p_c, p_r, r_c, r_r, h and the overall error, respectively.

It can be therefore concluded that the proposed framework allows to effectively manipulate deformable objects in a real setup as well as improved MM, LSR and APM modules together contribute to a significant better system than in [153].

Chapter 6

Conclusions

This thesis work aimed to investigate human multi-robot interaction from multiple perspectives. Indeed, coordinating multiple robots while letting them interact with human operators opens up several issues that need to be addressed for an effective collaboration. First, human safety must be ensured at all times in scenarios where humans and robots work side-by-side. This must be guaranteed regardless of the human dynamic behavior and the task. Next, proper strategies must be designed which allow to achieve synergy between human and robots according to the desired interaction, whether assistance or shared control. Then, the robots' control strategy must ensure the achievement of the desired human interaction while complying with possible constraints of the robotic system. In this regard, distributed architectures are generally desirable since they confer higher flexibility and robustness to faults to the system with respect to centralized ones.

In light of the above, this thesis presented solutions that allow to realize human multi-robot interaction to different extents by combining several methodologies from control theory, robotics and machine learning. More specifically, the problem of ensuring a *reliable* multi-robot system for human operators was first addressed. To this aim, a distributed fault detection and isolation strategy was proposed which, on the basis of residual signals and dynamic thresholds, enables each robot to monitor the state of health of all the teammates, even if no direct communication with them exists. Despite the potentiality of this approach to make the multi-robot system reliable, its main limitation is given by the need to define reasonable bounds for modeling uncertainties and disturbances in the operational space. This limitation can be mitigated by resorting to techniques for the identification of the robots model [126]. Moreover, as empathized in Section 2, it only allows to identify faults that affect the end effector dynamics. A *joint* space formulation should be carried out in the case this assumption is not reasonable for the system at hand. However, this would require each robot to access local information of the other robots in the team, i.e., their joint configurations, thus reducing the independence of the robots and their decoupling.

A workspace sharing scenario was then considered in which humans and robots are allowed to work side-by-side without explicit exchange of forces between them. To this aim, a trajectory scaling approach relying on distributed observers was designed to ensure a minimum level of human safety at all times while preserving as much as possible the planned path. In this case, the main criticality lies in the need to define the value of the minimum allowed human safety. Although a procedure was defined to calibrate this minimum safety in order to ensure a minimum human-robot *distance*, this may lead to rather conservative values that, in restricted work spaces, could compromise the effectiveness of the human multi-robot system. This can be mitigated by defining a minimum safety value based on the *perceived* safety of the person rather than on the minimum distance. However, this approach would not provide any guarantee on the human-robot minimum distance.

The possibility to physically interact with a controlled exchange of wrenches on a co-manipulated object was subsequently introduced. In this regard, a two-layer architecture was devised which, at the top layer, determines the desired object dynamics depending on the required interaction and, at the bottom layer, imposes this dynamics in a distributed fashion while controlling internal stresses. In particular, in the case of shared control, a Linear Quadratic Tracking problem was formulated which takes into account human and robot intentions, being the former estimated through Recursive Least Squares. In the case of physical assistance, an adaptive and robust control law was devised to guarantee human wrench regulation. Despite the formal proofs provided in Section 4, the main criticality of the proposed approaches lies in the assumption of knowing the human arm end-point model which is approximated as a damper-spring system. Although this assumption is commonly embraced in the research community, an extensive experimental campaign would be required to thoroughly validate the proposed solutions. This will be subject of future work.

Moreover, the unpredictable human behavior and the typical lack of reciprocal human-robot awareness may lead to the occurrence of *accidental* contacts besides *intentional* ones in physical interaction scenarios. A data-driven approach was therefore proposed which, based on recurrent neural networks and Gaussian mixture models, allows to detect and classify possible human contact, even in case the robotic task requires exchange of wrenches with the environment. In this case, the main criticality lies in the absence of a formal analysis that guarantees the detection and correct classification of human contact. In this regard, the experimental validation proved that the detection is generally correctly performed, while the recognition of the nature of contact suffers from more inaccuracies. However, the devised reaction strategies allow in each case to limit possible human harm.

Finally, the possibility of learning from human examples was included and a methodology was designed to perform visual planning on the basis of high-dimensional examples in the image space. To this aim, Variational Auto-Encoders were employed to map the high-dimensional observations into a lower-dimensional latent space. Based on this, a graph-based structure was built to plan sequence of actions (and respective images) from start to goal. Here, the main limitation of the proposed approach is that a *sufficient* number of examples (i.e., including different configurations of the system) must be provided as input to the solution in order to effectively plan from and to different possible states of the system. This limitation can be mitigated to a certain extent by including active learning techniques [176] that allow to learn online by exploiting information provided by human operators included in the planning loop.

6.1 Future work

Human-robot interaction is a fervent research field that potentially finds application in a multitude of contexts, ranging from industrial to domestic ones. Endowing the robots with the ability to interact and collaborate with human operators is a disruptive technology which is, however, still far from an effective practical integration in real systems.

As future work, the problem of detecting and recognizing the type of human contact in physical interaction scenarios will be extended to multi-robot setups. In this context, the problem arises of discerning possible internal wrenches in case of co-manipulation of a rigid object and, based on these, to recognize possible human contact. Then, it is necessary to establish proper reaction behaviors of the team when the human interacts, voluntarily or not, only with a subset of it. Moreover, the possibility to also learn from the human intentional interaction and adapt the robot task accordingly will be investigated and an extensive experimental campaign will be carried out to validate the approach in a real-world setting, by using both qualitative and quantitative measures.

Next, the understandability of visual plans will also be exploited for planning human-robot collaborative tasks. Indeed, the current framework envisages only the robot as *executive* agent. However, the inclusion of multiple agents that simultaneously and/or sequentially execute the planned actions can benefit the system in terms of performance. A *collaborative* visual action plan would then comprise a sequence of images representing the intermediate states to reach the goal configuration as well as the sequence of actions and the respective agents that must perform them in order to achieve the state transitions in the visual plan. In order to select the agents, optimization indices can be included which, for example, take into account the human level of fatigue. Note that the presence of images in the plans is particularly noteworthy for formulating collaborative tasks since they can be *intuitively* understood by human operators. The approach delineated above could also be relevant to realize co-manipulation of *deformable* objects by humans and robots. In this case, in fact, the presence of humans can be useful to overcome the complexity of manipulating deformable objects, and, at the same time, the use of visual information solves the problem of explicitly representing the state of these objects. This scenario, although potentially effective, is still unexplored in the current state of the art.

Finally, according to the most recent research directions, methodologies for awareness of the surrounding environment as well as human intentions will be investigated in order to make the robotic system proactive towards the human collaborator. In this way, depending on the context, the robot can try to identify the most favorable action to carry out according to what the human wants to achieve. It follows that the introduction of this awareness potentially enhances the reasoning skills of the robot system and makes it closer to a real collaborator.

Appendix A

Proofs for FDI scheme

A.1 Proof of Lemma 2.1

By differentiating (2.31) and after some straightforward steps, the following holds

$$\dot{\boldsymbol{v}}^\star = \dot{\hat{\boldsymbol{z}}}^\star - \boldsymbol{I}_N \otimes \bar{\boldsymbol{C}}\hat{\boldsymbol{\theta}}_g^\star - \boldsymbol{I}_N \otimes \bar{\boldsymbol{F}}\hat{\boldsymbol{z}}^\star \tag{A.1}$$

thus, from (2.32) it follows

$$\dot{\hat{\boldsymbol{v}}}^\star = -k_o\boldsymbol{L}^\star\hat{\boldsymbol{v}}^\star + k_o\boldsymbol{\Pi}^\star\tilde{\boldsymbol{v}}^\star. \tag{A.2}$$

By exploiting the following property of the Laplacian matrix [35]

$$\boldsymbol{L}^\star(\boldsymbol{1}_N \otimes \boldsymbol{v}) = (\boldsymbol{L}^\star \otimes \boldsymbol{I}_{2Np})(\boldsymbol{1}_N \otimes \boldsymbol{v}) = (\boldsymbol{L}\boldsymbol{1}_N) \otimes (\boldsymbol{I}_{2Np}\boldsymbol{v}) = \boldsymbol{0}_{2Np^2} \tag{A.3}$$

(A.2) can be rearranged as

$$\dot{\hat{\boldsymbol{v}}}^\star = -k_o\left(\boldsymbol{L}^\star + \boldsymbol{\Pi}^\star\right)\tilde{\boldsymbol{v}}^\star = -k_o\tilde{\boldsymbol{L}}^\star\tilde{\boldsymbol{v}}^\star. \tag{A.4}$$

By considering from (2.27) that $\dot{\boldsymbol{v}} = \bar{\boldsymbol{B}}(\boldsymbol{z})\left(\boldsymbol{f}_M - \bar{\boldsymbol{d}}\right)$, it holds

$$\dot{\tilde{\boldsymbol{v}}}^\star = \boldsymbol{1}_N \otimes \dot{\boldsymbol{v}} - \dot{\hat{\boldsymbol{v}}}^\star = -k_o\tilde{\boldsymbol{L}}^\star\tilde{\boldsymbol{v}}^\star + \boldsymbol{f}^\star - \bar{\boldsymbol{d}}^\star \tag{A.5}$$

where $\boldsymbol{f}^{\star} = \mathbf{1}_N \otimes \bar{\boldsymbol{B}}(\boldsymbol{z})\boldsymbol{f}_M$ and $\bar{\boldsymbol{d}}^{\star} = \mathbf{1}_N \otimes \bar{\boldsymbol{B}}(\boldsymbol{z})\bar{\boldsymbol{d}}$. Finally, in the absence of faults and uncertainties, the dynamics of $\tilde{\boldsymbol{v}}^{\star}$ is

$$\dot{\tilde{\boldsymbol{v}}}^{\star} = \mathbf{1}_N \otimes \dot{\boldsymbol{v}} - \dot{\hat{\boldsymbol{v}}}^{\star} = -k_o\tilde{\boldsymbol{L}}^{\star}\tilde{\boldsymbol{v}}^{\star}. \tag{A.6}$$

By leveraging the results in [55], it can be stated that $-\tilde{\boldsymbol{L}}^{\star}$ is Hurwitz provided that the communication graph is strongly connected. Thus, in the absence of faults and model uncertainties, (A.6) proves the lemma $\forall k_o > 0$.

A.2 Proof of Theorem 2.2

By considering the dynamics of $\boldsymbol{z}$ in (2.27) and of $\hat{\boldsymbol{z}}^{\star}$ in (2.32), and the Laplacian matrix property (A.3), in the absence of faults and uncertainties, the dynamics of $\tilde{\boldsymbol{z}}^{\star}$ can be written as

$$\dot{\tilde{\boldsymbol{z}}}^{\star} = -k_o\tilde{\boldsymbol{L}}^{\star}\tilde{\boldsymbol{v}}^{\star} + \boldsymbol{I}_N \otimes \bar{\boldsymbol{C}}\tilde{\boldsymbol{\theta}}_g^{\star} + \boldsymbol{I}_N \otimes \bar{\boldsymbol{F}}\tilde{\boldsymbol{z}}^{\star} \tag{A.7}$$

where $\tilde{\boldsymbol{\theta}}_g^{\star} = \mathbf{1}_N \otimes \boldsymbol{\theta}_g - \hat{\boldsymbol{\theta}}_g^{\star}$.

By virtue of Lemma 2.1, the term $-k_o\tilde{\boldsymbol{L}}^{\star}\tilde{\boldsymbol{v}}^{\star}$ is a disturbance converging to zero at steady state, thus the convergence properties of $\tilde{\boldsymbol{z}}^{\star}$ are those of the non-perturbed system

$$\dot{\tilde{\boldsymbol{z}}}^{\star} = \boldsymbol{I}_N \otimes \bar{\boldsymbol{C}}\tilde{\boldsymbol{\theta}}_g^{\star} + \boldsymbol{I}_N \otimes \bar{\boldsymbol{F}}\tilde{\boldsymbol{z}}^{\star}. \tag{A.8}$$

Based on (2.25), the auxiliary global input $\boldsymbol{\theta}_g$ can be written as

$$\boldsymbol{\theta}_g = \sum_{l=1}^{N} \boldsymbol{\Gamma}_{u_l}^T \boldsymbol{\Gamma}_{u_l}{}^{l}\hat{\boldsymbol{\theta}}_g = \sum_{l=1}^{N} \boldsymbol{\Pi}_{u_l}{}^{l}\hat{\boldsymbol{\theta}}_g \tag{A.9}$$

where $\boldsymbol{\Pi}_{u_l} \in \mathbb{R}^{Np\times Np}$ is a matrix that nullifies all the elements of

${}^{l}\hat{\boldsymbol{\theta}}_g$ but $\boldsymbol{\theta}_{g,l}$. Therefore, the i th element of $\tilde{\boldsymbol{\theta}}_g^{\star}$ can be written as

$$\begin{aligned} {}^{i}\tilde{\boldsymbol{\theta}}_g &= \sum_{l=1}^{N} \boldsymbol{\Pi}_{u_l}(\boldsymbol{K}^{l}\hat{\boldsymbol{z}} + \boldsymbol{\theta}_f) - \boldsymbol{K}^{i}\hat{\boldsymbol{z}} - \boldsymbol{\theta}_f = \sum_{l=1}^{N} \boldsymbol{\Pi}_{u_l}\boldsymbol{K}({}^{l}\hat{\boldsymbol{z}} \pm \boldsymbol{z}) - \boldsymbol{K}^{i}\hat{\boldsymbol{z}} \\ &= -\sum_{l=1}^{N} \boldsymbol{\Pi}_{u_l}\boldsymbol{K}^{l}\tilde{\boldsymbol{z}} + \boldsymbol{K}^{i}\tilde{\boldsymbol{z}} \end{aligned} \tag{A.10}$$

where the property $\sum_{l=1}^{N} \boldsymbol{\Pi}_{u_l}\boldsymbol{v} = \boldsymbol{v}$ for any vector $\boldsymbol{v}$ of proper dimensions has been exploited. From (A.10), $\tilde{\boldsymbol{\theta}}_g^{\star}$ is given by

$$\tilde{\boldsymbol{\theta}}_g^{\star} = (\boldsymbol{I}_N \otimes \boldsymbol{K})\tilde{\boldsymbol{z}}^{\star} - \boldsymbol{\Pi}_u^{\star}(\boldsymbol{I}_N \otimes \boldsymbol{K})\tilde{\boldsymbol{z}} = (\boldsymbol{I}_{Np^2} - \boldsymbol{\Pi}_u^{\star})(\boldsymbol{I}_N \otimes \boldsymbol{K})\tilde{\boldsymbol{z}}^{\star} \tag{A.11}$$

where $\boldsymbol{\Pi}_u^{\star} = \boldsymbol{1}_N \otimes \begin{bmatrix}\boldsymbol{\Pi}_{u_1} \ \boldsymbol{\Pi}_{u_2} \ \dots \ \boldsymbol{\Pi}_{u_N}\end{bmatrix} \in \mathbb{R}^{Np^2\times Np^2}$. It can be trivially verified that the following property holds for the matrix $\boldsymbol{I}_{Np^2} - \boldsymbol{\Pi}_u^{\star}$.

Property A.1. *Matrix* $\boldsymbol{I}_{Np^2} - \boldsymbol{\Pi}_u^{\star}$ *is idempotent, it is characterized by* Np *null rows and* $\operatorname{rank}(\boldsymbol{I}_{Np^2} - \boldsymbol{\Pi}_u^{\star}) = Np(N-1)$.

Based on the Property A.1, there exists a nonsingular permutation matrix $\boldsymbol{P} \in \mathbb{R}^{N^2\times N^2}$ such that

$$(\boldsymbol{P} \otimes \boldsymbol{I}_p)(\boldsymbol{I}_{Np^2} - \boldsymbol{\Pi}_u^{\star}) = \left[\begin{array}{c} \boldsymbol{0}_{Np\times Np^2} \\ \hline \boldsymbol{S} \end{array}\right] \tag{A.12}$$

where $\boldsymbol{S} \in \mathbb{R}^{Np(N-1)\times Np^2}$ is a full row rank matrix, i.e., $\operatorname{rank}(\boldsymbol{S}) = Np(N-1)$. From (A.8) and (A.11), the dynamics of $\tilde{\boldsymbol{z}}^{\star}$ is

$$\dot{\tilde{\boldsymbol{z}}}^{\star} = (\boldsymbol{I}_N \otimes \bar{\boldsymbol{F}})\tilde{\boldsymbol{z}}^{\star} + (\boldsymbol{I}_N \otimes \bar{\boldsymbol{C}})(\boldsymbol{I}_{Np^2} - \boldsymbol{\Pi}_u^{\star})(\boldsymbol{I}_N \otimes \boldsymbol{K})\tilde{\boldsymbol{z}}^{\star}. \tag{A.13}$$

Then, by considering the state transformation $\tilde{\boldsymbol{\omega}}^{\star} = (\boldsymbol{P} \otimes \boldsymbol{I}_{2p})\tilde{\boldsymbol{z}}^{\star}$ the system (A.13) can be rewritten as:

$$\begin{aligned} \dot{\tilde{\boldsymbol{\omega}}}^{\star} = {} & (\boldsymbol{P} \otimes \boldsymbol{I}_{2p})(\boldsymbol{I}_N \otimes \bar{\boldsymbol{F}})(\boldsymbol{P} \otimes \boldsymbol{I}_{2p})^{-1}\tilde{\boldsymbol{\omega}}^{\star} + \qquad \text{(A.14)} \\ & (\boldsymbol{P} \otimes \boldsymbol{I}_{2p})(\boldsymbol{I}_N \otimes \bar{\boldsymbol{C}})(\boldsymbol{I}_{Np^2} - \boldsymbol{\Pi}_u^{\star})(\boldsymbol{I}_N \otimes \boldsymbol{K})(\boldsymbol{P} \otimes \boldsymbol{I}_{2p})^{-1}\tilde{\boldsymbol{\omega}}^{\star}. \end{aligned}$$

By exploiting the *mixed-product property* of the Kronecker product [177]

$$(\boldsymbol{D}_1 \otimes \boldsymbol{D}_2)(\boldsymbol{D}_3 \otimes \boldsymbol{D}_4) = (\boldsymbol{D}_1\boldsymbol{D}_3) \otimes (\boldsymbol{D}_2\boldsymbol{D}_4) \tag{A.15}$$

with matrices $\boldsymbol{D}_1, \boldsymbol{D}_2, \boldsymbol{D}_3$, and $\boldsymbol{D}_4$ of proper dimensions, the following chains of equalities hold

$$\begin{aligned}(\boldsymbol{P} \otimes \boldsymbol{I}_{2p})(\boldsymbol{I}_N \otimes \bar{\boldsymbol{F}})(\boldsymbol{P} \otimes \boldsymbol{I}_{2p})^{-1} &= (\boldsymbol{P} \otimes \boldsymbol{I}_{2p})(\boldsymbol{I}_{N^2} \otimes \boldsymbol{F})(\boldsymbol{P}^{-1} \otimes \boldsymbol{I}_{2p}) \\ &= (\boldsymbol{P} \otimes \boldsymbol{F})(\boldsymbol{P}^{-1} \otimes \boldsymbol{I}_{2p}) = \boldsymbol{I}_N \otimes \bar{\boldsymbol{F}},\end{aligned} \tag{A.16}$$

$$\begin{aligned}(\boldsymbol{P} \otimes \boldsymbol{I}_{2p})(\boldsymbol{I}_N \otimes \bar{\boldsymbol{C}})(\boldsymbol{I}_{Np^2} - \Pi_u^\star) &= (\boldsymbol{P} \otimes \boldsymbol{I}_{2p})(\boldsymbol{I}_{N^2} \otimes \boldsymbol{C})(\boldsymbol{I}_{Np^2} - \Pi_u^\star) \\ &= (\boldsymbol{I}_{N^2} \otimes \boldsymbol{C})(\boldsymbol{P} \otimes \boldsymbol{I}_p)(\boldsymbol{I}_{Np^2} - \Pi_u^\star) \\ &= (\boldsymbol{I}_{N^2} \otimes \boldsymbol{C}) \left[\begin{array}{c} \boldsymbol{0}_{Np \times Np^2} \\ \hline \boldsymbol{S} \end{array}\right] = \left[\begin{array}{c} \boldsymbol{0}_{2Np \times Np^2} \\ \hline (\boldsymbol{I}_{N(N-1)} \otimes \boldsymbol{C})\boldsymbol{S} \end{array}\right].\end{aligned} \tag{A.17}$$

By considering (A.16), (A.17), and the structure of $\boldsymbol{I}_N \otimes \bar{\boldsymbol{F}}$, the system (A.14) can be partitioned as

$$\begin{aligned}\dot{\tilde{\boldsymbol{\omega}}}^\star &= \left[\begin{array}{c|c} \boldsymbol{I}_N \otimes \boldsymbol{F} & \boldsymbol{O}_{2Np,2Np(N-1)} \\ \hline \boldsymbol{O}_{2Np(N-1),2Np} & \boldsymbol{I}_{N(N-1)} \otimes \boldsymbol{F} \end{array}\right] \tilde{\boldsymbol{\omega}}^\star \\ &\quad + \left[\begin{array}{c} \boldsymbol{O}_{2Np \times Np^2} \\ \hline (\boldsymbol{I}_{N(N-1)} \otimes \boldsymbol{C})\boldsymbol{S} \end{array}\right] (\boldsymbol{I}_N \otimes \boldsymbol{K})\tilde{\boldsymbol{z}}^\star \\ &= \left[\begin{array}{c|c} \boldsymbol{F}_\omega^u & \boldsymbol{O}_{2Np,2Np(N-1)} \\ \hline \boldsymbol{O}_{2Np \times Np^2} & \boldsymbol{F}_\omega^c \end{array}\right] \tilde{\boldsymbol{\omega}}^\star + \left[\begin{array}{c} \boldsymbol{O}_{2Np \times Np^2} \\ \hline \boldsymbol{C}_\omega^c \end{array}\right] (\boldsymbol{I}_N \otimes \boldsymbol{K})\tilde{\boldsymbol{z}}^\star.\end{aligned} \tag{A.18}$$

System (A.18) can be viewed as a linear system with input $(\boldsymbol{I}_N \otimes \boldsymbol{K})\tilde{\boldsymbol{z}}^\star$; such a system is composed by the subsystem $\{\boldsymbol{F}_\omega^u, \boldsymbol{O}_{2Np \times Np^2}\}$ that is clearly uncontrollable, and the subsystem $\{\boldsymbol{F}_\omega^c, \boldsymbol{C}_\omega^c\}$, that is controllable if the system $\{\boldsymbol{F}, \boldsymbol{C}\}$ is controllable since the matrix $\boldsymbol{S}$ is a full row rank matrix. In order to make $\tilde{\boldsymbol{\omega}}^\star$ convergent to zero the following conditions must be met:

C1 The uncontrollable part $\{\boldsymbol{F}_\omega^u, \boldsymbol{O}_{Np \times Np}\}$ must be asymptotically stable. It is straightforward to recognize that this implies that the dynamics of system (2.26) is asymptotically stable;

C2 The control input gain $(\boldsymbol{I}_N \otimes \boldsymbol{K})$ has to be chosen in such a way to stabilize the subsystem $\{\boldsymbol{F}^c_\omega, \boldsymbol{C}^c_\omega\}$.

In order to guarantee that the conditions C1 and C2 are satisfied, the following choices of the control matrices must be done

- The matrix gain $\boldsymbol{K}_s$ has to be chosen in such a way that $\boldsymbol{F} = \boldsymbol{A}+\boldsymbol{C}\boldsymbol{K}_s$ is a Hurwitz matrix. It is worth noticing that, since $(\boldsymbol{A},\boldsymbol{C})$ represents a controllable system, the existence of such a matrix is ensured;

- The matrix gain $\boldsymbol{K}$ has to be chosen in such a way to stabilize the subsystem $\{\boldsymbol{F}^c_\omega, \boldsymbol{C}^c_\omega\}$. Again, the controllability of $(\boldsymbol{A}, \boldsymbol{C})$ ensures that such a matrix exists.

It is worth noticing that the choice of both $\boldsymbol{K}_s$ and $\boldsymbol{K}$ does not depend on a particular topology, thus the stability condition can be checked off-line on the basis of the knowledge of the sole system (2.26).

A.3 Proof of Theorem 2.3

From (A.5), after some algebraic steps, the dynamics of $\tilde{\boldsymbol{v}}^\star_k$, in the absence of faults, is

$$\dot{\tilde{\boldsymbol{v}}}^\star_k = -k_o \boldsymbol{\Gamma}^\star_k \tilde{\boldsymbol{L}}^\star \tilde{\boldsymbol{v}}^\star - \boldsymbol{\Gamma}^\star_k \bar{\boldsymbol{d}}^\star = -k_o \tilde{\boldsymbol{L}}^\star_k \tilde{\boldsymbol{v}}^\star_k - \boldsymbol{1}_N \otimes \boldsymbol{B}_k(\boldsymbol{z}_k)_k. \qquad \text{(A.19)}$$

From (2.36) and by virtue of Property 2.6, the residual ${}^i\boldsymbol{r}_k$, in the absence of faults, can be written as

$$\begin{aligned} {}^i\boldsymbol{r}_k = & \boldsymbol{\Gamma}_i \tilde{\boldsymbol{L}}^\star_k \tilde{\boldsymbol{v}}^\star_k = \boldsymbol{\Gamma}_i \tilde{\boldsymbol{L}}^\star_k \Big[\mathrm{e}^{-k_o \tilde{\boldsymbol{L}}^\star_k t}\, \tilde{\boldsymbol{v}}^\star_k(0) \\ & - \int_0^t \mathrm{e}^{-k_o \tilde{\boldsymbol{L}}^\star_k (t-\tau)} \left(\boldsymbol{1}_N \otimes \boldsymbol{B}_k(\boldsymbol{z}_k)_k(\tau) \right) d\tau \Big]. \end{aligned} \qquad \text{(A.20)}$$

Therefore it can be upper bounded as

$$\begin{aligned}\left\|{}^{i}\boldsymbol{r}_k\right\| \leq & \left\|\boldsymbol{\Gamma}_i \tilde{\boldsymbol{L}}_k^{\star}\right\| \left[\left\| \mathrm{e}^{-k_o \tilde{\boldsymbol{L}}_k^{\star} t} \tilde{\boldsymbol{v}}_k^{\star}(0) \right\| \right. \\ & \left. + \int_0^t \left\| \mathrm{e}^{-k_o \tilde{\boldsymbol{L}}_k^{\star}(t-\tau)} \left(\boldsymbol{1}_N \otimes \boldsymbol{B}_k(\boldsymbol{z}_k)_k(\tau) \right) d\tau \right\| \right] \\ \leq & \left\|\boldsymbol{\Gamma}_i \tilde{\boldsymbol{L}}_k^{\star}\right\| \left[\left\|\tilde{\boldsymbol{v}}_k^{\star}(0)\right\| \kappa \mathrm{e}^{-\lambda t} + \frac{\kappa \sqrt{N}\, \bar{d}}{\lambda\, \varepsilon_m(\bar{\boldsymbol{M}}_k)} (1 - \mathrm{e}^{-\lambda t}) \right]\end{aligned} \tag{A.21}$$

where $\varepsilon_m(\bar{\boldsymbol{M}}_k)$ is the minimum eigenvalue of $\bar{\boldsymbol{M}}_k$, while Property 2.6 and the following inequality

$$\left\|\boldsymbol{1}_N \otimes \boldsymbol{B}_k(\boldsymbol{z}_k)_k\right\| \leq \frac{\sqrt{N}}{\varepsilon_m(\bar{\boldsymbol{M}}_k)} \bar{d} \tag{A.22}$$

have been taken into account. Moreover, it holds [50]

$$\left\|\boldsymbol{\Gamma}_i \tilde{\boldsymbol{L}}_k^{\star}\right\| \quad \leq \quad \sqrt{2p} d_i + {}^{i}\delta_k \tag{A.23}$$

with d_i the in-degree of node i, and ${}^{i}\delta_k = 1$ if $i = k$ and ${}^{i}\delta_k = 0$ otherwise. Therefore, the right-hand member of (A.21) can be written as a time-varying threshold

$${}^{i}\mu_k(t) = (\sqrt{2p} d_i + {}^{i}\delta_k) \left(\left\|\tilde{\boldsymbol{v}}_k^{\star}(0)\right\| \kappa \mathrm{e}^{-\lambda t} + \frac{\kappa \sqrt{N}\, \bar{d}}{\lambda\, \varepsilon_m(\bar{\boldsymbol{M}}_k)} (1 - \mathrm{e}^{-\lambda t}) \right).$$

The inequality (A.21) and the above equation prove the theorem.

Remark A.1. *The calculation of the thresholds ${}^{i}\mu_k$ requires a reliable estimate of $\|\tilde{\boldsymbol{v}}_k^{\star}(0)\|$, λ and κ. The constant $\|\tilde{\boldsymbol{v}}_k^{\star}(0)\|$ can be estimated on the basis of approximate information about the initial state of the system by supposing that the robots are included on a known bounded area. Regarding λ and κ, if the Laplacian matrix of the system is known can be computed as in [56], otherwise the Laplacian matrix can be estimated by considering the worst case* *scenario.*

A.4 Proof of Theorem 2.4

In the presence of a fault affecting robot k, it holds $\|\boldsymbol{f}_k\| > 0$, therefore the dynamics of $\tilde{\boldsymbol{v}}_k^\star$ (A.19) becomes

$$\dot{\tilde{\boldsymbol{v}}}_k^\star = -k_o \tilde{\boldsymbol{L}}_k^\star \tilde{\boldsymbol{v}}_k^\star - \mathbf{1}_N \otimes \boldsymbol{B}_k(\boldsymbol{z}_k)\left({}_k + \boldsymbol{J}_{M,k}^T \boldsymbol{f}_k\right) \tag{A.24}$$

and, thus, by virtue of (A.20), the residual components ${}^i\boldsymbol{r}_k$ become

$$\begin{aligned} {}^i\boldsymbol{r}_k =& \boldsymbol{\Gamma}_i \tilde{\boldsymbol{L}}_k^\star \Bigg[\mathrm{e}^{-k_o \tilde{\boldsymbol{L}}_k^\star t} \tilde{\boldsymbol{v}}_k^\star(0) - \int_0^t \mathrm{e}^{-k_o \tilde{\boldsymbol{L}}_k^\star (t-\tau)} \left(\mathbf{1}_N \otimes \boldsymbol{B}_k(\boldsymbol{z}_k)_k(\tau)\right) d\tau \\ &+ \int_0^t \mathrm{e}^{-k_o \tilde{\boldsymbol{L}}_k^\star (t-\tau)} (\mathbf{1}_N \otimes \boldsymbol{B}_k(\boldsymbol{z}_k) \boldsymbol{J}_{M,k}^T \boldsymbol{f}_k(\tau)) d\tau \Bigg]. \end{aligned} \tag{A.25}$$

Equation (A.25) proves that the residual ${}^i\boldsymbol{r}_k$ is affected by the fault. On the other side, by considering that the matrix $\boldsymbol{\Gamma}_j^\star$ selects only the components of the vector $\boldsymbol{f}_M$ associated with the j th robot, which are null, the dynamics of ${}^i\boldsymbol{v}_j^\star$ with $j \neq k$, is

$$\dot{\tilde{\boldsymbol{v}}}_j^\star = -k_o \tilde{\boldsymbol{L}}_j^\star \tilde{\boldsymbol{v}}_j^\star - \mathbf{1}_N \otimes \boldsymbol{B}_j(\boldsymbol{z}_j)_j \tag{A.26}$$

which is not affected by the faulty robot. This proves the theorem.

Appendix B

Minimum safety index

In this Appendix, the computation of the minimum value $\bar{F}_{min}$ which ensures that $\bar{F} \geq \bar{F}_{min}$ implies $d^* \geq d_{min}$, being d^* the minimum human multi-robot distance in (3.8), is provided. To this aim, the case of single manipulator ($N = 1$) composed of n_l links is first considered. The required $\bar{F}_{min}$ is determined by evaluating the maximum value of the safety index $\bar{F}$ which can be obtained when a human point is at minimum distance $d^* = d_{min}$ from any point of the robot, for all possible human configurations. This ensures that if $\bar{F} \geq \bar{F}_{min}$ then certainly all points of the robot are at greater distance than d_{min} from all the points of the human operator. In light of this, it holds

$$\begin{aligned}\bar{F} &= \frac{1}{n_o}\sum_{j=1}^{n_o}\sum_{l=1}^{n_l}\int_0^1 \Big(\alpha_1(d_l^s) + \alpha_2(d_l^s, \dot{d}_l^s)\Big)\ ds \\ &\leq \frac{1}{n_o}\sum_{j=1}^{n_o}\sum_{l=1}^{n_l}\left(\int_0^1 \alpha_1(\|\boldsymbol{p}_l^0 + s(\boldsymbol{p}_l^1 - \boldsymbol{p}_l^0) - \boldsymbol{p}_{o,j}\|)\ ds + k\right) \\ &\leq \frac{1}{n_o}\sum_{j=1}^{n_o}\sum_{l=1}^{n_l}\left(\int_0^1 \alpha_1(\|\boldsymbol{p}_l^0 - \boldsymbol{p}_{o,j}\| + s\|\boldsymbol{p}_l^1 - \boldsymbol{p}_l^0\|)\ ds\right) + k\, n_l \\ &= \frac{1}{n_o}\sum_{j=1}^{n_o}\sum_{l=1}^{n_l}\left(\int_0^1 \alpha_1(\|\boldsymbol{p}_l^0 - \boldsymbol{p}_{o,j}\| + sL_l)\ ds\right) + k\, n_l\end{aligned} \tag{B.1}$$

being $L_l = \|\boldsymbol{p}_l^1 - \boldsymbol{p}_l^0\|$ the length of the lth link. Without loss of generality, let the human point j^* be the one at minimum distance d_{min} from a point $\boldsymbol{p}^*$ on the robot structure, that is $\|\boldsymbol{p}_{o,j^*} - \boldsymbol{p}^\star\| = d_{min}$; then, the following inequalities are verified

$$\begin{aligned}\|\boldsymbol{p}_l^0 - \boldsymbol{p}_{o,j^*} \pm \boldsymbol{p}^\star\| &\leq \|\boldsymbol{p}_l^0 - \boldsymbol{p}^\star\| + \|\boldsymbol{p}_{o,j^*} - \boldsymbol{p}^\star\| \\ &\leq \sum_{l=1}^{n_l} L_l + d_{min} = L + d_{min} \triangleq \bar{L}\end{aligned} \tag{B.2}$$

where L is the length of the overall robot structure, i.e., $L = \sum_{l=1}^{n_l} L_l$, and which are obtained by taking into account that the distance between two points of the robot structure is at most equal to the overall length L, thus it holds $\|\boldsymbol{p}_l^0 - \boldsymbol{p}^\star\| \leq L$. Similarly, since the human skeleton is modeled through rigid links, it is always possible to define a maximum constant distance L_o between any two points belonging to it, which is defined as follows

$$L_o = \max_{\forall\, i,k} \|\boldsymbol{p}_{o,i} - \boldsymbol{p}_{o,k}\|. \tag{B.3}$$

As a result, the following chain of inequalities holds

$$\|\boldsymbol{p}_l^0 - \boldsymbol{p}_{o,j} \pm \boldsymbol{p}^\star\| \leq L + \|\boldsymbol{p}_{o,j} \pm \boldsymbol{p}_{o,j^*} - \boldsymbol{p}^\star\| \tag{B.4}$$

$$\begin{aligned} &\leq L + \|\boldsymbol{p}_{o,j} - \boldsymbol{p}_{o,1}\| + \|\boldsymbol{p}_{o,j^*} - \boldsymbol{p}^\star\| \\ &\leq \bar{L} + L_o. \end{aligned} \tag{B.5}$$

Therefore, by considering (B.1) and (B.4), it follows

$$\bar{F} \leq \frac{1}{n_o} \sum_{j=1}^{n_o} \sum_{l=1}^{n_l} \left(\int_0^1 \alpha_1(\bar{L} + L_o + sL_l)\ ds \right) + k\, n_l.$$

By exploiting the Lipschitz property of the coefficient $\alpha_1(d)$ and by denoting with k_L the respective constant, one obtains

$$\begin{aligned} \bar{F} &\leq \frac{1}{n_o} \sum_{j=1}^{n_o} \sum_{l=1}^{n_l} \left(\int_0^1 |\alpha_1(\bar{L} + L_o + sL_l) \pm \alpha_1(\bar{L} + L_o)|\ ds \right) + k\, n_l \\ &\leq \frac{1}{n_o} \sum_{j=1}^{n_o} \sum_{l=1}^{n_l} \left(\int_0^1 \big(sk_L L_l + |\alpha_1(\bar{L} + L_o)| \big)\ ds \right) + k\, n_l \\ &= \frac{1}{n_o} \sum_{j=1}^{n_o} \sum_{l=1}^{n_l} \left(\frac{1}{2} k_L L_l + |\alpha_1(\bar{L} + L_o)| \right) + k\, n_l \end{aligned} \tag{B.6}$$

which leads to the following $\bar{F}_{min}$

$$\bar{F}_{min} = \frac{1}{2} k_L L + \big(|\alpha_1(\bar{L} + L_o)| + k \big) n_l. \tag{B.7}$$

By generalizing (B.1) to N robots, it holds

$$F \leq \frac{1}{n_o} \sum_{i=1}^{N} \sum_{j=1}^{n_o} \left[\sum_{l=1}^{n_i+1} \left(\int_0^1 \alpha_1(\|\boldsymbol{p}_{i,l}^0 - \boldsymbol{p}_{o,j}\| + sL_l)\ ds \right) + k\,(n_i + 1) \right] \tag{B.8}$$

where $L_i = \sum_{l=1}^{n_i+1} L_{i,l}$ is the length of the overall structure of robot i and also takes into account a virtual link to the team centroid. At this point, the structure composed by the ith manipulator and the one at minimum distance d_{min} can be analyzed in turn as an "aggregate" manipulator whose maximum overall length is $L_i + L_{max}$ with $L_{max} = \max\limits_{i \in 1,..,N} L_i$; therefore by also recalling (B.2) and (B.4), it holds

$$\|\boldsymbol{p}_{i,l}^{0} - \boldsymbol{p}_{o,j}\| \leq L_i + L_{max} + L_o + d_{min}. \tag{B.9}$$

By following the same reasoning as in (B.6) and by combining (B.8) and (B.9), it finally follows

$$\bar{F}_{min} = \sum_{i=1}^{N} \left[\frac{1}{2} k_L L_i + (|\alpha_1(L_i + L_{max} + L_o + d_{min})| + k)\,(n_i + 1)\right] \tag{B.10}$$

which is such as to ensure that each point on each manipulator is at least at distance d_{min} from the operator. As an example, by considering the formulation in (3.41), eq. (B.10) becomes

$$\begin{aligned}\bar{F}_{min} = \sum_{i=1}^{N} \Big[k_1 \Big(\frac{2n_i + 3}{2} L_i + (n_i + 1)(L_{max} + L_o + d_{min})\Big) \\ + k_2\,(n_i + 1)\Big].\end{aligned}$$

Appendix C

Additional material for physical interaction

C.1 Proof of Theorem 4.3

To prove Theorem 4.3, the closed loop dynamics or robot i is first derived. By replacing (4.43) in (2.10), it holds

$$\bar{\boldsymbol{M}}_i \dot{\boldsymbol{s}}_i = -\bar{\boldsymbol{C}}_i \boldsymbol{s}_i - \boldsymbol{K}_s \boldsymbol{s}_i - \Delta \boldsymbol{u}_i + \boldsymbol{h}_i + \bar{\boldsymbol{Y}}_i(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i, \boldsymbol{\rho}_i, \dot{\boldsymbol{\rho}}_i)\tilde{\boldsymbol{\pi}}_i. \tag{C.1}$$

The following Lyapunov function is considered

$$V = \frac{1}{2}\sum_{i=1}^{N}\left(\boldsymbol{s}_i^T \bar{\boldsymbol{M}}_i \boldsymbol{s}_i + \frac{1}{k_f}\hat{\Delta}\boldsymbol{u}_{f,i}^T \hat{\Delta}\boldsymbol{u}_{f,i} + \tilde{\boldsymbol{\pi}}_i^T \boldsymbol{K}_\pi \tilde{\boldsymbol{\pi}}_i\right). \tag{C.2}$$

By virtue of (C.1) and Property 2.3, the time derivative of V is

$$\begin{aligned}\dot{V} =& \sum_{i=1}^{N}\Big(\boldsymbol{s}_i^T\bar{\boldsymbol{M}}_i\dot{\boldsymbol{s}}_i+\frac{1}{2}\boldsymbol{s}_i^T\dot{\bar{\boldsymbol{M}}}_i\boldsymbol{s}_i+\hat{\Delta}\boldsymbol{u}_{f,i}^T\tilde{\boldsymbol{h}}_{int,i}-\tilde{\boldsymbol{\pi}}_i^T\boldsymbol{K}_\pi\dot{\hat{\boldsymbol{\pi}}}_i\Big)\\ =& \sum_{i=1}^{N}\Big(- \boldsymbol{s}_i^T\boldsymbol{K}_s\boldsymbol{s}_i + \frac{1}{2}(\dot{\bar{\boldsymbol{M}}}_i - 2\bar{\boldsymbol{C}}_i)\boldsymbol{s}_i + \boldsymbol{s}_i^T(\boldsymbol{h}_i - \Delta\boldsymbol{u}_i)\\ &+ \hat{\Delta}\boldsymbol{u}_{f,i}^T\tilde{\boldsymbol{h}}_{int,i} - \tilde{\boldsymbol{\pi}}_i^T(\boldsymbol{K}_\pi\dot{\hat{\boldsymbol{\pi}}}_i - \bar{\boldsymbol{Y}}_i^T(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i, \boldsymbol{\rho}_i, \dot{\boldsymbol{\rho}}_i)\boldsymbol{s}_i)\Big)\\ =& \sum_{i=1}^{N}\Big(-\boldsymbol{s}_i^T\boldsymbol{K}_s\boldsymbol{s}_i+\hat{\Delta}\boldsymbol{u}_{f,i}^T\tilde{\boldsymbol{h}}_{int,i}-\boldsymbol{s}_i^T(\boldsymbol{e}_{int,i} + k_f\hat{\Delta}\boldsymbol{u}_{f,i} + \frac{1}{N}{}^i\tilde{\boldsymbol{h}}_h)\\ &- \kappa_i(t)\boldsymbol{s}_i^T\boldsymbol{s}_i - \tilde{\boldsymbol{\pi}}_i^T(\boldsymbol{K}_\pi\dot{\hat{\boldsymbol{\pi}}}_i - \bar{\boldsymbol{Y}}_i^T(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i, \boldsymbol{\rho}_i, \dot{\boldsymbol{\rho}}_i)\boldsymbol{s}_i)\Big).\end{aligned} \tag{C.3}$$

Given the parameters update law in (4.45), (C.3) simplifies to

$$\begin{aligned}\dot{V} =& \sum_{i=1}^{N}\Big(- \boldsymbol{s}_i^T\boldsymbol{K}_s\boldsymbol{s}_i + \hat{\Delta}\boldsymbol{u}_{f,i}^T\tilde{\boldsymbol{h}}_{int,i} - \kappa_i(t)\|\boldsymbol{s}_i\|^2\\ &- (\tilde{\boldsymbol{\zeta}}_i + \hat{\Delta}\boldsymbol{u}_{f,i})^T(\tilde{\boldsymbol{h}}_{int,i} + \hat{\Delta}\boldsymbol{u}_{f,i}) + \frac{1}{N}\boldsymbol{s}_i^{T\,i}\tilde{\boldsymbol{h}}_h\Big).\end{aligned} \tag{C.4}$$

Thus, by choosing

$$\kappa_i(t) > \frac{\|\tilde{\boldsymbol{\zeta}}_i\|}{\|\boldsymbol{s}_i\|^2}\|\tilde{\boldsymbol{h}}_{int,i} + \hat{\Delta}\boldsymbol{u}_{f,i}\|$$

it holds

$$\dot{V} \leq \sum_{i=1}^{N}\big(- \boldsymbol{s}_i^T\boldsymbol{K}_s\boldsymbol{s}_i - \hat{\Delta}\boldsymbol{u}_{f,i}^T\hat{\Delta}\boldsymbol{u}_{f,i} + \frac{1}{N}\|\boldsymbol{s}_i^{T\,i}\|\|\tilde{\boldsymbol{h}}_h\|\big)$$

From Lemma 4.2, $\|\tilde{\boldsymbol{h}}_h\|$ converges to the origin after a finite time T_h; then after this time it holds

$$\dot{V} \leq \sum_{i=1}^{N}\big(- \boldsymbol{s}_i^T\boldsymbol{K}_s\boldsymbol{s}_i - {}^i\hat{\Delta}\boldsymbol{u}_{f,i}^T\,{}^i\hat{\Delta}\boldsymbol{u}_{f,i}\big)$$

which implies that $\dot{V}$ is semi-negative definite and, consequently, that V is bounded. By leveraging the boundedness of V and, then, of $\boldsymbol{s}_i$, $\hat{\Delta}\boldsymbol{u}_{f,i}$ and $\tilde{\boldsymbol{\pi}}_i$, it can be easily shown the $\ddot{V}$ is bounded as well. Thus, by virtue of Barbalat's lemma, $\dot{V}$ is uniformly continuous and converges to the origin, as well as $\boldsymbol{s}_i$ and $\hat{\Delta}\boldsymbol{u}_{f,i} = k_f \int_{t_0}^{t} \tilde{\boldsymbol{h}}_{int,i}\, d\tau$ (and, therefore, by definition $\tilde{\boldsymbol{h}}_{int,i}$). The main implication of the latter is that, since because of Lemma 4.2, $\tilde{\boldsymbol{h}}_{int,i}$ converges to the origin in finite time (that is the internal wrenches estimated via observer converges to the real one), then also $\boldsymbol{e}_{int,i}$, $\forall\, i$, converges to the origin.

In view of the expression of $\boldsymbol{s}_i$ in (4.41) and since $\hat{\Delta}\boldsymbol{u}_{f,i}$ converges to the origin, it follows

$$\dot{\boldsymbol{e}}_{x,i} + k_p \boldsymbol{e}_{x,i} = -\hat{\Delta}\boldsymbol{u}_{f,i}$$

which represents an asymptotically stable system (in the state variable $\boldsymbol{e}_{x,i}$) with vanishing input $-\hat{\Delta}\boldsymbol{u}_{f,i}$. Therefore, $\boldsymbol{e}_{x,i}$ asymptotically converges to the origin. Based on the expression of $\boldsymbol{e}_{x,i}$ in (4.40), it asymptotically holds

$$({}^{i}\hat{\boldsymbol{x}}_v - \boldsymbol{x}_i) + k_c \int_{t_0}^{t} \sum_{j \in \mathcal{N}_i} (\boldsymbol{x}_j - \boldsymbol{x}_i) d\tau \rightarrow \boldsymbol{0}_p. \tag{C.5}$$

Let ${}^{i}\tilde{\boldsymbol{x}}_v = \boldsymbol{x}_v - {}^{i}\hat{\boldsymbol{x}}_v \in \mathbb{R}^p$ denote the object trajectory estimate error and $\boldsymbol{e}_{v,i} = \boldsymbol{x}_v - \boldsymbol{x}_i \in \mathbb{R}^p$ the object trajectory tracking error. From (C.5), it asymptotically holds

$$(\boldsymbol{x}_v - \boldsymbol{x}_i) + k_c \int_{t_0}^{t} \sum_{j \in \mathcal{N}_i} (\boldsymbol{x}_j - \boldsymbol{x}_v - \boldsymbol{x}_i + \boldsymbol{x}_v) d\tau = -{}^{i}\tilde{\boldsymbol{x}}_v$$

which can be rewritten as

$$\boldsymbol{e}_{v,i} + k_c \int_{t_0}^{t} \sum_{j \in \mathcal{N}_i} (-\boldsymbol{e}_{v,j} + \boldsymbol{e}_{v,i}) d\tau = -{}^{i}\tilde{\boldsymbol{x}}_v. \tag{C.6}$$

By denoting with $\tilde{\boldsymbol{x}}_v \in \mathbb{R}^{Np}$ and $\boldsymbol{e}_v \in \mathbb{R}^{Np}$ the stacked vectors of the errors ${}^{i}\tilde{\boldsymbol{x}}_v$ and $\boldsymbol{e}_{v,i}$, respectively, (C.6) leads to

$$\boldsymbol{e}_v(t) = -k_c(\boldsymbol{L} \otimes \boldsymbol{I}_p) \int_{t_0}^{t} \boldsymbol{e}_v \, d\tau - \tilde{\boldsymbol{x}}_v \tag{C.7}$$

in which, from (4.38), $\tilde{\boldsymbol{x}}_v$ converges to the origin in finite-time. Finally, since the communication graph is connected, the immediate consequence of (C.7) is that $\int_{t_0}^{t} \boldsymbol{e}_{v,i} = \int_{t_0}^{t} \boldsymbol{e}_{v,j}$, $\forall\, i, j$ which, based on (4.40) and (C.5), implies that $\boldsymbol{e}_{v,i} = \boldsymbol{0}_p \;\forall\, i$. This completes the proof.

C.2 Complements to Lemma (4.4)

In this section, the expression of the coefficients $\bar{\boldsymbol{\Omega}}_1$, $\bar{\boldsymbol{\Omega}}_2$, $\bar{\boldsymbol{\omega}}_3$ and $\boldsymbol{\Omega}_1$, $\boldsymbol{\Omega}_2$, $\boldsymbol{\omega}_3$ in (4.48) of Lemma (4.4) is derived. In particular, by virtue of Assumption 4.1 and by considering virtual model tracking, from (4.1) it follows

$$\boldsymbol{x}_v = -\boldsymbol{K}_h^{-1}(\boldsymbol{h}_{h,d} + \boldsymbol{e}_h) + \boldsymbol{x}_{h,d} \tag{C.8}$$

whose time derivatives, by considering that a set-point desired wrench is assigned, i.e., $\dot{\boldsymbol{h}}_{h,d} = \dot{\boldsymbol{h}}_{h,d} = \boldsymbol{0}_p$, are

$$\begin{aligned} \dot{\boldsymbol{x}}_v &= \boldsymbol{K}_h^{-2}\dot{\boldsymbol{K}}_h(\boldsymbol{h}_{h,d} + \boldsymbol{e}_h) - \boldsymbol{K}_h^{-1}\dot{\boldsymbol{e}}_h + \dot{\boldsymbol{x}}_{h,d} \\ \ddot{\boldsymbol{x}}_v &= \boldsymbol{K}_h^{-2}(\ddot{\boldsymbol{K}}_h - 2\boldsymbol{K}_h^{-1}\dot{\boldsymbol{K}}_h^2)(\boldsymbol{h}_{h,d} + \boldsymbol{e}_h) + 2\boldsymbol{K}_h^{-2}\dot{\boldsymbol{K}}_h\dot{\boldsymbol{e}}_h \\ &\quad - \boldsymbol{K}_h^{-1}\ddot{\boldsymbol{e}}_h + \ddot{\boldsymbol{x}}_{h,d}. \end{aligned} \tag{C.9}$$

By replacing (C.8) and (C.9) into (4.46) and by considering $\boldsymbol{K}_h$ diagonal, the force error dynamics is obtained

$$\begin{aligned} &\ddot{\boldsymbol{e}}_h + (\boldsymbol{M}_v^{-1}\boldsymbol{D}_v - 2\boldsymbol{K}_h^{-1}\dot{\boldsymbol{K}}_h)\dot{\boldsymbol{e}}_h + (\boldsymbol{M}_v^{-1}\boldsymbol{K}_v + \boldsymbol{M}_v^{-1}\boldsymbol{K}_h \\ &-\boldsymbol{K}_h^{-1}\ddot{\boldsymbol{K}}_h + 2\boldsymbol{K}_h^{-2}\dot{\boldsymbol{K}}_h^2 - \boldsymbol{M}_v^{-1}\boldsymbol{D}_v\boldsymbol{K}_h^{-1}\dot{\boldsymbol{K}}_h)\boldsymbol{e}_h = -\boldsymbol{M}_v^{-1}\boldsymbol{K}_h\boldsymbol{u}_h \\ &+(\boldsymbol{K}_h^{-1}\ddot{\boldsymbol{K}}_h\boldsymbol{h}_{h,d} - 2\boldsymbol{K}_h^{-2}\dot{\boldsymbol{K}}_h^2\boldsymbol{h}_{h,d} + \boldsymbol{K}_h\ddot{\boldsymbol{x}}_{h,d} + \boldsymbol{M}_v^{-1}\boldsymbol{K}_h\boldsymbol{D}_v\dot{\boldsymbol{x}}_{h,d} \\ &+ \boldsymbol{M}_v^{-1}\boldsymbol{D}_v\boldsymbol{K}_h^{-1}\dot{\boldsymbol{K}}_h\boldsymbol{h}_{h,d} - \boldsymbol{M}_v^{-1}\boldsymbol{K}_v\boldsymbol{h}_{h,d} + \boldsymbol{M}_v^{-1}\boldsymbol{K}_h\boldsymbol{K}_v\boldsymbol{x}_{h,d}) \end{aligned} \tag{C.10}$$

which can be finally expressed as

$$\begin{aligned}\ddot{\boldsymbol{e}}_h + \bar{\boldsymbol{\Omega}}_1\dot{\boldsymbol{e}}_h + \bar{\boldsymbol{\Omega}}_2\boldsymbol{e}_h =& -\boldsymbol{\Omega}_1(t)\dot{\boldsymbol{e}}_h - \boldsymbol{\Omega}_2(t)\boldsymbol{e}_h + \boldsymbol{\omega}_3(t) + \bar{\boldsymbol{\omega}}_3 \\ & - \boldsymbol{M}_v^{-1}\boldsymbol{K}_h\boldsymbol{u}_h\end{aligned} \tag{C.11}$$

where

$$\begin{aligned}\bar{\boldsymbol{\Omega}}_1 =& \boldsymbol{M}_v^{-1}\boldsymbol{D}_v, \qquad \bar{\boldsymbol{\Omega}}_2 = \boldsymbol{M}_v^{-1}\boldsymbol{K}_v, \\ \bar{\boldsymbol{\omega}}_3 =& -\boldsymbol{M}_v^{-1}\boldsymbol{K}_v\boldsymbol{h}_{h,d}, \quad \boldsymbol{\Omega}_1 = -2\boldsymbol{K}_h^{-1}\dot{\boldsymbol{K}}_h, \\ \boldsymbol{\Omega}_2 =& \boldsymbol{M}_v^{-1}\boldsymbol{K}_h - \boldsymbol{K}_h^{-1}\ddot{\boldsymbol{K}}_h + 2\boldsymbol{K}_h^{-2}\dot{\boldsymbol{K}}_h^2 - \boldsymbol{M}_v^{-1}\boldsymbol{D}_v\boldsymbol{K}_h^{-1}\dot{\boldsymbol{K}}_h \\ \boldsymbol{\omega}_3 =& \boldsymbol{K}_h^{-1}\ddot{\boldsymbol{K}}_h\boldsymbol{h}_{h,d} - 2\boldsymbol{K}_h^{-2}\dot{\boldsymbol{K}}_h^2\boldsymbol{h}_{h,d} + \boldsymbol{M}_v^{-1}\boldsymbol{D}_v\boldsymbol{K}_h^{-1}\dot{\boldsymbol{K}}_h\boldsymbol{h}_{h,d} \\ & + \boldsymbol{K}_h\ddot{\boldsymbol{x}}_{h,d} + \boldsymbol{M}_v^{-1}\boldsymbol{K}_h\boldsymbol{D}_v\dot{\boldsymbol{x}}_{h,d} + \boldsymbol{M}_v^{-1}\boldsymbol{K}_h\boldsymbol{K}_v\boldsymbol{x}_{h,d}.\end{aligned} \tag{C.12}$$

This completes the proof.

Appendix D

Benchmark for bimanual cloth manipulation

A *benchmark* is a set of well-defined tasks to be performed in a standardized setup which needs to be easy to reproduce in different robotics laboratories. This allows to effectively evaluate and compare robotics methods [178].

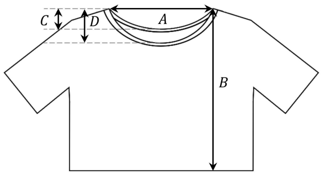
In this appendix, based on [155], three benchmarks are proposed relative to the following basic tasks in cloth manipulation, involving textile objects of different sizes and types: *(i)* spreading a tablecloth, which is preparatory for tasks like ironing or folding, *(ii)* folding a towel, which is a preparatory action before placing the cloth on a shelf or in a box for storage/packaging, *(iii)* partial dressing, i.e., without considering the sleeves, which forms the basis for more complex tasks like putting a T-shirt or a sweater on a human or mannequin. Protocols (RAL-SI-2020-P19-0832_1-V1.0, RAL-SI-2020-P19-0832_2-V1.0, RAL-SI-2020-P19-0832_3-V1.0 for the three tasks, respectively) can be found at the link[1] with their respective explanatory videos and benchmark documents. In the following, a summary of the benchmarks tasks, setup and evaluation is provided. For the sake of clearness, setup, sub-task decomposition and evaluation are common to three benchmarks and will thus be presented jointly. Further information can also be found

[1] https://ral-si.github.io/cloth-benchmark/

Code	Objects for manipulation	
	Tablecloth	IKEA Fullkomlig 1.45 × 2.4 m
[st]	Small towel	IKEA towel Hären 0.3 × 0.5 m or 0.3 × 0.3 m
[bt]	Big towel	IKEA towel Hären 0.5 × 1 m or 0.4 × 0.7 m
	T-Shirt	Any T-shirt in accordance to Figure D.1

Code	Environmental objects	
	Table	Any table with dimension in the range Length: $[1.2, 1.85]$ m Width: $[0.7, 0.8]$ m Height: $[0.72, 0.75]$ m
[sh]	Small head	Generate 3D model with provided script
[bh]	Big head	Generate 3D model with provided script

Table D.1: List of objects with instructions for acquisition



	T-shirt measures
A	[0.13, 0.25] m
B	0.5 m
C	[0.015, 0.05] m
D	[0.07, 0.13] m

Figure D.1: Representation of the allowed measures for the T-shirt; B measure is fixed to equalize the level of difficulty when performing the dressing.

on the website[1].

D.1 Tasks description

Task 1: Spreading a tablecloth

This task consists of grasping a tablecloth and spreading it on a table, using the table and the tablecloth indicated in Table D.1. An example of implementation is shown in Figure D.2-left. Similarly to the other tasks, this task requires to grasp the cloth at two grasping points, usually two of the corners, and then to manipulate it. For the first grasp, different starting cloth configurations are considered: from folded to crumpled on the table (see Figure D.3). Note that the large dimension of the cloth to manipulate repre-

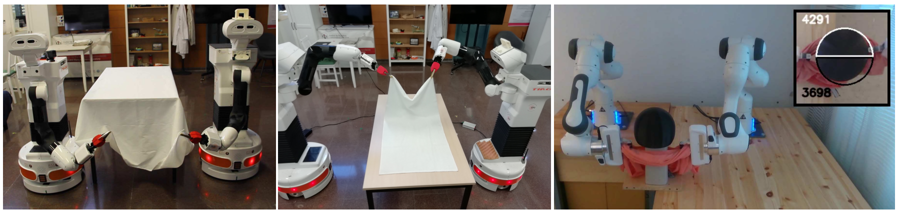


Figure D.2: From left to right, example of implementations of tasks one to three, respectively.

sents a critical issue for many existing solutions and . may call for additional strategies. No specific execution strategy is thus defined in the protocol to give more freedom to researchers to develop and compare innovative approaches.

Task 2: Folding a towel

This task consists of grasping a towel and folding it. The task uses the same table as the previous task and two different sizes of towels, as indicated in Table D.1. An example of implementation is shown in Figure D.2-middle. The following folding is considered: always fold in half and perform a maximum of three folds which are evaluate individually. This strategy has the advantage that can be easily evaluated by taking top view snapshots after every fold. Besides the starting crumpled configuration (as in Task 1), also the “flat on the table” configuration is considered, also shown in Figure D.3. When the cloth is crumpled, the main difference for grasping, compared to Task 1, is the size of the object. A small/medium versus large size would entail the need for different strategies to enable initial grasping.

Task 3: Partial dressing

The goal of the third task is to put a T-shirt over a simple head model starting from different initial configurations of the garment, as shown in Figure D.2-right. Putting on the sleeves is not included in the task. The complex geometric shape of T-shirts makes their manipulation towards desired states a difficult process that requires a tight integration of perceptive sensors, such as cameras and force/torque sensors, into the manipulation strategy. Analogously to the previous tasks, the success of the manipulation task highly depends on the way the garment is grasped, therefore several initial configurations of the T-shirt are considered which allow to explore different grasping strategies: crumpled, flat or folded on the table. Another important aspect is the relative size of the head with respect to the collar circumference: the larger the head is, the harder it is to execute the task. For this reason, head models of two different sizes are provided as reported in Table D.1.

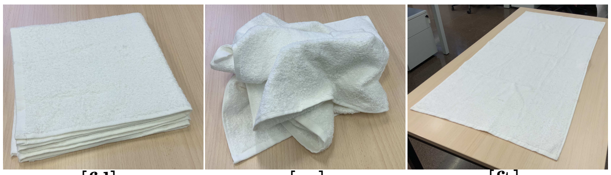
D.2 Setup description

Any bimanual setup with grasping capabilities can be employed and any sensor that can aid in completing the task is allowed.

Objects description

Table D.1 lists all the objects involved in the tasks with the link or information to acquire them. A *range* of measures that are accepted for the T-shirt is defined and reported in Figure D.1.In this way, greater flexibility is guaranteed compared to the case of a predefined single T-shirt and the possibility of adopting the benchmark is maintained despite continuous changes in fashion. The length of the T-shirt (measure B in Figure D.1) is fixed to allow comparability of different methods, since it determines the amount of garment that needs to pass through the head. Finally, concerning the towel for Task 2, two sizes are included, a small towel ([**st**]) and a bigger one ([**bt**]), where the latter is a step forward in the literature in terms of object size.

In addition, two environmental objects are required, a table (for Tasks 1 and 2) and a human-like head (for Task 3). Following the idea of flexibility to make the setup easy to reproduce, lightning conditions are not fixed and no specific table model is indicated but just an interval of table sizes. Concerning the human-like head, two different sizes are considered which are small ([**sh**]) and big ([**bh**]) and their models are defined according to the T-shirt measures.



[**fd**] [**cr**] [**ft**]

Figure D.3: Examples of starting configurations with a towel: [**fd**]: folded on the left, [**cr**]: crumpled on the middle and [**ft**]: flat on the right.

Initial cloth configuration descriptions

In general, when a task on cloth manipulation is attempted, the initial state of the cloth falls in one of these categories:

[**pg2**] Cloth is pre-grasped at two points.

[**pg1**] Cloth is pre-grasped at one point.

[**ft**] Cloth is lying flat on a table (Figure D.3-right).

[**fd**] Cloth is folded on a table (Figure D.3-left).

[**cr**] Cloth is crumpled on a table (Figure D.3-middle).

These starting configurations will be common for all the protocols benchmarking each task, although not all starting configurations are used for all task. For instance, it is pointless to consider the folded configuration for the folding task.

The parts of the cloth that need to be grasped will be referred to as grasping points, e.g., in a towel, these are usually the corners, instead, for a T-shirt, these strictly depend on the manipulation strategy.

D.3 Sub-Tasks description

Given a task, the respective sub-tasks are obtained by considering all the possible combinations of involved objects and initial cloth configurations: a tablecloth with 4 initial configurations for Task 1, two towels with 4 initial configurations for Task 2 and two head sizes with 5 initial configurations for Task 3.

In addition, each sub-task is decomposed in the following phases:

[**GR1**] Grasp first grasping point.

[**GR2**] Grasp second grasping point with other hand.

[**MAN**] Perform the manipulation (depending on the task).

Note that both [**GR1**] and [**GR2**] may require manipulation; for instance, in order to grasp a crumpled cloth from a table and reach the first grasping point, the cloth may need to be pre-manipulated, and all this actions constitute the [**GR1**] phase. Obviously, no [**GR1**] and [**GR2**] phases are required in case of starting configuration [**pg2**] as well as no [**GR1**] phase is executed for the initial configuration [**pg1**].

Users can submit all phases of one sub-task, or just one phase alone. This subdivision in phases and sub-tasks allows to achieve incremental complexity, letting the user choose the desired level of difficulty to face, e.g., dressing task with small head and [**pg2**] initial configuration is clearly less challenging than the case of big head with [**fd**] initial configuration.

D.4 Evaluation of results

To enhance progress, allow reproducibility of results and easy comparison between different works, the following list of performance metrics is proposed: *success of each phase*, *execution time*, *force measures* (if available) and *quality measures*. The choice not to provide a single value to assess quality but a set of values is motivated by the fact that, in such complex tasks, the former may be too reductive; in this way, instead, each user can focus on the aspects of interest, e.g solutions that require longer time but exert lower forces. In the following, the proposed performance metrics are detailed.

Success of each phase

In light of the phases subdivision in Section D.3, each phase can be evaluated individually in regards to completeness. Phases [**GR1**] and [**GR2**] are considered successfully completed if the grasping is performed and is held during the whole manipulation and, in Tasks 1 and 3, if the cloth is unfolded with starting configuration [**fd**]. The condition of success for phase [**MAN**] depends on the considered task: the tablecloth is successfully spread if it covers the table top; the folded towel is successfully folded if one fold is done and opposing corners are together (each fold is evaluated individually); for the dressing task, this phase is assumed to be accomplished when the neck hole of the T-shirt is put over the head and the entirety of the T-shirt lies below the head. Users can choose different grasping points according to their strategy, not to limit the possible approaches in the [**MAN**] phase. In case the manipulation phase is successful, the grasping phases will be considered successful in turn. To increase the flexibility of the benchmarks and promote participation, users can also report only the manipulation part [**MAN**], which may be the case for end-to-end learning-based approaches, or only the grasping part [**GR1**] and [**GR2**] if the group is strong in grasping but lacks the perception solutions to successfully execute the manipulation.

Execution time

The execution time comprises the times needed for all the phases, and it is measured from the moment the first robot starts to move until the end of the manipulation.

Force measures Force measures at the end effectors quantify the interaction between the robots and the environment; they are only acquired during phase [**MAN**] and minimum, maximum and average norms are considered. Note that, in order not to limit the possibility of using the benchmarks, force measures are not mandatory but are highly encouraged especially in dressing task, where monitoring of exerted forces on the head represents a key feature.

Quality measures

For the tasks of tablecloth spreading and folding, the quality of the result of the execution can be measured, e.g., poor results are achieved if the tablecloth is completely tilted or if towels are folded wrinkly or with the corners not matching. To take that into account, a quality function is defined that measures the percentage of error of the task result. Note that for the dressing task, no measures can be defined because of the binary nature of the task.

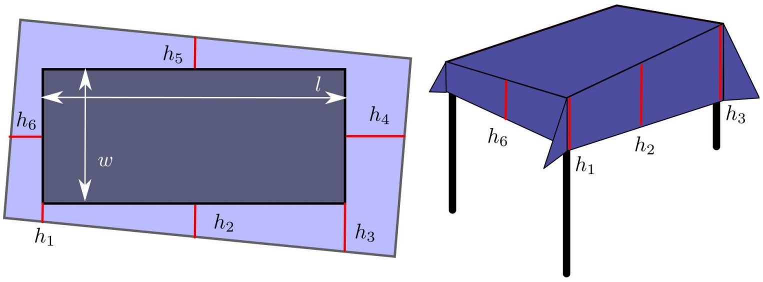


Figure D.4: Representation of the measures to evaluate how well the tablecloth has been placed.

Quality measures for Task 1

Rotation and translation of tablecloth with respect to the table are evaluated. To this aim, as represented in Figure D.4, a total of 6 tablecloth drop lengths at different sides of the table need to be measured after the tablecloth is spread. Measures can be taken from the middle of each table edge. For a table with length t_l and width t_w and a tablecloth with length c_l and width c_w, the proposed percentage of errors are:

$$\begin{aligned} \text{\% rotation error: } E_\alpha &= \frac{\arctan(\frac{|h_3-h_1|}{t_l})}{\pi/4} \\ \text{\% length translation error: } E_l &= \frac{|h_6-h_4|}{c_l-t_l} \\ \text{\% width translation error: } E_w &= \frac{|h_2-h_5|}{c_w-t_w} \end{aligned} \tag{D.1}$$

These quality functions can only be applied if the task has been successfully accomplished, meaning that the tablecloth is covering all the table top. Then, a 100% rotation error occurs when the tablecloth is rotated by $\pi/4$ radians, which is unlikely to happen if the tablecloth is fully covering the table. The maximum translation error occurs when one of the hanging parts is zero, meaning the table is almost uncovered. If the hanging part of the table cloth is touching the floor, one needs to measure the tablecloth drop length ignoring the floor. Note that this error measure is independent of the size of the table and tablecloth, thus allowing a fair comparison among different setups.
Quality measures for Task 2
A one fold manipulation is considered successful if the corners of the original spread cloth are matching two by two. This implies that if one of the corners is bent, the robot should correct it, otherwise the task cannot be reported as a success. In addition, the corners matching is measured by evaluating the ratio between the surface of the spread cloth before and after the fold. Then, the proposed quality function for this task is

$$\%\text{ of error in a fold } E_f = \frac{100}{0.5} \cdot \|\frac{A_f}{A_i} - 0.5\|, \tag{D.2}$$

where A_f is the final area of the cloth from the top view, and A_i is the initial area of the cloth. Assuming A_f will always be smaller than A_i, 100% error occurs when $A_i = A_f$, but also if A_f is less than half of A_i, which can only happen if there are wrinkles or extra folds.

D.4.1 Reporting results

Based on the above, it is required that, for each sub-task, five trials are performed and then, for each trial, measures in Section D.4 are acquired. In addition, videos of the experiments and snapshots (or equivalent stylized figures) clearly representing the grasping points must be provided. A summary table must be filled where, given a starting configuration, the success rate of each phase and average and variance of execution time, force measures and quality functions over the five trials are reported. When necessary, the size of the different elements must be reported as well, that are the table size for Tasks 1 and 2, the towel size for Task 2, and the head size for Task 3. Note that, in the folding task, results associated with each fold must be provided and top view pictures of each fold state have to be reported. Moreover, in order to assess the generality of the proposed approach, it is required to specify which assumptions (in a set in the respective scoring sheet) are made for completing the task, e.g., knowledge of the cloth color and pattern. In the case new assumptions are considered with respect to those in the scoring sheets, a detailed description on how they affect the solution must be reported. Finally, a discussion on:

- Employed hardware/software setup with specification of robots' details and respective number of motors;
- What makes the system successful;
- What makes the system fail;
- What is improved compared to other methods;

should be provided. A thorough description for the scoring of each task can be found in the provided Benchmark documents[1]. Finally, baseline solutions to the presented tasks and their respective evaluations can be found in [155] and on the website[1].

Bibliography

[1] M. Hermann, T. Pentek, and B. Otto, “Design principles for industrie 4.0 scenarios,” in *Hawaii Int. Conf. on System Sciences*, 2016, pp. 3928–3937.

[2] 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 *Int. Conf. on Computers and Industrial Engineering*, 2016, pp. 29–31.

[3] A. Vysocky and P. Novak, “Human-robot collaboration in industry,” *MM Science Journal*, vol. 9, no. 2, pp. 903–906, 2016.

[4] P. Tsarouchi, S. Makris, and G. Chryssolouris, “Human–robot interaction review and challenges on task planning and programming,” *Int. J. Comput. Integr. Manuf.*, vol. 29, no. 8, p. 916–931, Aug. 2016.

[5] V. Villani, F. Pini, F. Leali, and C. Secchi, “Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications,” *Mechatronics*, vol. 55, pp. 248 – 266, 2018.

[6] S. Thrun, “Toward a framework for human-robot interaction,” *Human–Computer Interaction*, vol. 19, no. 1-2, pp. 9–24, 2004.

[7] R. R. Murphy, “Human-robot interaction in rescue robotics,” *IEEE Trans. on Syst., Man and Cyber., Part C (Applications and Reviews)*, vol. 34, no. 2, pp. 138–153, 2004.

[8] E. Mielke, E. Townsend, D. Wingate, and M. D. Killpack, "Human-robot co-manipulation of extended objects: Data-driven models and control from analysis of human-human dyads," *arXiv preprint arXiv:2001.00991*, 2020.

[9] 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, no. 5, pp. 957–975, 2018.

[10] "Robots and robotic devices – safety requirements for industrial robots – Part 1: Robots," Int. Organization for Standardization, Standard ISO 10218-1:2011, 2011.

[11] "Robots and robotic devices – safety requirements for industrial robots – Part 2: Robot systems and integration," Int. Organization for Standardization, Standard ISO 10218-2:2011, 2011.

[12] "Robots and robotic devices – collaborative robots," Int. Organization for Standardization, Standard ISO/TS 15066:2016, 2016.

[13] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[14] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, and Y. Mezouar, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey," *Int. J. Robot. Res.*, vol. 37, no. 7, pp. 688–716, 2018.

[15] P. A. Lasota, T. Fong, J. A. Shah *et al.*, "A survey of methods for safe human-robot interaction," *Foundations and Trends® in Robotics*, vol. 5, no. 4, pp. 261–349, 2017.

[16] M. Lippi and A. Marino, "Human multi-robot safe interaction: A trajectory scaling approach based on safety assessment," *IEEE Trans. Control Syst. Technol.*, pp. 1–16, 2020.

[17] A. M. Zanchettin, N. M. Ceriani, P. Rocco, H. Ding, and B. Matthias, "Safety in human-robot collaborative manufacturing environments: Metrics and control," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 882–893, 2016.

[18] A. Casalino, A. M. Zanchettin, L. Piroddi, and P. Rocco, "Optimal scheduling of human-robot collaborative assembly operations with time petri nets," *IEEE Trans. Autom. Sci. and Engineering*, pp. 1–15, 2019.

[19] D. P. Losey, C. G. McDonald, E. Battaglia, and M. K. O'Malley, "A review of intent detection, arbitration, and communication aspects of shared control for physical human–robot interaction," *Applied Mechanics Reviews*, vol. 70, no. 1, p. 010804, 2018.

[20] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, pp. 2035 – 2057, 2006.

[21] C. Passenberg, A. Peer, and M. Buss, "A survey of environment-, operator-, and task-adapted controllers for teleoperation systems," *Mechatronics*, vol. 20, no. 7, pp. 787 – 801, 2010.

[22] A. Franchi, C. Secchi, M. Ryll, H. H. Bulthoff, and P. R. Giordano, "Shared control : Balancing autonomy and human assistance with a group of quadrotor uavs," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 57–68, 2012.

[23] L. Onnasch and E. Roesler, "A taxonomy to structure and analyze human–robot interaction," *Int. J. Social Robot.*, pp. 1–17, 2020.

[24] H. A. Yanco and J. Drury, "Classifying human-robot interaction: an updated taxonomy," in *IEEE Int. Conf. Syst., Man and Cyber.*, vol. 3, 2004, pp. 2841–2846.

[25] Z. Yan, N. Jouandeau, and A. A. Cherif, "A survey and analysis of multi-robot coordination," *Int. J. Advanced Robotic Syst.*, vol. 10, no. 12, p. 399, 2013.

[26] Z. Wang and M. Schwager, "Force-amplifying n-robot transport system (force-ants) for cooperative planar manipulation without communication," *Int. J. Robot. Res.*, vol. 35, no. 13, pp. 1564–1586, 2016.

[27] M. Essers and T. H. J. Vaneker, “Design of a decentralized modular architecture for flexible and extensible production systems,” *Mechatronics*, vol. 30, pp. 160 – 169, 2015.

[28] G. Gillini, M. Lippi, F. Arrichiello, A. Marino, and F. Pierri, “Distributed fault detection and isolation for cooperative mobile manipulators,” in *IEEE Int. Conf. Syst., Man and Cybern.*, 2019, pp. 1701–1707.

[29] ——, “Distributed fault detection and isolation strategy for a team of cooperative mobile manipulators,” in *Fault Diagnosis and Fault-tolerant Control of Robotic and Autonomous Systems*, ser. Control, Robotics & Sensors. Institution of Engineering and Technology (IET), 2020, pp. 143–166.

[30] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control.* Springer Science & Business Media, 2010.

[31] S. Chiaverini, B. Siciliano, and O. Egeland, “Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator,” *IEEE Trans. Control Syst. Techn.*, vol. 2, no. 2, pp. 123–134, 1994.

[32] O. Khatib, “A unified approach for motion and force control of robot manipulators: The operational space formulation,” *IEEE J. Robot. Autom.*, vol. 3, no. 1, pp. 43–53, 1987.

[33] R. Colbaugh, H. Seraji, and K. Glass, “Direct adaptive impedance control of robot manipulators,” *J. Robot. Syst.*, vol. 10, no. 2, pp. 217–248, 1993.

[34] O. Yonghwan and W. Chung, “Disturbance-observer-based motion control of redundant manipulators using inertially decoupled dynamics,” *IEEE/ASME Trans. Mechatronics*, vol. 4, no. 2, pp. 133–146, 1999.

[35] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks.* Princeton University Press, 2010, vol. 33.

[36] W. Ren and R. Beard, *Distributed Consensus in Multi-vehicle Cooperative Control*, ser. Communications and Control Engineering. Berlin: Springer, 2008.

[37] 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.

[38] X. Zhang, M. M. Polycarpou, and T. Parisini, "A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems," *IEEE Trans. Autom. Control*, vol. 47, no. 4, pp. 576–593, 2002.

[39] X. Zhang, T. Parisini, and M. Polycarpou, "Adaptive fault-tolerant control of nonlinear uncertain systems: an information-based diagnostic approach," *IEEE Trans. Autom. Control*, vol. 49, no. 8, pp. 1259–1274, 2004.

[40] L. Liu, Z. Wang, and H. Zhang, "Adaptive fault-tolerant tracking control for mimo discrete-time systems via reinforcement learning algorithm with less learning parameters," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 299–313, 2017.

[41] X. Zhang and Q. Zhang, "Distributed fault diagnosis in a class of interconnected nonlinear uncertain systems," *Int. J. of Control*, vol. 85, no. 11, pp. 1644–1662, 2012.

[42] R. Ferrari, T. Parisini, and M. Polycarpou, "Distributed fault detection and isolation of large-scale discrete-time nonlinear systems: An adaptive approximation approach," *IEEE Trans. Autom. Control*, vol. 57, no. 2, pp. 275–290, 2012.

[43] I. Shames, A. Teixeira, H. Sandberg, and K. Johansson, "Distributed fault detection for interconnected second-order systems," *Automatica*, vol. 47, no. 12, pp. 2757–2764, 2011.

[44] M. Davoodi, K. Khorasani, H. Talebi, and H. Momeni, "Distributed fault detection and isolation filter design for a network of heterogeneous multiagent systems," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 3, pp. 1061–1069, 2014.

[45] M. Davoodi, N. Meskin, and K. Khorasani, "Simultaneous fault detection and consensus control design for a network of multi-agent systems," *Automatica*, vol. 66, pp. 185 – 194, 2016.

[46] M. Khalili, X. Zhang, M. Polycarpou, T. Parisini, and C. Y., "Distributed adaptive fault-tolerant control of uncertain multi-agent systems," *Automatica*, vol. 87, pp. 142 – 151, 2018.

[47] A. R. Mehrabian and K. Khorasani, "Distributed formation recovery control of heterogeneous multiagent euler–lagrange systems subject to network switching and diagnostic imperfections," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 6, pp. 2158–2166, 2016.

[48] G. Chen, Y. Song, and F. L. Lewis, "Distributed fault-tolerant control of networked uncertain euler–lagrange systems under actuator faults," *IEEE Trans. Cyber.*, vol. 47, no. 7, pp. 1706–1718, 2017.

[49] L. Liu and J. Shan, "Distributed formation control of networked euler–lagrange systems with fault diagnosis," *J. of the Franklin Institute*, vol. 352, no. 3, pp. 952 – 973, 2015.

[50] F. Arrichiello, A. Marino, and F. Pierri, "Observer-based decentralized fault detection and isolation strategy for networked multirobot systems," *IEEE Trans. Control Syst. Techn.*, vol. 23, no. 4, pp. 1465–1476, July 2015.

[51] A. Marino, F. Pierri, and F. Arrichiello, "Distributed fault detection isolation and accommodation for homogeneous networked discrete-time linear systems," *IEEE Trans. Autom. Control*, vol. 62, no. 9, pp. 4840–4847, Sept 2017.

[52] C. Lopez-Limon, J. Ruiz-Leon, A. Cervantes-Herrera, and A. Ramirez-Trevino, "Formation and trajectory tracking of discrete-time multi-agent systems using block control," in *IEEE Conf. on Emerging Technologies Factory Automation*, 2013.

[53] G. Antonelli, F. Arrichiello, F. Caccavale, and A. Marino, "Decentralized time-varying formation control for multi-robot systems," *Int. J. Robot. Res.*, vol. 33, pp. 1029–1043, 2014.

[54] H. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, New Jersey: Prentice-Hall, 2002.

[55] G. Antonelli, F. Arrichiello, F. Caccavale, and A. Marino, "A decentralized controller-observer scheme for multi-agent weighted centroid tracking," *IEEE Trans. Autom. Control*, vol. 58, no. 5, pp. 1310 – 1316, 2013.

[56] G. Hu and M. Liu, "The weighted logarithmic matrix norm and bounds of the matrix exponential," *Linear algebra and its applications*, vol. 390, pp. 145–154, 2004.

[57] F. Romero Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer, "Speeded up detection of squared fiducial markers," *Image and Vision Computing*, vol. 76, pp. 38–47, June 2018.

[58] M. Lippi and A. Marino, "Distributed kinematic control and trajectory scaling for multi-manipulator systems in presence of human operators," in *Mediterranean Conf. on Control and Automation*, 2018, pp. 1–9.

[59] M. Lippi and A. Marino, "Safety in human-multi robot collaborative scenarios: a trajectory scaling approach," *IFAC-PapersOnLine*, vol. 51, no. 22, pp. 190 – 196, 2018, iFAC Symposium on Robot Control.

[60] B. Lacevic, P. Rocco, and A. M. Zanchettin, "Safety assessment and control of robotic manipulators using danger field," *IEEE Trans. Robot.*, vol. 29, no. 5, pp. 1257–1270, 2013.

[61] K. Ikuta, H. Ishii, and M. Nokata, "Safety evaluation method of design and control for human-care robots," *Int. J. Robot. Res.*, vol. 22, no. 5, pp. 281–297, 2003.

[62] D. Kulić and E. A. Croft, "Real-time safety for human–robot interaction," *Rob. Auton. Syst.*, vol. 54, no. 1, pp. 1 – 12, 2006.

[63] F. Flacco, T. Kröger, A. D. Luca, and O. Khatib, "A depth space approach to human-robot collision avoidance," in *IEEE Int. Conf. Robot. Autom.*, 2012, pp. 338–345.

[64] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” in *IEEE Int. Conf. Robot. Autom.*, vol. 2, 1985, pp. 500–505.

[65] P. A. Lasota, G. F. Rossano, and J. A. Shah, “Toward safe close-proximity human-robot interaction with standard industrial robots,” in *IEEE Int. Conf. Autom. Sci. Eng.*, 2014, pp. 339–344.

[66] M. Saveriano, F. Hirt, and D. Lee, “Human-aware motion reshaping using dynamical systems,” *Pattern Recogn. Lett.*, vol. 99, pp. 96 – 104, 2017.

[67] A. Dahlin and Y. Karayiannidis, “Adaptive trajectory generation under velocity constraints using dynamical movement primitives,” *IEEE Control Systems Letters*, vol. 4, no. 2, pp. 438–443, 2020.

[68] H. C. Ravichandar and A. P. Dani, “Human intention inference using expectation-maximization algorithm with online model learning,” *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 2, pp. 855–868, 2017.

[69] J. Mainprice and D. Berenson, “Human-robot collaborative manipulation planning using early prediction of human motion,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2013, pp. 299–306.

[70] S. Haddadin, H. Urbanek, S. Parusel, D. Burschka, J. Roßmann, A. Albu-Schäffer, and G. Hirzinger, “Real-time reactive motion generation based on variable attractor dynamics and shaped velocities,” in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2010, pp. 3109–3116.

[71] M. Ye, Q. Zhang, L. Wang, J. Zhu, R. Yang, and J. Gall, *A Survey on Human Motion Analysis from Depth Data.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 149–187.

[72] U. R. Acharya, K. P. Joseph, N. Kannathal, C. M. Lim, and J. S. Suri, “Heart rate variability: a review,” *Medical and Biological Engineering and Computing*, vol. 44, no. 12, pp. 1031–1051, 2006.

[73] O. H. Jafari, D. Mitzel, and B. Leibe, "Real-time rgb-d based people detection and tracking for mobile robots and head-worn cameras," in *IEEE Int. Conf. Robot. Autom.*, 2014, pp. 5636–5643.

[74] B. Siciliano, "A closed-loop inverse kinematic scheme for on-line joint-based robot control," *Robotica*, vol. 8, no. 3, p. 231–243, 1990.

[75] Fan-Tien Cheng, Tsing-Hua Chen, and York-Yih Sun, "Resolving manipulator redundancy under inequality constraints," *IEEE Trans. Robot. Autom.*, vol. 10, no. 1, pp. 65–71, 1994.

[76] Yunong Zhang, Jun Wang, and Yangsheng Xu, "A dual neural network for bi-criteria kinematic control of redundant manipulators," *IEEE Trans. Robot. Autom.*, vol. 18, no. 6, pp. 923–931, 2002.

[77] G. W. Y. Zhao, Z. Duan and G. Chen, "Distributed finite-time tracking for a multi-agent system under a leader with bounded unknown acceleration," *Systems & Control Lett.*, vol. 81, pp. 8–13, 2015.

[78] A. Marino, "Distributed adaptive control of networked cooperative mobile manipulators," *IEEE Trans. Control Systems Technology*, vol. PP, no. 99, pp. 1–15, 2017.

[79] F. Chen, Y. Cao, and W. Ren, "Distributed average tracking of multiple time-varying reference signals with bounded derivatives," *IEEE Trans. Autom. Control*, vol. 57, no. 12, pp. 3169–3174, 2012.

[80] J. Fu and J. Wang, "Observer-based finite-time coordinated tracking for general linear multi-agent systems," *Automatica*, vol. 66, pp. 231–237, 2016.

[81] A. Marino and A. Gasparri, "A distributed framework for k-hop control strategies in large-scale networks based on local interactions," *IEEE Trans. Autom. Control*, 2019.

[82] A. Filippeschi, N. Schmitz, M. Miezal, G. Bleser, E. Ruffaldi, and D. Stricker, "Survey of motion tracking methods based on inertial sensors: A focus on upper limb human motion," *Sensors*, vol. 17, no. 6, pp. 1–40, 2017.

[83] S. Musić and S. Hirche, "Control sharing in human-robot team interaction," *Annual Reviews in Control*, vol. 44, pp. 342 – 354, 2017.

[84] S. Haddadin, A. De Luca, and A. Albu-Schäffer, "Robot collisions: A survey on detection, isolation, and identification," *IEEE Trans. Robot.*, vol. 33, no. 6, pp. 1292–1312, 2017.

[85] N. Hogan, "Impedance Control: An Approach to Manipulation: Part I—Theory," *J. Dynamic Systems, Measurement, and Control*, vol. 107, no. 1, pp. 1–7, 1985.

[86] G. Tonietti, R. Schiavi, and A. Bicchi, "Design and control of a variable stiffness actuator for safe and fast physical human/robot interaction," in *IEEE Int. Conf. Robot. Autom.*, 2005, pp. 526–531.

[87] F. Dimeas and N. Aspragathos, "Online stability in human-robot cooperation with admittance control," *IEEE Trans. Haptics*, vol. 9, no. 2, pp. 267–278, 2016.

[88] C. T. Landi, F. Ferraguti, L. Sabattini, C. Secchi, and C. Fantuzzi, "Admittance control parameter adaptation for physical human-robot interaction," in *IEEE Int. Conf. Robot. Autom.*, 2017, pp. 2911–2916.

[89] T. Tsumugiwa, R. Yokogawa, and K. Hara, "Variable impedance control based on estimation of human arm stiffness for human-robot cooperative calligraphic task," in *IEEE Int. Conf. Robot. Autom.*, vol. 1, 2002, pp. 644–650 vol.1.

[90] Y. Li and S. S. Ge, "Human–robot collaboration based on motion intention estimation," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 3, pp. 1007–1014, 2014.

[91] B. Corteville, E. Aertbelien, H. Bruyninckx, J. De Schutter, and H. Van Brussel, "Human-inspired robot assistant for fast point-to-point movements," in *IEEE Int. Conf. Robot. Autom.*, 2007, pp. 3639–3644.

[92] J. R. Medina, T. Lorenz, and S. Hirche, "Synthesizing anticipatory haptic assistance considering human behavior uncertainty," *IEEE Trans. Robot.*, vol. 31, no. 1, pp. 180–190, 2015.

[93] H. N. Rahimi, I. Howard, and L. Cui, "Neural impedance adaption for assistive human–robot interaction," *Neurocomputing*, vol. 290, pp. 50 – 59, 2018.

[94] C. Masone, M. Mohammadi, P. R. Giordano, and A. Franchi, "Shared planning and control for mobile robots with integral haptic feedback," *Int. J. Robot. Res.*, vol. 37, no. 11, pp. 1395–1420, 2018.

[95] Y. Li, K. P. Tee, W. L. Chan, R. Yan, Y. Chua, and D. K. Limbu, "Continuous role adaptation for human–robot shared control," *IEEE Trans. Robot.*, vol. 31, no. 3, pp. 672–681, 2015.

[96] D. P. Losey and M. K. O'Malley, "Trajectory deformations from physical human–robot interaction," *IEEE Trans. Robot.*, vol. 34, no. 1, pp. 126–138, 2018.

[97] A. Mörtl, M. Lawitzky, A. Kucukyilmaz, M. Sezgin, C. Basdogan, and S. Hirche, "The role of roles: Physical cooperation between humans and robots," *Int. J. Robot. Res.*, vol. 31, no. 13, pp. 1656–1674, 2012.

[98] A. D. Luca and R. Mattone, "Sensorless robot collision detection and hybrid force/motion control," in *IEEE Int. Conf. Robot. Autom.*, 2005, pp. 999–1004.

[99] M. Geravand, F. Flacco, and A. De Luca, "Human-robot physical interaction and collaboration using an industrial robot with a closed control architecture," in *IEEE Int. Conf. Robot. Autom.*, 2013, pp. 4000–4007.

[100] A. Kouris, F. Dimeas, and N. Aspragathos, “A frequency domain approach for contact type distinction in human–robot collaboration,” *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 720–727, 2018.

[101] G. Cheng, E. Dean-Leon, F. Bergner, J. Rogelio Guadarrama Olvera, Q. Leboutet, and P. Mittendorfer, “A comprehensive realization of robot skin: Sensors, sensing, control, and applications,” *Proceedings of the IEEE*, vol. 107, no. 10, pp. 2034–2051, 2019.

[102] A. Albini and G. Cannata, “Pressure distribution classification and segmentation of human hands in contact with the robot body,” *Int. J. Robot. Res.*, vol. 39, no. 6, pp. 668–687, 2020.

[103] A.-N. Sharkawy and N. Aspragathos, “Human-robot collision detection based on neural networks,” *Int. J. Mechanical Engineering and Robot. Res.*, vol. 7, no. 2, pp. 150–157, 2018.

[104] A.-N. Sharkawy, P. N. Koustoumpardis, and N. Aspragathos, “Neural network design for manipulator collision detection based only on the joint position sensors,” *Robotica*, vol. 38, no. 10, pp. 1737–1755, 2020.

[105] Y. J. Heo, D. Kim, W. Lee, H. Kim, J. Park, and W. K. Chung, “Collision detection for industrial collaborative robots: a deep learning approach,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 740–746, 2019.

[106] N. Briquet-Kerestedjian, A. Wahrburg, M. Grossard, M. Makarov, and P. Rodriguez-Ayerbe, “Using neural networks for classifying human-robot contact situations,” in *European Control Conf.*, 2019, pp. 3279–3285.

[107] S. Golz, C. Osendorfer, and S. Haddadin, “Using tactile sensation for learning contact knowledge: Discriminate collision from physical interaction,” in *IEEE Int. Conf. Robot. Autom.*, 2015, pp. 3788–3794.

[108] D. Popov, A. Klimchik, and N. Mavridis, “Collision detection, localization classification for industrial robots with joint torque

sensors," in *IEEE Int. Symp. on Robot and Human Interactive Communication*, 2017, pp. 838–843.

[109] Y. Karayiannidis, L. Droukas, D. Papageorgiou, and Z. Doulgeri, "Robot control for task performance and enhanced safety under impact," *Frontiers in Robotics and AI*, vol. 2, p. 34, 2015.

[110] M. M. Rahman, R. Ikeura, and K. Mizutani, "Investigation of the impedance characteristic of human arm for development of robots to cooperate with humans," *JSME Int. J. Series C Mechanical Syst., Machine Elem. and Manufacturing*, vol. 45, no. 2, pp. 510–518, 2002.

[111] E. Burdet, D. W. Franklin, and T. E. Milner, *Human robotics: neuromechanics and motor control.* MIT press, 2013.

[112] M. S. Erden and A. Billard, "Hand impedance measurements during interactive manual welding with a robot," *IEEE Trans. Robot.*, vol. 31, no. 1, pp. 168–179, 2015.

[113] P. K. Artemiadis, P. T. Katsiaris, M. V. Liarokapis, and K. J. Kyriakopoulos, "Human arm impedance: Characterization and modeling in 3d space," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2010, pp. 3103–3108.

[114] H. Modares, I. Ranatunga, F. L. Lewis, and D. O. Popa, "Optimized assistive human–robot interaction using reinforcement learning," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 655–667, 2016.

[115] 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," *Rob. Auton. Syst.*, vol. 103, pp. 122 – 133, 2018.

[116] D. Cehajic, P. B. G. Dohmann, and S. Hirche, "Estimating unknown object dynamics in human-robot manipulation tasks," in *IEEE Int. Conf. Robot. Autom.*, 2017, pp. 1730–1737.

[117] M. Lippi and A. Marino, "Human multi-robot physical interaction: A distributed framework," *J. Intelligent & Robotic Systems*, vol. 101, no. 35, 2021.

[118] M. Selvaggio, F. Abi-Farraj, C. Pacchierotti, P. R. Giordano, and B. Siciliano, "Haptic-based shared-control methods for a dual-arm system," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 4249–4256, Oct 2018.

[119] T. Yoshikawa and X. Zheng, "Coordinated dynamic hybrid position/force control for multiple robot manipulators handling one constrained object," *Int. J. Robot. Res.*, vol. 12, no. 3, pp. 219–230, 1993.

[120] S. Haykin, *Adaptive Filter Theory.* Prentice-Hall, Inc., 1996.

[121] O. P. Malik, G. S. Hope, and S. J. Cheng, "Some issues on the practical use of recursive least squares identification in self-tuning control," *Int. J. Control*, vol. 53, no. 5, pp. 1021–1033, 1991.

[122] N. R. Sripada and D. G. Fisher, "Improved least squares identification," *Int. J. Control*, vol. 46, no. 6, pp. 1889–1913, 1987.

[123] R. J. Ansari, G. Giordano, J. Sjöberg, and Y. Karayiannidis, "Human grasp position estimation for human–robot cooperative object manipulation," *Robotics and Autonomous Systems*, vol. 131, p. 103600, 2020.

[124] A. Bhole, F. Ficuciello, A. Mashayekhi, S. Strano, M. Terzo, L. Villani, and B. Siciliano, "Online estimation of impedance parameters for a variable impedance controlled robotic manipulator," in *Advances in Italian Mechanism Science.* Springer International Publishing, 2019, pp. 267–274.

[125] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal control.* John Wiley & Sons, 2012.

[126] G. Golluccio, G. Gillini, A. Marino, and G. Antonelli, "Robot dynamics identification: A reproducible comparison with experiments on the kinova jaco2," *IEEE Robot. Autom. Mag.*, 2020.

[127] M. Lippi and A. Marino, "Cooperative object transportation by multiple ground and aerial vehicles: Modeling and planning," in *IEEE Int. Conf. Robot. Autom.*, 2018, pp. 1084–1090.

[128] Y. Li and S. S. Ge, "Force tracking control for motion synchronization in human-robot collaboration," *Robotica*, vol. 34, no. 6, p. 1260–1281, 2016.

[129] M. Lippi, A. Marino, and S. Chiaverini, "A distributed approach to human multi-robot physical interaction," in *IEEE Int. Conf. Syst., Man and Cyber.*, 2019, pp. 728–734.

[130] H. Seraji and R. Colbaugh, "Force tracking in impedance control," *Int. J. Robot. Res.*, vol. 16, no. 1, pp. 97–117, 1997.

[131] N. Fischer, R. Kamalapurkar, and W. E. Dixon, "Lasalle-yoshizawa corollaries for nonsmooth systems," *IEEE Trans. Autom. Control*, vol. 58, no. 9, pp. 2333–2338, 2013.

[132] M. Lippi and A. Marino, "Enabling physical human-robot collaboration through contact classification and reaction," in *IEEE Int. Conf. Robot and Human Interactive Communication*, 2020, pp. 1196–1203.

[133] M. Lippi, G. Gillini, A. Marino, and F. Arrichiello, "A data-driven approach for contact detection, classification and reaction in physical human-robot collaboration," in *IEEE Int. Conf. Robot. Autom.*, 2020.

[134] A. De Luca and F. Flacco, "Integrated control for pHRI: Collision avoidance, detection, reaction and collaboration," in *IEEE RAS EMBS Int. Conf. on Biomedical Robotics and Biomechatronics*, 2012, pp. 288–295.

[135] D. A. Reynolds, "Gaussian mixture models." *Encyclopedia of biometrics*, vol. 741, 2009.

[136] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Review*, vol. 26, no. 2, pp. 195–239, 1984.

[137] S. Calinon, T. Alizadeh, and D. G. Caldwell, "On improving the extrapolation capability of task-parameterized movement models," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2013, pp. 610–616.

[138] Q. Yuan and I.-M. Chen, "Localization and velocity tracking of human via 3 imu sensors," *Sensors and Actuators A: Physical*, vol. 212, pp. 25 – 33, 2014.

[139] A. D. Luca, G. Oriolo, and B. Siciliano, "Robot redundancy resolution at the acceleration level," *Laboratory Robotics and Automation*, vol. 4, pp. 97–106, 1992.

[140] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *IEEE Confer. Decis. Control*, 2014, pp. 6271–6278.

[141] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recogn.*, vol. 47, no. 6, pp. 2280–2292, 2014.

[142] E. A. Basso and K. Y. Pettersen, "Task-priority control of redundant robotic systems using control lyapunov and control barrier function based quadratic programs," 2020, https://arxiv.org/abs/2001.07547.

[143] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, and Y. Mezouar, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey," *Int. J. Robot. Res.*, vol. 37, no. 7, pp. 688–716, 2018.

[144] S. M. LaValle, *Planning Algorithms.* Cambridge U.K.: Cambridge University Press, 2006.

[145] C. Finn and S. Levine, "Deep visual foresight for planning robot motion," in *IEEE Int. Conf. Robot. Autom.*, 2017, pp. 2786–2793.

[146] T. Lesort, N. Díaz-Rodríguez, J.-F. Goudou, and D. Filliat, "State representation learning for control: An overview," *Neural Net.*, vol. 108, pp. 379–392, 2018.

[147] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *Int. Conf. Learn. Represent.*, 2015.

[148] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Int. Conf. Mach. Learn.*, 2014, pp. 1278–1286.

[149] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[150] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville, "Adversarially learned inference," in *Int. Conf. Learn. Represent.*, 2017.

[151] B. Ichter and M. Pavone, "Robot Motion Planning in Learned Latent Spaces," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2407–2414, 2019.

[152] A. Nair, D. Chen, P. Agrawal, P. Isola, P. Abbeel, J. Malik, and S. Levine, "Combining self-supervised learning and imitation for vision-based rope manipulation," in *IEEE Int. Conf. Robot. Autom.*, 2017, pp. 2146–2153.

[153] M. Lippi, P. Poklukar, M. C. Welle, A. Varava, H. Yin, A. Marino, and D. Kragic, "Latent space roadmap for visual action planning of deformable and rigid object manipulation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2020.

[154] ——, "Enabling visual action planning for object manipulation through latent space roadmap," *submitted to IEEE Trans. Robot.*, 2020.

[155] I. Garcia-Camacho, M. Lippi, M. C. Welle, H. Yin, R. Antonova, A. Varava, J. Borras, C. Torras, A. Marino, G. Alenyà, and D. Kragic, "Benchmarking bimanual cloth manipulation," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1111–1118, 2020.

[156] T. Lozano-Pérez and L. P. Kaelbling, "A constraint-based method for solving sequential manipulation planning problems," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. IEEE, 2014, pp. 3684–3691.

[157] L. P. Kaelbling and T. Lozano-Pérez, "Integrated task and motion planning in belief space," *Int. J. Robot. Res.*, vol. 32, no. 9-10, pp. 1194–1227, 2013.

[158] R. Bellman, "Curse of dimensionality," *Adaptive control processes: a guided tour. Princeton, NJ*, vol. 3, p. 2, 1961.

[159] A. Srinivas, A. Jabri, P. Abbeel, S. Levine, and C. Finn, "Universal planning networks," in *Int. Conf. Mach. Learn.*, 2018.

[160] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," *arXiv preprint arXiv:1811.04551*, 2018.

[161] A. Wang, T. Kurutach, P. Abbeel, and A. Tamar, "Learning robotic manipulation through visual planning and acting," in *Robotics: Science and Systems*, 2019.

[162] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recogn.*, vol. 2, 2006, pp. 1735–1742.

[163] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "β-vae: Learning basic visual concepts with a constrained variational framework," *Int. Conf. Learn. Represent.*, 2017.

[164] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, "Understanding disentangling in β-vae," *arXiv preprint arXiv:1804.03599*, 2018.

[165] D. Müllner, "Modern hierarchical, agglomerative clustering algorithms," *arXiv preprint arXiv:1109.2378*, 2011.

[166] R. R. Sokal, "A statistical method for evaluating systematic relationships." *Univ. Kansas, Sci. Bull.*, vol. 38, pp. 1409–1438, 1958.

[167] M. E. Celebi, *Partitional clustering algorithms.* Springer, 2014.

[168] C. C. Aggarwal and C. K. Reddy, "Data clustering," *Algorithms and applications. Chapman&Hall/CRC Data mining and Knowledge Discovery series, Londra*, 2014.

[169] R. P. Brent, "An algorithm with guaranteed convergence for finding a zero of a function," *The Computer Journal*, vol. 14, no. 4, pp. 422–425, 1971.

[170] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring Network Structure, Dynamics, and Function using NetworkX," in *Python in Science Conf.*, 2008, pp. 11–15.

[171] Unity Technologies, "Unity." [Online]. Available: https://unity.com

[172] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. on Computer Vision and Pattern Recogn.*, 2016, pp. 770–778.

[173] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.

[174] C. Burgess and H. Kim, "3d shapes dataset," https://github.com/deepmind/3dshapes-dataset/, 2018.

[175] D. Seita, A. Ganapathi, R. Hoque, M. Hwang, E. Cen, A. K. Tanwani, A. Balakrishna, B. Thananjeyan, J. Ichnowski, N. Jamali *et al.*, "Deep imitation learning of sequential fabric smoothing policies," *arXiv preprint arXiv:1910.04854*, 2019.

[176] B. Settles, *Active learning literature survey.* University of Wisconsin-Madison Department of Computer Sciences, 2009.

[177] H. Roger and R. Charles, *Topics in matrix analysis.* Cambridge: Cambridge University Press, 1994.

[178] F. Bonsignorio and A. P. del Pobil, "Toward replicable and measurable robotics research [from the guest editors]," *IEEE Robot. Autom. Mag.*, vol. 22, no. 3, pp. 32–35, 2015.

Acknowledgment

Now it is time to thank all the people that were part of this journey. Please don't be upset with me but I will switch to Italian for this part.

Un grazie speciale va ad Alessandro, un mentore ed un riferimento costante in questi anni. Ti sarò sempre profondamente grata per tutto ciò che mi hai insegnato, per il tuo assiduo supporto, per la tua dedizione (e, non ultimo, per lo sfrenato ottimismo che ti contraddistingue :P). Ringrazio il Prof. Pasquale Chiacchio per avermi dato la possibilità di conseguire questo titolo ed il Prof. Francesco Basile per aver rallegrato le giornate lavorative con i suoi racconti. Grazie ai miei colleghi che hanno rallegrato con risate e caffettini anche le ore più buie di questo percorso. Grazie ad Emi, per le infinite chiacchierate e per essermi stato sempre vicino (ma, soprattutto, per tutte le cene che mi hai promesso), a Luigi (il Maestro), per aver portato ritmo e leggerezza nelle giornate universitarie, a Enrico, per tutte le conversazioni sulla robotica (e sulle sagre), a Vincenzo, per i tuoi unici (e talvolta inquietanti) modi di scherzare, e ad Antonio e Kateryna, per avermi accolta e fatta sentire una piccola elettrotecnica insieme a loro. Grazie a Giovanni, per il tuo affetto e la tua gentilezza. Mi mancano i pranzi insieme!

Many thanks to all the people who made the experience in Stockholm unforgettable. Thanks to Dani, for giving me the opportunity to join her research group, and to Anastasiia, for being supportive in all the research activities. A special thanks to the folding girls: to Michael, for all the fun work together, the terrible jokes, the pancakes, and the (failed) whiskey lessons, and to Petra, for always bringing the light despite your role as language police. Ed, infine, grazie ad Agnese, per essere stata la migliore compagna di avventure, di kanelbulle e di serate a bere improbabile vino vedendo X Factor.

Un fortissimo grazie al gruppo di Cassino che mi resa parte della loro famiglia dal primo istante, riuscendo (quasi) a farmi apprezzare anche la nebbia di Cassino. Un grazie ai senior Filippo e Gianluca e ai junior Paolo, Daniele e Giacomo. E, poi, un grazie particolare a Giuseppe, per tutte le imprecazioni condivise lavorando con l'MDU e per avermi insegnato la saggia filosofia di Martellone.

Ringrazio, inoltre, il team Roma Tre. Grazie ad Andrea, per la passione per la ricerca che trasmetti, e grazie a Matteo. Nonostante gli svariati tentativi di lavorare insieme siano miseramente falliti, sei diventato uno degli amici più cari che ho e non potrei esserne più felice.

Un grazie gigante alla mia famiglia. Grazie a mamma e papà, per tutto l'amore ed il supporto in questi anni (e per non avermi cacciata di casa nei momenti in cui ero decisamente insopportabile), grazie a Checco, per essere un perfetto fratellone rompiscatole, grazie alla mia nonnina, per essere da sempre un modello di vita, e grazie a nonno Toto e nonna Bionda, che mi proteggono ogni giorno da lassù.

Un immenso grazie ad Alessandro per essere la mia certezza e per avermi sostenuta con amore in ogni momento di questo percorso. Grazie per le tue continue attenzioni e per la tua infinita dolcezza, mi rendi sicura con te al mio fianco.

Grazie ai miei amici di sempre, divenuti la mia seconda famiglia. A Marti, perché sei come una sorella, a Cate, per la tua dolcezza, a Cacco, per le lezioni da super coach, a Simo, per la tua schiettezza, a Bombolo, per gli anni compagne di banco, a Iduccia, per le lezioni di abbigliamento trash, a Carla, per le tue tontolate, e a Dario, per fornire sempre un sottofondo musicale.

Infine, grazie ad ogni persona che non ho menzionato ma che ha contribuito anche solo con un momento di gioia in questi tre anni di dottorato. "Happiness is real only when shared"!