
UNIVERSITÁ DI PISA
DOTTORATO DI RICERCA IN INGEGNERIA DELL'INFORMAZIONE

# ROBOT INTERACTION PLANNING AND CONTROL IN UNSTRUCTURED ENVIRONMENTS: EXPLOITING THE TRADE-OFF BETWEEN COMPLEXITY AND AUTONOMY

DOCTORAL THESIS

Author
**Pietro Balatti**

Tutors
**Dr. Arash Ajoudani**
**Prof. Lucia Pallottino**

Reviewers
**Dr. Fanny Ficuciello**
**Dr. Serena Ivaldi**

The Coordinator of the PhD Program
**Prof. Fulvio Gini**

Pisa, June 2021

Cycle XXXIII

# Acknowledgements

FIRST of all, I would like to express my gratitude to Dr. Arash Ajoudani for his constant and invaluable advice that steered my doctoral studies path always towards the right direction. Without his enthusiastic and valuable support, this Ph.D. thesis would not have been achievable. I feel honoured to have been an active witness of the outstanding growth of the Human-Robot Interfaces and physical Interaction laboratory.

I would also like to thank all the colleagues at the Istituto Italiano di Tecnologia that have lighten up every single day of the last 4 years. Constructive discussions, endless nights before deadlines, and the time spared out of the lab have contributed to solid growth of sincere friendships. A special thought to Marta and Edoardo, for the unforgettable moments we have been sharing during our entire Ph.D. path. Thanks also to Mattia, Dimitri, Fabio, Luca, Wansoo, and Yuqiang for their heartfelt friendship and invaluable help.

Special thanks to my parents, Piera and Ulisse, for their constant support and for always having believed in me. Thank you for passing on the desire to always dream high, and the perseverance to overcome all the difficulties life presents.

Lastly, the biggest thought goes to my beloved life partner, Lisa, for her unconditional love, support and patience throughout good and bad times that characterized these years.

*Pietro Balatti*

*Genova, June 2021*

# Summary

In the classical concept of industrial robotics, robots are deployed in structured environments of manufacturing plants, operating inside industrial cages. In this context, they are usually pre-programmed since they are aware of all their workspace features and properties. These characteristics are not changing over time, and therefore the required level of adaptation is trivial. Nowadays, robots are going beyond this outdated concept, and they are making their way into new unstructured application fields such as agriculture, disaster scenarios, small-batch manufacturing, healthcare, and logistics. These sectors demand an efficient interaction of the robots with human beings and unknown environments. To respond to the high uncertainty levels of such situations, the aim of this thesis resides in the development of novel context-aware and adaptive robotic behaviors able to distinguish expected interactions from external disturbances, to be able to react accordingly and appropriately. The novelty of the proposed methods consists in the development of novel robot interaction planning and control algorithms that can be regulated either by the robot itself (self-governing approach) or by an operator (human-in-the-loop approach).

In view of a lower task complexity, autonomous robotic behaviors should be preferable since they can achieve higher accuracy, perform tasks with faster motions, and also save costs related to the employment of less human resources. Therefore, a novel adaptive impedance controller able to regulate robot quasi-static parameters, i.e., stiffness and damping, will be presented. The tuning of such parameters is based on the concept of *interaction expectancy*. If no interaction with the environment is predicted, the robot maintains a compliant profile, to gently respond to unexpected external perturbations. On the contrary, if an interaction is foreseen, the impedance parameters are regulated, so as to achieve efficient results in the task execution. The self-governing approach will be addressed in Chapter 3.

With the rise of task complexity, autonomous robotic strategies alone can not satisfy adequately the requirements for time and resources efficiency. This is due to the growing environmental uncertainty and the decreasing situational awareness. Nevertheless, the introduction of a human in the loop approach can contribute to the improvement of the global system capabilities, compensating for scarce robot autonomy abilities. To

this end, new thinking and techniques dedicated to applications with high task complexity are introduced. Novel human-robot interfaces are presented for both close- and far-proximity applications, including robot physical guidance and teleoperation. The human-in-the-loop approach will be addressed in Chapter 4.

The results of this thesis illustrate a high potential in improving the robot interaction autonomy, which will enhance the robot capabilities, especially in unknown and unstructured environments. Exploiting the trade-off between complexity and autonomy will lead to a diversification of the strategies to be employed, with the common goal of improving the task execution efficiency. This will bring about the creation of resource-efficient manufacturing solutions, and help to reduce human physical stress, automating repetitive and cognitively unexciting industrial tasks.

# Sommario

TRADIZIONALMENTE, la robotica industriale vede i robot impiegati in ambienti molto strutturati, quali gli impianti manifatturieri, dove operano all'interno di gabbie. In questo contesto, vengono solitamente pre-programmati, avendo la precisa consapevolezza delle caratteristiche dello spazio di lavoro. Queste proprietà non subiscono cambiamenti nel tempo, e dunque il livello di adattamento richiesto è di poco conto. Tuttavia, oggigiorno i robot stanno superando questo concetto obsoleto e si stanno facendo strada verso campi di applicazione meno strutturati come l'agricoltura, gli scenari catastrofici, la produzione in serie di piccoli lotti, l'assistenza sanitaria e la logistica. Questi settori richiedono un'interazione efficiente dei robot con gli essere umani e con ambienti sconosciuti. Per reagire agli altri livelli di incertezza di questi scenari, l'obiettivo di questa tesi risiede nello sviluppo di nuovi comportamenti robotici che siano adattivi e abbiano una consapevolezza del contesto in cui sono impiegati, in modo da essere in grado di distinguere interazione previste da disturbi esterni e di reagire adeguatamente. L'originalità dei metodi proposti consiste nello sviluppo di nuovi algoritmi di controllo e pianificazione dell'interazione robotica che possono essere disciplinati o dai robot stessi (approccio *self-governing*) o da un operatore (approccio *human-in-the-loop*).

In considerazione di mansioni con bassa complessità, comportamenti robotici autonomi sono preferibili in quanto in grado di conseguire una elevata precisione, movimenti più rapidi e anche un risparmio economico dovuto al minor impiego di risorse umane. A questo scopo, sarà presentato un nuovo controllore d'impedenza adattivo, capace di regolare i parametri quasi-statici del robot, ovvero rigidezza e smorzamento. La messa a punto di questi è basato sul concetto di *aspettativa dell'interazione*. Se non si prevede alcuna interazione, il robot mantiene un profilo accondiscendente, per reagire delicatamente a perturbazioni esterne inattese. Al contrario, se ci si aspetta un'interazione, i parametri d'impedenza vengono regolati in modo da raggiungere risultati efficienti nell'esecuzione del lavoro. Questo approccio autogovernativo verrà presentato nel capitolo 3.

Con il crescere della complessità della mansione, le sole strategie robotiche autonome non sono in grado di soddisfare adeguatamente i requisiti per un'efficienza in

termini di tempo e risorse. Questo è causato dall'aumento dell'incertezza ambientale e dalla decrescita della consapevolezza situazionale. Ciò nondimeno, l'introduzione di un essere umano può contribuire a migliorare le capacità globali del sistema, compensando le scarse abilità autonome del robot. A questo fine, saranno presentate nuove teorie e tecniche dedicate ad applicazioni caratterizzate da alta complessità. Saranno introdotte originali interfacce uomo-robot sia per applicazioni con un livello di prossimità ravvicinato sia per quelle a distanza, compresa la guida fisica di robot e la loro teleoperazione. Questo approccio che comprende che affida all'uomo la parte decisionale verrò presentato nel capitolo 4.

I risultati di questa tesi mostrano il grande potenziale che risiede nel miglioramento dell'interazione autonoma da parte dei robot, che aumenterà le loro capacità specialmente in ambienti poco strutturati o sconosciuti. Lo sfruttamento del compromesso tra complessità e autonomia porterà a una diversificazione delle strategie da impiegare, con l'obiettivo comune di aumentare l'efficienza nello svolgimento delle mansioni. Ciò condurrà alla creazione di soluzioni per la produzione manufatturiera più efficienti in termini di risorse, e contribuirà alla riduzione dello stress fisico dei lavoratori, automatizzando mansioni ripetitive e cognitivamente non appaganti.

# List of publications

## International Journals

1. Wu, Y., Balatti, P., Lorenzini, M., Zhao, F., Kim, W., and Ajoudani, A. (2019, October). A teleoperation interface for loco-manipulation control of mobile collaborative robotic assistant. *IEEE Robotics and Automation Letters*. (Vol. 4(4), pp. 3593-3600). IEEE.

2. Balatti, P., Kanoulas, D., Tsagarakis, N., and Ajoudani, A. (2020, July. A method for autonomous robotic manipulation through exploratory interactions with uncertain environments. *Autonomous Robots*. (Vol. 44(8), pp. 1395-1410). Springer.

3. Balatti, P., Fusaro, F., Villa, N., Lamon, E., and Ajoudani, A. (2020, July). A Collaborative Robotic Approach to Autonomous Pallet Jack Transportation and Positioning. *IEEE Access*. (Vol. 8, pp. 142191-142204). IEEE.

4. Balatti, P., Leonori, M., and Ajoudani, A. A Flexible and Collaborative Approach to Robotic Box-Filling and Item Sorting. *Robotics and Autonomous Systems*. (Under review)

## International Conferences/Workshops with Peer Review

1. Balatti, P., Muratore, L., Peternel, L., Hoffman, E. M., Tsagarakis, N. G., and Ajoudani, A (2017, November). A Manipulation Framework for Debris Removal using WALK-MAN Humanoid. In *10th International Workshop on Human-Friendly Robotics*.

2. Balatti, P., Kanoulas, D., Rigano, G. F., Muratore, L., Tsagarakis, N. G., and Ajoudani, A. (2018, October). A self-tuning impedance controller for autonomous robotic manipulation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. (pp. 5885-5891). IEEE.

3. Balatti, P., Kanoulas, D., Tsagarakis, N. G., and Ajoudani, A. (2019, May). Towards robot interaction autonomy: Explore, identify, and interact. In *2019 International Conference on Robotics and Automation (ICRA)* (pp. 9523-9529). IEEE.

4. De Franco, A., Lamon, E., Balatti, P., De Momi, E., and Ajoudani, A. (2019, July). An Intuitive Augmented Reality Interface for Task Scheduling, Monitoring, and Work Performance Improvement in Human-Robot Collaboration. In *2019 IEEE International Work Conference on Bioinspired Intelligence (IWOBI)* (pp. 75-80). IEEE.

5. Kim, W., Balatti, P., Lamon, E., and Ajoudani, A. (2020, May). MOCA-MAN: A MObile and reconfigurable Collaborative Robot Assistant for conjoined huMAN-robot actions. In *2020 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 10191-10197). IEEE.

## Others

1. Balatti, P., De Franco, A., Lamon, E., De Momi, E., and Ajoudani, A. (2019, October) An Augmented Reality Interface for Improving Task Performance in Close-Proximity Teleoperation. *Institute for Robotics and Intelligent Machines Conference (I-RIM)*.

# List of Abbreviations

**A**

| | |
|---|---|
| AR | Augmented Reality. 77, 85–94, 106 |

**C**

| | |
|---|---|
| cobots | collaborative robots. 4, 15, 48, 62, 77, 78, 85, 106 |
| CoP | Center of Pressure. 96, 97, 99, 100, 102, 103, 106 |

**D**

| | |
|---|---|
| DoF | Degrees of Freedom. 10, 12, 14, 15, 17, 18, 70, 71, 95, 119 |
| DRC | DARPA Robotics Challenge. 18, 27 |

**E**

| | |
|---|---|
| EKF | Extended Kalman Filter. 17 |
| EMG | electromyography. 81 |

**F**

| | |
|---|---|
| FCI | Franka Control Interface. 16 |
| FRI | Fast Research Interface. 15 |
| FSM | Finite State Machine. 29–39, 41, 43, 52, 57, 58, 92, 93 |
| FT | Force/Torque. 79, 80, 83, 92–94 |

**G**

| | |
|---|---|
| GUI | Graphical user interface. 95 |

**H**

| | |
|---|---|
| HMD | head-mounted display. 86, 87 |
| HRC | Human-Robot Collaboration. 85, 86, 91 |

**I**

| | |
|---|---|
| IMU | inertial measurement unit. 92–94, 99 |
| IPS | Indoor Positioning System. 67 |

**M**

| | |
|---|---|
| MOCA | MObile Collaborative robotic Assistant. 16–18, 63, 68, 71, 73–75, 77, 78, 81–84, 96, 99, 100, 119, 120 |

**R**

| | |
|---|---|
| RANSAC | RANdom SAmple Consensus. 28 |
| ROS | Robot Operating System. 16, 35, 39, 52, 53, 58, 67, 71, 81, 88 |

**S**

| | |
|---|---|
| sEMG | surface electromyography. 82, 84, 98 |
| SVD | Singular Value Decomposition. 24 |

**T**

| | |
|---|---|
| TEB | Timed-Elastic-Band. 67, 68, 120 |

# Contents

**List of Abbreviations** **IX**

**1 Introduction** **1**
1.1 Motivation . . . . . 1
1.2 Contributions . . . . . 3
1.3 Outline . . . . . 4

**2 Fundamentals of robot interaction control** **6**
2.1 State-of-the-art overview . . . . . 6
2.1.1 Position control vs. interaction control . . . . . 7
2.1.2 Impedance control . . . . . 7
2.1.3 Force control . . . . . 8
2.1.4 Hybrid approaches . . . . . 9
2.1.5 Controllers comparison . . . . . 9
2.2 Interaction control of fixed based manipulators . . . . . 10
2.3 Interaction control of mobile manipulators with non-holonomic constraints 11
2.3.1 Whole-body Dynamics Analysis . . . . . 12
2.3.2 Whole-body Impedance Controller Implementation . . . . . 14
2.4 Robotic platforms used in this thesis . . . . . 15
2.4.1 Fixed base manipulators . . . . . 15
2.4.2 MOCA: MObile Collaborative robotic Assistant . . . . . 16
2.4.3 Other platforms . . . . . 17

**3 Self-governing robot interaction planning and control** **19**
3.1 The urge of robot interaction autonomy . . . . . 19
3.2 A self-tuning impedance controller . . . . . 20
3.2.1 Impedance regulation along Cartesian axes . . . . . 21
3.2.2 Impedance regulation along the motion vector . . . . . 23
3.3 Implementation and results . . . . . 27
3.3.1 Disaster response . . . . . 27
3.3.2 Agricultural setup . . . . . 34

3.3.3 Item sorting and box-filling . . . . . 48
3.3.4 Kinodynamically constrained vehicles repositioning . . . . . 62

**4 Human-in-the-loop robot interaction planning and control 76**
4.1 A close-proximity approach to robot interaction planning and control . 77
4.1.1 A MObile and reconfigurable Collaborative robot Assistant for conjoined huMAN-robot actions . . . . . 77
4.1.2 Human awareness enhancement through Augmented Reality . . 85
4.2 A far-proximity approach to robot interaction planning and control . . . 94
4.2.1 Teleoperation interface for loco-manipulation control . . . . . 94

**5 Conclusion 104**

**Bibliography 108**

**A Appendix 119**
A.1 Prioritized weighted whole-body inverse dynamics control . . . . . 119
A.2 Planning collision-free paths for holonomic robots . . . . . 120

CHAPTER 1

# Introduction

## 1.1 Motivation

Traditionally, the robotics field has always been strictly connected to industrial applications, where robots are deployed in highly structured environments, as manufacturing plants, operating only within cages and well-separated from human workers. Since these kinds of robots have no knowledge of their surrounding conditions, they represent a potential danger to human workers and thus they are bolted down to the factory floor behind fences so that no contact with humans can be established. This category of robots are mainly programmed a priori by human operators, typically with position control strategies, since all the properties and features of their workspace are known before carrying out the task. In this context, the required level of adaptation is trivial, since these characteristics do not change over time.

The more and more growing demand for flexibility in service applications and in the agile manufacturing sector has highlighted the need for intelligent systems, able to react and adapt to the external environment. To this end, two directions have been followed over the past decade. On the hardware level, robotic systems based on torque sensing and actuation or variable impedance mechanisms have been developed to make them compliant to their surroundings [8, 182]. On the software side instead, robots' perception autonomy has received a great level of attention, to capture the effects of appearance and context [76, 109]. However, there is still a big gap in the bridging action, i.e. associating perception to interaction in an autonomous way. This fundamental shortcoming has limited the application of robots in out-of-the-cage application scenarios, making a framework to enhance their physical interaction autonomy a critical requirement.

With the recent technological advances in robotic platforms hardware design, new control theories have been developed to improve the interaction with humans and envi-

ronmental constraints, especially in scarcely structured application fields such as agriculture, small-batch manufacturing, construction, disaster scenarios, healthcare, and entertainment. The characteristics of this kind of environments require an efficient interaction of the robots with human beings and unknown environments, since the latter set geometric path constraints that are to be followed by the end-effector. In such situations, commonly referred to as *constrained motion*, position control strategies do not represent anymore a feasible option. In fact, planning errors may give origin to a deviation of the end-effector from the desired trajectory, causing the reaction of the controller imposed to reduce the error, which may lead to damage at the interaction point, either on the human/environment side or to the robot itself.

To overcome this drawback, interaction controllers that take into account the contact force at the manipulator's end-effector have been proposed in the past decades. Different conditions require the employment of diverse strategies. On the one hand, there are situations in which substantial contact force values are needed to carry out a certain task, on the other, there are cases where the robot end-effector has to keep a soft profile to comply with the unexpected external disturbances. Since the environment variation can not be known a priori, or at least can not be foreseen with high accuracy, the employed controllers need to be adaptive, so as to modify the control behavior based on the external feedback.

Previous attempts to endow robots with adaptive interaction skills have pursued different directions. An intuitive and well-known approach can be developed based on learning from human demonstrations [96, 97, 111]. This technique has demonstrated promising results provided that a sufficient quantity of training data is available, but has also shown its limiting-factor in the high dependency on the quality of these training data sets. Another drawback is represented by the limited accuracy of sensory measurements related to physical interactions (e.g., forces and torques), especially while performing complex manipulation tasks. This also explains why most learning by demonstration techniques function on a kinematic level.

In order to provide a solution to this shortcoming, analytical solutions have focused on the use of impedance control [4, 48, 124], force control [157, 161, 162], or hybrid interaction controllers [11, 166]. Nevertheless, in the majority of cases, the proposed approaches present a substantial limitation: the control parameters are pre-selected by robot programmers in line with their previous experience in executing analogous tasks. With the presetting of these parameters, the framework can not adapt to varying task conditions, leading to a reduction of flexibility and adaptation capacity of robots to unexpected events [58, 194].

A step forward to equip control systems towards more versatility and adaptation has been represented by the introduction of several adaptive learning techniques. These solutions presented adaptive impedance controllers, also based on a fuzzy neural network [77, 192]. Their main limitation is given by the focus on a specific task, thus they lack of generality. Moreover, the flexibility is furtherly reduced since empirical constants need to be set manually, and the desired impedance matrices are assumed to be diagonal, limiting the adaptability to selective Cartesian axes [72]. Other methods, more generic and not task-dependent, have been presented where, in the absence of interaction, impedance control was reduced to position control. This was achieved by setting high position loop gains, so as to minimize the error between the desired and

actual trajectories [77]. Nevertheless, these systems are not able to distinguish desired interactions coming from the task and undesired contact forces (e.g., collisions), but they respond similarly to different types of contact.

To provide a solution to this problem, in this thesis we aim to propose the development of novel context-aware and adaptive robotic behaviors able to distinguish expected interactions from external disturbances, to be able to react accordingly and appropriately. The originality of the proposed methods consists in the development of novel robot interaction planning and control algorithms that can be regulated either by the robot itself (self-governing approach) or by an operator (human-in-the-loop approach). In the first approach, our aim is to endow the robot with self-adapting skills, so as to prove a certain level of autonomy. However, the more the task complexity grows, the more this becomes challenging. Therefore, acknowledging there is a trade-off between complexity and autonomy, in the second approach we let the human responsible for the high level decisions. Implementing different levels of shared autonomy, we aim to propose approaches for both close-proximity and far-proximity interaction planning and control.

## 1.2 Contributions

In view of the considerations stated in the previous section, the main goal of this thesis resides in the development of adaptive robotic behaviors that respond appropriately to a certain planning strategy, implemented either through robotic autonomous frameworks or entrusting the decision-making level to a human operator.

Self-governing robot strategies are preferable in many areas where they can help to reduce human physical stress and automate repetitive and cognitively unexciting industrial tasks [106], as well as in situations where the human intervention is at risk, as after natural catastrophes or man-made disasters [182]. However, when the task complexity arises, human high-level planning plays a fundamental role to achieve a successful task execution, that would not be possible otherwise.

We can reasonably state that the growth of the *task complexity* linearly corresponds to the difficulty in providing a certain level of *robot autonomy* that ensures enough ef-

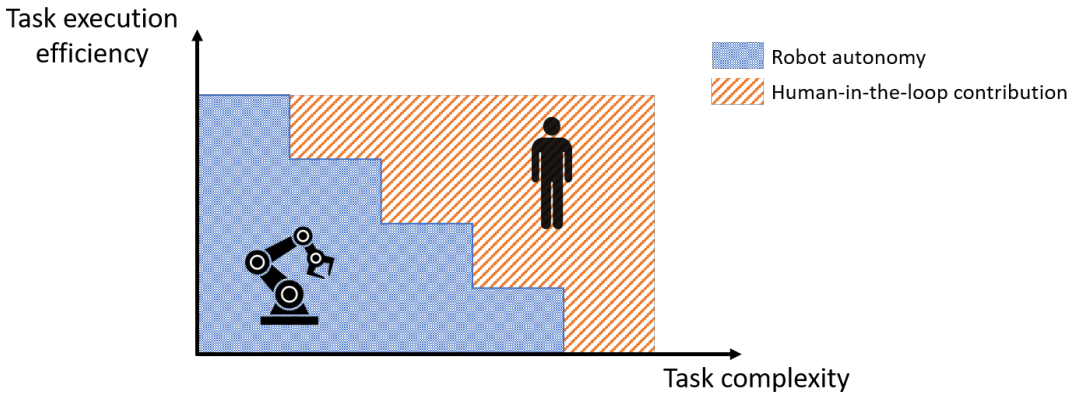


**Figure 1.1:** *When the task complexity raises, an efficient task execution can not be guaranteed solely by autonomous robotic behaviors. In such situations, the introduction of a human in the loop is added to the system, and compensates for the limited robot capabilities.*

ficiency in the task execution. This concept is illustrated in Figure 1.1. When the *task complexity* reaches a certain threshold, autonomous robotic strategies alone can not satisfy the minimum requirements for time and resources efficiency. This limit depends on many factors, as the task definition, the humans' skills, or the specific robot capacities. When the task complexity is low, robots are preferable with respect to humans, achieving higher accuracy, performing tasks with faster motions, and also saving costs related to the employment of fewer human resources. With the rise of task complexity, instead, robots may become slower and less precise, due to the environmental uncertainty and the decreasing situational awareness. Nevertheless, in order to respond to this shortcoming, the introduction of a human in the loop can contribute to improve the global system capabilities. In this way, the addition of the *human-in-the-loop contribution* can compensate for scarce *robot autonomy* abilities.

Based on these premises, the main contributions of this thesis can be summarized as follows:

- C1. A self-tuning impedance controller that adaptively regulates quasi-static parameters of the robot, by distinguishing between expected interactions and external disturbances (related to the scientific publications [18–20])
- C2. Adaptive planning strategies that, also by means of the results of C1, provide the system the ability to cope with unexpected environmental and operational changes (related to the scientific publications [17, 21])
- C3. Augmented Reality based human-robot interfaces to enhance task scheduling, monitoring and work performance (related to the scientific publications [16, 45])
- C4. Intuitive human-robot interfaces for close- and far-proximity teleoperation of mobile collaborative robots (related to the scientific publications [103, 191])

## 1.3 Outline

Figure 1.2 graphically represents the overview of the framework presented in this thesis. Robot interaction planning and control are at the core of this work. To provide the basis

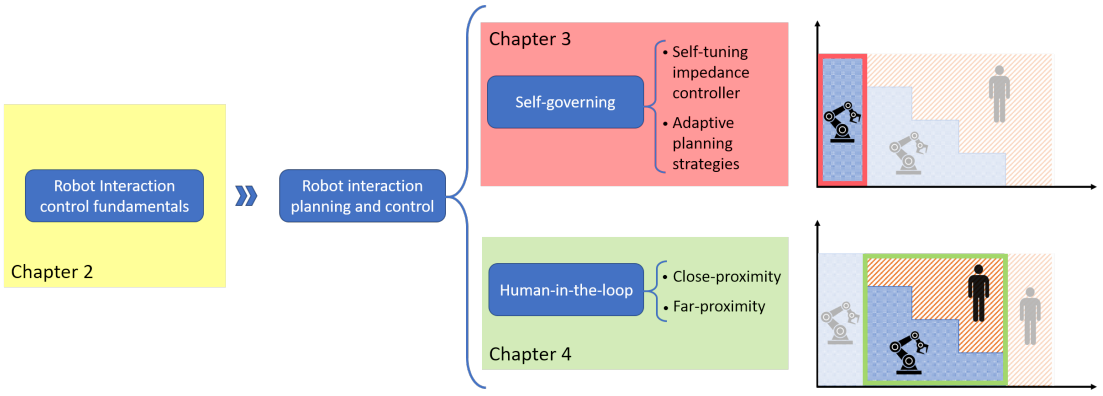


**Figure 1.2:** *Overview of the proposed framework. After the description of the fundamentals of interaction control (Chapter 2), the decision-making level is entrusted to robotic self-governing strategies (Chapter 3) or to human-in-the-loop approaches (Chapter 4).*

for a better comprehension of the methods proposed in the following parts, Chapter 2 illustrates the fundamentals of robot interaction control, enumerating the state-of-the-art control strategies that have been proposed over the last decades. This section includes also more in-depth details about the interaction control laws for fixed-base manipulators and for mobile manipulators with non-holonomic constraints. Finally, the robotic platforms that are going to be used in the next chapters are described, with a particular focus on the new mobile manipulators that have been developed to demonstrate the theoretical content herein presented.

The following chapters are dedicated to the novel contributions brought by this thesis. As shown in Figure 1.2, Chapter 3 treats the planning and control algorithms involved in the left part of the chart depicted in Figure 1.1, where, in view of a smaller amount of task complexity, the decision-making process is entirely entrusted to the robot. In this section, we introduce a self-tuning impedance controller that acts differently based on the type of interaction (C1), and adaptive planning strategies able to cope with unexpected environmental and operational changes, as unplanned human interventions (C2).

On the other hand, Chapter 4 focuses on the central part of the chart, dealing with applications that imply a higher task complexity, where the human-in-the-loop contribution is strictly necessary to reach the minimum requirements for a successful execution of the task. The development of novel human-robot interfaces guarantees performance improvements through Augmented Reality (C3), and favors intuitiveness in teleoperation applications, considering different proximity levels (C4).

Lastly, Chapter 5 summarizes the content of this work with a highlight on its contributions, also addressing the final conclusions and discussing future perspectives.

CHAPTER 2

# Fundamentals of robot interaction control

Among the multidisciplinary areas involved in the robotics field, the study of control theories is indisputably one of the more prominent, given that it plays a crucial role in any robotics application. In this chapter, we aim to provide an overview of the control theories that have been studied and adopted by (mobile) robotic manipulators, with a focus on the algorithms that deal with environmental interaction.

Section 2.1 presents the standard control theories employed in the robotics field, highlighting the distinction between motion and interaction control, and the diverse approaches of force control laws. The state-of-art controllers that represent the research focus of the last decades in this field are also introduced. The next sections provide more details about interaction control techniques for fixed base manipulators (Section 2.2) and mobile manipulators with non-holonomic constraints (Section 2.3). In Section 2.4, we describe the hardware components that will be used in this thesis, including commercial robotic manipulators and novel robotic platforms that were developed to demonstrate the contents illustrated in the next chapters.

## 2.1 State-of-the-art overview

The generation of the reference input for any motion control system is given by trajectory planning techniques. The problem of controlling a manipulator consists in ensuring these reference trajectories, by delivering appropriate generalized forces (force or torques) to be developed by the joint actuators. Based on the situation, different control laws may be employed, with the aim of carrying out the desired task towards a favorable outcome. In this section, after defining the role of interaction control in relation to motion control, we describe the different interaction control laws counted in literature.

### 2.1.1 Position control vs. interaction control

The most common control law in robotics is represented by position control since it is adequate for applications in the free space. This control strategy aims at tracking as closely as possible a motion trajectory, that is why it is widely employed in tasks that do not involve contact forces between the manipulator and the environment, such as spray painting, spot welding, and pick-and-place tasks. The main drawback of this approach is that, in the event of unexpected contacts, position-controlled robots will treat those contact forces as external perturbations, leading to position tracking displacements and undesired contact forces. This behavior shows the extremely high impedance typical of this control strategy, making it inappropriate for interaction tasks with the environment. In fact, successful interaction task execution could be achieved only in the case of accurate planning. If, on the one hand, a precise kinematic and dynamic modeling of the robot manipulator can be obtained, on the other, a detailed description of the environment geometry is quite hard to attain.

**Figure 2.1:** *Position control is widely employed in applications that do not involve contact forces between the manipulator and the environment, as spray painting, spot welding, and pick-and-place tasks.*

Nonetheless, many other manipulation applications in the robotics field highlighted the need of controllers able to deal with interaction forces coming from the environment, and not to treat them as disturbances. In fact, the success in effectively carrying out a manipulation task can be measured by the capacity to handle the interaction between the robot and the environment. To describe the interaction state, it is usual to refer to the contact forces at the end-effector level. Generally, it is desirable to avoid large contact force values to avoid unneeded stressing of the robot itself and the manipulated agent, as this would generate unnecessary high impedance, as in the position control case. We usually refer to *constrained motions* in situations where, while interacting with the environment, the latter sets geometric paths to be followed by the end-effector [169]. Typical practical tasks involved in such situations are the ones involving object manipulation and operations performed on a surface.

Based on the interaction forces application, control strategies can be subdivided in the ones performing *indirect force control* and those performing *direct force control*, as it will be illustrated in the remainder of this section.

### 2.1.2 Impedance control

With the aim of operating in a safer and more efficient manner also in view of unforeseen interaction situations, in the past decades there has been a large focus on robotic systems that can replicate the innate ability of humans to assume a compliant behavior. The dynamic response of a manipulator to its environment is defined with impedance,

inspired by the human motor mechanism. The techniques focused on the control of impedance belong to the *indirect force control* category, and aim to achieve the control of forces using as input motion control, and without explicitly closing a force feedback loop. To reproduce impedance behaviors on robotic manipulators, a twofold path has been undertaken, leading to significant results at both the hardware and the software level.

Hardware-based approaches [25, 88, 121, 140] to achieve impedance control have focused on the development of inherently compliant components, as stiffness-variable springs and variable dampers, exploiting the hardware intrinsic dynamics to vary the robot impedance [81–83]. However, most of the systems developed with this approach implement a passive impedance control, i.e., similarly as a spring, they are not able to deliver more energy than the one previously drawn from the environment [2]. A comprehensive review about variable impedance actuators has been presented in [186].

On the other hand, not to depend on customized hardware developed with inherent compliance, software-based approaches have been investigated. This second category is focused more on active impedance control, where regular actuators provide continuous system energy, and includes three main control strategies: position-based, torque-based and model-based. The first two methods have a similar structure implemented with two loops: the outer loop to compute the command motion trajectory where the desired impedance can be achieved, the inner loop to track the online command trajectory generated by the other. The latter is a position-servo controller in the position-based method and a torque-servo controller in the torque-based one. The main limitation of these two approaches resides in the fact that they are usually used for specific situations, i.e. to realize the desired interaction with a low (position-based) or high (torque-based) stiffness environments. The model-based method, instead, has a broader application range, although it requires an accurate prior knowledge of the original robot dynamics model [171]. This approach provides high flexibility in the design and implementation of a control system. Since the impedance parameters can be adapted online, to respond differently to diverse situations (see thesis contributions C1 and C2 in Section 1.2), the theoretical background of this method will be explained more in details in Section 2.2 and 2.3.

### 2.1.3 Force control

Through the above-presented technique, it is possible to indirectly control the contact forces by controlling appropriate end-effector motions. Nevertheless, only limited values of contact force can be considered given the rough estimate of the environment stiffness. Since many interaction tasks require a highly accurate contact force value, other strategies need to be employed. To apply a force regulation with the end-effector in contact with a compliant environment, *direct force control* is adopted. In this case, the system input is given by the desired force to be applied on a surface, with the control law operating on the error between the desired and measured force values. The force control scheme is realized thanks to the closure of an outer force regulation feedback loop that generates the control input for the manipulator motion control law. The inner loop can be either closed with position or velocity control. For more details on the implementation, refer to [169].

The force control effectiveness mainly depend on the geometry of the environment.

In fact, this approach is meaningful only when applied to the operational space directions in which there may arise interaction contact forces between the robot and the environment.

### 2.1.4 Hybrid approaches

To overcome the limitations of the former control approaches, a few hybrid strategies have been proposed over the years. A renowned method is represented by the hybrid position/force control [46, 91, 115, 117, 133], that combines position and force control together giving rise to another compliant control strategy. This method implies a clear division of the task space into two subspaces, i.e. the position-controlled subspace and the force-controlled subspace, where the respective control laws are adopted. This precondition represents the most challenging step in the application of this approach, since it requires that the geometry of the environment is accurately known. This turns out to be a great applicability limitation for all the tasks dealing with unstructured and dynamically changing environments [125]. The manipulator impedance is characterized by very high values in the position-controlled subspace and very low values in the force-controlled one.

Other hybrid strategies focused on hybrid impedance control laws, combining impedance and hybrid position/force control into a single control scheme [11]. Furthermore, approaches combining impedance and force have demonstrated their advantage with respect to pure force or impedance control methods, when the environmental interaction can not be predicted [166, 185]. Schindlbeck and Haddadin [166] presented a hybrid Cartesian force/impedance controller equipped with energy tanks to preserve passivity, also introducing a controller shaping function that robustly handles unexpected contact loss and avoids chattering behavior, that usually characterizes switching based approaches.

### 2.1.5 Controllers comparison

In view of the multiple approaches above-presented, some can be more convenient than the others, depending on the task requirements and on the level of information about the environment. Therefore, a comparison among them can facilitate the comprehension of their advantages and drawbacks.

Although position control constitutes the most frequently employed scheme in robotics, this only works effectively in the free-space, so it is not appropriate for any task that involves an interaction with the environment, since all the contacts are treated as disturbances to be rejected, as amply described in Section 2.1.1.

On the contrary, pure force control strategies allow for a precise definition of the contact force, but do not permit any control of the motion. This is due to the tendency of the robot to move passively with the environment, in order to achieve the reference contact force given as input to the control law. The effectiveness of force control strategies is guaranteed only in the case of constrained space, while, if no initial contact with the environment is established, the robot needs first to be driven to the contact pose with a motion based controller, as position control. This approach has a remarkable drawback, since at the moment of switching from one control to the other, instabilities can arise. In fact, if the contact pose is not accurately known, the controller switch can

happen asynchronously with respect to the contact with the environment. If, when the position control strategy is activated, the manipulator has already touched the environment, excessive contact forces can be produced. On the other hand, at the moment of leaving the contact surface, the transition to position control can be invoked after the manipulator is already in the free space, thus leading to undesired motions, that can be identified as "contact loss" [166].

This drawback can be overcome by the hybrid force/impedance approach. However, if stability needs to be ensured at all times, exact force regulation may not be achieved. The main disadvantage of hybrid force/position control strategies is represented by the difficulty to divide the task subspaces, i.e. which are the directions every controller needs to be applied to. This limits the applicability of this method to structured environment, that permit an accurate identification of the environmental geometry.

Position control and force control can be regarded as two extreme situations of impedance control [171], that results to be more robust in dynamically changing scenarios. In fact, this strategy as a higher flexibility, being capable of regulating online the robot compliance to meet the various needs of different manipulation phases. In addition, its adaptiveness implies the ability to be employed in constrained motion, free motion and during the transition from one motion to the other, without the need of switching the controller, therefore not suffering for the above-mentioned unstable responses.

## 2.2 Interaction control of fixed based manipulators

As described in the previous section, impedance control techniques can guarantee good results in interaction tasks with unstructured environment. Since this approach is at the basis of the work presented in the next chapters, hereafter we aim to provide the theoretical formulation of the classical Cartesian impedance control strategy for a fixed base robotic manipulator.

The general dynamic model of a manipulator with $n$ Degrees of Freedom (DoF) and joint coordinates $\boldsymbol{q} \in \mathbb{R}^n$ can be written as:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} + \boldsymbol{\tau}^{ext}, \tag{2.1}$$

where $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ is the symmetric and positive definite inertial matrix, $\boldsymbol{C} \in \mathbb{R}^{n\times n}$ is the Coriolis and centrifugal matrix, $\boldsymbol{g} \in \mathbb{R}^n$ is the gravity vector, $\boldsymbol{\tau} \in \mathbb{R}^n$ and $\boldsymbol{\tau}^{ext} \in \mathbb{R}^n$ are the commanded torque vector and external torque vector, respectively. In Eq. 2.1, it is maintained the property that $\dot{\boldsymbol{M}} - 2\boldsymbol{C} \in \mathbb{R}^{n\times n}$ is skew symmetric [169].

The relation between the joint angles $\boldsymbol{q}$ and the Cartesian space coordinates $\boldsymbol{x} \in \mathbb{R}^m$ is established through the forward kinematics equation $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{q})$, whose velocity and acceleration can be derived through the manipulator Jacobian matrix $\boldsymbol{J}(\boldsymbol{q})$ as:

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

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

In view of these equations, also the relation between the external torques $\boldsymbol{\tau}_{ext}$ and the external wrenches $\boldsymbol{F}_{ext}$ can be derived as:

$$\boldsymbol{\tau}_{ext} = \boldsymbol{J}(\boldsymbol{q})^\top \boldsymbol{F}_{ext}. \tag{2.4}$$

With the aim of achieving a desired compliant behavior, a relation between the Cartesian displacement, $\boldsymbol{\Delta x} = \boldsymbol{x} - \boldsymbol{x}_d$ , and the external wrenches $\boldsymbol{F}_{ext}$, is established through a mass-spring-damper relation:

$$\boldsymbol{F}_{ext} = \boldsymbol{\Lambda}_d \boldsymbol{\Delta \ddot{x}} + \boldsymbol{D}_d \boldsymbol{\Delta \dot{x}} + \boldsymbol{K}_d \boldsymbol{\Delta x}, \tag{2.5}$$

where $\boldsymbol{\Lambda}_d$, $\boldsymbol{D}_d$, and $\boldsymbol{K}_d$ are the positive definite matrices that represent the system virtual mass, damping and stiffness. The former dynamic relation between the task variables constitutes the main distinction with respect to the other classical control methods, i.e. force and position control [81–83].

Rewriting Eq. 2.1 and considering the relations in Eqs. 2.2, 2.3, and 2.4, we can obtain [100]:

$$\boldsymbol{\Lambda}(\boldsymbol{x})\ddot{\boldsymbol{x}} + \boldsymbol{\mu}(\boldsymbol{x}, \dot{\boldsymbol{x}})\dot{\boldsymbol{x}} + \boldsymbol{F}_g(\boldsymbol{q}) = \boldsymbol{F}_\tau + \boldsymbol{F}_{ext}, \tag{2.6}$$

where

$$\boldsymbol{\Lambda}(\boldsymbol{x}) = \boldsymbol{J}(\boldsymbol{q})^{-\top} \boldsymbol{M}(\boldsymbol{q}) \boldsymbol{J}(\boldsymbol{q})^{-1}, \tag{2.7}$$

$$\boldsymbol{\mu}(\boldsymbol{x}, \dot{\boldsymbol{x}}) = \boldsymbol{J}(\boldsymbol{q})^{-\top} (\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{M}(\boldsymbol{q}) \boldsymbol{J}(\boldsymbol{q})^{-1} \dot{\boldsymbol{J}}(\boldsymbol{q})) \boldsymbol{J}(\boldsymbol{q})^{-1}, \tag{2.8}$$

$$\boldsymbol{F}_g(\boldsymbol{q}) = \boldsymbol{J}(\boldsymbol{q})^{-\top} \boldsymbol{g}(\boldsymbol{q}), \tag{2.9}$$

$$\boldsymbol{F}_\tau = \boldsymbol{J}(\boldsymbol{q})^{-\top} \boldsymbol{\tau}. \tag{2.10}$$

The control input $\boldsymbol{F}_\tau$ can then be computed as:

$$\begin{aligned} \boldsymbol{F}_\tau = {} & \boldsymbol{\Lambda}(\boldsymbol{x})\ddot{\boldsymbol{x}}_d + \boldsymbol{\mu}(\boldsymbol{x}, \dot{\boldsymbol{x}})\dot{\boldsymbol{x}} + \boldsymbol{F}_g(\boldsymbol{q}) \\ & - \boldsymbol{\Lambda}(\boldsymbol{x})\boldsymbol{\Lambda}_d^{-1}(\boldsymbol{D}_d \boldsymbol{\Delta \dot{x}} + \boldsymbol{K}_d \boldsymbol{\Delta x}) + (\boldsymbol{\Lambda}(\boldsymbol{x})\boldsymbol{\Lambda}_d^{-1} - \boldsymbol{I})\boldsymbol{F}_{ext}, \end{aligned} \tag{2.11}$$

and consequently the joint torques can be computed as $\boldsymbol{\tau} = \boldsymbol{J}(\boldsymbol{q})^\top \boldsymbol{F}_{ext}$. To avoid any external wrenches feedback, we can set $\boldsymbol{\Lambda}_d = \boldsymbol{\Lambda}(\boldsymbol{x})$, thus transforming Eq. 2.11 in:

$$\boldsymbol{F}_\tau = \boldsymbol{\Lambda}(\boldsymbol{x})\ddot{\boldsymbol{x}}_d + \boldsymbol{\mu}(\boldsymbol{x}, \dot{\boldsymbol{x}}_d)\dot{\boldsymbol{x}} + \boldsymbol{F}_g(\boldsymbol{q}) - \boldsymbol{D}_d \boldsymbol{\Delta \dot{x}} - \boldsymbol{K}_d \boldsymbol{\Delta x}. \tag{2.12}$$

## 2.3 Interaction control of mobile manipulators with non-holonomic constraints

Fixed base manipulators have demonstrated their effectiveness especially in industrial scenarios. Nevertheless, to enhance the application range of interaction controlled robots, we need to consider also robotic platforms that, besides manipulation capabilities, are also able to navigate through the environment. To this end, many robotic systems have been developed in the last decades, including humanoid robots, either legged as HRP-4 [93], ASIMO [165], LOLA [131], WALK-MAN [182], CENTAURO [95], COMAN [183], or wheeled as ARMAR-III [14], TWENDY-ONE [89], Rollin' Justin [31], PR2 [29], and manipulators mounted on mobile bases as KUKA youBot [24], EL-E [90], and HERB 2.0 [174].

One of the main criteria to distinguish these platforms is to classify them based on the way they pursue locomotion: legged systems or wheeled platforms. Although

the former could have a high potential to be employed in a wider range of scenarios, as climbing a stair, they are more difficult to control. On the other hand, wheeled platforms have been proven to be very effective in many complex service tasks. Since the problem of balancing and stabilizing the gait has not to be considered (a part from platforms with less than three wheels [175]), they can focus on more complex manipulation tasks. Based on these considerations, in this section we present the whole-body impedance control formulation for a mobile manipulator with non-holonomic constraints.

### 2.3.1 Whole-body Dynamics Analysis

Before defining a unique system, we consider the dynamic models of the manipulator that is mounted on top of the mobile platform and the latter as two different systems, each one defined in its own base frame. Figure 2.2 represents a sketch of the entire system, with the reference frames useful to describe the entire platform, namely world frame $\Sigma_W$, mobile platform base frame $\Sigma_M$, and manipulator base frame $\Sigma_R$ and end-effector frame $\Sigma_{EE}$. As known, a robotic manipulator is a coupled, time-varying and nonlinear system. As already described by Eq. 2.1, and adding the subscript $r$ to distinguish the dynamic of the robotic arm from the one of the mobile platform, the general dynamic model of a manipulator with joint coordinates $\boldsymbol{q}_r \in \mathbb{R}^n$ can be written as:

$$\boldsymbol{M}_r(\boldsymbol{q}_r)\ddot{\boldsymbol{q}}_r + \boldsymbol{C}_r(\boldsymbol{q}_r, \dot{\boldsymbol{q}}_r) + \boldsymbol{g}_r(\boldsymbol{q}_r) = \boldsymbol{\tau}_r + \boldsymbol{\tau}_r^{ext}, \tag{2.13}$$

where $\boldsymbol{M}_r \in \mathbb{R}^{n\times n}$ is the symmetric and positive definite inertial matrix of the arm. $\boldsymbol{C}_r \in \mathbb{R}^n$ is the Coriolis and centrifugal force, $\boldsymbol{g}_r \in \mathbb{R}^n$ is the gravity vector, $\boldsymbol{\tau}_r \in \mathbb{R}^n$ and $\boldsymbol{\tau}_r^{ext} \in \mathbb{R}^n$ are the commanded torque vector and external torque vector, respectively. In Eq. 2.13, it is maintained the property that $\dot{\boldsymbol{M}}_r - 2\boldsymbol{C}_r \in \mathbb{R}^{n\times n}$ is skew symmetric.

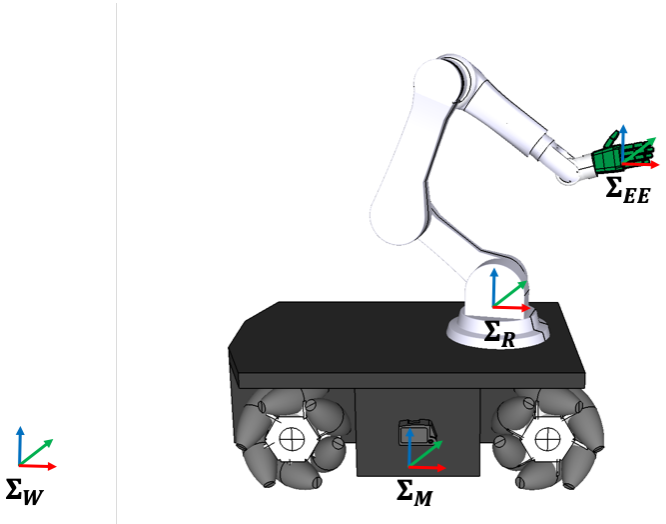


**Figure 2.2:** *Sketch of a mobile manipulator with its reference frames: world frame $\Sigma_W$, mobile platform base frame $\Sigma_M$, robotic manipulator base frame $\Sigma_R$, and robotic manipulator end-effector frame $\Sigma_{EE}$.*

On the other hand, the dynamics of a mobile platform with $m$ DoF, with virtual joint coordinates $\boldsymbol{q}_v \in \mathbb{R}^m$, can be obtained using the Lagrangian approach in the following

form [197]:

$$\boldsymbol{M}_v(\boldsymbol{q}_v)\ddot{\boldsymbol{q}}_v + \boldsymbol{C}_v(\boldsymbol{q}_v, \dot{\boldsymbol{q}}_v) = \boldsymbol{E}_v(\boldsymbol{q}_v)\boldsymbol{\tau}_v - \boldsymbol{A}_v^\top(\boldsymbol{q}_v)\boldsymbol{\lambda}, \tag{2.14}$$

where $\boldsymbol{M}_v \in \mathbb{R}^{m\times m}$ is the symmetric and positive definite inertial matrix of the mobile platform, $\boldsymbol{C}_v \in \mathbb{R}^m$ is the centrifugal force, $\boldsymbol{E}_v \in \mathbb{R}^{m\times m}$ is the input transformation matrix, $\boldsymbol{\tau}_v \in \mathbb{R}^m$ is the commanded torque vector, $\boldsymbol{A}_v \in \mathbb{R}^{m\times m}$ is the constraint matrix and $\boldsymbol{\lambda} \in \mathbb{R}^m$ is the Lagrange multiplier which denotes the constraint force vector.

When integrating the manipulator and the mobile platform into one mobile manipulation system, there exists a dynamic interaction between these two subsystems. The new dynamic equations of the manipulator and the mobile platform subject to each other are given by Eqs. 2.15 and 2.16 [193]:

$$\begin{aligned}\boldsymbol{M}_r(\boldsymbol{q}_r)\ddot{\boldsymbol{q}}_r + \boldsymbol{C}_r(\boldsymbol{q}_r, \dot{\boldsymbol{q}}_r) + \boldsymbol{C}_{rv}(\boldsymbol{q}_r, \dot{\boldsymbol{q}}_r, \dot{\boldsymbol{q}}_v) + \boldsymbol{g}_r(\boldsymbol{q}_r) \\ = \boldsymbol{\tau}_r + \boldsymbol{\tau}_r^{ext} - \boldsymbol{R}_r(\boldsymbol{q}_r, \boldsymbol{q}_v)\ddot{\boldsymbol{q}}_v,\end{aligned} \tag{2.15}$$

where $\boldsymbol{C}_{rv} \in \mathbb{R}^n$ represents Coriolis and centrifugal terms caused by angular motion of the mobile platform, and $\boldsymbol{R}_r \in \mathbb{R}^{n\times m}$ is the inertial matrix which represents the effect of the mobile platform dynamics on the manipulator.

$$\begin{aligned}&\boldsymbol{M}_v(\boldsymbol{q}_v)\ddot{\boldsymbol{q}}_v + \boldsymbol{C}_v(\boldsymbol{q}_v, \dot{\boldsymbol{q}}_v) + \boldsymbol{C}_{vr}(\boldsymbol{q}_v, \boldsymbol{q}_r, \dot{\boldsymbol{q}}_v, \dot{\boldsymbol{q}}_r) = \\ &\boldsymbol{E}_v(\boldsymbol{q}_v)\boldsymbol{\tau_v} - \boldsymbol{A}_v^\top(\boldsymbol{q}_v)\boldsymbol{\lambda} - \boldsymbol{M}_{vr}(\boldsymbol{q}_v, \boldsymbol{q}_r)\ddot{\boldsymbol{q}}_v - \boldsymbol{R}_v(\boldsymbol{q}_v, \boldsymbol{q}_r)\ddot{\boldsymbol{q}}_r,\end{aligned} \tag{2.16}$$

where $\boldsymbol{M}_{vr} \in \mathbb{R}^{m\times m}$ and $\boldsymbol{C}_{vr} \in \mathbb{R}^m$ denote the inertial term and Coriolis and centrifugal terms due to the presence of the manipulator, $\boldsymbol{R}_v \in \mathbb{R}^{m\times n}$ is the inertial matrix which reflects the dynamic effect of the manipulator motion on the mobile platform.

As, usually, mobile platforms present a velocity-based control and a high gain in the low level velocity controller, the dynamics of the mobile platform can be omitted and any external dynamic effect from the manipulator can be ignored. Considering that we aim at achieving a whole-body impedance control law, a force-torque interface is preferred. To this end, we introduce the definition of a Cartesian admittance controller based on the velocity interface:

$$\boldsymbol{M}_{adm}\ddot{\boldsymbol{q}}_v^{des} + \boldsymbol{D}_{adm}\dot{\boldsymbol{q}}_v^{des} = \boldsymbol{\tau}_v^{vir} + \boldsymbol{\tau}_v^{ext}, \tag{2.17}$$

where $\boldsymbol{M}_{adm} \in \mathbb{R}^{m\times m}$ and $\boldsymbol{D}_{adm} \in \mathbb{R}^{m\times m}$ are the virtual inertial and virtual damping, $\dot{\boldsymbol{q}}_v^{des} \in \mathbb{R}^m$ is the desired velocity sent to the mobile platform, $\boldsymbol{\tau}_v^{vir} \in \mathbb{R}^m$ and $\boldsymbol{\tau}_v^{ext} \in \mathbb{R}^m$ are the virtual and external torque interfaces.

Since the majority of the manipulation task, or at least the ones considered in this work, do not involve high dynamics, the mobile platform can be considered in a quasi-static state. Therefore we can consider that its motion does not have much effect on the manipulator, and, as a consequence, the dynamic coupling terms in Eq. 2.15 can be neglected. Consequently, the overall dynamics of the mobile manipulator can be formulated as:

$$\begin{aligned}&\begin{pmatrix}\boldsymbol{M}_{adm} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{M}_r\end{pmatrix}\begin{pmatrix}\ddot{\boldsymbol{q}}_v\\ \ddot{\boldsymbol{q}}_r\end{pmatrix} + \begin{pmatrix}\boldsymbol{D}_{adm} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{C}_r\end{pmatrix}\begin{pmatrix}\dot{\boldsymbol{q}}_v\\ \dot{\boldsymbol{q}}_r\end{pmatrix} + \begin{pmatrix}\boldsymbol{0}\\ \boldsymbol{g}_r\end{pmatrix}\\ &= \begin{pmatrix}\boldsymbol{\tau}_v^{vir}\\ \boldsymbol{\tau}_r\end{pmatrix} + \begin{pmatrix}\boldsymbol{\tau}_v^{ext}\\ \boldsymbol{\tau}_r^{ext}\end{pmatrix}.\end{aligned} \tag{2.18}$$

In this way, we obtained a simplified whole-body dynamic formula. The following assumptions allow us to decouple the dynamic of the two subsystems.

*Assumption 1:* Any external dynamic effect is compensated thanks to the high gain of the low level velocity controller.

*Assumption 2:* While manipulating an object, the mobile platform is in a quasi-static state, so its motion does not have much effect on the manipulator.

Based on the analysis carried out above, these assumptions are feasible. In the next section, we provide more details on the implementation of a whole-body impedance controller based on the decoupled dynamics.

### 2.3.2 Whole-body Impedance Controller Implementation

Considering the dynamic relationship of a manipulator with $n$ DoF and joint coordinates $\boldsymbol{q} \in \mathbb{R}^n$ presented in Eq. 2.5, we can define the Cartesian impedance controller input for the main task $\boldsymbol{\tau}_{imp}$ as following:

$$\boldsymbol{\tau}_{imp} = \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}(\boldsymbol{q})^\top(\boldsymbol{\Lambda}(\boldsymbol{x})\ddot{\boldsymbol{x}}_d + \boldsymbol{\mu}(\boldsymbol{x}, \dot{\boldsymbol{x}})\dot{\boldsymbol{x}}_{\boldsymbol{d}} - \boldsymbol{K}_d\boldsymbol{\Delta}\boldsymbol{x} - \boldsymbol{D}_d\dot{\boldsymbol{\Delta}}\boldsymbol{x}), \tag{2.19}$$

where all the terms have been defined above and will not be repeated.

For redundant robotic arms, the null-space task input is defined as:

$$\boldsymbol{\tau}_{null} = \boldsymbol{N}(\boldsymbol{q})(-\boldsymbol{D}_n\dot{\boldsymbol{q}} - \boldsymbol{K}_n(\boldsymbol{q} - \boldsymbol{q}_{d,0})), \tag{2.20}$$

where $\boldsymbol{K}_n \in \mathbb{R}^{n\times n}$ and $\boldsymbol{D}_n \in \mathbb{R}^{n\times n}$ are the desired Cartesian stiffness and damping of the null-space task, $\boldsymbol{q}_{d,0} \in \mathbb{R}^n$ is the virtual equilibrium position, $\boldsymbol{N}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is the projection matrix in order to prevent interference with the Cartesian impedance behavior. Here, the following dynamically consistent projection proposed by Khatib [100] is employed:

$$\boldsymbol{N}(\boldsymbol{q}) = \boldsymbol{I} - \boldsymbol{J}^\top(\boldsymbol{q})\boldsymbol{\Lambda}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})\boldsymbol{M}^{-1}(\boldsymbol{q}). \tag{2.21}$$

We need now to implement a whole-body Cartesian impedance control for a mobile manipulator, composed of a mobile platform with $m$ DoF and a robotic arm with $n$ DoF. To this end, the whole-body forward kinematics $\boldsymbol{x}_w(\boldsymbol{q}) \in \mathbb{R}^6$, the whole-body Jacobian $\boldsymbol{J}_w(\boldsymbol{q}) \in \mathbb{R}^{6\times(n+m)}$, the whole body Cartesian inertial $\boldsymbol{\Lambda}_w(\boldsymbol{x}_w) \in \mathbb{R}^{6\times6}$, the Cartesian Coriolis/centrifugal $\boldsymbol{\mu}(\boldsymbol{x}_w, \dot{\boldsymbol{x}}_{\boldsymbol{w}}) \in \mathbb{R}^{6\times6}$, and the null-space projection matrix $\boldsymbol{N}(\boldsymbol{q}) \in \mathbb{R}^{(n+m)\times(n+m)}$ need to be constructed.

From now on, we will assume the mobile platform to have $m = 3$ DoF, namely two translational on the plane and one rotational around the vertical axis. Since, in this work, we consider only holonomic platforms this assumption is guaranteed, and this will also allow us to provide more accurate details about the implementation.

The forward kinematics transform of a mobile manipulator at the end-effector frame $\Sigma_{EE}$ with respect to the world frame $\Sigma_W$, $\boldsymbol{T}^W_{EE}(\boldsymbol{q}) \in \mathbb{R}^{4\times4}$ can be derived as follows:

$$\boldsymbol{T}^W_{EE}(\boldsymbol{q}) = \boldsymbol{T}^W_M(\boldsymbol{q}_v)\boldsymbol{T}^M_R\boldsymbol{T}^R_{EE}(\boldsymbol{q}_r), \tag{2.22}$$

where $\boldsymbol{q} = (\boldsymbol{q}_v, \boldsymbol{q}_r)^\top \in \mathbb{R}^{(n+3)}$ represents the whole-body joint variables, that in details are given by $\boldsymbol{q}_v = (q_{vx}, q_{vy}, q_{vz})^\top \in \mathbb{R}^3$, $q_{vx}$, $q_{vy}$, $q_{vz}$ respectively represent the translation of mobile platform in x- and y-direction and rotation in z-direction, and $\boldsymbol{q}_r \in \mathbb{R}^n$ represents the joint coordinates of the manipulator. As shown in Fig. 2.2, $\boldsymbol{T}^W_M$, $\boldsymbol{T}^M_R$

and $\boldsymbol{T}_{EE}^{R}$ represent respectively the transformations from $\Sigma_M$ to $\Sigma_W$, from $\Sigma_R$ to $\Sigma_M$ (constant), and from $\Sigma_{EE}$ to its $\Sigma_R$.

The six dimensional representation of $\Sigma_{EE}$, with respect to $\Sigma_W$, $\boldsymbol{x}_w(\boldsymbol{q})$ can be extracted in angle-axis form, which is used by the robotic arm as follows:

$$\boldsymbol{x}_w(\boldsymbol{q}) = \boldsymbol{W}(\boldsymbol{x}(\boldsymbol{q}_r) + \boldsymbol{d}_0) + \boldsymbol{q}_v^{'}(\boldsymbol{q}_v), \tag{2.23}$$

where $\boldsymbol{W} = \begin{pmatrix} \boldsymbol{Rot}(\boldsymbol{z}, q_{vz}) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I^{3\times 3}} \end{pmatrix} \in \mathbb{R}^{6\times 6}$, $\boldsymbol{Rot}(\boldsymbol{z}, q_{vz})$ represents the rotation matrix around the *z-axis* with angle $q_{vz}$, $\boldsymbol{x}(\boldsymbol{q}_r) \in \mathbb{R}^6$ represents the six dimensional representation of $\Sigma_{EE}$ with respect to $\Sigma_R$, $\boldsymbol{d}_0 \in \mathbb{R}^6$ is the deviation between the $\Sigma_M$ and $\Sigma_R$, and $\boldsymbol{q}_v^{'} = (q_{vx}, q_{vy}, 0, 0, 0, q_{vz})^\top$.

The mobile platform general Jacobian matrix with respect to $\Sigma_W$ can be derived directly from equation Eq. 2.23:

$$\begin{aligned} \boldsymbol{J}_w(\boldsymbol{q}) &= \frac{\partial \boldsymbol{x}_w(\boldsymbol{q})}{\partial \boldsymbol{q}^\top} = \begin{pmatrix} \frac{\partial \boldsymbol{x}_w(\boldsymbol{q})}{\partial \boldsymbol{q}_v^\top} & \frac{\partial \boldsymbol{x}_w(\boldsymbol{q})}{\partial \boldsymbol{q}_r^\top} \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{J}_m & \boldsymbol{W}\boldsymbol{J}_r \end{pmatrix}, \end{aligned} \tag{2.24}$$

where $\boldsymbol{J}_m \in \mathbb{R}^{6\times 3}$ represents the contribution of the mobile platform velocity, and $\boldsymbol{J}_r \in \mathbb{R}^{6\times n}$ is the arm Jacobian matrix.

Based on the whole-body forward kinematics $\boldsymbol{x}_w(\boldsymbol{q})$, the whole-body Jacobian matrix $\boldsymbol{J}_w(\boldsymbol{q})$ and the dynamic model interface provided by the manipulator, the whole-body Cartesian inertial $\boldsymbol{\Lambda}_w(\boldsymbol{x}_w)$, the Cartesian Coriolis/centrifugal $\boldsymbol{\mu}(\boldsymbol{x}_w, \dot{\boldsymbol{x}}_{\boldsymbol{w}})$ and the null-space projection matrix $\boldsymbol{N}(\boldsymbol{q})$ can be computed from equation Eq. 2.7, Eq. 2.8 and Eq. 2.21, respectively.

## 2.4 Robotic platforms used in this thesis

In the next chapters, the theoretical contributions will be supported by experimental results, so in this section we aim to introduce the robotic platforms that will be employed to carry out the experiments that led to such results.

### 2.4.1 Fixed base manipulators

Two industrial collaborative robots have been used to perform experiments with fixed base manipulators: KUKA LBR iiwa 14 and Franka Emika Panda. Both robotic arms have 7 DoF and can be controlled with torque commands, that is a primary requirement for the development of impedance control strategies.

#### KUKA LBR iiwa 14

KUKA LBR iiwa 14 (Figure 2.3, on the left) is a 7-axis industrial robot arm that has been designed for safe human-robot collaboration. It is equipped with joint-torque sensors in all axes to perform precise motion commands and detect contacts with humans and objects. The payload of $14Kg$ places it among the top-rated cobots on the market. The software components we implemented, have been developed in a `C++` environment, and the robot was controlled with joint torque commands that were sent to the manipulator using Fast Research Interface (FRI) provided by KUKA at $500Hz$.

**Figure 2.3:** *On the left KUKA LBR iiwa 14 robotic arm, on the right Franka Emika Panda manipulator. LBR stands for "Leichtbauroboter" (German for lightweight robot), and iiwa for "intelligent industrial work assistant".*

#### Franka Emika Panda

Franka Emika Panda (Figure 2.3, on the right) is also composed of 7 joints, thus allowing to take advantage of a redundancy behavior, crucial aspect for a higher safety in human populated environments. Although its payload is limited to $3Kg$, it natively supports the robotics middleware Robot Operating System (ROS) [154], thus intrinsically enabling modular and flexible software development, particularly useful in research oriented projects. The software architectures that will be presented rely upon a tailored version of *franka_ros* metapackage, the ROS integration for Franka Emika robots. This package integrates *libfranka*, an open source C++ interface, into ROS and ROS Control. This interface communicates with the robot through the Franka Control Interface (FCI), that provides the current robot status and enables its direct control with an external workstation PC.

### 2.4.2 MOCA: MObile Collaborative robotic Assistant

A part of the work carried out within the scope of this thesis was dedicated to the design and development of a new research platform. The aim was to create a collaborative assistant, that can help humans in various tasks, with a focus on the industrial environment. The MObile Collaborative robotic Assistant (MOCA) [191], is composed of a lightweight manipulator arm, a Pisa/IIT SoftHand [36] (see also next section), and a mobile platform driven by four Omni-directional wheels. The idea has been to create a general purpose robot that can be employed for many applications. In fact, especially in small and medium-sized enterprises this is a crucial point, since they may not be able to afford multiple single-purpose, costly machines. On the other hand, they can exploit the flexibility of a multi-purpose machine, relieving the cost burden.

MOCA is the result of the integration of three components, as shown in Figure 2.4: Franka Emika Panda robotic arm, equipped with the underactuated Pisa/IIT SoftHand, which are mounted on top of Robotnik SUMMIT-XL STEEL mobile platform. A commercial camera supported by a pole can also be added to the mobile base.

Robotnik SUMMIT-XL STEEL is driven by four Omni-directional wheels, allowing the platform to avoid non-holonomic constraints. It can carry up to $130Kg$ payload with the intent to fulfill industrial applications, such as logistics, indoor transport, and

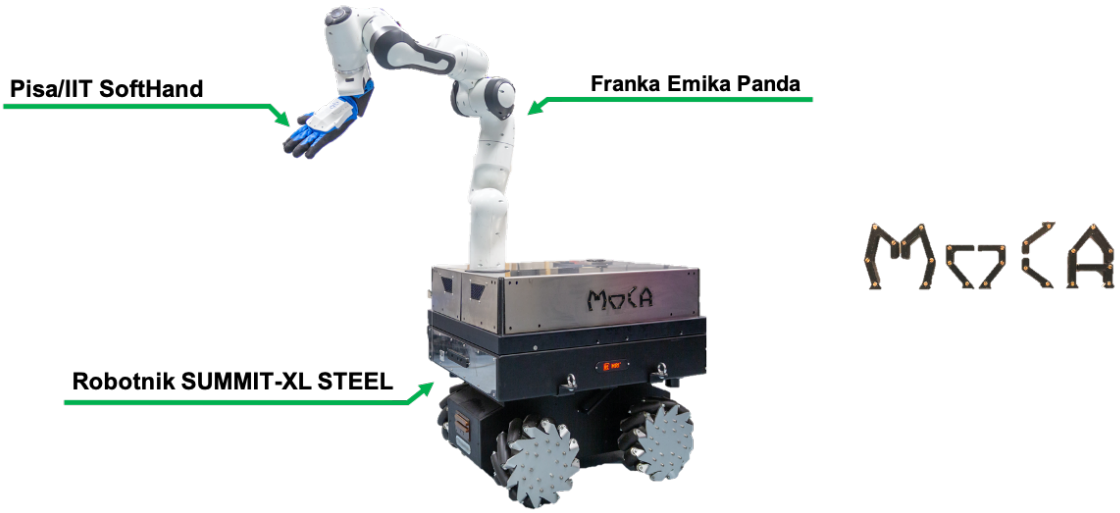


**Figure 2.4:** *The MObile Collaborative robotic Assistant (MOCA) is a research platform developed at IIT, composed of a lightweight manipulator arm, a Pisa/IIT SoftHand, and a mobile platform driven by four Omni-directional wheels.*

so on. The Cartesian velocity control interface is offered with a high gain in low level, which implies that the dynamics of the mobile platform can be omitted. It provides the odometry data, computed by Extended Kalman Filter (EKF) with the use of a high precision inertial measurement unit and the wheels' velocity. Moreover, the mobile platform is equipped with two laser scanners, two PTZ cameras and one 3D camera, which can be utilized for autonomous navigation and obstacle avoidance.

Although each of the above components possesses particular advantages, when integrated in one system, they present significant challenges. First of all, the control interface of Franka Emika Panda, Robotnik SUMMIT-XL STEEL, and Pisa/IIT SoftHand are respectively torque-based, velocity-based, and current-based (underactuated). Hence, the causal interactions in such a dynamical system must be considered in control of robot interaction controller [48]. Secondly, this integration introduces more redundancy (the manipulator and the mobile platform have 7-DoF and 3-DoF, respectively), which adds complexity to the control. Furthermore, the robotic arm and the mobile platform have different bandwidths: the mobile platform typically has a slower dynamic response than the manipulator [193], which must be taken into account.

### 2.4.3 Other platforms

In this section we introduce the robotic end-effector that has been largely adopted in the experiments that will be presented, i.e. the Pisa/IIT SoftHand. In addition, we also provide a brief overview of the humanoid robot WALK-MAN that was employed in a preliminary study illustrated at the beginning of Chapter 3.

#### Pisa/IIT SoftHand

The Pisa/IIT SoftHand [36], illustrated in the left part of Figure 2.5, is an underactuated robot hand, that combines robustness and ease of control with high grasping versatility, due to its aspect that is similar to that of the human hand. It has 19 joints, but only uses

**Figure 2.5:** *Left: the Pisa/IIT SoftHand simplifies grasp planning and control, through its synergy-driven design. Right: the humanoid robot WALK-MAN.*

1 actuator to activate its adaptive synergy. The design is very soft and safe, yet powerful and extremely robust, obtained through the use of innovative articulations and ligaments replacing conventional joint design. The idea of adaptive synergies, which comes from the combination of natural motor control principles, is the core of the Pisa/IIT SoftHand. As a result the hand can adapt itself according to the physical interaction of its body with the object, allowing to grasp a wide range of objects despite its single degree of actuation. Perfect recovery from large deformations and even disarticulation is allowed due to the innovative, biomorphic design.

The features of simplicity, lightness, robustness and compliance make it an ideal choice for integration to MOCA robot illustrated in the previous section.

#### WALK-MAN

WALK-MAN [182] (Figure 2.5, on the right) is an electrical motor driven full-size humanoid robot with 31 DoF. Its total mass is $118kg$ and it is $1.91m$ tall. Its Series Elastic high-end Actuation (SEA) units can reach velocities up to $19.5\frac{rad}{sec}$ and torques up to $400Nm$. The robot is equipped with two Pisa/IIT SoftHands, presented in the previous section. It has been developed as a system to be employed in disaster scenarios, to assist in tasks too hazardous for humans to perform. It showed its potential participating to the DARPA Robotics Challenge [1].

CHAPTER 3

# Self-governing robot interaction planning and control

This chapter aims to describe one of the two main aspects presented in this work, i.e. novel planning and control algorithms that entrust the decision-making process to the robot, in complete absence of human intervention. In the first part, a brief motivation on the need for robot interaction autonomy is presented. Next, we introduce the theoretical contribution of a novel self-tuning impedance controller (Section 3.2), and its implementation along with experimental results in various application fields (Section 3.3).

## 3.1 The urge of robot interaction autonomy

The last decades have highlighted the need for autonomous robotic agents in a wide range of application areas. The importance of the development of such systems has a twofold nature. On the one hand, autonomous robots can represent a substantial benefit in assisting humans to explore and interact with hazardous environments, where the human presence has to remain limited. This sectors include emergency and rescue [47], disaster scenarios [182], space exploration [66], and underwater applications [40]. On the other hand, the strong economic contribution of such robotics and autonomous systems highlights their importance in many other fields, that until a few decades ago were traditionally populated only by humans: industrial manufacturing [153], logistics [84], surveillance [35], maintenance [150], precision farming [143], rehabilitation [110], health care [145], surgery [59], assistive living [173], social interaction [12, 51], education [92], and entertainment [70].

The recent outbreak of the COVID-19 pandemic has underlined even more the urge of robotic systems [195] that can contribute to compensate the disruption of healthcare, social and economic systems of the majority of the countries around the world.

Autonomous systems in this situation can stress their importance in tackling essential challenges in fields as logistics, clinical care, reconnaissance, and also to in keeping active the socioeconomic functions.

All the fields mentioned above require the robots to be endowed with highly adaptive capabilities, since they need to interact with scarcely structured and unknown environments. Towards this direction, research solutions have focused mainly on two aspects, i.e. planning and control. Planning strategies have primarily aimed to avoid the possible contacts with the environment, both endowing mobile robots with adaptive navigation behavior through obstacle avoidance techniques [64, 116, 127, 146], and implementing collision avoidance strategies for fixed base manipulators [63, 74].

Nevertheless, also adaptive robotic behaviors that focused on the adjustments of robot control parameters have been studied. Previous attempts to endow robots with adaptive interaction skills have mainly focused on the development of analytical techniques related to the use of impedance control [4, 48, 124]. However, in the majority of cases, the control parameters are pre-selected by robot programmers based on their experience in carrying out analogous tasks. In such a way, the framework cannot adapt when task conditions change, hence, the full potential of such powerful control techniques cannot be exploited [58,194]. To provide a solution to this shortcoming, adaptive learning techniques have been proposed. In [192], an adaptive impedance controller for upper-limb rehabilitation, based on evolutionary dynamic fuzzy neural network was proposed to regulate the impedance profile between the impaired limb and the robot. However, this method lacks versatility, since the algorithm is limited to a specific task. In a similar work [72], empirical constants had to be set, which reduced the flexibility of the framework. In addition, the desired impedance matrices were assumed to be diagonal, resulting in limited adaptability in selective Cartesian axes. In [75], an online learning approach to tune the impedance parameters exploits recurrent fuzzy wavelet neural network. Variable impedance strategies have been presented in human-robot collaboration, e.g. in comanipulation tasks, with the aim of accommodating the human movement during physical interaction [60]. More generic methods have been introduced by reducing impedance control to position control (through high position loop gains) when there is no interaction [128] with the aim to minimize the error between the desired and actual trajectories [77]. This significantly reduced the system's ability to deliver a distinguished response to the desired (task) and undesired (e.g., collisions) interactions (see also [142]).

As [171] reports in the survey and comparison of impedance control techniques on robotic manipulation, variable impedance controllers need to determine *when* and *how* to vary the impedance parameters. Autonomously carrying out this non-trivial task falls within the primary aims of this chapter.

## 3.2 A self-tuning impedance controller

In the context of impedance adaptation, one of the biggest challenges resides in making such techniques work properly in unstructured environments, by distinguishing expected interactions from external disturbances. Taking into account this consideration will enable adaptation of the impedance parameters only to the interactions that are expected, while achieving a compliant behavior in response to the external disturbances

to avoid the generation of unnecessary high interaction forces (e.g. collisions with the environment).

To overcome this shortcoming, we propose a novel self-tuning impedance controller that is enabled only when an interaction is expected, exploiting an interaction expectancy value. The latter can be retrieved based on the task phase information in conjunction with visual feedback interaction values, i.e. when the robot end-effector enters in areas where a contact with the environment is predicted. We will show that in the first version of the presented control strategy the tuning of the impedance parameters is achieved in selective axes of the Cartesian space, while in a more mature version of this method, the impedance regulation can be achieved also along the direction of the motion.

### 3.2.1 Impedance regulation along Cartesian axes

With this strategy, we aim to endow the robot with the ability to adaptively tune the impedance parameters, i.e. stiffness and damping, in selective axes of the Cartesian space based on end-point sensory data, to avoid unnecessary stiffening/complying of the remaining axes. Figure 3.1 shows the block scheme of the control strategy.

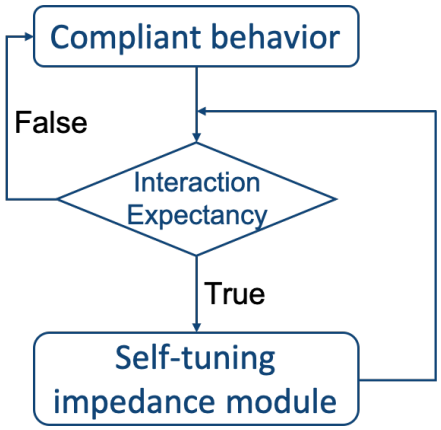


**Figure 3.1:** *Block scheme of the proposed self-tuning impedance control strategy.*

As presented in Section 2.2, in a torque-controlled robot, the relation between the external torques $\boldsymbol{\tau}_{ext}$ and the Cartesian external forces $\boldsymbol{F}_c \in \mathbb{R}^6$ is defined as:

$$\boldsymbol{\tau}_{ext} = \boldsymbol{J}(\boldsymbol{q})^\top \boldsymbol{F}_c + \boldsymbol{\tau}_{st}, \tag{3.1}$$

where $\boldsymbol{q} \in \mathbb{R}^n$ is the joint angles vector, with $n$ representing the number of joints, $\boldsymbol{J} \in \mathbb{R}^{6\times n}$ is the robot arm Jacobian matrix, and $\boldsymbol{\tau_{st}}$ the second task torques projected onto the null-space of $\boldsymbol{J}$. To achieve a compliant behavior, the following relation between the Cartesian displacement and the Cartesian desired forces is established:

$$\boldsymbol{F}_c = \boldsymbol{K}(\boldsymbol{X}_d - \boldsymbol{X}_a) + \boldsymbol{D}(\dot{\boldsymbol{X}}_d - \dot{\boldsymbol{X}}_a), \tag{3.2}$$

where $\boldsymbol{K} \in \mathbb{R}^{6\times6}$ and $\boldsymbol{D} \in \mathbb{R}^{6\times6}$ represent respectively the Cartesian stiffness and damping matrix, $\boldsymbol{X}_d$ and $\boldsymbol{X}_a \in \mathbb{R}^6$ the Cartesian desired and actual position, $\dot{\boldsymbol{X}}_d$ and $\dot{\boldsymbol{X}}_a \in \mathbb{R}^6$ their corresponding velocity profiles.

By default, the presented robot self-tuning impedance controller aims to achieve compliant Cartesian and nullspace behaviors. To this end, the default $\boldsymbol{K}$ and $\boldsymbol{K}_n$ matrices need to be selected to achieve a good tracking performance in the presence of joint friction, while providing a compliant response in case of accidental collisions. $\boldsymbol{K}_n$ is a constant nullspace stiffness matrix used in the secondary task of the self-tuning Cartesian impedance controller. To compute the secondary task joint torques, a stiffness-consistent nullspace projection [49] can be used. Since in this first version of the controller the tuning of the parameters is carried out only along the Cartesian axes, the nullspace stiffness can be kept as default all the time. This assumption is justified by the fact that most manipulation tasks are executed at the robot endpoint.

In order to retrieve if a physical interaction between the robot and the object/environment is to happen, we entrust this decision to a module that outputs a Boolean value named *interaction expectancy* value ($\boldsymbol{I}_e$), based on which the robot will keep the compliant behavior described above, or will tune the impedance parameters as explained in the next paragraph. This value is the result of a Boolean rule that correlates two other Boolean values. One of those, named *task interaction* value ($\boldsymbol{I}_{task}$), is defined based on the task phase, i.e. it is set to True only in the manipulation phases where a priori an external interaction is expected. The other one, named *interaction field* value ($\boldsymbol{I}_{field}$), is defined by a vision module that asserts if the robot end-effector resides within the interaction field area, i.e. the subspace that incorporates the object to be manipulated. These two values are the inputs of the following Boolean logic rule:

$$\boldsymbol{I}_e = \boldsymbol{I}_{task} \ \wedge \ \boldsymbol{I}_{field}. \tag{3.3}$$

As can be inferred from this formula, based on the logical AND operator, the *interaction expectancy* value $\boldsymbol{I}_e$ will be triggered only when the two conditions are satisfied at the same time.

When $\boldsymbol{I}_e$ is True, and so an expected interaction with the environment is detected, the self-tuning impedance module is activated. The Cartesian stiffness matrix $\boldsymbol{K}_c$, and consequently the damping matrix $\boldsymbol{D}_c$[1], are therefore subject to changes (increase or decrease), depending on the external forces sensing. We define as $\boldsymbol{\Delta F}_{ext,t}$ the variation of the external forces detected by the robot at time $t$ with respect to the ones measured in the previous control loop:

$$\boldsymbol{\Delta F}_{ext,t} = \boldsymbol{F}_{ext,t} - \boldsymbol{F}_{ext,t-1}. \tag{3.4}$$

First case: if $\boldsymbol{\Delta F}_{ext,t} < 0$, it means that the external forces coming from the interaction with the environment are going against the robot movement, and therefore the impedance parameters need to be increased on a certain Cartesian axis to ensure the motion is completed in that particular direction in a precise manner. For instance, when the robot picks a heavy object, the loading effect will introduce deviations from the desired trajectory. In this scenario the stiffness parameters are regulated as following:

$$\boldsymbol{\Delta X}_t = |\boldsymbol{X_d} - \boldsymbol{X_a}|, \tag{3.5}$$

$$\boldsymbol{K}_{c,t} = \boldsymbol{K}_{c,t-1} + \boldsymbol{\alpha \Delta X}_t \boldsymbol{\Delta T} \tag{3.6}$$

$$\text{subject to} \quad -\boldsymbol{f}_{max} < \boldsymbol{F}_{ext,t} < \boldsymbol{f}_{max}$$

[1] $\boldsymbol{K}_c \in \mathbb{R}^{3\times3}$ and $\boldsymbol{D}_c \in \mathbb{R}^{3\times3}$ represent the translational part of $\boldsymbol{K} \in \mathbb{R}^{6\times6}$ and $\boldsymbol{D} \in \mathbb{R}^{6\times6}$ introduced in Eq. 3.2.

$$\text{subject to } \boldsymbol{K}_{c,min} < \boldsymbol{K}_{c,t} < \boldsymbol{K}_{c,max},$$

where $\boldsymbol{\alpha}$ is a coefficient to be set, $\boldsymbol{\Delta T}$ the control loop sample time (here used as a constant parameter), $\boldsymbol{f}_{max}$ the maximum allowed interaction force level, $\boldsymbol{K}_{c,min}$ the minimum Cartesian stiffness value and $\boldsymbol{K}_{c,max}$ the maximum Cartesian stiffness value.

Second case: if $\boldsymbol{\Delta F}_{ext,t} > 0$, the external forces are favoring the robot motions, so the impedance parameters can be decreased on a certain Cartesian axis, switching back to compliant mode. For instance, when an object is laid down on a surface before being released, the robot needs to adapt to the surface forces. This will avoid the generation of high interaction forces between the object and the environment. In this case, the stiffness parameters are decreased according to the following law:

$$\boldsymbol{K}_{c,t} = \boldsymbol{K}_{c,t-1} - \boldsymbol{\beta \Delta F}_{ext,t}\boldsymbol{\Delta T} \tag{3.7}$$

$$\text{subject to } \boldsymbol{\Delta X}_t < \boldsymbol{\delta x}_{max}$$

$$\text{subject to } \boldsymbol{K}_{c,min} < \boldsymbol{K}_{c,t} < \boldsymbol{K}_{c,max},$$

where $\boldsymbol{\beta}$ is a coefficient to be set, $\boldsymbol{\Delta F}_{ext,t}$ is defined as in Eq. 3.4 and $\boldsymbol{\delta x}_{max}$ represents the maximum allowed displacement error from the desired trajectory, i.e. $\boldsymbol{\Delta X}_t$.

In both the situations presented above, the Cartesian damping matrix $\boldsymbol{D}_c$ is tuned online based on the resulting $\boldsymbol{K}_c$:

$$\boldsymbol{D}_c = \boldsymbol{\Lambda}_* \boldsymbol{D}_{diag} \boldsymbol{K}_{adj*} + \boldsymbol{K}_{adj*} \boldsymbol{D}_{diag} \boldsymbol{\Lambda}_*, \tag{3.8}$$

where $\boldsymbol{D}_{diag}$ is the diagonal matrix containing the damping factors ($\zeta = 0.7$), $\boldsymbol{K}_{adj*}\boldsymbol{K}_{adj*} = \boldsymbol{K}_c$ and $\boldsymbol{\Lambda}_*\boldsymbol{\Lambda}_* = \boldsymbol{\Lambda}$, where $\boldsymbol{\Lambda}$ is the desired end-effector mass matrix [8].

The experimental results we carried out to validate this controller can be found in Section 3.3.1.

### 3.2.2 Impedance regulation along the motion vector

In order to enhance the effectiveness of the former controller, we provide hereafter a solution to regulate the impedance parameters only along the direction of the motion through a multi-axes self-tuning impedance controller, keeping a compliant profile on the remaining directions. This approach guarantees more flexibility, since the adaptation is not limited to the Cartesian axes. With the aim of providing a clearer overview of this concept, Figure 3.2 presents a sketch of a robotic manipulator which interacts with various materials placed in a box, here representing the *interaction expectancy* area. When the robot is outside this area, the impedance profile is kept compliant as can be noticed by the length of the black arrows representing the stiffness amplitude on the direction of the motion. On the contrary, when the trajectory falls within the mentioned area, the stiffness assumes different values based on the properties of the materials.

In the previous section, we already demonstrated how to detect if an interaction with the environment is expected. In the same way, when no interaction is foreseen, being $\boldsymbol{I}_e$ False, the Cartesian stiffness matrix $\boldsymbol{K}_c$ in Eq. 3.2 is set to a default diagonal matrix with all the non-zero coefficients set to $\boldsymbol{k}_{min}$ (minimum) to deliver a compliant behavior. Note that, the impedance values to render softness, however, has to be chosen based on a trade-off between the position tracking accuracy (affected by the existence

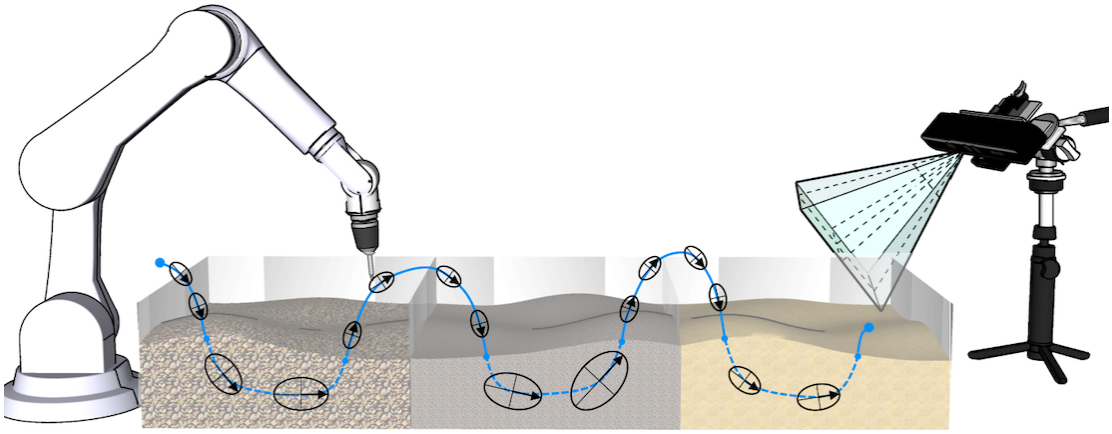

**Figure 3.2:** *The robotic arm explores the materials in its workspace, identifying and self-tuning its impedance parameters along the directions of interaction. These are represented by the principal axis of the geometric ellipsoids depicted in the figure. Longer arrows represent higher Cartesian stiffness and damping values.*

of unmodelled dynamics such as friction) and the force response, if an unexpected interaction occurs.

On the other hand, in correspondence of a positive value of $\boldsymbol{I}_e$, the Cartesian stiffness matrix $\boldsymbol{K}_c$ and consequently the damping matrix $\boldsymbol{D}_c$ are subject to changes increasing (or decreasing) the impedance parameters only along the direction of the desired movement defined by:

$$\overrightarrow{\boldsymbol{P}} = \boldsymbol{X}_{\boldsymbol{d},\boldsymbol{t}} - \boldsymbol{X}_{\boldsymbol{d},\boldsymbol{t-1}}, \tag{3.9}$$

(which can also be calculated from $\dot{\boldsymbol{X}}_d$) and keeping a compliant behavior, set to $\boldsymbol{k}_{min}$ and $\boldsymbol{d}_{min} = 2\zeta\sqrt{\boldsymbol{k}_{min}}$ [8], along the other axes. With this aim, the stiffness and damping matrices, as being symmetric and positive definite, can be expressed by:

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*, \tag{3.10}$$

which is known by the Singular Value Decomposition (SVD). In this way, the desired stiffness and damping, calculated with respect to the reference frame of desired motion vector $\overrightarrow{\boldsymbol{P}}$, can be projected onto the reference frame of the robot base, through such decomposition. In Eq. 3.10, $\boldsymbol{U} \in \mathbb{R}^{3\times3}$ and $\boldsymbol{V} \in \mathbb{R}^{3\times3}$ are orthonormal bases, and $\boldsymbol{\Sigma} \in \mathbb{R}^{3\times3}$ is a diagonal matrix whose elements are the singular values of matrix $\boldsymbol{A}$ sorted in decreasing order and representing the principal axes amplitudes of the resulting geometric ellipsoid. The columns of matrix $\boldsymbol{U}$ form a set of orthonormal vectors, which can be regarded as basis vectors.

Since our aim is to project the impedance onto direction of interaction, the first column of $\boldsymbol{U}$ is formed by the desired motion vector $\overrightarrow{\boldsymbol{P}}$, while the second and the third ones are derived from the first in such a way they form an orthonormal basis. Given that the Hermitian transpose $\boldsymbol{V}^* \in \mathbb{R}^{3\times3}$, and the resulting matrix $\boldsymbol{A}$, that represents the impedance values, is positive definite, we have that:

$$\boldsymbol{V}^* = \boldsymbol{V}^\top, \tag{3.11}$$

$$\boldsymbol{V} = \boldsymbol{U}. \tag{3.12}$$

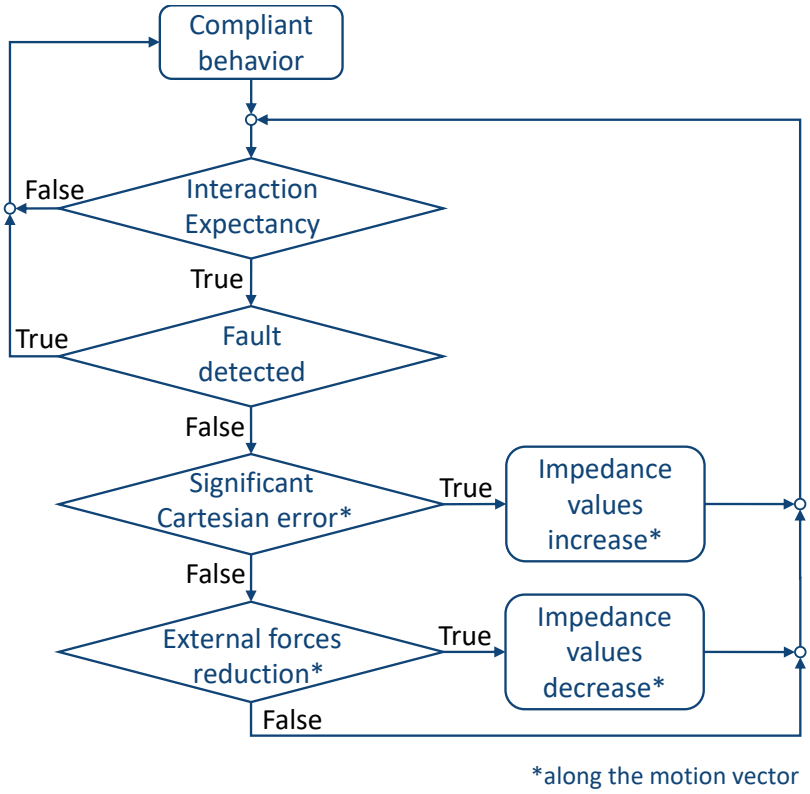


**Figure 3.3:** *The self-tuning impedance algorithm flow chart: if no interaction is expected, the robot keeps a compliant behavior, otherwise the impedance parameters can be subject to changes, if no faults are detected.*

Combining Eqs. 3.10, 3.11 and 3.12, we can derive the stiffness and the damping matrix as:

$$\boldsymbol{K}_c = \boldsymbol{U}\boldsymbol{\Sigma}_k\boldsymbol{U}^\top, \tag{3.13}$$

$$\boldsymbol{D}_c = \boldsymbol{U}\boldsymbol{\Sigma}_d\boldsymbol{U}^\top, \tag{3.14}$$

where the diagonal matrix $\boldsymbol{\Sigma}_k$ and $\boldsymbol{\Sigma}_d$ coefficients are respectively the desired stiffness and damping coefficients along the direction of the vectors composing the $\boldsymbol{U}$ basis. They are diagonal matrices defined by:

$$\boldsymbol{\Sigma}_k = \begin{bmatrix} \boldsymbol{k}_{st} & 0 & 0 \\ 0 & \boldsymbol{k}_{min} & 0 \\ 0 & 0 & \boldsymbol{k}_{min} \end{bmatrix}, \tag{3.15}$$

$$\boldsymbol{\Sigma}_d = \begin{bmatrix} \boldsymbol{d}_{st} & 0 & 0 \\ 0 & \boldsymbol{d}_{min} & 0 \\ 0 & 0 & \boldsymbol{d}_{min} \end{bmatrix}, \tag{3.16}$$

where $\boldsymbol{k}_{st}$ is the self-tuning stiffness coefficient to be set along the motion vector $\overrightarrow{\boldsymbol{P}}$ and $\boldsymbol{d}_{st}$ its correspondent damping element.

Similarly as the case of impedance tuning along the Cartesian axes, Eq. 3.6 can be rewritten to calculate $\boldsymbol{k}_{st}$ as:

$$\boldsymbol{k}_{st,t} = \boldsymbol{k}_{st,t-1} + \boldsymbol{\alpha}\boldsymbol{\Delta P}\boldsymbol{\Delta T}, \tag{3.17}$$

$$\boldsymbol{\Delta P} = |\boldsymbol{\Delta X} \cdot \widehat{\boldsymbol{P}}|, \tag{3.18}$$

$$\boldsymbol{\Delta X} = \boldsymbol{X_d} - \boldsymbol{X_a}, \tag{3.19}$$

where $\boldsymbol{\alpha}$ is the update parameter, $\boldsymbol{\Delta P}$ is the absolute value of the Cartesian error $\boldsymbol{\Delta X}$ projected onto the direction of the motion vector $\overrightarrow{\boldsymbol{P}}$, and normalized $\widehat{\boldsymbol{P}}$. $\boldsymbol{\Delta T}$ is the control loop sample time, that is a constant parameter. To avoid unnecessary and continuous changes, $\boldsymbol{k}_{st}$ is adapted only when $\boldsymbol{\Delta P}$ is beyond a certain threshold, defined as $\boldsymbol{\Delta P}_t$. The main reason behind this choice resides in the difficulties in achieving a small error, between the desired and actual position, in impedance controlled robots. This would be only possible by setting high impedance gains, thus leading to a behavior that could be compared to position control. On the contrary, through the introduction of this threshold, we let a small margin of error, avoiding any unneeded impedance variation if not required by the task, and arresting the impedance growth when a desired accuracy is achieved. In addition, unmodeled robot dynamics and small amount of friction at joints can also be the cause of this error, while the aim of the controller is to compensate only the error related to the task.

To enhance the effectiveness of adaptation in unknown situations, it is fundamental to converge to the proper impedance values in a reasonably short time. To this end, an effective choice of the parameter $\boldsymbol{\alpha}$ has to be taken. On the one hand, selecting a high value for this parameter would lead to a rapid convergence, on the other, this could cause needless stiffening of the robot that must be avoided. Hence, based on the situation, this parameter can assume very different values. In Section 3.3.2, we will provide the details on how this can be calculated.

So far, we described the positive variation of the impedance parameters, but, as illustrated in the previous section, there are also cases in which the impedance parameters adaptation has to be carried out in the opposite way, i.e. decreasing them. For instance, when $\boldsymbol{I}_e$ is still True but the external forces variation becomes positive, the robot should become more compliant. Similarly as in Eq. 3.7, the variation of $\boldsymbol{k}_{st}$ is regulated with:

$$\boldsymbol{k}_{st,t} = \boldsymbol{k}_{st,t-1} - \boldsymbol{\beta}\boldsymbol{\Delta F}_{ext,t}\boldsymbol{\Delta T}, \tag{3.20}$$

where $\boldsymbol{\beta}$ is given by $\boldsymbol{\alpha}$ scaled by a factor of $10^{-2}$, to implement a similar rate of adaptation as in Eq. 3.17, and the variation of the external forces along the motion vector is given by:

$$\boldsymbol{\Delta F}_{ext,t} = (\boldsymbol{F}_{ext,t} - \boldsymbol{F}_{ext,t-1}) \cdot \widehat{\boldsymbol{P}}. \tag{3.21}$$

Notice that, to avoid unnecessary changes caused by negligible force sensing difference, the positiveness of $\boldsymbol{\Delta F}_{ext,t}$ is defined considering a small $\varepsilon$.

Algorithm 1 proposes the pseudo-code of the presented method, where the impedance adaptation is subject also to the check of potential faults, along with the positiveness of the *interaction expectancy* value $\boldsymbol{I}_e$.

**Algorithm 1** Self-tuning impedance algorithm

**Input:** $\overrightarrow{\boldsymbol{P}}$
**Output:** $\boldsymbol{K}_c, \boldsymbol{D}_c$
*Initialization* :
$\boldsymbol{\Sigma}_k = \boldsymbol{k}_{min}\boldsymbol{I}_{3\times3},\ \boldsymbol{\Sigma}_d = \boldsymbol{d}_{min}\boldsymbol{I}_{3\times3}$
$\boldsymbol{k}_{st,0} = \boldsymbol{k}_{min},\ \boldsymbol{d}_{st,0} = \boldsymbol{d}_{min}$
*Control loop* :
$\boldsymbol{I}_e = \boldsymbol{I}_{task} \wedge \boldsymbol{I}_{field}$
**if** ($\boldsymbol{I}_e \wedge$ *No fault detected*) **then**
$\widehat{\boldsymbol{P}} = \dfrac{\overrightarrow{\boldsymbol{P}}}{\|\overrightarrow{\boldsymbol{P}}\|_2}$
$\boldsymbol{\Delta P} = |(\boldsymbol{X_d} - \boldsymbol{X_a}) \cdot \widehat{\boldsymbol{P}}|$
**if** ($\boldsymbol{\Delta P} \geq \boldsymbol{\Delta P}_t$) **then**
$\boldsymbol{k}_{st,t} = \boldsymbol{k}_{st,t-1} + \boldsymbol{\alpha}\boldsymbol{\Delta P}\boldsymbol{\Delta T}$
**else if** ($\boldsymbol{\Delta F}_{ext} \geq \varepsilon$) **then**
$\boldsymbol{k}_{st,t} = \boldsymbol{k}_{st,t-1} - \boldsymbol{\beta}\boldsymbol{\Delta F}_{ext,t}\boldsymbol{\Delta T}$
**end if**
$\boldsymbol{\Sigma}_{k,1,1} = \boldsymbol{k}_{st,t}$
$\boldsymbol{\Sigma}_{d,1,1} = 2\zeta\sqrt{\boldsymbol{k}_{st,t}}$
$\boldsymbol{U} = getOrthonormalBasis(\overrightarrow{\boldsymbol{P}})$
$\boldsymbol{K}_c = \boldsymbol{U}\boldsymbol{\Sigma}_k\boldsymbol{U}^\top$
$\boldsymbol{D}_c = \boldsymbol{U}\boldsymbol{\Sigma}_d\boldsymbol{U}^\top$
**else**
$\boldsymbol{K}_c = \boldsymbol{k}_{min}\boldsymbol{I}_{3\times3}$
$\boldsymbol{D}_c = \boldsymbol{d}_{min}\boldsymbol{I}_{3\times3}$
**end if**
**return** $\boldsymbol{K}_c, \boldsymbol{D}_c$

## 3.3 Implementation and results

To validate the theoretical contribution presented in the former section, we carried out several experiments in various application fields. Hereafter we demonstrate how to blend the impedance parameters adaptation with flexible planning strategies.

### 3.3.1 Disaster response

One of the fields that received a great attention in the development of robot autonomous interaction behavior is represented by the disaster response scenario. In fact, in the past decades, disasters caused by human negligences (e.g. Chernobyl nuclear accident), and natural catastrophes due to Earth's natural processes (e.g., tsunami, hurricanes, earthquakes, and their consequences) have pointed out the importance of the deployment of autonomous robotics systems that are able to carry out complex tasks in human-engineered environment. To encourage the progress of new investigations in this direction, in 2012, the US Defense Advanced Research Projects Agency (DARPA) announced a robotics competition named DARPA Robotics Challenge (DRC) [1]. To reproduce the simulation of a degraded disaster scenario, a set of manipulation tasks were included in this challenge, as removing debris blocking an entryway, opening doors, attaching a fire hose, using a cutting tool, and turning a valve near a leaking pipe. All these tasks are characterized by high uncertainties. Therefore, we decided to

apply the first version of the self-tuning impedance controller presented in Section 3.2.1 to one of these tasks: the removal of debris blocking a door. The results reported in this section led to the scientific publications [18] and [22].

As previously described, to distinguish expected interactions from external disturbances, the *interaction expectancy* value $\boldsymbol{I}_e$ is defined through the evaluation of two other Boolean values: $\boldsymbol{I}_{field}$ and $\boldsymbol{I}_{task}$. In this framework, the first value is computed by a vision system, by fitting an interaction field around the object to be manipulated or the area of the environment where an interaction is predicted, while the second one depends on the states of an autonomous Finite State Machine. In the following paragraphs, we describe the details of these two subsystems, before showing the experimental results obtained through their implementation.

#### Vision system

In order to yield the *interaction field* value $\boldsymbol{I}_{field}$, the framework needs to localize the objects to be manipulated, along with their grasping pose, that can be identified as $\hat{\mathbf{x}}_g$,$\hat{\mathbf{y}}_g$,$\hat{\mathbf{z}}_g$. Since the focus of this work is to remove debris from a door, we consider objects that have stick-like shape and that lie in front of a planar segment.

To sense the environment, we use a depth camera calibrated with respect to the robot end-effector. This camera provides, with a frequency of $30Hz$, the point cloud data, that allows us to segment the dominant plane using the RANdom SAmple Consensus (RANSAC) method [61]. Like this, it is possible to extract the normal of the plane $\hat{\mathbf{n}}_w$, that is identified in cyan color in Figure 3.4, and to segment all the points above the plane to different Euclidean clusters [94, 164]. Each cluster in the point cloud constitutes a potentially different object in the scene, and through RANSAC we can fit a 3D line for each cluster to detect which of those could represent a debris. The axis of

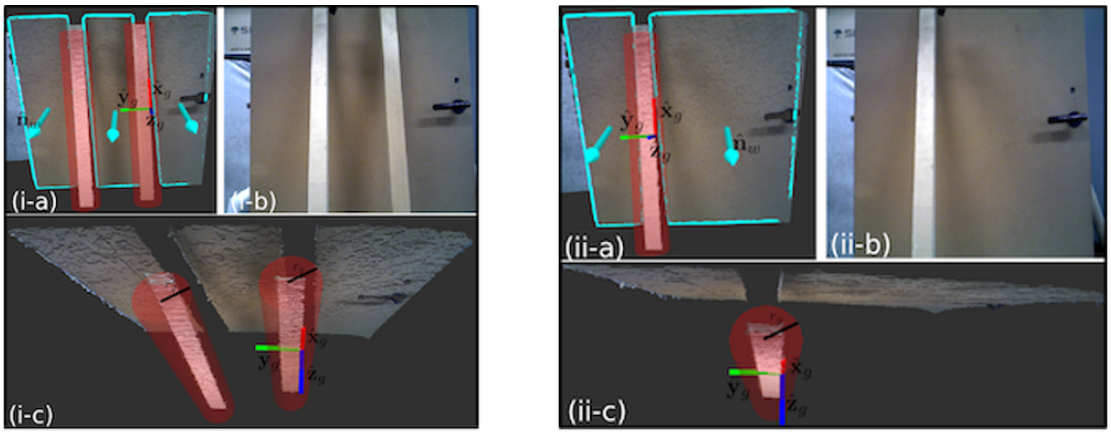


**Figure 3.4:** *On the left, two debris are detected in front of a door, (i-a) shows the point cloud and the door normal vector $\hat{\mathbf{n}}_w$ colored in cyan, the grasp frame of the right-most debris ($\hat{\mathbf{x}}_g$, $\hat{\mathbf{y}}_g$, $\hat{\mathbf{z}}_g$), and in red the interaction field cylinder around each debris. In (i-b), it is depicted the raw image of the scene detected by the camera, and in (i-c) the view of the point cloud from the top, highlighting the grasp axes and the interaction field cylinder with radius $r_g$. On the right, similar information are presented for the case with only one debris.*

this 3D line, illustrated in red in Figure 3.4, represents the $\hat{\mathbf{x}}_g - axis$ of the grasping frame, while the other two axes can be reconstructed as follows:

$$\hat{\mathbf{y}}_g = \hat{\mathbf{n}}_w \times \hat{\mathbf{x}}_g, \tag{3.22}$$

$$\hat{\mathbf{z}}_g = \hat{\mathbf{x}}_g \times \hat{\mathbf{y}}_g, \tag{3.23}$$

where $\hat{\mathbf{n}}_w$ is the plane normal vector and $\times$ is the cross product operator.

This frame is generated so as to facilitate the robotic hand grasping, needed to move away the debris. In fact, the height is maintained fixed with respect to the robot base frame, and the frame is localized at the edge of the object (left edge for left-handed grasps and right edge in the other case). This is done by merely extracting the point cloud nearest neighborhood for each cluster at a fixed height, and then selecting the left/right-most point.

As discussed above, this unit is also in charge of providing the state of the *interaction field* value $\boldsymbol{I}_{field}$. To this end, it is possible to identify the interaction field between the robotic end-effector and the object with a cylindrical shape, as shown in red in Figure 3.4. This can be extracted along the fitted 3D line axis for each debris. The cylinder radius $r_g$ is equal to the maximum size of the robotic hand fingers, while its height is identified considering the two extreme 3D points of the clusters extracted after the plane segmentation. Every time the robot end-effector fingers reside within this interaction field, the Boolean value $\boldsymbol{I}_{field}$ is True, while in all the other cases is set to False.

#### Finite State Machine

The second value contributing to the definition of the *interaction expectancy* value is triggered by the active states of a Finite State Machine (FSM), that has been designed to endow the framework with autonomous capabilities. Figure 3.5 graphically shows the FSM, whose states represent the framework motion primitives. The outgoing arrows allow the system to move from one state to the other, and are based on the feedback provided when each motion ends. If the motion is completed with a positive outcome, the FSM moves to the state indicated by the solid arrow, while if the action is halted due to a system failure, the recovery state to be reached is specified by the dashed arrows.

The FSM is characterized by 8 states, involving motion primitives where the robot needs to execute a trajectory and others where it actuates the opening/closure of the end-effector. The FSM control flow starts from the "Homing" state, where the robot is in the center of the workspace ready to initiate the task. This state also represents the recovery state for all the other states that involve the execution of a trajectory. After the vision system has computed the debris grasping frame as described above, the "Reach" state is executed, and the robot approaches the object.

Since in this work we are trying to simulate a situation in which many uncertainties can arise, we need to ensure that the pose detected by the vision module is accurate. In fact, in disaster response applications, the robot workspace may not be clear and bright, for instance due to poor lighting conditions or to the smoke caused by a nearby fire. To this end, the FSM next state, named "Adjust", aims to double-check the precision of the debris pose given by the perception unit. In this state, the framework reads the external wrenches sensed at the joint level and translated to the Cartesian space as:

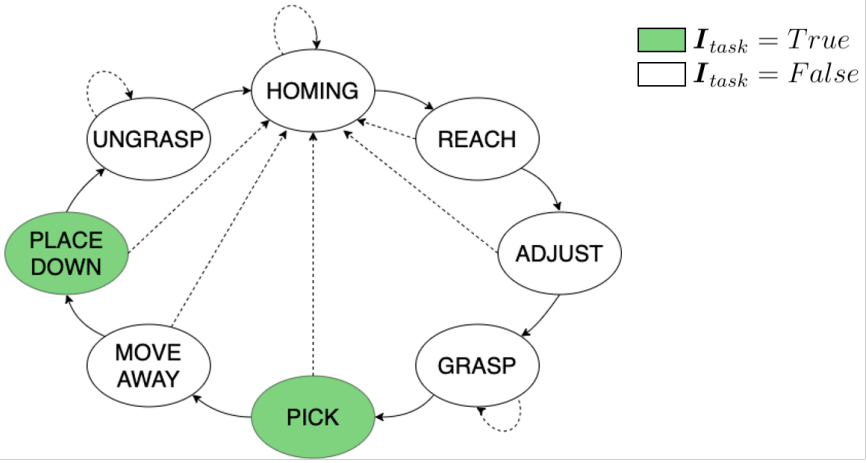


**Figure 3.5:** *The regulation of the framework phases is entrusted to a Finite State Machine, where each state represents a motion primitive and the outgoing arrows indicates the following motion to be executed in case of a successful execution (solid arrows) or a negative outcome (dashed arrows). The states highlighted in green are the ones where the task interaction Boolean value $\boldsymbol{I}_{task}$ is positive.*

$$\boldsymbol{F}_{c,ext} = \boldsymbol{J}^{+\top} \cdot \boldsymbol{\tau}_{ext} \tag{3.24}$$

where $\boldsymbol{J}^{+\top}$ represents the transpose of Jacobian pseudoinverse matrix and $\boldsymbol{\tau}_{ext}$ is the joint external torques vector. If no forces[2] are detected on the axis where the robot is supposed to be in contact with the debris, a movement along that axis is performed until external forces are detected, i.e. when $\boldsymbol{\Delta F}_{ext} \neq 0$. Performing end-effector movements against the object to sense the mentioned forces could lead to an accidental fall of the object, if it is placed in a precarious position, but since these adjustments are made with a very high compliance, the end-effector will not displace the object from its original pose.

After we ensured that a contact between the robotic hand and the debris surface has been established, the FSM switches to its following state, "Grasp", where the robot end-effector closure is invoked. The task goal is then to move the debris away to a safe portion of the workspace. To this end, the "Pick" state takes place, where the object is brought in front of the robot. In this phase the Boolean value $\boldsymbol{I}_{task}$ becomes positive since an interaction on the vertical axis is predicted, due to the external gravity load of the debris, as also highlighted by the corresponding green colored state in Figure 3.5. Next, the object is moved to the right (or to the left) through execution of the motion primitive of the "Move away" state. Here, the Boolean value related to the task turns to False, since no environmental interaction is foreseen and the impedance parameters of the controller do not need to be subject to any change. This ensures that, if during this motion a collision occurs, the robot keeps a compliant profile without leading to any damage to the environment and to the robot itself.

Once the robot has safely relocated the object, the "Place down" motion is responsible of reconfiguring its pose in a way it will not hinder future robot operations. This is done by rotating the debris upper-end so as to point in the robot opposite direction and by lowering it until a contact with the ground is detected. With this state, the $\boldsymbol{I}_{task}$

[2]From now on, the external wrenches term $\boldsymbol{F}_{c,ext}$ will consider only the force component on the three Cartesian axes.

value is activated again, so as to let the robot assume a more compliant behavior when the object makes contact with the ground. This is made possible since, in Eq. 3.4, $\boldsymbol{\Delta F}_{ext,t}$ assumes a positive value. Finally the FSM switches to the "Ungrasp" state, where the debris is released, and it comes back to the initial state to potentially restart this procedure and remove other debris.

#### Experimental results

To validate the theoretical content and the framework implementation presented above, we carried out experiments using a KUKA LBR IIWA 14 robotic arm equipped with the underactuated Pisa/IIT SoftHand, presented in Chapter 2. The camera to stream data to the vision module was an ASUS Xtion Pro RGB-D sensor. It was attached in the front part of the robot arm base, and calibrated with respect to its end-effector. The default values of the stiffness matrix $\boldsymbol{K}_c$ diagonal components were set to $100N/m$ on the translational ones, and to $20Nm/rad$ on the rotational ones. The update rate parameters in Eqs. 3.6 and 3.7 have been selected experimentally so as to minimize the Cartesian error within the trajectory period of the current state: $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ were set to 5000 and 50, respectively.

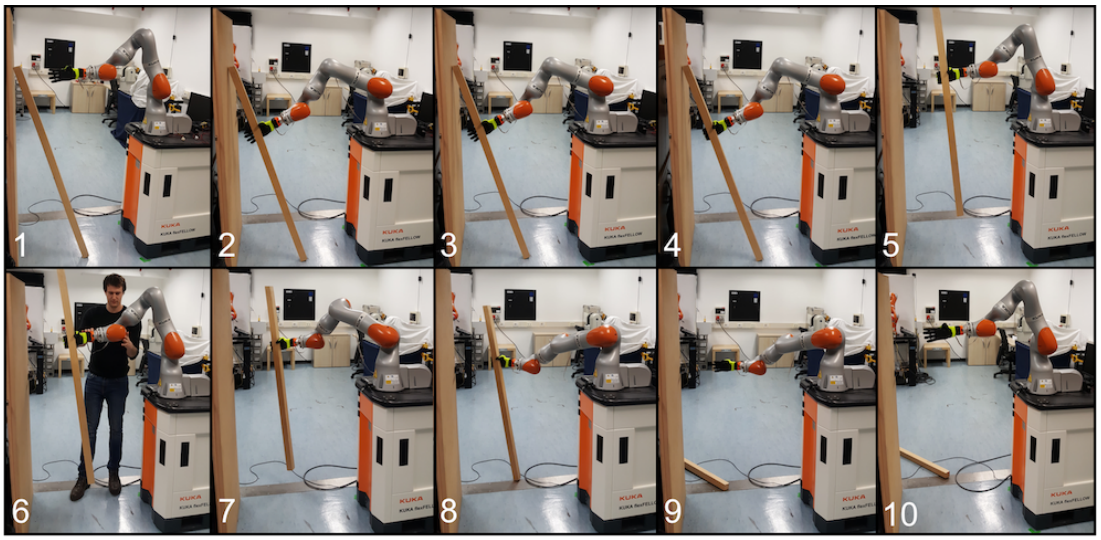


**Figure 3.6:** *The debris removal task has been performed with the KUKA LBR IIWA 14 robotic arm. FSM states: Homing (1), Reach (2), Adjust (3), Grasp (4), Pick (5), Unexpected external disturbances (6), Move Away (7), Place Down (8), Ungrasp (9), Homing (10).*

Figure 3.6 graphically shows the different phases of the task execution with reference to the FSM states. Figure 3.7, that plots the data collected during the experiment, illustrates the external forces in the Cartesian space $\boldsymbol{F}_{ext}$ (a), the Cartesian error between the desired and the actual robot end-effector pose $\boldsymbol{\Delta X}$ (b), the translational components of the Cartesian stiffness $\boldsymbol{K}_c$ (c), and the two Boolean values implied in the determination of the interaction expectancy value $\boldsymbol{I}_e$ (f), i.e. the FSM interaction value $\boldsymbol{I}_{task}$ (d), and the interaction field value $\boldsymbol{I}_{field}$ (e).

To describe the experimental results, we can follow the natural execution of the FSM states. Figure 3.6.1 shows the robot in the initial "Homing" state, that represents also the recovery pose the robot will assume in case of a failure in execution of the other

motion primitives. In Figure 3.6.2, the robot approaches the grasp pose received in input by the vision module ("Reach" state). In this state the robot end-effector follows within the area identified as the debris *interaction field* and $\boldsymbol{I}_{field}$ is triggered to a positive value, as can be noticed in the relative plot in Figure 3.7.e at $Time = 8s$. Nevertheless, the interaction expectancy does not get enabled since the value relative to the task is still False, and will not be activated before the switching to the "Pick" state.

Next, the robot ensures to be in contact with the debris by analyzing the sensed external forces (Figure 3.6.3, "Adjust" state), and, through the SoftHand closure, grasps the object (Figure 3.6.4, "Grasp" state). The FSM then switches to the "Pick" state, illustrated in Figure 3.6.5, in which the interaction value related to the task is enabled as can be noticed in the relative plot (Figure 3.7.d), at $Time = 13s$. Since in this phase both the interaction values are set to True, also the *interaction expectancy* value $\boldsymbol{I}_e$ is enabled and the controller impedance parameters become subject to changes. Being $\boldsymbol{\Delta F}_{ext,t} < 0$ on the *z-axis* (Figure 3.7.a), $\boldsymbol{K_c}(z)$ is increased following Eq. 3.6 (Figure 3.7.c).

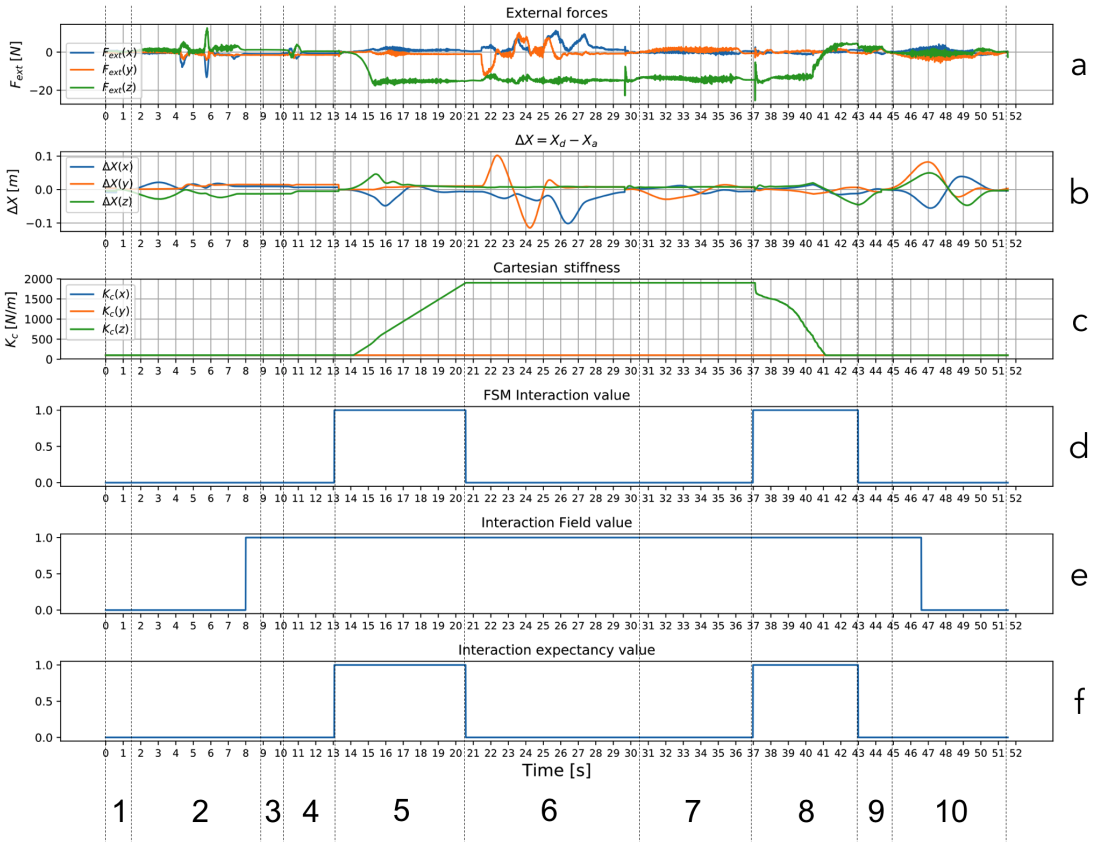


**Figure 3.7:** *The data logged during the experiment represent: external forces in the Cartesian space (a), the Cartesian displacement between desired and actual position (b), the curve trend of the Cartesian stiffness (c), FSM interaction value (d), interaction field value (e) and interaction expectancy value (f). To enhance the readability, the dashed vertical lines have been added to indicate when the different FSM states took place. The numbering follows the FSM states of Figure 3.6.*

With the switching between the "Pick" and the "Move away" state, the interaction value given by the FSM becomes False, and so does the global *interaction expectancy*

value $\boldsymbol{I}_e$. To demonstrate that, when the latter is not enabled, the Cartesian stiffness does not change even in view of unexpected external interactions, we let the robot remain in the current configuration for 10 seconds, allowing a person to interact with it. Figure 3.6.6 shows a subject while perturbing the robotic arm along all the Cartesian axes, as can be noticed in the plot related to the external forces (Figure 3.7.a) from time $20.5s$ to time $30.5s$. For the entire duration of this time window, the Cartesian stiffness is not affected by such deviations, regardless of the force and position displacements.

Subsequently, the natural control flow of the FSM is reestablished with the "Move Away" state (Figure 3.6.7). After having relocated the debris, in the next state (Figure 3.6.8, "Place Down" state), the task interaction value $\boldsymbol{I}_{task}$ is enabled one more time. Since the robot end-effector is still within the visual interaction field, also the $\boldsymbol{I}_e$ value gets enabled at $Time = 37s$. Here, the external forces variation is positive on the *z-axis*, this time the Cartesian stiffness $\boldsymbol{K}_c(z)$ decreases following Eq. 3.7, until the lower bound $\boldsymbol{K}_{c,min} = 100N/m$ is reached at $Time = 41s$. In this way, the debris is gently placed on the ground, and the external interaction with the environment is minimized. Together with the FSM switch to the next state, the task interaction value is set back to False, and within the "Ungrasp" state, the robotic hand opens, making the object fall to the ground. As soon as the debris is detected out of the vision interaction field, the relative value $\boldsymbol{I}_{field}$ is disabled. The final setup with the debris on the ground and the robot back to the "Homing" state is depicted in Figure 3.6.10. A video of the experiment is available at [18].

To demonstrate the system flexibility against varying conditions, we performed a similar experiment with another debris of different weight ($0.7Kg$) and length. Figure 3.8 shows the similar results achieved during the "Pick" state, where, in view of the lighter object weight, $\boldsymbol{K}_c(z)$ reaches a maximum value of $\approx$1330$N/m$ in comparison with the $\approx$1900$N/m$ of the first experiment where the weight of the object was $2Kg$.

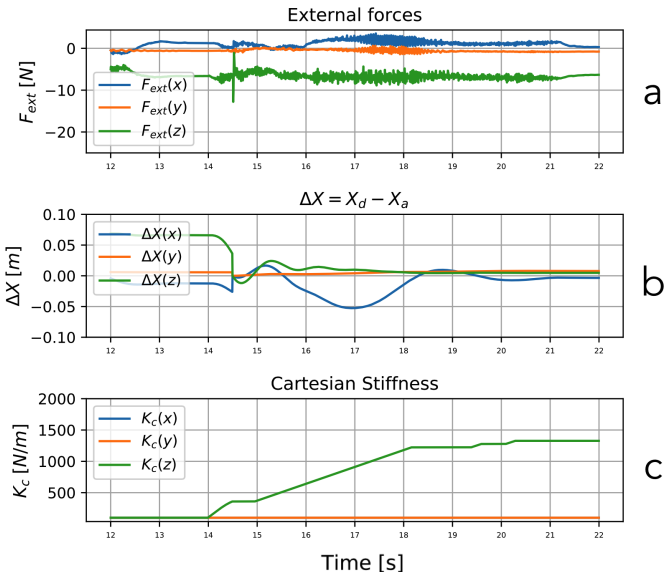


**Figure 3.8:** *The experiment repeated with a lighter object demonstrates how the Cartesian stiffness (c) reaches smaller values in the "Pick" state, in view of a smaller Cartesian displacement between desired and actual position (b). The external forces are shown in (a).*

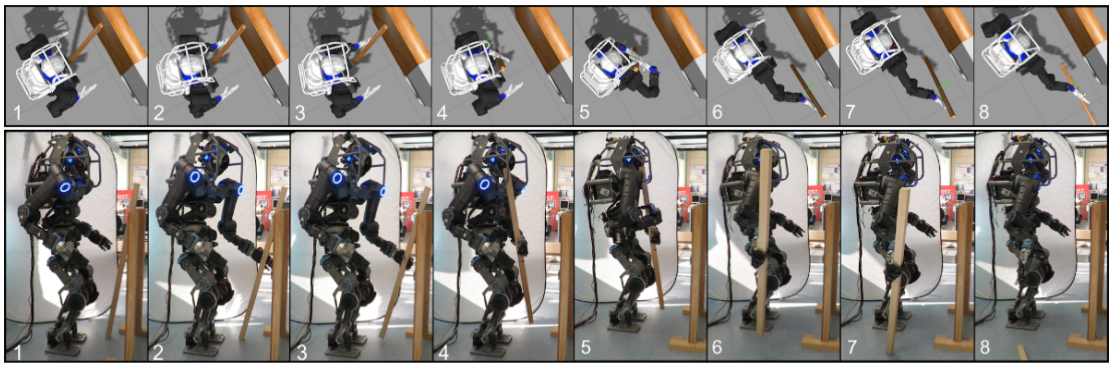


**Figure 3.9:** *The debris removal task performed with the humanoid robot WALK-MAN in the simulation environment Gazebo (top) and on the real robot (bottom). The motion primitives and FSM states depicted are: Homing (1), Reach (2), Grasp (3), Pick (4), Handover (5), Move Away (6), Place Down (7) and Ungrasp (8).*

The plots related to the other states are not reported here, since they are very similar to the ones presented above.

This framework has also demonstrated its effectiveness with the humanoid robot WALK-MAN. Figure 3.9 shows the execution of the phases presented above in a simulation environment and in the real world, respectively, during the final demonstration of the WALK-MAN project. Here an additional state, named "Handover" (step 5 in the figure), was implemented. The robot after having picked the debris with the left hand, hands it over to the right hand before moving it away. This motion primitive can be particularly useful in situations where the debris is not reachable by the arm that is on the side we want it to be relocated. A video of the experiment is available at [22].

### 3.3.2 Agricultural setup

In this section we aim to demonstrate the theoretical methods presented in Section 3.2.2, showing the multi-axes self-tuning impedance controller performing a task in an agricultural setup, highlighting the framework potential in one of the most promising fields in robotics. Here, the quasi-static robot parameters are adapted based on the prior experience gained in interacting with similar environments and the real-time sensory data. As above, the interaction expectancy value is computed through a visual perception unit and based on the task phases regulated by means of a FSM. In addition, this framework is endowed with the capability of detecting and reacting appropriately to unexpected events as faults.

Through the vision module, the framework is able to recognize and localize different materials within the robot workspace and can improve the robot internal knowledge through the analysis of their physical properties. In fact, in this work the robot aim is to *explore* an environment, *identify* its characteristics, and effectively *interact* with it. This behavior is inspired by the humans' way of adapting to their surroundings, by constantly building internal models of the external world, while exploring and identifying it. When interacting with new or similar environments, the prior knowledge is used as a preparatory strategy, while keeping open the possibility of adaption, to update our internal knowledge [99]. The human behavior is also affine to our framework, since when no interaction is expected, humans tend to relax their muscles so as to comply

with unexpected external disturbances (and to minimize energy consumption). In a similar way, the robot assumes a compliant strategy in such situations.

In this section we also present a comparison between the proposed self-tuning impedance controller and non-adaptive techniques, to highlight the full potential of the framework in interacting with uncertain environments. Since the continuous impedance parameters variation could lead to unstable behaviors, we also conducted an analysis to guarantee the stability of the controller. The results reported in this section led to the scientific publications [20] and [19].

Figure 3.10 shows the components that have been developed to build this framework and that will be illustrated hereafter: a visual perception module that locates the materials' positions in the robot workspace, a trajectory planner that provides different kind of motion, and a FSM that coordinates the data exchange between the other units, and regulates the task phases, also triggering the detection of system faults.

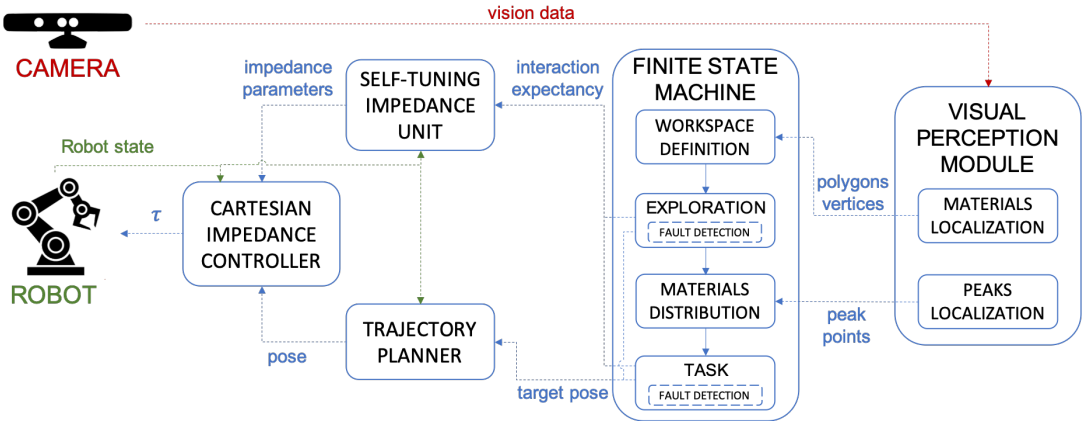


**Figure 3.10:** *The framework embeds different modules. Each one has been developed modularly as a ROS node. The data exchange between the modules is managed via ROS messages on the ROS topics illustrated with dotted lines. The messages in blue are published by the proposed software architecture, while the others represent the robot state (in green) and the vision data provided by the camera (in red).*

#### Vision system

The visual perception unit is responsible of providing information on the different materials that need to be manipulated by the robot. The vision system analyzes the point cloud data coming from an ASUS Xtion PRO sensor to accomplish two main goals. The first is to detect the different materials in the scene, based on RGB data, and their 3D localization inside a convex hull polygon, based on depth data (*Materials localization* sub-module). The second objective is to detect which are highest (i.e. peak) points of every material, based on depth data (*Peaks localization* sub-module). To this end, the camera is installed on a pole facing the workspace from a top-lateral view. The RGB and depth (point cloud) data are provided in an organized $640 \times 480$ grid-based structure at a frequency of $30Hz$ so as to allow fast nearest neighborhood computations.

*Materials Localization:* through this sub-system the vision module provides the polygon vertices of the areas where the materials are localized in the environment. These points are fundamental to build the interaction expectancy area, and therefore to

infer the positiveness of the *interaction field* value $\boldsymbol{I}_{field}$. To identify the location of the materials, we apply a color-based region growing segmentation, after transposing the point cloud data retrieved from the sensor in the robot base frame (*z-axis* upwards and *y-axis* towards left). In order to discard the points that are not to be considered, we filter the data using a pass-through filter to remove points below the robot base, and a crop-box filtering to eliminate the points lying outside the robot workspace. To guarantee the efficiency of this method, we maintain the original grid organization without removing the points. To do so, we set their values to *NaNs*, so as to filter those points out, without compromising the initial grid structure. Next, to classify similar points in clusters, we apply region growing based on color in two steps. First, points are sorted based on their local curvature and selected sequentially as seeds starting from the one with the minimum curvature value, so as to reduce the number of segments. If the color of two different points is similar, they are classified to be part of the same material. This process continues for a seed's neighboring point, until no further neighbors can be classified to the same segment. In the second stage, clusters whose color is close to the neighbor cluster or their size is small, are merged. The image in the center of Figure 3.11 shows the result of this process in which we identify the extreme points in the *xy-plane* (i.e. the 2.5D bounding box) so as to localize the convex hull polygon of each material. These points are the vertices $V_1, V_2, V_3, V_4$ that will be used by the FSM.

*Peaks Localization:* since the materials have already been localized, retrieving their highest point results straightforward. In fact, the peak point $p_i$, for a material $i$, is merely the point with the maximum z-value (Figure 3.11, on the right).

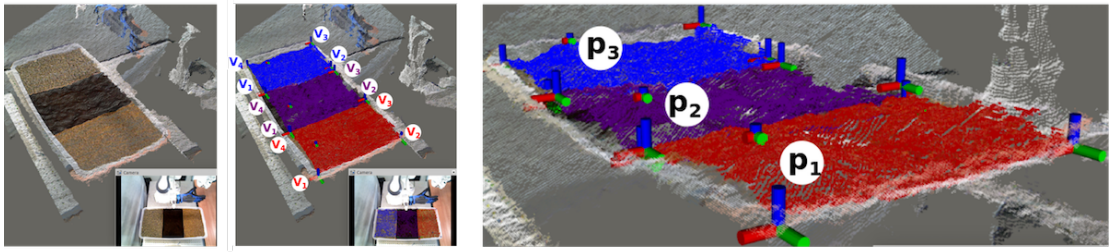


**Figure 3.11:** *The visual perception module retrieves the polygon vertices through the Materials localization sub-unit, and the materials' highest points by means of the Peaks localization sub-unit. On the left the RGB-D point cloud, acquired from the range sensor.*

#### Trajectory planner

Based on the motion required by the task, this unit offers the planning of three types of trajectories:

- *point-to-point motion*, which implements a fifth-order polynomial trajectory starting form the actual pose, and ending in the target pose received as input. The polynomial minimum jerk law is implemented (null initial and final velocity and acceleration), so as to ensure smoother trajectory profiles and to prevent impulsive jerks.

- *scooping motion*, that designs a half circular motion on the vertical axis, based on the starting and target pose, so as to replicate a scooping movement. This type of motion, designed with constant angular rate of rotation and constant speed, is helpful to collect materials when a scoop-like end-effector is attached to the robot flange.
- *shaking motion*, implements a rapid sinusoidal movement performed in place. The input is given by the shaking direction. This motion is useful to pour completely the material collected with the scoop-like end-effector.

The implementation of these trajectories is not explained in details since they are well-known in robotics literature. Nevertheless, a short introduction can be useful to clarify the tasks explained in the following sections.

#### Finite State Machine

In order to favor the enhancement of autonomous behavior, a FSM is responsible of the inter-connection between the framework individual components. As depicted in Figure 3.10, the FSM receives as input the data coming from the *visual perception module*, i.e. the polygon vertices and peak points described above in the vision system section. By fusing these data with the internal task states, the FSM computes the *interaction expectancy* value that is sent to the *Self-tuning impedance unit*, besides providing the target poses of each task phase to the *trajectory planner*.

Four macro-states compose the FSM flow. The first one, "Workspace definition", delimits the environment where the robot will need to carry out the next phases, thanks to the polygon vertices data coming from the vision module. Next, in the "Exploration" state, the robot inspects all the materials localized in the former phase. This is necessary to identify, for every material, an appropriate value of the stiffness parameter $\boldsymbol{k}_{st}$. To do so, the robot end-effector grasps a stick-like tool and reaches the material located in the leftmost workspace portion. This motion is carried out in a compliant way, since no interaction with the environment is foreseen. Subsequently, the robot dunks the tool into the material and the impedance parameters tuning is activated. In fact, the Boolean values $\boldsymbol{I}_{task}$ and $\boldsymbol{I}_{field}$ turn to True in view of the expected interaction with the material given by the task, and because the tool attached to the end-effector enters the interaction field reconstructed through the polygon vertices. The robot then executes a point-to-point motion towards the center of the polygon. Depending on the viscoelastic properties of the material, the displacement between the desired and the measured pose will vary, thus leading to different adaptations of the impedance parameters, according to Eq. 3.17. The resulting $\boldsymbol{k}_{st}$ is stored in an appropriate data structure that keeps track of the material properties. After pulling out the tool from the first material, this identification process is repeated for every material present in the workspace.

When the tool is dunked into the matter, it would be hard to retrieve its pose by means of the vision system, due to the occlusion of the material itself. Therefore, the value $\boldsymbol{I}_{field}$ would get erroneously disabled. Since the *interaction expectancy* area is fixed in the workspace, and given by the previous step, we can overcome this issue exploiting the robot forward kinematics. In fact, in this way, we can retrieve if the tool grasped by the robot end-effector is within such area or not, regardless of potential visual occlusions.

To respond to potential collisions with the boundary of the box containing the materials, we equipped the framework with a "Fault Detection" sub-unit. This increases the system robustness to unexpected events, for instance due to mismatch in the materials' localization provided by the perception module. A fault is detected when the external forces projected along the motion vector undergo a brusque increase. In this case, the motion is interrupted and the robot returns to its homing configuration. To estimate the trend of the external forces, that determines the fault, we employ a linear regression algorithm. This computation is made every $n$ samples, and if the linear regression slope $m$ goes beyond a threshold set to $m_{fault}$, the fault is triggered.

After the material exploration is completed, the FSM switches to the next state, "Materials distribution". In this step, the FSM retrieves which are the highest point for each material, reading the data computed by the visual perception sub-unit "Peaks localization". These points are needed in the next and final state, "Task", where the robot performs a scooping motion in all the materials to collect a small amount of each substance, so as to pour it in a pot, that will be handed by a second robot. To ensure a successful scooping of the substance, this motion starts from the mentioned peak points. The "Task" state coordinates the order in which the materials need to be scooped, and it is executed as follows.

After having grasped a scooping tool, as a scoop or a small shove, the robot reaches the first scheduled material. This motion is carried out assuming a compliant profile, i.e. by setting the stiffness values to $\boldsymbol{k}_{min}$ along every Cartesian axis. Next, the scooping motion is performed and, since the task requires an interaction with the environment, the value of $\boldsymbol{I}_{task}$ is set to True. As soon as the scooping tool enters the *interaction expectancy* area, also the $\boldsymbol{I}_{field}$ is enabled and so the impedance parameters are subject to adaptation in the direction specified by the motion vector. Thanks to the previous identification of the materials carried out in the "Exploration" state, the initial impedance value $\boldsymbol{k}_{st}$ is set to the one corresponding to the material to be scooped. In such a way, with the impedance appropriately set, the robot does not lag behind and the task can be executed with a high accuracy from the beginning. The scooping motion is carried out towards the polygon center, and the scooped material gets poured in a pot. The robot is then ready to start over the scooping for the other materials, according to the task sequence.

Since the viscoelastic characteristic of the materials can change over time, due to their intrinsic properties or because of external circumstances, even in the "Task" state the value of $\boldsymbol{k}_{st}$ can experience variations. However, to avoid substantial changes, we set a maximum value $\boldsymbol{k}_{st\_max}$ that can be reached by $\boldsymbol{k}_{st}$ as follows:

$$\begin{aligned} &\boldsymbol{k}_{st\_max,m} = \boldsymbol{k}_{st\_exploration,m} * (1+p) \\ &0 \leq p \leq 0.5 \quad \forall m, \end{aligned} \tag{3.25}$$

where $\boldsymbol{k}_{st\_exploration,m}$ is the value that was stored in the "Exploration" state for material $m$, and $p$ represents the percentage of variation that can be set. If the value of $\boldsymbol{k}_{st}$ goes beyond this threshold, the "Fault Detection" sub-unit is triggered, taking the robot to its initial configuration. This feature can help to detect substantial variations in the properties of the materials, that could be a symptom of malfunctions in industrial productions. In addition, with this method, also collisions with the container boundaries would be detected.

**Experimental results**

To show the potential of the presented framework, we conducted experiments using a Franka Emika Panda robotic arm, equipped with the Pisa/IIT SoftHand, that is responsible of carrying out all the phases presented above. Another Panda robot was employed as a support, to provide the main robot a pot where the scooped substances need to be poured. A container with three different materials was placed between the main robot and a support pole, where we mounted the ASUS Xtion Pro RGB-D camera. The materials were accurately selected using two criteria. First, their viscoelastic properties had to present a substantial difference among them, so as to demonstrate distinct impedance adaptation behaviors. Moreover, they also need to be used in large-scale in the agricultural sector. For these reasons, we selected seeds, soil and expanded clay. Figure 3.12 shows the experimental setup.

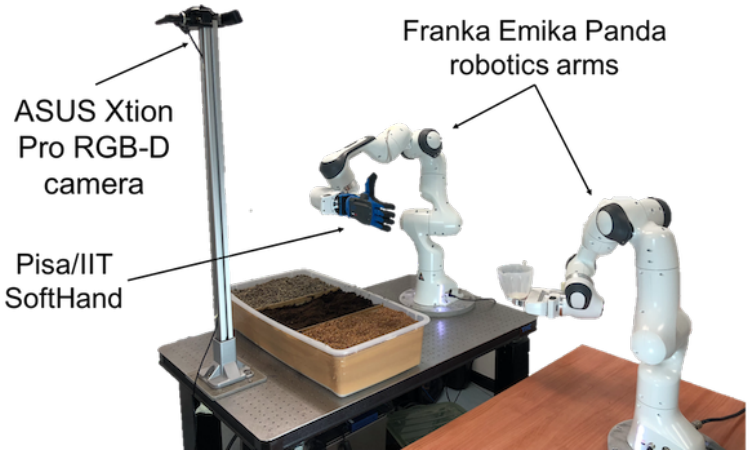


**Figure 3.12:** *The experimental setup includes two Franka Emika Panda robotic arms (one equipped with the Pisa/IIT SoftHand), a container with three materials, and an ASUS Xtion Pro RGB-D camera to sense the scene.*

From a software point of view, the communication among the modules, implemented as ROS nodes, is handled though the publisher/subscriber design pattern, exploiting the ROS infrastructure, as shown in Figure 3.10.

To describe the various phases of the experiment in a temporal order, we follow the task flow provided by the FSM. In the first state, the "Workspace definition" takes place exploiting the data received by the "Materials localization" perception sub-unit. Since the materials were distributed in a container with a rectangular shape, and divided in three rectangular sub-portions, the visual unit detects 12 Euclidean points that represent the *Polygon vertices* of the three areas. The relative column of Table 3.1 shows the data retrieved through the point cloud data.

At this stage, the robot is ready to switch to the "Exploration" state, grasping a $27cm$ long metal stick to analyze the materials exploiting Eq. 3.17. To this end, prior to the experiment, we had to set an appropriate value to the update parameter $\boldsymbol{\alpha}$. This choice had to represent a trade-off between fast convergence and stiffening performance. To select a value suitable for various materials, we performed this analysis on multiple substances such as soil, sand, rocks with different density, air and water, estimating, for every material $m$, a $\boldsymbol{\alpha}_m$, and then defining the general $\boldsymbol{\alpha}$ value through the arithmetic

| Material | Polygon vertices (x,y,z) | | | | Peak point (x,y,z) | $k_{st}[N/m]$ |
|---|---|---|---|---|---|---|
| | V1 | V2 | V3 | V4 | | |
| 1 | (0.66, 0.30, 0.10) | (0.36 0.30, 0.11) | (0.36 0.09, 0.09) | (0.66 0.08, 0.09) | (0.63, 0.22, 0.14) | 1100 |
| 2 | (0.66, 0.09, 0.08) | (0.36, 0.09, 0.09) | (0.36, -0.16, 0.10) | (0.66 -0.16, 0.09) | (0.62, -0.01, 0.12) | 1650 |
| 3 | (0.66, -0.14, 0.10) | (0.36, -0.14, 0.10) | (0.36, -0.36, 0.13) | (0.66, -0.36, 0.11) | (0.59, -0.29, 0.13) | 1330 |

**Table 3.1:** *The data structure containing the data coming from the Self-tuning impedance unit, relative to the stiffness $\boldsymbol{k}_{st}$ (rounded to integers), and from the Visual perception module, relative to the polygon vertices of the materials and their peak points.*

| $\boldsymbol{m}$ | $\boldsymbol{\alpha}_m$ |
|---|---|
| air | 16200 |
| water | 16800 |
| seeds | 19400 |
| soil | 22300 |
| expanded clay | 21800 |
| soil + water | 23500 |

**Table 3.2:** *To derive the value of $\boldsymbol{\alpha}$, different materials have been taken into account. The resulting average, computed through Eq. 3.26, was set to 20000.*

mean of all the $n$ materials taken into account in the analysis:

$$\boldsymbol{\alpha} = \frac{1}{n} \sum_{m=1}^{n} \boldsymbol{\alpha}_m. \tag{3.26}$$

Based on Eq. 3.26 and on the values reported in Table 3.2, $\boldsymbol{\alpha}$ was set to $20000$. To ensure a good level of compliance in case of unpredicted collisions, the value of $\boldsymbol{k}_{min}$ was set to $500N/m$. During the "Exploration" state the robot gets to the leftmost material, formed by seeds (material 1), dunking the metal stick into it. An interaction with the environment is expected to happen, and therefore the $\boldsymbol{I}_e$ value is activated, as shown at $t = 1.5s$ in the fourth plot of Figure 3.13. Consequently, the *Self-tuning impedance unit* is enabled.

While keeping the tool immersed in the material, the robot performs an $18cm$ long movement along the *x-axis*. Since $\boldsymbol{\Delta P}$ goes beyond the threshold $\boldsymbol{\Delta P}_t$ set to $1cm$, $\boldsymbol{k}_{st}$ increases following Eq. 3.17 as shown in the first plot of Figure 3.13. As a consequence, the Cartesian stiffness along the direction of interaction is adapted. In this case, the movement direction is performed only on the *x-axis*. With the increase of the impedance values, we can notice that $\boldsymbol{\Delta P}$, that represents the Cartesian error along the motion vector, gets reduced and goes below the threshold $\boldsymbol{\Delta P}_t$. The maximum value reached by $\boldsymbol{k}_{st}$ in this case is equal to $1100N/m$, and it gets associated to the relative material as reported in the right column of Table 3.1. To complete the "Exploration" state, the robot repeats the same described procedure for the other two materials. As expected, the soil (material 2) turns out to be the stiffest material, with $\boldsymbol{k}_{st}$ reaching a

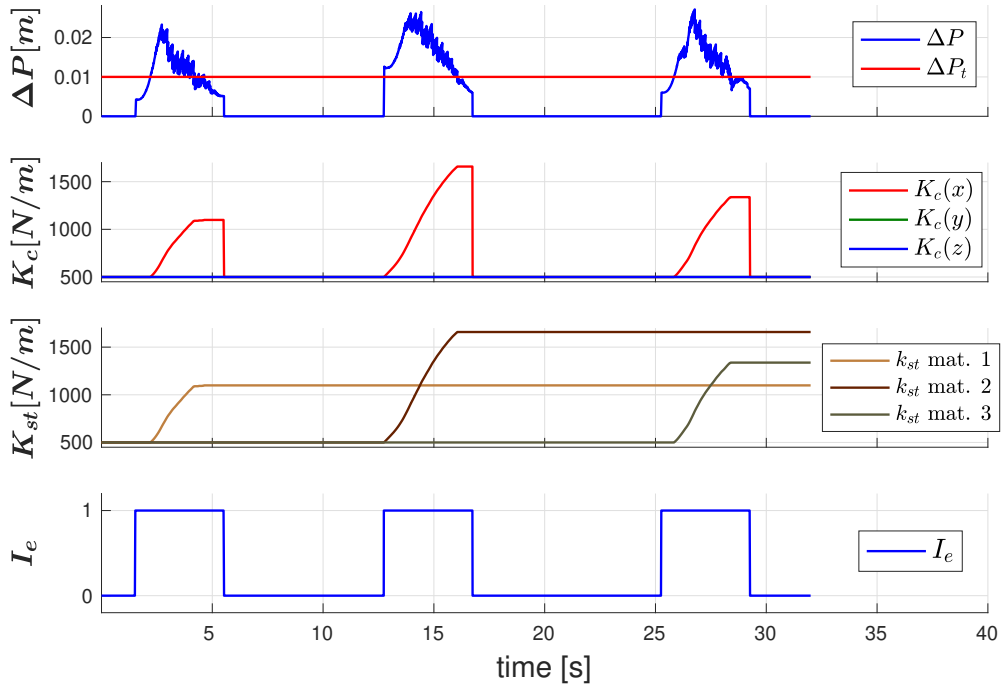


**Figure 3.13:** *"Exploration" state: when an interaction with the environment is predicted, the self-tuning stiffness $\boldsymbol{k}_{st}$ value grows based on the error in the direction of the motion ($\boldsymbol{\Delta P}$).*

value of $1650N/m$, and the expanded clay (material 3) is in between the other two, i.e. $1330N/m$. In the third plot of Figure 3.13, we can notice how these values are tuned.

After the identification of the impedance parameters, the FSM transits to the "Material distribution" state. The starting point of the scooping trajectories are detected by the "Peaks localization" unit of the perception module, and stored in the relative column of Table 3.1.

Then, the robotic hand grasps a scooping tool in order to carry out the "Task" state, subdivided in four substates. The robot scoops and pours in a plant pot, provided by the second robotic arm, the three materials following this sequence: soil (a), plant seeds (b), other soil (c), and expanded clay (d). These four substates are depicted in the plots of Figure 3.14 and Figure 3.15. In the latter, the green triangles represent the highest point of each substance provided by the "Peaks localization" perception module. To foster a deeper understanding, the axes of this figure are oriented to analyze the task from a lateral view. In this way, it is clear to see how the stiffness value $\boldsymbol{k}_{st}$ is adapted along the direction of the motion $\widehat{\boldsymbol{P}}$ inside the interaction expectancy area. Faint and shorter arrows symbolize lower stiffness values, while longer and more vivid arrows represent higher stiffness values. The direction of the motion vector in the Cartesian space is also specified in the plot related to the three components of the normalized motion vector $\widehat{\boldsymbol{P}}$ in Figure 3.14.

When no interaction is to happen, i.e. outside the containers, the robot keeps a compliant profile, and $\boldsymbol{k}_{st}$ is always set to $\boldsymbol{k}_{min}$, i.e. $500N/m$. Entering the interaction expectancy area leads to a rapid adaptation of $\boldsymbol{k}_{st}$, that assumes the value stored in Table 3.1 relative to each material. This can be noticed by the sudden growth in the arrows length and color intensity. In case $\boldsymbol{\Delta P}$ keeps its value below the threshold, it

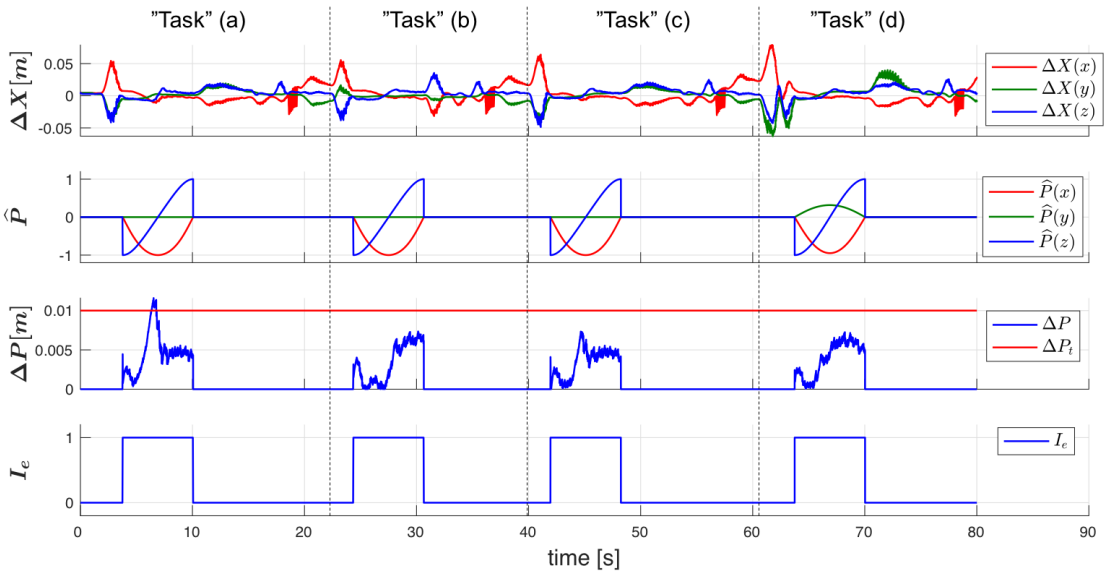


**Figure 3.14:** *"Task" state: starting from the impedance values already tuned in the "Exploration" state, the framework allows the robot not to lag behind, so that $\boldsymbol{\Delta P}$ remains below the $\boldsymbol{\Delta P}_t$ threshold. Only in "Task" (a), $\boldsymbol{\Delta P}$ goes beyond the threshold, since the material viscoelastic properties have been intentionally changed.*

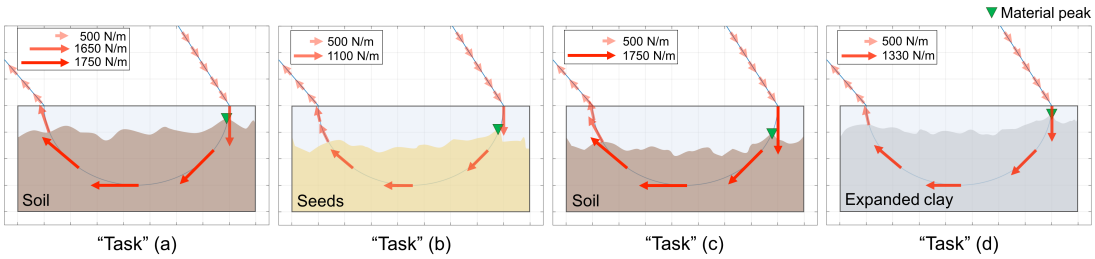


**Figure 3.15:** *A lateral view of the four "Task" substates. $\boldsymbol{k}_{st}$ is projected onto the direction of motion and here represented by means of red arrows, whose intensity grows with their value.*

means that the viscoelastic properties of the material did not change, and so there is no need for further adaptation. When the scooping is over, but still being inside the container, $\boldsymbol{k}_{st}$ gets reduced according to Eq. 3.20, as can be seen the last part of the scooping. Notice that, negligible variations could lead to unnecessary changes, so we designed a moving average window to calculate $\boldsymbol{\Delta F}_{ext}$. In the last part of the depicted motion, the robot exits the interaction expectancy area, and $\boldsymbol{k}_{st}$ is restored to its default compliant value, i.e. $\boldsymbol{k}_{min}$.

To show that the impedance self-tuning would occur also in case of viscoelastic properties changes, we decided to pour some water in the soil between the "Exploration" and the "Task" states. This adaptation is visible when the scooping tool enters the soil during "Task"(a), and it is caused by $\boldsymbol{\Delta P}$ exceeding the threshold $\boldsymbol{\Delta P}_t$ as shown in the third plot of Figure 3.14 at $t = 6.3s$ when the *Self-tuning impedance unit* is activated again. The value of $\boldsymbol{k}_{st}$ for material 2 gets increased from $1650N/m$ to $1750N/m$. This is highlighted by the difference between the first and the other arrows inside the leftmost container in Figure 3.15.

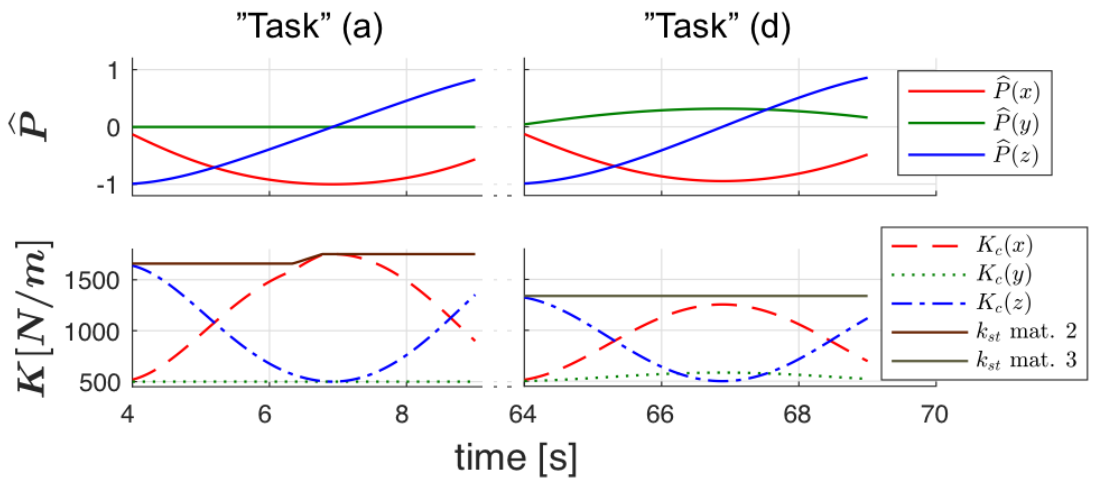


**Figure 3.16:** *"Task" state: the tuning of the Cartesian stiffness is achieved only in the direction of movement $\widehat{\boldsymbol{P}}$. The vectorial sum of the three diagonal components is always equal to $\boldsymbol{k}_{st}$.*

In Figure 3.16, we show how the tuning of the Cartesian stiffness is achieved only in the direction of movement $\widehat{\boldsymbol{P}}$, when the tool is inside the materials in two of the "Task" state sub-phases. In "Task"(a), $\boldsymbol{k}_{st}$ of material 2 is adapted at $t = 6.5s$, since the soil viscoelastic properties were changed as explained above. We can notice that the sum of the three Cartesian stiffness diagonal components is always equal to $\boldsymbol{k}_{st}$. In "Task"(d), we see the adaptation also on $\boldsymbol{K}_c(y)$.

To show that the framework reliability is ensured by means of the two "Fault Detection" sub-units, we repeated the experiment simulating a fault in the perception unit by changing the pose of the box containing the materials both during the "Exploration" state and the "Task" state execution. Like this, following the desired trajectory, the tool grasped by the robot end-effector collides with one of the container sides.

Figure 3.17 shows an execution of the "Exploration" state performed to retrieve the first material $\boldsymbol{k}_{st}$. As can be noticed in the third subplot, performing a linear regression (red solid curve) on the measured external forces data (blue scattered curve) allows to define in the fourth subplot the linear regression slope $\boldsymbol{m}$ (blue curve) that, when it goes beyond the threshold $\boldsymbol{m}_{fault}$ set to $-15$, triggers a fault at $t = 4.5s$. The robot stops performing the task and goes back to its homing position in a compliant way, as illustrated in the second subplot that shows the Cartesian stiffness parameters. The linear regression was performed every $500$ samples, i.e. every $0.5s$.

On the other hand, Figure 3.18 shows the behavior of the "Fault Detection" sub-unit associated to the "Task" state. Since in the "Exploration" state (performed without faults), $\boldsymbol{k}_{st}$ for material 1 reached the value $1100N/m$, by applying Eq. 3.25 with $p = 0.3$, we obtain that $\boldsymbol{k}_{st\_max,1} = 1433N/m$. As shown in the plots, at the time of the collision, i.e. $t = 6s$, $\boldsymbol{\Delta P}$ increases suddenly and so $\boldsymbol{k}_{st,1}$ goes beyond $\boldsymbol{k}_{st\_max,1}$ computed above. The execution halts, the robot exits the material and goes back to its initial configuration.

To further demonstrate the validity and the effectiveness of the proposed method, we decided to perform another experiment inserting an obstacle inside one of the materials, so as to simulate an uncertain environment, and carrying out one more time the FSM

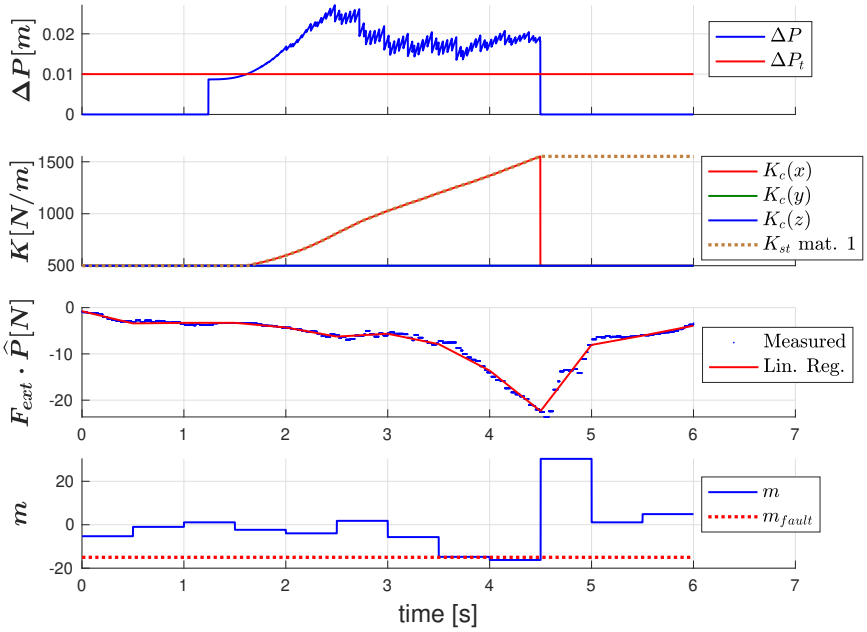


**Figure 3.17:** *"Fault Detection" during "Exploration" state: if the external forces linear regression slope $\boldsymbol{m}$ goes beyond the threshold $\boldsymbol{m}_{fault}$, the robot stops the current motion and goes back to its homing configuration in a compliant way.*

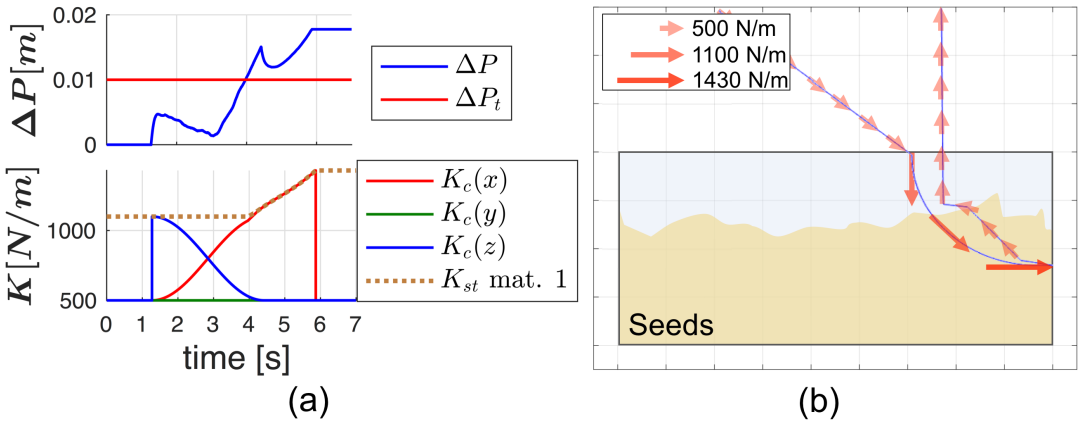


**Figure 3.18:** *"Fault Detection" during "Task" state: if $\boldsymbol{k}_{st}$ goes beyond the $\boldsymbol{k}_{st\_max}$ threshold, the execution halts and the robot goes back to its initial configuration in a compliant way.*

"Task" state. We put a piece of wood inside the seeds (material 1), placing it in the middle of the path of the expected reference trajectory. In this way, the tool held by the robot had to react adapting to the wood shape. We repeated the experiment three times. At first, we removed the *Self-tuning impedance unit* from the framework, so that the impedance parameters were not subjected to changes, even if an interaction was expected. With this configuration, we performed the experiment with high and low impedance parameters and we compared the obtained results. Afterwards, we carried out the task with the same setup, but with the impedance regulation enabled. To evaluate

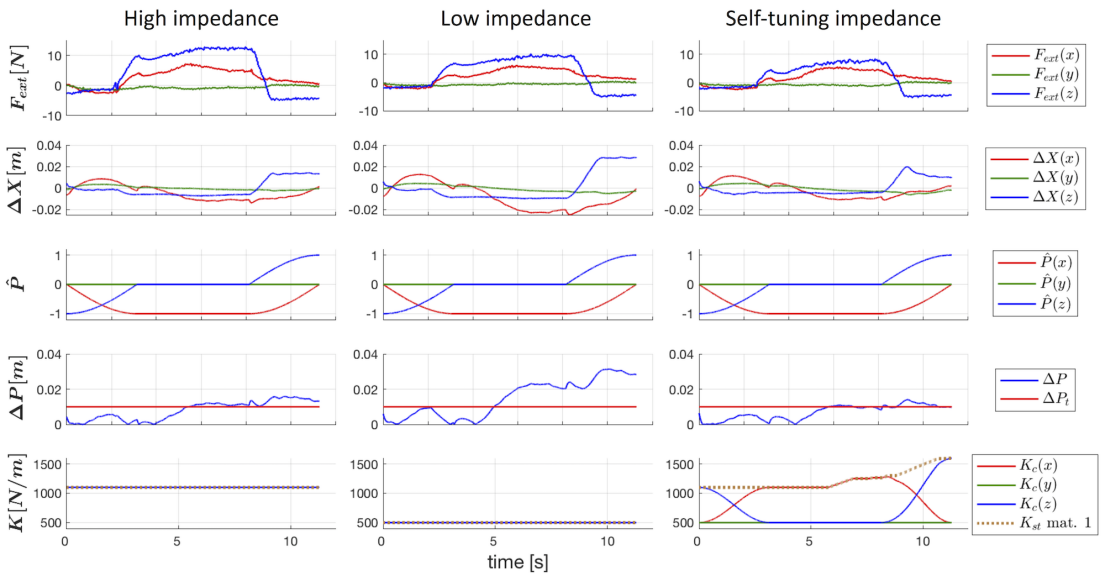


**Figure 3.19:** *The plots show three different executions of the "Task" state with an object placed inside material 1 (setup shown in Figure 3.20): on the left without applying the presented method and always keeping high impedance values, in the center as the previous case but with low impedance parameters, and on the right the trial with the Self-tuning impedance unit enabled. In the latter case, there are less external interaction forces with respect to the first case, and the error in the direction of the movement $\boldsymbol{\Delta P}$ is substantially less with respect to the second trial.*

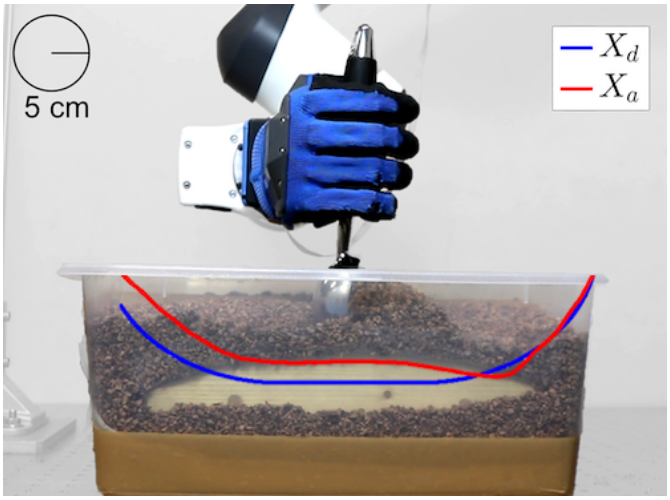


**Figure 3.20:** *To simulate an uncertain environment, a piece of wood has been inserted inside material 1. The curves in the picture represent the desired trajectory (blue line) and the actual trajectory (red line).*

the three described trials, whose plots are illustrated in Figure 3.19, we decided to compare the external interaction forces acting on the robot end-effector, i.e. $\boldsymbol{F}_{ext}$, and the Cartesian error projected onto the direction of the movement, i.e. $\boldsymbol{\Delta P}$, under the different conditions of the impedance parameters.

The first column represents the data acquired while keeping always a high level of the impedance parameters, i.e. $1100N/m$ (see bottom plot), that is the value reached by

$\boldsymbol{k}_{st}$ for material 1 during the "Exploration" state. In this case, although the tracking of the error $\boldsymbol{\Delta P}$ does not exceed excessively the imposed threshold $\boldsymbol{\Delta P}_t$, the interaction forces on the *z-axis*, represented by the blue line on the top plot, reached a quite high value ($\approx 13N$). Therefore, this approach could lead to a system failure caused by a tool/robot damage. Notice that, with higher obstacle curvatures, the external forces measurements could scale quite rapidly easily leading to more substantial failures.

The second column shows the plots of the trial with lower impedance parameters, set at $500N/m$, as no interaction was ever expected. The robot is able to better comply with the external environment, as highlighted by lower interaction forces on the *z-axis*, that reach a maximum value of $\approx 10N$, and therefore damages are more likely prevented. Nevertheless, complying both with the expected and unexpected interactions with the environment leads also to a loss in terms of performances. This can be seen in the plots representing $\boldsymbol{\Delta P}$, where the robot lags behind the desired trajectory up to $3cm$. This behavior can not be considered desirable, since the task is not carried out as expected.

Lastly, the third column depicts the data logged applying the method presented in this work. Stiffness and damping are updated on-line, based on the interaction expectancy and on the direction of the movement $\widehat{\boldsymbol{P}}$. The external interaction forces are further reduced, reaching at maximum $\approx 8N$. $\boldsymbol{\Delta P}$ is significantly less with respect to to the case with low impedance.

In Figure 3.20, we report the setup used in this experiment enhanced with the reference trajectory (blue curve) and the measured path (red curve) logged during the last trial, when the *Self-tuning impedance unit* was enabled. A video of the experiment is available at [19].

#### Stability analysis

Since the presented controller is based on continuous variations of the impedance parameters, we must demonstrate that the passivity of the system, and so its stability, is guaranteed. Following the approach presented in [58], we implemented a tank-based approach to monitor the system stability. Formally, the model of the robot in the task space is given by:

$$\begin{aligned} \boldsymbol{\Lambda}(\boldsymbol{x})\boldsymbol{\Delta}\ddot{\boldsymbol{X}} + (\boldsymbol{\mu}(\boldsymbol{x},\dot{\boldsymbol{x}}) + \boldsymbol{D}_d)\boldsymbol{\Delta}\dot{\boldsymbol{X}} + \\ \boldsymbol{K}_{const}\boldsymbol{\Delta X} - \boldsymbol{\omega} x_t = \boldsymbol{F}_{ext}, \\ \dot{x}_t = \frac{\sigma}{x_t}(\boldsymbol{\Delta}\dot{\boldsymbol{X}}^\top \boldsymbol{D}_d \boldsymbol{\Delta}\dot{\boldsymbol{X}}) - \boldsymbol{\omega}^\top \boldsymbol{\Delta}\dot{\boldsymbol{X}}, \end{aligned} \tag{3.27}$$

where the desired stiffness is equal to $\boldsymbol{K}_d(t) = \boldsymbol{K}_{const} + \boldsymbol{K}'(t)$, being $\boldsymbol{K}_{const} \in \mathbb{R}^{6\times6}$ the constant stiffness term and $\boldsymbol{K}' \in \mathbb{R}^{6\times6}$ the time-varying stiffness. $\boldsymbol{\Lambda}(\boldsymbol{x}) \in \mathbb{R}^{6\times6}$ and $\boldsymbol{\mu}(\boldsymbol{x},\dot{\boldsymbol{x}}) \in \mathbb{R}^{6\times6}$ are the Cartesian inertia and Coriolis/centrifugal matrices respectively. The scalar $x_t \in \mathbb{R}$ is the state associated with the tank and the tank energy $T \in \mathbb{R}^+$, with $T = \frac{1}{2}x_t^2$. $\sigma \in \mathbb{R}$ and $\boldsymbol{\omega} \in \mathbb{R}^6$ respectively are

$$\sigma = \begin{cases} 1 & if\, T(x_t) \leq \bar{T}_u \\ 0 & otherwise \end{cases}, \tag{3.28}$$

$$\boldsymbol{\omega} = \begin{cases} -\frac{\boldsymbol{K}'(t)\tilde{\boldsymbol{x}}}{x_t} & if\, T(x_t) \geq \bar{T}_l \\ \mathbf{0} & otherwise \end{cases}, \tag{3.29}$$

where $\bar{T}_u \in \mathbb{R}^+$ is a suitable, application dependent, upper bound that avoids excessive energy storing in the tank, while $\bar{T}_l \in \mathbb{R}^+$ is a lower bound below which energy cannot be extracted by the tank for avoiding singularities in Eq. 3.27, and thus the time-varying stiffness $\boldsymbol{K}'(t)$ will be removed. For a detailed analysis of the system passivity, please refer to [58].

Figure 3.21 shows the stability analysis performed during the entire duration of the experiment, i.e. including the "Exploration" and the "Task" states, when no faults occurred. As we can see from the bottom subplot, the tank energy was above the lower bound ($\bar{T}_l = 0.5J$) during all the phases, which means that the full stiffness including the constant (set to the compliant value, $500N$) and varying parts can be realized without loss of stability.

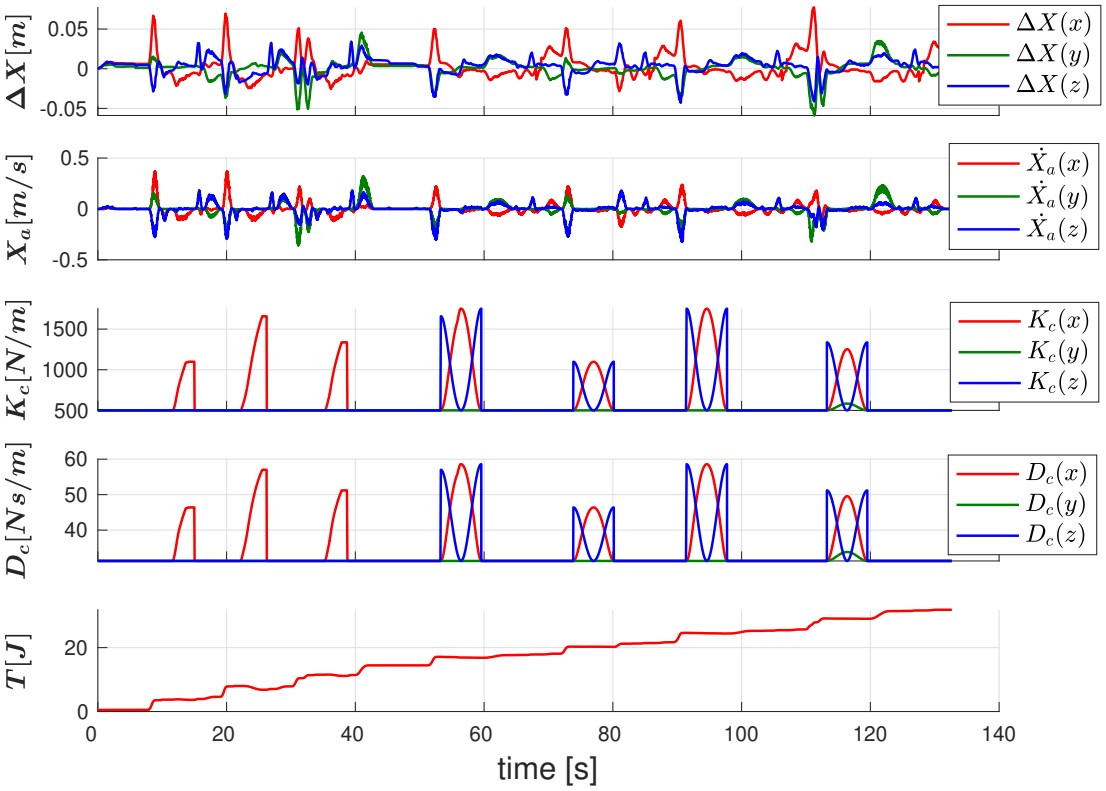


**Figure 3.21:** *The analysis reveals that the tank energy is above the lower bound $\bar{T}_l$ set to $0.5J$ for the entire experiment, thus guaranteeing there is no loss of stability.*

### 3.3.3 Item sorting and box-filling

In this section, we aim to provide a novel adaptive solution in the direction of flexible warehouses automation. In fact, recent collaborative robotic technologies have the potential to add high levels of flexibility to the manufacturing processes, due to their versatility and human-centric design [6]. They can not only contribute to the creation of hybrid (human-plus-robot) and resource-efficient manufacturing solutions, but also can help to reduce human physical stress and automate repetitive and cognitively unexciting industrial tasks [106]. Despite this, cobots of today are mostly exploited in structured manufacturing environments, where a precise knowledge of the surroundings is required for their operation. In such a way, the true potential of cobots cannot be demonstrated, which is fast adaptation to the variabilities arising from the environment and human co-workers.

Several research works have aimed to improve the manipulation flexibility of cobots in performing repetitive tasks such as pick-and-place [57, 86], sorting [56], and boxing [200], through adaptive control systems [39, 53, 196] that exploit sensory information such as vision [139], and force [113]. These operations are particularly important from a flexible automation perspective, since they are of a repetitive nature, and involve large body movements that can lead to musculoskeletal disorders in long term [73, 120].

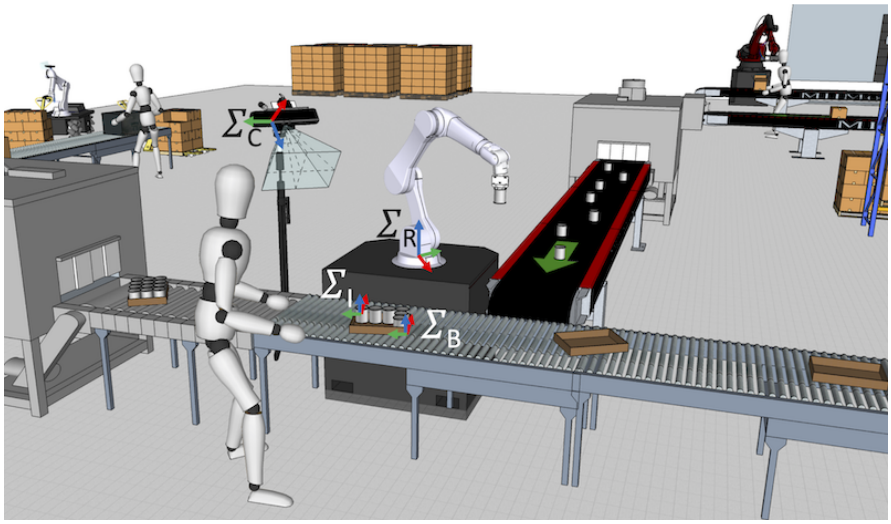


**Figure 3.22:** *Concept illustration of flexible and collaborative robotic box-filling and item-sorting in manufacturing industry. The depicted reference frames represent the robot base (R), the camera (C), the box (B), and the item (I).*

To respond to the flexibility needs of common industrial tasks such as pick-and-place and item-sorting, in this section, we propose a new adaptive framework for collaborative robots. The aim is to provide the system with the ability to cope with unexpected environmental and operational changes (e.g., human interventions and box position/orientation changes). To this end, we first decided to perform human experiments to understand how humans perform typical pick-and-place and item-sorting tasks subject to perception uncertainties. Starting from this analysis, we defined a novel human-inspired *items placing strategy* that has been embedded in the presented framework. The results reported in this section led to the scientific publication [21].

#### Observations of human behavior

To implement an efficient *items placing strategy* for the robotic box-filling, we decided to get inspiration from human behavior, asking 15 human subjects to fill in some items in a box in the most natural way possible. The objective was to understand the underlying perception-action principles for this particular task, and to possibly replicate them in our robot control framework.

In the depicted scenario, the robotic end-effector can be regarded as the human hand, and the camera perception system as the human eyesight. However, we know that steerable human sight is much more accurate with respect to the artificial perception sensing of a fixed camera, especially in unstructured and dynamically changing environments. Hence, we decided to carry out the experiment twice for each subject: once with the eyes uncovered to simulate a perfect visual perception system, and once with the subjects being blindfolded to reproduce a scenario with complete lack of visual sensing.

The subjects were asked to place one item in the lower-right edge of a box (see "goal" reference frame $\mathbf{\Sigma}_G$ in Figure 3.23a), while their hand pose ($\mathbf{\Sigma}_H$) was tracked with a *Intel RealSense Depth* camera through the OpenPose library [34]. Conventionally, in industrial pick-and-place machine tools and robotic systems, the items are directly placed in the desired pose, relying on accurate sensing of the environment and precise robot trajectory planning. Therefore, the main objective of this analysis was to infer how the items were placed in the goal position that was instructed to the subjects (the box edge).

To retrieve the human items placing strategy, we decided to analyze the trajectory executed from the moment the item was in contact with the box layer till it was released in its goal pose. Figure 3.23a shows the three different poses of the human arm before reaching the contact with the box (P1), during the box layer contact (P2), and in the ending pose where the item had to be placed (P3). Figure 3.23b depicts the human pose with respect to the Box frame $\Sigma_B$. From here it is possible to notice that, after having reached pose P2, the pose of the human hand on the *z-axis* (blue curve) is always null, so we can deduce that, between P2 and P3, there is only a sliding motion on the horizontal plane *xy*.

Based on the former considerations, we can illustrate the trajectory computed by the human hand in 2D since the contact with the box layer, i.e. from P2 to P3. Figure 3.23c shows the same trajectory tracked in Figure 3.23a-b on the box plane, while Figure 3.23d shows the Euclidean distance between the hand pose and the goal pose, $\boldsymbol{d}_{\overline{HG}}$, over time. It is worth to notice that the traditional positioning of the item directly on the goal pose (classical *Peg-in-hole* strategy) would make P2 coincide with P3, since the goal pose would be reached simultaneously to the contact with the box plane. With such a strategy, in Figure 3.23d, the curve would be degenerated in a single point at the origin of the axes, since a null distance $\boldsymbol{d}_{\overline{HG}}$ would be achieved instantaneously at $time = 0s$.

Fifteen healthy subjects (twelve males and three females; age, $28.5 \pm 3.9$ years) participated in the overall experiments. Figure 3.24 shows the results of the 15 trials for both the uncovered eyes case and the blindfolded one. Similarly as in Figure 3.23c, we illustrate the motion primitives executed during all the subject trials in the box plane (top), and the mean $\sigma$ and standard deviation $\mu$ of $\boldsymbol{d}_{\overline{HG}}$ (bottom).

As can be seen from the results reported in Figure 3.24, no subject placed the item

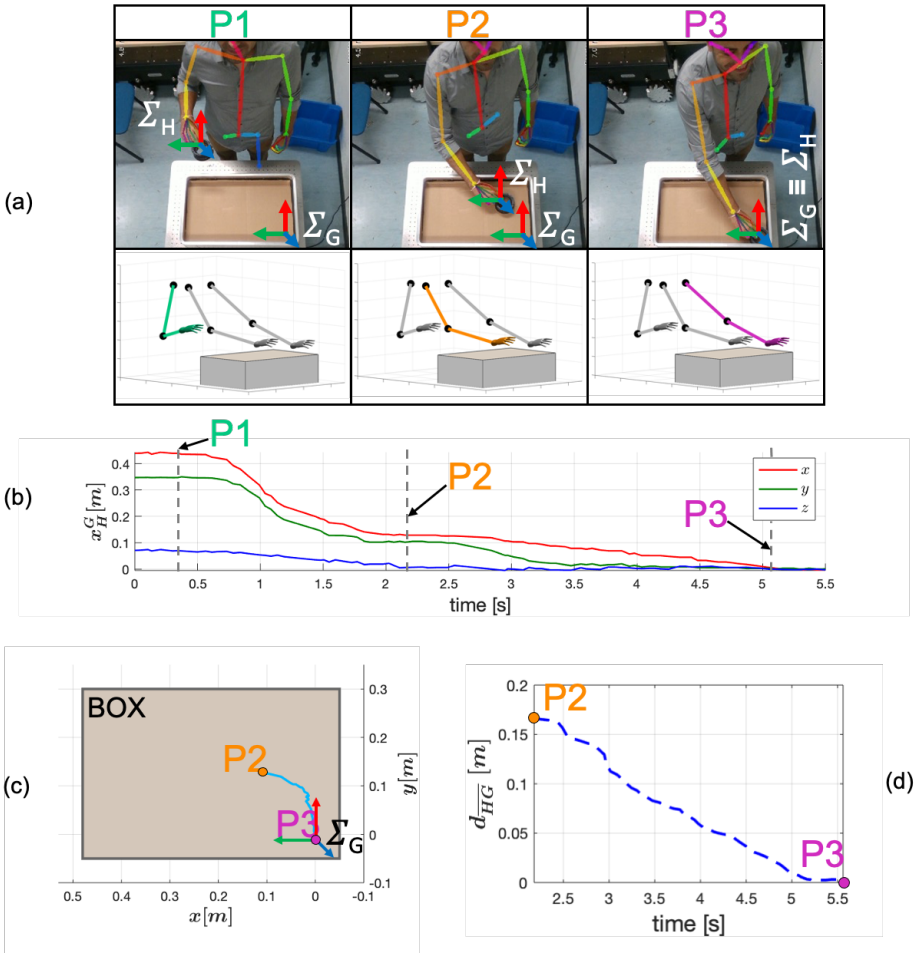


**Figure 3.23:** *Three different poses of the human arm have been illustrated in this figure: before reaching a contact with the box (P1), during the contact (P2), and the pose where the item had to be placed (P3). Subfigure (a) depicts the experimental snapshots of the three poses and their graphical representations, while (b) shows the human hand pose $\boldsymbol{X}_H^G$ with respect to the goal reference frame $\boldsymbol{\Sigma}_G$: when $\boldsymbol{X}_H^G(z)$ tends to a null value we can consider the start of the sliding phase on the plane (P2). On the bottom, the plot of the sliding phase (P2 to P3) is represented in the xy-plane (c), as well as the distance between the hand and the goal $\boldsymbol{d}_{\overline{HG}}$ over time (d).*

directly from the top to the desired pose (*Peg-in-hole* strategy). Instead, although in the first scenario the subjects were coadiuvated by a perfect visual sensing (Uncovered eyes), the strategy employed was to first place the item in an empty area of the box, with an initial distance $\boldsymbol{d}_{\overline{HG}} = 0.05 \pm 0.08m$, and then slide it towards the goal pose. We assume that the reason of this choice stands in the intrinsic confidence of possessing a higher accuracy in sensing the external forces with respect to our visual perception. This becomes even more evident in the blindfolded scenario, where the awareness of the environment is very poor because of the lack of eyesight feedback, when $\boldsymbol{d}_{\overline{HG}}$ at time $t = 0s$ reaches values three times as high as the first case ($0.16 \pm 0.08m$).

The results of this analysis show that the subjects barely chose to place items directly from the top to the desired pose (*Peg-in-hole* strategy), while instead they placed objects starting from a "safer" initial guess and then adjusted their pose with the help of force

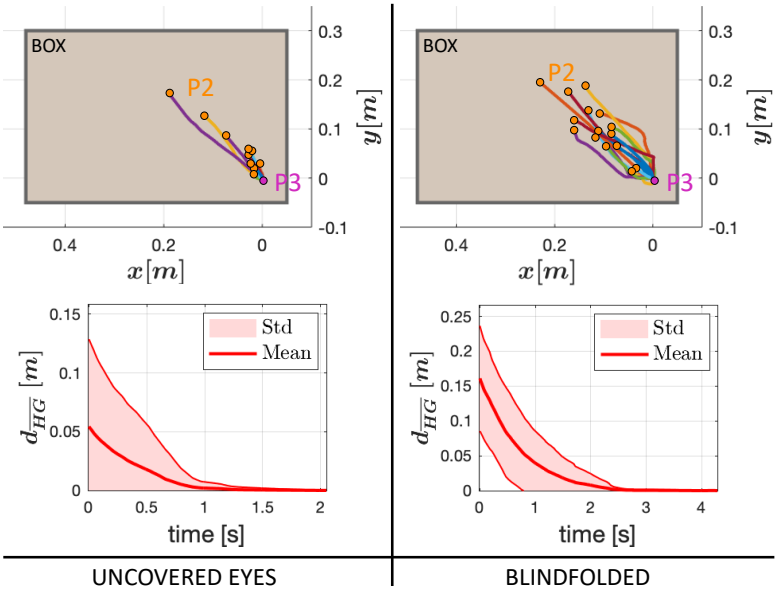


**Figure 3.24:** *Fifteen subjects performed the item placing experiment: once with uncovered eyes, and once blindfolded. With poorer visual information, the employed sliding strategy is even more evident.*

feedback. This distance, in our opinion, was decided based on the level of perception uncertainty and the required accuracy. In fact, studies in neuroscience found out that the best performances in reaching tasks are achieved when humans can control both sensory and motor accuracy at the same time, and therefore overcoming visuomotor-only uncertainties [23]. Furthermore, studies on Fitts tapping task [62], simulated through virtual reality, have demonstrated that eye-hand coordination achieves higher performances when enhanced with force feedback [13]. Other results also revealed than blind people possess enhanced tactile acuity, that they developed to overcome the lack of visual feedback [68]. This can also suggest why in our experiment, when blindfolded, subjects relied much more on the force sensing.

Based on these results, we decided to implement an *Items placing strategy* for our robotic system, that will be presented below. In summary, the algorithm will replicate human behavior in placement of the items in their desired locations (decided by an occupancy grid map and a planner), where a safe initial distance will be chosen experimentally based on the the level of uncertainty present in the perception-action system.

#### Robot framework

The presented framework aims to improve adaptability in the typical industrial task of pick and place. The concept of flexibility is at the core of this methodology, since the environment can be continuously subject to changes induced by the external agents, that can be identified by humans, other robots, or a combination of both.

The framework's software architecture has been built with a high modularity, to facilitate its understanding, and to encourage the reusability of its components. To this end, three new modules have been built, as shown in Figure 3.25:

- a perception module able to detect the location of the items to be picked from the

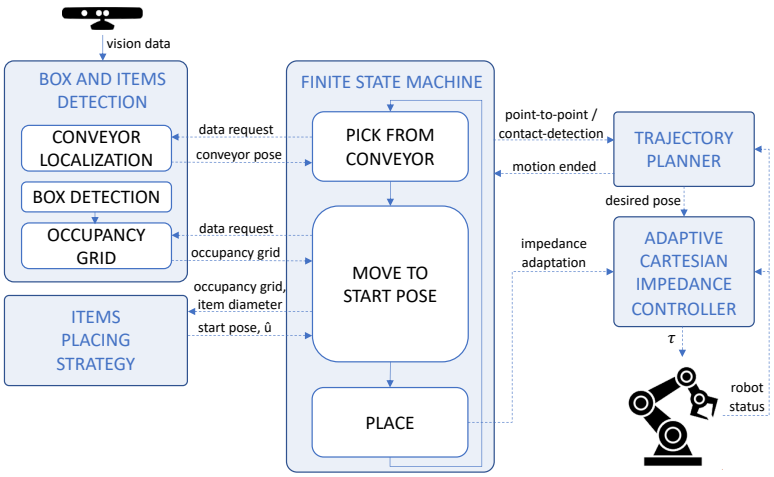


**Figure 3.25:** *The software architecture of the framework composed of 5 modules implemented as ROS nodes. The dashed line represent the data exchanged among the units.*

conveyor, the pose of the box where they need to be placed, and the position of the items already laying inside the box;

- an items placing strategy to place the items inside the box based on the vision information provided by the previous module;
- a Finite State Machine that coordinates the data exchanged across the other modules.

Furthermore, the framework includes the self-tuning impedance controller presented in Section 3.2.2 and a trajectory planner to define the robot desired motions. The latter offers two different types of trajectories. The first one is a classical *point-to-point* motion, computed by means of a fifth-order polynomial, that given a starting and a target pose, computes the intermediate waypoints. The second one is given by a *contact-detection* motion, that when receives as input a unit vector and a threshold force, starts moving the robot end-effector from the current pose towards the unit vector direction until the forces along that direction go beyond a given threshold. This unit, which continuously reads the robot status, also regulates the switching between the FSM states by triggering motion-ending acknowledgments.

Herafter, we illustrate the three novel components that are listed above: Box and items detection, Items placing strategy, and Finite State Machine.

#### Box and items detection

During close-proximity motions, each agent (e.g. humans, robots, automated machines, etc.) demands a considerable level of flexibility and adaptability. For this reason, a reliable and accurate perception system is crucial to perform successful and harmless tasks. The vision components implemented and integrated on the framework are presented, and they are in charge of the conveyor localization, detection of the container and identification of the occupancy of the items laying inside it.

*Conveyor Localization:* the first phase of every pick and place task consists in picking the items from a certain location, in our case represented by a moving conveyor. To

augment the framework flexibility, we decided to detect the picking location at every iteration, localizing the conveyor through vision. Since the grasping phase should be performed with a fair precision, we used *ArUco* markers to locate the picking pose of the items on the conveyor. The strength of this approach is the reliability and speed in the marker pose computation, as presented in [67] and [158]. The *ASUS Xtion PRO Live* camera, mounted on top on the table, provides RGB images to the *aruco_detect* ROS package[3], which estimate the pose of the marker. Therefore, a fixed transformation is applied to grasp the object properly.

*Box Detection:* the second perception requirement consists in the detection and the pose estimation of the box to be filled in. Since ideally the container will be transported throughout the production process and then shipped to other logistic departments, we decided not to use the *ArUco* markers concept applied above. As a matter of fact, every rectangular box is characterized by four corners whose location lies on the same plane. Therefore, we implemented a detection algorithm exploiting the geometric shape of the box. The system processes the point cloud, acquired by the depth sensor integrated in the *ASUS* camera, by applying a pass-through filter. The aim is to remove the points which belong outside the region of interest, defined as $d \leq d_{threshold}$, where $d_{threshold}$ is the parameter which defines the distance between the camera and the table. Through a segmentation procedure, the pre-processed point cloud is clustered in smaller sets of points. Furthermore, the algorithm looks for a set of points that respects specific geometrical features. Thus, by calculating the center of the object and analyzing the principal components, we obtained the reference frame of the object and, accordingly to this, we estimated four candidate corners of the box, based on the dimensions of the elaborated point cloud. A positive detection is acquired when at least a point is in proximity of each candidate corner.

*Occupancy grid:* in order to obtain the desired item placement position, the framework evaluates a sorting strategy based on an occupancy grid of the detected box content. The high mutability of the scenario, introduced by the human agent which cooperate with the cobot to accomplish the task, requires a dynamic update of the candidate placement positions. The framework should be capable to neglect non-static objects captured during the detection phase (human upper limb). Therefore, it is necessary to obtain different camera depth frames and merge the information to obtain a single occupancy grid.

To this end, the algorithm processes the point clouds using a pass-through filter, which exploits the pose and the dimensions resulting from the box detection. The bottom of the box is removed from the resulting point clouds, in order to consider just the border of the box and its content. Then, since the occupancy grid is a discretized 2D representation of the space, a projection of the point cloud on the *xy-plane* is required to distinguish the occupied cells from the free ones. In [180], the authors proposed an approach to iteratively update the cell occupancy state. The following formula is based on binary Bayes filter in *log odds* form with an inverse measurement model:

$$l_t^i = l_{t-1} + \log \frac{p(x_i|z_t)}{1 - p(x_i|z_t)} - \log \frac{p(x_i)}{1 - p(x_i)}, \tag{3.30}$$

where $l_t^i$ is the *log odds* at time $t$ of the $i$-th cell, $x_i$ is the occupancy state of the cell,

[3] `http://wiki.ros.org/aruco_detect`

$p(x_i)$ is the probability that the $i$-th cell is occupied, while $p(x_i|z_t)$ is the probability that the $i$-th cell is occupied given the depth sensor data $z$ at time $t$. Since, the $i$-th cell have the same probability to occupied rather than free, we assumed $p(x_i) = 0.5$. In this way, Eq 3.30 can be rewritten as:

$$l_t^i = l_{t-1} + \log \frac{p(x_i|z_t)}{1 - p(x_i|z_t)}. \tag{3.31}$$

By calculating the *belief* factor, function of *log odds ratio*, the algorithm assigned a value from 0 (free cell) to 1 (occupied cell) to each cell, representing the occupancy probability:

$$bel_t(x_i) = 1 - \frac{1}{1 - \exp l_t^i}. \tag{3.32}$$

#### Items placing strategy

Several bin packing strategies, addressing the problem of item sorting, have been introduced over the years in the field of operational research [188]. Although both exact algorithms [37, 135] and heuristics [69, 130, 187] have been considered, they mainly rely on a priori knowledge of the items order, besides being computationally expensive. Furthermore, these studies do not consider the variations brought by the human counterparts (e.g., changes in item sorting) and are mainly evaluated and performed in simulation studies. Hence, they could find their way into practice only with a perfect knowledge of the surrounding environment and an accurate robot trajectory planning, which are mainly typical of machine tools that do not allow for collaborative approaches.

To overcome this shortcomings, in this framework we present a novel items placing strategy, based on the observation of the human behavior presented above. To ensure flexibility during the placing phase, we assume that at every iteration the items inside the box can be subject to pose changes, for instance due to the collaboration with human agents performing the task in the same container. Mainly, the environmental variables in these situations are identified by two elements: the change of the items quantity/position inside the box and the variation of the box pose in the environment.

In order to correctly place the picked items in a box, we designed a strategy that ensures robustness to potential environmental changes between the placing of an item and its successor, based on the observations on humans performing a similar task. The human-inspired strategy includes an initial placing of the object in a collision-free area where no object/border is detected, and then a sliding motion on the box plane towards the desired goal pose identified by the algorithm.

Algorithm 2 shows the pseudo-code implementation of the method hereafter described. The algorithm receives as inputs an Occupancy Grid ($\boldsymbol{OG}$), and the diameter of the item $\boldsymbol{d}$, and returns the starting pose $\boldsymbol{start}$ where the item needs to be placed in the box, before being pushed in the direction given by the other output variable, the unit vector $\hat{\boldsymbol{u}}$. This strategy is needed to compact the items inside the box, and augments the tolerance to perception accuracy.

Although, as a proof of concept, we consider the shape of the items to be cylindrical, another item shape would not affect the algorithm core, e.g., for a square-shaped object,

**Algorithm 2** Items placing algorithm

**Input:** $\boldsymbol{OG}, \boldsymbol{d}$
**Output:** $\boldsymbol{start}, \hat{\boldsymbol{u}}$
  $\boldsymbol{rows} = \boldsymbol{OG}_{height}$
  $\boldsymbol{columns} = \boldsymbol{OG}_{width}$
  $\boldsymbol{D} = ceil(\boldsymbol{d}/\boldsymbol{OG}_{resolution})$
  **for** $\boldsymbol{r}$ = 0 to $\boldsymbol{rows}$-$\boldsymbol{D}$ **do**
    **for** $\boldsymbol{c}$ = 0 to $\boldsymbol{columns}$-$\boldsymbol{D}$ **do**
      *goalGrid* $\rightarrow$ ($\boldsymbol{r}$ : $\boldsymbol{r}$+$\boldsymbol{D}$, $\boldsymbol{c}$ : $\boldsymbol{c}$+$\boldsymbol{D}$)
      **if** *emptyGrid(goalGrid)* **then**
        $\boldsymbol{goal}_x = (c + D/2) \cdot \boldsymbol{OG}_{resolution}$
        $\boldsymbol{goal}_y = (r + D/2) \cdot \boldsymbol{OG}_{resolution}$
        *borderGrids = getBorderGrids(goalGrid)*
        **for** each *bg* in *borderGrids* **do**
          **if** *emptyGrid(bg)* **then**
            $bg_{weight} = getOuterCellsOP(bg)$
            *candidateStartGrids.add(bg)*
          **end if**
        **end for**
        **if** *size(candidateStartGrids)* $> 0$ **then**
          *startGrid = MinWGrid(candidateStartGrids)*
        **else**
          *startGrid = goalGrid*
        **end if**
        $\boldsymbol{start}_x = (startGrid_x + D/2) \cdot \boldsymbol{OG}_{resolution}$
        $\boldsymbol{start}_y = (startGrid_y + D/2) \cdot \boldsymbol{OG}_{resolution}$
        $\overrightarrow{\boldsymbol{u}} = \boldsymbol{goal} - \boldsymbol{start}$
        $\hat{\boldsymbol{u}} = \frac{\overrightarrow{\boldsymbol{u}}}{|\overrightarrow{\boldsymbol{u}}|}$
        **return** $\boldsymbol{start}, \hat{\boldsymbol{u}}$
      **end if**
    **end for**
  **end for**

we can consider its side as $\boldsymbol{d}$. The $\boldsymbol{OG}$ information, besides the occupancy probability (*OP*) of each cell, include the *height* of the grid that can be regarded as the grid $\boldsymbol{rows}$, its *width* corresponding to the number of $\boldsymbol{columns}$, its *resolution*, and its *frame* calculated with respect to the camera frame. We define $\boldsymbol{D}$ as the minimum number of grid cells where $\boldsymbol{d}$ can fit, by finding the smallest integer greater than or equal to $\boldsymbol{d}$ divided by the grid *resolution*.

The essence of the algorithm resides in the search of an $\boldsymbol{OG}$ subgrid, named *goalGrid*, of dimension $\boldsymbol{D} \times \boldsymbol{D}$ with all empty cells, i.e. with *OP* less than $0.2$, and in finding another empty subgrid, named *borderGrid*, of dimension $\boldsymbol{D} \times \boldsymbol{D}$ along the contour of the first one. Like that, the item can be first placed in the *borderGrid* and then pushed towards the *goalGrid* until a contact with the box border or with other already placed items is found. To this end, the algorithm scans all the possible *goalGrids* starting from the one localized at the bottom-right, which corresponds to the $\boldsymbol{OG}$ element *(0,0)*, assigning at each iteration a new potential *goalGrid* and checking if it is empty. Once it is found, the $\boldsymbol{goal}$ pose coordinates *(x,y)* are set as the central element of the examined subgrid multiplied by the $\boldsymbol{OG}$ *resolution*. The algorithm moves forward to look for the best $\boldsymbol{start}$ pose coordinates by looking for the *borderGrids*, that can be regarded as the

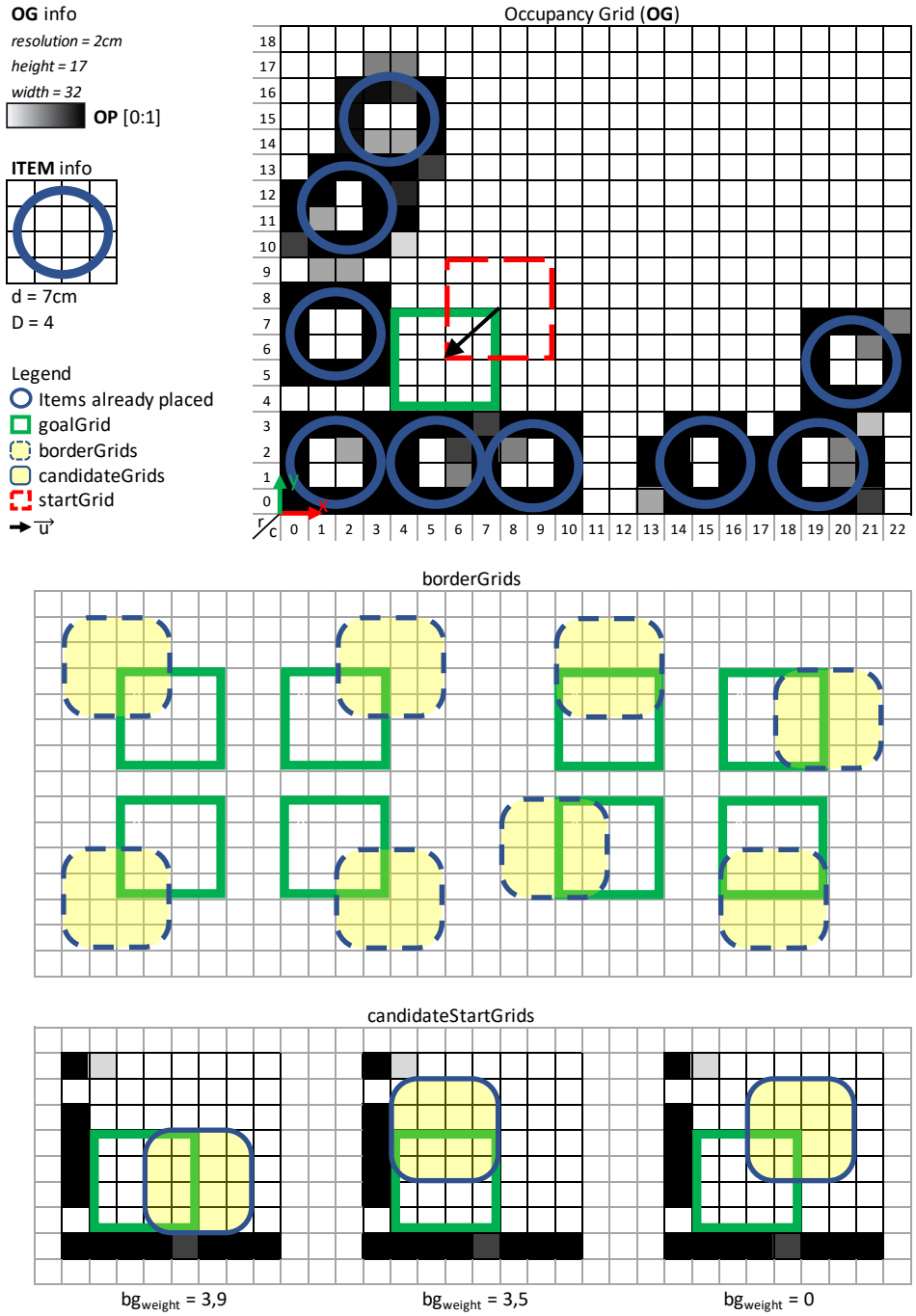


**Figure 3.26:** *Occupancy grid example (top) with the illustration of the potential borderGrids (center), and the candidateStartGrids (bottom) that would be extracted with the occupancy grid above.*

8 subgrids around the *goalGrid* (see the central part of Figure 3.26). Each *borderGrid* that is found to be empty is added to the *candidateStartGrids* list, with associated a *weight* that is the result of the sum of the *OP* of each cell laying on the external outline of the subgrid. Next, the *startGrid* is selected as the one with minimum *weight* among the *candidateStartGrids* list. If the *candidateStartGrids* list is empty, it means that only one ***OG*** subgrid is empty, so the *startGrid* is set equal to the *goalGrid*, implying a

peg-in-hole like motion. Similarly as explained above for the *goalGrid*, the $\boldsymbol{start}$ pose coordinates *(x,y)* are set as the central element of the examined subgrid multiplied by the $\boldsymbol{OG}$ *resolution*. Finally, the unit vector $\hat{\boldsymbol{u}}$ can be computed normalizing $\overrightarrow{\boldsymbol{u}}$, that is the vector connecting $\boldsymbol{start}$ and $\boldsymbol{goal}$ coordinates.

#### Finite State Machine

The design of a FSM is needed to regulate the data exchanged among the presented modules. Every state communicates with the *Trajectory planner*, asking for a desired motion (*point-to-point* or *contact-detection*) and receiving a *motion ended* acknowledgment, that serves as input to move on to the next state.

The FSM primitive motions can be summarized in three main states (depicted in Figure 3.25): pick an item from the conveyor, move it to a starting pose in the box, and place it towards the border/other items. "Pick from conveyor" sends a data request to the perception module "Conveyor localization" subunit and waits until the conveyor pose is sent back. Notice that, since a successful box detection is given only when static objects are perceived, if a human agent is simultaneously placing some items, the robot pauses its motion until he/she moves away. This enhances the framework's safety for the workers.

Once the conveyor pose is retrieved, the FSM sends to the *Trajectory planner* first a request for a *point-to-point* motion that leads the robot end-effector above the conveyor, and then a *contact-detection* motion that moves the robot down until a contact with the conveyor is detected. The robotic gripper can now grasp the item.

Next, in the "Move to start pose" state, the robot needs to reach the pose identified in Algorithm 2 as $\boldsymbol{start}$. To do so, a data request is sent to the *Box and items detection* unit, that returns the current box **OG**, containing the data of the probability occupancy of every cell grid, its height, width and resolution, and the origin of the box reference frame $\Sigma_B$. This information, along with the item diameter, are then sent to the *Items placing strategy* module that returns the $\boldsymbol{start}$ pose and the unit vector $\hat{\boldsymbol{u}}$, that are translated in the robot frame through the transformation:

$$\boldsymbol{T}_I^R = \boldsymbol{T}_C^R \boldsymbol{T}_B^C \boldsymbol{T}_I^B \tag{3.33}$$

where $\boldsymbol{T}_I^R$ represent the transformation from the item reference frame $\boldsymbol{\Sigma}_I$ to the robot reference frame $\boldsymbol{\Sigma}_R$ (see Figure 3.22).

The robot first moves above the $\boldsymbol{start}$ pose with a *point-to-point* motion, and then down until a contact with the box is detected (with a *contact-detection* motion), as in the previous state. Finally, in the "Place" state, the robot moves towards the direction given by $\hat{\boldsymbol{u}}$ with a *contact-detection* motion until a contact with the border or other items already present in the box is found. This is done activating the *impedance adaptation* strategy along the direction of the motion presented in Section 3.2.2, so as to ensure a more efficient item placing. In this case the *interaction expectancy* value depends only on the task, so on the FSM state and, in Eq. 3.15, $\boldsymbol{k}_{st}$ is kept fixed with a significantly higher value with respect to $\boldsymbol{k}_{min}$. Figure 3.27 shows the reason behind this choice: being stiff only along the direction of the motion and compliant in the others, lets the robot adapt to the constraints that can be found along the defined trajectory. The described 3-states process is repeated until there are items on the conveyor and/or there is a box with empty occupancy subgrids available.

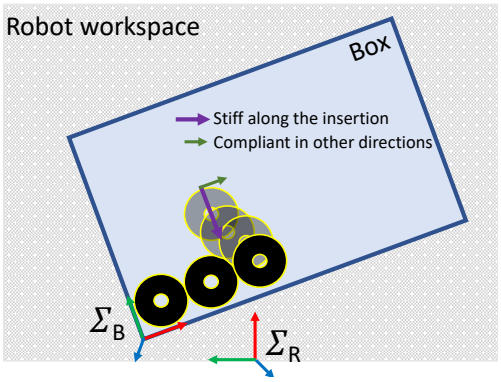


**Figure 3.27:** *The adaptive impedance controller implements a stiff profile along the motion vector, and a soft profile in other directions, to allow adaptation to the constraints along robot path.*

#### Experimental results

To validate the proposed method, we carried out experiments replicating a pick and place industrial scenario, that involves human-robot collaboration. We used a Franka Emika Panda robotic arm equipped with its original two-finger gripper, that was modified with longer fingers to allow bigger objects grasping. The presented software architecture, depicted in Figure 3.25, relies upon the ROS middleware, every module is implemented as a ROS node, and the data among them are exchanged through ROS services and messages. The perception data were streamed at a frequency of $30Hz$ through an ASUS Xtion Pro Live RGB-D sensor, that was mounted above the robot workspace facing downward, so as to have a top view of the workspace, and calibrated with respect to the robot base frame. As items, we used industrial actuator shells. Figure 3.28 shows the experimental setup (a), the camera view (b), and the experimental snapshots of the three FSM states (c): "Pick from conveyor", "Move to start pose", and "Place". In the proposed experiments, the ***OG** resolution* was set to $1cm$.

To describe the different task phases, we follow the FSM flow through its three main states. Figure 3.29 shows the data associated to an experiment that was carried out to fill in an entire box in collaboration with a human subject. The three subplots represent the desired and measured position of the robot end-effector, the sensed external forces, and the stiffness profiles, along the three Cartesian axes. For every iteration of the three FSM states, above each relative plot, we represent the state of the **OG** taken during "Move to start pose" phase, and the snapshot of the robot and the human placing the items related to the "Place" state. The first one is composed by the raw image recorded by the RGB-D camera placed above the robot, and the **OG** retrieved by the *Box and items detection* module with that image. On the grid we also represent the data computed by the *Items placing strategy* following the nomenclature of Algorithm 2: the green square represents the ***goal*** pose, the red dashed square identifies the ***start*** pose, and the black arrow constitutes the unit vector $\hat{\boldsymbol{u}}$.

In this scenario, to completely fill in the box, the FSM states were iterated 7 times, as numbered in the snapshots associated to the plots. In every iteration we can distinguish the three FSM states: "Pick from conveyor" is highlighted in light red, "Move to start pose" in light green, and "Place" in light blue. The first phase is similar for all the iterations, the robot reaches the conveyor pose and moves down on the *z-axis* until a

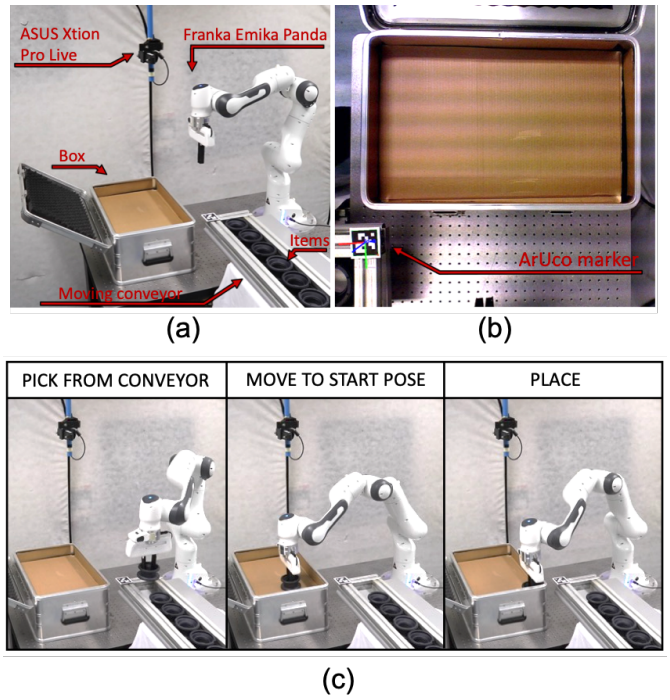


**Figure 3.28:** *(a) The experimental setup includes a Franka Emika Panda robotic arm, an ASUS Xtion Pro Live camera, a moving conveyor with motor shells, and a box placed on the robot workbench. (b) The view from the camera shows the box and the detected ArUco marker sticked to the moving conveyor. (c) Experimental snapshots of the three Finite State Machine primitives.*

contact with the conveyor is detected, i.e. when $||\boldsymbol{F}_{ext}(z)|| \geq \boldsymbol{F}_{ext,th}(z)$. The threshold $\boldsymbol{F}_{ext,th}(z)$ was set to $3N$, each time added to the different force bias sensed at the beginning of the relative motion. On the other hand, the second and the third phase change at every iteration, therefore hereafter we provide a complete description of all of them. A video of the experiment is available at [21].

*Iteration 1:* the computed $\boldsymbol{goal}$ pose lies on the bottom-left corner, since the box is empty and the algorithm starts taking into account the $\boldsymbol{OG}$ cells with row and column equals to 0. There are 3 feasible *candidateStartGrids*, but the one from the top and the one from the right are assigned with a high weight since they are close to the grid border. On the contrary, the selected $\boldsymbol{start}$ pose has $weight = 0$ since the cells on its outer border are all empty. The item, after being moved to the $\boldsymbol{start}$ pose, is pushed against the borders until a contact on $x$ and $y$ is detected ($time = 20s$). The threshold $\boldsymbol{F}_{ext,th}(x, y)$ was set to $10N$, projected along the motion vector direction. Therefore in this case $\boldsymbol{F}_{ext,th}(x) \approx \boldsymbol{F}_{ext,th}(y) \approx 7N$ (also here added to the different force bias sensed at the beginning of the relative motion). In the third subplot, the stiffness profile adaptation is depicted, the robot is stiffer along the motion vector and more compliant in the other directions. In Eq. 3.15, $\boldsymbol{k}_{st}$ was set to $1400N/m$ and $\boldsymbol{k}_{min}$ to $200N/m$.

*Iteration 2:* similar to the previous iteration. In addition, a subject starts to add items in the box as well, as can be seen from the picture. The 2 items added by the human are also visible in the next iteration, where the detected $\boldsymbol{OG}$ includes 4 items.

*Iteration 3:* the computed $\boldsymbol{goal}$ pose lies on the bottom-right corner, and the only feasible *candidateStartGrid* is represented by the depicted $\boldsymbol{start}$ pose. The "Place"

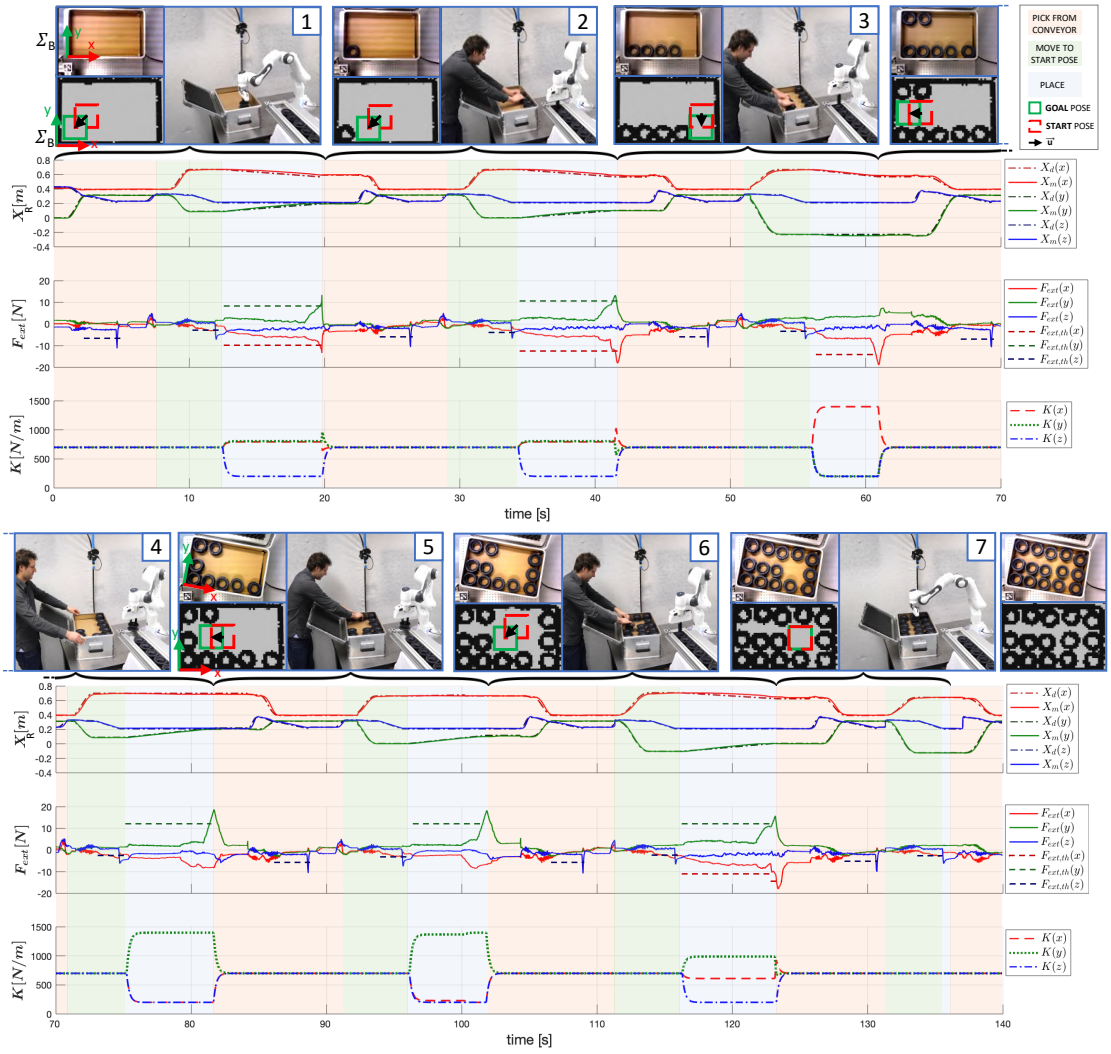


**Figure 3.29:** *The robot and the human agent fill in the box in collaboration. The plots show the desired and measured position in the robot reference frame* $\boldsymbol{\Sigma_R}$*, the sensed external forces, and the stiffness profiles, along the Cartesian axes. The experimental snapshot, the camera view, and the occupancy grid are shown above the plots.*

motion is only along the robot *x-axis*, as can be noticed also in the impedance parameters regulation. Being compliant on the other axes ensures the success of the motion, since the item can be inserted in the empty spot regardless of any potential obstruction, given either by the other item or by the right border. In parallel, the subject places other 2 items in the top-left part of the grid.

*Iteration 4:* similarly as in iteration 3, only one feasible *candidateStartGrid* is found by the *Items placing strategy*. This time, the motion takes place only along the robot *y-axis*. At the end of this phase, the subject moves the position of the box on the workbench. This has been done to show the flexibility of the framework, robust to changing conditions. In fact, the box pose is computed at the beginning of every iteration, and it is part of the information associated to the $\boldsymbol{OG}$.

*Iteration 5:* the *Items placing strategy* outcome is similar to the one computed in the previous iteration. In addition, here we can see that the stiffness parameters are tuned

also along the robot *x-axis*, since, despite the motion on the grid looks identical, the box has been rotated. During the "Place" state, the human puts other 2 items on the grid top-right.

*Iteration 6:* the computed unit vector $\hat{\boldsymbol{u}}$ is similar to the one of the first 2 iteration. However, due to the previously mentioned box rotation, the direction of the motion in the world frame is quite different. At the end of the "Place" state ($time \approx 123s$), we can see that the contact on $y$ is detected slightly before the one along the robot *x-axis*, so $\boldsymbol{K}(x)$ increases and $\boldsymbol{K}(y)$ decreases. Also $\boldsymbol{F}_{ext,th}(x)$ raises since, once the contact along $y$ is detected, the motion on that direction is stopped and continues only along $x$. At the end of the robot motion, the subject places other 2 items in the box.

*Iteration 7:* only one empty spot is left on the grid. In the *Items placing strategy*, since no *candidateStartGrid* is found, the $\boldsymbol{start}$ pose is set equal to the $\boldsymbol{goal}$ pose. The third state, "Place", involves only the item ungrasping, since, after the contact detected on the *z-axis* during the previous state, there is no need to move it on the *xy-plane*.

**Comparison to the stiff controller** To better show the advantages of the online impedance regulation along the motion vector, we performed an additional experiment starting from the same condition, with and without the impedance parameters adaptation. Figure 3.30 shows on the left the experiment carried out with the impedance regulation enabled, as can be seen from the third subplot. The placing motion does not stop where the $\boldsymbol{goal}$ pose was estimated, but it continues along the robot *x-axis* until the border of the box is found. Being compliant on the *y-axis* makes the robot insert the piece between the two items in the box bottom-right. In the first subplot, we can notice that the desired motion on $y$ does not change while the measured one moves away, thanks to the above-mentioned compliance. The two camera snapshots, taken before and after

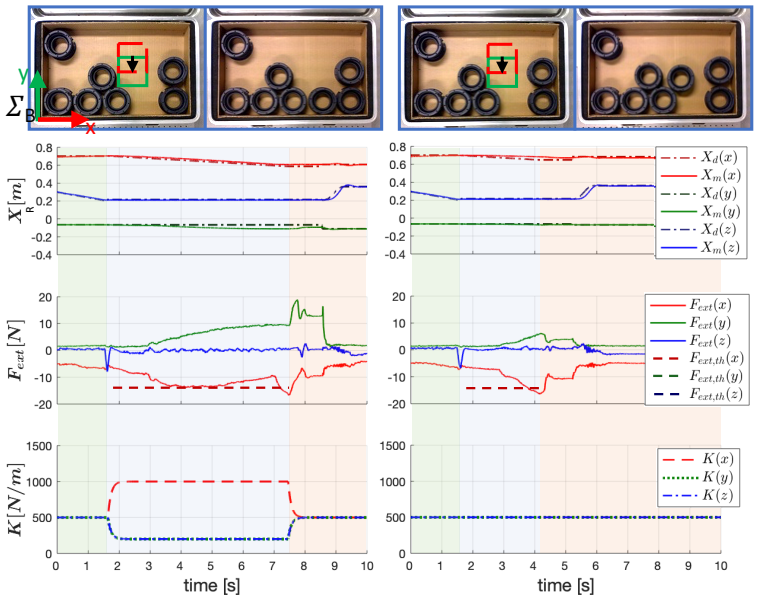


**Figure 3.30:** *Starting from the same* $\boldsymbol{OG}$ *initial condition, the same experiment repeated twice: on the left with the impedance parameter adaptation, on the right without. Being stiffer along the motion vector, and more compliant in the other directions, improves the complete filling of empty spaces.*

the plotted data, are shown above the plots.

On the other hand, when the online tuning of the impedance profiles is not active, the situation explained above does not hold anymore. When the estimated $\boldsymbol{goal}$ pose is reached, the sensed external forces on the robot *x-axis* go beyond the threshold $\boldsymbol{F}_{ext,th}(x)$ ($t = 4s$), and the item is released from the gripper.

### 3.3.4 Kinodynamically constrained vehicles repositioning

In this section, we present another adaptive planning strategy blended with the robot control strategy presented above, employed in a logistics scenario, with the aim of extending the robot capabilities in the direction of flexible warehouse automation. In fact, the role of automation in logistics has been growing more and more in the last decades. This is due to the increasing demands imposed by the "e-commerce revolution" [132], whose constant progress cannot be responded by the decreasing available workforce in aging societies. The target activities with automation potential are mainly those that do not require a significant level of human expertise, such as handling, sorting, storage, and transportation, typical of warehouse environments.

Recent developments in robotized and automated warehouse systems have presented advantages such as full-time availability, and savings on space, labor costs, and on other operational costs, such as heating and lighting [15]. Nevertheless, the existing solutions require large initial investments, since they often need to be applied to customized production processes. Additionally, these solutions are not flexible enough, so they can be hardly employed in dynamically changing environments such as small and medium sized warehouses. For this reason, only 5% of the existing warehouses worldwide have been semi-automated [33].

One fundamental roadblock towards extending the automation capacity of the warehouses has been posed by the automatic yet flexible management of the pallets, which demand for their recognition, manipulation, and relocation (transportation and positioning through pallet jacks). Current solutions have aimed to tackle this through bespoke machines, designed for specific purposes. For instance, in [178], a voice-commandable prototype has been introduced, to enable human supervisors to use speech and pen-based gestures to assign tasks to the forklift. Other similar works focused on forklifts control [177, 199], pallet manipulation [122, 167], and pallet recognition [32, 44], to separately tackle the issues related to the pallet repositioning tasks. Although such systems can be beneficial in large and fully autonomous warehouses, however, they impose additional costs and equipment waiting times to the others. In addition, since most of the small and medium sized warehouses have been designed and created for humans, the introduction of such automation systems would require a major re-design of the spaces, affecting their economic sustainability.

Cobots, on the other hand, have demonstrated the potential to execute a large range of manufacturing and logistics tasks alongside their human partners [6], and add certain flexibility levels to the processes [26]. In logistics scenarios in particular, they can reconfigure from autonomous picking and palletizing [112] to navigating in warehouse environments [134, 198].

Towards understanding the potential of cobots in addressing the pallet management problem, in this section, we propose a novel loco-manipulation control framework for the execution of complex tasks with kinodynamic constraints using mobile manipula-

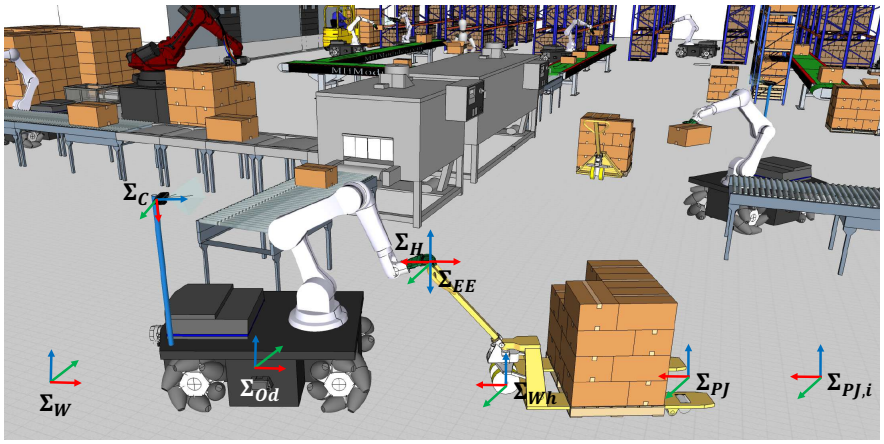


**Figure 3.31:** *Concept illustration of a mobile collaborative robot while transporting and positioning of a manual pallet jack inside a warehouse. The depicted reference frames represent world (W), platform odometry (Od) and end-effector (EE), camera (C), pallet jack handle (H), wheels (Wh), current pose (PJ) and initial pose (PJ,i).*

tors. As a representative example, we consider the handling and re-positioning of pallet jacks (or lifts/carriers with similar characteristics) in unstructured environments. This task is associated with significant challenges in terms of locomotion, due to the mobility constraints that are imposed by their limited kinematics while moving, and manipulation, due to the existence of dynamic uncertainties while grasping and handling of pallet jacks. To tackle these challenges, our solution enables the robotic platform to autonomously reach a pallet jack location while avoiding the obstacles, and to detect and manipulate its handle by fusing the perception and the contact force data. Subsequently, the transportation of the pallet jack is achieved through a prioritized weighted whole-body inverse dynamics control algorithm (see Appendix Section A.1), and a trajectory planner which takes into account the mobility constraints of the robot-pallet jack chain.

To the best of the authors' knowledge, all the existing frameworks tackling the pallet transportation and positioning problem are related to autonomous forklifts development [177, 189]. Instead, in this work, we propose a solution designed for a *multi-purpose platform*, MOCA, that has a high potential to perform several logistics tasks. It is important to notice that, this work does not claim to compare the performances that can be achieved with unmanned forklifts, but aims to provide a solution to this problem with a general purpose robot that can be employed for many applications. In small and medium-sized enterprises this is a crucial point, since they may not be able to afford multiple single-purpose, costly machines. On the other hand, they can exploit the flexibility of a multi-purpose machine, relieving the cost burden. In fact, the reconfigurability of this platform has already demonstrated to be effective also in other industrial scenarios, i.e., in human-collaborative manufacturing [107] and palletizing [119] tasks, and in teleoperation applications [191] (presented in Section 4.2.1).

The presented framework can be subdivided into four operational modules, that include a *Pallet jack reaching* unit to navigate towards a desired pallet jack location, a *Handle detection* unit to localize more accurately the pallet jack handle, a *Handle preparation* unit to grasp and pull down the pallet jack handle, and a *Pallet jack repositioning* unit to move the pallet jack to a new location in the workspace. Before explain-

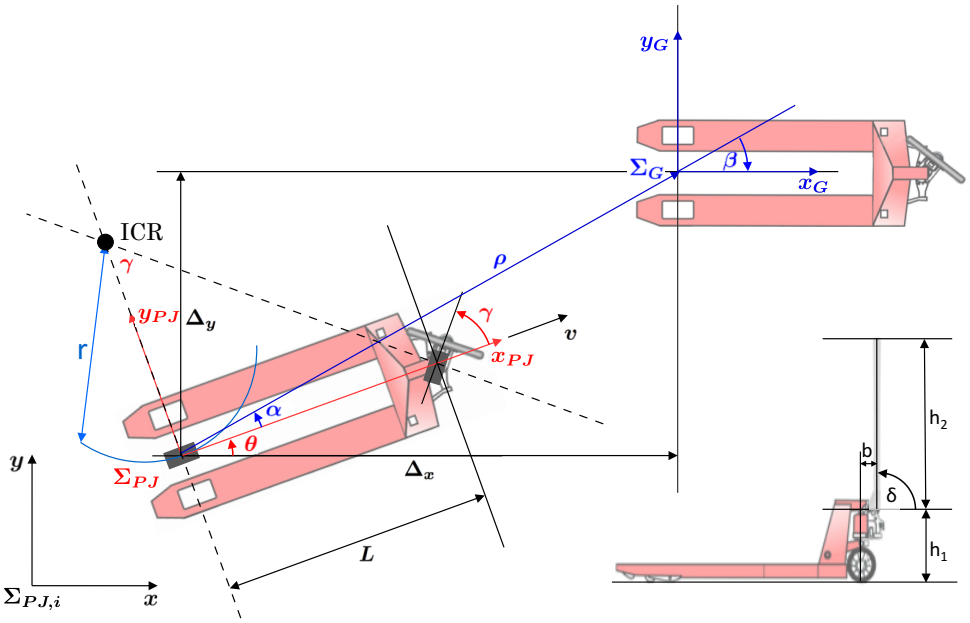


**Figure 3.32:** *Bicycle model applied to the pallet jack. The two dark grey wheels represent the bicycle approximation. $\gamma$ represented the steering wheel angle. The two wheel axes are extended by the dashed lines and intersect at the Instantaneous Center of Rotation (ICR). $\Delta_x$ and $\Delta_y$ represent the error between $\Sigma_{PJ}$ and $\Sigma_G$ in Cartesian coordinates. The distance between $\Sigma_{PJ}$ and the goal pose $\Sigma_G$ is denoted with$\rho$, $\alpha$ is the angle between $x_{PJ}$ and $\rho$, and $\beta$ is the angle of $\rho$ with respect to $\Sigma_{PJ,i}$. In the right bottom part, the lateral view is shown with the corresponding notation.*

ing the details of these modules, we introduce how the trajectory planning of vehicles with kinematic constraints was implemented. The results reported in this section led to the scientific publication [17].

#### Kinematically-constrained trajectory planning for holonomic robots

In this paragraph we want to address the problem of generating trajectories for an holonomic robot that is pulling a wheeled object. The goal consists in designing trajectories for the end-effector of the robot, given the initial and goal pose of the pulled object, and the kinematics constraints that the object imposes on the range of possible motions of the holonomic robot. In this particular scenario, the pulled wheeled object is a pallet jack. To be able to embed its kinematics limitation onto the trajectory generator, we model the kinematic structure of the pallet jack and how it is coupled with the robot end-effector.

**Pallet jack kinematic model and trajectory generation** The pallet jack kinematic structure is composed by two rear fixed wheels and one front steered wheel. A quite common kinematic model for this structure is represented by the *bicycle model* (Figure 3.32) [43]. In this model, the configuration of the vehicle is represented by the generalized coordinates $\mathbf{q} = \begin{bmatrix} x & y & \theta \end{bmatrix}^\top$, where the $\mathbf{q}$ is described by the vehicle frame $\Sigma_{PJ}$ in the frame $\Sigma_{PJ,i}$ shown in Figure 3.32, with its *x-axis* in the vehicle's forward direction and its origin in the center of the rear fixed wheel axle. Its velocity is defined as $v$ in the vehicle's $x$-direction, zero in the $y$-direction (the wheels cannot slip sideways). In the

vehicle frame $\mathbf{\Sigma}_{PJ}$:

$$\dot{x}^{PJ} = v, \quad \dot{y}^{PJ} = 0. \tag{3.34}$$

The dashed lines in Figure 3.32 are the no motion lines, so the direction along which the wheels cannot move. They intersect in a point called Instantaneous Centre of Rotation (ICR). The vehicle frame $\mathbf{\Sigma}_{PJ}$ follows a circular path with radius $r$ around ICR with an angular velocity:

$$w = \frac{v}{r}, \quad r = \frac{L}{\tan\gamma}. \tag{3.35}$$

Transforming the velocities in the frame $\mathbf{\Sigma}_{PJ,i}$ and applying Eq. 3.35, we obtain the motion equations:

$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= \frac{v}{L}\tan\gamma \end{aligned} . \tag{3.36}$$

As we can notice, when $v = 0$, then $\dot{\theta} = 0$, so it is not possible to change orientation without applying a linear velocity. If $\gamma = \pi/2$, then the vehicle cannot move forward (the front wheel is orthogonal with respect to the back one) and the model enters in an undefined region.

Defined the kinematics, now the problem is to drive the pallet jack to a desired pose $(x^*, y^*, \theta^*)$. With the aim of generating feasible trajectories for the pallet jack model, a controller is introduced. We consider the coordinate transformation into polar coordinates using the notation shown in Figure 3.32 [43]:

$$\begin{aligned} \rho &= \sqrt{{\Delta_x}^2 + {\Delta_y}^2} \\ \alpha &= atan2(\Delta_y, \Delta_x) - \theta \cdot \\ \beta &= -\theta - \alpha \end{aligned} \tag{3.37}$$

This results in a system description, in the new polar coordinates

1. if $\alpha \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right]$ (the goal is in front of the vehicle):

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -\cos\alpha & 0 \\ \frac{\sin\alpha}{\rho} & -1 \\ -\frac{\sin\alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}, \tag{3.38}$$

2. if $\alpha \in \left(-\pi, -\frac{\pi}{2}\right] \cup \left(\frac{\pi}{2}, \pi\right]$ (the goal is behind the vehicle):

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} \cos\alpha & 0 \\ -\frac{\sin\alpha}{\rho} & 1 \\ \frac{\sin\alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}. \tag{3.39}$$

Now we need to design the control signals $v$ and $w$ to drive the vehicle to a unique equilibrium at $(\rho, \alpha, \beta) = (0, 0, 0)$. Moreover, to avoid discontinuity at $\rho = 0$, we define the linear control law as:

$$\begin{aligned} v &= \sigma k_\rho \rho \\ w &= \sigma(k_\alpha \alpha + k_\beta \beta). \end{aligned} \tag{3.40}$$

where the terms $k_\rho\rho$ and $k_\alpha\alpha$ drive the vehicle along a line toward $\boldsymbol{\Sigma}_G$, $k_\beta\beta$ rotates the line such that $\beta \to 0$ and $\sigma$ is the direction of the motion: $+1$ if the goal is in front, $-1$ otherwise. Combining Eqs. 3.38 and 3.40, we obtain the closed-loop system:

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -\sigma k_\rho \rho \cos\alpha \\ -\sigma(-k_\rho \sin\alpha + k_\alpha \alpha + k_\beta \beta) \\ -\sigma k_\rho \sin\alpha \end{bmatrix}. \tag{3.41}$$

This controller drives the vehicle to the pose $(0, 0, 0)$. To drive to an arbitrary pose $(x^*, y^*, \theta^*)$, we apply a change of coordinates:

$$x' = x - x^*, \quad y' = y - y^*, \quad \theta' = \theta, \quad \beta = \beta' + \theta^*. \tag{3.42}$$

The control parameters that makes the closed-loop system stable are:

$$k_\rho > 0, \quad k_\beta < 0, \quad k_\alpha - k_\rho > 0. \tag{3.43}$$

However, since we aim for a robust position control, we want to ensure that the vehicle does not change direction during its approach to the goal. To this end, we apply the strong stability condition [170]:

$$k_\rho > 0; \quad k_\beta < 0; \quad k_\alpha + \frac{5}{3}k_\beta - \frac{2}{\pi}k_\rho > 0. \tag{3.44}$$

**End-effector trajectory planner** Since we want to compute the trajectory to be carried out by the pallet jack expressed with respect to its initial frame, $\boldsymbol{\Sigma}_{PJ,i}$, we first need to derive its transformation with respect to the world frame $\boldsymbol{\Sigma}_W$:

$$\boldsymbol{T}^W_{PJ,i} = \boldsymbol{T}^W_{Od}\boldsymbol{T}^{Od}_{EE}\boldsymbol{T}^{EE}_{H}\boldsymbol{T}^{H}_{Wh}\boldsymbol{T}^{Wh}_{PJ}, \tag{3.45}$$

where $\boldsymbol{T}^W_{Od}$ represents the robot odometry, i.e. the transformation from $\boldsymbol{\Sigma}_W$ to $\boldsymbol{\Sigma}_{Od}$, and $\boldsymbol{T}^{Od}_{EE}$ the transformation from $\boldsymbol{\Sigma}_{Od}$ to the end-effector frame $\boldsymbol{\Sigma}_{EE}$. Since we suppose that after the grasping the end-effector is rigidly coupled with the pallet jack handle, the translational component of their frames, $\boldsymbol{\Sigma}_{EE}$ and $\boldsymbol{\Sigma}_H$, results to be the same, while the orientation of the handle is rotated by $\pi\ rad$ on the *y-axis*:

$$\boldsymbol{T}^{EE}_{H} = \begin{bmatrix} \boldsymbol{R}_y(\pi) & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{3.46}$$

The transformation of the wheel frame $\boldsymbol{\Sigma}_{Wh}$ with respect to the handle frame $\boldsymbol{\Sigma}_H$ is calculated as:

$$\boldsymbol{T}^{H}_{Wh} = \begin{bmatrix} \boldsymbol{I}_{3\times3} & \begin{bmatrix} x^H_{Wh} \\ x^H_{Wh} tan\gamma \\ z^H_{Wh} \end{bmatrix} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{3.47}$$

where $x_{Wh}^{H}$ and $z_{Wh}^{H}$ are computed as:

$$\begin{aligned} x_{Wh}^{H} &= -(b + h_2 cos\delta) \\ z_{Wh}^{H} &= -(h_1 + h_2 sin\delta). \end{aligned} \tag{3.48}$$

Lastly, the transformation from the pallet jacket to its wheels is constant and represented by:

$$\boldsymbol{T}_{PJ}^{Wh} = \begin{bmatrix} & \begin{bmatrix} -L \\ 0 \\ 0 \end{bmatrix} \\ \boldsymbol{I}_{3\times3} & \\ \mathbf{0} & 1 \end{bmatrix}. \tag{3.49}$$

After modeling and designing a controller for the pallet jack, we have to bring it to a desired pose taking into account it is not actuated. Giving as input a desired pose $(x^*, y^*, \theta^*)$, the trajectory is simulated through the controller. From this, we obtain the pose $(x, y, \theta)$ and hence $\boldsymbol{T}_{PJ}^{PJ,i}$, and the steering wheel angle $\gamma$, at each time-step. At this point, it is possible to retrieve the robot end-effector desired trajectory $\boldsymbol{T}_{EE}^{W}$ through Eq. 3.45:

$$\begin{aligned} \boldsymbol{T}_{PJ}^{W} &= \boldsymbol{T}_{PJ,i}^{W} \boldsymbol{T}_{PJ}^{PJ,i} \\ \boldsymbol{T}_{PJ}^{EE} &= \boldsymbol{T}_{H}^{EE} \boldsymbol{T}_{Wh}^{H} \boldsymbol{T}_{PJ}^{Wh}. \\ \boldsymbol{T}_{EE}^{W} &= \boldsymbol{T}_{PJ}^{W} (\boldsymbol{T}_{PJ}^{EE})^{-1} \end{aligned} \tag{3.50}$$

#### Operational modules

The proposed autonomous pallet jack recognition and handling framework can be divided into four main execution modules, as shown in Figure 3.33: (A) a *Pallet jack reaching* unit in which the mobile robot navigates towards a desired pallet jack location while avoiding obstacles, (B) a *Handle detection* unit that combines the data coming from a perception module and the interaction forces at hand, (C) a *Handle preparation* unit responsible for grasping and pulling down the pallet jack handle, and (D) a *Pallet jack repositioning* unit to move the pallet jack to a new location in the workspace.

**Pallet jack reaching** The presented system is designed to be operated in environments like warehouses, where typically a point-to-point direct motion is not feasible for autonomous robots. As mentioned earlier, human workers, pallets, and other obstacles must be taken into account in planning of the robot mobility. Therefore, the obstacle avoidance algorithm presented above, capable of preventing collisions with both fixed and moving agents, is integrated into the *Pallet jack reaching* module. The ROS package "move_base", which utilizes the ROS Global Planner along with the Timed-Elastic-Band (TEB) local planner [159], was used for this implementation, which eliminates the need for a map initialization. Given the target location, that can be retrieved in multiple ways, e.g. through a beacon system, exploiting cameras placed around the area, or using any other Indoor Positioning System (IPS), the algorithm is able to generate the mobile base velocity commands and navigate it through the unknown area. This is done by updating a cost map which fuses the data perceived by the perception sensors, such as lasers and cameras, and the odometry information. The data received

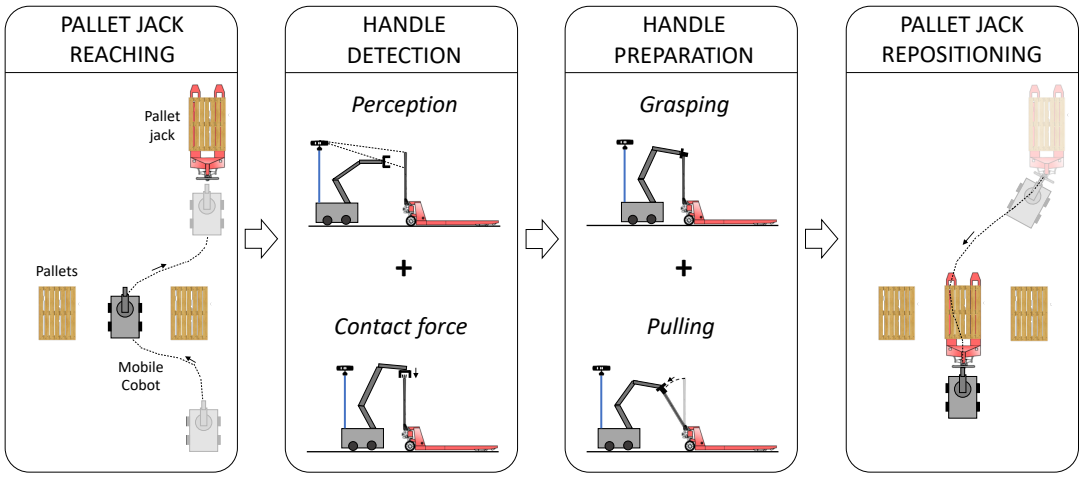


**Figure 3.33:** *The framework consists of four main modules, that also represent the different phases of a pallet repositionning task. At first, the mobile cobot reaches the pallet jack location while avoiding the obstacles along its path. Fusing the data coming from perception and contact forces, MOCA is able to detect the pose of the pallet jack handle, that is grasped and pulled in the next phase. Lastly, the pallet jack is transported to the new desired location within the warehouse.*

from the sensors is employed to either insert or remove obstacle information into/from the costmap. Then, the costmap occupied cells are converted into a set of non-convex (concave) polygons in the following steps:

- Clusters are determined using the DBSCAN Algorithm [55].
- Clusters are converted to convex polygons and then to concave ones [149].

The obstacle poses are updated at each time-step. Thus, the TEB local planner is able to calculate a collision-free path related to the received information as explained in the Appendix Section A.2.

**Handle detection** Subsequent to locating the robot in front of the pallet jack, its handle pose must be detected with a high level of accuracy to ensure a successful prosecution of the next phases. To this end, we designed a handle detection algorithm that combines the vision and the interaction forces data.

*Perception:* the following section explains the vision algorithm implemented in the *handle detection* phase. To be able to detect and manipulate the handle, the mobile manipulator should identify the grasping 3D point along the pallet jack handle, together with the shaft orientation to allow the mobile robot to approach the pallet jack from a suitable angle. These two variables constitute the handle pose $\boldsymbol{T}_H^W$.

A simple approach to detect the handle grasping pose was to attach a colored band on the handle and proceed with the color segmentation of the colored point cloud. However, this approach is subject to the sensitivity of pixel colors to the environmental light. Moreover, the detection of a particular color in the scene depends on a set of color-defining parameters that have to be preset carefully. Therefore, this approach was considered unfeasible for the real warehouse environments, since, in addition to these issues, it requires also customization of the pallet jacks.

As reported in [79], learning-based and template matching techniques are among the most practical yet accurate for object recognition. Learning-based techniques are very strong in object recognition, even for a multiple of them, but require expensive training and can be an over-fit to our relatively simple handle detection problem. On the other side, the template matching based algorithms can ensure a good approximation of the object position and orientation in the space. This technique also allows us to select a certain point on the template, for instance, a feasible grasping point on the handle, which leads to a simple handle grasping strategy. It also eliminates the need for any additional customization of the standard pallet jacks.

To apply this technique, first, a point cloud model of the handle was extracted from the scene. Following [129], a cluster extraction was performed from the observed depth point cloud, in order to select the candidates for the template alignment. Then, thanks to SHOT estimator algorithm, the SHOT descriptors are extracted both from the model and from the scene [181]. The correspondences between the model and the scene are computed by means of KdTree, a space-partitioning data structure that enables efficient nearest neighbor searches [141]. Finally, to evaluate the correspondences, we applied geometric consistency grouping, that is a clustering algorithm that enforces simple geometric constraints between pairs of correspondences [38]. The output of the algorithm consists in a transformation matrix from the camera reference frame to a new reference, i.e., the rotated original frame[4] of the template, which aligns the model cloud with the real handle in the environment. Once the template fitting was performed, a rigid transformation from the new reference frame to the grasping point on the template was applied. The algorithm's output matrix $\boldsymbol{T}_H^C$, which corresponds to the pose of the handle in the camera frame $\boldsymbol{\Sigma}_C$, was transformed in the world frame $\boldsymbol{\Sigma}_W$ by:

$$\boldsymbol{T}_H^W = \boldsymbol{T}_{Od}^W \boldsymbol{T}_C^{Od} \boldsymbol{T}_H^C, \tag{3.51}$$

$$\boldsymbol{T}_C^{Od} = \begin{bmatrix} \boldsymbol{R}_z(-\frac{\pi}{2})\boldsymbol{R}_x(-\frac{\pi}{2}+\boldsymbol{\phi}) & \boldsymbol{p}_C^{Od} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{3.52}$$

where $\boldsymbol{\phi}$ is the camera angle around the *x-axis* of its frame.

*Contact force:* once the handle position $\boldsymbol{T}_H^W$ is localized, the robot end-effector must reach the handle from the top in order to grasp it firmly. The whole-body impedance controller is activated, with low impedance values to smoothly respond to unexpected collisions, while a higher mobility is assigned to the *arm* through Eq. A.10. The robot is first sent to $\hat{\boldsymbol{T}}_H^W$, that represents $\boldsymbol{T}_H^W$ translated of $8cm$ on the *z-axis*. From $\hat{\boldsymbol{T}}_H^W$, the robot end-effector starts to lower its position on the *z-axis* until the interaction forces on that axis, i.e., $F_z$, reach the threshold $F_{z,th}$, which identifies a physical contact.

**Handle preparation** Once the contact with the handle is established, the pallet jack needs to be prepared for the final phase. This state is subdivided into two actions.

*Grasping:* this primitive simply makes the gripper/robotic hand close. In this way the robot and the pallet jack are physically coupled and they can be regarded as part of the same kinematic chain.

*Pulling:* the goal of this state is to make the pallet jack handle rotate $\delta_p rad$, so as to enable the pallet jack repositioning in the next phase. To do so, a circular trajectory with origin at the handle joint is designed exploiting Eq. 3.48 with $\delta$ going from

[4] The reference frame of the model cloud when it was recorded.

$\pi/2rad$ to $\delta_p$. In order to ensure better interaction performances while pulling the handle, the self-tuning impedance controller presented in Section 3.2.2 is activated, and the impedance parameters are tuned in such a way they are more stiff on the direction of the motion, and more compliant along the other axes. From now on, the controller weight matrix $\boldsymbol{H}$, introduced in Eq. A.10, is designed in such a way all the mobile base rotations about the *z-axis* are reduced, while, at the same time, the motion of the base is boosted with respect to the arm motion. In order to do that, we set different weights in correspondence of the virtual torques of the mobile platform:

$$\boldsymbol{H} = \begin{bmatrix} diag\{\eta_{B_x}, \eta_{B_y}, \eta_{B_{yaw}}\} & \mathbf{0}_{m\times n} \\ \mathbf{0}_{n\times m} & \eta_A \boldsymbol{I}_{n\times n} \end{bmatrix}, \tag{3.53}$$

where $\eta_{B_{yaw}} \gg \eta_A \geq \eta_{B_x} = \eta_{B_y}$. In this way, high virtual torques related to the yaw of the mobile base are discouraged, in favor of other DoF. In the particular case of $\eta_A > \eta_{B_x} = \eta_{B_y}$, the motion of the base platform in the *xy-plane* is dominant, and if the end-effector frame has to perform a rotation about the *z-axis*, the controller will try to actuate more the arm joints than the wheels. Noteworthy, the algorithm does not constrain the motion of the robot: whenever the rotation is not achievable just performing an arm motion, the wheels will be actuated.

**Pallet jack repositioning** The pallet jack is now ready to be relocated in the desired pose of the workspace. First, a trajectory is generated by taking into account the kinematic constraints of the pallet jack, as explained in the paragraph *Kinematically-constrained trajectory planning for holonomic robots*. Next, the whole-body impedance controller executes the trajectory with the aforementioned weights. Once the pallet jack desired goal is reached, the task is accomplished.

#### Experimental results

Before conducting experiments on the real hardware, we validated the repeatability and the accuracy of the pallet jack repositioning algorithm in a simulation environment. Figure 3.34 illustrates four typical scenarios with different ending desired positions and orientations. These simulation results provided a primary evidence in achieving

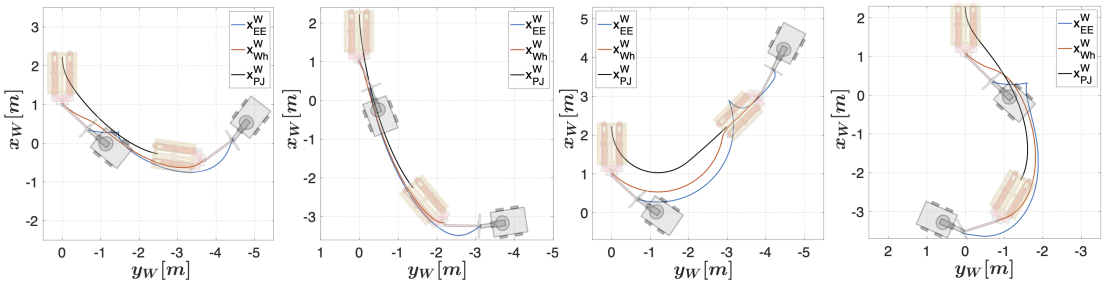


**Figure 3.34:** *Simulation results: the four plots show the trajectories for different goals starting from the same configuration , i.e.* $\boldsymbol{q}_{PJ}^W = [2.2\ m\ \ 0.0\ m\ \ \pi\ rad]^\top$. *From left to right, the final desired configurations were set to:* $\boldsymbol{q}_{PJ}^W = [-0.2\ m\ \ -2.5\ m\ \ -\pi/2\ rad]^\top$, $\boldsymbol{q}_{PJ}^W = [-2.2\ m\ \ -1.4\ m\ \ -3\pi/4rad]^\top$, $\boldsymbol{q}_{PJ}^W = [2.3m\ -3.0m\ -\pi/4rad]^\top$, $\boldsymbol{q}_{PJ}^W = [-2.2m\ -1.4m\ 3\pi/4rad]^\top$. *The average errors of the final desired position and orientation across all trials were* $0.0199m$ *and* $0.1262rad$, *respectively.*

the pallet jack desired pose with high accuracy in the repositioning algorithm. Notice that, the sketches of the pallet jack and the robot were added at a post-processing stage on the plots, to foster a deeper understanding of the results.

Next, we conducted experiments inside a real warehouse, using a pallet jack with loaded pallets to demonstrate the validity of the proposed method in a typical industrial environment. The pallet jack used in our experiments is a Tractel Pioneer 2500, weights $87Kg$, and has these physical characteristics, following the notation of the Figure 3.32: $L = 1.2\ m$, $h_1 = 0.4\ m$, $h_2 = 0.85\ m$ and $b = 0.09\ m$. An ASUS Xtion Pro RGB-D camera is used by the vision module, and was mounted on a pole attached to the backside of MOCA (see Figure 3.33). In the presented experiments, the values in Eq. 3.51 were set to: $\boldsymbol{p}_C^{Od} = [-0.225\ m,\ -0.183\ m,\ 1.56\ m]^\top$ and $\boldsymbol{\phi} = 0.24\ rad$. The software architecture relies upon ROS, using C++ and Matlab as client libraries.

To clearly show the validity of the framework, we first illustrate the performance of the proposed Weighted WB impedance controller for repositioning of a pallet jack. Next, an overview of the results of the four phases of the experiments (presented in previous sections), is provided.

For the *Pallet jack repositioning* phase, the controller's parameters to calculate the trajectory of the pallet jack were set to

$$k_\rho = 0.2, \quad k_\beta = -\frac{6}{5}, \quad k_\alpha = -\frac{5}{3}k_\beta + \frac{2}{\pi}k_\rho + 1.$$

Moreover, we decided to saturate the pallet jack linear velocity $v$ at $0.3m/s$ and the steering wheel angle $\gamma$ at $\pi/4rad$.

We initially carried out this phase with two different controllers: the whole-body impedance controller introduced in Section 2.3.2 (WB controller), and its improved version presented in Section A.1 that includes the prioritized weighted inverse dynamics algorithm (Weighted WB controller). In the latter, in order to limit the rotational movements of the robot base, so as to achieve better tracking performances, we set the weights of Eq. 3.53 as $\eta_A = \eta_{B_x} = \eta_{B_y} = 1$, $\eta_{B_{yaw}} = \sqrt{20}$. Figure 3.35 shows the different behaviors of the two controllers with the same pallet jack desired goal, $\boldsymbol{q}_{PJ}^W = [0.1\,m\ 4.0\,m\ \pi/2\,rad]^\top$, represented in the world frame $\boldsymbol{\Sigma}_W$. As can be noticed from Figure 3.35a, the desired end-effector pose $x_{dEE}^W$, projected on the *xy-plane*, is tracked more accurately with the Weighted WB controller. In fact, with the WB controller, the robot trajectory is subject to continuous variations as can be also noticed in Figure 3.35b, where the position errors of the two controllers are compared. In the third subplot, after an initial transient of $15s$, the Weighted WB controller position error norm (blue curve) falls below the threshold of $3cm$, while the one related to the WB controller (red curve) is still subject to high fluctuations for a longer time producing a chattering phenomenon. The Cartesian controller generates a force according to the Cartesian tracking error. At first, when this error is small, due to the high compliance, such force is not enough to pull the pallet jack, because of its load and friction. As a result, the tracking error increases until the generated force overcomes friction and gravity, moving the pallet jack, and hence, reducing again the tracking error. The WB controller generally tries to exploit all the DoF of MOCA to achieve the desired motion, combining highly dynamical arm motion with slower base motion. For instance, the rotation of the end-effector around the *z-axis* is achievable by means of combined arm motion and the third joint of the base. The effect of the base motion results slower

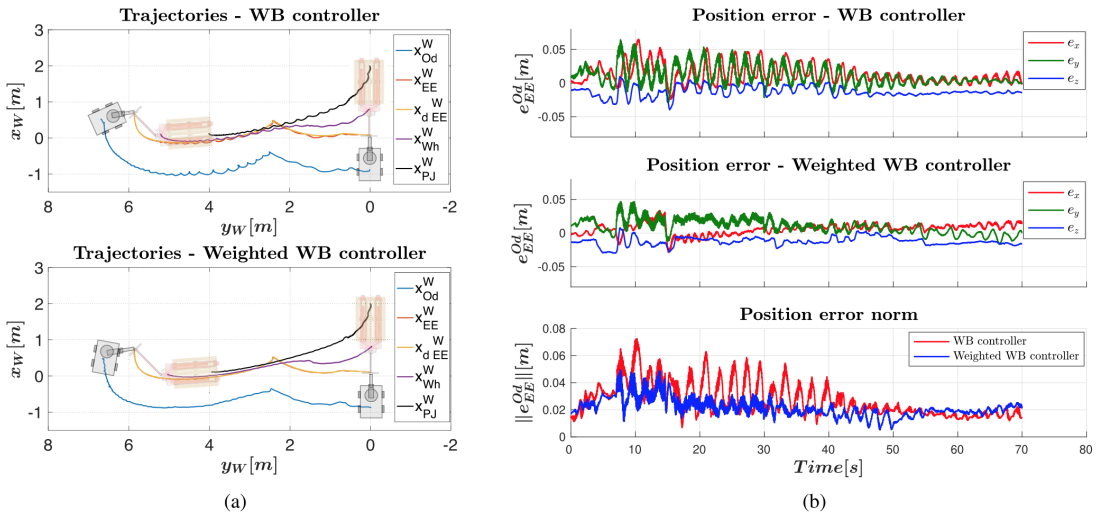


**Figure 3.35:** *Pallet jack repositioning phase: trajectory (a) and position error (b) comparison between the whole-body controller (WB) and the weighted inverse dynamics algorithm whole-body controller (Weighted WB).*

because of its higher inertia, and partially delayed with respect to the arm motion, due to its lower control loop frequency. This delay might affect the end-effector pose limiting the tracking performance. To limit this effect, we opted to weight the rotational motions of the base around the *z-axis*, favoring the combined motion of the arm and just the linear motion of the base. The effects of this choice can be noticed in Figure 3.35b, where the norm of tracking error results lower.

In an additional set of experiments, we evaluated the performance of the proposed

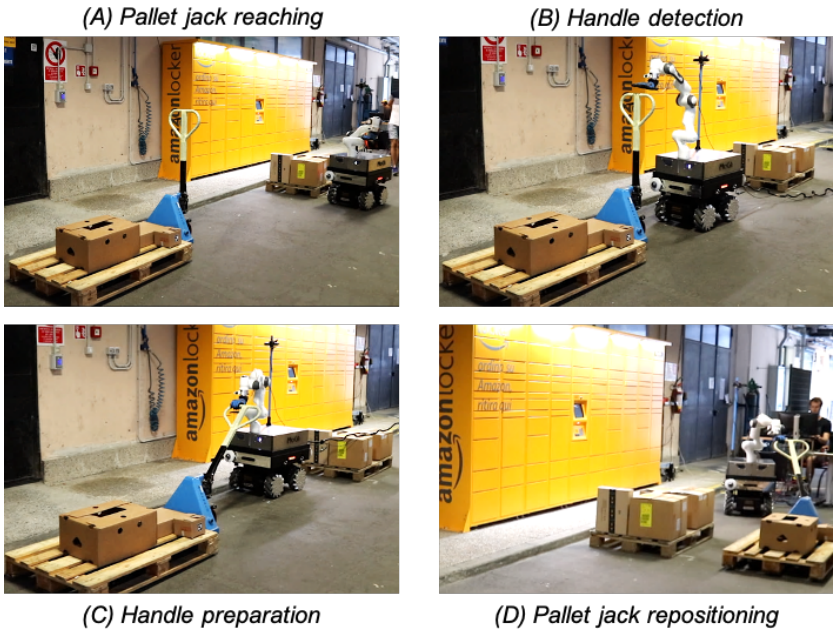


**Figure 3.36:** *Snapshots of the experiment which was performed in a real warehouse environment.*

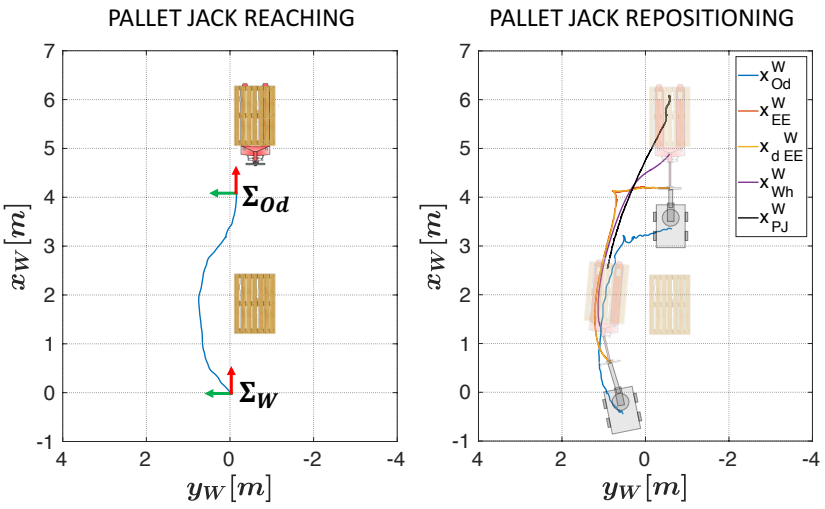


**Figure 3.37:** *Experimental results of phases A and D.*

framework (with Weighted WB impedance controller), in four phases of the task, whose snapshots are illustrated in Figure 3.36. A video of the experiment is available at [17].

In the *Pallet jack reaching* phase, MOCA had to reach the pallet jack avoiding obstacles (notice the filled pallet in Figure 3.36, which was placed in front of MOCA) after having received the target pose. In a more realistic scenario, the pallet jack could be localized within the warehouse in many ways, e.g. through a beacon system or exploiting cameras placed around the area. However, for the sake of simplicity in our proof-of-concept experiment, we assumed the pallet jack pose to be roughly known.

Figure 3.37 left plot depicts the path of the robot during this phase. Since a pallet (as an obstacle) was autonomously detected on the way to the goal, the algorithm updated the trajectories to avoid it. To detect the obstacles, the MOCA platform takes advantage of the perception sensors mounted on the Robotnik SUMMIT-XL STEEL mobile base, namely a HOKUYO-10LX Laser and an ASUS Xtion Pro RGB-D camera located in front.

In the next phase, the *Handle detection* unit retrieved the pose of the handle in the Odometry frame $\boldsymbol{\Sigma}_{Od}$, i.e. $\boldsymbol{p}_H^{Od} = [0.75\ \text{-}0.42\ 1.27]\ m$, as shown on the point cloud in Figure 3.38. The white point cloud corresponds to the real sight of the robot, the green one is the extracted handle cluster, and the red point cloud corresponds to the aligned template.

Subsequently, the robot end-effector reached $\hat{\boldsymbol{T}}_{ee}^{W}$ that can be noticed in the first column of the top plot in Figure 3.39. The impedance values were set to be very compliant, $300N/m$ in every Cartesian direction, to allow a safe and compliant interaction in case of unexpected collisions. Next, in the *Contact force* phase, the robot lowered its hand until a contact was detected, i.e., at time $t = 4.4s$, when $F_z$ goes beyond the threshold $F_{z,th}$ set to $7N$, as shown in the bottom subplot.

This contact triggered the *Handle preparation* phase, where the Pisa/IIT SoftHand grasped the pallet jack handle at time $t = 5 - 6s$ (*Grasping* subphase). The hand only uses 1 actuator to activate its adaptive synergy, so its intrinsic compliance makes it adapt to the object that is being grasped, therefore we set the closure percentage to 100%, to

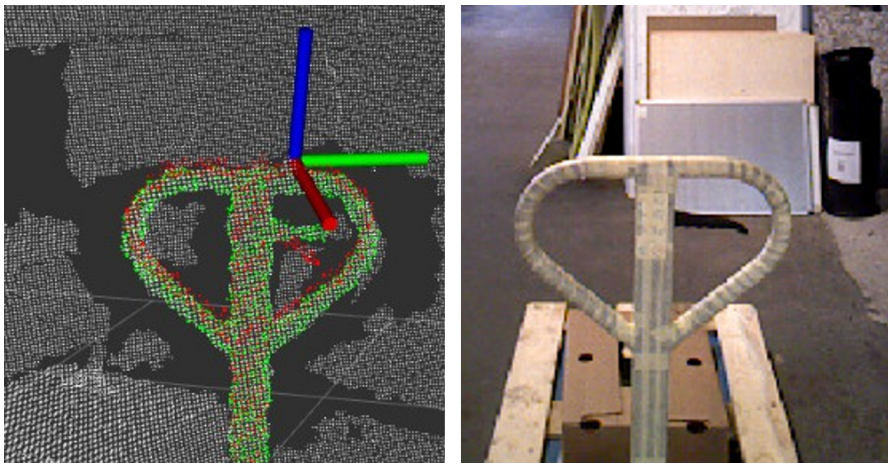

**Figure 3.38:** *Handle detection through perception module.*

ensure a stable grasping that guarantees the kinematic chain continuity between the robot and the pallet jack. Next, the circular trajectory described in Eq. 3.48 is executed with $\delta_p$ set to $\pi/4$ (*Pulling* subphase).

Here, the *interaction expectancy* value depends only on the task and the impedance adaptation value $\boldsymbol{k}_{st}$ and $\boldsymbol{k}_{min}$ in Eq. 3.15 were kept constant and set to $900N/m$ and $300N/m$ respectively. In the second plot of Figure 3.39, it is possible to notice the regulation of the impedance parameters based on the direction of the motion vector during *Pulling*. Here, the parameters of the weight matrix $\boldsymbol{H}$ were modified such

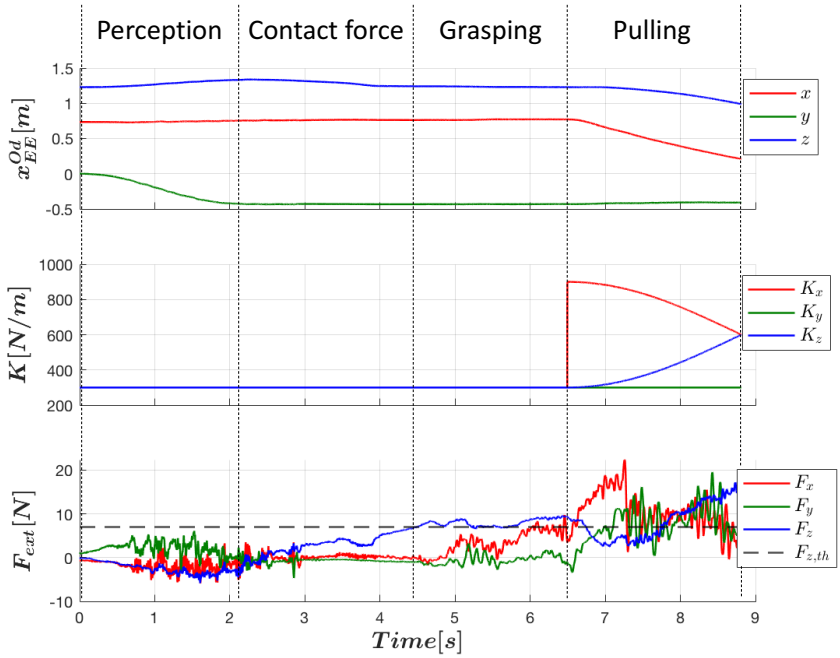


**Figure 3.39:** *Experimental results of phases B-C: MOCA end-effector pose, Cartesian stiffness and external forces.*

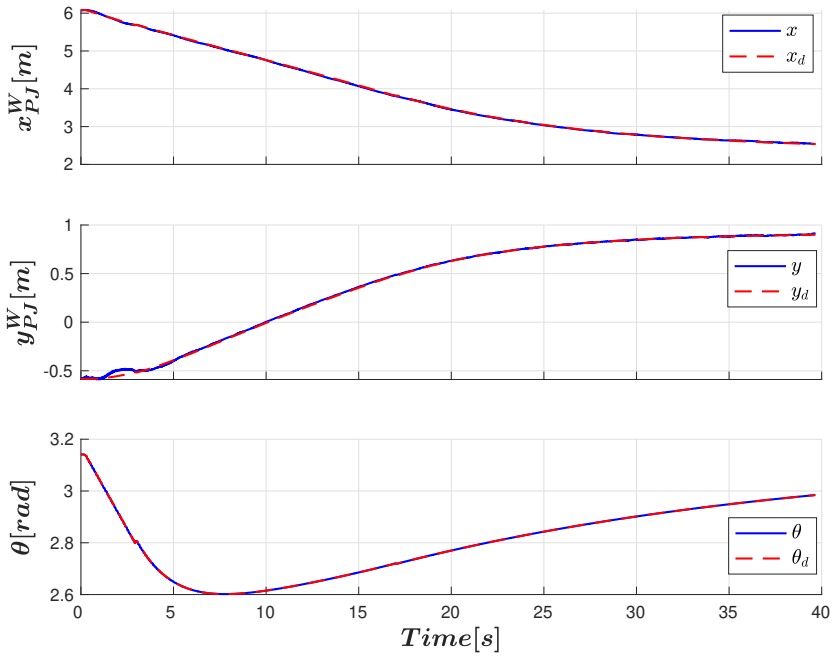


**Figure 3.40:** *Experimental results of phase D. The desired trajectories (red dashed lines) are accurately tracked through MOCA.*

that $\eta_A = \eta_{B_x} = \eta_{B_y} = 1$, and $\eta_{B_{yaw}} = \sqrt{20}$. The right plot in Figure 3.37 represents the results during the *Pallet jack repositioning* phase. Specifically, it shows the trajectories in the world frame $\boldsymbol{\Sigma}_W$ projected on the *xy-plane*. In this experiment, the goal position for the *Pallet jack repositioning* phase was set aside another filled pallet, which was positioned in front of a warehouse storage drawer, so it was set to $\boldsymbol{q}_{PJ}^{W} = [2.5\ m\ 0.9\ m\ \pi\ rad]^\top$.

Figure 3.40 shows the desired pallet jack configuration $\boldsymbol{q}_d = [x_d \quad y_d \quad \theta_d]^\top$ computed by the kinematically-constrained trajectory planner at each time-step, projected in the world frame $\boldsymbol{\Sigma}_W$. Moreover, it shows the estimated pallet jack configuration $\boldsymbol{q} = [x \quad y \quad \theta]^\top$ calculated reversing Eq. 3.50 and using the measured $\boldsymbol{T}_{EE}^{W}$.

CHAPTER 4

# Human-in-the-loop robot interaction planning and control

In this chapter, the second area of this thesis' contributions is introduced. We present new thinking and techniques dedicated to applications with high task complexity, where the robot autonomy alone can not satisfy the requirements to guarantee the task execution efficiency, both in terms of resources and the execution time (see also Figure 1.1). In such situations, the human-in-the-loop contribution is essential to achieve successful results. To this end, we introduce novel human-robot interfaces that have been developed, based on the human proximity levels with respect to the robot location. Figure 4.1 shows the different strategies that have been followed. In Section 4.1, a close-proximity approach is considered, where the human can plan the robot motion establishing a direct view contact with the robot. This can be done either physically guiding the robot in the

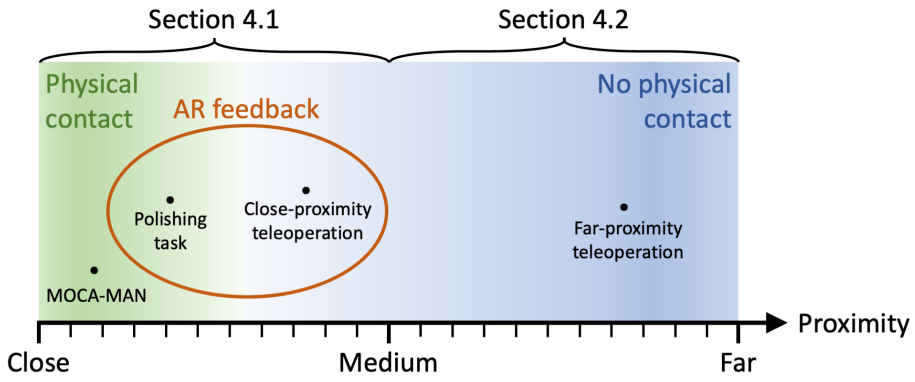


**Figure 4.1:** *Based on the level of human proximity, with respect to the robot location, different strategies will be presented in this chapter.*

workspace (MOCA-MAN), or commanding it through Augmented Reality interfaces that enhance the user awareness about the task, either with a physical contact (Polishing task), or without (Close-proximity teleoperation). Section 4.2, instead, presents novel interfaces that entrust the human with the decision-making process without involving any direct visual/physical contact. Within this strategy, we present a far-proximity teleoperation interface for remote robot loco-manipulation control.

The experiments presented in this Chapter were approved by the ethics committee Azienda Sanitaria Locale Genovese (ASL) N.3 (Protocollo IIT HRII 001 (rif. interno: 108/2018)).

## 4.1 A close-proximity approach to robot interaction planning and control

### 4.1.1 A MObile and reconfigurable Collaborative robot Assistant for conjoined huMAN-robot actions

In the previous chapter, we demonstrated the potential of cobots, and in particular of the MOCA platform, in carrying out tasks in an autonomous fashion, adapting to the surrounding environment. In this section, the aim is to prove that, in view of a higher task complexity, the same robotic platforms can be reconfigured in such a way they can respond to human-guided planning, temporarily interrupting their fully autonomous behavior.

The introduction of collaborative robotic technologies, embedded with rich actuation and sensing systems, opened a new horizon of automation opportunities for agile manufacturing, industrial scenarios, and even disaster response. This was due to their main distinctive features that promoted safety when working with their human counterparts [6], and enabled adaptation to their unknown surroundings [7, 42].

Collaborative technologies can take forms of exoskeletons [104, 137, 176], supernumerary limbs [147], and robotic manipulators (cobots) [42]. Supernumerary limbs, which can augment human capabilities via additional legs [148], dual-arms [41], and fingers [87], pose less mobility constraints to their wearers in comparison to exoskeletons. However, since the extra limbs are still carried by the users, they can cause fatigue. Cobots, on the other hand, have demonstrated a high potential for improving productivity [168] and contributing to better ergonomics and comfort of their human counterparts [106]. For the time being, the functional capacities of the supernumerary limbs (or exoskeletons) and cobots are perceived as complementary and synergistic, especially in industrial domains. However, this consideration adds costs, and requires a range of personnel expertise required for training and maintenance of the substantially different systems. An effective solution to these shortcomings, through lean thinking approach [138], would be a unified system that can reconfigure quickly between cobots and supernumerary limbs, to be able to cover a large range of industrial tasks' requirements.

To this end, the work presented here aims to exploit the reconfiguration potential of MOCA, to subsume the advantages of cobots and supernumerary limbs. In this way, the system can function autonomously (as already demonstrated in Chapter 3), or be physically coupled to the human counterpart as a supernumerary body, when necessary. The physical coupling is achieved through a mechanical admittance interface, that can en-

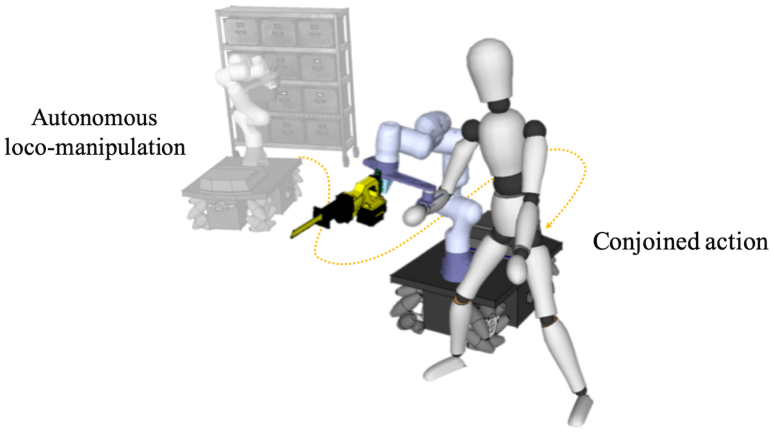


**Figure 4.2:** *The proposed control framework aims to incorporate the advantages of cobots and supernumerary limbs into a unified collaborative framework, to cover a large range of industrial task requirements.*

able the human desired pose inputs to the robot whole-body controller. Through a hand gesture recognition system, the controller can give higher priority to the mobile base (e.g., for long distance co-carrying tasks) or the arm movements (e.g., for manipulating tasks), when performing conjoined actions. Meanwhile, the admittance interface translates the user forces into end-effector trajectories in space, so as to enable the execution of joint actions.

As a result, the user does not have to carry the weight of the supernumerary limb, and can connect to it only when he/she feels the need. In all the other cases, MOCA can reconfigure to a mobile collaborative robot and perform its routine work. This, besides contributing to the user's comfort, also increases the productivity, since it helps to reduce equipment waiting and setup times, typical of small and medium enterprises [9].

We named this system MOCA-MAN, that stands for "MObile and reconfigurable Collaborative robot Assistant for conjoined huMAN-robot actions". Hereafter we present the human-robot admittance interface that was implemented, the control framework that allows the switching from autonomous behavior and MOCA-MAN conjoined actions, and the evaluation of the framework performances in two different use-cases: one that requires large mobility and one for close-proximity manipulation. The results reported in this section led to the scientific publication [103].

#### Human-robot admittance interface

In order to enable a physical coupling between the human and the robot for conjoined actions, we developed a human-robot admittance interface. Hereafter, we provide the details about the interface hardware implementation, and the ones regarding the admittance model that was developed at the control level.

**Hardware** Figure 4.3 illustrates the human-robot admittance interface. From the robot side, it is designed as an extension link connected to the robot end-effector flange, while from the human side is attached to the subject wrist. The mechanical connection between the two sides is achieved through a magnetic clamp that enables rapid and straightforward attachment/detachment. When the two sides are mechanically con-

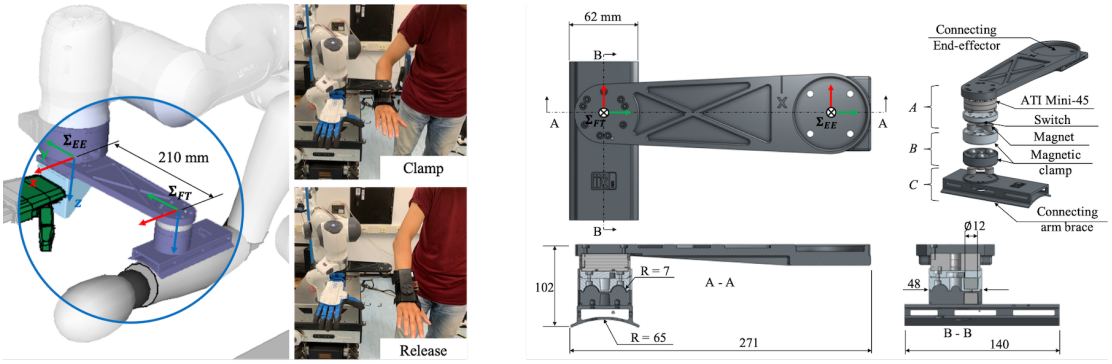


**Figure 4.3:** *The components of the developed human-robot admittance interface: A-robot mounting part, B-magnetic clamp part and C-wearable part. The admittance interface can be attached/detached through a magnetic clamp.*

nected, the interaction forces applied by the human are transmitted to the robot, so as to perform conjoined human-robot movements. The magnetic clamp, besides guaranteeing a fast reconfigurability, ensures a high safety level for the human, in the case unexpected robot behaviors occur.

The admittance interface consists of three parts, depicted in Figure 4.3b: the robot mounting part ($A$), the wearable part ($C$), and the magnetic clamp part ($B$). The device dimension is $271 \times 140 \times 102mm$, and its weight is $0.47kg$, including the arm brace and the Force/Torque (FT) sensor employed to measure human interaction forces.

The robot mounting part is attached to the robot end-effector flange, and is placed between the end-effector and the wrist flange. The flange is designed so as not to constrain robot end-effector movements when the robot operates unaccompanied. This part is composed of a six-axis ATI Mini-45 FT sensor and a push switch. The FT sensor measures the human-robot interaction forces. In this way, such forces can be distinguished from the interaction forces that arise from the robot interaction with the external environment. The human-intended forces are then transmitted through the admittance interface, so as to obtain the desired trajectories for the conjoined motion through the admittance control algorithm, whose details are explained in the next subsection (*Admittance behavior for conjoined motion*). The FT sensor is displaced from the end-effector frame $\Sigma_{EE}$ by $210mm$ along the negative *y-axis*. The push switch is placed in the last layer of the mounting part, to detect whether the wearable part is clamped (or not) through the magnetic clamp. This facilitates a fast switching between the autonomous and the conjoined human-robot mode, and also ensures human safety when the wearable part is released from the robot mounting part. Notice that, although the conjoined mode is terminated as soon as a user detaches his/her arm from the robot, a time interval of $3$ seconds is considered to switch the control mode. This is done to let the subject move away from the robot in a safe way.

The wearable part is assembled with a comfortable arm brace, that the user can wear on the hand. To ensure that the interaction forces are directly transmitted to the FT sensor, the wearable part and human interface are connected tightly through three Velcro loop bands. The shape of the bottom surface is designed to adapt to the human body arm and wrist with a curved surface. Additionally, to enable intuitive control of

the robot end-effector, the attachment parts are designed to align the human hand and the robot hand during the conjoined task executions.

The clamping force can vary based on the magnetic field's strength, that can be decided based on the threshold of interaction forces that are considered to be safe. The magnetic interface includes three disk magnets that possess a diameter of $12mm$, and a height of $10mm$. When two unequal magnetic poles attract each other at a distance of $4mm$, the attractive force is approximately $12.18N$. Accordingly, interaction forces can be transferred up to $36.54N$ via the interface. Hence, when the interaction force coming from the human overcomes this threshold, the magnetic clamp is released and the robot control is stopped.

**Admittance behavior for conjoined motion** The human-robot admittance interface is developed to enable conjoined movements when performing manipulation tasks or carrying loads. For this reason, we exploit an admittance control loop to transfer contact wrenches to desired end-effector movements, through an admittance model [39, 123]. The whole-body behavior of the robot when following the trajectories can be regulated based on the task requirements and the mobility considerations, as explained in Section 3.3.4.

Let $\mathbf{F}_m = [\mathbf{f}_m^\top \quad \boldsymbol{\mu}_m^\top]^\top \in \mathbb{R}^6$ denote the contact wrenches measured from the external FT sensor (mounted on the admittance interface), and $\boldsymbol{\vartheta}_d = \begin{bmatrix}\boldsymbol{\upsilon}_d^\top & \boldsymbol{\omega}_d^\top\end{bmatrix}^\top \in \mathbb{R}^6$, the spatial desired velocity vector, where $\boldsymbol{\upsilon}_d$ and $\boldsymbol{\omega}_d$ correspond to linear and angular velocities, respectively. The dynamic relationship of the admittance model is given by:

$$\boldsymbol{\Lambda}_d \dot{\boldsymbol{\vartheta}}_d + \mathbf{D}_d \boldsymbol{\vartheta}_d = \mathbf{F}_m, \tag{4.1}$$

where $\boldsymbol{\Lambda}_d$ and $\mathbf{D}_d$ are $6 \times 6$ positive definite diagonal inertia and damping matrices that can be tuned to obtain a gentle robot behavior. Through Eq. 4.1, that describes a decoupled dynamic behavior, the desired velocity vector $\boldsymbol{\vartheta}_d$ is transformed into the incremental motion of desired task $\mathbf{x}_d$ via a discrete-time integration. Noteworthy, since the installed FT sensor's frame $\Sigma_{FT}$ is displaced from the $\Sigma_{EE}$, the measured wrenches are transformed into $\Sigma_{EE}$ to obtain the interaction forces, which is then used for the admittance control law.

**MOCA-MAN control framework**

Figure 4.4 illustrates the unified MOCA-MAN control framework. The control flow consists of two modes: the autonomous mode and the conjoined mode.

The autonomous mode represent actions as manipulation, navigation, etc., which can be autonomously executed by the robot. As a representative example of the autonomous mode, the robot approaches the human partner, when it is called for a conjoined action using a hand gesture. For this purpose, the position of the human partner is sent to the robot whole-body impedance controller.

On the other hand, the conjoined mode enables the execution of coupled MOCA-huMAN actions. Examples include co-manipulation (e.g., drilling with heavy or vibrating tools), or co-carrying of heavy or potentially dangerous items. In this mode, the *Prioritized weighted whole-body inverse dynamics control*, included in the Appendix (Section A.1), can set a higher priority to the arm or mobile base movements. For example, if the task requires precise manipulation at a certain position, the controller is able

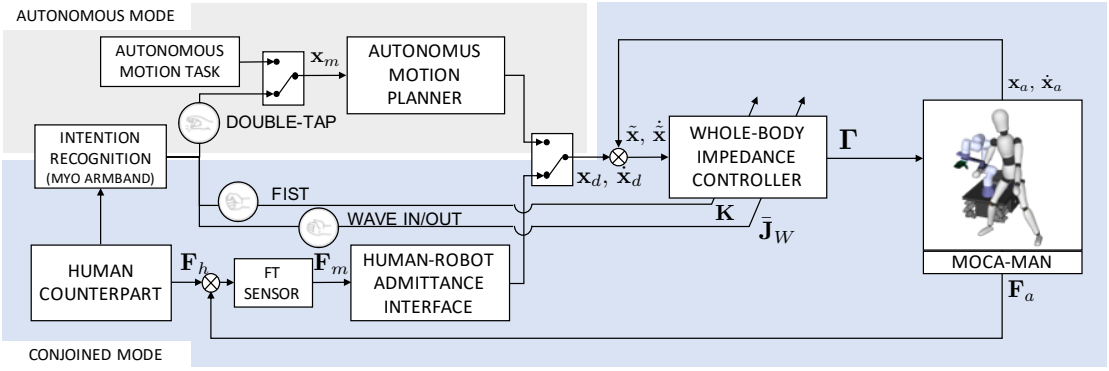


**Figure 4.4:** *The overall control architecture of the presented collaborative framework split in the two control modes: the gray part represents the autonomous mode, while the light blue one the conjoined mode.*

to give a higher priority to the motion of the robotic arm. On the contrary, if the task requires long distance transportation, the mobile base provide a higher contribution, with respect to the arm, in the task execution.

To switch between the two modes, we use a Myo armband (Thalmic Labs$^{TM}$), which integrates eight electromyography sensors, that can provide the recognition of human gestures. The switching between the autonomous and the conjoined mode is triggered by a *double tap* hand gesture, which makes the robot approach the user. Then, when the subject realizes the clamping motion, the conjoined mode is activated. Additional hand gestures such as *wave in/out* and *fist* are considered to assign higher mobility to the arm/mobile base movements, and also to open or close the Pisa/IIT SoftHand, respectively.

To compensate the unknown weight of the external object held by the robot end-effector, in the conjoined mode, the desired stiffness of Cartesian impedance controller is set to a higher value along the *z-axis*. This feature is activated when the *fist* gesture is detected, which implies that an object is about to be grasped and picked by the SoftHand.

#### Experimental results

In this section, we first provide experimental validation of the framework's switching capability from autonomous to conjoined mode, where MOCA approaches to its calling partner to perform a joint action. We consider two experimental tasks, i.e., long distance carrying and close-proximity co-manipulation. Next, we validate the effectiveness of the proposed collaborative framework in conjoined mode through the electromyography (EMG) analysis, where we demonstrate a coherent reduction of five subject's physical efforts while performing a manipulation task.

**Setup** During the experiments, the human subjects wore the Myo armband, as well as the wearable part of the developed admittance interface. Instead, the mounting part was attached to MOCA. The whole-body impedance controller was executed at $1KHz$ frequency, and communicated with the gesture recognition module at $10Hz$ via ROS.

For the first experiment, MOCA was located $0.6m$ far from its human partner, where

the target object (a carrying case, $3.1kg$, $440mm \times 340mm \times 140mm$) and the final goal position were located at $0.8m$ and $2.8m$ distance from the starting position of MOCA, respectively.

To call MOCA and make it autonomously navigate towards the human partner to perform a conjoined action, the subject made a *double tap* gesture. To determine the position of the user, we used an external Optitrack motion tracking system with a single marker set, that was mounted on the wearable part of the admittance interface. The task of conjoined motion was to carry an object to a final goal position, known only by the human. The weighted Jacobian (see Section A.1) and the Cartesian impedance parameters were switched when the human partner realized the *wave in/out* gestures, so as to give high preeminence to the arm movements when reaching to grasp/release the object, and to the mobile base when carrying it. When the given task was accomplished, the human partner released the admittance interface to end the conjoined motion.

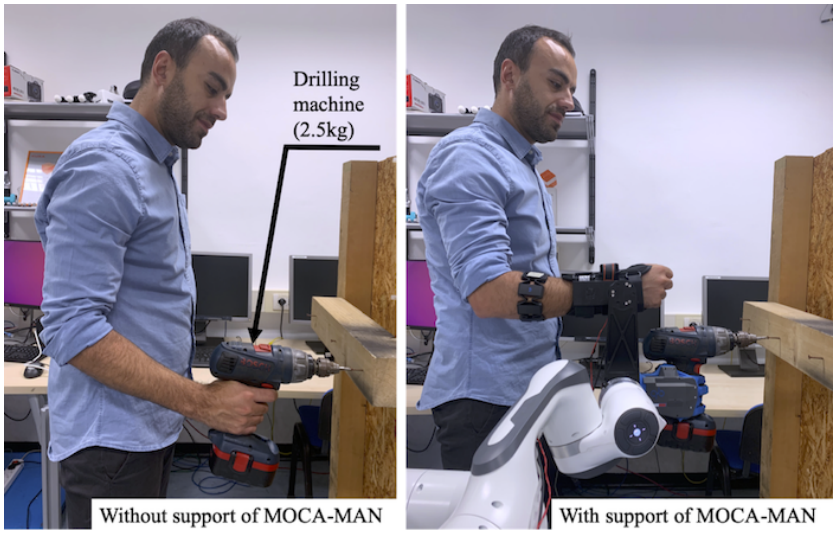


**Figure 4.5:** *The experimental setup for the manipulation task, carried out to compare the effort with and without the support of MOCA-MAN.*

In a complementary co-manipulation experiment, we compared the human muscle activities of the upper limb, measured by surface EMG sensors (sEMG), with and without the support of MOCA-MAN. Five volunteers (1 female and 4 males) participated to this experimental session. The task was to perform a screwing operation at $1m$ height with a drilling machine ($2.5kg$). The task was repeated in two conditions: i) in the conjoined human-MOCA mode where the drill was held and operated by MOCA-MAN, and ii) with the subjects' own hands and without MOCA's support (see also Figure 4.5). Six sEMG sensors were placed on the arm of each subject, on the anterior deltoid (AD), the posterior deltoid (PD), the biceps (BC), the triceps (TC), the flexor carpi radialis (FC), the extensor carpi radialis (EC) muscles. The objective was to demonstrate a coherent reduction of the effort across all subjects when MOCA was operated in conjoined mode and assisted its human partner in the execution of the task.

**Results** The results of the co-carrying experiment are illustrated in Figure 4.6. The sequences of the task execution, which are distinguished by the hand gestures, are

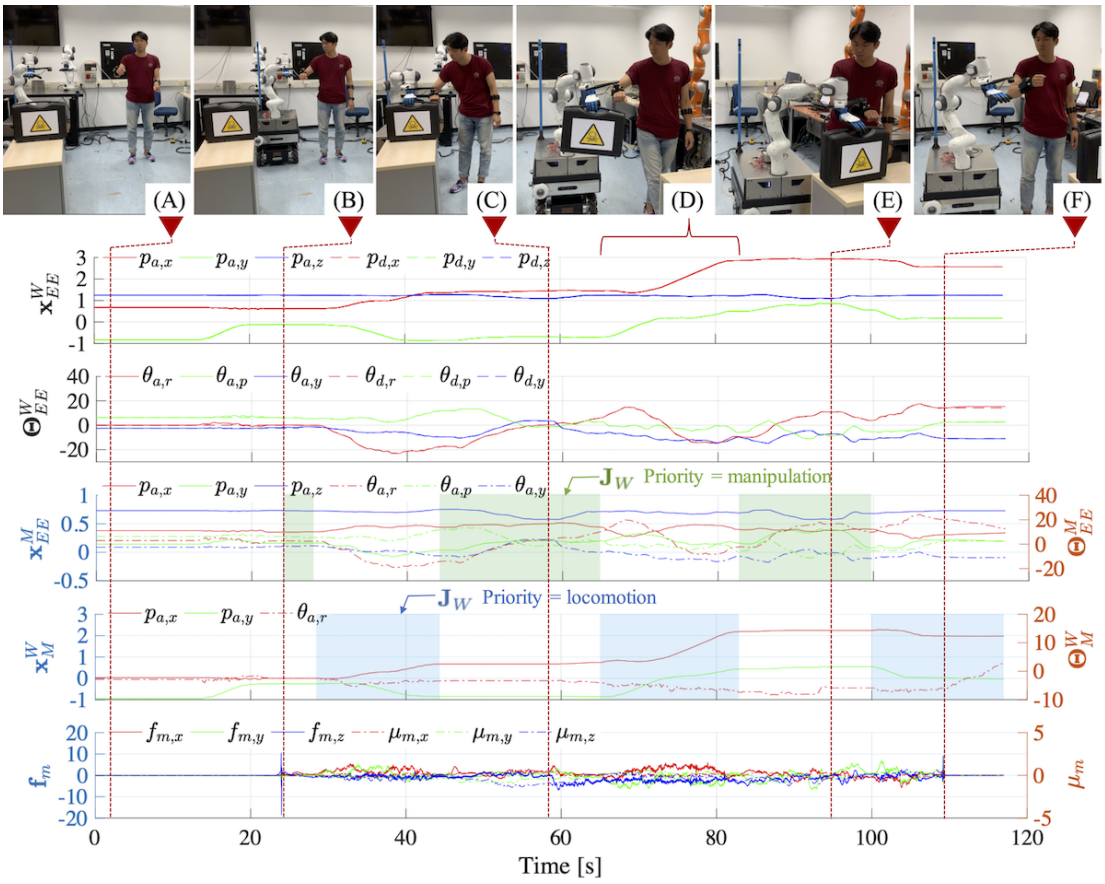


**Figure 4.6:** *Experimental results of the proposed collaborative framework during the co-carrying task: snapshots of the experiment (top), the trajectory of the whole-body motion during the execution of the conjoined action (first two plots), trajectories of the robotic arm (third plot) and the mobile base (fourth plot), the interaction forces sensed through the admittance interface (bottom plot). The dashed lines represent the desired motion, and the solid ones represent the measured motion.*

represented above the plots. The first two plots represent the position and orientation of $\Sigma_{EE}$ with respect to $\Sigma_W$, the third and fourth plots depict the position of $\Sigma_{EE}$ with respect to $\Sigma_R$ and position of $\Sigma_M$ with respect to $\Sigma_W$, to show the prioritized arm and mobile base movements due to different priority set in Eq. A.10. The bottom plot shows the measured forces at the FT sensor, that was placed between MOCA and the human in the admittance interface, so as to detect human intended movements.

In the sub-phase (A), MOCA started to move towards the human partner's position when the *double tap* gesture was recognized. During this phase, MOCA used the mobile base motion to approach its human partner. Subsequently, the conjoined action started, as soon as the admittance interface switch was enabled due to the clamping action (sub-phase (B)). In the conjoined action mode, the human partner guided MOCA through the environment, applying interaction forces that were sent to the admittance model, as shown in the bottom plot. The desired motion was obtained by Eq. 4.1, where:

$$\mathbf{\Lambda}_d = \text{diag}\begin{pmatrix} 3 & 3 & 3 & 0.1 & 0.1 & 0.1 \end{pmatrix} \left[\text{kg}, \text{kgm}^2\right],$$

$$\mathbf{D}_d = \text{diag}\begin{pmatrix}20 & 20 & 20 & 0.5 & 0.5 & 0.5\end{pmatrix} [\text{Ns/m, Nms/rad}] .$$

When MOCA reached the vicinity of the object targeted by the user, the priority motion was given to the manipulator, through the *wave-in* gesture. This was done to approach and grasp precisely the object. To carry the object, the robot had to compensate the imposed gravitational forces, caused by the object weight. As mentioned previously, this gravity-compensation feature was activated when the SoftHand performed the grasping action, which was triggered by the *fist* gesture (sub-phase (C)). Hence, the parameter of the Cartesian stiffness matrix $\boldsymbol{K}$ along the *z-axis* increased automatically from $300N/m$ to $1800N/m$. Although MOCA, as a supernumerary body, was physically coupled to human, the effect of the object weight was not transmitted to the human partner, who instead perceived the physical support. The payload assistance provided by MOCA was also evident in the carrying phase, where the mobile base had a higher priority during the conjoined movement (sub-phase (D)). In fact, in this phase the measured interaction forces along the *z-axis* kept an almost constant value of $2.71 \pm 0.97N$, until the release of the object. The SoftHand ungrasping was triggered by the *fist* hand gesture (sub-phase (E)). In the sub-phase (F), the human partner detached the admittance interface to terminate the conjoined mode. The detachment force $F_{safe} = 39.06 \pm 2.39N$, and moment $T_{safe} = 2.1 \pm 0.1Nm$, values were consistent with the safety considerations we took into account when designing the magnetic interface. A video of the experiment is available at [103].

Table 4.1 shows that the overall sEMG data collected during the manipulation task. These data confirmed the reduction of the muscles' activity when MOCA-MAN was used for the screwing task. The decrement ratios across the five subjects were: $49.42 \pm 4.23\%$ in the AD, $22.97 \pm 4.23\%$ in the PD, $67.96 \pm 2.13\%$ in the BC, $48.10 \pm 3.69\%$ in the TC, $76.04 \pm 1.99\%$ in the FC, and $80.49 \pm 1.82\%$ in the EC.

**Table 4.1:** *Experimental results of five subjects performing a screwing task. The decrement ratio (%) of the muscle activity is reported by: mean (standard error of the mean).*

| | AD | PD | BC | TC | FC | EC |
|---|---|---|---|---|---|---|
| Subject 1 | 69.87 | 46.79 | 59.29 | 76.25 | 86.01 | 93.74 |
| | (2.10) | (1.34) | (4.49) | (0.59) | (0.11) | (0.39) |
| Subject 2 | 34.19 | 50.57 | 76.18 | 54.19 | 76.03 | 87.93 |
| | (2.34) | (2.56) | (2.71) | (1.03) | (5.02) | (1.93) |
| Subject 3 | 77.13 | 8.61 | 82.39 | 47.23 | 74.03 | 78.34 |
| | (7.56) | (11.88) | (0.36) | (0.24) | (3.27) | (0.34) |
| Subject 4 | 21.12 | 1.71 | 53.25 | 43.91 | 85.52 | 73.15 |
| | (6.89) | (6.66) | (4.12) | (1.22) | (0.16) | (0.57) |
| Subject 5 | 44.77 | 7.20 | 68.69 | 18.91 | 58.63 | 69.29 |
| | (0.82) | (7.50) | (0.88) | (3.30) | (6.19) | (1.84) |
| Mean | 49.42 | 22.97 | 67.96 | 48.10 | 76.04 | 80.49 |
| (std. error) | (4.23) | (4.23) | (2.13) | (3.69) | (1.99) | (1.82) |

The statistical differences between the two conditions (i.e. task with and without MOCA) were tested with post-hoc t-tests, where the level of statistical significance was $.05$. The measured muscle activity of the AD, BC, TC, FC, and EC showed a significant difference between performing the task with MOCA and without it. The differences of those muscles were: $36.4 \pm 4.57\% (p \leq .05)$ in the AD, $42.6 \pm 2.41\% (p \leq .05)$

in the BC, $33.0 \pm 5.31\% \, (p \leq .05)$ in the TC, $32.8 \pm 4.01\% \, (p \leq .05)$ in the FC and $46.6 \pm 3.40\% \, (p \leq .05)$ in the EC. On the other hand, the difference of muscle activity in the PD was statistically insignificant, being $12.2 \pm 4.16\% \, (p = .13)$. The muscle activity of the PD slightly reduced, but not as much as in the other muscles. Nevertheless, we can conclude that the overall effort among subjects decreased significantly with the support of MOCA-MAN interface. Future works will investigate the ergonomic aspects of this interface, i.e. what is its impact on the human posture and how human locomotion is affected. Moreover a usability analysis will be conducted, evaluating the interface user experience.

### 4.1.2 Human awareness enhancement through Augmented Reality

As we have also seen in the previous section, in the human-robot collaboration (HRC) field, an open challenge is to create a simple and intuitive system in which humans and cobots could interact in the most natural way. In human working teams, people communicate their intentions, besides spoken language, through gestures, facial expressions and, in general, with body motions. The absence of these rich and immediate communication means in human-robot teams decreases the trust of the human on the robot and the perception of robot user-friendliness, even if the environment is safe for the human. To face the challenge of a simple interface that allows human and robot to interact and coordinate, several devices have been developed, for instance, wearable and haptic devices [6].

A more recent trend in HRC consists in the introduction of Augmented Reality (AR) applications in industrial working environments. AR combines real-world and computer-generated data to enhance the user experience, differently from a purely virtual environment, in which the user is totally immersed. Several studies already showed that AR enhances HRC performances, in particular, improving task efficiency by helping workers to understand robot intent [71]. Furthermore, AR applications proved reduction of costs up to $25\%$, and an improvement in performance up to $30\%$, in the manufacturing industry [179].

Other applications of AR in industrial scenarios have been presented. For example, Boeing workers exploited Google Glasses to help workers in the construction of aircraft wire harnesses [28]. To overcome the discomfort of the platform, wearable devices that projects information on surfaces have been presented, as a system based on a projector that supports employees in assembly tasks by virtual information and instructions [65]. To provide additional information about the workplace, a head-up display integrated into an industrial helmet has also been presented [163]. With the same goal, another work introduced an Android-based application for programming and monitoring industrial robots using a tablet [136]. Even if the tablet presents a simple interface to communicate with the robot, workers cannot work and interact with the system at the same time. In fact, in this application, a worker holds the tablet and another performs the collaborative task. In general, hand-free devices are more suitable for physical tasks in industrial environments.

In this section, we propose two simple and intuitive AR interfaces for human workers in collaborative tasks. The main purpose is to improve the user experience, increasing worker confidence in the system and his awareness of the robot actions during collaborative tasks. We refer in particular to situations in which human and robot do not

only coexist in the same space, but also interact between themselves and the environment. We will exploit the mixed reality smartglasses (Microsoft HoloLens), a device that allows not only the projection of holograms in a head-mounted display (HMD) but also presents new features like voice commands, as well as gestures recognition and gaze tracking, opening new prospects on the way human can interact and control robots. The results reported in this section led to the scientific publications [45] and [16].

#### Augmented Reality interface for Human-Robobt collaborative tasks

To obtain the desired human-cobot collaboration, we would like to exploit human cognitive skills with cobot's physical power generation capacity. To do that, a real-time communication between the two agents is needed [156]. The AR interface, indeed, should be able to receive feedback from the smart work cell and eventually elaborate these data to obtain meaningful information for the human. In this way, workers will be more aware of the work cell's current status and future plans. On the other hand, human commands could improve both task scheduling, by creating an intuitive mean of task scheduling, and task performance by coordinating the two systems.

The AR device, i.e. Microsoft HoloLens, allows the human user to send inputs to the robotic system in different ways: vocal input, hand gesture input, and gaze. At the same time, the user could receive system information in the form of holograms (visual feedback) and sound (audio feedback). Although non-verbal interaction has already been introduced in HRC, as in the case of haptic cues [50], other devices are not able to provide a wide range of input/output data, streaming simultaneously audio and visual feedback while allowing the human user to interact in different ways.

The first goal of the presented AR interface is to help workers to interact in the most user-friendly way possible with the cobot. In this way, we tried to identify the type of data that can be the useful for the task execution, which should be streamed to the worker, either using the visual or audio feedback, and sent if requested. Another important aspect of the interface considers the overload of information. We want to provide the user with the smallest amount of information needed to achieve the task, not to stress or mentally overload them. Moreover, this additional data should not conflict with other environmental aspects of the real scene (e.g., occluding the workspace), to make the AR experience as less invasive as possible. The AR interface has to provide instant feedback from the robotic system to the user, from high level information like the schedule of the task plan, to lower-level system status data, in terms of Cartesian positions and velocity, force and torques applied to/from the end-effector. These data are not just useful for safety reasons to improve workers awareness in the execution of the task, but also might be required for a correct task execution.

On the other hand, the user inputs can be used to coordinate cobot performances with the one of the workers. For instance, in order to give the cognitive lead of the task execution, we will allow the human worker to trigger the system status progress. After the performance of certain actions, the system will switch to a break state, waiting for the worker's input, that might be either a voice command or a gesture, according to the particular environment or task. The architecture of the proposed coordination system is depicted in Figure 4.7.

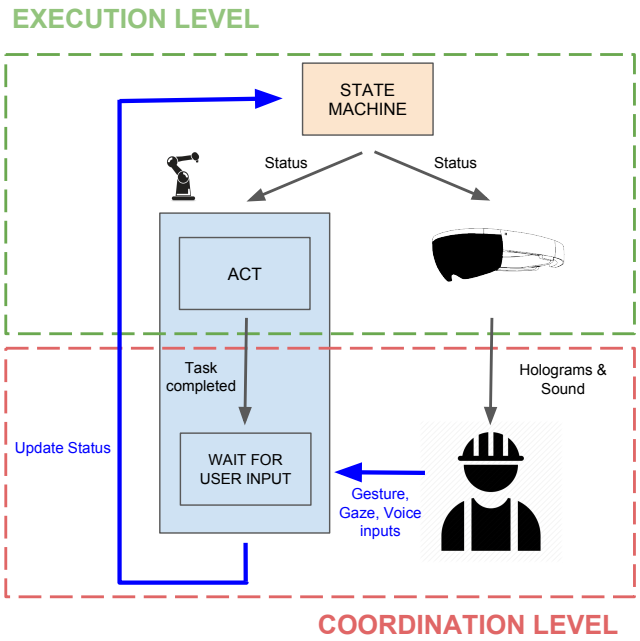


**Figure 4.7:** *At the Execution level, a state machine sends the status information to the HoloLens and the robot. Worker wearing HoloLens receives AR feedback (Holograms and Sounds) while the robot accomplishes the task. At the coordination level, robot switches in a break state, waiting for the workers' input. Human operator can trigger the system through a gesture, gaze or vocal command.*

### System architecture

In this section, we will introduce the collaborative framework that we exploited in the experiments. Figure 4.8 shows the two main components of the system: the AR device Microsoft HoloLens and a collaborative robot. The HoloLens is a fully standalone HMD device that provides immersive AR. The device runs on Windows 10 and features a Holographic Processing Unit and a 32-bit Intel CPU. The sensory system contained in the device hardware provides excellent stability of the 3D holograms in the real world. With Microsoft HoloLens, it is possible to place 3D-holograms into space through the

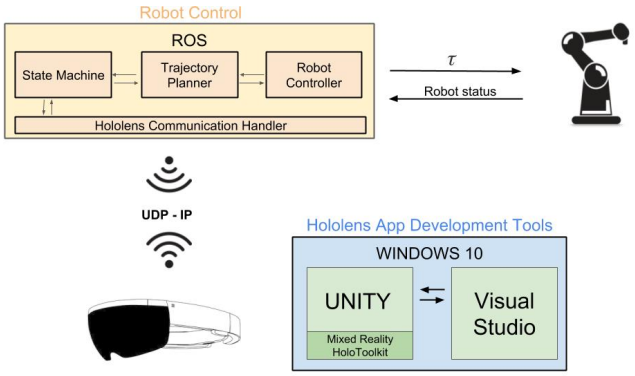


**Figure 4.8:** *The system architecture enables double channel HoloLens-Robot communication via WiFi (UDP-IP). Hololens App Development Tools represent the environment in which a holographic 3D application is developed. The main components involved in the robot control are shown above.*

environmental scanning. Moreover, the integrated gesture and speech recognition allows for a real-time interaction with the holograms. The gaze direction, used to place holograms in the environment, is estimated using the orientation and position of the user's head. Finally, the vocal input is possible thanks to four microphones embedded in the device, and through a speech recognition algorithm.

The system architecture allows double channel communication, from the robot to the HoloLens and viceversa, through User Datagram Protocol (UDP). UDP provides a real-time data transmission between the devices via WiFi. The robot control architecture is implemented using ROS. Microsoft provides different software to develop applications for Universal Windows Platforms. One of these is the Unity development software: a commercial cross-platform game creation system for creating interactive media. The Unity platform provides many tools for holographic 3D development and simulation. In our case, it is used to create the AR scene in which the user is immersed. Moreover, we exploited Microsoft Mixed Reality Toolkit, a Unity library containing scripts, that allows the HoloLens user to interact with holograms. To develop and build `C#` scripts, Visual Studio was used. The result is a holographic 3D application that can be executed on the HoloLens.

#### Experimental results

In the next paragraphs, we demonstrate how the potential of this framework can be exploited in two different application scenarios, that involve a direct physical contact with the environment (collaborative polishing), and the planning of robot actions from the vicinity of the robot workspace without any direct physical interaction (close-proximity teleoperation).

**Collaborative polishing** In this experiment the worker should perform the polishing of a rough wooden surface with a polisher, following a linear trajectory. The polisher is mounted on the end-effector of the robot through the Pisa/IIT SoftHand. The weight of the polisher is completely sustained by the robot thanks to the Cartesian impedance control with high impedance on the robot *z-axis*, but the operator can easily move the robot in the other 2 directions since the robot assumes a very compliant behavior. The polishing task was performed by 10 subjects, between 25 and 35 years old, who were not aware of the purpose of the experiment. Subjects were required to keep the force applied perpendicular to the surface (on the *x-axis*) as close as possible to $10N$, for 30 seconds. The same experiment was performed in three different experimental settings in the following order: without any force feedback to the human operator, with a force feedback displayed on a screen, and with the force feedback projected on the HoloLens display. Please note that, in this Section, the force feedback is merely visual.

The latter interface was illustrated to the user with two different holograms: one with a number representing the interaction force, expressed in Newtons, and another with an arrow that changes color (green if the applied force is close to the desired force, red if really far from the desired force, and yellow in between the two) and dimension according to the force value (the higher the force, the longer the arrow). Figure 4.9 shows an example. The visual force feedback corresponds to the estimated force applied to the end-effector of the robot. A video of the experiment is available at [45].

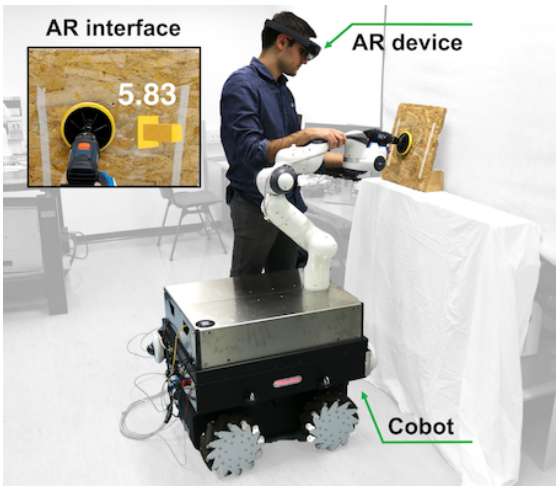


**Figure 4.9:** *Human-Robot collaborative task supported by an Augmented Reality interface. The human operator receives instant feedback about the applied force.*

Participants were asked, after the execution of each trial, to complete a questionnaire based on the Likert scale. The questionnaire was as follows:
Q.1 The polishing was easy to perform;
Q.2 It was difficult to maintain the desired force;
Q.3 It was easy to perceive the amount of force applied;
Q.4 It was difficult to follow the desired trajectory;
Q.5 It was easy to keep the focus on the task execution;
Q.6 It was physically tiresome to accomplish the task;
Q.7 It was psychologically tiresome to accomplish the task;

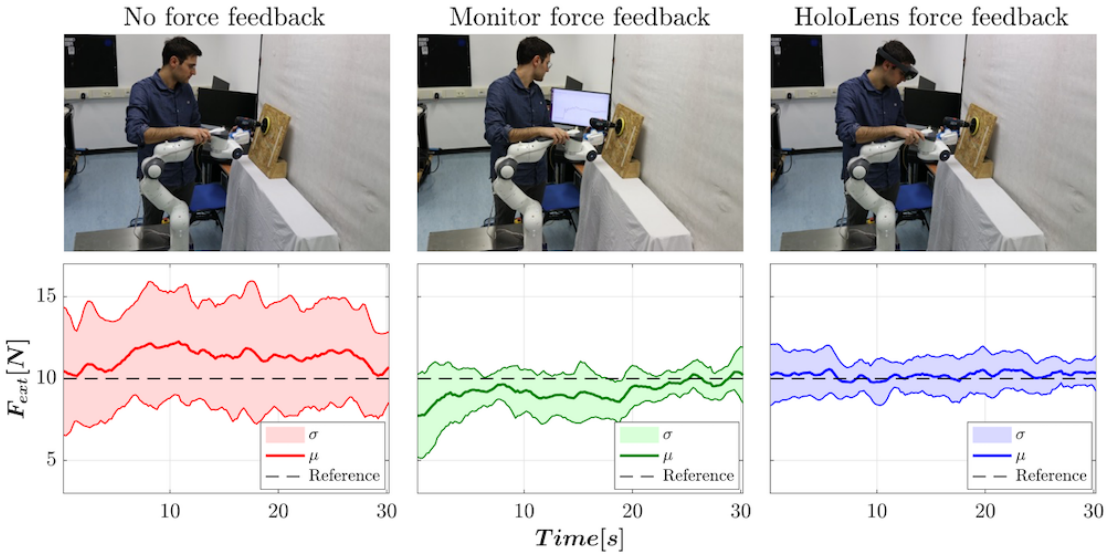


**Figure 4.10:** *Experimental collaborative polishing task. The photo above show the experimental setup in the three different experimental settings specified by the title. The task force in the three experimental settings is depicted in the plots below. The average value of the force ($\mu$ ) is shown by the marked line, the standard deviation ($\sigma$) by the faded area, and the reference value ($10N$) by the dotted line.*

Q.8 Overall, I felt satisfied with the current task performance.

A score is associated with each possible answer, from strongly disagree to strongly agree, with an assigned score of -5 and +5, respectively.

To compare the measured task force between the three different experimental settings, we computed the average and the standard deviation among the ten subjects throughout the experiment for each trial: no force feedback (trial 1), monitor force feedback (trial 2), and HoloLens force feedback (trial 3). The results shown in Figure 4.10, suggest that the subjects performed more accurately the polishing task with a feedback interface. In fact, in trials 2 and 3 the standard deviation of the task force is lower with respect to trial 1. However, the average value in trial 2 is still oscillating and it is not centered on the desired value set to $10N$. Using the AR interface (trial 3), instead, allows the users to keep an effort almost constant, hence we can conclude that the AR interface provides a better feedback than the monitor. To further support this hypothesis, we performed a one-way analysis of variance (ANOVA test). The boxplots of each trial are displayed in Figure 4.11. It resulted that the differences between the data in trial 2 and 3 are statistically significant (p-value $< 0.01$).

These improvements are confirmed by the questionnaire results shown in Figure 4.12. In the answers to the question 5 (Q.5) the subjects stated that it was easy to keep the focus on the task execution in the third trial, unlike the second trial. The second trial is the most physically tiring for the subjects (Q.6) and it is the least easy to perform (Q.1), but it introduced improvements in the perception of the applied force compared to the first trial (Q.3). However, the subjects confirmed that with HoloLens the perception of the force applied is improved even more. Also with regard to the trajectory followed during the polishing task, the subjects preferred the AR feedback with respect to the other approaches (Q.4). In particular, the trial with force feedback displayed on the monitor was the most difficult to accomplish, since the subjects could not focus on the task because of the monitor placement. Overall, the participants felt satisfied with the proposed AR interface (Q.8) and the amount of psychological effort required (Q.7). An

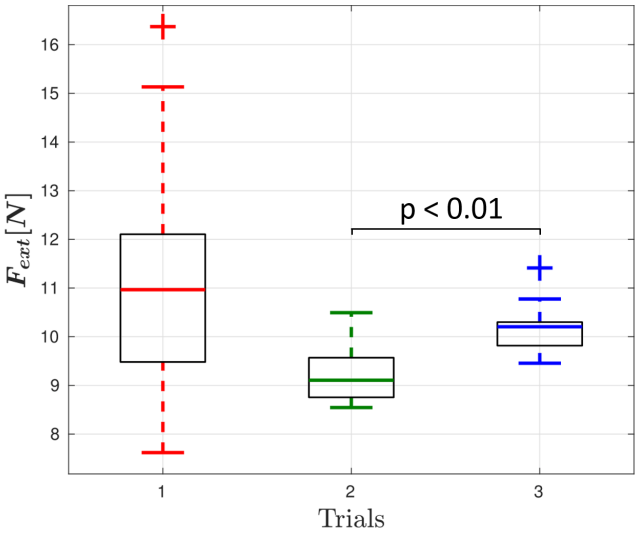


**Figure 4.11:** *Boxplots of the task force average value in the three experimental settings (trials). The p-value shows that Trial 3 is statistically significant with respect to Trial 2.*

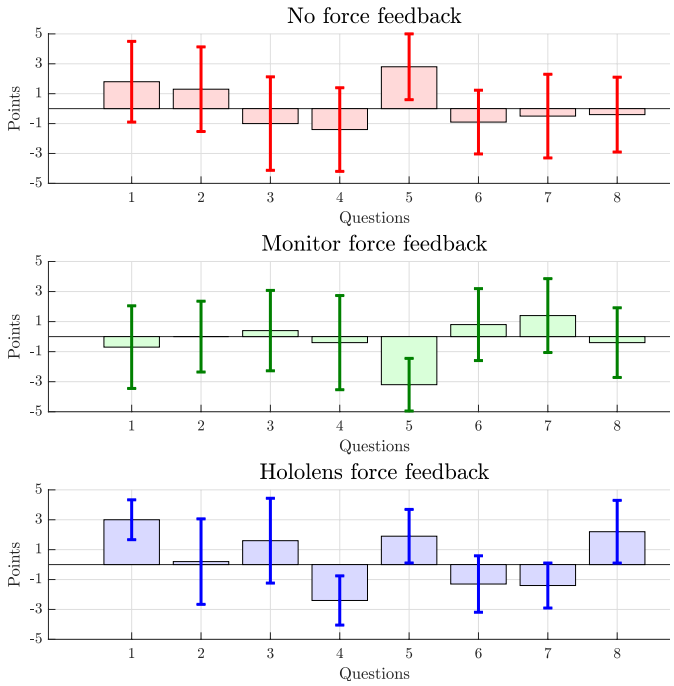


**Figure 4.12:** *Likert scale questionnaire scores for the collaborative polishing experiment in the three experimental settings.*

interesting suggestion came from one of the participants who advised us to show in the HoloLens display, instead of the current value of force, the force signal as in the case of the monitor (see Figure 4.9 and 4.10).

**Close-proximity teleoperation** Although teleoperation applications are mainly associated with robots operating in remote, far-off sites, a large number of tasks in the industrial field requires teleoperation because of their potential danger, such as management of toxic waste, or human sourced contamination, such as in clean isolator systems. In some cases, the operator can be as close as possible to the isolated environment, to have a clear view of the task space for better situational awareness. There are applications based on a direct visual feedback, especially in the industrial field, that has been largely investigated for inspection and maintenance operations in hazardous environments [30], or with the aim of avoiding human sourced contamination, such as, for instance, teleoperated cleaning robots for radioactive environments [102]. Nevertheless, precise manipulation of hazardous or toxic substances, such as radioactive materials, usually requires the employment of shielded glove box isolators. In such cases, the operator directly insert his/her hands into the isolator gloves, to perform tasks inside the box without breaking containment. This kind of operation can lead to potential danger for the operator, being in close contact with the toxic substances. A teleoperation interface is potentially able to address these issues, provided that an accurate task feedback is given to the user.

To this end, in this paragraph we introduce a novel close-proximity teleoperation framework, through which the user awareness is enriched using an AR interface. This choice was led by the effectiveness recently demonstrated by AR interfaces in HRC

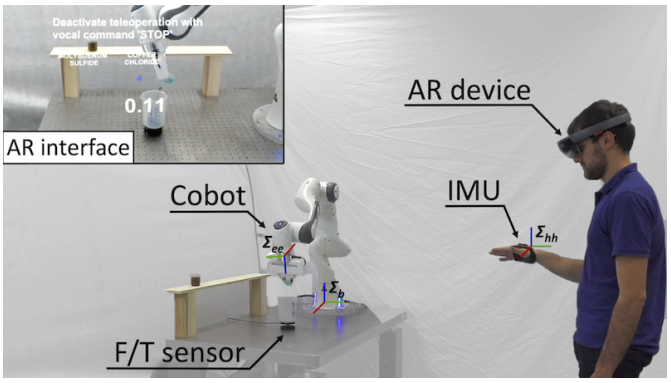


**Figure 4.13:** *The close-proximity teleoperation task is supported by an instant feedback shown through the AR interface.*

scenarios [45, 114]. Although the system is not limited to a specific application, as a proof of concept we demonstrate how teleoperation can be employed in place of glove box isolators, preventing the operator to be in close contact with potentially hazardous materials, or avoiding a human sourced contamination. Figure 4.13 shows the setup of the experiment.

Three main modules have been developed and integrated to build the software architecture illustrated in Figure 4.14: 1) a robot controller that retrieves the robot status and commands the robot motions, 2) an AR communication handler necessary to receive user inputs from the AR device, and to send the information to render the AR holograms, and 3) a FSM in charge of coordinating all the components of the system.

The robot controller embeds a Cartesian impedance controller that is responsible of computing the robot joint torques needed to reach the target pose received by the FSM. This is achieved by means of a fifth-order polynomial trajectory planner as illustrated in Section 3.3.2. This unit, reading the robot status, also regulates the switching between

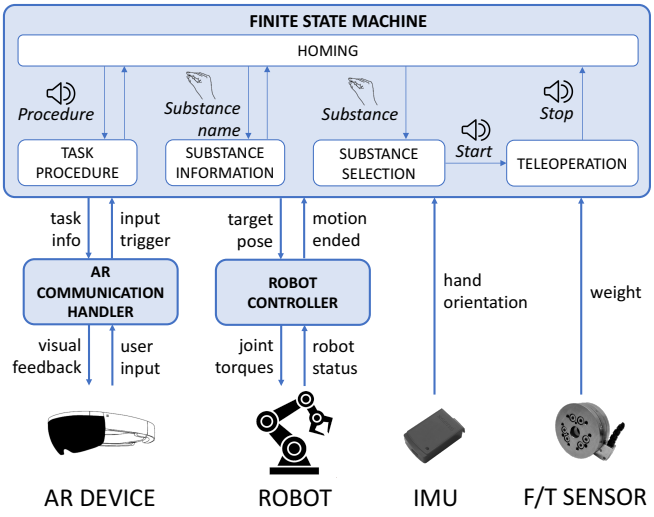


**Figure 4.14:** *The system architecture is composed by three units that communicate with an AR device, a robotic manipulator, an IMU and a FT sensor. The arrows within the Finite State Machine show the user inputs needed to trigger the different states.*

FSM states by triggering motion ending acknowledgments.

The AR communication handler communicates with the AR device and the FSM. It receives from the latter the task info that are translated in visual feedback shown on the AR device to augment the user awareness, and sends back the trigger given by the user input. This can be either specified by voice command or hand gesture acting upon the targeted element, based on the eye's gaze as presented above.

Lastly, we designed a FSM that coordinates the data coming from the aforementioned modules and the ones received by two other sensors, i.e. an inertial measurement unit (IMU) and a FT sensor. The IMU is mounted on the operator right hand and provides information about the hand orientation, needed for the robotic gripper teleoperation. The FT sensor is placed in the hostile environment with an empty beaker placed on top, and acts as a scale to retrieve the quantity of the substances that will be poured inside the container. The FSM initial and central state is given by the "Homing" state, where the system returns after performing each one of the three possible actions hereafter described, and graphically represented in Figure 4.15.

*a) Task procedure:* the operator, by sending the vocal command "Procedure" can list the task process shown as an hologram that appears in front of him/her. This list can be moved in a more convenient position with the *tap* hand gesture, i.e. simulating a mouse click with the thumb and the forefinger, not to hinder the operator's view.

*b) Substance information:* looking at the substance's name located on the shelf, the operator can list the material information as holograms with the *tap* hand gesture.

*c) Substance selection and Teleoperation:* the operator can select the desired sub-

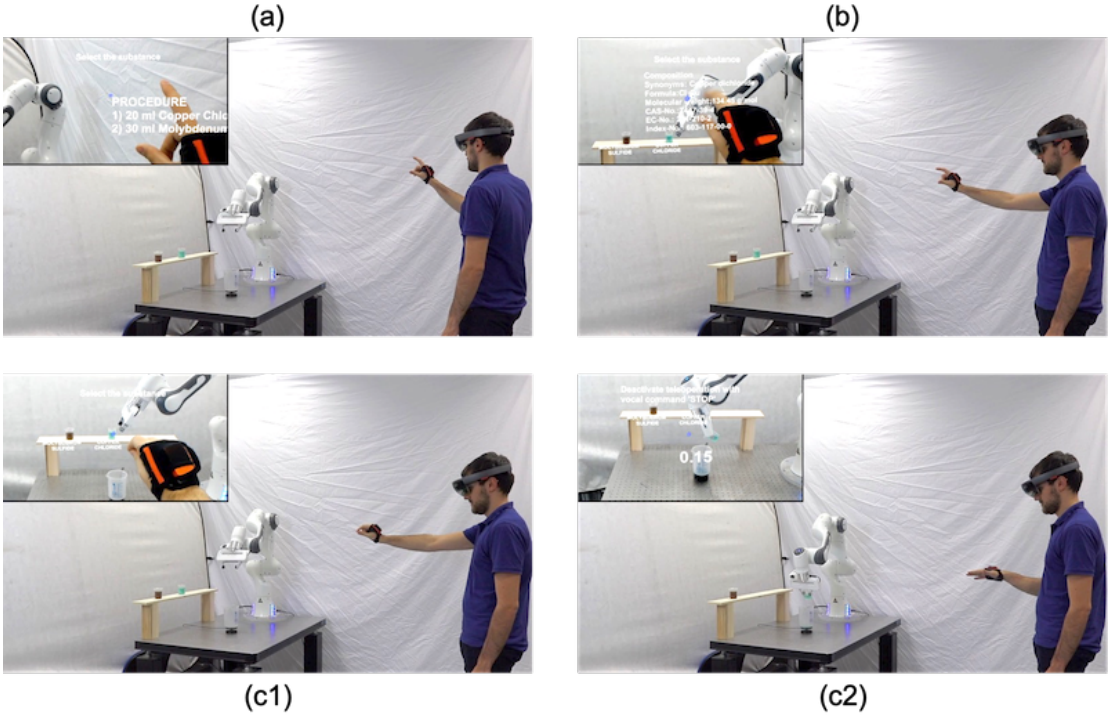


**Figure 4.15:** *Three actions can be selected by the user: Task procedure (a), Substance information (b), and Substance selection and Teleoperation (c1 - c2).*

stance with a *tap* hand gesture, while pointing the gaze on the relative beaker. The robot then picks the substance, and moves it on top of the main beaker placed on the scale. With the vocal command "Start", the operator enables the teleoperation phase and can pour the desired quantity of the substance by rotating his/her right hand. The weight of the main beaker content is shown through the AR interface. During teleoperation, the robot end-effector pose with respect to its base, $T_b^{ee}$, is computed as follows:

$$T_b^{ee} = T_{b\_initial}^{ee} T_{hh\_initial}^{hh} \tag{4.2}$$

where $T_{b\_initial}^{ee}$ represents $T_b^{ee}$ at the beginning of the phase and $T_{hh\_initial}^{hh}$ the transformation of the current human hand with respect to its initial value. To reduce the chances of human errors and increase the efficiency of the method, most of values of $T_{hh\_initial}^{hh}$ are kept fixed, and the only variable that is subject to changes is the rotation on the axis in the direction of the human arm. Once the user pours the right substance quantity, he/she can disable the teleoperation with the voice input "Stop", the robot places back the beaker on the shelf, and comes back to the "Homing" state, ready to start over another task.

The cobot employed in this experiment was the Franka Emika Panda robotic arm, the FT sensor was an ATI Nano17 Force/Torque sensor, and as IMU we used one element of the MVN Biomech suit (Xsens Tech$^{TM}$). Copper Chloride and Molybdenum Sulfide have been used as substances. For this feasibility study, the experiment was performed by 5 subjects in two different scenarios, with and without the AR interface feedback, measuring the time to complete the task and the accuracy. Without feedback the subjects could rely only on the graduate beakers, marked on the side with lines indicating the volume contained, while the AR interface enriched the situational awareness with holograms showing the amount of poured substance. The average time to pour the two substances without the AR feedback was $20.77s$, while with the feedback $16.87s$, i.e. $18.8\%$ less. Without AR feedback, the experiments showed on average an error of $13.5\%$, while with the holograms feedback the subjects reached a much more accurate level of precision, resulting in a $2.2\%$ error. A video of the experiment is available at [16].

## 4.2 A far-proximity approach to robot interaction planning and control

In the previous section, we demonstrated how the human-in-the-loop contribution was needed to plan the robot movements in a scenario where a direct physical or visual contact was feasible. In this section, instead, we consider another approach, in which the human and the robot can not come into contact, either because the human cannot enter the robot area (e.g. for safety reasons), or because the robot is located in a remote area, with respect to the human (e.g. in space applications). To this end, we developed an intuitive teleoperation interface for remote loco-manipulation control.

### 4.2.1 Teleoperation interface for loco-manipulation control

Teleoperation technologies are widely used in space exploration [27], disaster relief [108], surgery [126] and surveillance due to the existence of risks to humans or unreachable physical distances. A typical teleoperation system is usually constructed in a master-slave architecture, in which the key components are control methods developed for

slave robot, human-robot interfaces sending commands to robot while receiving a feedback for human, and communication support to avoid large time delays [54]. Position- or velocity-based teleoperation systems with force feedback are among the most common to benefit from such an architecture [144].

More recently, the introduction of active impedance control techniques in the teleoperation control architecture has made significant improvements in robot responses to the uncertainties arising from the remote task dynamics or the environment [184]. In most cases, however, the impedance parameters in teleoperation systems are kept constant, which limits the interaction performance of robots in the execution of highly dynamic tasks such as drilling, door opening, valve turning, chipping, and may result in task failure.

Towards delivering a more intuitive and effective way to adjust the impedance parameters of teleoperated robots in such interaction scenarios, Ajoudani, et al., proposed the concept of tele-impedance [5], as an alternative technique to unilateral and bilateral teleoperation. Tele-impedance enriches the command sent to the slave robot by combing the master's estimated position and the stiffness references. The compound reference commands are then realized by the remote impedance controller without explicit force feedback to the operator. The application of tele-impedance in remote control of robotic arm [3], hand [80], and dual-arm [118] systems has demonstrated this control concept's high potential. Nevertheless, tele-impedance control method has only been employed in the teleoperation of fixed-base platforms.

On the other hand, modern teleoperation systems call for the crucial contribution of mobility to enable their users to navigate the systems to the points of interest. Additionally, the mobile base extends the workspace of the manipulator and provides improved flexibility brought by the extra DoF. However, although a wide variety of mobile robots with legged (e.g., WALK-MAN [182], TORO [52]) or wheeled (e.g., Rollin' Justin [31], TWENDY-ONE [89]) locomotion capabilities exist, the lack for an intuitive and interaction-efficient interface for the control of robot loco-manipulation has prohibited their deployment in challenging remote interaction scenarios. In fact, most of the current mobility based teleoperation interfaces (e.g. [151, 172]) focus on the mobility aspect, and do not provide effective solutions for the control of remote interactions in challenging manipulation tasks.

Although mobility brings more flexibility, it also increases the difficulty to design a friendly loco-manipulation interface for the operator to control a mobile manipulator. In most cases, a joystick interface is used due to its simplicity [190]. However, when dealing with many DoF, it could be confusing for the users. The choice of a joystick can also be employed for the locomotion primitive only, leaving the manipulation mode associated to human whole-body movements [152]. Graphical user interface (GUI) running on a computer is also widely adopted in the teleoperation of mobile manipulators [78], although a complex GUI may not be intuitive and effective enough for the operators. Haptic devices such as Falcon are also used in some cases to get force feedback [10], although most of them have only six DoF and are designed to control a fixed manipulator, not a mobile one. In a few situations [85, 155], an exoskeleton or an extra manipulator is employed to measure the position of human arm and provide force feedback to human, which makes teleoperation more intuitive. However, these interfaces increase the system complexity.

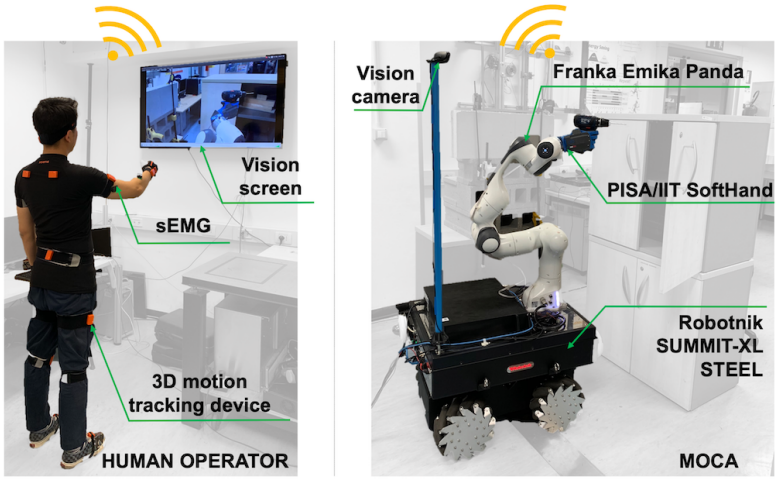


**Figure 4.16:** *The human operator remotely teleoperates MOCA by moving his/her body, thanks to the visual feedback shown on the screen and provided by the camera mounted on the robot.*

Accordingly, the aim of this section is to propose a novel teleoperation interface for control of robot loco-manipulation in remote environments, exploiting the potential of the MOCA platform we developed, along with the implementation of the whole-body impedance controller introduced in Section 2.3.2. To merge the tele-impedance control with robot locomotion ability, an advanced teleoperation interface is presented to enable MOCA control in two modes: Locomotion and Manipulation. After describing the control architecture that was developed for this framework, we illustrate the experimental results that contributed to its validation in the locomotion and manipulation control of MOCA in remote environments, in applications such navigation, door opening, wall drilling, and valve turning. The results reported in this section led to the scientific publication [191].

#### Human-Robot loco-manipulation interface

In this section, a human-robot loco-manipulation interface is developed to enable intuitive control of MOCA mobility and arm interaction. Figure 4.17 illustrates the two control modes, i.e., Manipulation and Locomotion.

For the Locomotion mode, MOCA arm and base controllers are decoupled to avoid asynchronized dynamic responses between the manipulator and the mobile platform[1]. In this phase, the virtual torques of the mobile-base admittance controller are computed based on the estimation of human Center of Pressure (CoP) resulting from the body inclinations, which correspond to the directions of motion in remote environment.

On the other hand, for the Manipulation mode, the operator's arm position and stiffness commands are tracked in real-time and replicated by the MOCA's whole-body impedance controller. Details of the human-robot interface are explained below.

**Locomotion Mode** In the Locomotion mode, the motion of the mobile platform is controlled on the basis of the position of the whole-body CoP of the subject, which is

[1] Although this can be avoided using the coupled whole-body controller, however, several hard constraints must be imposed to avoid the motion of arm with respect to the mobile base.

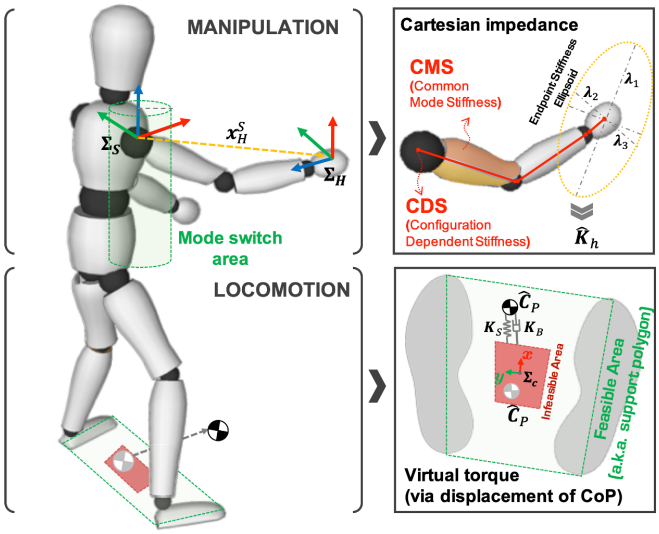


**Figure 4.17:** *Whole-body Manipulation (top) is performed through the tele-impedance interface, Locomotion (bottom) is controlled with virtual torques, that are based on the CoP displacement.*

estimated by means of a Statically Equivalent Serial Chain (SESC) technique, as described in [105], using data collected through the 3D motion tracking of the subject. In this phase, the manipulator is controlled independently with a Cartesian impedance controller and no command is sent from human side. In order to make it compliant with possible obstacles, low stiffness parameters are set in the controller.

By taking into account the human's support polygon (see bottom picture in Figure 4.17), that is the horizontal region bounded by the top and the bottom of the right and left foot, we defined a "dead-zone" area which corresponds to a specific percentage of the support polygon (by default the $50\%$, but it can be set differently) and is treated as a no-movements area for the mobile platform. This consideration is to avoid undesired movements of the mobile platform resulted by small, involuntary body sways of the operator.

Accordingly, we computed the displacement of the CoP $\boldsymbol{\Delta C}_P$, which is set to zero if the CoP lies inside the sub-polygon, while it is equal to the distance between the CoP and the closest side of the sub-polygon, otherwise. The resulting CoP displacement is then used to compute the virtual torques as:

$$\boldsymbol{\tau}_v^{vir} = \boldsymbol{K}_S \boldsymbol{\Delta C}_P + \boldsymbol{K}_D \boldsymbol{\Delta \dot{C}}_P, \tag{4.3}$$

where $\boldsymbol{K}_S \in \mathbb{R}^{2\times 2}$ and $\boldsymbol{K}_D \in \mathbb{R}^{2\times 2}$ are the virtual stiffness and damping matrices respectively. Here, only translational motions on $x$ and $y$ axes are considered. The computed torque $\boldsymbol{\tau}_v^{vir}$ is sent to the mobile base admittance interface, to employ the Locomotion mode.

The virtual stiffness and damping matrices values were experimentally chosen based on the resulting mobile-based velocity, which is achieved by body inclinations of the operator.

**Whole-body Manipulation Mode** Tele-impedance [5] is a control paradigm developed in the last five years. It consists in tele-operating a robot through an impedance controller by measuring and replicating the user's limb pose and impedance on the slave robot in real-time. The user's impedance is estimated by monitoring the muscles' activity through the use of sEMG and interpreted to estimate the impedance of the human limb. The estimation may involve a detailed muscle model, usually a Hill-based one or a derivation of it.

Towards the tracking of the human physical interaction behavior in 3D space using a principled simplification approach, recently [3], the tele-impedance concept has been extended based on the dependency of the arm endpoint stiffness to both geometric human arm configuration (Configuration Dependent Stiffness - CDS) and muscular activity (Common Mode Stiffness - CMS).

CDS includes the effect of arm configuration and muscle moment arms that contribute to the variations in the geometry of the arm endpoint stiffness. The arm kinematics is retrieved by the 3D motion tracking system, which enables the computation of the arm Jacobian ($\boldsymbol{J}_a(\boldsymbol{q}_a)$), with $\boldsymbol{q}_a \in \mathbb{R}^7$ being the arm joint angles. Using the muscle attachment points, and the length variations over the joint angles, i.e. the muscle Jacobian ($\boldsymbol{J}_{musc}(\boldsymbol{q}_a)$) can be computed online [3].

CMS, on the other hand, implements a coordinated co-activation of the arm muscles, while its tracking is achieved by a co-contraction index ($a_{cc}$), calculated from the dominant and easily accessible muscles of the arm for surface electromyography measurements, i.e. the Biceps Brachii (BB) and Triceps Brachii (TB). Through a pre-define muscle synergy matrix ($\boldsymbol{K}_{syn}$), this index contributes to modifications in the volume of the endpoint stiffness ellipsoid.

Finally, the congruence conservative transformation from the joint space to the Cartesian space of the human arm can be written to obtain the estimated endpoint stiffness $\hat{\boldsymbol{K}}_h$ (see details in [3]):

$$\hat{\boldsymbol{K}}_h = \boldsymbol{J}_a^{+T}(\boldsymbol{q}_a)(\boldsymbol{J}_{musc}^{\top}(\boldsymbol{q}_a)a_{cc}\,\boldsymbol{K}_{syn}\boldsymbol{J}_{musc}(\boldsymbol{q}_a))\boldsymbol{J}_a^{+}(\boldsymbol{q}_a). \tag{4.4}$$

The identification of the parameters in Eq. 4.4 was achieved in an off-line experimental phase as described in [3].

#### Control architecture

The components illustrated in the previous sections need to be integrated in a unique framework, in order to build an effective system capable of following the operator's commands. The control flow of the software architecture is depicted in Figure 4.18. The operator, assuming a standing pose, is provided with two ways to command the robot, the Locomotion mode and the Manipulation mode, and is able to switch between them at any time through the *Mode switch* action. This command mode change is enabled by some predefined operator's arm gestures. When the user arms are at his/her sides, the Locomotion mode is activated, while by raising the right arm, the user can switch to Manipulation mode. This motion is detected when the human hand position (i.e., position at the origin of $\boldsymbol{\Sigma}_H$ with respect to $\boldsymbol{\Sigma}_S$) goes beyond a *mode switch area* shaped as a cylindrical constraint (see Figure 4.17). To switch back to the Locomotion mode, the operator needs first to raise also his/her left arm, and then to take down both arms simultaneously, thus going back to the starting pose with both arms at his/her

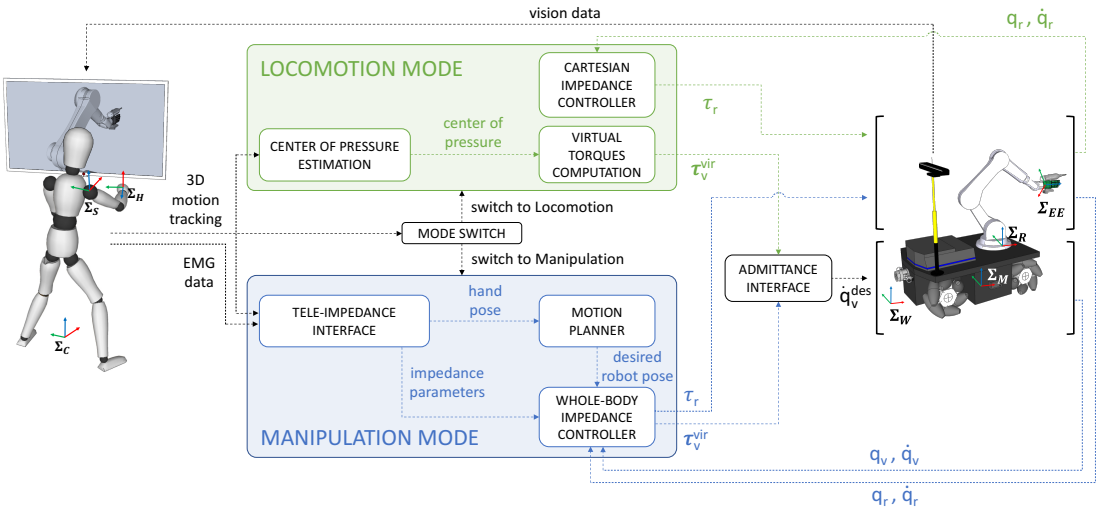


**Figure 4.18:** *The control architecture of the presented framework. The dotted lines represent the data exchanged by the modules: in green the data exchanged during Locomotion, in blue during Manipulation, and in black the data that always flow when the system is active.*

sides. Note that, simply lowering the right arm to switch mode, without using the left arm, would not be a feasible option, since the robotic arm would still be subject to teleoperation movements.

In the Locomotion mode, the *Center of Pressure Estimation* module receives as input the 3D motion tracking data and estimates the human whole-body CoP, that is given in turn to the *Virtual Torques Computation* module, whose output are the virtual torques $\boldsymbol{\tau}_v^{vir}$ that are sent to the mobile platform *Admittance Interface*. The robotic arm is instead controlled by a standard Cartesian impedance control. In this mode, the mobile platform and the robotic arm are controlled independently to avoid unnecessary excessive movements of the upper part of the system and to achieve a smoother behavior.

On the contrary, in the Manipulation mode, the upper and the lower part of the system are commanded by a *Whole-body Impedance Controller* as a unified framework. The *Motion Planner* unit takes as input the human hand displacement with respect to its initial pose (with $\boldsymbol{\Sigma}_H$ as reference frame) and, at every time step, it adds this displacement to the initial robot pose computed in the world frame $\boldsymbol{\Sigma}_W$. The human impedance, estimated by the *Tele-impedance Interface*, is directly mapped to the robot Cartesian impedance.

#### Experimental results

To validate the proposed method, we carried out experiments switching between the two modes. The operator teleoperated the robot, localized in a remote environment, relying on the visual data provided by a camera mounted on MOCA. These data were streamed on a screen located in front of him/her. The 3D whole-body motion tracking data were retrieved thanks to a MVN Biomech suit (Xsens Tech$^{TM}$) provided with 17 inter-connected IMUs. The frequency of the human-robot loco-manipulation interface is based on the 3D motion tracking frequency, i.e. $80Hz$, while the impedance controllers run in real-time at $1KHz$, and the admittance interface at $300Hz$.

In order to show the framework's full potential, the user had to accomplish the fol-

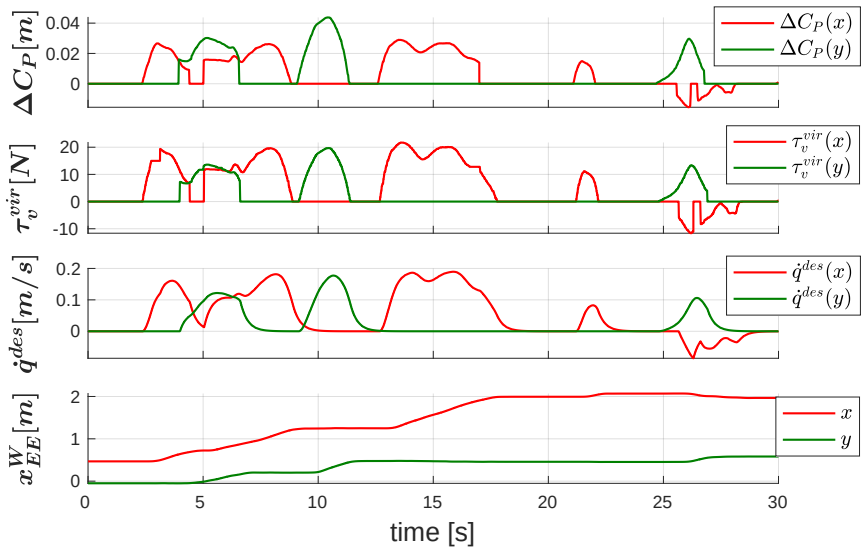


**Figure 4.19:** *Locomotion mode: the human operator, by bending frontally and/or laterally, modifies his/her CoP position. Based on its displacement $\boldsymbol{\Delta C}_P$, the virtual torques $\boldsymbol{\tau}_v^{vir}$ are computed and the Admittance Interface translates them in a desired velocity $\dot{\boldsymbol{q}}^{des}$ through which MOCA changes its pose $\boldsymbol{X}_{EE}^W$.*

lowing subtasks: using the Locomotion mode, guide the robot in front of a closet, switch to Manipulation mode to open the closet door, change back to the first mode to move forward, and finally switch again to Manipulation mode to grab a drill placed inside the closet and pierce a wall on the left.

Figure 4.19 represents the Locomotion mode, in which the user drove the robot in front of the closet. The plots show the operator CoP displacement $\boldsymbol{\Delta C}_P$, on which are based the virtual torques $\boldsymbol{\tau}_v^{vir}$ calculated as in Eq. 4.3, where $\boldsymbol{K}_S = 300N/m$ and $\boldsymbol{K}_D = 2\zeta\sqrt{\boldsymbol{K}_S}$, with $\zeta = 1$. $\boldsymbol{\tau}_v^{vir}$ are used as input to the *Admittance Interface* module, that computed the mobile platform desired velocity $\dot{\boldsymbol{q}}^{des}$ through Eq. 2.17, where $\boldsymbol{M}_{adm} = diag(32.5\ 32.5\ 6.5)\ [Kg\ Kg\ Kgm^2]$ and $\boldsymbol{D}_{adm} = diag(104\ 104\ 20.8)$ $[Ns/m\ Ns/m\ Nms/rad]$. The last row of the figure describes how the robot end-effector pose changes with respect to the world frame $\boldsymbol{\Sigma}_W$.

Once the robot was guided close enough to the closet, the operator raised the right arm to switch to Manipulation mode. In this stage, depicted on Figure 4.20a, the user had to open the door of the closet. The first two plots highlight the coupling between the movements of the human hand with respect to his/her shoulder ($\boldsymbol{x}_H^S$) and the motion of the robot end-effector with respect to the world frame ($\boldsymbol{x}_{EE}^W$). The third plot depicts the diagonal values of the Cartesian stiffness matrix $\boldsymbol{K}$ estimated by the *Tele-impedance Interface* module and set to the robot *Whole-body Impedance Controller*. Since the door opening had to be carried out mainly on the *x-axis*, as shown by the external interaction forces $\boldsymbol{F}_{ext}$, we can notice that the impedance gains reached high values only on that axis remaining compliant on $y$ and $z$ axes. Only in this way a successful execution of the task was possible, in fact to open the door the robot had to comply with the door constraints especially in $y$ direction while remaining stiff in $x$ direction to be able to open it. This avoided the generation on unnecessary high interaction forces in $y$ and $z$ axes.

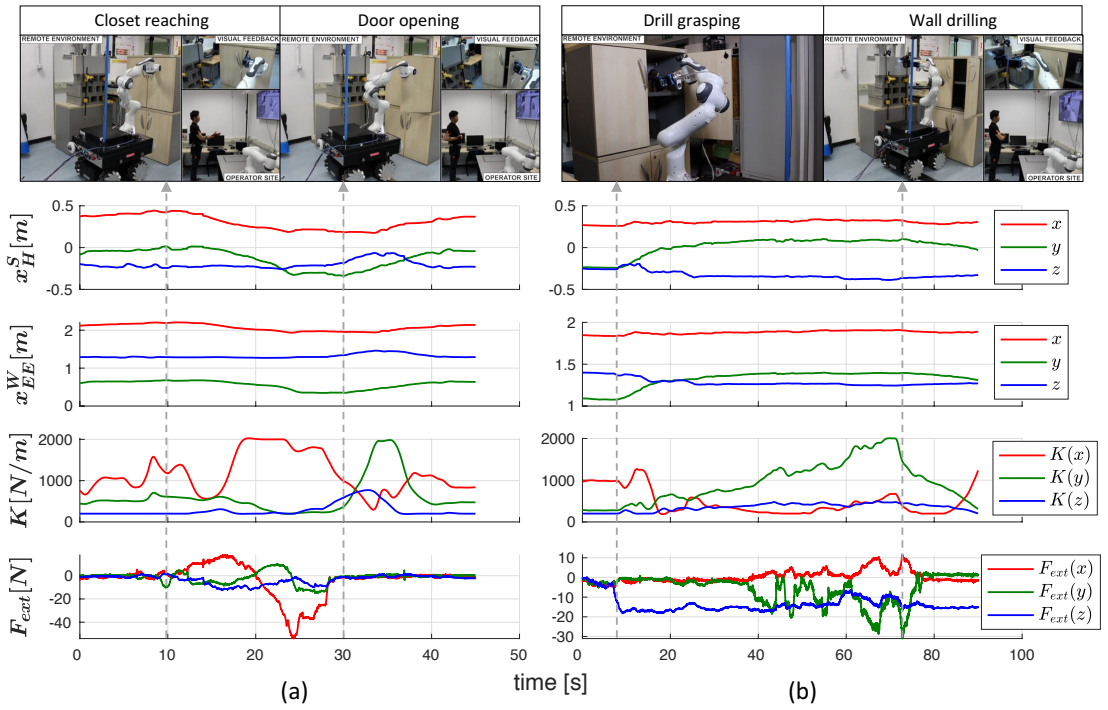


**Figure 4.20:** *Manipulation mode: the human operator performs the door opening (a), and the wall drilling (b). The first two plots show the coupling between the position of the human hand with respect to his/her shoulder frame $\boldsymbol{X}_H^S$, and the robot end-effector pose expressed in the world frame $\boldsymbol{X}_{EE}^W$. The third plot depicts the Cartesian stiffness estimated by the Tele-impedance Interface that is mapped onto the robot Whole-body Impedance Controller. In the last plot, the external interaction forces are represented, highlighting the pulling of the closet handle on x-axis (a), and the wall piercing on the y-axis (b).*

After having opened the closet door, the operator switched back to Locomotion mode and moved in a configuration in which it was feasible to both grab the drill inside the closet and pierce a wall on its left. We omit the relative plots since they are very similar to the ones shown in Figure 4.19. The plots describing the last Manipulation mode are depicted on Figure 4.20b. In the first part ($time \simeq 8s$) the operator grabbed the drill, as highlighted by the sudden negative variation of the external interaction forces $\boldsymbol{F}_{ext}$ on the *z-axis* given by the tool weight ($2Kg$). In this phase, the Cartesian stiffness $\boldsymbol{K}$ gains are high only in $x$ direction since the human arm is fully extended frontally to reach the tool inside the closet. Next, the operator led the robot to the left as it can be noticed by the positive variation on *y-axis* in the first two plots, and pierced multiple times the wall. This time the Cartesian stiffness $\boldsymbol{K}$ values are high only in $y$ direction, as the external interaction forces $\boldsymbol{F}_{ext}$. Due to the increase of stiffness in *y-axis*, the drill can penetrate inside the wall, and remain relatively compliant in other axes so that any misalignment is gently treated.

To show the adaptability of the framework to different tasks and people, another subject performed a valve turning task. Since the results were similar, the plots are not shown here. However, the experiment is included in the multimedia attachment of the manuscript, that can be found at [191].

#### Discussion

The whole-body impedance controller improves the robot interaction performance in terms of accuracy and safety, in comparison to a decoupled system, i.e. Cartesian impedance control on the robotic arm and admittance control on the mobile platform. A good example is characterized by the door opening phase: the *x-axis* forces exerted by the door handle while pulling it, make the mobile platform move back on the same axis. Without this coupled control, only the arm would move back and the door would crash against the mobile platform, resulting in a failure of the task and causing a damage to the environment and to the robot itself.

Future developments will include the extension of the framework to dual-arm manipulation. In this case, the CoP interface brings even more benefits, since the two human arms/hands are dedicated to command the tele-impedance interface. Nevertheless, to highlight the importance and intuitiveness-of-use of the developed CoP interface in this work, we performed additional experiments and compared its performance to a case where robot mobility was controlled using a joystick. The task was to navigate the mobile platform using the two locomotion interfaces (CoP or joystick), and to perform a painting task on a wall in two different areas (see Figure 4.21). The operators were asked to zoom in the camera view manually, using a button, after each painting action to control the quality of the painting tasks. This implied that, when using the joystick, the subjects had to grasp and switch between the joystick and the camera button repeatedly. Ten healthy subjects participated to the experiments.

After the execution of the two tasks, they were asked to fill out a Likert scale questionnaire to compare the CoP and the joytick interface. The questionnaire included 8 statements:
Q.1 Accomplishing the task with the CoP interface was less mentally demanding;
Q.2 Coordination of the robot mobility and manipulation was more intuitive using the

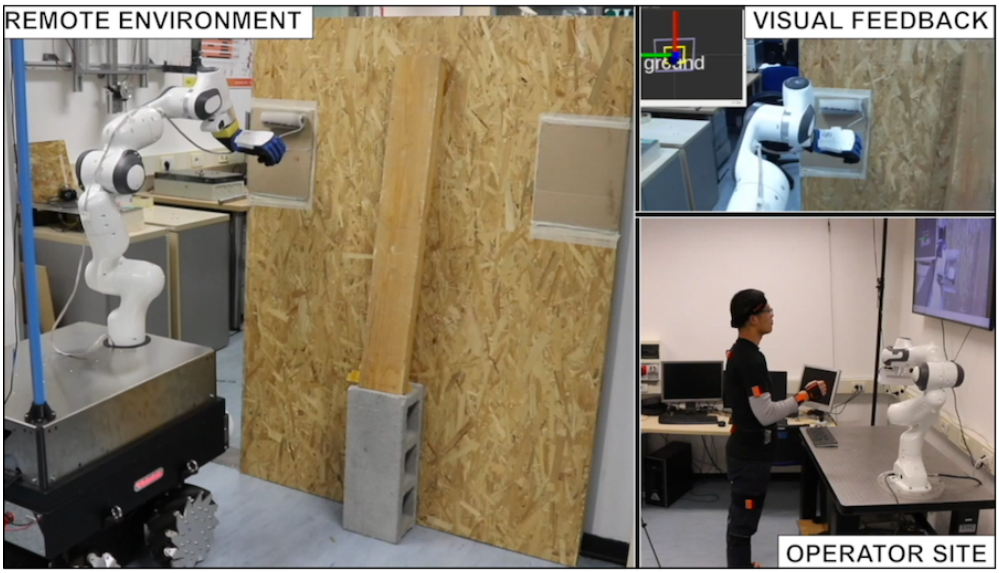


**Figure 4.21:** *A painting task was performed both with the CoP interface presented in this work and with a joystick interface. The former allows both hands to be free, while with the latter reduces the operator capabilities in case they need the hands to perform other tasks.*

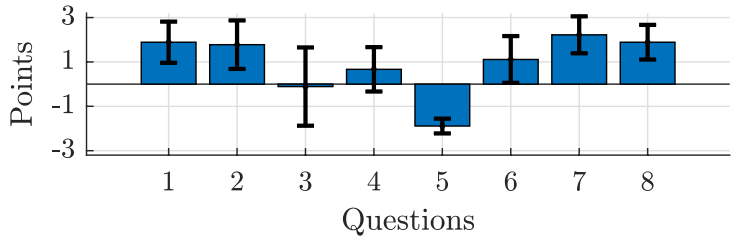


**Figure 4.22:** *Likert scale questionnaire scores about the human subjective evaluation comparing the CoP and the joystick interfaces.*

CoP interface;
Q.3 Accomplishing the task with the joystick interface was more physically demanding;
Q.4 I had to work harder to accomplish my level of performances using the joystick interface;
Q.5 I felt more discouraged, stressed, and annoyed using the CoP interface;
Q.6 I felt more discouraged, stressed, and annoyed using the joystick interface;
Q.7 The CoP interface has a higher potential to make the execution of complex remote loco-manipulation tasks easier for a user;
Q.8 Overall, I felt I was improving the mixed loco-manipulation performance with the CoP interface.

The possible answers ranged from *strongly disagree* to *strongly agree*, with an assigned score of -3 and +3, respectively. The results, illustrated in Figure 4.22, show that the users found the presented interface more intuitive, less mentally demanding, and with a higher potential to make the execution of complex loco-manipulation tasks easier, in comparison with a joystick interface.

We also measured the average time to complete the task with the two modes obtaining $117,7s$ for the CoP interface and $158,9s$ using the joystick, across all subjects and the experiments.

CHAPTER 5

# Conclusion

The thesis herein presented introduced novel control algorithms and interaction policies aimed to enhance the robot interaction capabilities, so as to deal with unknown and unstructured environments. To this end, two different approaches have been considered, based on the complexity level of the tasks carried out by robotic systems. In view of a lower task complexity, autonomous robotic behaviors have demonstrated to guarantee the necessary efficiency in achieving their goal. On the other hand, when the complexity of the task rises, a collaborative scenario is preferable, where a human is introduced in the loop in order to plan the actions to be taken.

Following the first approach, we developed a framework that integrates multiple components to achieve a context-aware and adaptive interaction between the robot and uncertain environments, whose core resides in an adaptive impedance controller able to regulate robot quasi-static parameters, i.e., stiffness and damping, based on the robot sensory data and vision. To this end, we introduced the concept of *interaction expectancy*, that allows for a clear distinction between expected interactions and external disturbances. When no interactions with the environment are predicted, the impedance controller parameters assume very compliant profiles, in order to avoid the generation of unnecessary high interaction forces. In fact, gently responding to unforeseen external perturbations, as collisions, minimizes the damages to the environment and to the robot itself. On the contrary, if an interaction is expected, the stiffness and damping parameters needs to be adapted. We demonstrated that in view of a significant Cartesian error, so when the robot lags behind with respect to its desired trajectory, the impedance values need to be increased, so as to carry out the task as planned. On the other hand, if the robot is accurately tracking the designed path and there is a reduction in the sensed external forces, the impedance parameters have to be decreased, in order to avoid unmotivated large values.

At first, we evaluated the impedance regulation along selective Cartesian axes, experimentally validating the framework in a disaster response scenario, i.e. carrying out a debris removal task with a fixed base manipulator and the humanoid robot WALK-MAN. Since this preliminary work highlighted the potential of such framework, we extended the impedance regulation developing a more mature version of the controller, where the Cartesian stiffness and damping profiles are adapted in any arbitrary direction, which coincides with the direction of interaction. This adaptation is based on the previous experience in interacting with similar environments and the real-time robot sensory data, following a three stages paradigm: *explore*, *identify*, and *interact*. This concept was inspired by the humans' way of adapting to their surroundings, by constantly building internal models of the external world, while exploring and identifying it. When interacting with new or similar environments, the prior knowledge is used as a preparatory strategy, while keeping open the possibility of adaption, to update our internal knowledge [99]. This idea was experimentally validated in an agricultural setup, where we also pointed out the advantages of the proposed multi-axes self-tuning impedance controller with respect to fixed-impedance control methodologies.

The algorithm scalability has been also proven in other scenarios, in the direction of flexible warehouse automation. Adaptive planning strategies have been introduced to provide the system with the ability to cope with unexpected environmental and operational changes. Inspired by the observation of the human behavior, an item sorting strategy was designed, implementing a trade-off between the task execution accuracy and environmental perception uncertainty. The robot control module exploits the self-tuning Cartesian impedance controller, to implement an adaptive strategy for the picking, placement, and sorting of the items in a box. A vision module has been developed for the detection and tracking of the environment (e.g., box and the items), creating an occupancy grid in real-time, to continuously update the robot trajectory planner with the occupied portions of the detected box and their coordinates. The efficacy of the framework in performing a flexible box-filling task using a robot, autonomously or in collaboration with a human, has been evaluated through several experiments.

Next, we presented a novel collaborative robotic approach to the problem of autonomous transportation and positioning of vehicles that present kinodynamic constraints, using a standard industrial pallet jack as a representative example. The proposed framework can navigate a mobile cobot to a target pose, detect the handle of the pallet jack and approach to grasp it, and transport it to a desired position and orientation in a warehouse environment. The experimental results demonstrated the high potential of the proposed framework in seamless and accurate positioning of the pallets. This was achieved in spite of the dynamic uncertainties in perception, navigation, and manipulation, and the imposed kinematic constraints of the exploited standard pallet jack. Future works will aim to improve the framework reliability in response to uncertainties, such as checking the feasibility of the maneuver space in narrow environments, and ensuring collision-free paths during the repositioning phase, for instance by pausing the motion if a dynamic obstacle is moving along the planned path. Although this work only focused on the problem of pallet jack transportation and repositioning, the proposed framework has the potential to automatize similar repetitive and effort intensive tasks, such as repositioning of large industrial carts, airport mobile stairways, etc. This is because, such systems exhibit similar mobility constraints that can be dealt with,

using the proposed controller.

In the second part of this thesis, new thinking and techniques have been presented to deal with high task complexity. The human-in-the-loop contribution has proved to guarantee a successful execution in different application areas, both in close- and far-proximity scenarios. In the first case, we demonstrated how the user can command the robot motion, while establishing a direct view contact with the robot. Different approaches have been studied. With the aim of physically guiding a cobot through the environment, a human-robot interface has been developed to subsume the advantages of cobots and supernumerary limbs. An admittance interface was developed to transfer human-intended forces to desired robot end-effector trajectories, which were then executed by a whole-body impedance controller. A hand gesture recognition system was implemented to enable the activation of human-robot conjoined actions, and to provide the user with the ability to assign a high priority to the arm or the mobile base movements in different phases of the task. We proved that, although the robot, as a supernumerary body, was physically coupled to human, the effect of weight was not transmitted to the human partner, who in turn perceived the physical support.

Within the close-proximity level, AR interfaces have also been investigated, with the aim of enhancing the user awareness about the task. The primary objective of this development was to assist workers, in such a way they could feel the interaction with the cobot as user-friendly as possible. Two different experimental scenarios have been considered. In the first one, the user had to perform a collaborative task polishing a rough wooden surface with a polisher sustained by the cobot, while the second one was represented by a teleoperation task. In both cases, through the AR device, the user received real-time information about the task, and could send inputs to the framework by means of vocal commands and hand gestures. A subjective evaluation was performed, and the subjects involved in the experiments confirmed that, through the AR interface, they felt an improvement in the task performances (that was also measured through an objective evaluation of the data).

Novel interfaces that entrust the human with the decision-making process, but without involving any direct visual/physical contact, have also been presented. In particular, a far-proximity teleoperation interface for remote robot loco-manipulation control has been developed. Robot locomotion was commanded based on the CoP displacement of the user, while whole-body manipulation exploited a whole-body impedance controller, enriched with the tele-impedance paradigm, that allows for a replication of the user's limb pose and impedance on the slave robot in real-time. The interface was extensively validated in various applications such navigation, door opening, wall drilling, valve turning, and painting. A comparison with the more traditional use of a joystick was carried out, highlighting the potential of the proposed interface. In fact, besides proving to be more efficient, the presented interface was also preferred by the subjects involved in the experiments. Through a subjective evaluation, the users stated that they found the presented interface more intuitive, less mentally demanding, and with a higher potential to make the execution of complex loco-manipulation tasks easier, in comparison with a joystick interface.

In this work, the equation dynamic of mobile platform and manipulator were considered decoupled because of the following assumptions: i) while performing manipulation tasks, the mobile platform was only involved in quasi-static movements; ii) any

external dynamic effect was compensated thanks to the high gain of the low level velocity controller. Furthermore, the limited payload of the arm ($3Kg$) and the much larger weight of the mobile base ($\sim 130Kg$) never resulted in tilting of the base. However, in case future application will not allow these assumptions, the modeling of the coupled terms can be added, and the destabilizing effect of the payload should be identified.

The results presented in this thesis showed how the issue of scarce robot interaction autonomy has been addressed, providing a general idea that has then been applied to many applications, exploiting the trade-off between task complexity and autonomy. Nevertheless, the experimental results applicability has been proved in multiple scenarios as a proof-of-concept. Therefore, future works will aim to validate the presented framework in real world environments, where the uncertainty level is even more prominent. To this end, the methods and concepts presented in this work will be extensively tested especially within the manufacturing area, with the industrial partners we started collaborating with. The final goal is to apply the presented theories in real factories, where the introduced framework will make their way into, both creating resource-efficient manufacturing solutions (human-in-the-loop approach), and helping to reduce human physical stress, automating repetitive and cognitively unexciting industrial tasks (self-governing approach).

# Bibliography


[1] Evan Ackerman and Erico Guizzo. Darpa robotics challenge finals: Rules and course. *IEEE,[Online]. Available: https://spectrum.ieee.org/automaton/robotics/humanoids/drc-finals-course*, 2015.

[2] Gabriel Aguirre-Ollinger, J. Edward Colgate, Michael A. Peshkin, and Ambarish Goswami. Active-impedance control of a lower-limb assistive exoskeleton. In *2007 IEEE 10th international conference on rehabilitation robotics*, pages 188–195. IEEE, 2007.

[3] Arash Ajoudani, Cheng Fang, Nikos Tsagarakis, and Antonio Bicchi. Reduced-complexity representation of the human arm active endpoint stiffness for supervisory control of remote manipulation. *The International Journal of Robotics Research*, 37(1):155–167, 2018.

[4] Arash Ajoudani, Jinoh Lee, Alessio Rocchi, Mirko Ferrati, Enrico Mingo Hoffman, Alessandro Settimi, Darwin G. Caldwell, Antonio Bicchi, and Nikos G. Tsagarakis. A manipulation framework for compliant humanoid coman: Application to a valve turning task. In *14th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 664–670, 2014.

[5] Arash Ajoudani, Nikolaos G. Tsagarakis, and Antonio Bicchi. Tele-impedance: Towards transferring human impedance regulation skills to robots. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 382–388. IEEE, 2012.

[6] Arash Ajoudani, Andrea Maria Zanchettin, Serena Ivaldi, Alin Albu-Schäffer, Kazuhiro Kosuge, and Oussama Khatib. Progress and prospects of the human–robot collaboration. *Autonomous Robots*, 42(5):957–975, 2018.

[7] Alin Albu-Schäffer, Sami Haddadin, Christian Ott, Andreas Stemmer, Thomas Wimböck, and Gerhard Hirzinger. The dlr lightweight robot: design and control concepts for robots in human environments. *Industrial Robot: an international journal*, 34(5):376–385, 2007.

[8] Alin Albu-Schäffer, Christian Ott, Udo Frese, and Gerd Hirzinger. Cartesian impedance control of redundant robots: recent results with the DLR-light-weight-arms. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3704–3709, 2003.

[9] Ali Allahverdi and HM Soroush. The significance of reducing setup times/setup costs. *European Journal of Operational Research*, 187(3):978–984, 2008.

[10] Víctor Hugo Andaluz, Washington X. Quevedo, Fernando A Chicaiza, José Varela, Cristian Gallardo, Jorge S Sánchez, and Oscar Arteaga. Transparency of a bilateral tele-operation scheme of a mobile manipulator robot. In *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*, pages 228–245. Springer, 2016.

[11] Robert J. Anderson and Mark W. Spong. Hybrid impedance control of robotic manipulators. *IEEE Journal on Robotics and Automation*, 4(5):549–556, 1988.

[12] Salvatore M. Anzalone, Sofiane Boucenna, Serena Ivaldi, and Mohamed Chetouani. Evaluating the engagement with social robots. *International Journal of Social Robotics*, 7(4):465–478, 2015.

[13] Roland Arsenault and Colin Ware. Eye-hand co-ordination with force feedback. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 408–414, 2000.

[14] Tamim Asfour, Kristian Regenstein, Pedram Azad, Joachim Schroder, Alexander Bierbaum, Nikolaus Vahrenkamp, and Rüdiger Dillmann. ARMAR-III: An integrated humanoid platform for sensory-motor control. In *2006 6th IEEE-RAS international conference on humanoid robots*, pages 169–175. IEEE, 2006.

[15] Kaveh Azadeh, René De Koster, and Debjit Roy. Robotized and automated warehouse systems: Review and recent developments. *Transportation Science*, 53(4):917–945, 2019.

[16] Pietro Balatti, Alessandro De Franco, Edoardo Lamon, Elena De Momi, and Arash Ajoudani. An augmented reality interface for improving task performance in close-proximity teleoperation. In *Institute for Robotics and Intelligent Machines Conference (I-RIM)*, 2019. Video available at: `https://youtu.be/cLwX-imezi0`.

[17] Pietro Balatti, Fabio Fusaro, Nicola Villa, Edoardo Lamon, and Arash Ajoudani. A collaborative robotic approach to autonomous pallet jack transportation and positioning. *IEEE Access*, 8:142191–142204, 2020. Video available at: `https://youtu.be/HzruNzmwaHo`.

[18] Pietro Balatti, Dimitrios Kanoulas, Giuseppe F Rigano, Luca Muratore, Nikos G. Tsagarakis, and Arash Ajoudani. A self-tuning impedance controller for autonomous robotic manipulation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5885–5891. IEEE, 2018. Video available at: `https://youtu.be/9RdbkIxB-_Y`.

[19] Pietro Balatti, Dimitrios Kanoulas, Nikos Tsagarakis, and Arash Ajoudani. A method for autonomous robotic manipulation through exploratory interactions with uncertain environments. *Autonomous Robots*, pages 1–16, 2020. Video available at: `https://youtu.be/D_tQye61gfs`.

[20] Pietro Balatti, Dimitrios Kanoulas, Nikos G. Tsagarakis, and Arash Ajoudani. Towards robot interaction autonomy: Explore, identify, and interact. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9523–9529. IEEE, 2019.

[21] Pietro Balatti, Mattia Leonori, and Arash Ajoudani. A flexible and collaborative approach to robotic box-filling and item sorting. *Robotics and Autonomous Systems*, Under review. Video available at: `https://youtu.be/EoG-xTIaw18`.

[22] Pietro Balatti, Luca Muratore, Luka Peternel, Enrico Mingo Hoffman, Nikos G. Tsagarakis, and Arash Ajoudani. A manipulation framework for debris removal using WALK-MAN humanoid. In *10th International Workshop on Human-Friendly Robotics (HFR)*, 2017. Video available at: `https://youtu.be/65cZ3C4K1EM`.

[23] Peter W. Battaglia and Paul R. Schrater. Humans trade off viewing time and movement duration to improve visuomotor accuracy in a fast reaching task. *Journal of Neuroscience*, 27(26):6984–6994, 2007.

[24] Rainer Bischoff, Ulrich Huggenberger, and Erwin Prassler. KUKA youBot - a mobile manipulator for research and education. In *2011 IEEE International Conference on Robotics and Automation*, pages 1–4. IEEE, 2011.

[25] Joaquin A. Blaya and Hugh Herr. Adaptive control of a variable-impedance ankle-foot orthosis to assist drop-foot gait. *IEEE Transactions on neural systems and rehabilitation engineering*, 12(1):24–31, 2004.

[26] Richard Bloss. Collaborative robots are rapidly providing major improvements in productivity, safety, programing ease, portability and cost while addressing many new applications. *Industrial Robot: An International Journal*, 43(5):463–468, 2016.

[27] William Bluethmann, Robert Ambrose, Myron Diftler, Scott Askew, Eric Huber, Michael Goza, Fredrik Rehnmark, Chris Lovchik, and Darby Magruder. Robonaut: A robot designed to work with humans in space. *Autonomous robots*, 14(2-3):179–197, 2003.

[28] Boeing. Boeing tests augmented reality in the factory. `https://www.boeing.com/features/2018/01/augmented-reality-01-18.page`, 2018.

[29] Jonathan Bohren, Radu Bogdan Rusu, E. Gil Jones, Eitan Marder-Eppstein, Caroline Pantofaru, Melonee Wise, Lorenz Mösenlechner, Wim Meeussen, and Stefan Holzer. Towards autonomous robotic butlers: Lessons learned with the PR2. In *2011 IEEE International Conference on Robotics and Automation*, pages 5568–5575. IEEE, 2011.

[30] Wayne Book and Lonnie Love. Teleoperation telerobotics telepresence. *Handbook of Industrial Robotics*, pages 167–186, 1999.

[31] Christoph Borst, Thomas Wimbock, Florian Schmidt, Matthias Fuchs, Bernhard Brunner, Franziska Zacharias, Paolo Robuffo Giordano, Rainer Konietschke, Wolfgang Sepp, Stefan Fuchs, et al. Rollin' Justin-mobile platform with variable base. In *2009 IEEE International Conference on Robotics and Automation*, pages 1597–1598. IEEE, 2009.

[32] Roger Bostelman, Tsai Hong, and Tommy Chang. Visualization of pallets. In *Intelligent Robots and Computer Vision XXIV: Algorithms, Techniques, and Active Vision*, volume 6384, page 638408. International Society for Optics and Photonics, 2006.

[33] Gülçin Büyüközkan and Fethullah Göçer. Digital supply chain: literature review and a proposed framework for future research. *Computers in Industry*, 97:157–177, 2018.

[34] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2D pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[35] Mattia Castelnovi, Paolo Musso, Antonio Sgorbissa, and Renato Zaccaria. Surveillance robotics: analyzing scenes by colors analysis and clustering. In *Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation. Computational Intelligence in Robotics and Automation for the New Millennium (Cat. No. 03EX694)*, volume 1, pages 229–234. IEEE, 2003.

[36] Manuel G. Catalano, Giorgio Grioli, Edoardo Farnioli, Alessandro Serio, Cristina Piazza, and Antonio Bicchi. Adaptive synergies for the design and control of the pisa/iit softhand. *The International Journal of Robotics Research*, 33(5):768–782, 2014.

[37] Chinsheng Chen, Shen-Ming Lee, and Qiusheng Shen. An analytical model for the container loading problem. *European Journal of Operational Research*, 80(1):68–76, 1995.

[38] Hui Chen and Bir Bhanu. 3D free-form object recognition in range images using local surface patches. *Pattern Recognition Letters*, 28(10):1252–1262, 2007.

[39] Andrea Cherubini, Robin Passama, André Crosnier, Antoine Lasnier, and Philippe Fraisse. Collaborative manufacturing with physical human–robot interaction. *Robotics and Computer-Integrated Manufacturing*, 40:1–13, 2016.

[40] Swagat Chutia, Nayan M. Kakoty, and Dhanapati Deka. A review of underwater robotics, navigation, sensing techniques and applications. In *Proceedings of the Advances in Robotics*. Association for Computing Machinery, 2017.

[41] Andrea S. Ciullo, Manuel G. Catalano, Antonio Bicchi, and Arash Ajoudani. A supernumerary soft robotic hand-arm system for improving worker ergonomics. In *International Symposium on Wearable Robotics*, pages 520–524. Springer, 2018.

[42] Ed Colgate, Witaya Wannasuphoprasit, and Michael Peshkin. Cobots: robots for collaboration with human operators. In *Proceedings of the 1996 ASME International Mechanical Engineering Congress and Exposition*, pages 433–439. ASME, 1996.

[43] Peter Corke. *Robotics, vision and control: fundamental algorithms in MATLAB® second, completely revised*, volume 118. Springer, 2017.

[44] Rita Cucchiara, Massimo Piccardi, and Andrea Prati. Focus based feature extraction for pallets recognition. In *BMVC*, pages 1–10, 2000.

[45] Alessandro De Franco, Edoardo Lamon, Pietro Balatti, Elena De Momi, and Arash Ajoudani. An intuitive augmented reality interface for task scheduling, monitoring, and work performance improvement in human-robot collaboration. In *2019 IEEE International Work Conference on Bioinspired Intelligence (IWOBI)*, pages 75–80. IEEE, 2019. Video available at: `https://youtu.be/4MSIy4TC6zs`.

[46] Alessandro De Luca and Costanzo Manes. Hybrid force-position control for robots in contact with dynamic environments. *IFAC Proceedings Volumes*, 24(9):177–182, 1991.

[47] Jeffrey Delmerico, Stefano Mintchev, Alessandro Giusti, Boris Gromov, Kamilo Melo, Tomislav Horvat, Cesar Cadena, Marco Hutter, Auke Ijspeert, Dario Floreano, et al. The current state and future outlook of rescue robotics. *Journal of Field Robotics*, 36(7):1171–1191, 2019.

[48] Alexander Dietrich, Kristin Bussmann, Florian Petit, Paul Kotyczka, Christian Ott, Boris Lohmann, and Alin Albu-Schäffer. Whole-body impedance control of wheeled mobile manipulators. *Autonomous Robots*, 40(3):505–517, 2016.

[49] Alexander Dietrich, Christian Ott, and Alin Albu-Schäffer. An overview of null space projections for redundant, torque-controlled robots. *The International Journal of Robotics Research*, 34(11):1385–1400, 2015.

[50] Julie Dumora, Franck Geffard, Catherine Bidard, Thibaut Brouillet, and Philippe Fraisse. Experimental study on haptic communication of a human in a shared human-robot collaborative task. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5137–5144. IEEE, 2012.

[51] Gautier Durantin, Scott Heath, and Janet Wiles. Social moments: a perspective on interaction for social robotics. *Frontiers in Robotics and AI*, 4:24, 2017.

[52] Johannes Englsberger, Alexander Werner, Christian Ott, Bernd Henze, Maximo A. Roa, Gianluca Garofalo, Robert Burger, Alexander Beyer, Oliver Eiberger, Korbinian Schmid, et al. Overview of the torque-controlled humanoid robot toro. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 916–923. IEEE, 2014.

[53] Mustafa Suphi Erden and Bobby Marić. Assisting manual welding with robot. *Robotics and Computer-Integrated Manufacturing*, 27(4):818–828, 2011.

[54] Felipe Espinosa, Marcelo Salazar, Daniel Pizarro, and Fernando Valdés. Electronics proposal for telerobotics operation of p3-dx units. In *Remote and Telerobotics*. InTech, 2010.

[55] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, pages 226–231, 1996.

[56] Patrik Fager, Martina Calzavara, and Fabio Sgarbossa. Kit preparation with cobot-supported sorting in mixed model assembly. *IFAC-PapersOnLine*, 52(13):1878–1883, 2019.

[57] Hongchao Fang, Soh Khim Ong, and Andrew Y. C. Nee. Interactive robot trajectory planning and simulation using augmented reality. *Robotics and Computer-Integrated Manufacturing*, 28(2):227–237, 2012.

[58] Federica Ferraguti, Cristian Secchi, and Cesare Fantuzzi. A tank-based approach to impedance control with variable stiffness. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4948–4953, 2013.

[59] Fanny Ficuciello, Guglielmo Tamburrini, Alberto Arezzo, Luigi Villani, and Bruno Siciliano. Autonomy in surgical robots and its meaningful human control. *Paladyn, Journal of Behavioral Robotics*, 10(1):30–43, 2019.

[60] Fanny Ficuciello, Luigi Villani, and Bruno Siciliano. Variable impedance control of redundant manipulators for intuitive human–robot physical interaction. *IEEE Transactions on Robotics*, 31(4):850–863, 2015.

[61] Martin A. Fischler and Robert C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Comm. ACM*, 24(6):381–395, 1981.

[62] Paul M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology*, 47(6):381, 1954.

[63] Fabrizio Flacco, Torsten Kröger, Alessandro De Luca, and Oussama Khatib. A depth space approach to human-robot collision avoidance. In *2012 IEEE International Conference on Robotics and Automation*, pages 338–345. IEEE, 2012.

[64] Atsushi Fujimori, Peter N. Nikiforuk, and Madan M. Gupta. Adaptive navigation of mobile robots with obstacle avoidance. *IEEE Transactions on Robotics and Automation*, 13(4):596–601, 1997.

[65] Markus Funk. Augmented reality at the workplace: a context-aware assistive system using in-situ projection. *Dissertation*, 2016.

[66] Yang Gao and Steve Chien. Review on space robotics: Toward top-level science through space exploration. *Science Robotics*, 2(7), 2017.

[67] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Rafael Medina-Carnicer. Generation of fiducial marker dictionaries using mixed integer linear programming. *Pattern Recognition*, 51:481–491, 2016.

[68] Daniel Goldreich and Ingrid M. Kanics. Tactile acuity is enhanced in blindness. *Journal of Neuroscience*, 23(8):3439–3445, 2003.

[69] José Fernando Gonçalves and Mauricio G. C. Resende. A biased random key genetic algorithm for 2D and 3D bin packing problems. *International Journal of Production Economics*, 145(2):500–510, 2013.

[70] Birgit Graf and Oliver Barth. Entertainment robotics: Examples, key technologies and perspectives. *Safety*, 6(7):8–12, 2002.

[71] Scott A. Green, Mark Billinghurst, XiaoQi Chen, and J. Geoffrey Chase. Human-robot collaboration: A literature review and augmented reality approach in design. *International journal of advanced robotic systems*, 5, 2008.

[72] Elena Gribovskaya, Abderrahmane Kheddar, and Aude Billard. Motion learning and adaptive impedance for robot control during physical interaction with humans. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4326–4332, 2011.

[73] Luca Gualtieri, Erwin Rauch, and Renato Vidoni. Emerging research fields in safety and ergonomics in industrial collaborative robotics: A systematic literature review. *Robotics and Computer-Integrated Manufacturing*, 67:101998, 2020.

[74] Sami Haddadin, Alessandro De Luca, and Alin Albu-Schäffer. Robot collisions: A survey on detection, isolation, and identification. *IEEE Transactions on Robotics*, 33(6):1292–1312, 2017.

[75] Mohammad Hossein Hamedani, Maryam Zekri, Farid Sheikholeslam, Mario Salvaggio, Fanny Ficuciello, and Bruno Siciliano. Recurrent fuzzy wavelet neural network variable impedance control of robotic manipulators with fuzzy gain dynamic surface in an unknown varied environment. *Fuzzy Sets and Systems*, 2020.

[76] Maaike Harbers, Marieke M. M. Peeters, and Mark A. Neerincx. Perceived autonomy of robots: Effects of appearance and context. In *A World with Robots*, pages 19–33. Springer, 2017.

[77] Wei He and Yiting Dong. Adaptive fuzzy neural network control for a constrained robot using impedance learning. *IEEE Transactions on Neural Networks and Learning Systems*, 29(4):1174–1186, 2018.

[78] Abdelfetah Hentout, Mohamed Riad Benbouali, Isma Akli, Brahim Bouzouia, and Lamia Melkou. A telerobotic human/robot interface for mobile manipulators: A study of human operator performance. In *2013 International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 641–646. IEEE, 2013.

[79] Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *2011 international conference on computer vision*, pages 858–865. IEEE, 2011.

[80] Elif Hocaoglu and Volkan Patoglu. Tele-impedance control of a variable stiffness prosthetic hand. In *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1576–1582. IEEE, 2012.

[81] Neville Hogan. Impedance control: An approach to manipulation: part I — Theory. *ASME Journal of Dynamic Systems, Measurement, and Control*, 107:1–7, 1985.

[82] Neville Hogan. Impedance control: An approach to manipulation: part II — Implementation. *ASME Journal of Dynamic Systems, Measurement, and Control*, 107:8–16, 1985.

[83] Neville Hogan. Impedance control: An approach to manipulation: part III — Applications. *ASME Journal of Dynamic Systems, Measurement, and Control*, 107:17–24, 1985.

[84] George Q. Huang, Michael Z. Q. Chen, and Jia Pan. Robotics in ecommerce logistics. *HKIE Transactions*, 22(2):68–77, 2015.

[85] Thomas Hulin, Katharina Hertkorn, Philipp Kremer, Simon Schätzle, Jordi Artigas, Mikel Sagardia, Franziska Zacharias, and Carsten Preusche. The DLR bimanual haptic device with optimized workspace. In *2011 IEEE International Conference on Robotics and Automation*, pages 3441–3442. IEEE, 2011.

[86] Gaël Humbert, Minh Tu Pham, Xavier Brun, Mady Guillemot, and Didier Noterman. Comparative analysis of pick & place strategies for a multi-robot application. In *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*, pages 1–8. IEEE, 2015.

[87] Irfan Hussain, Gionata Salvietti, Giovanni Spagnoletti, Monica Malvezzi, David Cioncoloni, Simone Rossi, and Domenico Prattichizzo. A soft supernumerary robotic finger and mobile arm support for grasping compensation and hemiparetic upper limb rehabilitation. *Robotics and Autonomous Systems*, 93:1–12, 2017.

[88] Shahid Hussain, Sheng Q. Xie, and Prashant K. Jamwal. Adaptive impedance control of a robotic orthosis for gait rehabilitation. *IEEE transactions on cybernetics*, 43(3):1025–1034, 2013.

[89] Hiroyasu Iwata and Shigeki Sugano. Design of human symbiotic robot TWENDY-ONE. In *2009 IEEE International Conference on Robotics and Automation*, pages 580–586. IEEE, 2009.

[90] Advait Jain and Charles C. Kemp. EL-E: an assistive mobile manipulator that autonomously fetches objects from flat surfaces. *Autonomous Robots*, 28(1):45, 2010.

[91] Doyoung Jeon and Masayoshi Tomizuka. Learning hybrid force and position control of robot manipulators. *IEEE Transactions on Robotics and Automation*, 9(4):423–431, 1993.

[92] Sung Eun Jung and Eun-Sok Won. Systematic review of research trends in robotics education for young children. *Sustainability*, 10(4):905, 2018.

[93] Kenji Kaneko, Fumio Kanehiro, Mitsuharu Morisawa, Kazuhiko Akachi, Go Miyamori, Atsushi Hayashi, and Noriyuki Kanehira. Humanoid robot hrp-4-humanoid robotics platform with lightweight and slim body. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4400–4407. IEEE, 2011.

[94] Dimitrios Kanoulas, Jinoh Lee, Darwin G. Caldwell, and Nikos G. Tsagarakis. Center-of-mass-based grasp pose adaptation using 3D range and force/torque sensing. *International Journal of Humanoid Robotics*, 15(04):1850013, 2018.

[95] Navvab Kashiri, Lorenzo Baccelliere, Luca Muratore, Arturo Laurenzi, Zeyu Ren, Enrico Mingo Hoffman, Malgorzata Kamedula, Giuseppe Francesco Rigano, Jorn Malzahn, Stefano Cordasco, et al. Centauro: A hybrid locomotion and high power resilient manipulation platform. *IEEE Robotics and Automation Letters*, 4(2):1595–1602, 2019.

[96] Dov Katz, Yuri Pyuro, and Oliver Brock. Learning to manipulate articulated objects in unstructured environments using a grounded relational representation. *Robotics: Science and Systems IV*, page 254, 2009.

[97] Dov Katz, Arun Venkatraman, Moslem Kazemi, J Andrew Bagnell, and Anthony Stentz. Perceiving, learning, and exploiting object affordances for autonomous pile manipulation. *Autonomous Robots*, 37(4):369–382, 2014.

[98] Lydia E. Kavraki, Petr Svestka, Jean-Claude Latombe, and Mark H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, 12(4):566–580, 1996.

[99] Mitsuo Kawato. Internal models for motor control and trajectory planning. *Current opinion in neurobiology*, 9(6):718–727, 1999.

[100] Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.

[101] Oussama Khatib. Inertial properties in robotic manipulation: An object-level framework. *The International Journal of Robotics Research*, 14(1):19–36, 1995.

[102] Kiho Kim, Jangjin Park, Hohee Lee, and Keechan Song. Teleoperated cleaning robots for use in a highly radioactive environment of the dfdf. In *2006 SICE-ICASE International Joint Conference*, pages 3094–3099. IEEE, 2006.

[103] Wansoo Kim, Pietro Balatti, Edoardo Lamon, and Arash Ajoudani. MOCA-MAN: A mobile and reconfigurable collaborative robot assistant for conjoined human-robot actions. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10191–10197. IEEE, 2020. Video available at: `https://youtu.be/Zg1NzO3hHSY`.

[104] Wansoo Kim, Heedon Lee, Donghwan Kim, Jungsoo Han, and Changsoo Han. Mechanical design of the hanyang exoskeleton assistive robot (hexar). In *2014 14th international conference on control, automation and systems (ICCAS 2014)*, pages 479–484. IEEE, 2014.

[105] Wansoo Kim, Jinoh Lee, Nikos Tsagarakis, and Arash Ajoudani. A real-time and reduced-complexity approach to the detection and monitoring of static joint overloading in humans. In *Rehabilitation Robotics (ICORR), 2017 International Conference on*, pages 828–834. IEEE, 2017.

[106] Wansoo Kim, Marta Lorenzini, Pietro Balatti, Phuong D. H. Nguyen, Ugo Pattacini, Vadim Tikhanoff, Luka Peternel, Claudio Fantacci, Lorenzo Natale, Giorgio Metta, et al. Adaptable workstations for human-robot collaboration: A reconfigurable framework for improving worker ergonomics and productivity. *IEEE Robotics & Automation Magazine*, 26(3):14–26, 2019.

[107] Wansoo Kim, Marta Lorenzini, Pietro Balatti, Yuqiang Wu, and Arash Ajoudani. Towards ergonomic control of collaborative effort in multi-human mobile-robot teams. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[108] Tobias Klamt, Diego Rodriguez, Max Schwarz, Christian Lenz, Dmytro Pavlichenko, David Droeschel, and Sven Behnke. Supervised autonomous locomotion and manipulation for disaster response with a centaur-like robot. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018.

[109] Iuliia Kotseruba and John K. Tsotsos. 40 years of cognitive architectures: core cognitive abilities and practical applications. *Artificial Intelligence Review*, 53(1):17–94, 2020.

[110] Hermano Igo Krebs, Jerome Joseph Palazzolo, Laura Dipietro, Mark Ferraro, Jennifer Krol, Keren Rannekleiv, Bruce T. Volpe, and Neville Hogan. Rehabilitation robotics: Performance-based progressive robot-assisted therapy. *Autonomous robots*, 15(1):7–20, 2003.

[111] Klas Kronander and Aude Billard. Learning compliant manipulation through kinesthetic and tactile human-robot interaction. *IEEE Transactions on Haptics*, 7(3):367–380, 2014.

[112] Robert Krug, Todor Stoyanov, Vinicio Tincani, Henrik Andreasson, Rafael Mosberger, Gualtiero Fantoni, and Achim J Lilienthal. The next step in robot commissioning: Autonomous picking and palletizing. *IEEE Robotics and Automation Letters*, 1(1):546–553, 2016.

[113] Jörg Krüger and Dragoljub Surdilovic. Robust control of force-coupled human–robot-interaction in assembly processes. *CIRP annals*, 57(1):41–44, 2008.

[114] Dennis Krupke, Frank Steinicke, Paul Lubos, Yannick Jonetzko, Michael Görner, and Jianwei Zhang. Comparison of multimodal heading and pointing gestures for co-located mixed reality human-robot interaction. In *2018 IEEE/RSJ International Conf. on Intelligent Robots and Systems (IROS)*, pages 1–9, 2018.

[115] Naveen Kumar, Vikas Panwar, Nagarajan Sukavanam, Shri Prakash Sharma, and Jin-Hwan Borm. Neural network based hybrid force/position control for robot manipulators. *International Journal of Precision Engineering and Manufacturing*, 12(3):419–426, 2011.

[116] Voemir Kunchev, Lakhmi Jain, Vladimir Ivancevic, and Anthony Finn. Path planning and obstacle avoidance for autonomous mobile robots: A review. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 537–544. Springer, 2006.

[117] Chiman Kwan. Hybrid force/position control for manipulators with motor dynamics using a sliding-adaptive approach. *IEEE Transactions on Automatic Control*, 40(5):963–968, 1995.

[118] Marco Laghi, Michele Maimeri, Mathieu Marchand, Clara Leparoux, Manuel G. Catalano, Arash Ajoudani, and Antonio Bicchi. Shared-autonomy control for intuitive bimanual tele-manipulation. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2018.

[119] Edoardo Lamon, Mattia Leonori, Wansoo Kim, and Arash Ajoudani. Towards an intelligent collaborative robotic system for mixed case palletizing. In *2020 International Conference on Robotics and Automation (ICRA)*, pages 9128–9134. IEEE, 2020.

[120] Kurt Landau, Holger Rademacher, Herwig Meschke, Gabriele Winter, Karlheinz Schaub, Marc Grasmueck, Ingo Moelbert, Michael Sommer, and Jens Schulze. Musculoskeletal disorders in assembly jobs in the automotive industry with special reference to age management aspects. *International Journal of Industrial Ergonomics*, 38(7-8):561–576, 2008.

[121] Kirsten F. Laurin-Kovitz, James Edward Colgate, and Steven D. R. Carnes. Design of components for programmable passive impedance. In *ICRA*, volume 2, pages 1476–1481, 1991.

[122] Daniel Lecking, Oliver Wulf, and Bernardo Wagner. Variable pallet pick-up for automatic guided vehicles in industrial environments. In *2006 IEEE Conference on Emerging Technologies and Factory Automation*, pages 1169–1174. IEEE, 2006.

[123] Hee-Don Lee, Byeong-Kyu Lee, Wansoo Kim, Jung-Soo Han, Kyoo-Sik Shin, and Chang-Soo Han. Human–robot cooperation control based on a dynamic model of an upper limb exoskeleton for human power amplification. *Mechatronics*, 24(2):168–176, 2014.

[124] Jinoh Lee, Arash Ajoudani, Enrico Mingo Hoffman, Alessio Rocchi, Alessandro Settimi, Mirko Ferrati, Antonio Bicchi, Nikolaos G. Tsagarakis, and Darwin G. Caldwell. Upper-body impedance control with variable stiffness for a door opening task. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 713–719, 2014.

[125] Kwang-Kyu Lee and Martin Buss. Force tracking impedance control with variable target stiffness. *IFAC Proceedings Volumes*, 41(2):6751–6756, 2008.

[126] Joshua Leven, Darius Burschka, Rajesh Kumar, Gary Zhang, Steve Blumenkranz, Xiangtian Donald Dai, Mike Awad, Gregory D. Hager, Mike Marohn, Mike Choti, et al. DaVinci canvas: a telerobotic surgical system with integrated, robot-assisted, laparoscopic ultrasound capability. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 811–818. Springer, 2005.

[127] Baoguo Li and Chunxi Zhang. Adaptive fuzzy control for mobile robot obstacle avoidance based on virtual line path tracking. In *2006 IEEE International Conference on Robotics and Biomimetics*, pages 1454–1458. IEEE, 2006.

[128] Yanan Li, Shuzhi Sam Ge, and Chenguang Yang. Learning impedance control for physical robot–environment interaction. *International Journal of Control*, 85(2):182–193, 2012.

[129] Kaiqi Liu, Wenguang Wang, and Jun Wang. Pedestrian detection with lidar point clouds based on single template matching. *Electronics*, 8(7):780, 2019.

[130] Andrea Lodi, Silvano Martello, and Daniele Vigo. Heuristic algorithms for the three-dimensional bin packing problem. *European Journal of Operational Research*, 141(2):410–420, 2002.

[131] Sebastian Lohmeier, Thomas Buschmann, and Heinz Ulbrich. Humanoid robot LOLA. In *2009 IEEE International Conference on Robotics and Automation*, pages 775–780. IEEE, 2009.

[132] Sara Lone, Adlen Khelladi, and Shaun Packiarajah. European ecommerce report: 2018 edition. *Brussels: Ecommerce Europe Publications*, 2018.

[133] Rogelio Lozano and Bernard Brogliato. Adaptive hybrid force-position control for redundant manipulators. In *29th IEEE Conference on Decision and Control*, pages 1949–1950. IEEE, 1990.

[134] Matthias Lutz, Christian Verbeek, and Christian Schlegel. Towards a robot fleet for intra-logistic tasks: Combining free robot navigation with multi-robot coordination at bottlenecks. In *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–4. IEEE, 2016.

[135] Silvano Martello, David Pisinger, Daniele Vigo, Edgar Den Boef, and Jan Korst. Algorithm 864: General and robot-packable variants of the three-dimensional bin packing problem. *ACM Transactions on Mathematical Software (TOMS)*, 33(1):7–es, 2007.

[136] Carlos Mateo, Alberto Brunete, Ernesto Gambao, and Miguel Hernando. Hammer: An Android based application for end-user industrial robot programming. In *2014 IEEE/ASME 10th International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, pages 1–6. IEEE, 2014.

[137] Pauline Maurice, Jernej Čamernik, Dasa Gorjan, Benjamin Schirrmeister, Jonas Bornmann, Luca Tagliapietra, Claudia Latella, Daniele Pucci, Lars Fritzsche, Serena Ivaldi, et al. Evaluation of PAEXO, a novel passive exoskeleton for overhead work. *Computer Methods in Biomechanics and Biomedical Engineering*, 22(sup1):S448–S450, 2019.

[138] Trish Melton. The benefits of lean manufacturing: what lean thinking has to offer the process industries. *Chemical engineering research and design*, 83(6):662–673, 2005.

[139] Ady-Daniel Mezei, Levente Tamás, and Lucian Buşoniu. Sorting objects from a conveyor belt using active perception with a POMDP model. In *2019 18th European Control Conference (ECC)*, pages 2466–2471. IEEE, 2019.

[140] Toshio Morita and Shigeki Sugano. Design and development of a new robot joint using a mechanical impedance adjuster. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, volume 3, pages 2469–2475. IEEE, 1995.

[141] Marius Muja and David Lowe. Flann-fast library for approximate nearest neighbors user manual. *Computer Science Department, University of British Columbia, Vancouver, BC, Canada*, 2009.

[142] Bojan Nemec, Nejc Likar, Andrej Gams, and Aleš Ude. Human robot cooperation with compliance adaptation along the motion trajectory. *Autonomous Robots*, 42(5):1023–1035, 2018.

[143] Vinh Nguyen, Quyen Vu, Oksana Solenaya, and Andrey Ronzhin. Analysis of main tasks of precision farming solved with the use of robotic means. In *MATEC Web of Conferences*, volume 113, page 02009. EDP Sciences, 2017.

[144] Allison M. Okamura. Methods for haptic feedback in teleoperated robot-assisted surgery. *Industrial Robot: An International Journal*, 31(6):499–508, 2004.

[145] Allison M. Okamura, Maja J. Matarić, and Henrik I. Christensen. Medical and health-care robotics. *IEEE Robotics & Automation Magazine*, 17(3):26–37, 2010.

[146] Anish Pandey, Shalini Pandey, and Dayal R. Parhi. Mobile robot navigation and obstacle avoidance techniques: A review. *Int Rob Auto J*, 2(3):00022, 2017.

[147] Federico Parietti, Kameron Chan, and H. Harry Asada. Bracing the human body with supernumerary robotic limbs for physical assistance and load reduction. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 141–148. IEEE, 2014.

[148] Federico Parietti, Kameron Chan, Banks Hunter, and H. Harry Asada. Design and control of supernumerary robotic limbs for balance augmentation. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5010–5017. IEEE, 2015.

[149] Jin-Seo Park and Se-Jong Oh. A new concave hull algorithm and concaveness measure for n-dimensional datasets. *Journal of Information science and engineering*, 28(3):587–600, 2012.

[150] Lynne E Parker and John V Draper. Robotics applications in maintenance and repair. *Handbook of industrial robotics*, 2:1023–1036, 1998.

[151] Luigi Penco, Brice Clément, Valerio Modugno, Enrico Mingo Hoffman, Gabriele Nava, Daniele Pucci, Nikos G. Tsagarakis, Jean Baptiste Mouret, and Serena Ivaldi. Robust real-time whole-body motion retargeting from human to humanoid. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pages 425–432. IEEE, 2018.

[152] Luigi Penco, Nicola Scianca, Valerio Modugno, Leonardo Lanari, Giuseppe Oriolo, and Serena Ivaldi. A multimode teleoperation framework for humanoid loco-manipulation: An application for the iCub robot. *IEEE Robotics & Automation Magazine*, 26(4):73–82, 2019.

[153] J. Norberto Pires and Amin S. Azar. Advances in robotics for additive/hybrid manufacturing: robot control, speech interface and path planning. *Industrial Robot: An International Journal*, 2018.

[154] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. ROS: an open-source robot operating system. In *ICRA workshop on open source software*, 2009.

[155] Joao Rebelo, Thomas Sednaoui, Emiel Boudewijn den Exter, Thomas Krueger, and Andre Schiele. Bilateral robot teleoperation: A wearable arm exoskeleton featuring an intuitive user interface. *IEEE Robotics & Automation Magazine*, 21(4):62–69, 2014.

[156] Kyle Reed and Michael A. Peshkin. Physical collaboration of human-human and human-robot teams. *IEEE Transactions on Haptics*, 1(2):108–120, July 2008.

[157] Ludovic Righetti, Mrinal Kalakrishnan, Peter Pastor, Jonathan Binney, Jonathan Kelly, Randolph C. Voorhies, Gaurav S. Sukhatme, and Stefan Schaal. An autonomous manipulation system based on force control and optimization. *Autonomous Robots*, 36(1-2):11–30, 2014.

[158] Francisco J. Romero-Ramirez, Rafael Muñoz-Salinas, and Rafael Medina-Carnicer. Speeded up detection of squared fiducial markers. *Image and vision Computing*, 76:38–47, 2018.

[159] Christoph Rösmann, Frank Hoffmann, and Torsten Bertram. Planning of multiple robot trajectories in distinctive topologies. In *2015 European Conference on Mobile Robots (ECMR)*, pages 1–6. IEEE, 2015.

[160] Christoph Rösmann, Frank Hoffmann, and Torsten Bertram. Integrated online trajectory planning and optimization in distinctive topologies. *Robotics and Autonomous Systems*, 88:142–153, 2017.

[161] Loris Roveda. Adaptive interaction controller for compliant robot base applications. *IEEE Access*, 7:6553–6561, 2018.

[162] Loris Roveda, Niccolò Iannacci, and Lorenzo Molinari Tosatti. Discrete-time formulation for optimal impact control in interaction tasks. *Journal of Intelligent & Robotic Systems*, 90(3-4):407–417, 2018.

[163] Emanuele Ruffaldi, Filippo Brizzi, Franco Tecchia, and Sandro Bacinelli. Third point of view augmented reality for robot intentions visualization. In *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*, pages 471–478. Springer, 2016.

[164] Radu Bogdan Rusu. *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD thesis, Computer Science department, TUM, Germany, October 2009.

[165] Yoshiaki Sakagami, Ryujin Watanabe, Chiaki Aoyama, Shinichi Matsunaga, Nobuo Higaki, and Kikuo Fujimura. The intelligent ASIMO: System overview and integration. In *IEEE/RSJ international conference on intelligent robots and systems*, volume 3, pages 2478–2483. IEEE, 2002.

[166] Christopher Schindlbeck and Sami Haddadin. Unified passivity-based cartesian force/impedance control for rigid and flexible joint robots via task-energy tanks. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 440–447, 2015.

[167] Michael Seelinger and John-David Yoder. Automatic visual guidance of a forklift engaging a pallet. *Robotics and Autonomous Systems*, 54(12):1026–1038, 2006.

[168] Jane Shi, Glenn Jimmerson, Tom Pearson, and Roland Menassa. Levels of human and robot collaboration for automotive manufacturing. In *Proceedings of the Workshop on Performance Metrics for Intelligent Systems*, pages 95–100. ACM, 2012.

[169] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.

[170] Roland Siegwart, Illah Reza Nourbakhsh, and Davide Scaramuzza. *Introduction to autonomous mobile robots*. MIT press, 2011.

[171] Peng Song, Yueqing Yu, and Xuping Zhang. A tutorial survey and comparison of impedance control on robotic manipulation. *Robotica*, 37(5):801–836, 2019.

[172] Alessandro Spada, Marco Cognetti, and Alessandro De Luca. Locomotion and telepresence in virtual and real worlds. In *Human Friendly Robotics*, pages 85–98. Springer, 2019.

[173] Matthew Spenko, Haoyong Yu, and Steven Dubowsky. Robotic personal aids for mobility and monitoring for the elderly. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(3):344–351, 2006.

[174] Siddhartha S. Srinivasa, Dmitry Berenson, Maya Cakmak, Alvaro Collet, Mehmet R. Dogar, Anca D. Dragan, Ross A. Knepper, Tim Niemueller, Kyle Strabala, Mike Vande Weghe, et al. Herb 2.0: Lessons learned from developing a mobile manipulator for the home. *Proceedings of the IEEE*, 100(8):2410–2428, 2012.

[175] Mike Stilman, Jon Olson, and William Gloss. Golem krang: Dynamically stable humanoid robot for mobile manipulation. In *2010 IEEE International Conference on Robotics and Automation*, pages 3304–3309. IEEE, 2010.

[176] Nahema Sylla, Vincent Bonnet, Frédéric Colledani, and Philippe Fraisse. Ergonomic contribution of able exoskeleton in automotive industry. *International Journal of Industrial Ergonomics*, 44(4):475–481, 2014.

[177] Tua Agustinus Tamba, Bonghee Hong, and Keum-Shik Hong. A path following control of an unmanned autonomous forklift. *International Journal of Control, Automation and Systems*, 7(1):113–122, 2009.

[178] Seth Teller, Matthew R. Walter, Matthew Antone, Andrew Correa, Randall Davis, Luke Fletcher, Emilio Frazzoli, Kim Glass, Jonathan P. How, Albert S. Huang, Jeong hwan Jeon, Sertac Karaman, Brandon Luders, Nicholas Roy, and Tara Sainath. A voice-commandable robotic forklift working alongside humans in minimally-prepared outdoor environments. In *2010 IEEE International Conference on Robotics and Automation*, pages 526–533, May 2010.

[179] Graziano Terenzi and Giuseppe Basile. Smart maintenance–an augmented reality platform for training and field operations in the manufacturing industry. *ARMEDIA Augmented Reality Blog*, 2014.

[180] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[181] Federico Tombari, Samuele Salti, and Luigi Di Stefano. A combined texture-shape descriptor for enhanced 3D feature matching. In *2011 18th IEEE International Conference on Image Processing*, pages 809–812, 2011.

[182] Nikolaos G. Tsagarakis, Darwin G. Caldwell, Francesca Negrello, Wooseok Choi, Lorenzo Baccelliere, Vo-Gia Loc, Jerryll Noorden, Luca Muratore, Alessio Margan, Alberto Cardellino, et al. Walk-man: A high-performance humanoid platform for realistic environments. *Journal of Field Robotics*, 34(7):1225–1259, 2017.

[183] Nikos G. Tsagarakis, Stephen Morfey, Gustavo Medrano Cerda, Li Zhibin, and Darwin G. Caldwell. Compliant humanoid coman: Optimal joint stiffness tuning for modal frequency control. In *2013 IEEE International Conference on Robotics and Automation*, pages 673–678. IEEE, 2013.

[184] Muhammad Tufail and Clarence W. de Silva. Impedance control schemes for bilateral teleoperation. In *2014 9th International Conference on Computer Science & Education*, pages 44–49. IEEE, 2014.

[185] Ana Lucia Pais Ureche, Keisuke Umezawa, Yoshihiko Nakamura, and Aude Billard. Task parameterization using continuous constraints extracted from human demonstrations. *IEEE Transactions on Robotics*, 31(6):1458–1471, 2015.

[186] Bram Vanderborght, Alin Albu-Schäffer, Antonio Bicchi, Etienne Burdet, Darwin G. Caldwell, Raffaella Carloni, Manuel G. Catalano, Oliver Eiberger, Werner Friedl, Ganesh Ganesh, et al. Variable impedance actuators: A review. *Robotics and autonomous systems*, 61(12):1601–1614, 2013.

[187] Fan Wang and Kris Hauser. Stable bin packing of non-convex 3D objects with a robot manipulator. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8698–8704. IEEE, 2019.

[188] Gerhard Wäscher, Heike Haußner, and Holger Schumann. An improved typology of cutting and packing problems. *European journal of operational research*, 183(3):1109–1130, 2007.

[189] Augie Widyotriatmo and Keum-Shik Hong. Configuration control of an autonomous vehicle under nonholonomic and field-of-view constraints. *International Journal of Imaging and Robotics*, 15(3):126–139, 2015.

[190] Jeffrey D Will, Kevin L Moore, and Ian K Lynn. Optimizing human-robot teleoperation interfaces for mobile manipulators. *Industrial Robot: An International Journal*, 40(2):173–184, 2013.

[191] Yuqiang Wu, Pietro Balatti, Marta Lorenzini, Fei Zhao, Wansoo Kim, and Arash Ajoudani. A teleoperation interface for loco-manipulation control of mobile collaborative robotic assistant. *IEEE Robotics and Automation Letters*, 4(4):3593–3600, 2019. Video available at: `https://youtu.be/kgod5ePdZpI`.

[192] Guozheng Xu, Aiguo Song, and Huijun Li. Adaptive impedance control for upper-limb rehabilitation robot using evolutionary dynamic recurrent fuzzy neural network. *Journal of Intelligent & Robotic Systems*, 62(3-4):501–525, 2011.

[193] Yoshio Yamamoto. *Control and coordination of locomotion and manipulation of a wheeled mobile manipulators*. PhD thesis, Citeseer, 1994.

[194] Chenguang Yang, Gowrishankar Ganesh, Sami Haddadin, Sven Parusel, Alin Albu-Schäffer, and Etienne Burdet. Human-like adaptation of force and impedance in stable and unstable interactions. *IEEE Transactions on Robotics*, 27(5):918–930, 2011.

[195] Guang-Zhong Yang, Bradley J. Nelson, Robin R. Murphy, Howie Choset, Henrik Christensen, Steven H. Collins, Paolo Dario, Ken Goldberg, Koji Ikuta, Neil Jacobstein, et al. Combating covid-19—the role of robotics in managing public health and infectious diseases, 2020.

[196] Bitao Yao, Zude Zhou, Lihui Wang, Wenjun Xu, Quan Liu, and Aiming Liu. Sensorless and adaptive admittance control of industrial robot in physical human–robot interaction. *Robotics and Computer-Integrated Manufacturing*, 51:158–168, 2018.

[197] Xiaoping Yun and Yoshio Yamamoto. Internal dynamics of a wheeled mobile robot. In *Intelligent Robots and Systems' 93, IROS'93. Proceedings of the 1993 IEEE/RSJ International Conference on*, volume 2, pages 1288–1294. IEEE, 1993.

[198] Zheng Zhang, Qing Guo, Juan Chen, and Peijiang Yuan. Collision-free route planning for multiple AGVs in an automated warehouse based on collision classification. *IEEE Access*, 6:26022–26035, 2018.

[199] Zhilu Zhang and Benxian Xiao. The influence of cargo moving and sliding mode control strategy for forklift. *IEEE Access*, 8:16637–16646, 2020.

[200] Carola Zwicker and Gunther Reinhart. Human-robot-collaboration system for a universal packaging cell for heavy electronic consumer goods. In *Enabling Manufacturing Competitiveness and Economic Sustainability*, pages 195–199. Springer, 2014.

APPENDIX $\mathcal{A}$

# Appendix

## A.1 Prioritized weighted whole-body inverse dynamics control

Through the controller presented in Section 2.3.2, it is possible to plan Cartesian forces (and hence trajectories, thanks to the impedance controller) in the end-effector frame, but it is not possible to regulate how the Cartesian task is projected at the joint level. To overcome this shortcoming and to increase the capabilities of the controller, a prioritized weighted whole-body inverse dynamics control algorithm, obtained by solving an optimization problem, is presented. This controller allows to regulate the interaction forces with the environment, i.e. to shape the way the operational wrench is projected into joint torques, ensuring at the same time that the dynamic equations of the motion are satisfied. In this way, the redundancy provided by the 10 DoF of the MOCA robot can be exploited, determining which joints have to be employed more, in some defined operational modes, such as manipulation and locomotion. In manipulation mode, for instance, the arm motion in close-proximity manipulation tasks needs to be favored, by projecting the Cartesian force mainly on the arm joints, and similarly for the locomotion mode, favoring the mobile base mobility when navigating in free spaces.

To achieve these behaviors, the robot joint torques vector $\boldsymbol{\tau} \in \mathbb{R}^n$ is computed according to a prioritized weighted inverse dynamics algorithm. The controller is obtained by solving the problem of finding the torque vector $\boldsymbol{\tau}$ closest to some desired $\boldsymbol{\tau_0}$ equal to:

$$\boldsymbol{\tau_0} = -\boldsymbol{D}_n\dot{\boldsymbol{q}} - \boldsymbol{K}_n(\boldsymbol{q} - \boldsymbol{q}_{d,0}) \tag{A.1}$$

(see also Eq. 2.20), that realizes the operational forces $\boldsymbol{F}$, according to the norm induced by the positive definite weighting matrix $\boldsymbol{W} \in \mathbb{R}^{(n+m)\times(n+m)}$:

$$\begin{aligned} &\min_{\boldsymbol{\tau}} \frac{1}{2} \|\boldsymbol{\tau} - \boldsymbol{\tau_0}\|^2_{\boldsymbol{W}} \\ &\text{subject to} \quad \bar{\boldsymbol{J}}\boldsymbol{\tau} = \boldsymbol{F} \end{aligned} \tag{A.2}$$

where $\|\boldsymbol{\tau} - \boldsymbol{\tau_0}\|^2_{\boldsymbol{W}}$ indicates the weighted norm $(\boldsymbol{\tau} - \boldsymbol{\tau_0})^2\boldsymbol{W}(\boldsymbol{\tau} - \boldsymbol{\tau_0})$ and

$$\bar{\boldsymbol{J}} = (\boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{J}^\top)^{-1}\boldsymbol{J}\boldsymbol{M}^{-1} \tag{A.3}$$

is the *dynamically consistent pseudo-inverse of* $\boldsymbol{J}(\boldsymbol{q})$, and the constraint

$$\bar{\boldsymbol{J}}\boldsymbol{\tau} = \boldsymbol{F} \tag{A.4}$$

is the general relationship between the generalized joint torques and the operational forces [101]. Notice that, the external torque computation $\boldsymbol{\tau}^{ext} = \boldsymbol{J}^\top\boldsymbol{F}^{ext}$, presented in Chapter 2, is just one solution of Eq. A.4. The solution

of the problem introduced in Eq. A.2 is retrieved by means of the Lagrangian multipliers method. Consider the Lagrangian cost function

$$\boldsymbol{\mathcal{L}} = \frac{1}{2}(\boldsymbol{\tau} - \boldsymbol{\tau_0})^\top \boldsymbol{W}(\boldsymbol{\tau} - \boldsymbol{\tau_0}) + \boldsymbol{\lambda}^\top(\boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{J}^\top\boldsymbol{F} - \boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{\tau}), \tag{A.5}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^6$ is the vector of unknown multipliers of the constraint. The solution requires to satisfy the following:

$$\begin{aligned}\left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{\tau}}\right)^\top &= \boldsymbol{W}(\boldsymbol{\tau} - \boldsymbol{\tau_0}) - \boldsymbol{M}^{-1}\boldsymbol{J}^\top\boldsymbol{\lambda} = \boldsymbol{0}, \\ \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}}\right)^\top &= \boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{J}^\top\boldsymbol{F} - \boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{\tau} = \boldsymbol{0}.\end{aligned} \tag{A.6}$$

The desired optimal solution is obtained by combining the mentioned conditions:

$$\begin{aligned}\boldsymbol{\tau} =& \boldsymbol{W}^{-1}\boldsymbol{M}^{-1}\boldsymbol{J}^\top\boldsymbol{\Lambda}_{\boldsymbol{W}}\boldsymbol{\Lambda}^{-1}\boldsymbol{F} + \\ &+ (\boldsymbol{I} - \boldsymbol{W}^{-1}\boldsymbol{M}^{-1}\boldsymbol{J}^\top\boldsymbol{\Lambda}_{\boldsymbol{W}}\boldsymbol{J}\boldsymbol{M}^{-1})\boldsymbol{\tau_0},\end{aligned} \tag{A.7}$$

where

$$\boldsymbol{\Lambda}_{\boldsymbol{W}} = \boldsymbol{J}^{-\top}\boldsymbol{M}\boldsymbol{W}\boldsymbol{M}\boldsymbol{J}^{-1} \tag{A.8}$$

can be regarded as the *weighted Cartesian inertia*, analogous to the Cartesian inertia in $\boldsymbol{\Lambda} = \boldsymbol{J}^{-\top}\boldsymbol{M}\boldsymbol{J}^{-1}$ (see Eq. 2.7). In Eq. A.7, we can distinguish the two tasks in the controller, the Cartesian force task $\boldsymbol{F}$ with higher priority, and the joint-space torque $\boldsymbol{\tau_0}$ projected in the null space of the first task through a *dynamically consistent null-space projector*, analogously to Eqs. 2.19, 2.20, and 2.21. The formulation in Eq. A.7 contains the prioritized tasks, but the input torques of the controller needs also to compensate for the other terms that are not present in the formulation of the problem in Eq. A.2, like gravity and Coriolis terms, such as in Eq 2.19. Moreover, $\boldsymbol{F}$ and $\boldsymbol{\tau_0}$ can be computed according to Eqs. 2.5 and A.1 respectively.

Only the choice of $\boldsymbol{W}$ is left. The structure of the matrix can be defined as

$$\boldsymbol{W}(\boldsymbol{q}) = \boldsymbol{H}^\top\boldsymbol{M}^{-1}(\boldsymbol{q})\boldsymbol{H} \tag{A.9}$$

where $\boldsymbol{H} \in \mathbb{R}^{(n+m)\times(n+m)}$ is the tunable positive definite weight matrix of the controller. In particular, in this work, $\boldsymbol{H}$ is diagonal and dynamically selected depending on the task. A possible choice is represented by

$$\boldsymbol{H} = \begin{bmatrix} \eta_B\boldsymbol{I}_{m\times m} & \boldsymbol{0}_{m\times n} \\ \boldsymbol{0}_{n\times m} & \eta_A\boldsymbol{I}_{n\times n} \end{bmatrix} \tag{A.10}$$

where $m$ and $n$ are the dimensions of the joint space of the base and the arm, respectively, and $\eta_B$, $\eta_A > 0$ constant scalar values. For instance, to obtain higher mobility of the arm with respect to the base, we set $\eta_B > \eta_A$.

Noteworthy, if $\boldsymbol{H} = \boldsymbol{I}$, then $\boldsymbol{W} = \boldsymbol{M}^{-1}$ and $\boldsymbol{\Lambda}_{\boldsymbol{W}} = \boldsymbol{\Lambda}$. These simplifications leads to the well-known [100]:

$$\boldsymbol{\tau} = \boldsymbol{J}^\top\boldsymbol{F} + (\boldsymbol{I} - \boldsymbol{J}^\top\boldsymbol{\Lambda}\boldsymbol{J}\boldsymbol{M}^{-1})\boldsymbol{\tau_0}. \tag{A.11}$$

This means that the whole-body impedance controller presented in Section 2.3.1 is equal to the solution of the problem in Eq. A.2, if weighting the norm with the inverse of the joint-space inertia matrix $\boldsymbol{M}$.

## A.2 Planning collision-free paths for holonomic robots

Typically, in cluttered environments such as warehouses, simple point-to-point direct motions are not suitable for autonomous mobile robots, due to the possible collision with dynamic obstacles. Human workers, machinery, conveyor belts, and other obstacles must be taken into account in planning the robot mobility. Therefore, the MOCA robot was equipped with an obstacle avoidance algorithm capable of preventing collisions with fixed and moving obstacles. The algorithm exploits a global planner along with the TEB local planner [159], without the need for a map initialization. Given a target location, the planner generates velocity commands for the mobile base, allowing it to navigate through the unknown area. This is obtained by updating a cost map which fuses the data sampled by the perception systems, such as front and rear lasers and the front camera, with the odometry estimation. Here we give an overview of the algorithm.

The trajectory optimization is described as the problem of finding the optimal control inputs $\boldsymbol{u}^* \in \mathbb{R}^m$:

$$\boldsymbol{x}^* = argmin\ J(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathcal{X} \tag{A.12}$$

$$\boldsymbol{u}^* = \boldsymbol{k}(\boldsymbol{x}^*), \tag{A.13}$$

where $\boldsymbol{x}$ is the optimization variable, $J(\boldsymbol{x})$ defines a nonlinear cost function, and $\boldsymbol{k} : \mathcal{X} \rightarrow \mathbb{R}^m$ denotes the mapping from $\boldsymbol{x}$ to the controls $\boldsymbol{u}$. Explicit definition of $J(\boldsymbol{x})$ can be found in [160].

In the case of mobile robot navigation, $J(\boldsymbol{x})$ is computed in order to minimize the trajectory in terms of time or distance between current and target pose. Additional terms are added to ensure a minimal distance from the obstacles, where the maximum is set at the center of the obstacle. The planning algorithm presented in [159] computes planar trajectories and it does not make assumptions on the obstacles shape or representation. A trajectory, in terms of subsequent robot positions, is defined as $s = \{\boldsymbol{z}_k \in \mathbb{R}^2 \,|\, k = 1, 2, \ldots, N\}$.

An exploration graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is constructed for grouping an initial subset of admissible paths given the robot current $\boldsymbol{z}_s$ and goal $\boldsymbol{z}_g$ positions, and a set of obstacle regions $\mathcal{O} = \{\mathcal{O}_l \,|\, l = 1, 2, \ldots, R\}$ with the set of vertices defined as $\mathcal{V} = \{\boldsymbol{z}_s, \boldsymbol{\zeta}_i, \boldsymbol{z}_g \in \mathbb{C} \,|\, \forall \boldsymbol{\zeta}_i \notin \mathcal{O},\, i = 1, 2, \ldots, I\}$ and $\boldsymbol{\zeta}_i \in \mathbb{C}$ the waypoint samples. Due to the arbitrary shape of the obstacles, the waypoint sampling is computed with the probabilistic roadmap method [98].

Then, the set of edges $\mathcal{E}$ is constructed from the sampling of the waypoints. Each edge connects a pair of vertices $v_1,\, v_2 \in \mathcal{V}$ if:

- the direction with respect to the goal heading is forward oriented, i.e. $\dfrac{\Re[\overline{(v_2 - v_1)}(z_g - z_s)]}{\mid v_2 - v_1 \mid\mid z_g - z_s \mid} > \theta,\ \theta \in [0, 1]$;
- the line segment $\mathcal{L} = \{v_1 + t(v_2 - v_1) \,|\, \forall t \in [0, 1]\}$ does not intersect any obstacle $\mathcal{O}_l$, i.e. $\mathcal{L} \cap \mathcal{O} = \emptyset$.

The paths between $\boldsymbol{z}_s$ and $\boldsymbol{z}_g$ are extracted from the generated graph $\mathcal{G}$ by a depth-first search augmented by a visited list.

Finally, the trajectory has to be optimized within the robot control feedback loop. At the beginning, the set of admissible trajectories $T$ is empty. Hence, a new graph is created by seeding random samples in a region of interest. Thanks to the modified depth first search, the initialization of a single representative trajectory is computed for each class in $T$. The globally optimal trajectory $\boldsymbol{x}^*$ is selected according to Eq. A.12, and then the sub-optimal control $\boldsymbol{u}^*$ is calculated. The current start and goal poses, $\boldsymbol{z}_0$ and $\boldsymbol{z}_g$ respectively, are updated considering the actual robot state and perceptions.