POLITECNICO DI MILANO
DIPARTIMENTO DI ELETTRONICA, INFORMAZIONE E BIOINGEGNERIA
DOCTORAL PROGRAMME IN INFORMATION TECHNOLOGY

A GENERAL FRAMEWORK FOR SHARED CONTROL IN ROBOT TELEOPERATION WITH FORCE AND VISUAL FEEDBACK

Doctoral Dissertation of:
Davide Nicolis

Supervisor:
Prof. Paolo Rocco

Tutor:
Prof. Luca Bascetta

The Chair of the Doctoral Program:
Prof. Barbara Pernici

Year 2018 – Cycle XXXI

Abstract

SINCE the '50s, robot teleoperation has been employed in a variety of applications where a human user is required to operate from a distance a robotic device, often a robot manipulator. The use of telerobotics is often motivated by the inaccessibility of the environment where the task must be performed, caused by hostile conditions, such as on-orbit maintenance, and decommissioning of hazardous materials, or simply by the different scale of the workspace in robot-assisted surgery.

Currently, the topic of interaction between user and robotic devices has been receiving increasing attention from the research community and the industry. As teleoperation applications and platforms grow more complex, the employed control framework should be able to relieve the user of some of the burden caused by operating such devices, establishing a sort of shared control.

This work aims at proposing a comprehensive control framework for teleoperation systems comprising robot manipulators. At a local lower control level, sliding mode control theory is employed to achieve a prescribed system behavior, by robustly shaping master and slave manipulators impedances irrespective of uncertainties. An outer hierarchical optimization layer considers control and motion constraints. To help and guide the operator, the specification of hard and soft virtual fixtures is tackled at this level, with virtual force feedback rendered through the analysis of the dual solution of the optimization. A stability analysis of the overall control scheme in presence of variable communication delays during contact is performed by relying on small gain theorem and absolute stability require-

ments, which provide clear tuning guidelines for master and slave robot control parameters.

Furthermore, optical feedback by means of visual servoing is integrated and experimentally validated on a teleoperated dual-arm platform. The proposed controller helps the user in navigating cluttered environments and keep a line of sight with its target by completely avoiding occlusions, reducing the operator workload required to complete a reaching task. Finally, machine learning techniques are employed to infer the user intention and predict his/her motion to actively assist in task execution and reduce fatigue.

Contents

CHAPTER 1

Introduction

1.1 Background

Many branches of robotics are currently striving for complete system autonomy. Self-driving cars and fully autonomous drones are pushing the boundaries of robot capabilities, aiming to relieve humans from the burden of performing most mundane and repetitive tasks, but with an ever increasing attention towards more complex and unpredictable scenarios.

A parallel research line in robotics is instead trying to bring together human and robot in a more seamless way, by making users of these technologies an integral part of the design process. One reason behind this kind of approach resides in the very nature of a robotic platform, which is bound to interact with its environment, and thus also ourselves. Another one is instead quite practical, as the purpose of robots is to be of some service to us, we should take great care in considering the behavior and the needs of the human users during the design phase, and "close the loop" around them.

Nonetheless, a critical factor in current robotics technology is that it is presently unable to cope with situations where complex high-level reasoning and/or manipulation is required. On one hand, these settings demand the presence of a human for his/her decision making capabilities. On the other one, these environments are often harsh and dangerous for human presence, or simply hard to reach.

Telerobotics aims to bridge this gap by letting the human user directly in control of a remote slave robot device through a local master mechanism, and complete tasks in difficult environments, while ensuring personnel safety. Indeed, the user presence is often key to the task successful completion, which is likely to fail by employing solely an autonomous robot.

Robot teleoperation has seen its first applications in the ’40s and ’50s in nuclear material handling. In 1954 the electromechanical master-slave manipulator invented by Goertz [1] laid the foundations of modern telerobotics and force reflecting devices, replacing the pure mechanical and hydraulic architectures of the time, which required to be closely coupled. Nowadays, that invention has spread to the most diverse industries and research areas, with noteworthy applications including minimally invasive surgeries, where a remotely controlled robot can minimize the procedure invasiveness and reduce tremors [2], space robotics for on-orbit servicing and planetary exploration [3], as well as search-and-rescue operations [4]. The range of industries interested in this technology is growing to include sterile drug manufacturing [5] and all those fields where access to a potentially hazardous environment is required while keeping a high degree of safety, or simply where a human himself would not be able to operate effectively.

The increasing attention for the application of these systems in such harsh conditions has also seen the interest of institutional organizations, taking part in funding initiatives. This is for example the case of the H2020 project *SMARTsurg* [6], which aims at improving robot-assisted minimally invasive surgery by designing wearable master devices for optimal perception, and extending these techniques to other surgical procedures. In the H2020 RoMaNS project, instead, the focus is on the autonomous handling and sorting of nuclear waste and radiocative material. The highly unstructured and diverse items to be picked has encouraged the research of innovative teleoperation interfaces where user decision and system autonomy are tightly coupled, and control is shared by employing multi-arm systems and visual servoing techniques [7]. Within the *WALK-MAN* project, a humanoid teleoperated robotic platform has been successfully tested in disaster scenarios, highlighting the focus on perception, manipulation and mobility, in order to navigate, map and interact with environments riddled with obstacles after earthquake events. 3D first person stereoscopic visual feedback has also been adopted for better immersion and telepresence [8]. In [9], the group of O. Khatib proposed a robotic avatar for underwater exploration and discovery, with the purpose of substituting current ROV technology by endowing the system with human-like manipulation abilities and sensing via grounded haptic devices. The user input and the self-stabilizing platform control allow gentle interaction for the safe retrieval of fragile artifacts.

In recent years, a large boost to research in telerobotics has been given by the interest of space agencies and companies, to define a framework for remote control applied to on-orbit servicing, and driving of rovers in planetary exploration. Early experiments were first conducted in the '90s with *ROTEX* [10], which employed predictive simulation concepts to compensate for the inherent earth-to-orbit communication delay. Later experiments in the *Kontur* projects focused on the design and performance of teleoperation controllers in presence of real world internet communication links and data loss with the use of a force reflecting joystick from the International Space Station [11]. The most recent results have come out of the *METERON* project, a joint effort of ESA, NASA, and DLR [12]. The experiments shifted from the classical formulation of direct teleoperation to the concept of supervised autonomy, where an astronaut commanded from the ISS a robot on the ground, with the user responsible for the selection of high level actions and the robot in charge of the task autonomous execution. This architecture has proved to be highly effective in presence of very long delays and communication loss, increasing task success rate.

These recent innovations and growing number of applications make telerobotics a topic of primary interest for robotics research.

1.2 Research challenges

Bilateral teleoperation provides the user with a sense of presence at the remote site, closing the loop between slave and master devices with an appropriate kinesthetic or tactile feedback. Historically, research has focused its attention on the design of stable teleoperation controllers, in presence of time delay in the communication between master and slave robots. Indeed, as a teleoperation system is required to interact with highly uncertain environment dynamics on the slave side, and with user dynamics on the master one, the time necessary for communication of information may produce instability issues due to non-passive behavior of the channel.

[13] and [14] give an overview of established control techniques and problems, while [15–17] provide detailed reviews and comparisons of the controllers proposed in the literature. The grounds for modern teleoperation controllers stability analysis and design were given by Anderson and Spong [18] and Niemeyer and Slotine [19] with the scattering and wave variables

approaches, that are able to ensure overall system passivity by modeling the communication as a transmission line. More recent results have however shifted the paradigm towards less conservative techniques such as time-domain passivity and energy tanks.

Other analyses on force reflecting systems applied to interaction with virtual environments showed a trade-off between the maximum displayable impedance at master side and system stability, mainly due to sampling and discretization effects [20]. In [21] these notions were extended to any pair of impedance or admittance devices and virtual environments, giving also guidelines for the stabilization of such systems via the introduction of a virtual coupler.

The main challenge in telerobotics remains this trade-off between system stability and transparency. To guarantee the practical usability of the system, some feeling of presence at the remote location has to be sacrificed, inevitably degrading the user perception of the remote environment. This compromise has been analyzed in detail with the formalism proposed by Lawrence [22].

In these settings, the control of master and slave impedance remains a key point in system analysis, especially when physical interactions are an integral part of the task at hand. Such applications are not limited to the industrial and construction sectors [23, 24], but have seen widespread use also in whole-body control [25], and the medical field [26], where physicians are required to operate a hands-on robotic device, and impedance selection greatly affects accuracy, physician fatigue, and the delicate contact with patient tissues. In such unstructured environments the robust control of the interaction dynamics are paramount, in order to also ensure the accurate tracking of desired task trajectory profiles and master reference. The problems of impedance control and teleoperation stability have been deeply investigated in [27], which proposed impedance tuning guidelines for the intercontinental control of a humanoid robot subject to delays of hundreds of milliseconds.

Together with kinesthetic and tactile feedback, the development of interfaces and control algorithms for visual feedback has seen an increased appeal to the research community. Especially in delicate scenarios, it is critical to allow the operator to understand what is happening at the remote site with the aid of visual cues, while further exploiting them to improve the system usability. While interfaces such as virtual reality headsets have been employed to increase the sense of immersion, and superimpose information and virtual elements on the visual feed [8], visual servoing has also been considered. In [28] classical visual servoing aids a physician in the execution of a tele-ecography, by exploiting image features to automatically turn the robot tool while optimizing the ultrasound image quality. Image based visual servoing (IBVS) was instead applied to a two slaves system for remote nuclear waste sorting and handling in [7].

Overall, the application of control techniques to visual feedback together with force cues outlines a fertile area for research, which is focusing more and more on the interaction between the user and the control system in the form of shared control architectures. The goal is to split the burden of operation between user and robot, to relieve operators of some the fatigue required to use the system and offer a more intuitive experience. This idea becomes more beneficial as tasks become more difficult, demanding a higher cognitive load, and where physical load is not negligible, such as the handling and installation of heavy materials [29]. Shared human-robot controllers might be the answer for an efficient and effective cooperation in the execution of tasks that the human alone might find hard to perform. Therefore, estimation and prediction of user intention and behavior can further improve the interaction between user and teleoperation system.

Further coverage of the state of the art related to specific topics is covered in the introductions of the chapters of the thesis.

1.3 Thesis contributions

In this scenario, the aim of the present thesis is to propose a complete control framework for the teleoperation of robot manipulators, that tackles the problem both at the local manipulator control level, and in terms of stability of the overall communication loop.

The approach of the thesis leverages the notion of shared human-robot control and autonomy. In particular, this idea is expanded in terms of virtual fixture and virtual force feedback by employing constrained control techniques, and through visual servoing, exploiting the robot autonomy to improve the user visual feedback and work experience, via a simplification of auxiliary camera positioning in difficult environments. Furthermore, preliminary results on machine learning applied to human-robot collaboration are presented, in order to tighten the interdependence between user and robotic system by actively assisting the user through intention inference.

In this regard, this dissertation provides the following contributions

1. A robust controller exploiting sliding mode control techniques is proposed for redundant robot manipulators, to arbitrarily and reliably assign the robot impedance in the task space and track the user input in a teleoperation scenario

2. An optimization-based high level model predictive controller is defined to enforce motion and actuation constraints. This architecture is exploited to allow the straightforward definition of virtual fixtures and their rendering to the operator via force feedback.

3. System stability is tackled in terms of absolute stability criteria, providing insights into the tuning of the overall controller.

4. The framework is extended to dual-arm slaves equipped with cameras. Visual servoing is used to improve user interaction with the system and reduce cognitive load, by introducing some autonomy into the slave robot in terms of camera control.

5. Machine learning in the form of neural networks is exploited to predict user intention and design an assistance controller to help operators during task execution.

All the methods discussed in this thesis are experimentally validated in realistic scenarios employing industrial robots (ABB YuMi prototype and ABB IRB140) and haptic devices (Novint Falcon).

The work is organized as follows.

Chapter 2 starts by addressing the teleoperation of position-controlled slaves and presents a novel controller for bilateral teleoperation based on hierarchical constrained optimization techniques. Motion constraints are exploited for the definition of virtual fixtures, both rigid and compliant, without the need of relative weight tuning. Kinesthetic feedback due to each constraint is directly evaluated from the dual solution of the optimization algorithm. A preliminary transparency and stability analysis in case of communication delays is provided by relying on the classical two-port network formalism and linear control theory.

Chapter 3 shifts the focus on torque-controlled devices and the definition of a robust centralized controller for impedance control and reference tracking of redundant manipulators. The proposed approach takes advantage of the robustness properties of sliding mode control (SMC) and the constraint specification characteristics of model predictive control (MPC). Unmodeled system dynamics and disturbances are compensated to simultaneously ensure accurate tracking and desired end-point impedance in contact with the environment. These features are exploited by the higher level MPC to guarantee constraint fulfillment and the definition of additional tasks, based on the nominal feedback linearized robot model. A detailed analysis of the controllers and their interaction is given also in case of delays acting on the control input.

Chapter 4 adapts the controller of the previous chapter to teleoperation systems for robust impedance shaping. A three-plus-one channel architecture is proposed, with an in-depth analysis of its stability and transparency properties based on Llewellyn's absolute stability theorem. Impedance parameters tuning criteria are derived and the proposed scheme performance is compared with a time-domain passivity approach.

Chapter 5 extends the proposed controller to dual-arm slaves, with one arm equipped with a camera. The problem of visual feedback is tackled, since occlusions can lead to visual servoing failure, and degrade user navigation performance due to the obstructed vision of elements of interest. Occlusion avoidance requirements are formulated inside the optimization as constraints in the image space, with guaranteed robustness against noisy measurements and dynamic environment. Occlusion-free tasks are carried out by autonomously executing evasive camera maneuvers, while keeping the arm teleoperated by the user in the field of view, and ensuring user-friendly movements.

Chapter 6 presents preliminary results on the prediction of user intent in human-robot collaborative tasks. An approach involving a proactive robot behavior that assists in the cooperative execution of trajectories towards desired goals is also proposed. To this end, recurrent neural networks are employed to predict and classify cooperative motions, on the basis of a set of predefined target goals in the workspace and model-based generated data of human movements. The assistance provided by the robot is shown to reduce user fatigue by facilitating the task.

Chapter 7 briefly reviews the thesis contributions and limitations, while also providing suggestions for further developments.

The results and findings of this thesis are based on the following publications and submitted material.

- D. Nicolis, A. M. Zanchettin, P. Rocco, "A Hierarchical Optimization Approach to Robot Teleoperation and Virtual Fixtures Rendering", in *IFAC-PapersOnLine (20th IFAC World Congress)*, vol. 50, no. 1, pp. 5672-5679, July 2017.

- D. Nicolis, M. Palumbo, A. M. Zanchettin and P. Rocco, "Occlusion-Free Visual Servoing for the Shared Autonomy Teleoperation of Dual-Arm Robots", in *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 796-803, April 2018.
- D. Nicolis, A. M. Zanchettin, P. Rocco, "Human Intention Estimation based on Neural Networks for Enhanced Collaboration with Robots", in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2018.
- D. Nicolis, F. Allevi, P. Rocco, "Robust Impedance Shaping of Redundant Teleoperators with Time-Delay via Sliding Mode Control", submitted to the *IEEE Robotics and Automation Letters*, September 2018.

The following publications related to the author's contributions to robot force control are not part of the thesis.

- D. Nicolis, A. M. Zanchettin, P. Rocco, "Constraint-Based and Sensorless Force Control With an Application to a Lightweight Dual-Arm Robot", in *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 340-347, January 2016.
- M. Parigi Polverini, D. Nicolis, A. M. Zanchettin, P. Rocco, "Implicit Robot Force Control Based on Set Invariance", in *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1288-1295, July 2017.
- M. Parigi Polverini, D. Nicolis, A. M. Zanchettin, P. Rocco, "Robust Set Invariance for Implicit Robot Force Control in Presence of Contact Model Uncertainty", *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6393-6399, October 2017.

CHAPTER 2

Hierarchical optimal control for position-controlled teleoperators

In this chapter we focus specifically on teleoperation systems with impedance-type master devices and position-controlled slave robots. By position-controlled, we mean manipulators where a tight position control loop has been already designed to obtain satisfactory reference tracking capabilities. In this case, we can assume that the slave device dynamics are akin to those of a chain of integrators, with the states being the reference joint position and velocity.

A novel controller based on constrained optimization is presented to achieve stability and good performance in a bilateral teleoperation scenario in terms of master tracking error and virtual force feedback. Resorting to a hierarchical formulation, we show how both rigid and compliant virtual fixtures can be defined without the need of unintuitive relative weight tuning. Furthermore, we illustrate how to render haptic feedback by exploiting the dual solution of the optimization problem, which allows the generation of a force feedback even for hard constraints, avoiding unwanted violations of forbidden regions. Through a classical 2-port network representation and tools from linear control theory, a preliminary transparency performance and stability analysis is given in presence of delays in the communication channel between master and slave devices.

In Sec. 2.1 an overview of previous work on virtual fixturing and optimization-based control of manipulators and teleoperators is given. A background on hierarchical constrained optimization is provided in Sec. 2.2, while in Sec. 2.3 and 2.4 the teleoperation controller is presented along with the haptic rendering method based on the optimization dual solution. Transparency and stability properties are then analyzed. Finally, an experimental validation is carried out in Sec. 2.5 in an object tracking application on a system composed of a 7-axes slave robot and a 3 d.o.f. master haptic device.

2.1 Background

Although force feedback generated by the interaction with the real world is substantial in giving a sense of presence, shared control teleoperators often employ *virtual fixtures* to constrain master and/or slave motion [30]. In their most basic form, virtual fixtures are user-defined constraints or workspace regions, whose purpose is to limit the device motion to help the user during task execution, reducing its complexity, as well as the cognitive and physical burden of interacting with the system. Their role is also to enrich the haptic feedback and give the user kinesthetic and tactile information about the task and what is going on at the slave station, which allows task completion in a steadier and quicker way.

In [31] the design and analysis of guidance virtual fixtures (GVF) and forbidden region virtual fixtures (FRVF) is discussed. The former ones usually trace a reference that the user should follow to execute the task, often the operator has the chance to deviate from this nominal behavior, generating a feedback and adapting to task changes that may present themselves only at runtime. On the other hand, the latter ones create regions that the device is not allowed to enter for safety-related requirements, or simply to delimit the robot workspace. Even in this case a force feedback informs the user of the prohibited movements he/she is trying to make. These approaches are often penalty-based and require, to some extent, the penetration of the virtual fixture for the generation of a corresponding force feedback [32]. An exception is given by *reference direction virtual fixtures* combined with master devices of the admittance type: in this case appropriate modulation of the system compliance allows the definition of infinitely stiff fixtures and prevents their violation [33].

The increasing popularity of optimization algorithms for kinematic inversion [34] and reactive trajectory generation [35] for autonomous robots has also seen their application in teleoperation since the mid '90s [36]. The attractive property of these schemes lies in their ability to account for multiple motion constraints, either in the form of equalities or inequalities, and to minimize a cost function of weighted tasks. This is especially convenient in scenarios where redundancies are present due to the task or the robot kinematics, and allows the simultaneous definition of a motion controller and virtual fixtures.

In [37] the authors proposed a model predictive controller to account for communication delays, but did not consider neither the inclusion of virtual fixtures nor an explicit proof of stability of the algorithm. Virtual fixtures primitives for constrained hands-on and remote operation were defined in [38] and later employed in [39] and [40] for endoscopic sinus surgery and knot positioning. In those works the feedback was simply given by the hands-on nature of the application, or computed as a displacement from the desired position, since the master was position-controlled.

A survey by Bowyer et al. [41] on virtual fixtures and active constraints shows how current virtual fixtures haptic rendering methods are independent from the employed teleoperation controller. Methods relying on constraint proximity or on the arbitrary definition of potential fields may result in a possible penetration of the constraint even when it is supposed to be perfectly rigid. Constrained optimization methods, instead, allow the definition of rigid virtual fixtures but lack a way to render haptic feedback for force-controlled impedance-type master devices. Most optimization-based controllers also require the accurate tuning of the cost function weights if both hard and soft virtual fixtures are to be included [42].

2.2 Hierarchical constrained optimization

In the following, the main ideas behind hierarchical kinematic inversion in robotic applications and constrained optimization will be summarized to be later used for the teleoperation controller synthesis. For constrained optimization we will focus on the class of quadratic programming (QP) problems subject to linear constraints, as they can model and solve a wide range of robotic tasks with efficient computational methods. Nonetheless a generalization can be made for nonlinear optimization by employing sequential quadratic programming (SQP) techniques.

Robotic applications often require the fulfillment of multiple tasks defined as equality constraints on the robot motion. When a high number of degrees of freedom is available, a hierarchy can be established between the tasks, and kinematic inversion algorithms can be employed to compute the corresponding joint-space motion based on the task-space requirements priority. Given a set of p prioritized tasks $\boldsymbol{A}_i\dot{\boldsymbol{q}} = \boldsymbol{b}_i$, where $\boldsymbol{A}_i$ represents the i^{th} task Jacobian and $\dot{\boldsymbol{q}}$ the robot joint velocities, the solution of the hierarchical kinematic inversion problem is given recursively by [43]

$$\dot{\boldsymbol{q}}_p = \sum_{i=1}^{p} (\boldsymbol{A}_i \boldsymbol{P}_{i-1})^{\dagger} (\boldsymbol{b}_i - \boldsymbol{A}_i \dot{\boldsymbol{q}}_{i-1}) \tag{2.1}$$

where $\dot{\boldsymbol{q}}_i$ is the optimum solution up to the i^{th} task and $\boldsymbol{P}_i$ the null space projector of matrix $[\boldsymbol{A}_1^T \dots \boldsymbol{A}_i^T]^T$, with $\boldsymbol{P}_0 = \boldsymbol{I}$, $\dot{\boldsymbol{q}}_0 = 0$. While (2.1) provides a closed-form solution to the kinematic inversion, it can only account for equality constraints, that is, only guidance virtual fixtures in a teleoperation formulation. In [7] the authors following this approach had to separately define a potential field to avoid the forbidden areas specified by joint limits in visual-based telemanipulation.

If also linear inequality constraints have to be considered, their priority can be accounted for by formalizing the problem as a cascade of QP optimizations. Here without loss of generality we consider only inequalities, since they can be used to describe also equality task constraints. Given the requirement $\boldsymbol{A}_i\dot{\boldsymbol{q}} \leq \boldsymbol{b}_i$ subject to some constraints $\boldsymbol{A}_{i-1}\dot{\boldsymbol{q}} \leq \boldsymbol{b}_{i-1}$, we can write the following to obtain the optimum joint velocities

$$\{\dot{\boldsymbol{q}}_i, \boldsymbol{w}_i^*\} = \underset{\dot{\boldsymbol{q}}, \boldsymbol{w}_i}{\arg\min} \|\boldsymbol{w}_i\|^2 \tag{2.2a}$$

$$\text{s.t.} \quad \boldsymbol{A}_{i-1}\dot{\boldsymbol{q}} \leq \boldsymbol{b}_{i-1} \tag{2.2b}$$

$$\boldsymbol{A}_i\dot{\boldsymbol{q}} \leq \boldsymbol{b}_i + \boldsymbol{w}_i \tag{2.2c}$$

where $\boldsymbol{w}_i$ is a vector of slack variables, and $\{\dot{\boldsymbol{q}}_i, \boldsymbol{w}_i^*\}$ the solution of the optimization. If a lower priority constraint $\boldsymbol{A}_{i+1}\dot{\boldsymbol{q}} \leq \boldsymbol{b}_{i+1}$ should be considered, the QP problem can be rewritten by using $\boldsymbol{w}_i^*$

$$\{\dot{\boldsymbol{q}}_{i+1}, \boldsymbol{w}_{i+1}^*\} = \underset{\dot{\boldsymbol{q}}, \boldsymbol{w}_{i+1}}{\arg\min} \|\boldsymbol{w}_{i+1}\|^2 \tag{2.3a}$$

$$\text{s.t.} \quad \boldsymbol{A}_{i-1}\dot{\boldsymbol{q}} \leq \boldsymbol{b}_{i-1} \tag{2.3b}$$

$$\boldsymbol{A}_i\dot{\boldsymbol{q}} \leq \boldsymbol{b}_i + \boldsymbol{w}_i^* \tag{2.3c}$$

$$\boldsymbol{A}_{i+1}\dot{\boldsymbol{q}} \leq \boldsymbol{b}_{i+1} + \boldsymbol{w}_{i+1} \tag{2.3d}$$

The solution $\dot{\boldsymbol{q}}_{i+1}$ is optimal also for the optimization problem (2.2) but tries to satisfy (2.3d) in the best possible way. This is ensured by fixing $\boldsymbol{w}_i = \boldsymbol{w}_i^*$, which guarantees the optimality of the higher priority layer, reducing 2.3c to a hard constraint for the lower priority task. Repeating the process for all the p prioritized requirements produces the hierarchical solution like in (2.1), while also potentially taking into consideration inequality constraints (e.g. forbidden region virtual fixtures). It is then clear that, with respect to task (2.3c), (2.3b) represents hard constraints (perfectly rigid virtual fixtures), while (2.3d) defines soft constraints (compliant virtual fixtures) for which a violation is allowed. Note that with this formulation there is no need to tune the weights of the constraints as it has been done in [42] to achieve a similar result, but with the risk of ill-conditioning the problem if their relative magnitude is too large. Although solving a cascaded optimization problem is computationally more intensive than (2.1), especially with an increasing number of variables and constraints, efficient algorithms exist, like the one proposed in [44].

2.3 Teleoperation optimal controller

This section will present the hierarchical controller for bilateral teleoperation. Throughout the discussion we will consider a teleoperation system consisting of a gravity-compensated force-controlled master device of the impedance-type, and a position-controlled slave robot. Furthermore we will assume to be working inside the control bandwidth of the slave position control loop. Under these assumptions the slave robot can be modeled as a system of double integrators, while the master has the following end effector dynamic model

$$\boldsymbol{M}_m\ddot{\boldsymbol{x}}_m + \boldsymbol{D}_m\dot{\boldsymbol{x}}_m = \boldsymbol{F}_h - \boldsymbol{F}_{v,m} \tag{2.4}$$

where $\boldsymbol{M}_m$, $\boldsymbol{D}_m$ and $\boldsymbol{x}_m$ denote the the master inertia, damping, and position, respectively, while $\boldsymbol{F}_h$ and $\boldsymbol{F}_{v,m}$ represent the force applied by the human on the master and the force feedback. In the following subscript s will refer to slave quantities.

To obtain a smooth slave trajectory, the hierarchical controller is defined to optimize a cost function over the slave joint accelerations which work as inputs to the slave robot, while master position and velocity are inputs to the controller. The first objective of the teleoperator is to achieve kinematic coordination between master and slave, in particular we want

$$\boldsymbol{x}_m(t) - \boldsymbol{x}_s(t) \to 0, \quad t \to \infty \tag{2.5}$$

Following the task description used in Sec. 2.2, we have to write (2.5) in the form

$$\boldsymbol{A}\ddot{\boldsymbol{q}}_s = \boldsymbol{b} \tag{2.6}$$

A possible solution is to employ the differential equation

$$-\ddot{\boldsymbol{x}}_s + \boldsymbol{K}_D(\dot{\boldsymbol{x}}_m - \dot{\boldsymbol{x}}_s) + \boldsymbol{K}_P(\boldsymbol{x}_m - \boldsymbol{x}_s) = 0 \tag{2.7}$$

which can be rewritten in terms of the slave Jacobian as

$$\boldsymbol{J}_s\ddot{\boldsymbol{q}}_s = \boldsymbol{K}_D(\dot{\boldsymbol{x}}_m - \dot{\boldsymbol{x}}_s) + \boldsymbol{K}_P(\boldsymbol{x}_m - \boldsymbol{x}_s) - \dot{\boldsymbol{J}}_s\dot{\boldsymbol{q}}_s \tag{2.8}$$

where all the elements on the right hand side are known. $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$ are positive definite diagonal matrices. If hard constraints have to be enforced, such as forbidden region virtual fixtures to avoid unwanted collisions, or simply kinematic limitations, we can plug them and (2.8) in (2.2) to obtain the first stage of the optimization

$$\ddot{\boldsymbol{q}}_{s,1} = \arg\min_{\ddot{\boldsymbol{q}}_s} \|-\ddot{\boldsymbol{x}}_s + \boldsymbol{K}_D(\dot{\boldsymbol{x}}_m - \dot{\boldsymbol{x}}_s) + \boldsymbol{K}_P(\boldsymbol{x}_m - \boldsymbol{x}_s)\|^2_{\boldsymbol{Q}} \tag{2.9a}$$

$$\text{s.t.} \quad \boldsymbol{A}_H\ddot{\boldsymbol{q}}_s \leq \boldsymbol{b}_H \tag{2.9b}$$

The subscript H denotes the hard constraints, while $\boldsymbol{Q}$ is a diagonal matrix of weights that will be useful for haptic rendering. For a list of useful virtual fixture primitives in teleoperation see [38, 39] and [41].

As shown in (2.3), to define a hierarchy with both hard and soft virtual fixtures it is necessary to propagate the optimality conditions from the first stage. For (2.9) this amounts to adding the constraint

$$\boldsymbol{J}_s\ddot{\boldsymbol{q}}_s = \boldsymbol{J}_s\ddot{\boldsymbol{q}}_{s,1} \tag{2.10}$$

with $\ddot{\boldsymbol{q}}_{s,1}$ is the optimum of the first stage. This equation implicitly requires that the second optimization stage provides a solution that is equally optimal with respect to the cost function (2.9a). From the user point of view, this results in the slave tracking the master even if one of the soft constraints is violated, with the user being warned via force feedback as it will be discussed in Sec. 2.4. This kind of approach is useful in order to define a safety region that is accessed before a forbidden one, or to induce a soft guidance in the user motion without affecting position coordination. Denoting with subscript S the soft virtual fixture constraints, the second stage of the optimization can be formalized as

$$\{\ddot{\boldsymbol{q}}_{s,2}, \boldsymbol{w}^*_S\} = \arg\min_{\ddot{\boldsymbol{q}}_s, \boldsymbol{w}_S} \|\boldsymbol{w}_S\|^2_{\boldsymbol{Q}_S} \tag{2.11a}$$

$$\text{s.t.} \quad \boldsymbol{A}_H\ddot{\boldsymbol{q}}_s \leq \boldsymbol{b}_H \tag{2.11b}$$

$$\boldsymbol{J}_s\ddot{\boldsymbol{q}}_s = \boldsymbol{J}_s\ddot{\boldsymbol{q}}_{s,1} \tag{2.11c}$$

$$\boldsymbol{A}_S\ddot{\boldsymbol{q}}_s \leq \boldsymbol{b}_S + \boldsymbol{w}_S \tag{2.11d}$$

$\boldsymbol{Q}_S$ is a positive definite diagonal matrix. The optimum $\ddot{\boldsymbol{q}}_{s,2}$ of the second stage is also the solution of the hierarchical optimization problem, and the reference for the position-controlled slave robot.

The procedure can be summarized as follows: given master and slave positions and velocities, minimize the kinematic coordination error while satisfying the hard virtual fixtures by solving (2.9), then, using the obtained solution, minimize the soft virtual fixtures violation with the remaining degrees of freedom by solving (2.11). The result is the hierarchical optimum.

2.4 Haptic rendering via Lagrange multipliers

Current optimization-based teleoperation controllers often require either the definition of additional structures (e.g. potential fields), or constraint penetration to provide the user with haptic feedback revealing the presence of a virtual fixture. We propose here a method for haptic rendering that exploits the optimization procedure and hierarchical nature of our controller, by relying on the dual solution of the algorithm.

For the first stage of the optimization (2.9), the optimal solution must satisfy the Karush-Kuhn-Tucker (KKT) optimality conditions [45]

$$\left(\boldsymbol{A}_H\ddot{\boldsymbol{q}}_{s,1} - \boldsymbol{b}_H\right)\boldsymbol{\lambda}_{H,1} = 0 \tag{2.12a}$$

$$\boldsymbol{\nabla} f(\ddot{\boldsymbol{q}}_{s,1}) + \boldsymbol{A}_H^T\boldsymbol{\lambda}_{H,1} = 0 \tag{2.12b}$$

where $\boldsymbol{\lambda}_{H,1}$ is the dual optimum associated to the hard constraints for the first stage, and $\boldsymbol{\nabla} f(\ddot{\boldsymbol{q}}_{s,1})$ the gradient of the cost function (2.9a) evaluated in the optimum. Whenever a constraint is active (the equality sign holds, so we are in contact with a virtual fixture), the corresponding Lagrange multiplier λ is different from zero. By writing (2.12b) explicitly we obtain

$$\begin{aligned} &2\boldsymbol{J}_s^T\boldsymbol{Q}\left(-\ddot{\boldsymbol{x}}_s(\ddot{\boldsymbol{q}}_{s,1}) + \boldsymbol{K}_D(\dot{\boldsymbol{x}}_m - \dot{\boldsymbol{x}}_s) + \boldsymbol{K}_P(\boldsymbol{x}_m - \boldsymbol{x}_s)\right) = \\ &= \boldsymbol{A}_H^T\boldsymbol{\lambda}_{H,1} \end{aligned} \tag{2.13}$$

Pre-multiplying for the left Moore-Penrose pseudo-inverse of the slave transpose Jacobian $\boldsymbol{J}_s^T$ and defining

$$\boldsymbol{M}_s = 2\boldsymbol{Q} \tag{2.14a}$$

$$\boldsymbol{D}_s = 2\boldsymbol{Q}\boldsymbol{K}_D \tag{2.14b}$$

$$\boldsymbol{K}_s = 2\boldsymbol{Q}\boldsymbol{K}_P \tag{2.14c}$$

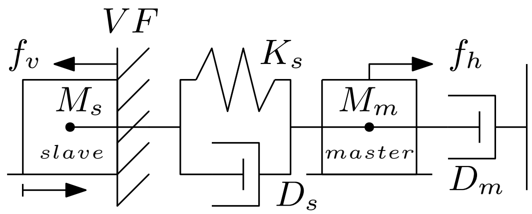


Figure 2.1: *Mechanical representation of the master-slave system*

equation (2.13) becomes

$$\boldsymbol{M}_s \ddot{\boldsymbol{x}}_s(\ddot{\boldsymbol{q}}_{s,1}) = \boldsymbol{F}_s - \boldsymbol{F}_v \tag{2.15a}$$

$$\boldsymbol{F}_s = \boldsymbol{D}_s(\dot{\boldsymbol{x}}_m - \dot{\boldsymbol{x}}_s) + \boldsymbol{K}_s(\boldsymbol{x}_m - \boldsymbol{x}_s) \tag{2.15b}$$

$$\boldsymbol{F}_v = \boldsymbol{J}_s^{T\dagger} \boldsymbol{A}_H^T \boldsymbol{\lambda}_{H,1} \tag{2.15c}$$

Using a Lagrangian mechanics interpretation, (2.15a) represents the dynamic equations of a slave robot with inertia $\boldsymbol{M}_s$ controlled by a PD teleoperator ($\boldsymbol{F}_s$), and in contact with a rigid environment (i.e. the virtual fixtures) producing a reaction force $\boldsymbol{F}_v$. It is worth noticing that with this interpretation the first optimization stage (2.9) basically tries to minimize the slave desired impedance error.

In this case the Lagrange multipliers $\boldsymbol{\lambda}_{H,1}$ represent the intensity of the equivalent reaction torques in the joint space, while the gradients $\boldsymbol{A}_H^T$ their direction in the same space. The contribution of each virtual fixture can be evaluated independently since the corresponding Lagrange multiplier is found while solving the optimization. The reaction forces of interest can be computed to generate the haptic feedback on the master

$$\boldsymbol{F}_{v,m} = \boldsymbol{J}_s^{T\dagger} \sum_i \boldsymbol{a}_{H,i}^T \lambda_{H,1,i} \tag{2.16}$$

where lowercase $\boldsymbol{a}_{H,i}$ refers to the i^{th} row of $\boldsymbol{A}_H$. Figure 2.1 gives a 1 d.o.f. mechanical representation of the teleoperation system given by (2.4), (2.15). Note that, when in free motion, $\boldsymbol{F}_{v,m} = 0$ and the user only feels the master dynamics.

For the second stage (2.11), applying the KKT conditions gives

$$\boldsymbol{A}_S^T \boldsymbol{K}_S \boldsymbol{w}_S^* = -\boldsymbol{A}_H^T \boldsymbol{\lambda}_{H,2} - \boldsymbol{J}_s^T \boldsymbol{\lambda}_{s,2} \tag{2.17}$$

where $\boldsymbol{\lambda}_{H,2}$ and $\boldsymbol{\lambda}_{s,2}$ are the dual optima associated with the constraints for the second stage, and $\boldsymbol{K}_S = 2\boldsymbol{Q}_S$. Assuming that the soft constraint is violated ($\boldsymbol{w}_S^* \neq 0$), at least one of the terms on the right-hand side is different from zero. In particular the first one can be interpreted as the joint torque contribution of the hard virtual fixtures to the soft constraints. In this situation the hard constraints are active and the soft ones are violated simultaneously. The second term, on the other hand, denotes the joint torque component due to the slave tracking the master inside the soft virtual fixtures. The two combined produce a reaction torque pushing the slave away from the soft virtual fixture along its gradient, while the first one alone still pushes the slave out of it, but along the quickest parting direction from the hard constraint. As before, pre-multiplying (2.17) by $\boldsymbol{J}_s^{T\dagger}$ produces the operational space force feedback for the master

$$\boldsymbol{F}_{v,m} = \boldsymbol{J}_s^{T\dagger} \begin{bmatrix} \boldsymbol{A}_H^T & \boldsymbol{J}_s^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_{H,2} \\ \boldsymbol{\lambda}_{s,2} \end{bmatrix} \tag{2.18}$$

Force components due to hard and soft virtual fixtures can then be combined to obtain the final force feedback. Notice that the choice of the diagonal matrix $\boldsymbol{K}_S$ defines the soft virtual fixtures stiffness, we can however avoid the simultaneous use and difficult tuning of very large and very small weights to simulate a rigid surface, since hard constraints are separately defined in the first optimization stage. We also prevent the appearance of an unwanted master-slave coordination error during constraint penetration.

2.4.1 Transparency and stability

To study the transparency and stability properties of the proposed optimization controller [22], we adopt a two-port network model for our teleoperation system. Taking advantage of the previous results, the system equations for master and slave devices are given by

$$\boldsymbol{M}_m \ddot{\boldsymbol{x}}_m + \boldsymbol{D}_m \dot{\boldsymbol{x}}_m = \boldsymbol{F}_h - \boldsymbol{F}_{v,m} = \boldsymbol{F}_h - \boldsymbol{F}_v \tag{2.19}$$

$$\boldsymbol{M}_s \ddot{\boldsymbol{x}}_s = \boldsymbol{D}_s(\dot{\boldsymbol{x}}_m - \dot{\boldsymbol{x}}_s) + \boldsymbol{K}_s(\boldsymbol{x}_m - \boldsymbol{x}_s) - \boldsymbol{F}_v \tag{2.20}$$

For simplicity let's consider the 1 d.o.f. case and define

$$Z_m = M_m s + D_m \tag{2.21a}$$

$$Z_s = M_s s \tag{2.21b}$$

$$C = D_s + \frac{K_s}{s} \tag{2.21c}$$

where Z_m, Z_s and C are the impedances of master and slave, and the coupler transfer function respectively. We can then write the associated two-port network (Fig. 2.2) in terms of the hybrid model

$$\begin{bmatrix} F_h \\ -\dot{x}_s \end{bmatrix} = \begin{bmatrix} Z_m & 1 \\ -\dfrac{C}{Z_s + C} & \dfrac{1}{Z_s + C} \end{bmatrix} \begin{bmatrix} \dot{x}_m \\ F_v \end{bmatrix} \tag{2.22}$$

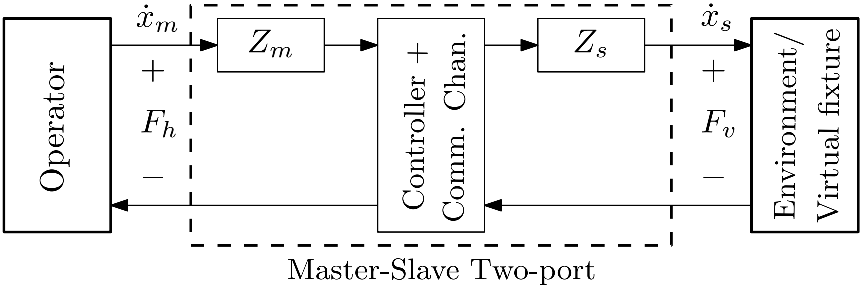


Figure 2.2: *Network block diagram of the teleoperation system*

To achieve perfect transparency, we need

$$Z_m \to 0 \tag{2.23a}$$

$$\frac{C}{Z_s + C} \to 1 \tag{2.23b}$$

$$\frac{1}{Z_s + C} \to 0 \tag{2.23c}$$

In turn, this amounts to having low master and slave impedances and high gains K, D. It is straightforward to notice, however, that $Z_s \neq 0$ since the parameter $M = 2Q$ appears in the first stage optimization cost function and needs to be greater than zero so as to be well-defined.

Given the environment/virtual fixture impedance $Z_e = M_e s + D_e + \frac{K_e}{s}$, the transmitted impedance Z_{th} to the user is

$$Z_{th} = Z_m + \frac{C Z_e}{Z_e + Z_s + C} \tag{2.24}$$

Considering the two limit conditions of free motion ($Z_e = 0$) and rigid environment ($Z_e \to \infty$) we have

$$Z_{th} \to Z_m + C, \quad Z_e \to \infty \tag{2.25a}$$

$$Z_{th} \to Z_m, \quad Z_e \to 0 \tag{2.25b}$$

Again, to display high impedance virtual fixtures we need high controller gains, while in free motion only the master impedance contributes. If the master dynamic model is known, theoretically by applying impedance control we can arbitrarily assign its dynamics Z_m.

To study the system stability, let's consider a delay τ in the communication channel between master and slave. The new hybrid matrix becomes

$$\begin{bmatrix} f_h \\ -\dot{x}_s \end{bmatrix} = \begin{bmatrix} Z_m & e^{-\tau s} \\ -\dfrac{C}{Z_s + C} e^{-\tau s} & \dfrac{1}{Z_s + C} \end{bmatrix} \begin{bmatrix} \dot{x}_m \\ f_v \end{bmatrix} \tag{2.26}$$

Defining the human impedance $Z_h = M_h s + D_h + \frac{K_h}{s}$, the loop transfer function for the whole system is the following one:

$$L(s) = \frac{C Z_e}{(Z_h + Z_m)(Z_e + Z_s + C)} e^{-2\tau s} \tag{2.27}$$

Since all the elements at the denominator of (2.27) are passive, all open loop poles are on the left-hand-side of the complex plane. To find conditions on the control parameters that ensure system stability despite the delay, we can apply the small-gain theorem. This amounts to:

$$|C Z_e| < |Z_h + Z_m||Z_e + Z_s + C|, \ \forall \omega \tag{2.28}$$

In free motion $Z_e = 0$ and the system is in open loop, the slave follows the master without providing any feedback, and thus stability is guaranteed. For $Z_e \to \infty$, (2.28) becomes

$$|C| < |Z_h + Z_m|, \ \forall \omega \tag{2.29}$$

Expanding the computations yields

$$D^2 + \frac{K^2}{\omega^2} < (D_m + D_h)^2 + \left((M_m + M_h)\omega - \frac{K_h}{\omega} \right)^2 \tag{2.30}$$

Since (2.30) has to hold at all frequencies, we must have

$$K < K_h, \quad \omega \to 0 \tag{2.31}$$

That is, at low frequencies the user's grip on the master's handle must be steady and more rigid than the controller proportional action. This however is critical only if the delay is large enough to produce a substantial phase loss at low frequency. As a rule of thumb, from (2.30), selecting

$$D \ll D_m + D_h \tag{2.32}$$

guarantees stability. This can be achieved either by choosing a small derivative action D, thus sacrificing the display of stiff virtual fixtures (2.25a), or by increasing the master damping D_m and give up transparency in free motion (2.25b), since the master will be more sluggish. Passivation via damping injection has also been used in the PD teleoperator proposed in [46].

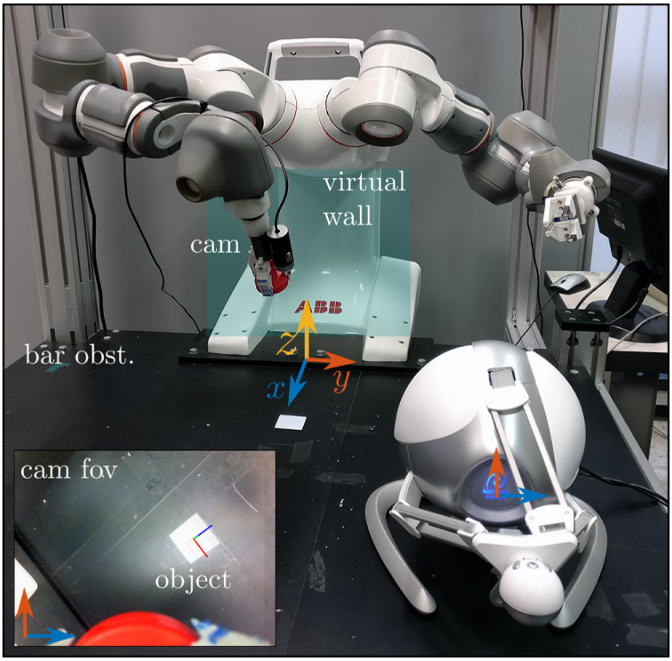


Figure 2.3: *The experimental setup. The camera-equipped robot arm position is controlled by the 3 d.o.f. master device on the bottom right by mapping the joystick movements in the camera frame.*

2.5 Experiments

The controller has been validated by performing an object tracking task with a camera mounted on a teleoperated slave robot. The following section presents the experimental setup and the obtained results.

The setup consists of a slave 7 d.o.f. industrial collaborative robot prototype from ABB endowed with an open research interface, a Microsoft LifeCam webcam mounted on the slave robot in a eye-in-hand configuration, and a 3 d.o.f. Novint Falcon master haptic device. Master and slave robots are connected to two Linux PCs and communicate via Ethernet. The master's controller runs at 1kHz and makes use of Force Dimension haptic libraries, while the slave's one runs at 250Hz. The camera has a $640 * 480\text{px}$ resolution and has been setup with OpenCV to detect a square workpiece in the camera frame.

The task goal is to keep the object inside the teleoperated camera field of view (Fig. 2.3). The user can move freely in the workspace, although subject to both hard and soft virtual fixtures that constrain motion and inform him/her on the task state via kinesthetic feedback. Several hard virtual

Table 2.1: *Object tracking experiment: optimization problem.*

$\min_{\ddot{\boldsymbol{q}}_s} \|-\ddot{\boldsymbol{x}}_s + \boldsymbol{K}_D(\dot{\boldsymbol{x}}_m - \dot{\boldsymbol{x}}_s) + \boldsymbol{K}_P(\boldsymbol{x}_m - \boldsymbol{x}_s)\|^2_{\boldsymbol{Q}}$
$x_{elbow} \geq x_{bar}$ $x_s \geq x_{wall}$ $z_s \geq z_{table}$ $\underline{\boldsymbol{p}} \leq \boldsymbol{p}_o \leq \overline{\boldsymbol{p}}$
$-\ddot{\Phi}_s - \boldsymbol{K}_{D_\Phi}\dot{\Phi}_s + \boldsymbol{K}_{P_\Phi}(\Phi_{ref} - \Phi_s) = 0$

From top to bottom: the kinematic coordination cost function and the hard and soft virtual fixtures

fixtures have been added in the form of collision avoidance constraints. In particular, one of them prevents the robot arm collision with the bar placed on the robot base right side, while two others define a vertical and an horizontal virtual wall to avoid impact of the end-effector with the robot body and the table. An additional hard virtual fixture constrains the object inside the camera field of view. Its point image features $\boldsymbol{p}_o = [u_o \; v_o]^T$ in the camera frame are related to joint variables via the interaction matrix $\boldsymbol{L}$ and the camera Jacobian expressed in the camera frame $\boldsymbol{J}^c_c(\boldsymbol{q}_s)$

$$\dot{\boldsymbol{p}}_o = \boldsymbol{L}\boldsymbol{J}^c_c(\boldsymbol{q}_s)\dot{\boldsymbol{q}}_s \tag{2.33}$$

The virtual fixture to apply is thus the following one:

$$\underline{\boldsymbol{p}} \leq \boldsymbol{p}_o \leq \overline{\boldsymbol{p}} \tag{2.34}$$

where $\underline{\boldsymbol{p}}$, $\overline{\boldsymbol{p}}$ represent the minimum and maximum allowed position of the features in the image plane, respectively. For the implementation we used the robust constraint formulation proposed in [47] that can let us take into account noise and estimation errors of the features position. Since the robot is highly redundant with respect to the master device, motions of the master correspond to Cartesian movements expressed in the camera frame for eye-in-hand navigation. A soft virtual fixture that guides the slave end-effector to a fixed orientation $\Phi_s = \Phi_{ref}$ has been introduced

$$-\ddot{\Phi}_s - \boldsymbol{K}_{D_\Phi}\dot{\Phi}_s + \boldsymbol{K}_{P_\Phi}(\Phi_{ref} - \Phi_s) = 0 \tag{2.35}$$

where $\Phi_s = [\phi_s \; \theta_s \; \psi_s]^T$ are the slave XYZ Euler angles. Table 2.1 concisely presents the optimization problem for the task at hand, while Table 2.2 shows the controller and virtual fixtures parameters values.

Table 2.2: *Object tracking experiment: parameters.*

Parameter	Value
$\boldsymbol{K}_P$ ($\boldsymbol{K_s}$)	$1000\boldsymbol{I}_3$ ($1000\boldsymbol{I}_3\frac{N}{m}$)
$\boldsymbol{K}_D$ ($\boldsymbol{D_s}$)	$126.5\boldsymbol{I}_3$ ($126.5\boldsymbol{I}_3\frac{Ns}{m}$)
$\boldsymbol{Q}$ ($\boldsymbol{M_s}$)	$0.5\boldsymbol{I}_3$ ($1\boldsymbol{I}_3 kg$)
x_{bar}	$0.20m$
x_{wall}	$0.35m$
z_{table}	$0.1m$
$\underline{\boldsymbol{p}}$	$[32\ 24]^T px$
$\overline{\boldsymbol{p}}$	$[608\ 384]^T px$
$\boldsymbol{K}_{P\Phi}$	$100\boldsymbol{I}_3$
$\boldsymbol{K}_{D\Phi}$	$40\boldsymbol{I}_3$
$\boldsymbol{Q}_{\Phi}$	$0.05\boldsymbol{I}_3$

$\boldsymbol{I}_3$ is the 3-by-3 identity matrix

2.5.1 Preliminary experiment

We conducted a first experiment to evaluate the system behavior during the activation of a single hard virtual fixture. We scaled up the master position by a factor 2 due to the limited workspace of the haptic device, and led the slave to interact with the vertical wall. Fig. 2.4 shows the experiment results in terms of master and slave position and generated force feedback. While in free motion the slave correctly tracks the master, when the user tries to bring the robot inside the forbidden region, the controller makes the slave decelerate in time to avoid virtual fixture penetration. Upon contact with the virtual environment, the proposed approach generates a force feedback informing the user of the occurred interaction. It can be observed from the force and position signals that the main contribution to the impedance perceived by the user is given by the controller gains, as shown in (2.25a).

2.5.2 Object tracking with virtual fixtures

In the second experiment[1], we performed the object tracking task and evaluated the system response during the activation of multiple constraints, both hard and soft.

Since scaling the master position and velocity to match the whole slave robot workspace resulted in a too sensitive response, we adopted an ap-

[1] Video available at `https://youtu.be/GZZt5hcUsTc`

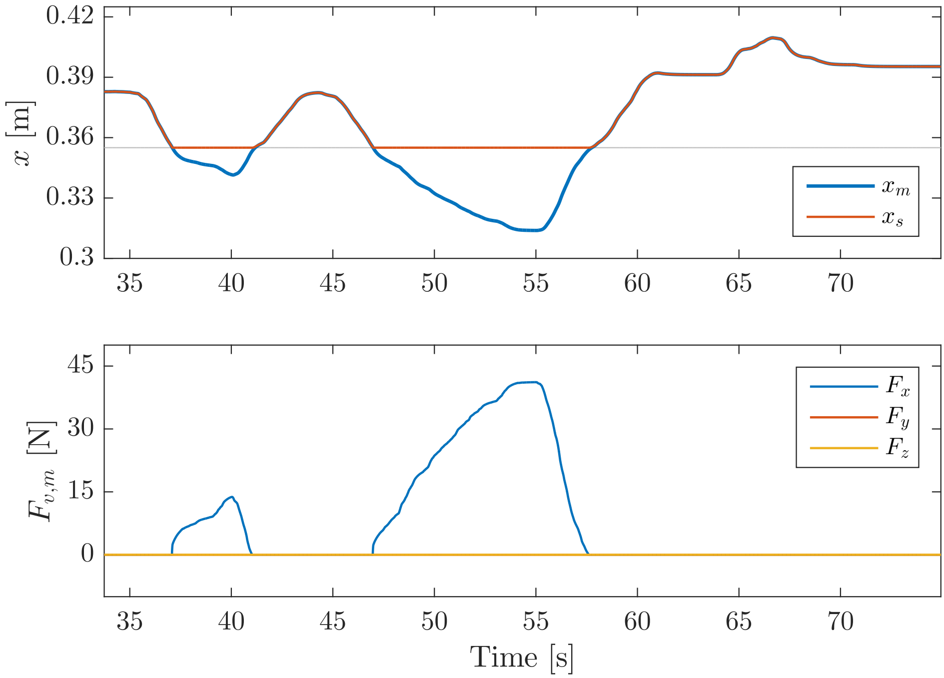


Figure 2.4: *Preliminary experiment. Master and slave positions with force feedback expressed in the slave robot base frame.*

proach analogous to the bubble technique used for navigation in virtual environments [48]. The approach consists in the definition of a volume in an haptic device workspace: the virtual environment is position-controlled while the device is inside the volume, and rate-controlled while outside. In a similar manner, for each axis of the master workspace we identified a threshold delimiting the rate-controlled region from the position-controlled one. When the master position on a certain axis exceeded the threshold, the haptic device position for that direction was interpreted as a reference velocity for the slave, increasing with the distance. This allows the use of the whole slave workspace by moving in rate control mode and to execute finer movements while in position control.

Figures 2.5 and 2.6 show quantities of interest for the experiment. In the interval from 40 to 60 seconds, the user commands a motion that would result in the object leaving the camera field of view (ξ_1 in Fig. 2.6). In return, to keep the object inside, the soft guidance virtual fixture on the slave orientation is violated and a corresponding force feedback is provided (Fig. 2.5 middle, 2.6 bottom). Here we took only the first term on the right-

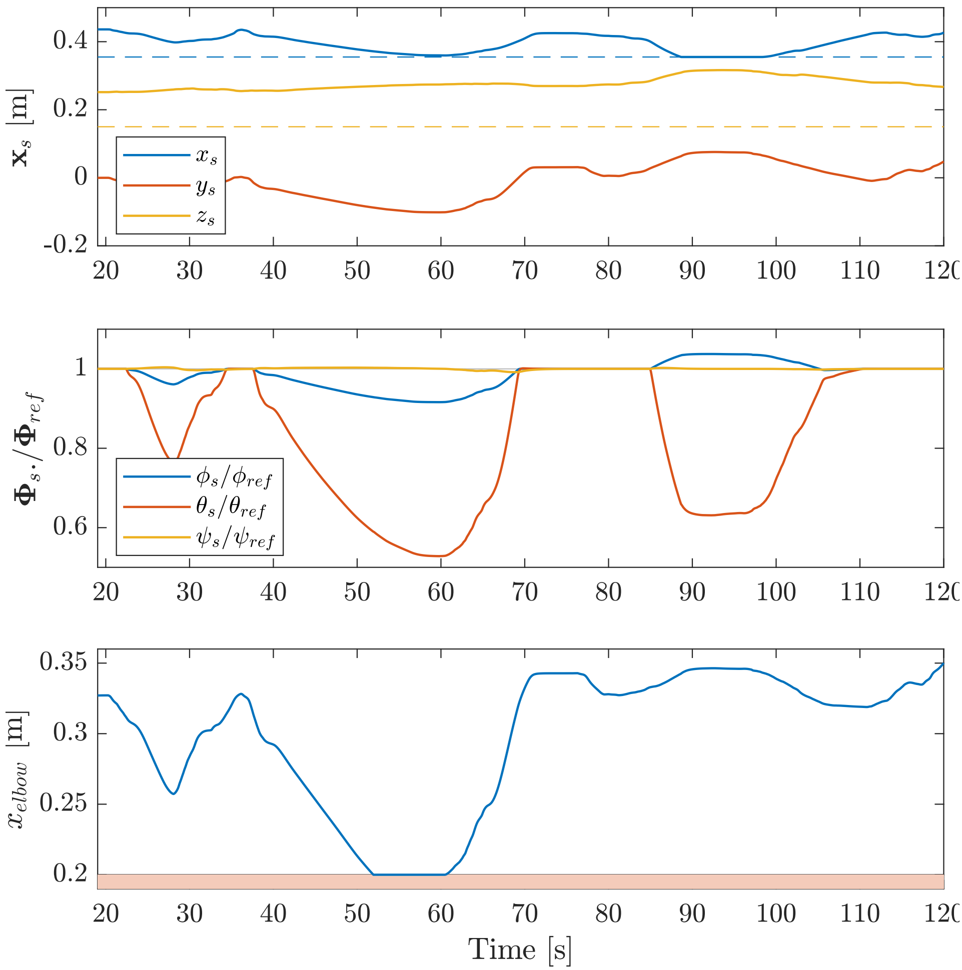


Figure 2.5: *Object tracking experiment. From top to bottom: slave robot position, slave robot XYZ Euler angles normalized over their reference value, slave robot elbow x axis position.*

hand side of (2.17), thus the resulting force acts along the gradient of the feature constraint. Note that, in this interval $(50 - 60\text{s})$ the elbow reaches the forbidden region preventing collision with the bar (Fig. 2.5 bottom), however no force feedback is felt by the user due to this constraint, since it can be satisfied with the slave redundant degree of freedom. The user can then continue with the desired movement undisturbed. From 85 to 105

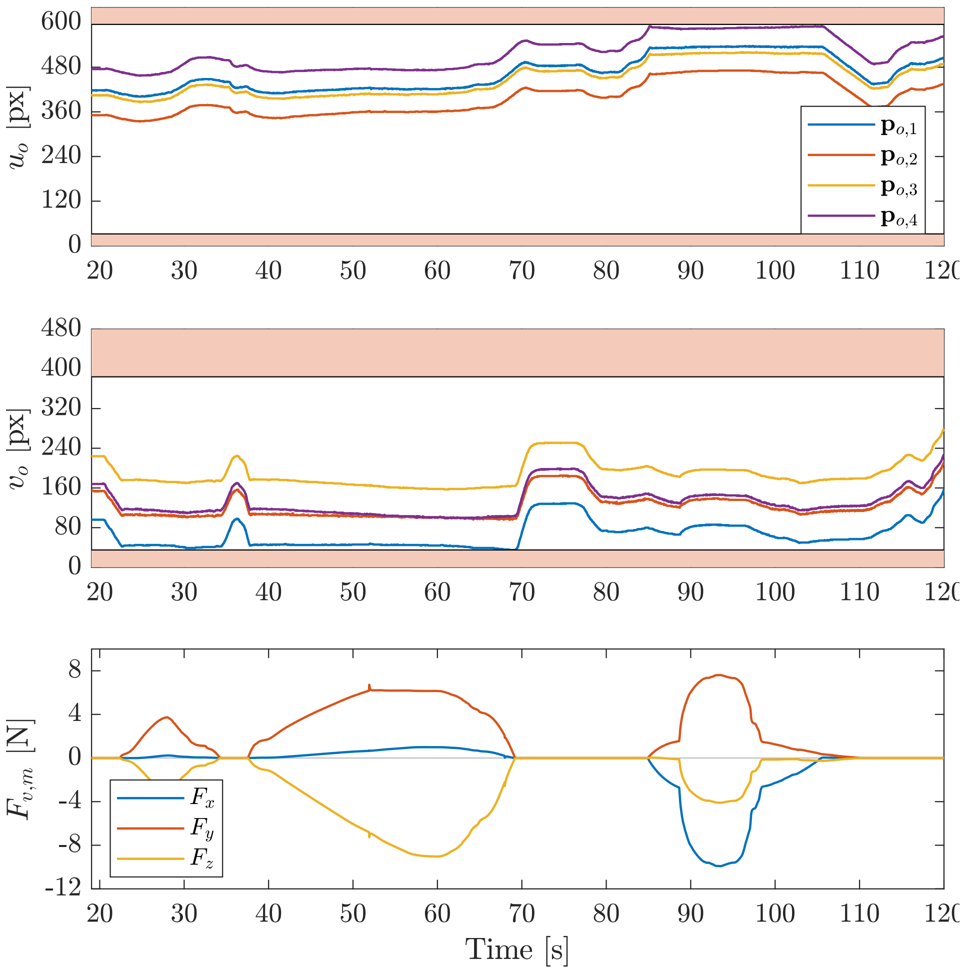


Figure 2.6: *Object tracking experiment. From top to bottom: object features position on the camera horizontal axis, object features position on the camera vertical axis, force feedback in the master frame of reference.*

seconds a similar event happens (see ξ_4 approaching the upper bound in Fig. 2.6), however at 90 seconds the master tries to lead the slave inside the forbidden area of the virtual wall (Fig. 2.5 top). The user immediately feels this interaction and the additional force feedback is summed to the one given by the soft constraint violation: the sudden increase is clearly visible at the bottom of Fig. 2.6.

We would like to remark that, at any time instant, none of the hard virtual fixtures are violated, and a force feedback is provided whenever the constraint cannot be satisfied with the slave inherent redundancy alone. On the other hand, the soft constraint is often violated since higher priority is given to the kinematic coordination between the two devices, but the user is still informed with an appropriate proportional haptic feedback.

2.6 Closing comment

In this chapter we provided an introduction to the bilateral teleoperation control problem for position-controlled slave manipulators and presented a solution in terms of constrained optimization, with focus on virtual fixtures inclusion in the framework and force feedback. A preliminary stability analysis has also been carried out. In the next chapter we will start to extend the approach to torque-controlled manipulators by discussing the robust control of these devices.

CHAPTER 3

Robust control of constrained redundant manipulators

In Chap. 2, we proposed an optimization-based teleoperation controller for virtual fixtures and force feedback rendering, however we assumed the presence of an inner position control loop in the slave device, limiting the considerations at the kinematics level and inside the controller bandwidth. In order to tackle robot interaction with real environments as well as to increase system performance, it is inevitable to consider the full robot dynamics, extending the control approach to torque-controlled robots. Before we proceed in the investigation of the bilateral teleoperation control problem for these devices, we must first consider the robust operational space control of a generic manipulator.

In the following we present a robust impedance control framework for trajectory tracking and interaction control based on the robustness properties of sliding mode control and the constraint specification features of model predictive control. A first inverse dynamics stage provides compensation of the robot nominal dynamics, while the second model predictive sliding mode control (MPSMC) stage rejects the residual uncertain dynamics and ensures the enforcement of the desired end-effector impedance, as well as accurate task execution. Unlike previous approaches, the sliding manifold is more intuitively formalized directly in the task space and modified via null-space projections to take into account manipulator redundancy. We show the controller accuracy and robustness even for small impedance gains, and against delays acting on the control inputs, together with an in-depth analysis of the SMC and MPC controllers properties, and how they influence each other.

Sec. 3.1 gives an overview of the literature on sliding mode and model predictive control, while in Sec. 3.2 a background on standard impedance and inverse dynamics control is provided. Sec. 3.3 presents the robust operational space sliding mode controller accompanied by the relative proofs. Sec. 3.4 adapts the previous architecture to an integral second order scheme to provide robustness from the initial time instant and for chattering attenuation. In 3.5, the concept of sliding mode redundancy is introduced to extend the approach to redundant manipulators. The overall MPSMC control scheme is detailed in Sec. 3.6, discussing the interaction between the two control components and the compensation of possible delays in the application of the control torque. Sec. 3.7 and 3.8 are devoted to the simulations conducted on a 4 d.o.f. planar robot and to the experiments carried out on a ABB YuMi 7 d.o.f. industrial robot, respectively.

3.1 Background

Since its conception in the '80s, impedance control [49] has played a primary role in robotics. Shaping the manipulator impedance allows the modification of the system behavior during contact, in order to achieve bounded forces or assign system dynamics suitable for cooperation with a human [50]. Realizing the desired impedance profile becomes increasingly important when tolerances are small, such as in insertion tasks [51], or when machining hard surfaces [52], where a lack of robustness may lead to large contact forces.

In [53] the authors employed a Cartesian impedance controller for cooperative tasks with a human, and exploited the redundant degree of freedom to shape the robot apparent inertia to be as close as possible to the desired one. In return this increases the range of selectable stabilizing inertia and damping values. Indeed, non-collocation of sensing and actuation is known to generate instability when the desired apparent inertia becomes increasingly small [54]. Buerger et al. [55] proposed a loop shaping technique to numerically obtain the controller parameters that minimize the difference between desired and attainable impedance, while ensuring stability.

These techniques focus more on the choice of stabilizing impedance parameters, based on the nominal model. Instead, robust robot control is more concerned with stability and performance in presence of uncertain model parameters and nonlinear friction phenomena, with the most well-known approaches resorting to passivity-based arguments, or variable structure schemes [56]. Among the most notable algorithms, the one proposed by Spong [57] employed a variable structure controller to compensate bounded uncertainties of the dynamic parameters. The resulting control law closely resembles the discontinuous control usually obtained with SMC [58].

The sliding mode approach has been extensively applied to robotic platforms. Early results by Lu et al. [59] used a first order sliding mode to assign a desired impedance for grinding tasks, however their scheme did not consider inertia shaping and reverted to a PI controller in proximity of the sliding manifold. In [60] the authors designed two sliding mode observers to estimate uncertain kinematics and unmodeled robot dynamics, in order to achieve zero reference tracking error in finite time. Supervisory SMC has also been applied to multi-robot systems where a safe cooperative grasp has to be guaranteed in terms of both positioning and interaction forces [61]. Although such controllers are able to perfectly cancel bounded matched uncertainties, they present some drawbacks in terms of discontinuity and chattering of the control variables. This behavior limits the applicability of first order sliding mode controllers to robotic systems, due to their tendency to excite mechanical resonances and increase joint gear wearing.

Higher order, and in particular second order algorithms [62], have been proposed to mitigate these effects by ensuring finite time convergence of the sliding variable and also its derivatives through state augmentation. Chattering is reduced at the price of more complex tuning and bounding assumptions on the derivatives of the system uncertain dynamics. Further improvements have been made with control gain adaptation, to decrease chattering amplitude when the plant is closer to the nominal model [63, 64]. The suboptimal second order algorithm has been applied to the motion control of a 3 d.o.f. planar manipulator in [65]. The authors individually designed for each joint a sliding surface based on the tracking error with independent control gains, and showed improved performance over a classical high gain PD controller. Nonetheless, the particular suboptimal solution still presented some oscillations at steady state. The same approach in observer form has also been employed in fault detection for robot manipulators [66].

It is worth noting that hydraulic robots are particularly suited to this type of control, since oil viscosity and hydraulic servovalves cause the system to suffer from highly nonlinear, time-varying dynamics [67, 68].

In the literature, SMC has been considered for overactuated systems, mainly in its application to fault tolerant control [69], where the control signal is reallocated to the remaining actuators upon a failure. Nevertheless, in robotic manipulation the redundant degrees of freedom may be used to complete additional tasks even during normal operation, and not only when an actuator failure occurs.

Redundancy resolution and impedance assignment in torque-controlled robots is usually performed by projection of the required torques in the null-space task Jacobian [70]. Such methods often avoid inertia shaping in order to circumvent the need for force sensors, but lack in robustness against uncertain dynamics, since an accurate impedance enforcement is frequently not required. A recent work by Lee et al. [71] employs the null-space projection formalization on a humanoid robot. Robustness is guaranteed by the simultaneous application of sliding mode and time-delay estimation to obtain sufficiently accurate approximations of inertia and residual coupling terms respectively. Unfortunately, the approach requires the measurement or otherwise estimation also of joint accelerations, inevitably introducing additional uncertainties.

Other approaches split the control of dynamics and kinematics, with a lower level controller ensuring joint reference tracking, and a higher level one dealing with inverse kinematics, thus solving the redundancy. A review of some schemes for operational control and redundancy resolution is given in [72].

Optimization in the form of MPC has been employed together with SMC to improve robustness. In [73], these techniques were applied to a solar air conditioning plant affected by variable time delay and showed considerable improvements over MPC alone. The sliding control component is responsible for compensating the disturbances, while the model predictive one only has to consider the delay acting on the nominal system. The authors in [74], instead, proposed a cascaded control structure that used MPC to compute and update the optimal parameters of a sliding manifold and achieve minimum energy or minimum time performance objectives for nonlinear mechanical systems.

One of the few instances of combined application of MPC and SMC to robot manipulators is the one proposed by Incremona et al. [75]. The authors employed a filtered first order integral sliding mode to reject the disturbances generated by imperfect dynamics cancellation. The design focused on trajectory tracking in joint space, with independently tuned sliding surfaces similar to [65], and an outer MPC layer guaranteeing optimal evolution of the constrained system, without any need for computationally intensive formulations such as tube-based MPC.

3.2 Operational space impedance control

In this section we provide some background on impedance control in the operational space. We consider a n d.o.f. rigid manipulator dynamic model of the form

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{\tau} - \boldsymbol{J}(\boldsymbol{q})^T \boldsymbol{F}_e \tag{3.1}$$

where $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}} \in \mathbb{R}^n$ are the robot joint positions, velocities and accelerations respectively, $\boldsymbol{B}(\boldsymbol{q}) \in \mathbb{R}^{n \times n}$ is the symmetric positive definite inertia matrix, $\boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^n$ is a vector comprising Coriolis, gravitational, and friction terms, $\boldsymbol{\tau} \in \mathbb{R}^n$ is the actuation torque, $\boldsymbol{F}_e \in \mathbb{R}^m$ is the external generalized force due to interaction with the environment, and finally $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{m \times n}$ is the robot end effector Jacobian, with $m \leq n$, the equality holding for non-redundant manipulators. For simplicity, only end effector forces and impedance will be considered. Nevertheless, given the proper Jacobian, the generalization to forces and impedances at intermediate points of the kinematic chain, as well as joint space impedance, can be easily derived if torque sensors are available at the joints.

In traditional impedance control, a cancellation of the nominal robot dynamics is first performed via feedback linearization, in order to remove the system's coupled nonlinear behavior. For full impedance control with inertia shaping, the inverse dynamics control torque is the following

$$\boldsymbol{\tau} = \hat{\boldsymbol{B}}\boldsymbol{v} + \hat{\boldsymbol{n}} + \boldsymbol{J}^T \hat{\boldsymbol{F}}_e \tag{3.2}$$

where the joint dependence notation has been dropped for simplicity. The hat indicates estimated quantities, such that in general $\hat{\boldsymbol{B}} \neq \boldsymbol{B}$, $\hat{\boldsymbol{n}} \neq \boldsymbol{n}$, $\hat{\boldsymbol{F}}_e \neq \boldsymbol{F}_e$, while $\boldsymbol{v} \in \mathbb{R}^n$ is the auxiliary control input that will be used to assign the new impedance. In case inertia shaping is not needed, the first term in (3.2) can be replaced with a generic auxiliary torque, and force measurements are not needed [70].

By substituting (3.2) in (3.1) we obtain the system equation

$$\ddot{\boldsymbol{q}} = \boldsymbol{B}^{-1}\hat{\boldsymbol{B}}\boldsymbol{v} + \boldsymbol{B}^{-1}(\tilde{\boldsymbol{n}} + \boldsymbol{J}^T \tilde{\boldsymbol{F}}_{\boldsymbol{e}}) = \boldsymbol{B}^{-1}\hat{\boldsymbol{B}}\boldsymbol{v} + \boldsymbol{B}^{-1}\boldsymbol{\eta} \tag{3.3}$$

with the tilde indicating the estimation error, e.g. $\tilde{\boldsymbol{n}} = \hat{\boldsymbol{n}} - \boldsymbol{n}$, and $\boldsymbol{\eta}$ lumping together the uncertain terms. It is immediately clear that the system is linear and decoupled only if perfect cancellation of the dynamics is achieved, which is not the case in general.

The auxiliary input is selected in order to assign the desired dynamics. The ultimate goal is to achieve the following impedance at the end point

$$\boldsymbol{M}\ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{D}\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}\tilde{\boldsymbol{x}} = \hat{\boldsymbol{F}}_e \tag{3.4}$$

$\boldsymbol{M}$, $\boldsymbol{D}$, $\boldsymbol{K} > 0 \in \mathbb{R}^{m\times m}$ are the desired inertia, damping and stiffness matrices respectively, while $\tilde{\boldsymbol{x}} = \boldsymbol{x}_r - \boldsymbol{x} \in \mathbb{R}^m$ is the end effector tracking error with $\boldsymbol{x}_r$ being the reference trajectory. Notice that it is not necessary to impose a desired impedance for all m directions, e.g. only translational components can be considered, thus introducing task redundancy.

Given (3.4), the auxiliary control can be found as a solution to the equation

$$\boldsymbol{M}\boldsymbol{J}\boldsymbol{v} = \boldsymbol{M}(\ddot{\boldsymbol{x}}_{\boldsymbol{r}} - \dot{\boldsymbol{J}}\dot{\boldsymbol{q}}) + \boldsymbol{D}\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}\tilde{\boldsymbol{x}} - \hat{\boldsymbol{F}}_e \tag{3.5}$$

Since $\hat{\boldsymbol{B}}$ is an estimated inertia matrix, it makes sense to choose it positive definite and thus invertible, then we can pre-multiply (3.3) with $\boldsymbol{M}\boldsymbol{J}\hat{\boldsymbol{B}}^{-1}\boldsymbol{B}$ and substitute (3.5). After rearrangement we get the closed-loop end effector dynamics

$$\boldsymbol{M}\ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{D}\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}\tilde{\boldsymbol{x}} = \hat{\boldsymbol{F}}_e - \boldsymbol{M}\boldsymbol{J}\hat{\boldsymbol{B}}^{-1}(\tilde{\boldsymbol{B}}\ddot{\boldsymbol{q}} + \boldsymbol{\eta}) \tag{3.6}$$

Note that this becomes equal to the desired impedance (3.4) only when there is no estimation error. In all other cases a deviation from the desired behavior should be expected both during transient and at steady state. Moreover, depending on the choice of the impedance parameters and the estimation accuracy, the system may even become unstable.

As a simple example, consider the 1 d.o.f. version of (3.6) for regulation to zero, and with $\boldsymbol{n}$ simply characterized by a linear viscous friction with coefficient μ

$$\frac{mb}{\hat{b}}\ddot{x} + \left(d - \frac{m}{\hat{b}}\tilde{\mu}\right)\dot{x} + kx = -\hat{f}_e + \frac{m}{\hat{b}}\tilde{f}_e \tag{3.7}$$

Since m, b, $\hat{b}$, $k > 0$, the coefficient of $\dot{x}$ must be positive to guarantee stability, i.e.

$$\frac{d}{m} > \frac{\hat{\mu} - \mu}{\hat{b}} \tag{3.8}$$

Depending on the accuracy of the estimated dynamics, there are limits on the values of d and m that can be selected. In this case, for a positive damping d, a lower bound on the damping-mass ratio must be satisfied if the system viscosity is overestimated.

It is now clear that a robust rejection of the disturbances in (3.6) is not only desirable, but in some cases also necessary for stability. The considerations made so far are valid for both non-redundant and redundant manipulators. However, for redundant ones or for task-redundant applications, proper controllers should be designed for the remaining degrees of freedom to achieve stability of the whole manipulator. How to guarantee the satisfaction of such requirements is presented next.

3.3 Operational space sliding mode control

Sliding mode control belongs to the class of variable structure controllers (VSC) [58]. Depending on the value of a sliding variable, function of the system state $\boldsymbol{\sigma}(\boldsymbol{z})$, a switching control action is applied to the system, in order to reach in a finite time t_f (*reaching phase*) and remain on (*sliding mode*), the sliding surface $\boldsymbol{\sigma} = 0$. Once on the manifold, the system will evolve according to the reduced dynamics of $\boldsymbol{\sigma}$. The control action $\boldsymbol{v}$ is usually chosen as follows

$$\boldsymbol{v} = \boldsymbol{v}_0 + \boldsymbol{v}_{smc}(\boldsymbol{\sigma}) \tag{3.9}$$

where $\boldsymbol{v}_0$ (*nominal control*) is a controller stabilizing the nominal system, while $\boldsymbol{v}_{smc}(\boldsymbol{\sigma})$ (*sliding mode control*) is a sliding variable-dependent control. Despite the possible discontinuities, SMC is able to robustly reject any bounded matched uncertainty (i.e. uncertain terms affecting the system on the control channel), as well as to decouple the design of the desired reduced system dynamics (the sliding manifold) from that of a stabilizing robust controller.

In what follows we derive the equations for the robust operational space impedance control of a robot manipulator.

After applying the feedback linearizing control (3.2), we can rewrite (3.3) in state-space normal form

$$\begin{cases} \dot{\boldsymbol{z}}_0 = \boldsymbol{z}_1 \\ \dot{\boldsymbol{z}}_1 = \boldsymbol{z}_2 \\ \dot{\boldsymbol{z}}_2 = \boldsymbol{B}^{-1}\hat{\boldsymbol{B}}\boldsymbol{v} + \boldsymbol{B}^{-1}\boldsymbol{\eta} \end{cases} \tag{3.10}$$

where $\boldsymbol{z} = [\boldsymbol{z}_0 \ \boldsymbol{z}_1 \ \boldsymbol{z}_2]^T = [\int \boldsymbol{q} \ \boldsymbol{q} \ \dot{\boldsymbol{q}}]^T$, with an extra integrator for reasons that will shortly become clear. Since we want to enforce the end effector dynamics (3.4), a naive approach would be to select as sliding manifold the equation

$$\boldsymbol{\sigma} = \boldsymbol{I} = \boldsymbol{M}\ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{D}\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}\tilde{\boldsymbol{x}} - \hat{\boldsymbol{F}}_e = 0 \tag{3.11}$$

Clearly, in this case the sliding manifold depends, not only on the system state, but also on joint accelerations. This requires the use of an acceleration estimator in order to compute $\boldsymbol{\sigma}$ and then select the correct sliding mode control, unavoidably introducing inaccuracies in its evaluation. One solution could be to remove the inertial term, however, here we start developing our approach from the idea proposed in [76] for a 1 d.o.f. manipulator, that allows full impedance specification.

Proposition 3.3.1. *Consider the sliding vector*

$$\boldsymbol{\sigma} = \int_0^t \boldsymbol{I} d\tau = \boldsymbol{M}\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{D}\tilde{\boldsymbol{x}} + \boldsymbol{K}\int_0^t \tilde{\boldsymbol{x}} d\tau - \int_0^t \hat{\boldsymbol{F}}_e d\tau \tag{3.12}$$

If $\boldsymbol{\sigma} = 0, \forall t \geq t_f$, then, $\forall t \geq t_f$, $\boldsymbol{I} = 0$, and the desired impedance relation (3.4) *is correctly imposed.*

Proof. The proof is trivial: if $\int \boldsymbol{I}$ is identically zero beginning from a time t_f, then, starting from the same instant also its derivative is zero, thus $\boldsymbol{I} = 0$. Note that such condition is only sufficient. □

By using (3.12), we have implicitly augmented the system with an additional state associated with the position integral, which explains (3.10). The controlled system is of order $3n$, while during the sliding mode the reduced dynamics are of order $3n - m$, since $\boldsymbol{\sigma} \in \mathbb{R}^m$. Given the sliding vector definition, we must now compute the control (3.9) so that the sliding mode is enforced.

3.3.1 Nominal control $\boldsymbol{v}_0$

No particular restrictions exist for $\boldsymbol{v}_0$. Its objective is to provide a stabilizing behavior given the nominal system, noting that its choice will later influence the tuning and control effort relative to $\boldsymbol{v}_{smc}$. In principle $\boldsymbol{v}_0 = 0$ could be taken.

For our purposes, let us first consider $\dot{\boldsymbol{\sigma}}$. After substituting (3.3) in the time derivative of (3.12), we obtain

$$\begin{aligned}\dot{\boldsymbol{\sigma}} =& \boldsymbol{M}\ddot{\boldsymbol{x}}_r - \boldsymbol{M}\boldsymbol{J}\boldsymbol{B}^{-1}\hat{\boldsymbol{B}}\boldsymbol{v} - \boldsymbol{M}\boldsymbol{J}\boldsymbol{B}^{-1}\boldsymbol{\eta} - \boldsymbol{M}\dot{\boldsymbol{J}}\dot{\boldsymbol{q}} + \\ &+ \boldsymbol{D}\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}\tilde{\boldsymbol{x}} - \hat{\boldsymbol{F}}_e\end{aligned} \tag{3.13}$$

If we are in sliding mode, then the desired impedance is enforced (Prop. 3.3.1), and $\dot{\boldsymbol{\sigma}} = 0$. From (3.13) we can compute

$$\begin{aligned}\boldsymbol{v}_{eq}^{th} =& (\boldsymbol{J}\boldsymbol{B}^{-1}\hat{\boldsymbol{B}})^{\dagger}\boldsymbol{M}^{-1}(\boldsymbol{M}\ddot{\boldsymbol{x}}_r - \boldsymbol{M}\boldsymbol{J}\boldsymbol{B}^{-1}\boldsymbol{\eta} + \\ &- \boldsymbol{M}\dot{\boldsymbol{J}}\dot{\boldsymbol{q}} + \boldsymbol{D}\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}\tilde{\boldsymbol{x}} - \hat{\boldsymbol{F}}_e)\end{aligned} \tag{3.14}$$

where † indicates the Moore-Penrose pseudo-inverse. $\boldsymbol{v}_{eq}^{th}$ is the so-called equivalent control, i.e. the theoretical continuous control signal that, if applied, maintains the system on the manifold. Unfortunately, such control cannot be computed due to its dependence on the uncertain terms. The best that can be done is to neglect these, as if we were working on the nominal plant. This also motivates the addition of a discontinuous control component $\boldsymbol{v}_{smc}$ to ensure the sliding mode.

By enforcing this approximation in (3.14), we obtain

$$\boldsymbol{v}_{eq} = \boldsymbol{J}^\dagger \boldsymbol{M}^{-1} \left(\boldsymbol{M}(\ddot{\boldsymbol{x}}_r - \dot{\boldsymbol{J}}\dot{\boldsymbol{q}}) + \boldsymbol{D}\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}\tilde{\boldsymbol{x}} - \hat{\boldsymbol{F}}_e \right) \tag{3.15}$$

Comparing (3.15) with (3.5), we notice that the first equation is a solution to the second. Indeed, given the sliding vector (3.12), the best that we can do in terms of equivalent control is to apply the nominal impedance controller of Sec. 3.2. Due to these considerations, we may choose $\boldsymbol{v}_0 = \boldsymbol{v}_{eq}$.

3.3.2 Sliding mode control $\boldsymbol{v}_{smc}$

The sliding mode control must be selected to ensure that the system reaches the sliding manifold in finite time and remains on it. Given a candidate Lyapunov function $V(\boldsymbol{\sigma})$, to reach the manifold in finite time, $\boldsymbol{v}_{smc}(\boldsymbol{\sigma})$ has to be chosen so that

$$\dot{V}(\boldsymbol{\sigma}) < -\mu \|\boldsymbol{\sigma}\| \tag{3.16}$$

where μ is a positive scalar. For first order SMC it is usually sufficient to take the scaled element-wise signum function of the manifold vector $\boldsymbol{v}_{smc} = k\text{sgn}(\boldsymbol{\sigma})$, with $k > 0$ large enough to compensate the uncertainties. Unfortunately, here $\boldsymbol{v}_{smc} \in \mathbb{R}^n$, while $\boldsymbol{\sigma} \in \mathbb{R}^m$ with $m \leq n$, so that this choice of discontinuous control is not feasible for redundant manipulators or in presence of task-redundancy.

To circumvent this problem we define a new sliding variable $\boldsymbol{\sigma}_q \in \mathbb{R}^n$. The following result holds

Proposition 3.3.2. *Consider a n d.o.f. manipulator, a sliding vector $\boldsymbol{\sigma} \in \mathbb{R}^m$ as in* (3.12)*, with $m \leq n$, and define*

$$\boldsymbol{\sigma}_q = \boldsymbol{J}^\dagger \boldsymbol{\sigma}, \quad \boldsymbol{\sigma}_q \in \mathbb{R}^n \tag{3.17}$$

If the Jacobian $\boldsymbol{J}$ is full rank, and $\boldsymbol{\sigma}_q = 0$, $\forall t \geq t_f$, then, $\forall t \geq t_f$, $\boldsymbol{I} = 0$ and the desired impedance is correctly imposed.

Proof. By hypothesis, for $t \geq t_f$, we can write the overdetermined system of equations

$$\boldsymbol{J}^\dagger \boldsymbol{\sigma} = \boldsymbol{J}^\dagger \int_0^t \boldsymbol{I} d\tau = 0 \tag{3.18}$$

If $\boldsymbol{J}$ is full rank, the robot is far from kinematic singularities, and $rank \boldsymbol{J}^{\dagger} = dim \boldsymbol{I}$, the Rouché-Capelli theorem holds and gives the unique trivial solution

$$\boldsymbol{\sigma} = \int_0^t \boldsymbol{I} d\tau = 0 \tag{3.19}$$

Then, by applying Prop. 3.3.1, we have $\boldsymbol{I} = 0$ and the desired impedance requirement is fulfilled. □

Corollary 3.3.2.1. *When on the sliding manifold $\boldsymbol{\sigma}_q = 0$, the equivalent control computed from $\dot{\boldsymbol{\sigma}}_q = 0$ is the same as the one obtained from $\dot{\boldsymbol{\sigma}} = 0$ in* (3.14).

Proof. Since $\boldsymbol{\sigma}_q = 0 \Rightarrow \boldsymbol{\sigma} = 0$, on the manifold we can write

$$\dot{\boldsymbol{\sigma}}_q = \dot{\boldsymbol{J}}^{\dagger} \boldsymbol{\sigma} + \boldsymbol{J}^{\dagger} \dot{\boldsymbol{\sigma}} = \boldsymbol{J}^{\dagger} \dot{\boldsymbol{\sigma}} = 0 \tag{3.20}$$

$\boldsymbol{J}$ being full rank implies $\dot{\boldsymbol{\sigma}} = 0$, which gives (3.14). □

These results suggest that $\boldsymbol{\sigma}_q$ can be considered when enforcing the desired dynamics, and more easily compute a suitable $\boldsymbol{v}_{smc}(\boldsymbol{\sigma}_q)$. Thanks to Corol. 3.3.2.1, it is also unnecessary to modify $\boldsymbol{v}_{eq}$ and thus $\boldsymbol{v}_0$.

To finally obtain the sliding mode control, let us first define the candidate Lyapunov function

$$V = \frac{1}{2} \boldsymbol{\sigma}^T \boldsymbol{M}^{-1} \boldsymbol{\sigma} \tag{3.21}$$

Taking the time derivative, and substituting (3.9) and (3.15) in (3.13), we get

$$\dot{V} = \boldsymbol{\sigma}^T \boldsymbol{J}(-\boldsymbol{B}^{-1} \hat{\boldsymbol{B}} (\boldsymbol{v}_{eq} + \boldsymbol{v}_{smc}(\boldsymbol{\sigma})) - \boldsymbol{B}^{-1} \boldsymbol{\eta} + \boldsymbol{v}_{eq}) \tag{3.22}$$

Noting that $\boldsymbol{B}^{-1} \hat{\boldsymbol{B}} = \boldsymbol{I}_n + \boldsymbol{B}^{-1} \tilde{\boldsymbol{B}}$, with $\boldsymbol{I}_n$ the $n \times n$ identity matrix, and collecting all the terms affected by uncertainty, yields

$$\dot{V} = \boldsymbol{\sigma}^T \boldsymbol{J}(-\boldsymbol{v}_{smc}(\boldsymbol{\sigma}) - \boldsymbol{\chi}) \tag{3.23}$$

with $\boldsymbol{\chi} = \boldsymbol{B}^{-1}(\tilde{\boldsymbol{B}}(\boldsymbol{v}_{eq} + \boldsymbol{v}_{smc}(\boldsymbol{\sigma})) + \boldsymbol{\eta})$. Rewriting the derivative in terms of $\boldsymbol{\sigma}_q$ by using (3.17) gives

$$\dot{V} = \boldsymbol{\sigma}_q^T \boldsymbol{T}(-\boldsymbol{v}_{smc}(\boldsymbol{\sigma}_q) - \boldsymbol{\chi}) \tag{3.24}$$

with $\boldsymbol{T} = \boldsymbol{J}^T \boldsymbol{J}$.

Although one may be lead to take $\boldsymbol{v}_{smc}(\boldsymbol{\sigma}_q) = k\text{sgn}(\boldsymbol{\sigma}_q)$ and consider each robot joint separately thanks to the decoupling obtained via the feedback linearization (e.g. as done in [75]), the dynamics are coupled again due to the choice of the sliding manifold, hence the presence of $\boldsymbol{T}$ in the previous equation. A better choice is to employ the unit vector $\frac{\boldsymbol{\sigma}_q}{\|\boldsymbol{\sigma}_q\|}$, $\boldsymbol{v}_{smc}(\boldsymbol{\sigma}_q) = k\frac{\boldsymbol{\sigma}_q}{\|\boldsymbol{\sigma}_q\|}$, $k > 0$ [77]. The Euclidean norm is used here and in the following.

To ensure that the manifold is reached in finite time (manifold μ-reachability), we must have

$$\dot{V} = \boldsymbol{\sigma}_q^T \left(-\boldsymbol{T}\frac{k}{\|\boldsymbol{\sigma}_q\|}\right)\boldsymbol{\sigma}_q + \boldsymbol{\sigma}_q^T(-\boldsymbol{T}\boldsymbol{\chi}) \leq -\mu\|\boldsymbol{\sigma}_q\| \tag{3.25}$$

Multiplying both sides by $\|\boldsymbol{\sigma}_q\|$ and bounding the second term on the left-hand side, we can write

$$\boldsymbol{\sigma}_q^T\left(-k\boldsymbol{T} + \|\boldsymbol{T}\boldsymbol{\chi}\|\boldsymbol{I}_n\right)\boldsymbol{\sigma}_q \leq \boldsymbol{\sigma}_q^T\left(-\mu\boldsymbol{I}_n\right)\boldsymbol{\sigma}_q \tag{3.26}$$

where we assumed that $\|\boldsymbol{T}\boldsymbol{\chi}\|$ is finite. Indeed, by applying matrix norm properties, we have

$$\|\boldsymbol{T}\boldsymbol{\chi}\| \leq \|\boldsymbol{T}\|\|\boldsymbol{\chi}\| \leq \|\boldsymbol{T}\|\|\boldsymbol{B}^{-1}\|(\bar{\tilde{B}}(\bar{v}_{eq} + k) + \bar{\eta}) \tag{3.27}$$

where the bar indicates the uncertainties norm upper bound, $\bar{\tilde{B}} = \max_{\boldsymbol{q}}\|\tilde{\boldsymbol{B}}(\boldsymbol{q})\|$, $\bar{\eta} = \max_{\boldsymbol{q}}\|\boldsymbol{\eta}(\boldsymbol{q})\|$, and we assumed bounded inertia matrix and Jacobian norms [78]. Note that we also made the reasonable assumption of bounded equivalent control $\|\boldsymbol{v}_{eq}\| \leq \bar{v}_{eq}$. This will be explicitly considered in Sec. 3.6.2 via a saturation constraint.

Using (3.27) with (3.26) we obtain the matrix inequality

$$k(\boldsymbol{T} - \|\boldsymbol{T}\|\|\boldsymbol{B}^{-1}\|\bar{\tilde{B}}\boldsymbol{I}_n) \geq (\mu + \|\boldsymbol{T}\|\|\boldsymbol{B}^{-1}\|(\bar{\tilde{B}}\bar{v}_{eq} + \bar{\eta}))\boldsymbol{I}_n \tag{3.28}$$

In case $\boldsymbol{T} > 0$ (i.e. non-redundant manipulator), a sufficient condition is obtained by choosing k as follows

$$k \geq \frac{\mu + \overline{\lambda}(\boldsymbol{T})\|\boldsymbol{B}^{-1}\|(\bar{\tilde{B}}\bar{v}_{eq} + \bar{\eta})}{\underline{\lambda}(\boldsymbol{T}) - \overline{\lambda}(\boldsymbol{T})\|\boldsymbol{B}^{-1}\|\bar{\tilde{B}}} \tag{3.29}$$

where $\overline{\lambda}(\boldsymbol{T})$ and $\underline{\lambda}(\boldsymbol{T})$ indicate the largest and smallest singular values of $\boldsymbol{T}$. For k to be greater than zero, we must have

$$\bar{\tilde{B}} < \frac{\underline{\lambda}(\boldsymbol{T})}{\overline{\lambda}(\boldsymbol{T})\|\boldsymbol{B}^{-1}\|} \tag{3.30}$$

Note that this means that the uncertainty on the inertia matrix has to be small enough compared to the scaled inverse of the condition number of $\boldsymbol{T}$.

Indeed, as the robot moves closer to a singular configuration, $\underline{\lambda}(\boldsymbol{T})$ becomes increasingly small (large condition number), while k approaches infinity. Therefore it is not possible to compensate uncertainties in the direction of the eigenvector corresponding to that particular singular value, when close to a singularity.

A similar issue occurs if $\boldsymbol{T} \geq 0$ (redundant manipulator). In this case the smallest singular value is exactly zero, and k can be computed only to ensure disturbance rejection in the row space of $\boldsymbol{J}$ by taking in (3.29) the smallest non-zero singular value of $\boldsymbol{T}$, while guaranteeing that the sliding manifold $\boldsymbol{\sigma} = 0$ will be reached. The problem of uncertainty compensation in the null space of $\boldsymbol{J}$ will be tackled more extensively in Sec. 3.5.

The result in (3.29) is however conservative. In practice smaller gains or adaptation laws [63,64] can be tuned independently for each element of the sliding vector (i.e. each joint), so that convergence in finite time still holds. Finally, if we had chosen $\boldsymbol{v}_0 = 0$, in (3.29) the numerator would have been $\mu + \overline{\lambda}(\boldsymbol{T})(\bar{v}_{eq} + \|\boldsymbol{B}^{-1}\|\bar{\eta})$, thus requiring a higher gain due to the absence of $\bar{\tilde{B}}$ pre-multiplying the equivalent control.

Given these considerations, the auxiliary control (3.9) is given by the following:

$$\boldsymbol{v} = \boldsymbol{v}_{eq} + k\frac{\boldsymbol{\sigma}_q}{\|\boldsymbol{\sigma}_q\|} \tag{3.31}$$

3.4 Second order integral sliding mode

In Sec. 3.3 we derived the first order sliding mode controller to enforce the desired impedance. The main drawback is that a first order algorithm produces undesirable chattering effects of the control variable. Moreover, the robot behaves with the desired impedance only after a finite time t_f, while during the transient, for $t_0 \leq t < t_f$, the system evolves in a way that depends on the uncertainty. To address these issues, we apply the integral sliding mode (ISM) approach, that will ensure $\boldsymbol{I} = 0$ from the initial time instant t_0, and the super-twisting algorithm (STA) to obtain a second order sliding mode and alleviate chattering.

3.4.1 Integral sliding mode

Let us first define the nominal version of (3.10) without uncertainty, and controlled solely by the nominal control $\boldsymbol{v}_0$. The resulting system is a chain of integrators

$$\dot{\boldsymbol{z}} = \boldsymbol{f}(\boldsymbol{z}, \boldsymbol{v}_0) = \begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \\ \boldsymbol{v}_0 \end{bmatrix} \tag{3.32}$$

To ensure the sliding mode from t_0, $\boldsymbol{\sigma}$ is usually modified as follows [79]

$$\Sigma(t) = \boldsymbol{\sigma}(t) - \boldsymbol{\lambda}(t) \tag{3.33}$$

where $\Sigma(t)$ is the new modified sliding vector, and $\boldsymbol{\lambda}(t)$ is the *reaching function*

$$\boldsymbol{\lambda}(t) = \int_{t_0}^{t} \left(\frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{z}} \boldsymbol{f}(\boldsymbol{z}, \boldsymbol{v}_0) + \dot{\Gamma} \right) d\tau + \boldsymbol{\sigma}(t_0) \tag{3.34}$$

where Γ comprises the terms in $\boldsymbol{\sigma}$ that do not depend on the system state.

Definition 3.4.1. *We say that the control $\boldsymbol{v}_{eq}$ is a nominal solution to $\dot{\boldsymbol{\sigma}}$, if it solves the equation $\dot{\boldsymbol{\sigma}} = 0$ evaluated with uncertainty-free system dynamics.*

We can prove the following result.

Proposition 3.4.1. *Let $\boldsymbol{v}_0 = \boldsymbol{v}_{eq}$* (3.15) *be a nominal solution to $\dot{\boldsymbol{\sigma}}$, the integral sliding vector formulation* (3.33) *simplifies to*

$$\Sigma(t) = \boldsymbol{\sigma}(t) - \boldsymbol{\sigma}(t_0) \tag{3.35}$$

Furthermore, $\boldsymbol{I} = 0, \forall t \geq t_0$, and the desired impedance is correctly imposed from the initial time instant.

Proof. Let us define $\mathsf{x}(\boldsymbol{z}) = [\int \boldsymbol{x} \ \boldsymbol{x} \ \dot{\boldsymbol{x}}]^T$, $\Theta = [\boldsymbol{K} \ \boldsymbol{D} \ \boldsymbol{M}]$, $\Gamma = \Theta \mathsf{x}_r - \int \hat{\boldsymbol{F}}_e$. Eq. (3.12) can be written in compact form

$$\boldsymbol{\sigma} = -\Theta \mathsf{x}(\boldsymbol{z}) + \Gamma \tag{3.36}$$

Noting that $\frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{z}} = \frac{\partial \boldsymbol{\sigma}}{\partial \mathsf{x}} \frac{\partial \mathsf{x}}{\partial \boldsymbol{z}}$, we can substitute (3.36) in (3.34) and obtain

$$\boldsymbol{\lambda}(t) = \int_{t_0}^{t} \left(-\Theta \frac{\partial \mathsf{x}}{\partial \boldsymbol{z}} \boldsymbol{f}(\boldsymbol{z}, \boldsymbol{v}_0) + \dot{\Gamma} \right) d\tau + \boldsymbol{\sigma}(t_0) \tag{3.37}$$

Given the well-known relation between operational and joint space coordinates, we have also

$$\frac{\partial \mathsf{x}}{\partial \boldsymbol{z}} \boldsymbol{f}(\boldsymbol{z}, \boldsymbol{v}_0) = \begin{bmatrix} \frac{\partial \int \boldsymbol{x}}{\partial \boldsymbol{z}_0} & \frac{\partial \int \boldsymbol{x}}{\partial \boldsymbol{z}_1} & 0 \\ 0 & \boldsymbol{J} & 0 \\ 0 & \frac{\partial \boldsymbol{J}}{\partial \boldsymbol{z}_1} \boldsymbol{z}_2 & \boldsymbol{J} \end{bmatrix} \begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \\ \boldsymbol{v}_0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{x} \\ \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{J}} \boldsymbol{z}_2 + \boldsymbol{J} \boldsymbol{v}_0 \end{bmatrix} \tag{3.38}$$

Substituting the expressions for Θ, Γ and remembering that $\boldsymbol{z}_2 = \dot{\boldsymbol{q}}$, $\boldsymbol{\lambda}(t)$ becomes

$$\begin{aligned}\boldsymbol{\lambda}(t) = &\int_{t_0}^{t} \left(\boldsymbol{M}(\ddot{\boldsymbol{x}}_{\boldsymbol{r}} - \boldsymbol{J}\boldsymbol{v}_0 - \dot{\boldsymbol{J}}\dot{\boldsymbol{q}}) + \boldsymbol{D}\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}\tilde{\boldsymbol{x}} - \hat{\boldsymbol{F}}_e\right) d\tau + \\ &+ \boldsymbol{\sigma}(t_0)\end{aligned} \tag{3.39}$$

By hypothesis, $\boldsymbol{v}_0 = \boldsymbol{v}_{eq}$, if we use (3.15) in the previous equation, all the terms in the integral cancel out

$$\boldsymbol{\lambda}(t) = \boldsymbol{\sigma}(t_0) \tag{3.40}$$

From (3.33) we then obtain (3.35).
For the second part, it is sufficient to use an argument similar to the one used to prove Prop. 3.3.1. Due to the ISM, $\Sigma(t) = 0, \forall t \geq t_0$. Then $\int \boldsymbol{I} = \boldsymbol{\sigma}(t_0)$, and since $\boldsymbol{\sigma}(t_0)$ is a constant, by taking the derivative we have $\boldsymbol{I} = 0, \forall t \geq t_0$. □

We can make the following observations in regards to the application of the ISM to the proposed sliding manifold:

Remark 1. Since (3.35) simply shifts $\boldsymbol{\sigma}$ by a constant value, $\dot{\Sigma} = \dot{\boldsymbol{\sigma}}$, and the new theoretical equivalent control is the same as (3.14), which is consistent with the initial choice of taking $\boldsymbol{v}_0 = \boldsymbol{v}_{eq}$.

Remark 2. So far we have considered the sliding vector expressed in the operational space. Nonetheless, in order to compute the sliding mode control $\boldsymbol{v}_{smc}$, similarly to (3.17) it is useful to define

$$\Sigma_q = \boldsymbol{J}^{\dagger}\Sigma, \quad \Sigma_q \in \mathbb{R}^n \tag{3.41}$$

Given Prop. 3.4.1, it is straightforward to see that Prop. 3.3.2 and Corol. 3.3.2.1 also hold for the pair Σ, Σ_q, but from a time t_0.

Remark 3. Although one may try to apply the ISM directly to $\boldsymbol{\sigma}_q$ instead of $\boldsymbol{\sigma}$, particular attention should be paid to the following. While (3.15) is a nominal solution for $\dot{\boldsymbol{\sigma}}$, this is not the case for $\dot{\boldsymbol{\sigma}}_q$ when not on the manifold. Indeed, evaluating $\dot{\boldsymbol{\sigma}}_q$ with the uncertainty-free dynamics (3.32)

$$\dot{\boldsymbol{\sigma}}_q = \dot{\boldsymbol{J}}^{\dagger}\boldsymbol{\sigma} + \boldsymbol{J}^{\dagger}\dot{\boldsymbol{\sigma}} = \dot{\boldsymbol{J}}^{\dagger}\boldsymbol{\sigma} \neq 0 \tag{3.42}$$

Therefore, Prop. 3.4.1 does not hold for $\boldsymbol{\sigma}_q$ if we want to maintain the same nominal control (3.15). Nonetheless, the ISM may still be applied directly to $\boldsymbol{\sigma}_q$ if desired, as long as its full formulation (3.33) is used.

For the ISM, keeping the same nominal control is necessary to impose the desired impedance. We can show that the following is true:

Proposition 3.4.2. *If an ISM* (3.33) *with control $\boldsymbol{v} = \boldsymbol{v}_0 + \boldsymbol{v}_{smc}$ is applied to the uncertain system* (3.3) *with sliding vector $\boldsymbol{\sigma}$* (3.12)*, then the system will evolve according to the nominal feedback-linearized operational space dynamics imposed by the control $\boldsymbol{v}_0$, with $\boldsymbol{v}_{smc}$ acting as an uncertainty estimator and compensator.*

Proof. Let us take the derivative of (3.33), and substitute the system dynamics (3.3) to obtain the control that allows to remain on the manifold

$$\dot{\Sigma} = \dot{\boldsymbol{\sigma}} - \dot{\boldsymbol{\lambda}} = \boldsymbol{M}\boldsymbol{J}(\boldsymbol{v}_0 - \boldsymbol{B}^{-1}\hat{\boldsymbol{B}}\boldsymbol{v} - \boldsymbol{B}^{-1}\boldsymbol{\eta}) = 0 \tag{3.43}$$

$$\boldsymbol{v} = \boldsymbol{v}_0 + \boldsymbol{v}_{smc} = (\boldsymbol{J}\boldsymbol{B}^{-1}\hat{\boldsymbol{B}})^{\dagger}\boldsymbol{J}(\boldsymbol{v}_0 - \boldsymbol{B}^{-1}\boldsymbol{\eta}) \tag{3.44}$$

Using the previous equation and pre-multiplying (3.3) by $\boldsymbol{J}$, we get

$$\ddot{\boldsymbol{x}} = \boldsymbol{J}\boldsymbol{v}_0 + \dot{\boldsymbol{J}}\dot{\boldsymbol{q}} \tag{3.45}$$

which is just the expression of the operational space feedback-linearized dynamics with compensated uncertainties controlled by $\boldsymbol{v}_0$. □

It is now clear that if an ISM is applied, the choice of $\boldsymbol{v}_0$ affects the closed loop dynamics, unlike the standard sliding mode of Sec. 3.3.

3.4.2 Alternative ISM formulation

Given proposition 3.4.2, we can also provide an alternative formulation to the ISM of the previous section, that directly assigns Σ_q. We can exploit the fact that, if an ISM is applied, $\boldsymbol{v}_{smc}(\Sigma_q)$ acts as an uncertainty estimator and compensator, and the system dynamics are solely governed by our choice of $\boldsymbol{v}_0$. Under this assumption, applying the auxiliary control (3.9) guarantees that the system evolves with the completely feedback linearized and decoupled dynamics (3.32).

We can prove the following result:

Proposition 3.4.3. *Consider the partially feedback linearized system* (3.3) *and the control* (3.9)*. Let*

$$\Sigma_q(t) = \dot{\boldsymbol{q}}(t) - \dot{\boldsymbol{q}}_0(t) \tag{3.46}$$

be the selected sliding vector, with $\dot{\boldsymbol{q}}_0(t) = \int_{t_0}^{t} \boldsymbol{v}_0 d\tau + \dot{\boldsymbol{q}}(t_0)$. On the sliding manifold $\Sigma_q = 0$, the system evolves with the ideal dynamics (3.32)*. Moreover, this holds beginning from the initial time instant t_0.*

Proof. To prove the proposition we have to show that (3.46) produces an integral sliding mode.

Let us define an auxiliary generic sliding vector

$$\boldsymbol{\sigma}_q^*(t) = (\dot{\boldsymbol{q}}(t) - \dot{\boldsymbol{q}}_r(t)) + \delta(\boldsymbol{q}(t) - \boldsymbol{q}_r(t)) \tag{3.47}$$

where δ is a positive gain and the subscript r indicates a reference joint trajectory. By applying the definition of integral sliding mode to $\boldsymbol{\sigma}_q^*$ [79], we may write

$$\Sigma_q(t) = \boldsymbol{\sigma}_q^*(t) - \int_{t_0}^{t} \left(\frac{\partial \boldsymbol{\sigma}_q^*}{\partial \boldsymbol{z}} \boldsymbol{f}(\boldsymbol{z}, \boldsymbol{v}_0) + \frac{\partial \boldsymbol{\sigma}_q^*}{\partial \boldsymbol{z}_r} \dot{\boldsymbol{z}}_r \right) d\tau - \boldsymbol{\sigma}_q^*(t_0) \tag{3.48}$$

Substituting (3.32), (3.47) in the integral, we obtain

$$\Sigma_q(t) = \boldsymbol{\sigma}_q^*(t) - \int_{t_0}^{t} \left[(\boldsymbol{v}_0 - \ddot{\boldsymbol{q}}_r) + \delta(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_r) \right] d\tau - \boldsymbol{\sigma}_q^*(t_0) \tag{3.49}$$

By simplifying, the sliding vector becomes the following

$$\Sigma_q(t) = \dot{\boldsymbol{q}}(t) - \int_{t_0}^{t} \boldsymbol{v}_0 d\tau - \dot{\boldsymbol{q}}(t_0) \tag{3.50}$$

which is exactly (3.46). Hence, the proposed sliding vector generates an integral sliding mode. Indeed, we have that $\Sigma_q(t_0) = 0$, and the system will remain on the manifold if $\boldsymbol{v}_{smc}$ is chosen appropriately.

To show that (3.32) describes the system dynamics when $\Sigma_q = 0$, let us compute $\dot{\Sigma}_q = 0$, which is a necessary condition to remain on the manifold. Substituting (3.3) in $\dot{\Sigma}_q = 0$, we have

$$\boldsymbol{B}^{-1} \hat{\boldsymbol{B}} \boldsymbol{v} + \boldsymbol{B}^{-1} \tilde{\boldsymbol{n}} - \boldsymbol{v}_0 = 0 \tag{3.51}$$

Thus, using (3.9), and making $\boldsymbol{v}_{smc}$ explicit

$$\boldsymbol{v}_{smc} = -\hat{\boldsymbol{B}}^{-1} (\tilde{\boldsymbol{B}} \boldsymbol{v}_0 + \tilde{\boldsymbol{n}}) \tag{3.52}$$

Substituting back in (3.9) and then in (3.3), it is clear that the system will evolve with the nominal dynamics (3.32). □

Therefore, by choosing the manifold (3.46), we simply have to appropriately select $\boldsymbol{v}_0$ to obtain the desired impedance at the end effector, as shown in Sec. 3.3.1 by computing $\boldsymbol{v}_{eq}$.

3.4.3 Super-twisting algorithm

Higher order sliding modes ensure finite time convergence of the sliding vector derivatives up to an order ρ. They have been proposed to alleviate the chattering effect by ensuring continuity of the control variable, and to apply SMC to systems with relative degree $r \geq 2$ [62].

If we consider Σ as a system output, it is straightforward to see that $r = 1$. We have to ensure $\rho \geq r = 1$ in order for the control variable to appear in the sliding vector derivatives, which justifies the feasibility of the first order sliding mode discussed up to this point. Therefore, the lowest sliding mode order required to obtain a continuous control is $\rho = 2$. In this case, the discontinuity is moved onto the control first derivative, mitigating chattering of the actual control.

In the following we apply the super-twisting second order algorithm proposed by Levant [80]. Given the sliding vector dynamics obtained with (3.10)

$$\ddot{\Sigma} = \left(\frac{\partial \dot{\Sigma}}{\partial t} + \frac{\partial \dot{\Sigma}}{\partial \boldsymbol{z}} \dot{\boldsymbol{z}} \right) + \frac{\partial \dot{\Sigma}}{\partial \boldsymbol{v}} \dot{\boldsymbol{v}} \tag{3.53}$$

we have to ensure the boundedness of the terms

$$\left\| \frac{\partial \dot{\Sigma}}{\partial t} + \frac{\partial \dot{\Sigma}}{\partial \boldsymbol{z}} \dot{\boldsymbol{z}} \right\| \leq \Phi, \quad 0 < \underline{\Psi} \leq \left\| \frac{\partial \dot{\Sigma}}{\partial \boldsymbol{v}} \right\| \leq \overline{\Psi} \tag{3.54}$$

The first one basically calls for the reference jerk and the interaction force time derivative, along with system uncertainties, to be bounded in the control input domain. This is reasonable if the reference trajectory is defined at jerk level or if we consider saturated accelerations in a discrete time application. Instead, for the force derivative we are asking to avoid collisions with very stiff environments when selecting a stiff impedance profile, which may provoke a brief detachment from the sliding manifold. The second term simply requires $\|\boldsymbol{B}^{-1}\hat{\boldsymbol{B}}\|$ to be lower and upper bounded, which is reasonable for bounded inertia matrix uncertainties.

Given these assumptions, the control input for the super-twisting ISM can be written as

$$\boldsymbol{v}_{smc}(\Sigma_q) = k_1 \frac{\Sigma_q}{\sqrt{\|\Sigma_q\|}} + k_2 \int_{t_0}^{t} \frac{\Sigma_q}{\|\Sigma_q\|} d\tau \tag{3.55}$$

(see [81] for a general tuning procedure of the gains k_1, k_2). Hence, the overall auxiliary control (3.9) becomes the following:

$$\boldsymbol{v} = \boldsymbol{v}_{eq} + k_1 \frac{\Sigma_q}{\sqrt{\|\Sigma_q\|}} + k_2 \int_{t_0}^{t} \frac{\Sigma_q}{\|\Sigma_q\|} d\tau \tag{3.56}$$

3.5 Sliding mode redundancy

In the previous sections we considered generic n d.o.f. robots, and a sliding vector $\Sigma \in \mathbb{R}^m$, so that $m \leq n$. If a strict inequality holds, the result is that by enforcing such a sliding mode we are controlling only a subspace of the robot dynamics. Indeed, even intuitively, there exist an infinite number of joint configurations that satisfy $\Sigma = 0$, and thus the desired impedance $\boldsymbol{I} = 0$.

Unfortunately this also means that disturbances and uncertainties acting in the null space of the impedance task will not be robustly compensated. Take system (3.10), and rewrite the additive uncertainty as the sum of two terms

$$\boldsymbol{B}^{-1}\boldsymbol{\eta} = \boldsymbol{\delta}_R + \boldsymbol{\delta}_N \;\in \mathbb{R}^n \tag{3.57}$$

where $\boldsymbol{\delta}_N \in \mathcal{N}(\boldsymbol{J})$ is the uncertainty component belonging to the null space of $\boldsymbol{J}$, and $\boldsymbol{\delta}_R \in \mathcal{R}(\boldsymbol{J}^T)$ its orthogonal complement, such that $\mathcal{N}(\boldsymbol{J}) \cup \mathcal{R}(\boldsymbol{J}^T) = \mathbb{R}^n$. Computing $\dot{\Sigma}$ as in (3.13) with (3.57), we have

$$\begin{aligned} \dot{\Sigma} =& \boldsymbol{M}\ddot{\boldsymbol{x}}_r - \boldsymbol{M}\boldsymbol{J}\boldsymbol{B}^{-1}\hat{\boldsymbol{B}}\boldsymbol{v} - \boldsymbol{M}\boldsymbol{J}(\boldsymbol{\delta}_R + \boldsymbol{\delta}_N) + \\ & - \boldsymbol{M}\dot{\boldsymbol{J}}\dot{\boldsymbol{q}} + \boldsymbol{D}\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}\tilde{\boldsymbol{x}} - \hat{\boldsymbol{F}}_e \end{aligned} \tag{3.58}$$

By definition of null space, $\boldsymbol{J}\boldsymbol{\delta}_N = 0$, and the sliding vector dynamics are insensitive to any uncertainty or disturbance $\boldsymbol{\delta}_N \in \mathcal{N}(\boldsymbol{J})$. This implies that also Σ_q will not be able to detect and reject such uncertainties and disturbances via the sliding mode control $\boldsymbol{v}_{smc}(\Sigma_q)$.

In robotic manipulators, redundancy resolution is often performed by projection of a lower priority task in the null space of a higher priority one, either at velocity, acceleration, or torque level. In order to guarantee the robust control of the whole kinematic chain, we follow the same idea by performing a sequential projection of sliding surfaces defined directly in the task space.

Consider a task that we want to execute to solve the redundancy, in the following general form

$$\boldsymbol{J}_i(\boldsymbol{q}, \dot{\boldsymbol{q}})\ddot{\boldsymbol{q}} - \boldsymbol{b}_i(\boldsymbol{q}, \dot{\boldsymbol{q}}) = 0 \tag{3.59}$$

with $\boldsymbol{J}_i \in \mathbb{R}^{m_i \times n}$ is the task Jacobian, and $\boldsymbol{b}_i \in \mathbb{R}^{m_i}$. Using similar arguments as those at the beginning of Sec. 3.3 for impedance control, by integration we can define a sliding vector that, on the sliding manifold, realizes the task and does not depend on joint accelerations (i.e. Prop. 3.3.1 holds)

$$\boldsymbol{\sigma}_i = \int_0^t (\boldsymbol{J}_i \ddot{\boldsymbol{q}} - \boldsymbol{b}_i) d\tau \tag{3.60}$$

If desired, ISM can be applied also to this sliding vector obtaining Σ_i, however, since the chosen nominal controller $\boldsymbol{v}_0 = \boldsymbol{v}_{eq}$ is not a nominal solution to $\dot{\boldsymbol{\sigma}}_i$ in general, Prop. 3.4.1 does not hold and the full form of (3.33) has to be used.

We can redefine Σ_q, ensuring the sliding hierarchy as follows

$$\Sigma_q = \boldsymbol{J}^\dagger \Sigma + \boldsymbol{P}\boldsymbol{J}_i^\dagger \Sigma_i, \quad \Sigma_q \in \mathbb{R}^n \tag{3.61}$$

where $\boldsymbol{P}$ is a null space projector of $\boldsymbol{J}$, such that $\boldsymbol{J}\boldsymbol{P} = 0$. Prop. 3.3.2 still holds for the first term in (3.61) if $\boldsymbol{J}$ is full rank, since $\boldsymbol{J}\Sigma_q = \Sigma$, however this is not necessarily true for the second one. Indeed, we also have that $\Sigma_q = 0 \Rightarrow \Sigma_i = 0$ only if $rank(\boldsymbol{P}\boldsymbol{J}_i^\dagger) = m_i \leq n - m$, that is, if the secondary task is not conflicting with the primary one and can be completed with the remaining degrees of freedom. Nonetheless, we expect to execute the secondary task *at best* to begin with, given the priority. Simultaneously, thanks to the second term of (3.61), we are able to detect and compensate disturbances in a larger subset of $\mathbb{R}^n$.

The projection procedure may be extended to an arbitrary number of tasks p, in order to use up all the degrees of freedom (i.e. $\sum_{i=1}^p m_i \geq n$) and be sensitive to disturbances in all $\mathbb{R}^n$

$$\Sigma_q = \boldsymbol{J}^\dagger \Sigma + \sum_{i=1}^{p} \boldsymbol{P}_{i-1}\boldsymbol{J}_i^\dagger \Sigma_i, \quad \Sigma_q \in \mathbb{R}^n \tag{3.62}$$

where $\boldsymbol{P}_i$ is the null space projector of the augmented Jacobian

$$\bar{\boldsymbol{J}}_i = \begin{bmatrix} \boldsymbol{J}^T & \boldsymbol{J}_1^T & \dots & \boldsymbol{J}_i^T \end{bmatrix}^T, \quad \bar{\boldsymbol{J}}_0 = \boldsymbol{J} \tag{3.63}$$

We mentioned that the ISM can be applied also to the lower priority manifolds, however, as we observed with Prop. 3.4.2, the removal of the reaching phase makes the robot dynamics evolve as dictated by the nominal controller $\boldsymbol{v}_0$. It is clear that if ISM is applied we confer robustness to the system from a time t_0, but we are forced to modify the nominal control in order to satisfy the low priority tasks, while maintaining the optimality of the impedance one.

At this point, we can simply assume to work with the nominal system of integrators (3.32), and modify $\boldsymbol{v}_0$ as if we were considering only the robot

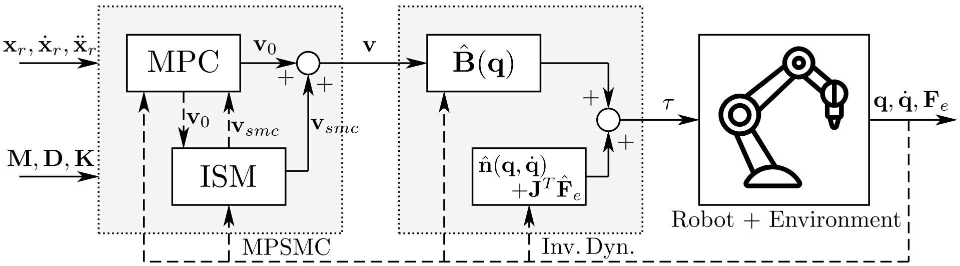


Figure 3.1: *The proposed model predictive sliding mode control scheme.*

kinematics, by applying a standard optimal task redundancy resolution algorithm [82]

$$\boldsymbol{v}_0 = \boldsymbol{v}_{0,p} = \boldsymbol{v}_{eq} + \sum_{i=1}^{p} (\boldsymbol{J}_i \boldsymbol{P}_{i-1})^\dagger (\boldsymbol{b}_i - \boldsymbol{J}_i \boldsymbol{v}_{0,i-1}) \tag{3.64}$$

where $\boldsymbol{v}_{0,i}$ is the optimal nominal control up to the i^{th} task, $\boldsymbol{v}_{0,0} = \boldsymbol{v}_{eq}$.

Note that (3.64) is still a nominal solution to $\dot{\Sigma} = \dot{\boldsymbol{\sigma}}$, so that the considerations made in Sec. 3.3 and 3.4 still hold. In fact, the second term is canceled when pre-multiplied by $\boldsymbol{J}$, since $\boldsymbol{J}(\boldsymbol{J}_i \boldsymbol{P}_{i-1})^\dagger = 0, \forall i = 1 \dots p$, thanks to the null space projector properties.

3.6 Model predictive sliding mode control

The formulation (3.64) of the nominal controller allows the definition of a strict task hierarchy, however it does not allow to consider constraints such as actuators saturation, or even simple joint motion range limitations. To cope with this kind of requirements we employ an optimization-based controller for the computation of a suitable $\boldsymbol{v}_0$ that takes them into account, then it will be adapted to a model predictive formulation in Sec. 3.6.4 to address actuation delays.

The combined MPC and SMC controllers define a global scheme that we call Model Predictive Sliding Mode Controller (MPSMC) (Fig. 3.1). The advantages of both techniques are exploited: the sliding mode component guarantees robust rejection of disturbances via joint space projections of operational sliding surfaces (3.62), whereas the model predictive component ensures the feasibility of the constrained motion considering only the ideal feedback-linearized system, and therefore just the robot kinematics.

In the following we outline the optimization-based model predictive controller. We show how it is adapted to solve the redundancy by enforcing a task hierarchy, and to work in conjunction with the SMC presented in the previous section.

3.6.1 Hierarchy preserving MPC

If the sliding mode is enforced by the SMC, the system evolves as if there were no uncertainties, with dynamics given by the chain of integrators (3.32), controlled by $\boldsymbol{v}_0$ (Prop. 3.4.2).

The optimization problem must be designed so that, if no constraint is active, the obtained control $\boldsymbol{v}_0$ is a nominal solution to $\dot{\boldsymbol{\sigma}}$. In order to maintain an efficient quadratic programming (QP) formulation of the problem, and avoid the non-linearities due to the Jacobian dependence on future control inputs, without loss of generality we select for the time being a prediction horizon $N = 0$, that also decreases the computational complexity for real-time implementation.

Given these considerations, the hierarchical formulation of MPC can be expressed in a form that is analogous to the one used in Sec. 2.2 to obtain the hierarchical teleoperation controller, whereas in this case we must consider the derivative of the sliding manifolds in the cost functions. Indeed, the optimization problem must be designed so that, if no constraint is active, the obtained control is a nominal solution to $\dot{\boldsymbol{\sigma}}$. Therefore, we define the following QP

$$\boldsymbol{v}_{0,0} = \arg\min_{\boldsymbol{v}_0} \|\dot{\boldsymbol{\sigma}}\|^2_{\boldsymbol{Q}} \tag{3.65a}$$

$$s.t. \quad \underline{\boldsymbol{\tau}} \leq \boldsymbol{\tau}(\boldsymbol{v}_0, \boldsymbol{v}_{smc}) \leq \overline{\boldsymbol{\tau}} \tag{3.65b}$$

$$\underline{\boldsymbol{q}} \leq \boldsymbol{q} \leq \overline{\boldsymbol{q}} \tag{3.65c}$$

$$\boldsymbol{A}\boldsymbol{v}_0 \leq \boldsymbol{b} \tag{3.65d}$$

If no constraint is active, by minimizing the squared norm of the sliding vector derivative evaluated with the nominal dynamics, we obtain the nominal solution (3.15). Otherwise, a compromise has to be reached in order to satisfy torque saturation constraints (3.65b), joint limits (3.65c), and other inequality constraints (3.65d) (e.g. forbidden workspace regions). $\boldsymbol{Q} > 0$ is a weight matrix that penalizes sliding vector components upon constraint activation, e.g. if it is not possible to obtain the desired impedance in all directions, one may decide to bias the solution to favor one that is more relevant to the task.

As mentioned at the end of Sec. 3.5, if we apply ISM to the lower priority manifolds $\boldsymbol{\sigma}_i$, the nominal controller has to be modified in order to solve the redundancy and satisfy these additional tasks (3.64). To do so in an optimization-based architecture, we adopt the approach in [34, 83] and used in Sec. 2.2 and 2.3, that enforces the task hierarchy with a cascade of QP optimizations. Therefore, for each task we may write a new QP as follows

$$\boldsymbol{v}_{0,i} = \arg\min_{\boldsymbol{v}_0} \|\dot{\boldsymbol{\sigma}}_i\|^2_{\boldsymbol{Q}_i} \tag{3.66a}$$

$$s.t. \quad \textit{constraints up to task } i-1 \tag{3.66b}$$

$$\boldsymbol{J}_{i-1}\boldsymbol{v}_0 = \boldsymbol{J}_{i-1}\boldsymbol{v}_{0,i-1} \tag{3.66c}$$

Each level of the optimization tries to minimize its sliding vector derivative aiming to obtain its nominal solution, while respecting the constraints defined at the higher priority levels (3.66b), as well as adding a constraint that guarantees the optimality of the new solution $\boldsymbol{v}_{0,i}$ with respect to $\boldsymbol{v}_{0,j}$ $j = 0...i-1$ (3.66c).

In the end, we can select the hierarchy consistent solution

$$\boldsymbol{v}_0 = \boldsymbol{v}_{0,p} \tag{3.67}$$

In absence of constraint activation, this procedure provides the same result found via the standard projection approach (3.64).

3.6.2 Control torque saturation

In order to limit the maximum torque requested by the overall MPSMC, the sliding mode control $\boldsymbol{v}_{smc}$ has to be taken into account in the optimization formulation (3.65b). Failure to do so may generate a nominal control that satisfies torque constraints, however the subsequent addition of $\boldsymbol{v}_{smc}$ can produce a total control torque that exceeds the prescribed bounds.

Since $\boldsymbol{v}_{smc}$ inherently acts as a compensator of the uncertain dynamics, from the optimization point of view, we can just consider the nominal estimated robot dynamics to be the true one. Using (3.2) and (3.9), we get the constraint

$$\underline{\boldsymbol{\tau}} - \overbrace{\hat{\boldsymbol{B}}\boldsymbol{v}_{smc}}^{\boldsymbol{\tau}_{smc}} \leq \hat{\boldsymbol{B}}\boldsymbol{v}_0 + \hat{\boldsymbol{n}} + \boldsymbol{J}^T\hat{\boldsymbol{F}}_e \leq \overline{\boldsymbol{\tau}} - \overbrace{\hat{\boldsymbol{B}}\boldsymbol{v}_{smc}}^{\boldsymbol{\tau}_{smc}} \tag{3.68}$$

Depending on the sign of $\boldsymbol{\tau}_{smc}$, the sliding mode can prevent the selection of a nominal control that would produce a too large torque. On the other hand, it may allow to use a higher nominal torque, when the uncertainty implicitly works in favor of the requested motion, since it is afterwards compensated by the sliding mode. In this case, we are basically exploiting the uncertainties to perform more dynamically demanding tasks.

3.6.3 Sliding manifold adaptation

Upon activation of a constraint, the optimization solution will be different from the nominal one, and the cost function will be non-zero. Indeed, we will have $\dot{\boldsymbol{\sigma}} \neq 0$.

It is clear that a non-zero derivative will generate a detachment from the sliding manifold, which will be detected by the SMC. The constraint is interpreted as a disturbance to be compensated, thus causing its violation, and making the MPC architecture useless.

To avoid this, we need a way to communicate to the sliding mode controller the change in control policy to accommodate for the constraints. The idea is to progressively adjust the sliding surface based on the optimization cost, avoiding interference with the SMC. Naturally, the price to be paid is that we will not be able to ensure the desired end effector impedance anymore.

The drift from the sliding manifold, and therefore the required adaptation, can be computed by integration

$$\Delta(t) = \int_{t_0}^{t} \dot{\boldsymbol{\sigma}} d\tau + \boldsymbol{\sigma}(t_0) \tag{3.69}$$

where $\dot{\boldsymbol{\sigma}}$ has been evaluated with the nominal dynamics (3.32) and the control (3.67), giving as a result

$$\begin{aligned} \Delta(t) =& \boldsymbol{M}\left(\dot{\boldsymbol{x}}_r - \boldsymbol{J}\int_{t_0}^{t} \boldsymbol{v}_0 d\tau\right) + \boldsymbol{D}\tilde{\boldsymbol{x}} + \boldsymbol{K}\int_{t_0}^{t} \tilde{\boldsymbol{x}} d\tau \\ &- \int_{t_0}^{t} \hat{\boldsymbol{F}}_e d\tau + \boldsymbol{\sigma}(t_0) \end{aligned} \tag{3.70}$$

The new sliding manifold is then the following

$$\Sigma(t) = \boldsymbol{\sigma}(t) - \Delta(t) \tag{3.71}$$

Remarkably, the previous equation provides the same result that would be obtained by application of the standard ISM (3.33), (3.34). Indeed, when a constraint activates, the hypothesis of Prop. 3.4.1 does not hold anymore, and we are left with the full ISM formulation.

From this we can come to the following conclusion, valid also for the lower priority tasks. If no ISM is employed in the SMC formulation, the adaptation here proposed is mandatory for each manifold to ensure constraint satisfaction, albeit at the expense of task execution. Instead, if ISM is already implemented by taking $\boldsymbol{v}_0 = \boldsymbol{v}_{0,p}$, no adaptation is required, and we are guaranteed to satisfy the constraints and lay on the sliding manifold since t_0, thus enforcing the desired impedance and executing at best the remaining tasks when no constraint is active.

3.6.4 Control torque delay

Although the proposed controller is robust against matched uncertainties, it shows its weakness when unmatched effects are concerned, such as actuation delays. These may induce oscillatory behaviors of the system and limit cycles around the sliding manifold. To tackle this problem we will make the assumption that the dynamics between the desired control torque and the true one can be reasonably approximated as a pure delay.

First, we discretize the system and perform a prediction of state and sliding vector. By doing so, we exploit the model predictive optimization procedure to compute a suitable control $\boldsymbol{v}_0$ that can cope with the delay. The same prediction is also employed in the evaluation of the sliding manifold, in order to compute a predictive sliding mode control $\boldsymbol{v}_{smc}$.

Since all we know are the estimated parameters (3.2), the best that we can do to derive a predictor is to consider the perfectly feedback-linearized dynamics. The system becomes a chain of integrators controlled by the auxiliary control $\boldsymbol{v}$, and can be discretized as follows

$$\begin{aligned} \boldsymbol{z}_{k+1} &= \boldsymbol{H}\boldsymbol{z}_k + \boldsymbol{G}v_{k-d} = \\ &= \begin{bmatrix} 1 & T_s & \frac{T_s^2}{2} \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \int \boldsymbol{q}_k \\ \boldsymbol{q}_k \\ \dot{\boldsymbol{q}}_k \end{bmatrix} + \begin{bmatrix} \frac{T_s^3}{6} \\ \frac{T_s^2}{2} \\ T_s \end{bmatrix} \boldsymbol{v}_{k-d} \end{aligned} \tag{3.72}$$

T_s is the discretization sampling time, index k designates the time instant, and v_{k-d} the control input, where d is the estimated number of delay steps.

Once the relation between joint and operational space variables has been made explicit, the sliding vector derivative $\dot{\boldsymbol{\sigma}}$ at time k, evaluated with the delayed dynamics (3.72), becomes

$$\dot{\boldsymbol{\sigma}}_k = \Theta \left(\dot{\mathbf{x}}_{r,k} - \begin{bmatrix} \mathcal{T}(\boldsymbol{q}_k) \\ \boldsymbol{J}_k \dot{\boldsymbol{q}}_k \\ \dot{\boldsymbol{J}}_k \dot{\boldsymbol{q}}_k + \boldsymbol{J}_k \boldsymbol{v}_{0,k-d} \end{bmatrix} \right) - \hat{\boldsymbol{F}}_{e,k} \tag{3.73}$$

where $\Theta = [\boldsymbol{K}\ \boldsymbol{D}\ \boldsymbol{M}]$, $\mathbf{x}_{r,k} = [\int \boldsymbol{x}_{r,k}\ \boldsymbol{x}_{r,k}\ \dot{\boldsymbol{x}}_{r,k}]^T$, $\boldsymbol{J}_k = \boldsymbol{J}(\boldsymbol{q}_k)$, $\dot{\boldsymbol{J}}_k = \dot{\boldsymbol{J}}(\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k)$, and $\mathcal{T}(\boldsymbol{q}_k) = \boldsymbol{x}_k$ is the robot forward kinematics.

The first time instant where the sliding vector derivative depends on the current control input is $k + d$. In fact, $\dot{\boldsymbol{\sigma}}_{k+d}(\boldsymbol{q}_{k+d}, \dot{\boldsymbol{q}}_{k+d}, \boldsymbol{v}_k, \dot{\mathbf{x}}_{r,k+d}, \hat{\boldsymbol{F}}_{e,k+d})$.

Unfortunately, since these quantities are unknown at time k, we can only perform a d-step prediction by propagating the dynamics (3.72)

$$\boldsymbol{z}^*_{k+d} = \boldsymbol{H}^d \boldsymbol{z}_k + \sum_{j=0}^{d-1} \boldsymbol{H}^{d-1-j} \boldsymbol{G} v_{k-d+j} \tag{3.74}$$

where $(\cdot)^*$ indicates that the quantity is a prediction and not the true one. Similarly, for the vector $\dot{\mathbf{x}}_{r,k+d}$, if the trajectory is pre-planned we already know the future reference and we can use it, otherwise we can assume it to be constant $\dot{\mathbf{x}}^*_{r,k+d} = \dot{\mathbf{x}}_{r,k}$. Since we do not have higher order information on the force, we can simply select $\hat{\boldsymbol{F}}^*_{e,k+d} = \hat{\boldsymbol{F}}_{e,k}$.

Therefore, we obtain the predicted sliding variable derivative $\dot{\boldsymbol{\sigma}}^*_{k+d}$, that depends on $\boldsymbol{v}_0$ at the current instant k

$$\dot{\boldsymbol{\sigma}}^*_{k+d} = \Theta \left(\dot{\mathbf{x}}^*_{r,k+d} - \begin{bmatrix} \mathcal{T}(\boldsymbol{q}^*_{k+d}) \\ \boldsymbol{J}^*_{k+d} \dot{\boldsymbol{q}}^*_{k+d} \\ \dot{\boldsymbol{J}}^*_{k+d} \dot{\boldsymbol{q}}^*_{k+d} + \boldsymbol{J}^*_{k+d} \boldsymbol{v}_{0,k} \end{bmatrix} \right) - \hat{\boldsymbol{F}}^*_{e,k+d} \tag{3.75}$$

Using $\dot{\boldsymbol{\sigma}}^*_{k+d}$ in (3.65) allows to consider the actuation delay when computing the nominal control $\boldsymbol{v}_0$.

We can do the same for the sliding variable

$$\boldsymbol{\sigma}^*_{k+d} = \Theta \left(\mathbf{x}^*_{r,k+d} - \begin{bmatrix} \int \mathcal{T}(\boldsymbol{q}^*_{k+d}) \\ \mathcal{T}(\boldsymbol{q}^*_{k+d}) \\ \boldsymbol{J}^*_{k+d} \dot{\boldsymbol{q}}^*_{k+d} \end{bmatrix} \right) - \int_0^k \hat{\boldsymbol{F}}^*_{e,k+d} d\tau \tag{3.76}$$

Thus, following the discussion of Sec. 3.3 and 3.4, this leads to $\Sigma^*_{q,k+d}$, which is used to obtain the predictive sliding mode control $\boldsymbol{v}_{smc}(\Sigma^*_{q,k+d})$ (3.55). The same should be done for the other sliding surfaces $\boldsymbol{\sigma}_i$ and their derivatives, as well as the constraints in the MPC controller, in order to obtain a consistent formulation.

Applying the prediction also to the inverse dynamics (3.2), we obtain the following predictive control torque:

$$\boldsymbol{\tau}_k = \hat{\boldsymbol{B}}(\boldsymbol{q}^*_{k+d}) \boldsymbol{v}_k + \hat{\boldsymbol{n}}(\boldsymbol{q}^*_{k+d}, \dot{\boldsymbol{q}}^*_{k+d}) + \boldsymbol{J}^{*T}_{k+d} \hat{\boldsymbol{F}}^*_{e,k+d} \tag{3.77a}$$

$$\boldsymbol{v}_k = \boldsymbol{v}_0(\dot{\boldsymbol{\sigma}}^*_{k+d}) + \boldsymbol{v}_{smc}(\Sigma^*_{q,k+d}) \tag{3.77b}$$

3.7 Simulations

To validate the approach, in MATLAB we performed a series of simulations on a $n = 4$ d.o.f. planar manipulator with revolute joints (Fig. 3.2).

Table 3.1: *4 d.o.f. simulations: robot dynamic parameters.*

True (Est.) value	Link 1	Link 2	Link 3	Link 4
$m\ (kg)$	10 (12)	10 (8)	6 (7.2)	6 (7.2)
$I\ (kgm^2)$	3 (3.6)	3 (2.4)	2 (2.4)	2 (2.4)
$l\ (m)$	1 (1.2)	1 (0.8)	1 (1.2)	1 (1.2)
$a\ (m)$	2 (2)	2 (2)	2 (2)	2 (2)
$b\ (Ns)$	25 (20)	25 (20)	50 (40)	50 (40)
$\tau_c\ (Nm)$	5 (4)	5 (4)	15 (12)	15 (12)
$\tau_s\ (Nm)$	20 (16)	20 (16)	40 (32)	40 (32)
$\alpha\ (s/m)$	15 (12)	15 (12)	15 (12)	15 (12)

We assumed knowledge of the kinematic parameters (link length a), but we introduced a severe 20% uncertainty on the other ones, namely link mass m, link inertia I, and link CoM position l. We considered gravity to be present in the $-y$ direction, and we introduced joint friction utilizing a viscous Stribeck model

$$\tau_f = b\dot{q} + \left(\tau_c + (\tau_s - \tau_c)^{-\alpha|\dot{q}|}\right) \operatorname{sgn}(\dot{q}) \tag{3.78}$$

with b the viscosity coefficient, τ_c and τ_s the Coulomb and static friction respectively, while α is a parameter that tunes the slope in Stribeck friction range. A list of the values of true and estimated parameters is given in Table 3.1.

In all tests the objective is to either follow a point-to-point trajectory or regulate the robot end effector position $\boldsymbol{x} \in \mathbb{R}^{m_0}$ to a given set point, with the desired impedance (Task 0, $m_0 = 2$). The redundancy is resolved by regulating the orientation $\phi \in \mathbb{R}^{m_1}$ of the end effector, so that the last link is parallel to the x axis (Task 1, $m_1 = 1$), and the position $\boldsymbol{x}_2 \in \mathbb{R}^{m_2}$ of the third joint to a fixed point (Task 2, $m_2 = 2$), in a descending priority order (see Fig. 3.2). The tasks and the respective sliding variables have been defined as in (3.59). For example Task 1 can be written as follows

$$\boldsymbol{J}_1\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_1\dot{\boldsymbol{q}} + K_{d1}\dot{\phi} - K_{p1}e_\phi = \boldsymbol{J}_1\ddot{\boldsymbol{q}} - b_1 = 0 \tag{3.79}$$

where ϕ is the orientation angle, $e_\phi = \phi_r - \phi$ the regulation error, $\boldsymbol{J}_1 \in \mathbb{R}^{m_1 \times n}$ the Jacobian relating joint velocities to angle derivative. K_{p1}, K_{d1} are PD gains, that can be interpreted as stiffness and damping terms, however external forces are considered only for the higher priority impedance task. Table 3.2 gives the relevant parameters for each task, as well as the respective initial and target values for the simulations.

Table 3.2: *4 d.o.f. simulations: task parameters, initial and target values.*

Task i	$\boldsymbol{M}$	$\boldsymbol{D}$ / $\boldsymbol{K}_{di}$	$\boldsymbol{K}$ / $\boldsymbol{K}_{pi}$	Initial	Target
T_0: $\boldsymbol{x}$	$1\boldsymbol{I}_2$	$4\boldsymbol{I}_2$	$4\boldsymbol{I}_2$	$\begin{bmatrix} 0.897 \\ -0.518 \end{bmatrix} m$	$\begin{bmatrix} 2 \\ 3 \end{bmatrix} m$
T_1: ϕ	1	4	4	$-1.833\, rad$	$0\, rad$
T_2: $\boldsymbol{x}_2$	$1\boldsymbol{I}_2$	$10\boldsymbol{I}_2$	$10\boldsymbol{I}_2$	$\begin{bmatrix} 0 \\ 2.828 \end{bmatrix} m$	$\begin{bmatrix} -1 \\ 2.5 \end{bmatrix} m$

$\boldsymbol{I}_2$ is the 2-by-2 identity matrix

Unless otherwise noted, for the sliding mode control $\boldsymbol{v}_{smc}$ the super-twisting integral formulation (3.55) has been used, with gains $k_1 = 15$, $k_2 = 50$. Moreover, to further smoothen the discontinuity and reduce chattering, we approximated the unit vector computation with the following expression

$$\frac{\Sigma_q}{\|\Sigma_q\|} \approx \frac{\Sigma_q}{\|\Sigma_q\| + 10^{-3}} \tag{3.80}$$

For the nominal control $\boldsymbol{v}_0$ we employed the proposed MPC architecture (3.65), (3.66), with $\boldsymbol{Q}_i = 1, \; i = 0, 1, 2$. The overall control scheme is discretized with a sampling time $T_s = 4ms$.

In all regulation and tracking tests the position error is calculated with respect to the evolution of a system with the exact desired dynamics. Furthermore, in tracking tests, the reference trajectory has been generated with a trapezoidal velocity profile (TVP) with maximum velocity $\dot{x}_{max} = 0.6m/s$ and maximum acceleration $\ddot{x}_{max} = 0.3m/s^2$.

3.7.1 Impedance control/MPSMC comparison

We first compared the performance of our approach with that obtainable with standard impedance control (Sec. 3.2). To achieve comparable results we did not consider any constraint in our architecture so that our control becomes equivalent to a sliding mode controller with the nominal control (3.64).

Fig. 3.2 shows the starting and final robot configurations for the two control strategies, along with each joint trajectory. It is clear that the standard approach is heavily affected by uncertainties, while with the proposed one we are able to reach the desired end effector configuration. Fig. 3.3 and 3.4 give a quantitative evaluation for the regulation case. If we look at the top two plots, the standard impedance controller (blue) exhibits steady state error, while the proposed approach (dashed green) not only reaches

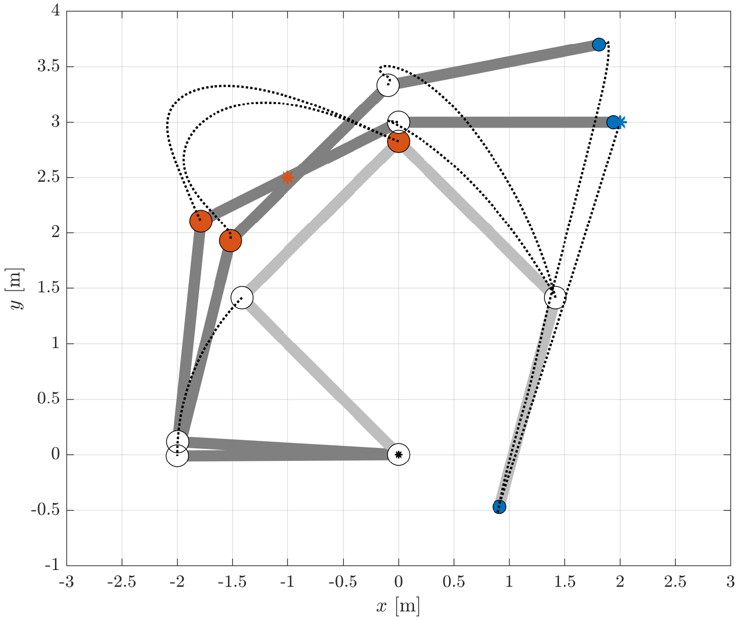


Figure 3.2: *Robot configuration for the first simulation. The robot base and first joint are centered in* $(0, 0)$*. Light gray: the robot initial configuration. Dark gray: the robot final configurations. In blue and red, the robot end effector and third joint, with the respective target positions. The proposed controller is able to regulate the robot to the desired configuration, while the impedance controller visibly fails.*

the desired target, but is able to do so with the desired impedance dynamics (black dots). In Fig. 3.5, we are able to achieve small errors for both regulation and tracking, although we employed very small gains, while the standard controller shows inferior performance.

Comparing our strategy with and without ISM (red dash-dot), it shows how at the beginning the absence of the integral sliding mode gives rise to a reaching phase, with the end effector not behaving as desired during the transient. This is also visible in Fig. 3.6, where with the ISM we start already on the manifold and thus with the desired impedance (left), while without integral strategy we have a transient with $\dot{\boldsymbol{\sigma}}$ reaching high values, and therefore an incorrect impedance profile.

In Fig. 3.5, the comparison between the acceleration profiles for the first and fourth joints outlines the role of the sliding mode control $\boldsymbol{v}_{smc}$, to balance the uncertainties so that the real system behaves like the nominal one considered by the MPC component. Indeed, $\ddot{\boldsymbol{q}} \approx \boldsymbol{v}_0$ for the plots on the left, with the spikes mostly due to change in sign of the velocity, and

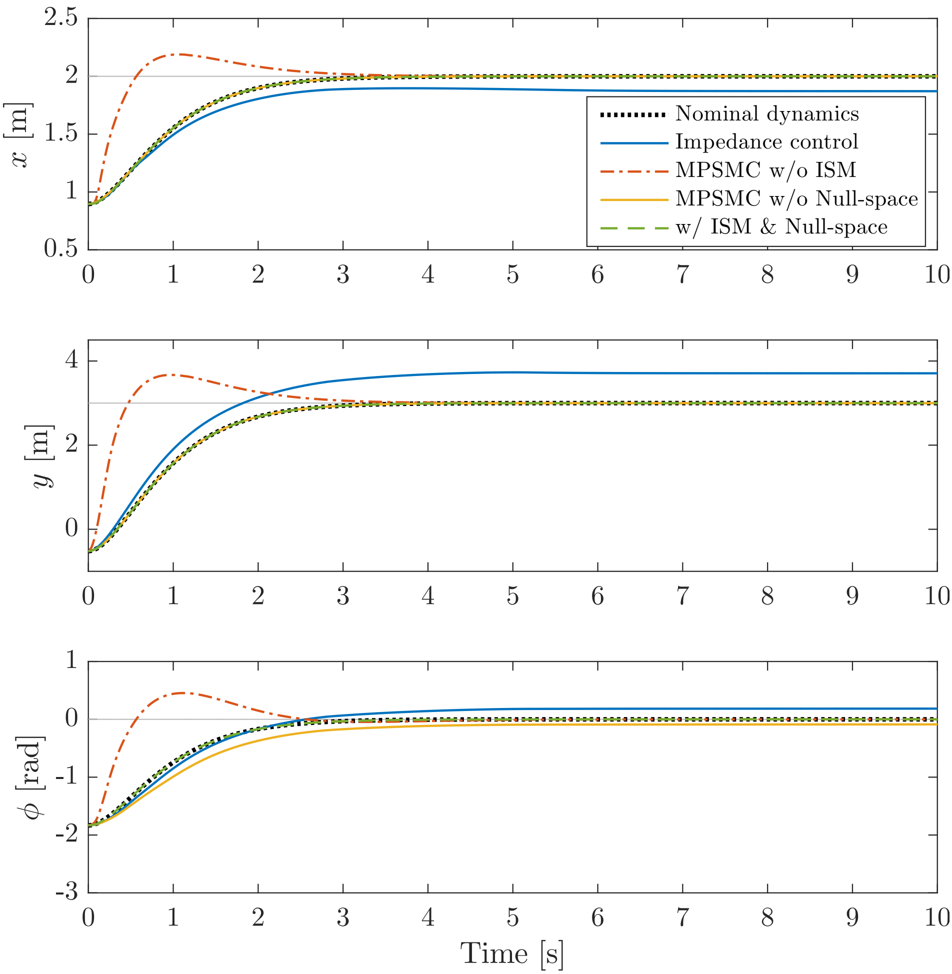


Figure 3.3: *Comparison of the proposed controller with and without ISM and null-space sliding mode, against the standard impedance controller. End effector Cartesian coordinates. The black dotted nominal dynamics line represents the theoretical desired task evolution.*

thus high non-linearity due to friction.

The benefits of the sliding mode redundancy scheme detailed in Sec. 3.5 are visible when looking at the lower priority tasks in Fig. 3.3 and 3.4. Without null-space projection of the manifolds (yellow), the orientation task presents an error both during transient and at steady state, although

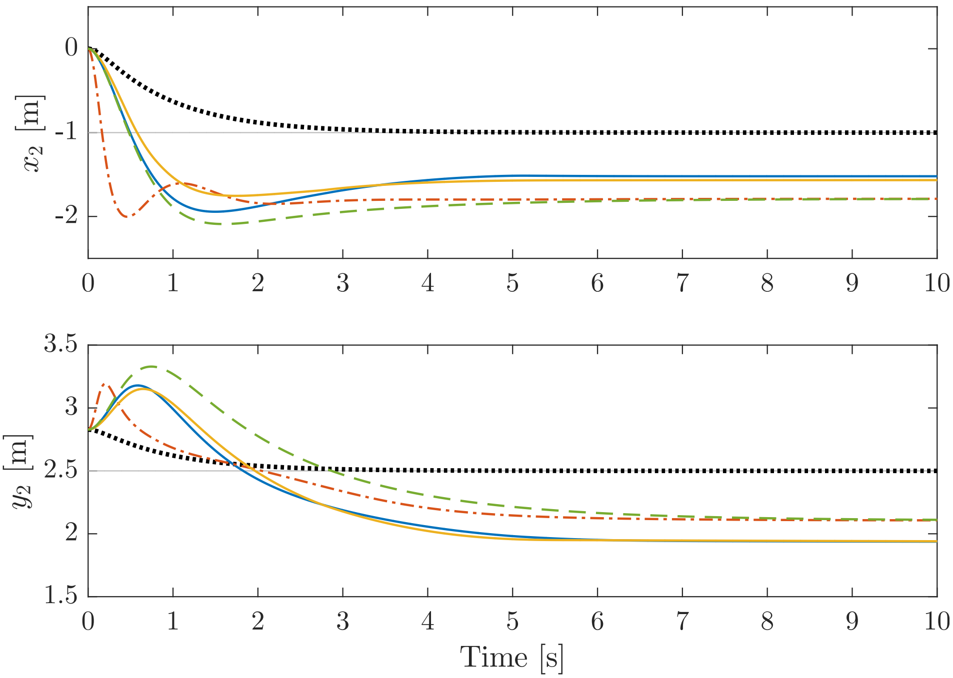


Figure 3.4: *Comparison of the proposed controller with and without ISM and null-space sliding mode, against the standard impedance controller. Cartesian coordinates of the third robot joint. The black dotted nominal dynamics line represents the theoretical desired task evolution.*

the impedance task is perfectly executed. On the other hand, for $\boldsymbol{x}_2$ we understandably do not reach the desired target with any controller due to the conflicts with the higher priority tasks ($\sum m_i = 5 > n = 4$). Nonetheless our formulation still rejects the disturbances in all of the configuration space.

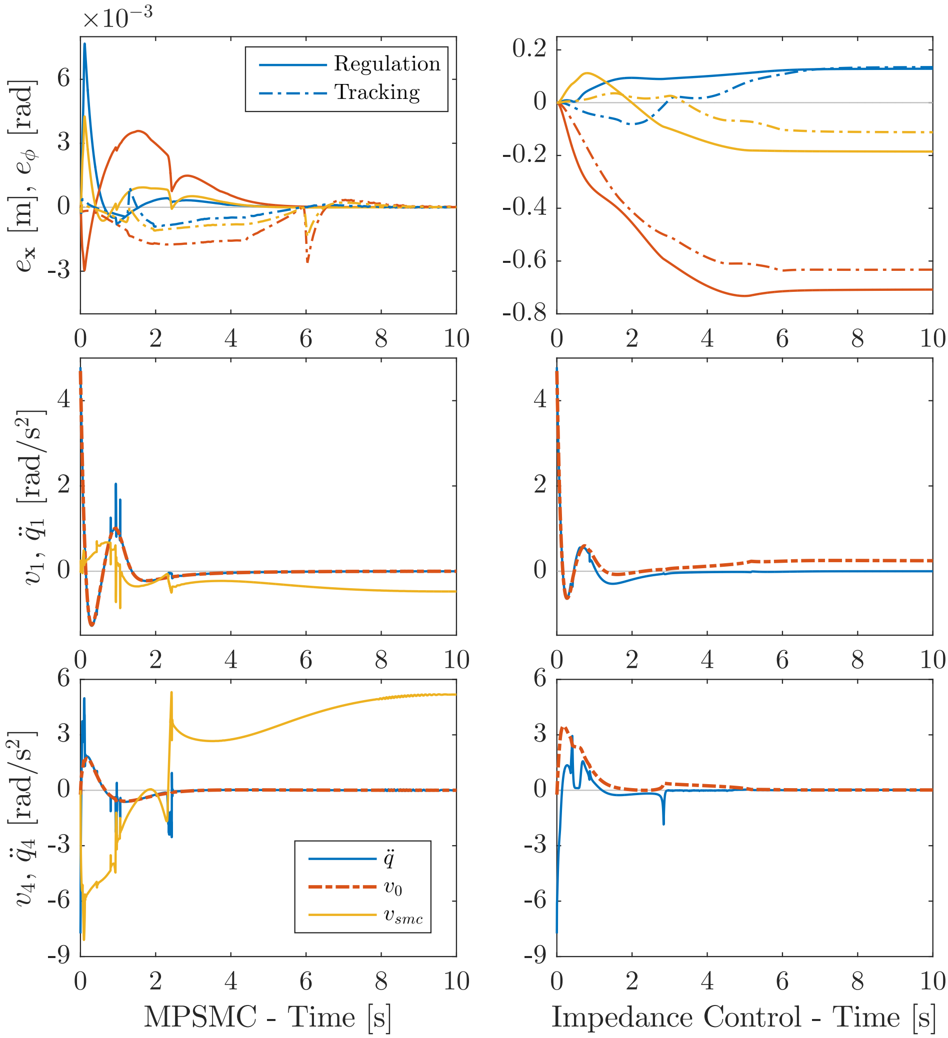


Figure 3.5: *Comparison between the proposed approach and standard impedance control. Top: the task error for both regulation (solid) and tracking (dashed). Bottom: joint accelerations, nominal, and sliding mode controls.*

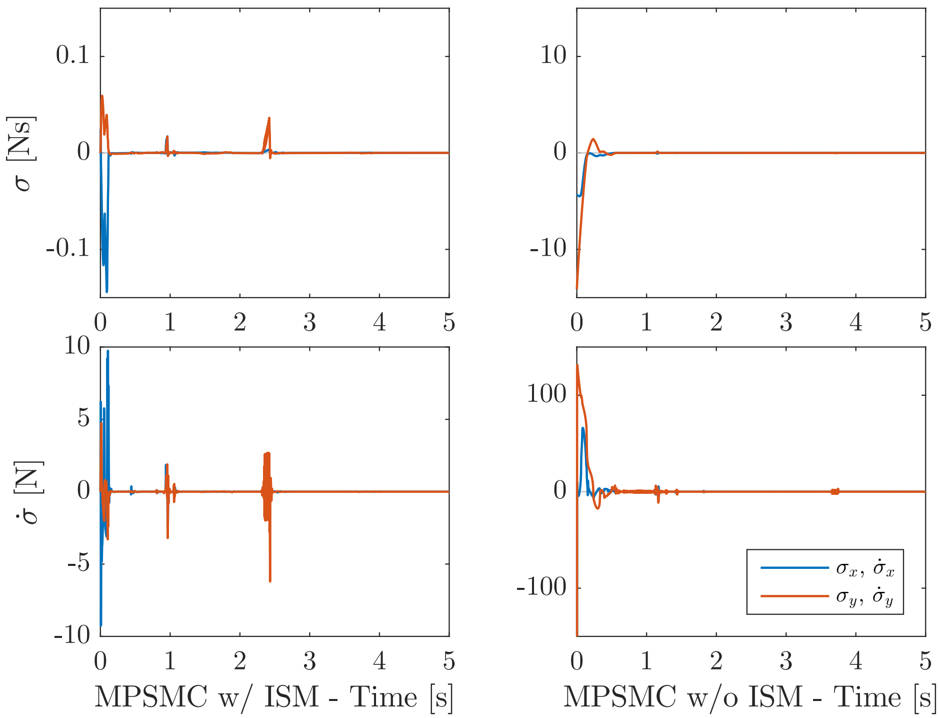


Figure 3.6: *Comparison of the proposed approach with and without ISM. Top: sliding manifold for the impedance task. Bottom: sliding manifold derivative, and thus impedance equation.*

3.7.2 Interaction with the environment

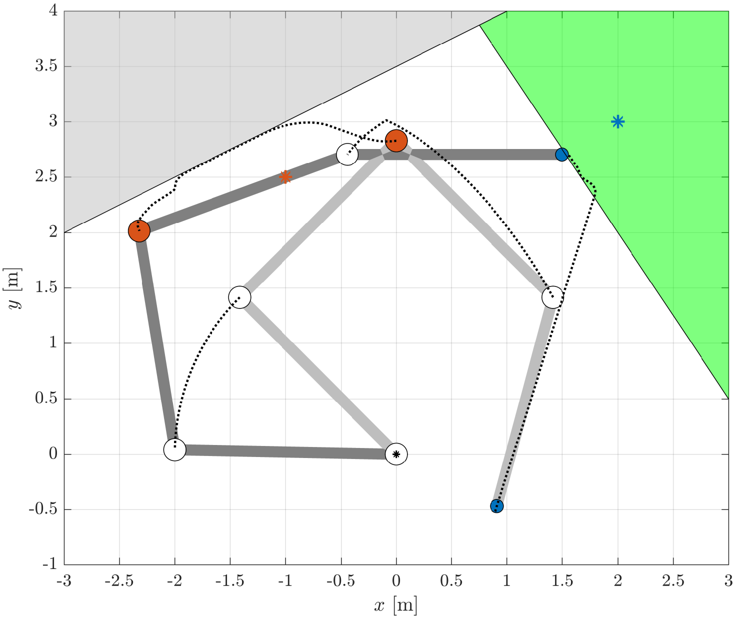


Figure 3.7: *Robot configuration during the simulation with environment interaction. Gray: the forbidden area. Green: the environment wall. The dotted lines indicate the trajectory of each joint and the end effector.*

In a second simulation, we tested the algorithm in presence of external interaction. We added a wall in the $x-y$ plane with stiffness coefficient $K_e = 100N/m$ and damping $D_e = 350Ns/m^2$ so that the end effector has to establish contact to reach the target position. The generated force in the direction normal to the surface is given by

$$F_e = \begin{cases} -K_e\delta & \text{if } \delta > 0,\ \dot{\delta} \leq 0 \\ -K_e\delta - D_e\delta\dot{\delta} & \text{if } \delta > 0,\ \dot{\delta} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.81}$$

where δ is the penetration of the surface. Moreover, we added a constraint in the optimization in the form of a virtual wall (3.65d), to define a forbidden region that the robot has to avoid with the redundant degrees of freedom.

Fig. 3.7 shows the 2D plane with the robot trajectory and the environment (green) as well as the forbidden area (gray). The robot correctly uses the redundant degrees of freedom to *slide* on the perimeter of the virtual

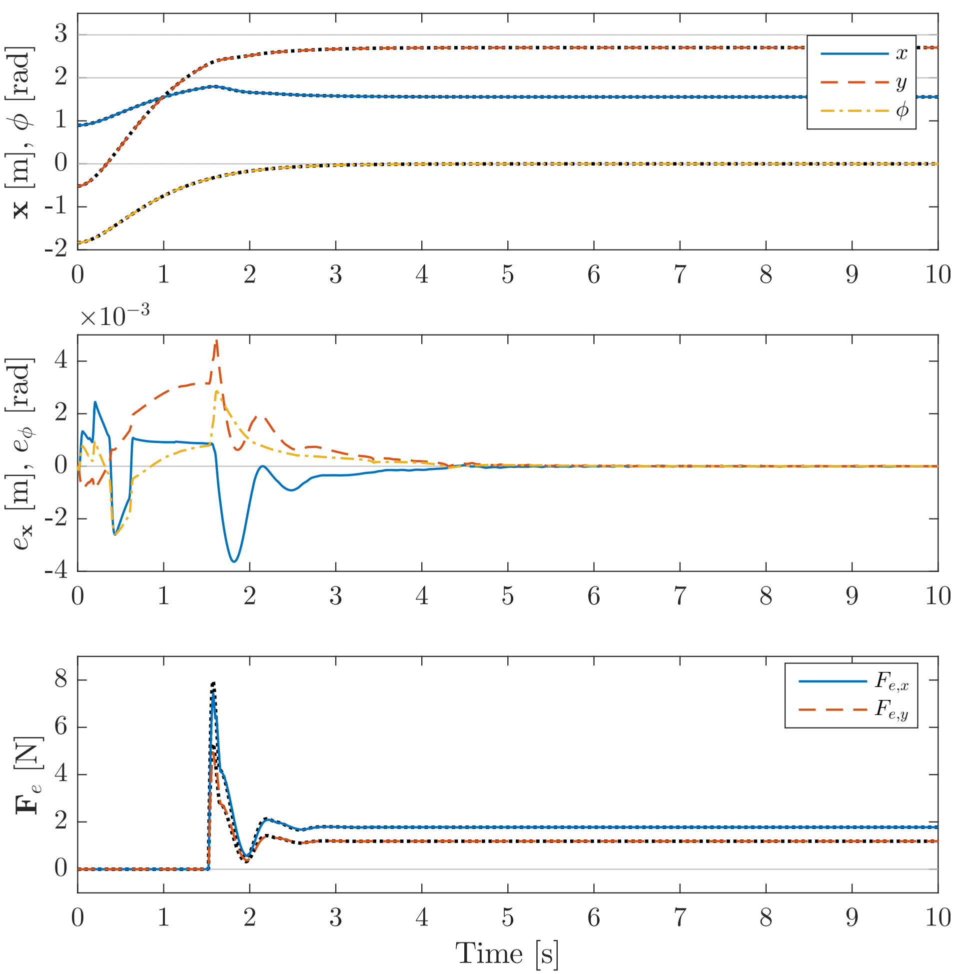


Figure 3.8: *Second simulation: regulation. Top: position and orientation. Middle: error with respect to the desired dynamics. Bottom: contact force.*

wall at the expense of the third task, without violating the constraint. The second task, instead, can still be accomplished, maintaining the horizontal orientation. From Fig. 3.8 and 3.9, we see how the end effector moves as requested, keeping a small error with respect to the theoretical evolution (dotted black), even when contact with the environment is established, both for tracking and regulation. The requested end effector impedance is displayed also during contact, keeping bounded interaction forces.

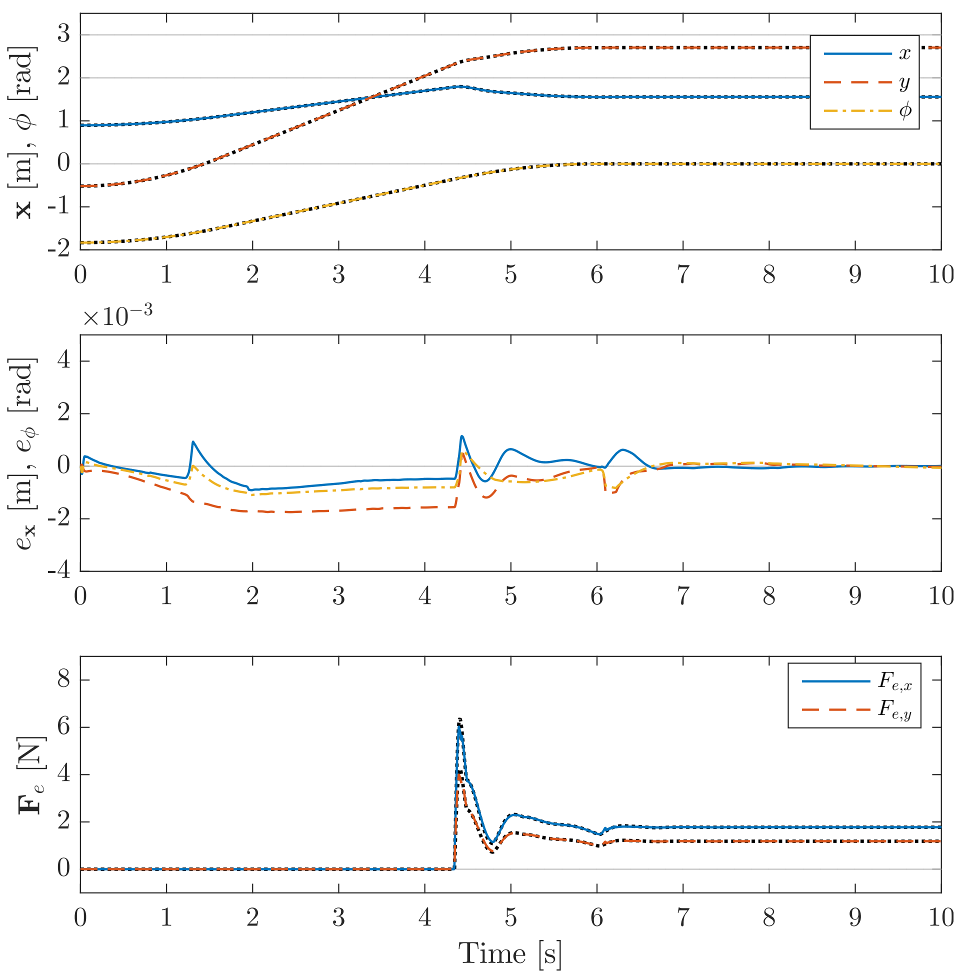


Figure 3.9: *Second simulation: tracking. Top: position and orientation. Middle: error with respect to the desired dynamics. Bottom: contact force.*

The present and the previous simulation illustrate how we can simultaneously achieve accurate regulation and tracking performance in free motion, while exhibiting the desired impedance once contact is achieved. We are able to do so, even when very small control/impedance gains are used and in presence of unknown dynamics.

3.7.3 Torque saturation and joint limits

A third simulation consisted in the introduction of torque saturation and joint limits in the optimization (3.65b), (3.65c). In Fig. 3.10, we compare the system behavior with (red) and without (blue) the compensation of $\boldsymbol{\tau}_{smc}$ (3.68). If the sliding mode torque contribution is not considered, the SMC component tries to compensate the error that would appear due to the saturation, inevitably requesting a control torque much higher than the prescribed bounds. On the other hand, if $\boldsymbol{\tau}_{smc}$ is explicitly taken care of, the overall torque remains in the requested range, at the cost of an error arising with respect to ideal unsaturated dynamics. Furthermore, there is little to no chattering thanks to the second order sliding mode formulation (3.55) coupled with the smoothened unit vector function (3.80). Note that some chattering still remains around $3s$ in both cases, this is however due to the high friction nonlinearity at low velocity when the robot is about to stop.

Similarly with joint limits in Fig. 3.11, if the adaptation scheme of Sec. 3.6.3 is not employed (blue), $\boldsymbol{v}_0$ and $\boldsymbol{v}_{smc}$ generate conflicting control inputs. The first one tries to avoid constraint violation, while the second one tries to ensure adherence to the sliding manifold, inevitably producing an erratic behavior of the system, with possible constraint violations (bottom plot). If the sliding surface is adapted (red), the constraints are fulfilled with a small impedance error, thanks to the exploitation of the robot redundancy.

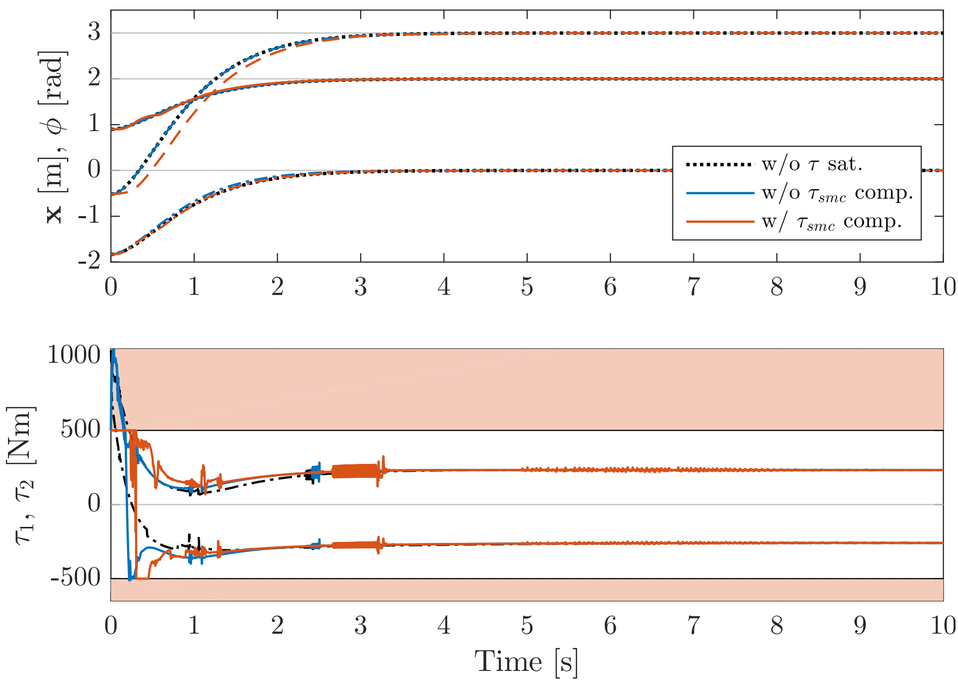


Figure 3.10: *Simulation with torque saturation at* $500Nm$. *Top: end effector position and orientation. Bottom: torques for the first and second joint.*

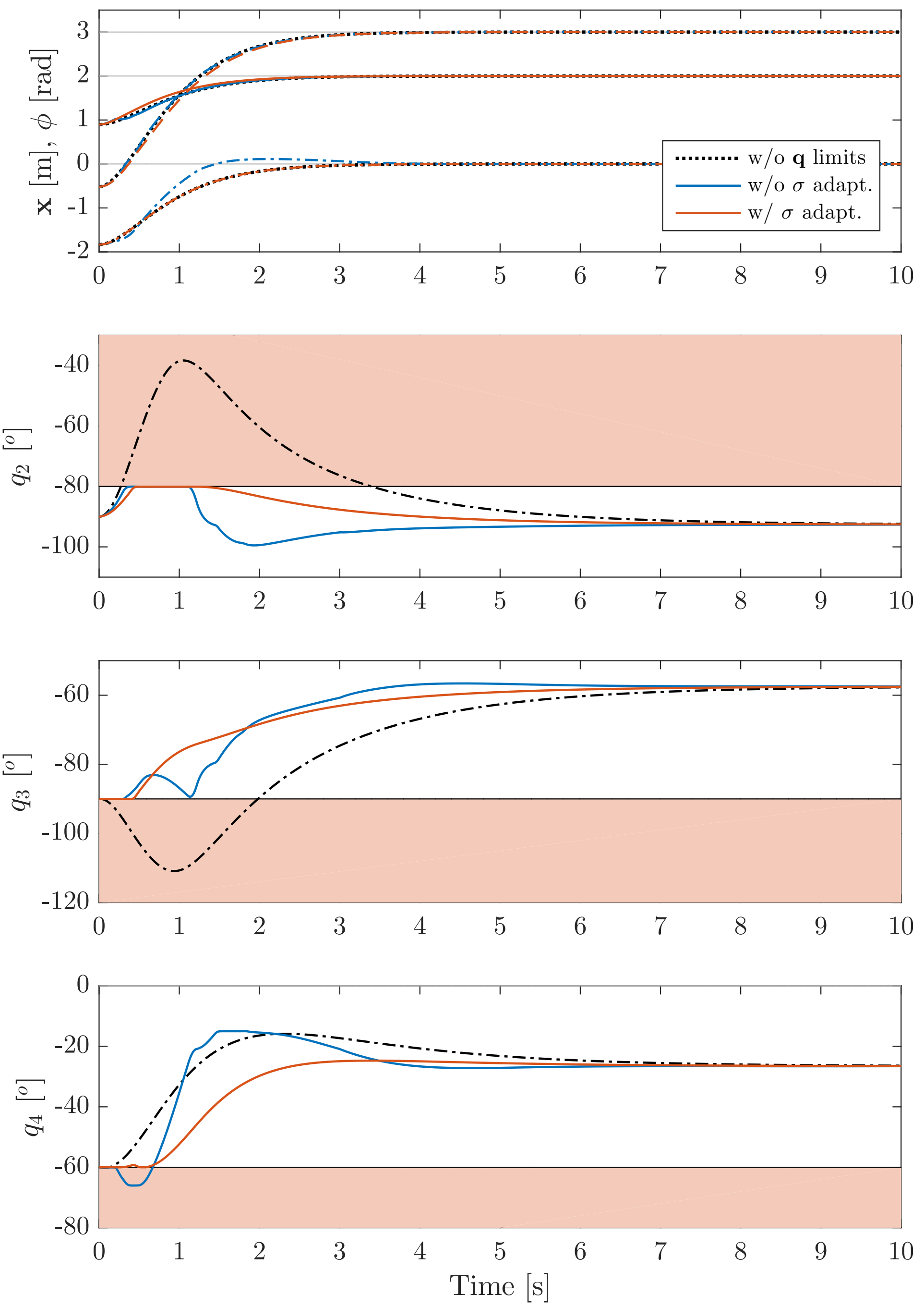


Figure 3.11: *Simulation with joint limits. From top to bottom: end effector position and orientation, second, third and fourth joint angles.*

3.7.4 Control torque delay

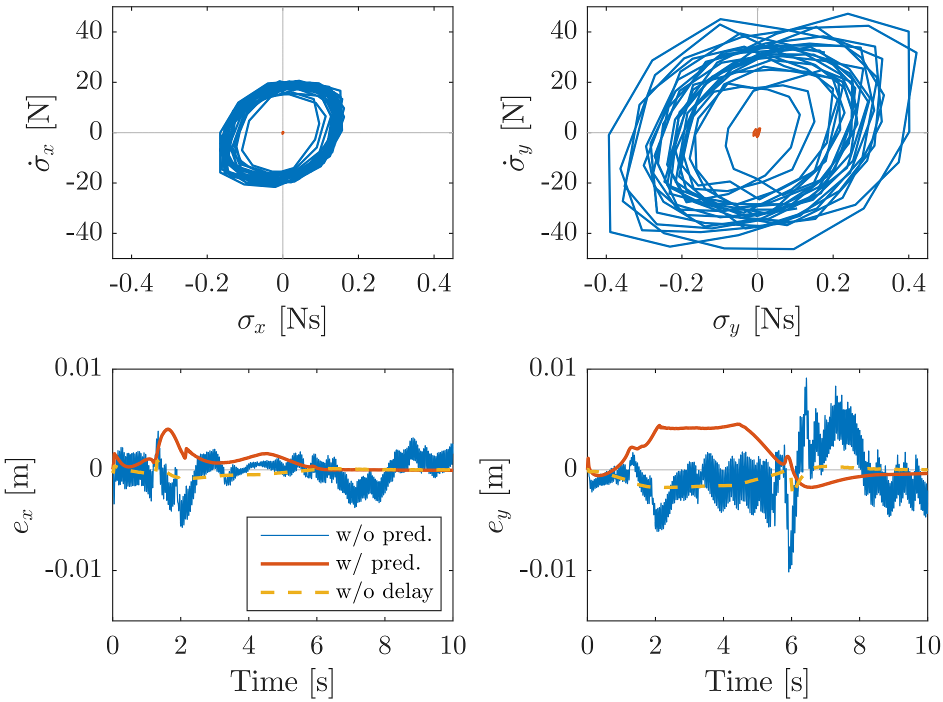


Figure 3.12: *Simulation with torque delay: tracking. Top: sliding manifold phase plane evolution. Bottom: tracking error comparison.*

In a final simulation, we tested the predictive approach of Sec 3.6.4, by considering a two-step delay $d = 2$ on the input torque (corresponding to an actuation time delay of $8ms$). The results in Fig. 3.12 illustrate how the limit cycle that arises from neglecting the delay is quite large (blue), with a $\pm 20N$ error with respect to the desired impedance, especially compared to the behavior when prediction is performed (red). For comparison, the position error without any delay active on the system is displayed in yellow. With the prediction we achieve slightly worse results, but with remarkably less oscillations of the end effector.

In Fig. 3.13, contact with the environment is robustly established even in presence of delay when the prediction is employed, although there is some performance deterioration in terms of position and orientation error.

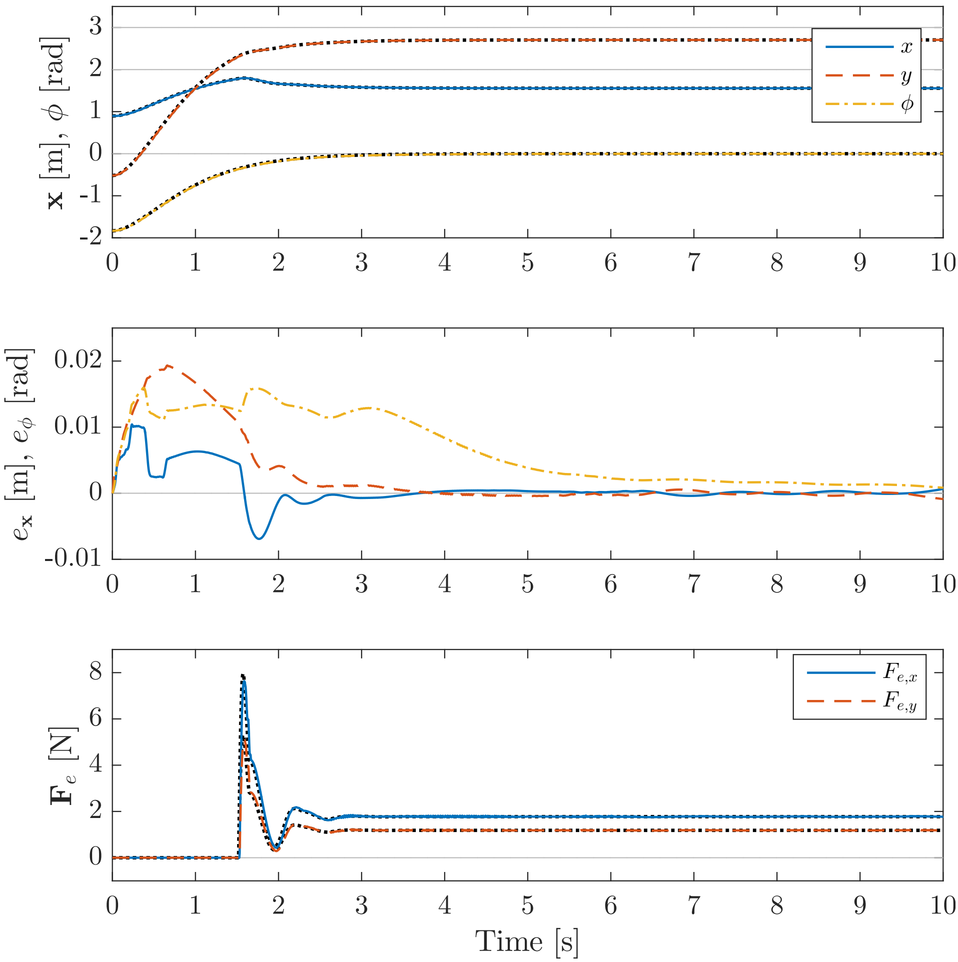


Figure 3.13: *Simulation with torque delay: regulation. Top: position and orientation. Middle: error with respect to the desired dynamics. Bottom: contact force.*

3.8 Experiments

Experiments have been performed on a platform consisting of a YuMi robot prototype, where only the left 7 d.o.f. arm was considered. In the experiments the arm was controlled to track a reference TVP profile and then impact against metal slab placed in the middle of the path (Fig. 3.14).

In order to act directly on the motor torques of our setup, the industrial

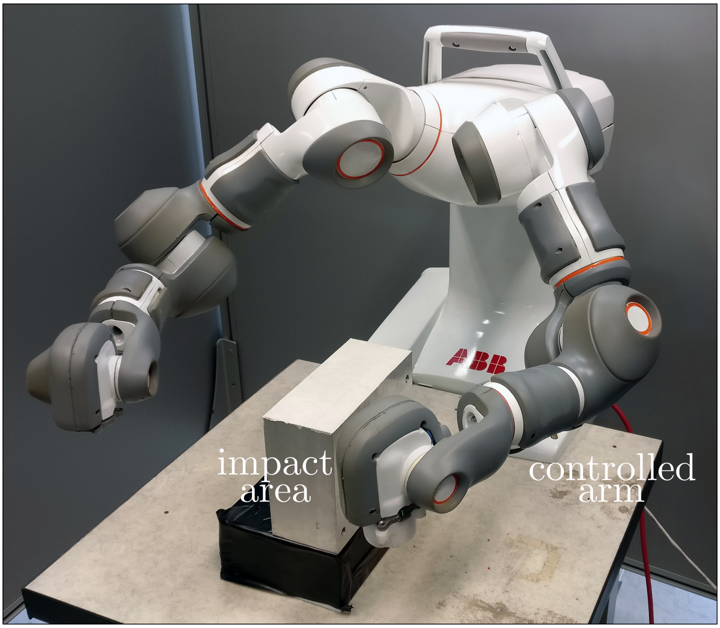


Figure 3.14: *The experimental setup used to validate the proposed approach.*

P-PI control loop was opened and we injected our control torque through the torque feed-forward channel, which unfortunately presented a built-in FIR filter with $10Hz$ cut-off frequency and a 2-step actuation delay. These dynamics were in part considered by performing a 3-step prediction of joint measurements to account for the phase loss as described in Sec. 3.6.4.

Due to the absence of force sensors, interaction forces were reconstructed by employing a momentum observer [84]. Overall, the only available measurements were actuation torques, and joint positions, with joint velocities obtained by filtered differentiation with a $30Hz$ second order LPF. The robots inertia, Coriolis, and gravitational terms were computed based on the robot CAD drawings, while friction was approximated with a Coulomb and viscous model, linearized close to zero velocity to avoid discontinuities. The controller was discretized with a sampling time of $T_s = 4ms$. For the sliding mode control gains, we selected by trial and error $\boldsymbol{k}_1 = [30\ 30\ 35\ 35\ 20\ 20\ 20]$ and $\boldsymbol{k}_2 = [35\ 35\ 40\ 40\ 20\ 20\ 20]$. The redundancy was solved by keeping a constant wrist orientation and minimizing joint velocities.

In a first experiment we employed the classical non-robust impedance control approach for comparison purposes. We selected the following parameters: $\boldsymbol{M} = 30\boldsymbol{I}_3 kg$, $\boldsymbol{D} = 100\boldsymbol{I}_3 Ns/m$, $\boldsymbol{K} = 83.3\boldsymbol{I}_3 N/m$. Fig. 3.15

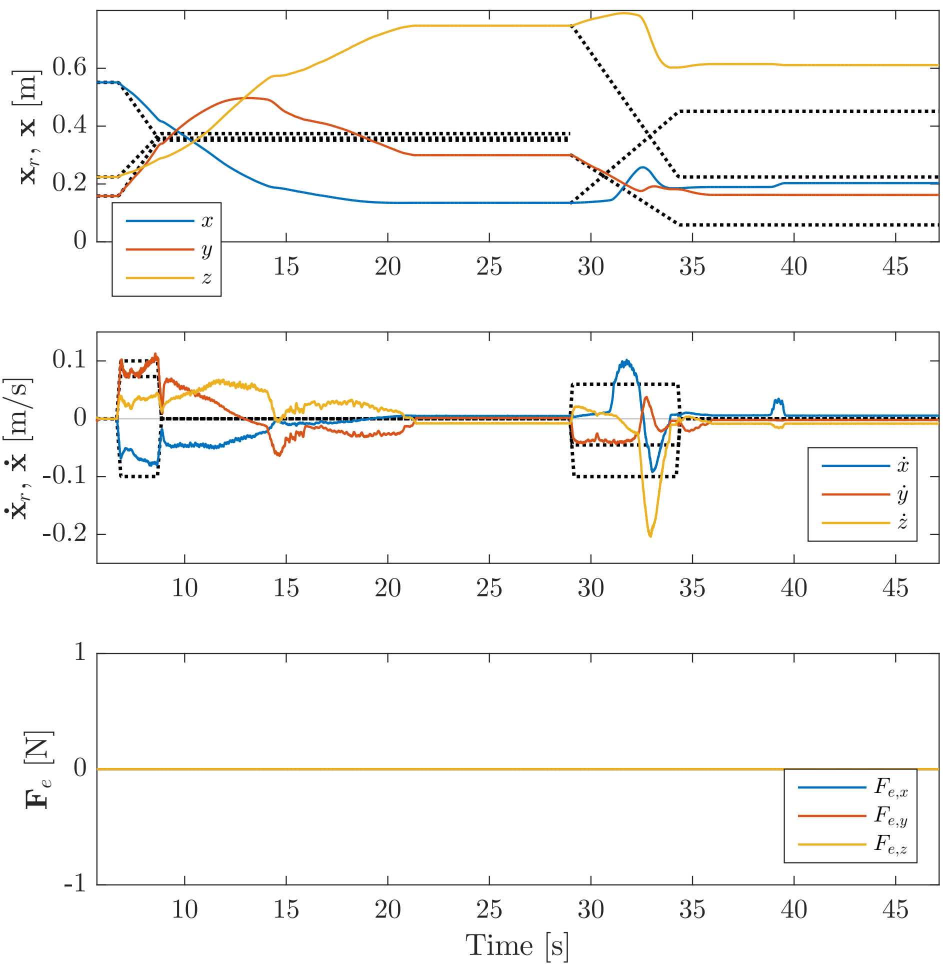


Figure 3.15: *Experiment with the standard impedance controller. Top and middle: end effector position and velocity (solid) and TVP reference (dotted). Bottom: environment force acting on the end effector.*

presents the experiment results. The performance is unacceptable, and the controller is not able to provide sufficient tracking accuracy even in absence of external forces, with the friction terms being the main cause of task failure. Indeed, if we do not use the friction model for at least partial compensation, the robot does not move at all by applying only the nominal control.

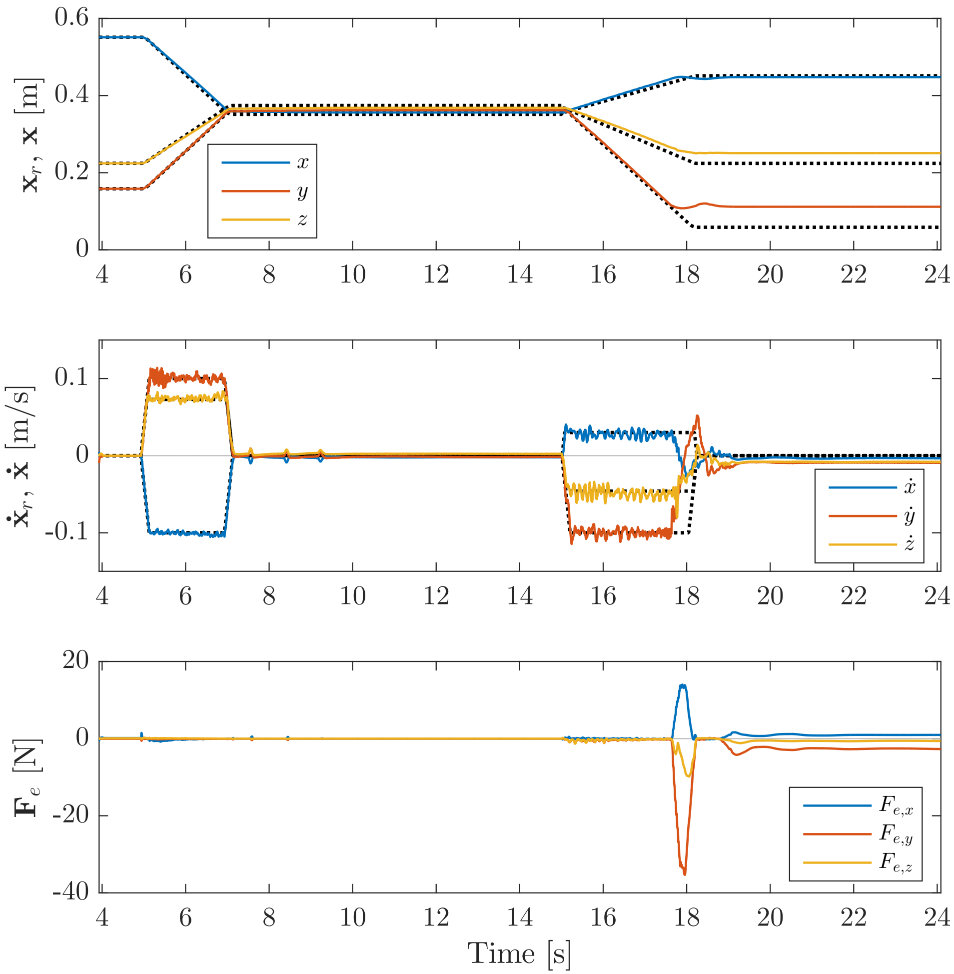


Figure 3.16: *Experiment with the proposed approach and tuning. Top and middle: end effector position and velocity (solid) and TVP reference (dotted). Bottom: environment force acting on the end effector.*

Next we applied the proposed approach and reduced the impedance parameters by a factor 10. In Fig. 3.16, we see a clear improvement in tracking the desired reference, even for such small impedance gains, which is also confirmed by the small sliding variable values (Fig. 3.17 bottom). Unfortunately, some non negligible tracking errors sometimes remain present (in the order of millimeters), due to the neglected industrial controller dy-

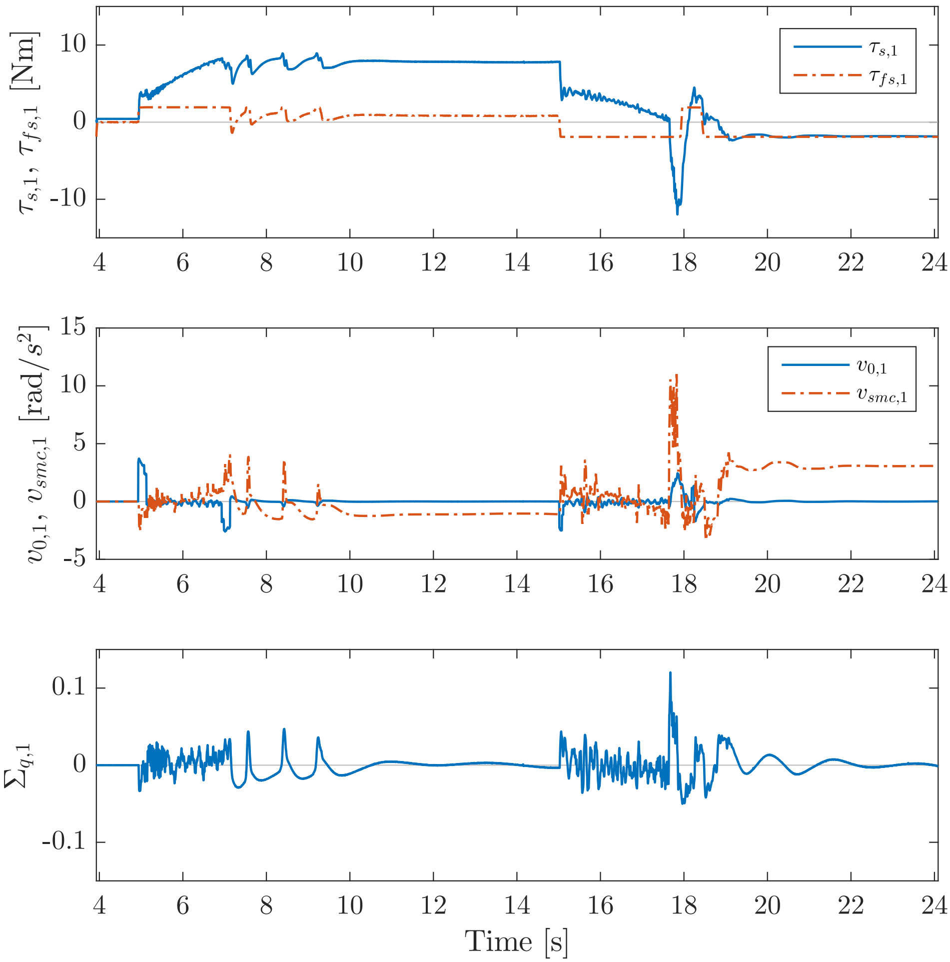


Figure 3.17: *Experiment with the proposed approach and tuning. From top to bottom: Total motor torque $\boldsymbol{\tau}_s$ and friction $\boldsymbol{\tau}_{fs}$, auxiliary control comparison, and sliding variable for the first joint of the robot.*

namics, but mostly caused by Coulomb and static friction. In the considered system a significant part of the total torque is used to overcome these terms (Fig. 3.17).

After gravity compensation, friction terms dominate the dynamic ones, as highlighted also by the large sliding mode auxiliary control $\boldsymbol{v}_{smc}$ compared to the nominal one $\boldsymbol{v}_0$. As the friction model grows more inaccurate close to zero velocity, a higher sliding mode control is required, however a complete rejection is difficult to obtain due to the high non-linearity. Although performance remains acceptable, the use of ad-hoc friction compensation techniques could prove beneficial, if the sliding mode gains cannot be increased due to torque limitations [85].

When the robot impacts against the metal surface towards the end of the trajectory, the system behaves as expected. After the initial transient, contact is stabilized as desired, therefore achieving compliance during interaction as well as tracking accuracy in free motion with the same controller.

3.9 Closing comment

In this chapter we discussed the robust control of torque-controlled robots and proposed a control architecture based on sliding mode and model predictive control to cope with system uncertainties and other unmodeled effects such as input filtering and torque actuation delays. We also tackled the problem of robust impedance control in the operational space for redundant manipulators. The next chapter will try to apply these results to a full teleoperation system and give an in-depth stability discussion and guarantees.

CHAPTER 4

Impedance shaping of teleoperators with time-delay

In this chapter we apply the robust impedance controller of Chap. 3 to a bilateral teleoperation system to provide a general control formulation for torque-controlled teleoperators. The objective is to obtain a controller that allows the exploitation of the optimization control structure for virtual fixture rendering as presented in Chap. 2 for position-controlled robots, as well as robustness in presence of contact with the real environment.

We rely on the MPSMC of Chap. 3 to ensure accurate impedance tracking both for master and slave devices and reject the uncertainties of imperfect inverse dynamics. Unlike previous approaches in sliding mode teleoperation, chattering of control torques is alleviated with the integral second order sliding mode of Sec. 3.4, without the estimation of robot accelerations. The model predictive optimization-based component of the controller enforces the desired end effector impedance dynamics of master and slave manipulators, while taking into account redundancy, control and kinematic constraints.

Llewellyn's absolute stability criterion [86] is employed to tune the impedance parameters in case of communication delays. Such stability analysis will also clarify the importance of accurate impedance tracking in teleoperation, to maximize transparency properties while retaining stability of the delayed system. The in-depth discussion of the architecture stability and transparency properties highlights some tuning aspects that have been neglected in previous works.

In the resulting "three-plus-one" channel architecture, only the slave force is fed back to the master station, while velocity, user interaction force and also a delayed version of slave contact forces are fed forward to the slave device. Most notably, the inherent robustness also allows the feed-forward of master accelerations without numerical derivations. The performance of the proposed algorithm is also compared with a time-domain passivity (TDP) approach.

The remainder of the chapter is organized as follows. In Sec. 4.1 we present previous attempts on the use of sliding mode control in teleoperation, and Llewellyn's absolute stability. Sec. 4.2 briefly recalls the robust inverse dynamics results of Chap. 3. In Sec. 4.3 the hierarchical optimization-based approach is detailed for master and slave devices. Stability and transparency properties of the teleoperation controller are discussed in Sec. 4.4 and 4.5, and compared with a passivity-based algorithm in simulation (Sec. 4.6). Results of the experimental validation are given in Sec. 4.7.

4.1 Background

Many approaches have been proposed to overcome stability limitations while maintaining an acceptable degree of transparency in bilateral teleoperation [16, 87]. While the most established techniques resort to damping injection or wave variables, recent approaches favor time-domain passivity, in order to reduce performance only upon loss of passivity. In [88] passivity observers and controllers monitor and dissipate part of the system energy, when the communication channel displays an active behavior that might destabilize contact. This fundamentally allows damping injection only in critical situations, preserving transparency properties otherwise. A two-layer approach is employed in [89], where virtual reservoirs store surplus energy that would be otherwise dissipated and drain it when the channel becomes active, to preserve overall passivity.

Absolute stability criteria are often employed to tune the control gains so that the system remains stable whatever the external environment and operator dynamics are, as long as they satisfy a reasonable assumption of passivity. In [90], the authors proposed an in-depth analysis of two and four-channel teleoperation control structures, based on Llewellyn's criterion and Lawrence formalism. Although four-channel controllers provide substantially better transparency results, two-channel architectures were shown to be remarkably more stable and easier to tune, especially position-force schemes, where only the environmental force is fed back to the user, removing delay induced forces and slave dynamics reflection. The same authors analyzed the tuning of three-channel architectures [91], giving insights on the parameters choice depending on the application and the local feedback controller. Indeed, these provide improved transparency via operator force feed-forward and still manageable tuning, especially if some environment knowledge is available [92]. Nonetheless, a force/torque sensor becomes necessary also on the master device.

Sliding mode control theory has been applied also in the context of teleoperation, starting with the work of Buttolo et al. [93]. The authors discussed the definition of a classical sliding surface with a linear combination of position and velocity tracking errors. They highlighted the improvements with respect to a classical PD controller of a position-position architecture, but delays were not considered. Cho et al. tackled the communication delay problem in [76], where they defined a sliding surface based on the integral of the desired impedance relation, in order to obtain both accurate tracking of the master and compliance during contact. A similar approach has been proposed in [94], but with a higher order sliding mode to avoid chattering. The method unfortunately requires the use of an observer to estimate the acceleration for the sliding surface computation. While previous attempts considered simple 1 d.o.f. devices, an operational space sliding mode was applied to a multi-dof teleoperator in [95], without considering redundancies. In all these works, some critical limitations in the choice of impedance parameters for both master and slave were neglected. A mixed sliding mode/fuzzy logic controller was proposed in [96], to reduce the chattering effect while still employing a first order algorithm, while the four-channel scheme of Hace et al. [97] was completely chattering-free but did not converge to the desired sliding manifold in finite time.

4.2 Robust inverse dynamics

Here we briefly recall some of the equations presented in Chap. 3 and apply the robust control framework to a teleoperation system. For both master and slave devices we consider generic n d.o.f. robots, possibly redundant, characterized by model (3.1)

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{\tau} - \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{F} \tag{4.1}$$

with $\boldsymbol{F} \in \mathbb{R}^m$ a vector of external forces, which can be environment contact forces for the slave ($\boldsymbol{F}_e$) or forces applied by the user for the master ($\boldsymbol{F}_h$). Substituting the nominal inverse dynamics control torque (3.2) in (4.1) we obtain the partially feedback-linearized dynamics (3.3)

$$\ddot{\boldsymbol{q}} = \boldsymbol{B}^{-1}\hat{\boldsymbol{B}}\boldsymbol{v} + \boldsymbol{B}^{-1}\tilde{\boldsymbol{\eta}} \tag{4.2}$$

The system is fully decoupled only with perfect knowledge of the dynamics. Therefore, in a teleoperation system, the application of a standard impedance controller does not guarantee neither a zero tracking error of the slave device with respect to the master, nor the enforcement of the desired dynamics, possibly invalidating absolute stability properties, and resulting in instability due to communication delay.

In the following we will employ the integral second order sliding mode formulation proposed in Sec. 3.4.2 and 3.4.3 to obtain a robust feedback linearization, with

$$\Sigma_q(t) = \dot{\boldsymbol{q}}(t) - \dot{\boldsymbol{q}}_0(t), \quad \dot{\boldsymbol{q}}_0(t) = \int_{t_0}^{t} \boldsymbol{v}_0 d\tau + \dot{\boldsymbol{q}}(t_0) \tag{4.3}$$

$$\boldsymbol{v} = \boldsymbol{v}_0 + \boldsymbol{v}_{smc}(\Sigma_q) = \boldsymbol{v}_0 - k_1\sqrt{|\Sigma_q|} \odot \mathrm{sgn}(\Sigma_q) - k_2 \int_{t_0}^{t} \mathrm{sgn}(\Sigma_q) d\tau \tag{4.4}$$

By doing so, what remains is to select the desired impedances for master and slave to obtain $\boldsymbol{v}_0$.

4.3 Master/slave impedance control

After applying control (4.4), we can simply consider the nominal system (3.32) for both master and slave robot. The impedance controllers are detailed here.

4.3.1 Master device

For the user operated master device we consider an impedance model where the slave interaction force is used for haptic feedback. This allows to avoid the reflection of slave dynamics onto the master, as well as high robustness in free motion. Then:

$$\boldsymbol{M}_m\ddot{\boldsymbol{x}}_m + \boldsymbol{D}_m\dot{\boldsymbol{x}}_m + \boldsymbol{K}_m\boldsymbol{x}_m = \boldsymbol{F}_h - k_f\boldsymbol{F}_e^d \tag{4.5}$$

where $\boldsymbol{M}_m$, $\boldsymbol{D}_m$ and $\boldsymbol{K}_m$ are the desired inertia, damping and stiffness respectively, with subscript m indicating master quantities. $\boldsymbol{x}_m$ is the end effector position, while $\boldsymbol{F}_h$ is the force applied by the user on the master end effector. $\boldsymbol{F}_e^d(t) = \boldsymbol{F}_e(t - d_2)$ is the force exerted on the slave, with d_2 the slave-to-master delay, and k_f a scaling factor.

Considering the relation between master end effector acceleration and joint velocity and acceleration via the robot Jacobian $\ddot{\boldsymbol{x}}_m = \boldsymbol{J}_m\ddot{\boldsymbol{q}}_m + \dot{\boldsymbol{J}}_m\dot{\boldsymbol{q}}_m$, using (3.32), and then substituting in (4.5), we obtain the impedance equation $\boldsymbol{I}_m = 0$

$$\boldsymbol{I}_m = \boldsymbol{M}_m\boldsymbol{J}_m\boldsymbol{v}_{0,m} - \boldsymbol{b}_m = 0 \tag{4.6}$$

$$\boldsymbol{b}_m = -\boldsymbol{M}_m\dot{\boldsymbol{J}}_m\dot{\boldsymbol{q}}_m - \boldsymbol{D}_m\dot{\boldsymbol{x}}_m - \boldsymbol{K}_m\boldsymbol{x}_m + \boldsymbol{F}_h - k_f\boldsymbol{F}_e^d$$

Therefore, the least squares solution of (4.6) gives the nominal auxiliary control enforcing the desired impedance

$$\boldsymbol{v}_{0,m} = \boldsymbol{J}_m^\dagger\boldsymbol{M}_m^{-1}\boldsymbol{b}_m \tag{4.7}$$

With reference to Lawrence formalism, the transfer functions of the slave signals to master torques are the following

$$\boldsymbol{\tau}_{m,\boldsymbol{F}_e} = C_2(s)\boldsymbol{F}_e = -k_f\hat{\boldsymbol{B}}_m\boldsymbol{J}_m^\dagger\boldsymbol{M}_m^{-1}\boldsymbol{F}_e e^{-sd_2} \tag{4.8}$$

$$\boldsymbol{\tau}_{m,\dot{\boldsymbol{x}}_s} = C_4(s)\dot{\boldsymbol{x}}_s = 0 \tag{4.9}$$

4.3.2 Slave device

In the design of the slave impedance we must ensure the correct tracking of the reference master trajectory and compliance in case of contact forces. Hence, analogously to (2.7), we select

$$\boldsymbol{M}_s\ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{D}_s\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_s\tilde{\boldsymbol{x}} = -\boldsymbol{F}_e \tag{4.10}$$

where subscript s indicates slave quantities, and $\tilde{\boldsymbol{x}} = \boldsymbol{x}_s - k_p\boldsymbol{x}_m^d$, with k_p a scaling factor accounting for the possibly different robot workspaces. $\boldsymbol{x}_m^d(t) = \boldsymbol{x}_m(t - d_1)$ is the delayed master position with d_1 the master-to-slave delay.

Similar to the master case, we have

$$\boldsymbol{I}_s = \boldsymbol{M}_s\boldsymbol{J}_s\boldsymbol{v}_{0,s} - \boldsymbol{b}_s = 0 \tag{4.11}$$

$$\boldsymbol{b}_s = -\boldsymbol{M}_s\dot{\boldsymbol{J}}_s\dot{\boldsymbol{q}}_s + \boldsymbol{M}_s k_p\ddot{\boldsymbol{x}}_m^d - \boldsymbol{D}_s\dot{\tilde{\boldsymbol{x}}} - \boldsymbol{K}_s\tilde{\boldsymbol{x}} - \boldsymbol{F}_e$$

While selection of the slave impedance as in (4.5) resembles a two-channel position-force architecture, the presence of the master acceleration in the previous equation is critical. To solve this problem we can simply compute the acceleration from (4.5)

$$\ddot{\boldsymbol{x}}_m^d = -\boldsymbol{M}_m^{-1}\left(\boldsymbol{D}_m\dot{\boldsymbol{x}}_m^d + \boldsymbol{K}_m\boldsymbol{x}_m^d - \boldsymbol{F}_h^d + k_f\boldsymbol{F}_e^{dd}\right) \tag{4.12}$$

where $\boldsymbol{F}_e^{dd}(t) = \boldsymbol{F}_e(t - d_1 - d_2)$ is the twice delayed slave contact force forwarded by the master and necessary for a correct computation of the acceleration. Substituting the previous equation in (4.11), the nominal auxiliary control is obtained as in (4.7) via least squares. Such control depends on the master position/velocity and operator force, plus the additional feedforward of the delayed slave external force. Overall, we obtain a "three-plus-one" channel controller. This gives the following remaining transfer functions of the Lawrence scheme

$$\boldsymbol{\tau}_{s,\boldsymbol{F}_h} = C_3(s)\boldsymbol{F}_h = k_p\hat{\boldsymbol{B}}_s\boldsymbol{J}_s^\dagger\boldsymbol{M}_m^{-1}\boldsymbol{F}_h e^{-sd_1} \tag{4.13}$$

$$\begin{aligned}\boldsymbol{\tau}_{s,\dot{\boldsymbol{x}}_m} = C_1(s)\dot{\boldsymbol{x}}_m = k_p\hat{\boldsymbol{B}}_s\boldsymbol{J}_s^\dagger\Big(&\boldsymbol{M}_s^{-1}\boldsymbol{D}_s + \\ &-\boldsymbol{M}_m^{-1}\boldsymbol{D}_m + \frac{\boldsymbol{M}_s^{-1}\boldsymbol{K}_s - \boldsymbol{M}_m^{-1}\boldsymbol{K}_m}{s}\Big)\dot{\boldsymbol{x}}_m e^{-sd_1}\end{aligned} \tag{4.14}$$

$$\boldsymbol{\tau}_{s,\boldsymbol{F}_e^d} = C_+(s)\boldsymbol{F}_e^d = -k_pk_f\hat{\boldsymbol{B}}_s\boldsymbol{J}_s^\dagger\boldsymbol{M}_m^{-1}\boldsymbol{F}_e^d e^{-sd_1} \tag{4.15}$$

Note that the computation of the master acceleration in (4.12) is valid only because the sliding mode cancels out the disturbance terms remaining from the inverse dynamics, and (3.32) is enforced. If a robust feedback linearization is not employed at master side, the accuracy of (4.12) depends on the magnitude of the uncertainties.

4.3.3 Teleoperation optimal controller

To take into account redundancies and control constraints, we employ the optimization-based approach first seen in Sec. 2.3 and detailed in Sec. 3.6 in its model predictive form.

Since the objective is to ensure the end effector impedances (4.5), (4.10), in a first optimization stage we minimize the impedance errors (4.6), (4.11) subject to the desired constraints (e.g. control, kinematic, and hard virtual fixtures), noticing that they are the manifold derivative used in Sec. 3.6

$$\boldsymbol{v}_{0,*}^0 = \arg\min_{\boldsymbol{v}_{0,*}} \|\boldsymbol{I}_*\|_{\boldsymbol{Q}_*}^2 \tag{4.16a}$$

$$s.t. \quad \underline{\boldsymbol{\tau}}_* \le \boldsymbol{\tau}_*(\boldsymbol{v}_{0,*}, \boldsymbol{v}_{smc,*}) \le \overline{\boldsymbol{\tau}}_* \tag{4.16b}$$

$$\underline{\boldsymbol{q}}_* \le \boldsymbol{q}_*(\boldsymbol{v}_{0,*}) \le \overline{\boldsymbol{q}}_* \tag{4.16c}$$

$$\boldsymbol{A}\boldsymbol{v}_{0,*} \le \boldsymbol{b} \tag{4.16d}$$

Given p other tasks and soft virtual fixtures specifications $\boldsymbol{A}^i_* \boldsymbol{v}_{0,*} \leq \boldsymbol{b}^i_*$ with decreasing priority as in Sec. 2.3, a cascade of optimizations guarantees virtual fixtures satisfaction and solves the robot redundancy (3.66).

$$\{\boldsymbol{v}^i_{0,*}, \boldsymbol{w}^{i*}_*\} = \underset{\boldsymbol{v}_{0,*}, \boldsymbol{w}^i_*}{\arg\min} \; \|\boldsymbol{w}^i_*\|^2_{\boldsymbol{Q}^i_*} \tag{4.17a}$$

$$s.t. \quad \textit{constraints up to task } i-1 \tag{4.17b}$$

$$\boldsymbol{w}^{i-1}_* = \boldsymbol{w}^{i-1*}_* \tag{4.17c}$$

$$\boldsymbol{A}^i_* \boldsymbol{v}_{0,*} \leq \boldsymbol{b}^i_* + \boldsymbol{w}^i_* \tag{4.17d}$$

where (4.17c) takes into account the higher priority layers, maintaining their optimality. The solution to the lowest priority optimization $\boldsymbol{v}_{0,*} = \boldsymbol{v}^p_{0,*}$ is the final nominal auxiliary control that enforces the desired impedance and solves the robot redundancy subject to the specified constraints, and possibly implements desired virtual fixtures.

4.4 Teleoperation stability

Impedance parameters tuning for the proposed teleoperation framework is discussed in the following, based on stability and transparency requirements.

Given the local controllers of Sec. 4.2 and 4.3, we can assume that, from the user and the environment perspective, the overall system can be described by the impedance equations (4.5), (4.10). For simplicity we consider diagonal parameter matrices and study the scalar case without loss of generality. We may highlight the relation between *efforts* (forces) and *flows* (velocities) as follows

$$Z_m \dot{x}_m = F_h - k_f e^{-sd_2} F_e \tag{4.18}$$

$$Z_s \dot{x}_s - k_p e^{-sd_1} Z_s \dot{x}_m = -F_e \tag{4.19}$$

where $Z_m = M_m s + D_m + K_m/s$ and $Z_s = M_s s + D_s + K_s/s$ are the master and slave impedances, respectively, with s the Laplace variable. The system can be represented as a *two-port* network element, described by

$$\begin{bmatrix} F_h \\ -\dot{x}_s \end{bmatrix} = \boldsymbol{H} \begin{bmatrix} \dot{x}_m \\ F_e \end{bmatrix} \tag{4.20}$$

with $\boldsymbol{H}$ the hybrid matrix

$$\boldsymbol{H} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} = \begin{bmatrix} Z_m & k_f e^{-sd_2} \\ -k_p e^{-sd_1} & Z_s^{-1} \end{bmatrix} \tag{4.21}$$

We apply Llewellyn's absolute stability criterion to prove the system stability independently of operator and environment, as long as they exhibit passive behavior.

System (4.21) is said to be absolutely stable if and only if

- h_{11} and h_{22} have no poles with positive real part
- each pole of h_{11} and h_{22} on the imaginary axis is simple and with positive real residual
- at all frequencies: $Re(h_{11}) \geq 0$ and
  $2Re(h_{11})Re(h_{22}) - Re(h_{12}h_{21}) - |Re(h_{12}h_{21})| \geq 0$

where h_{**} indicates the element of matrix $\boldsymbol{H}$.

The first two conditions are satisfied by choosing impedances with positive parameters, while $Re(h_{11}) = D_m \geq 0$ is also easily fulfilled. The last condition implies the following inequality

$$\Lambda(\omega) = \frac{2D_m D_s \omega^2}{D_s^2\omega^2 + (M_s\omega^2 - K_s)^2} + k_p k_f(\cos(d_{rt}\omega) - 1) \geq 0, \quad \forall\omega \tag{4.22}$$

with $d_{rt} = d_1 + d_2$ the round-trip delay. In absence of delay it is straightforward to see that the second term disappears and (4.22) is always satisfied for positive dampings. On the other hand, in presence of delay, there exists a frequency where Λ crosses zero and becomes negative. Indeed, by taking $\lim\limits_{\omega\to\infty} \Lambda(\omega)$, the first term approaches zero, while the second one is not defined and oscillates between 0 and $-2k_pk_f$. This is particularly critical if contact with high stiffness environments is made, since they may excite those frequencies where Λ is negative, and stability would not be guaranteed anymore. Moreover, since $\lim\limits_{\omega\to 0} \Lambda(\omega) = 0$, depending on the function convexity close to steady state, we may have $\Lambda < 0$ at a low frequency range. Therefore, it is necessary to define tuning guidelines to avoid these two phenomena.

4.4.1 Low frequency

To keep $\Lambda > 0$ near steady state, since its first derivative in the origin is always zero, we have to ensure that the second derivative of Λ is sufficiently large and positive in $\omega = 0$. That is

$$\left.\frac{\partial^2\Lambda(\omega)}{\partial\omega^2}\right|_{\omega=0} = \frac{4D_m D_s}{K_s^2} - k_p k_f d_{rt}^2 \gg 0 \tag{4.23}$$

From the previous equation we obtain an upper bound on K_s by taking the worst case round-trip delay $\bar{d}_{rt}$ and assuming that the other parameters have already been chosen

$$K_s \ll 2\sqrt{D_m D_s / (k_p k_f \bar{d}_{rt}^2)} = K_c \tag{4.24}$$

Intuitively, higher delays or an increase of the feed-forward coupling terms require a smaller stiffness of the slave end effector. Fig. 4.1 shows the effect of this constraint at low frequencies.

4.4.2 High frequency

In order to obtain a relation between the zero-crossing frequency and the parameters, we take the lower envelope of Λ which removes the delay dependency

$$\underline{\Lambda}(\omega) = \frac{2 D_m D_s \omega^2}{D_s^2 \omega^2 + (M_s \omega^2 - K_s)^2} - 2 k_p k_f \tag{4.25}$$

Solving the biquadratic equation $\underline{\Lambda}(\omega) = 0$ and taking the solution with largest positive value (the other positive one is relative to the low frequency case), we obtain the relationship between the crossing frequency and the tuning parameters

$$\omega_0 = f(D_m, M_s, D_s, K_s, k_p, k_f) \tag{4.26}$$

A sensitivity analysis of Λ and ω_0 is proposed in Fig. 4.2). A larger D_m increases $\underline{\Lambda}$ and the crossing frequency improving stability, however a too high master damping should be avoided as it interferes with the operator maneuvers. Since there is no dependence on M_m and K_m, they can be chosen as desired. As for M_s, a lower slave mass increases ω_0, but a lower limit exists, due to the intrinsic robot elasticity [55]. A small D_s increases the curve peak, while a higher value has a flattening effect, however ω_0 is not monotonic with the damping and a trade-off value should be selected. Finally, K_s affects the curve mostly at low frequency, with little variation of ω_0. Small k_p and k_f improve stability, however their choice is usually dictated by task and robot workspaces. Note also that scaling the slave parameters by the same factor has the inverse effect of D_m.

Given these considerations, we should acknowledge that no tuning exists, such that the system is guaranteed to be absolutely stable for all environment and user impedances. Indeed, it is always possible to find a stiff enough environment and a user with a sufficiently compliant grip so that frequencies above ω_0 are excited and instability ensues (Fig. 4.4), although

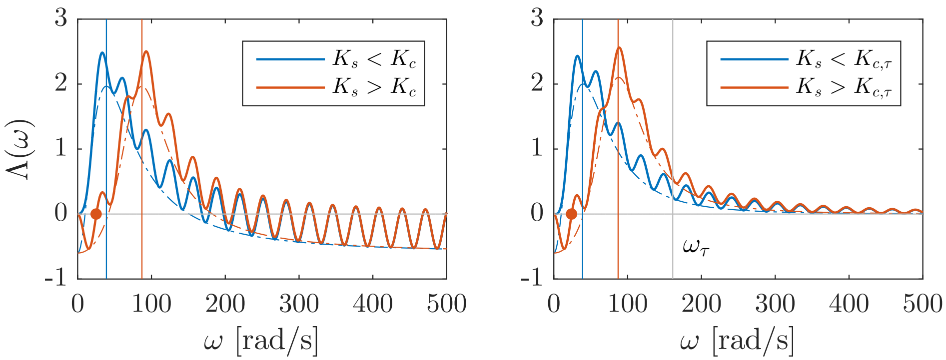


Figure 4.1: *Llewellyn's absolute stability condition.* $\Lambda(\omega)$ *with optimal tuning* (4.31) *(*$B_m = 50$, $M_s = .5$, $B_s = 39$, $K_s = 759$*) compared to the case with* K_s *violating* (4.24)*, on the right the same comparison with filtered force feecback. The impedance natural frequency is highlighted.*

the crossing frequency can be arbitrarily increased to improve robustness. To solve this problem we propose a modification of the force feedback.

4.4.3 Force feedback filtering

By filtering the force feedback with a second order filter, the h_{12} entry of the hybrid matrix (4.21) is modified as follows

$$h_{12} = \frac{k_f}{(1+s\tau)^2} e^{-sd_2} \tag{4.27}$$

where τ is the filter time constant. Llewellyn condition (4.22) becomes

$$\begin{aligned}\Lambda(\omega) = &\frac{2D_m D_s \omega^2}{D_s^2\omega^2 + (M_s\omega^2 - K_s)^2} + \frac{k_p k_f}{1+\omega^2\tau^2}\cdot \\ &\cdot \left(\frac{(1-\omega^2\tau^2)\cos(d_{rt}\omega) - 2\omega\tau\sin(d_{rt}\omega)}{1+\omega^2\tau^2} - 1\right) \geq 0, \quad \forall\omega\end{aligned} \tag{4.28}$$

At low frequency we obtain a similar constraint on K_s

$$K_s \ll 2\sqrt{D_m D_s/(k_p k_f(2\tau + \bar{d}_{rt})^2)} = K_{c,\tau} \tag{4.29}$$

At high frequency, both the first term and the trigonometric one become infinitesimals of the same order, therefore it is sufficient to choose τ so that they decrease at least at the same rate to avoid any zero crossing, obtaining the following condition

$$\tau \geq M_s\sqrt{k_p k_f/(D_m D_s)} = \tau_c \tag{4.30}$$

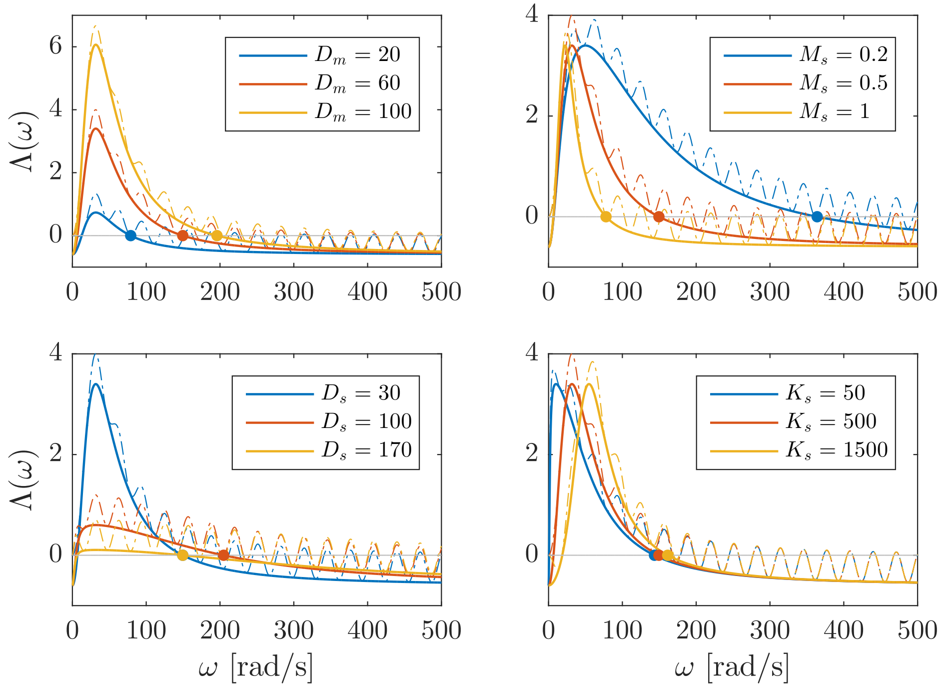


Figure 4.2: *Llewellyn's absolute stability condition. Sensitivity analysis of* (4.22) *for varying parameters with the respective high frequency cross-over.*

By filtering, we ensure the absolute stability requirement at all frequencies, thus guaranteeing a stable behavior for every passive environment and user. Theoretically, the remaining parameters can be chosen as desired, even though with a transparency trade-off.

Finally, in free motion $F_e = 0$, and equivalently $k_f = 0$, in $\Lambda(\omega)$ only the first term of (4.22) appears, which is positive at all frequencies. Hence, the system in free motion is always absolutely stable for positive parameters.

Although all tunings are acceptable with force feedback filtering as long as (4.29), (4.30) are satisfied, to confer robustness in presence of neglected dynamics, the parameters should be obtained by solving the following nonlinear optimization problem, where we try to maximize the crossing frequency of the unfiltered scheme

$$\max_{D_m, M_s, D_s, K_s} \quad \omega_0 \tag{4.31a}$$

$$s.t. \quad 0 < D_m \leq \overline{D}_m \tag{4.31b}$$

$$0 < \underline{M}_s \leq M_s \tag{4.31c}$$

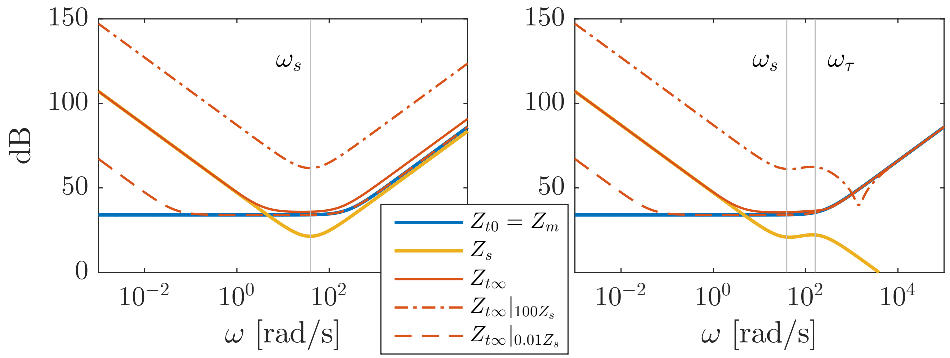


Figure 4.3: *Proposed approach transparency characteristic. Transmitted impedance transfer function for varying slave impedance scaling factors in free motion and with rigid environment, without (left) and with (right) filtering.*

$$0 < D_s \tag{4.31d}$$

$$0 < K_s \ll K_{c,\tau} \tag{4.31e}$$

$$\tau = \tau_c \tag{4.31f}$$

$$D_s = 2\sqrt{M_s K_s} \tag{4.31g}$$

with $\overline{D}_m$ and $\underline{M}_s$ the maximum master damping and minimum slave inertia allowed. The last equality guarantees well-damped poles, while τ_c is the smallest time constant that avoids any zero crossing and filters as little as possible the force feedback. Fig. 4.1 presents the tuning obtained by solving (4.31), with and without force feedback filtering and the effect of the stiffness constraint $K_s \ll K_{c,\tau}$.

4.5 Teleoperation transparency

As far as transparency is concerned, to obtain perfect force reflection and master tracking, the following expression for the hybrid matrix is necessary

$$\boldsymbol{H} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{4.32}$$

Unfortunately, such result is not attainable due to stability requirements. Indeed, in presence of delay the Llewellyn function would be always below zero.

To evaluate teleoperation transparency the transmitted impedance Z_t is usually considered, i.e. the impedance displayed to the user during contact. Given $Z_e = F_e/\dot{x}_s$ the environment impedance, with filtered force feedback the transmitted one becomes

$$Z_t = \frac{F_h}{\dot{x}_m} = Z_m + \frac{k_p k_f Z_s Z_e e^{-s(d_1+d_2)}}{(Z_s + Z_e)(1+s\tau)^2} \tag{4.33}$$

It is interesting to investigate the limit conditions, in free motion and with infinitely rigid environment

$$Z_{t0} = \lim_{Z_e \to 0} Z_t = Z_m \tag{4.34}$$

$$Z_{t\infty} = \lim_{Z_e \to \infty} Z_t = Z_m + \frac{k_p k_f Z_s}{(1+s\tau)^2} e^{-s(d_1+d_2)} \tag{4.35}$$

Clearly, for maximum transparency Z_t should be as close as possible to Z_e, therefore we should aim for a small master impedance, a large slave impedance across the frequencies of interest, and high frequency filter poles (small τ). The force feedback fidelity K_t gives the same result, as we would like $K_t \approx k_f$

$$K_t = \frac{F_h}{F_e} = k_f e^{-sd_2} + \frac{Z_m(Z_s + Z_e)e^{s(d_1+d_2)}}{k_p Z_s Z_e} \tag{4.36}$$

$$K_{t\infty} = \lim_{Z_e \to \infty} K_t = k_f e^{-sd_2} + \frac{Z_m e^{s(d_1+d_2)}}{k_p Z_s} \tag{4.37}$$

Fig. 4.3 shows the transmitted impedance behavior for varying scaling of Z_s. In free motion the displayed impedance is always the master's one, while in contact it is approximately given by the upper envelope of Z_m, Z_s. There is no anti-resonant peak if the slave impedance is critically damped, unless the master zero is at a sufficiently low frequency and their plots intersect. Clearly, scaling up slave parameters generally provides better transparency across all frequencies but at the expense of stability. However, with force feedback filtering, performance after the filter poles is only determined by the master impedance.

4.6 Simulations

Simulations were performed in MATLAB SIMULINK to compare the behavior of our system with and without force feedback filtering, and against a time-domain passivity (TDP) approach [88]. We modeled the user as a position regulator [98], while we adopted a spring-damper model with stiffness K_e and damping D_e for the environment, placing it at $x_e = 0.1m$. The following parameters were selected based on (4.31): $M_m = 0.2Kg$, $D_m =$

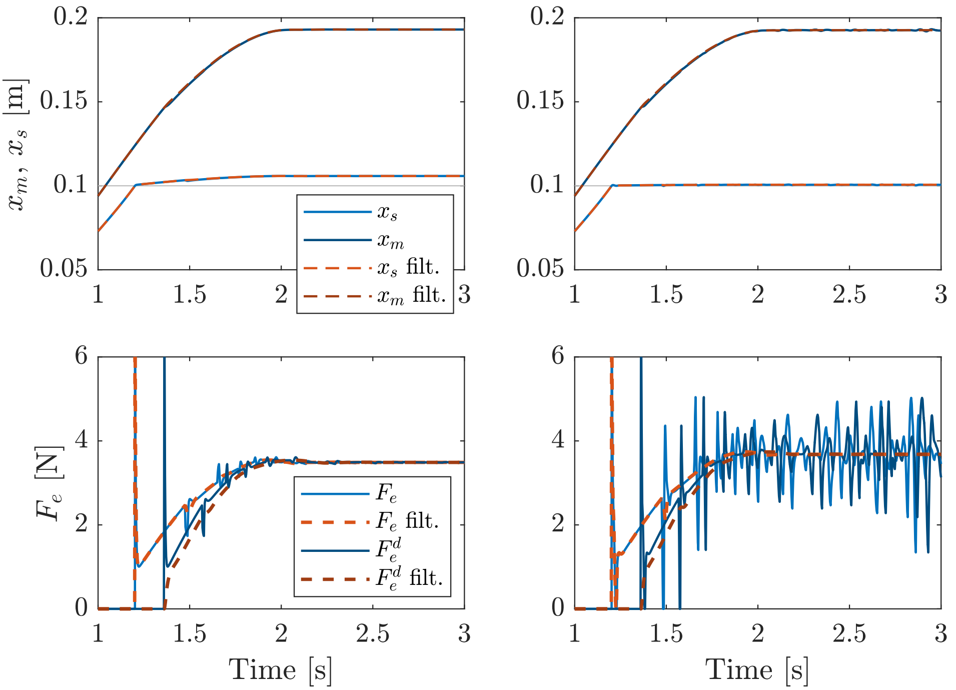


Figure 4.4: *Simulation with and without force feedback filter. Left: low environment impedance. Right: high impedance. F_e^d is the actual force feedback used by the master.*

$20Ns/m$, $K_m = 0N/m$, $M_s = 0.2Kg$, $D_s = 6Ns/m$, $K_s = 40N/m$, $k_p = k_f = 1$, $\tau = 0.0189s$. A variable communication delay between $20ms$ and $200ms$ was added, for a maximum $400ms$ round-trip delay.

We first compared the same tuning with and without force feedback filtering with two different environments, a compliant one with $K_e = 600N/m$, $D_e = 50Ns/m$ and a slightly stiffer one with $K_e = 6000N/m$, $D_e = 50Ns/m$. The results are shown in Fig. 4.4. With compliant environment both schemes are stable, with the unfiltered one able to provide better transparency, displaying also the initial contact force spike. However by increasing stiffness the unfiltered one slowly turns unstable, amplifying the force spikes and eventually losing contact, while the filtered version maintains the same robust behavior.

In the second simulation we considered the TDP approach of [88]. In TDP the energy flow entering the *two-port* on master and slave side is mon-

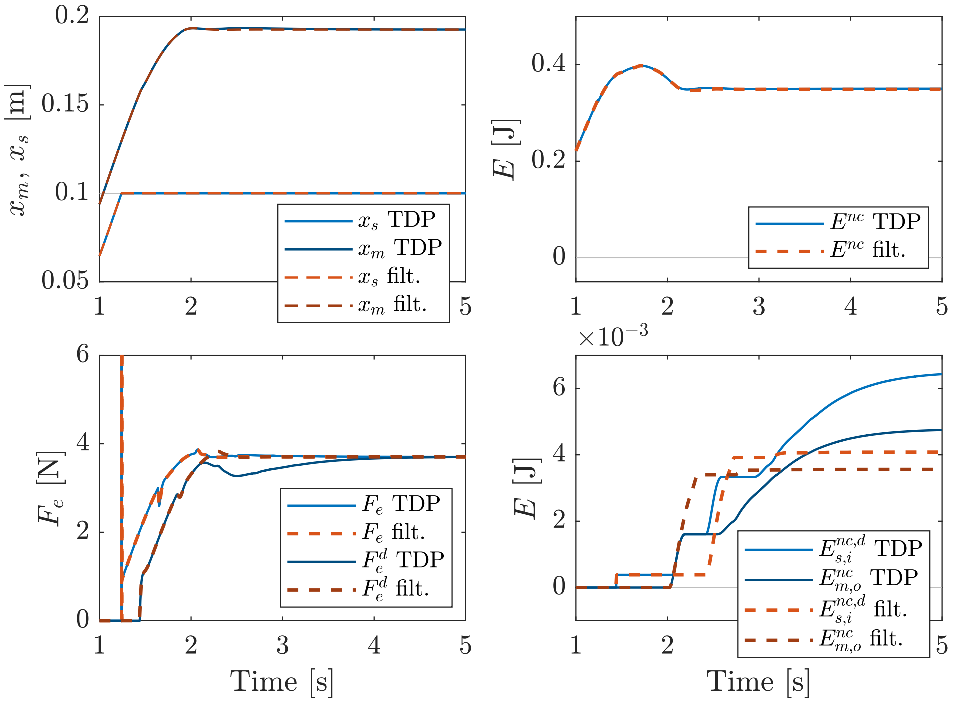


Figure 4.5: *Comparison between the proposed and the TDP approach. Left: master and slave positions and forces. Right: total system energy (top) and energy components at the master port, for both approaches.*

itored. The following must be ensured to guarantee a passive stable system

$$E(t) = E_m(t) + E_s(t) = \int_{t_0}^{t} F_h \dot{x}_m - F_e \dot{x}_s d\tau \geq 0, \ \forall t \geq t_0 \tag{4.38}$$

where E_m, E_s is the energy entering from the master and the slave port respectively. From (4.5), (4.10), multiplying by $\dot{x}_m$, $\dot{\tilde{x}}$ and considering only the non-conservative energy components (superscript nc) we have

$$\begin{aligned} E_m^{nc}(t) + E_s^{nc}(t) = &\int_{t_0}^{t} \left(D_m \dot{x}_m^2 + k_f F_e^d \dot{x}_m\right) d\tau + \\ &+ \int_{t_0}^{t} \left(D_s \dot{\tilde{x}}^2 - k_p F_e \dot{x}_m^d\right) d\tau \geq 0, \ \forall t \geq t_0 \end{aligned} \tag{4.39}$$

Due to the presence of delay, inequality (4.39) cannot be monitored as a whole, therefore an observer must be employed at each side of the system, so that the following conditions must be met separately to achieve passivity

$$E_{s,i}^{nc,d}(t) - E_{m,o}^{nc}(t) \geq 0 \qquad \text{at master side} \tag{4.40}$$
$$E_{m,i}^{nc,d}(t) - E_{s,o}^{nc}(t) \geq 0 \qquad \text{at slave side} \tag{4.41}$$

where subscripts i and o indicate positive ingoing and outgoing energy at each port, and d the delayed communicated energy information. Whenever (4.40) and (4.41) are not satisfied, F_e^d and $\dot{x}_m^d$ are modified at master and slave station to dissipate enough energy and maintain passivity. Notice that thanks to the robust impedance assignment approach, the system energy can be explicitly computed via the master and slave parameters as in (4.40), making it possible to compensate for the position drift that usually occurs in this implementation by adapting the delayed master reference for the slave.

Fig. 4.5 presents a comparison between the TDP approach and the proposed one for very high environment stiffness $K_e = 6 \cdot 10^5 N/m$, $D_e = 5 \cdot 10^3 Ns/m$. While both are stable, TDP sacrifices some transparency to meet (4.40), lowering the force feedback magnitude (bottom left). However this is unnecessary, because should we measure the true system energy, we would see that the system remains passive even without force feedback modulation (top right). Therefore, we can provide less conservative results, although passivity is not guaranteed in general.

4.7 Experiments

The approach is validated on the same experimental platform of Sec. 3.8, consisting of a YuMi dual-arm robot prototype, with 7 d.o.f. for each arm. The right arm was operated by a user as a master device, while the second one was employed as the slave manipulator. In the experiments the user held the wrist of the right arm and moved it in order to interact through the slave with a stiff metal slab (Fig. 4.6).

Analogously to the experiments of Chap. 3, to compensate the built-in $10Hz$ FIR filter and the 2-step actuation delay, a 3-step prediction of joint measurements was performed. Interaction forces were reconstructed with a momentum observer, joint velocities obtained by filtered differentiation, and friction approximated with a Coulomb and viscous model.

As in the simulations, we artificially injected a variable communication delay between the two arms, up to $200ms$ in each direction, plus the controllers discretization time of $T_s = 4ms$. Based on (4.31), we chose $\boldsymbol{M}_m = \boldsymbol{M}_s = 3\boldsymbol{I}_3 kg$, $\boldsymbol{D}_m = \boldsymbol{D}_s = 10\boldsymbol{I}_3 Ns/m$, $\boldsymbol{K}_m = 0_3 N/m$, $\boldsymbol{K}_s = 8.33\boldsymbol{I}_3 N/m$, $\tau = 0.3464s$, and $k_p = k_f = 1$ since the two arms

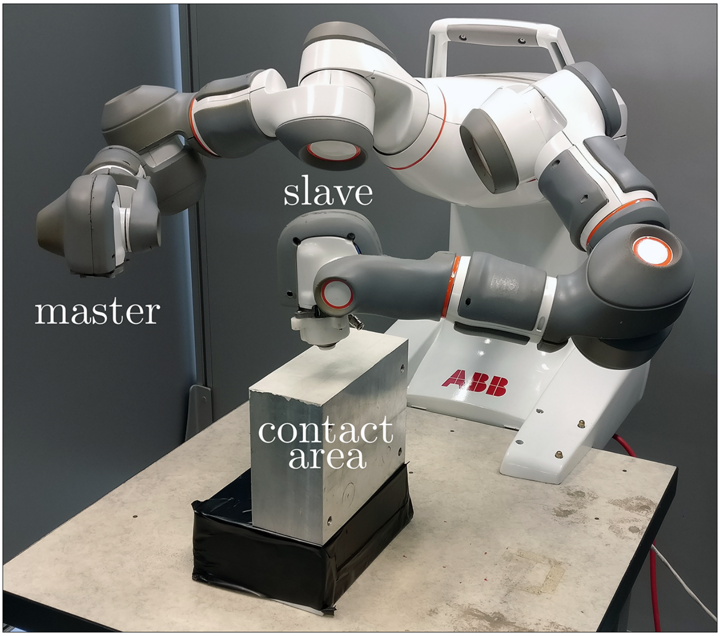


Figure 4.6: *The experimental setup for the experiments validating the proposed teleoperation scheme.*

are kinematically identical. The sliding mode control gains were selected as follows: $\boldsymbol{k}_1 = [12\ 12\ 14\ 14\ 10\ 10\ 10]$ and $\boldsymbol{k}_2 = [8\ 7\ 13\ 13\ 10\ 10\ 10]$. For both arms the redundancy was solved simply by keeping a constant wrist orientation and minimizing joint velocities.

Fig. 4.7 presents the master and slave positions and contact forces during interaction for the proposed tuning. The user was easily able to maintain a stable contact with the environment surface, while the master position was tracked in free motion. As seen in Sec. 3.8, steady state tracking error is sometimes present, with large friction torques dominating the robot dynamics, and the system being mostly driven by the sliding control component (Fig. 4.8).

In a second experiment, the slave impedance parameters were increased by a factor of 3, thus violating the criteria of Sec. 4.4. The results in Fig. 4.9 show the generation of large oscillations that bring the system close to practical instability, and require additional stabilizing effort from the user. This behavior validates the robust approach taken against parasitic dynamics in the definition of (4.31).

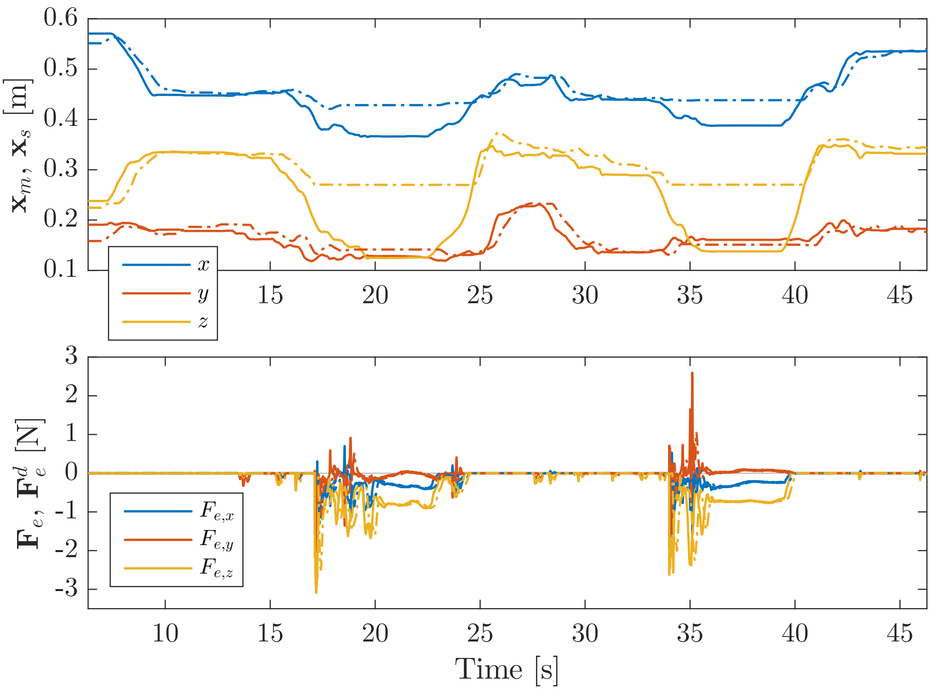


Figure 4.7: *Experiment with the proposed approach and tuning. Top: master (solid) and slave (dashed) end effector position. Bottom: environment force acting on the slave (solid), and delayed filtered force feedback (dashed).*

4.8 Closing comment

We applied the results of the two previous chapters to a bilateral teleoperation system with torque-controlled devices. In particular the virtual fixture-oriented optimization controller of Chap. 2 and the robust operational space MPSMC controller of Chap. 3 have been joined together to obtain a robust bilateral teleoperation architecture. A complete stability and transparency analysis has been carried out, giving precise guidelines to tune the system impedance parameters, and the approach viability has been experimentally tested on a real industrial platform. The following chapter will extend the proposed architecture to multi-arm robots equipped with vision sensors in order to endow the system with partial autonomy via shared control and visual servoing, in presence of complex slave environments.

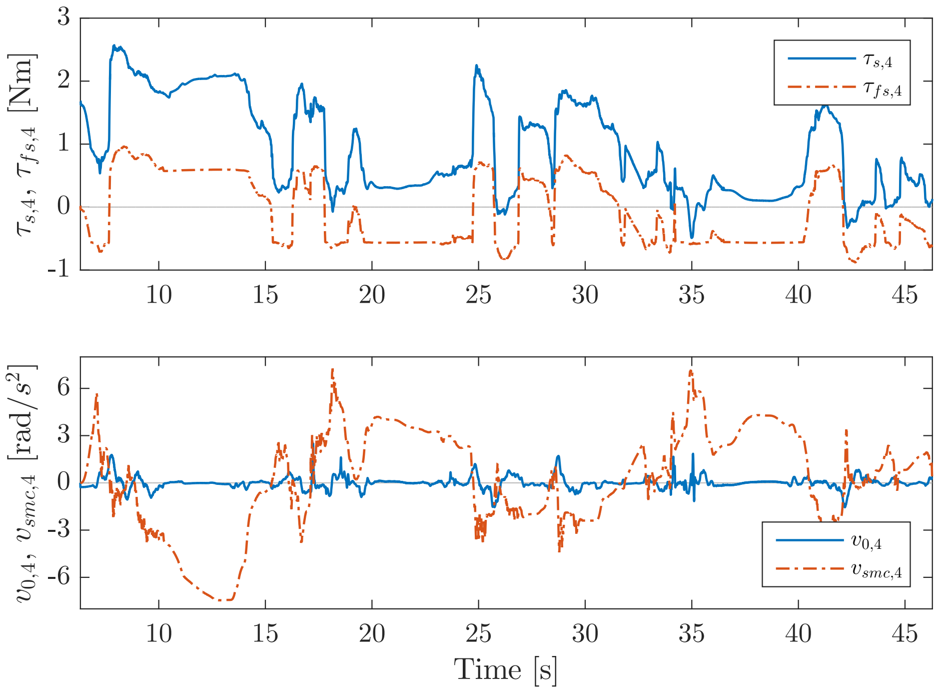


Figure 4.8: *Total motor torque $\boldsymbol{\tau}_s$, friction $\boldsymbol{\tau}_{fs}$, and auxiliary control comparison for the fourth joint of the slave robot.*

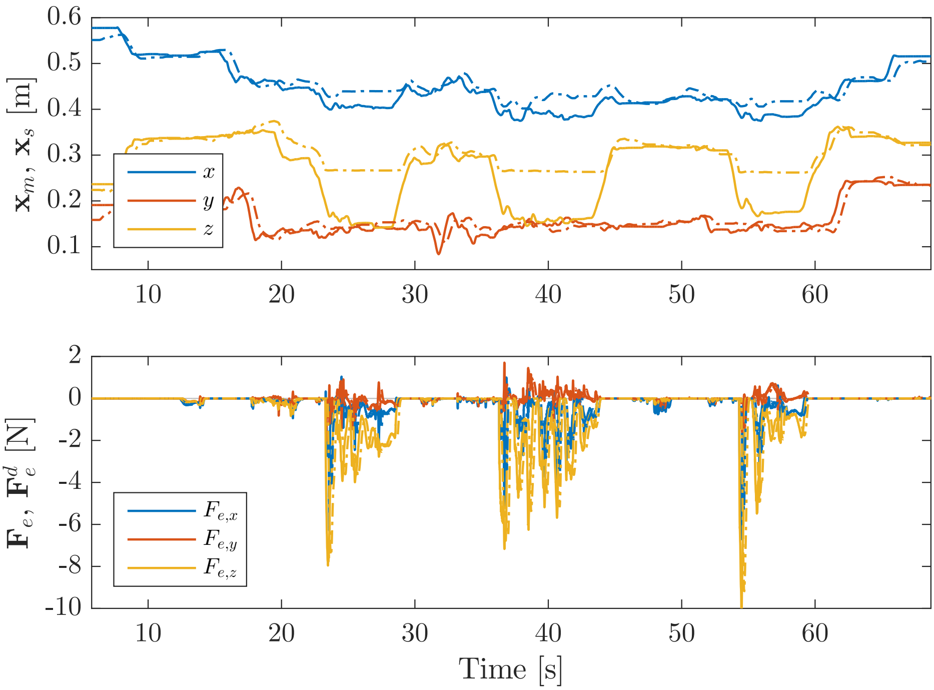


Figure 4.9: *Experiment with unstable tuning. Top: master (solid) and slave (dashed) end effector position. Bottom: environment force acting on the slave (solid), and delayed filtered force feedback (dashed).*

CHAPTER 5

Occlusion-free visual servoed teleoperation

Up to now we have analyzed the proposed teleoperation scheme, in terms of interaction with the environment and force feedback for the user. It is clear that in a remote control application, the visual feedback perceived by the user plays a primary role in the correct execution of the desired task. Even more so, when the environment is highly cluttered and can compromise the remote camera vision, making the control of the slave robot extremely difficult.

In the experimental section of Chap. 2, we briefly considered a visual servoing application. Here we aim to expand the use of visual servoing to dual-arm systems where one arm is remotely operated, while the other one is equipped with a camera and in charge of maintaining at all times the teleoperated tool and goal visible. Such setup enables the shared control of the platform, by the user through the teleoperated arm, and by the control system through the autonomous camera. This allows a more user-friendly teleoperation, since the operator is always aware of the remote location state, but without the burden of controlling the camera himself.

Furthermore, we assume the environment, as well as operator intentions, to be dynamic and potentially unknown. For this reason we adopt a reactive approach, where risks of occlusion are dealt with on-line, without previous path-planning. The hierarchical optimization controller of Chap. 2 and 4 is extended with the inclusion of additional occlusion constraints able to model arbitrary convex occluding objects. Their formulation is derived by highlighting the analogy between occlusion avoidance in the image space and collision avoidance in the 3D space.

Finally, a finite state machine (FSM) applies a switching control policy that performs an evasive camera motion when risk of occlusion is detected, while keeping an intuitive display of the remote scene and control of the teleoperated robot for the user.

The approach is validated experimentally with a reaching task executed on a dual-arm 14 d.o.f. ABB YuMi robot equipped with an eye-in-hand RGB camera, and one arm teleoperated by a 3 d.o.f. Novint Falcon device. Robustness is tested against noise on the image features, as well as dynamic occluding objects and goals. Possible applications may include handling of hazardous material, and surveying in barely accessible disaster area, where debris can occlude the camera view.

In Sec. 5.1 a background on visual servoing in teleoperation is given along with results on current occlusion avoidance techniques. Sec. 5.2 presents the problem settings along with some visual servoing basics, while Sec. 5.3 introduces the occlusion avoidance constraint and its robust formulation. In Sec. 5.4 we define the FSM responsible for switching the controller and solving the impending occlusion. Finally, Sec. 5.5 shows the validation results on our experimental platform.

5.1 Background

In [99] machine vision techniques were used in teleoperation system to compute a force feedback via virtual fixtures, simplifying a remote maintenance task, while Kofman et al. [100] used cameras at the local site to estimate user hand position and orientation, adapted as reference for the teleoperated robot, and employed visual servoing to achieve finer semi-autonomous positioning. Abi-Farraj et al. proposed in [7] a teleoperation system composed of two independent slave robots controlled via visual servoing for radioactive material handling.

Critical issues arise in visual servoing whenever an object of interest is occluded either by itself, by another object, or by the robot arm, or the corresponding image features escape the camera field of view (FoV). When this occurs, the relevant object features are lost, which may result in a control failure, especially if IBVS is used without any recovery technique. In a teleoperation setup this is also undesirable since it removes direct visual feedback of the object in the scene, making navigation more difficult and less intuitive.

Some authors focused on feature estimation to recover control properties during occlusions, either by making environment model assumptions and resorting to Kalman estimators to reconstruct the 3D object state of motion [101], or by using nonlinear observers to estimate point feature depth and thus position in the workspace [102]. Other authors adopted a geometric approach to reconstruct a dynamic 3D object characterized by both point and line features, by relying on the information from multiple cameras [103].

Another approach to deal with occlusions is to plan a camera trajectory that would altogether avoid their occurrence. Kazemi et al. [104] presented an extensive overview of the main path-planning techniques used in visual servoing to guarantee occlusion-free and collision-free trajectories, as well as to consider FoV limitations.

Other authors used potential fields [105] or variable weighting LQ control laws [106] to preserve visibility and avoid self-occlusions. Although these approaches are suitable for real-time implementation, they may exhibit local minima and possible unwanted oscillations [107].

Global path-planning can overcome these limitations by computing a priori occlusion-free trajectories, however it often requires long computational time and accurate workspace knowledge, unsuitable for reactive motions in dynamic environments [108]. The applicability of the navigation functions proposed in [109] is instead limited to simple scenarios.

Time complexity is a major obstacle also in optimization-based path-planning, where the solver computes a suitably parametrized optimal trajectory, while respecting camera constraints [110]. These approaches have the merit of considering arbitrary constraints but are hardly applicable in complex contexts, as the optimization might become non-convex.

Inherently reactive control techniques, such as MPC, are also time demanding when increasing the time horizon [111], although a local or global optimization can be conducted based on the knowledge of the environment.

In [112] the authors proposed a two-step path-planning algorithm merging together potential fields with convex optimization techniques to ensure feature visibility in the image plane, while in [113] qualitative visual servoing was proposed to trade-off feature positioning and visibility, by expressing only a confidence interval for said features. Nevertheless, in a teleoperation framework the resulting solution without proper redundancy management could generate undesired camera motions.

Folio et al. [114] proposed a solution to the occlusion avoidance problem for mobile robots by switching controller whenever an imminent occlusion is detected. The algorithm tries to push the occluding object outside the camera FoV via IBVS, however it is limited to planar camera motions and rotations around the vertical axis.

Quadratic programming techniques have seen increasing popularity also in visual servoing robotic application. Differently from the discussed global trajectory optimizers, these solutions find an optimal robot motion in real-time based on camera sensor measurements. In [115] nonlinear optimization was applied to UAVs for visual servoed target tracking in cluttered environments. These algorithms have been employed also in complex systems of high dimensionality, such as the whole-body control of a humanoid robot [116], where visual servoing was integrated with FoV and occlusion constraints, although in a simple example where the wall obstructing the robot gaze essentially divided the image plane in two half-planes.

5.2 System requirements and visual servoing

The considered robotic system is composed of two slave robot arms and a master device (Fig. 5.1). One arm is remotely operated by a user via the master device that controls the robot TCP velocity. The other arm is autonomous and camera-equipped with the purpose of producing an optimal view of the workspace for the user by keeping the teleoperated tool and the goal inside the FoV, as well as avoiding occlusions. The camera feed is the only visual cue available to the user.

Overall, the requirements for the system during a remote reaching task can be summarized as follows:

1. Autonomous continuous camera positioning. The user should not worry about controlling the camera robot.

2. Kinematic coordination (tracking) between master and slave teleoperated robot.

3. Teleoperated TCP and user-selected goal visible at all times in the camera FoV.

4. Autonomous camera repositioning to avoid occlusions due to additional objects in the scene.

5. Natural and intuitive camera motion and reference mapping of the master device in the camera frame. The user should clearly under-

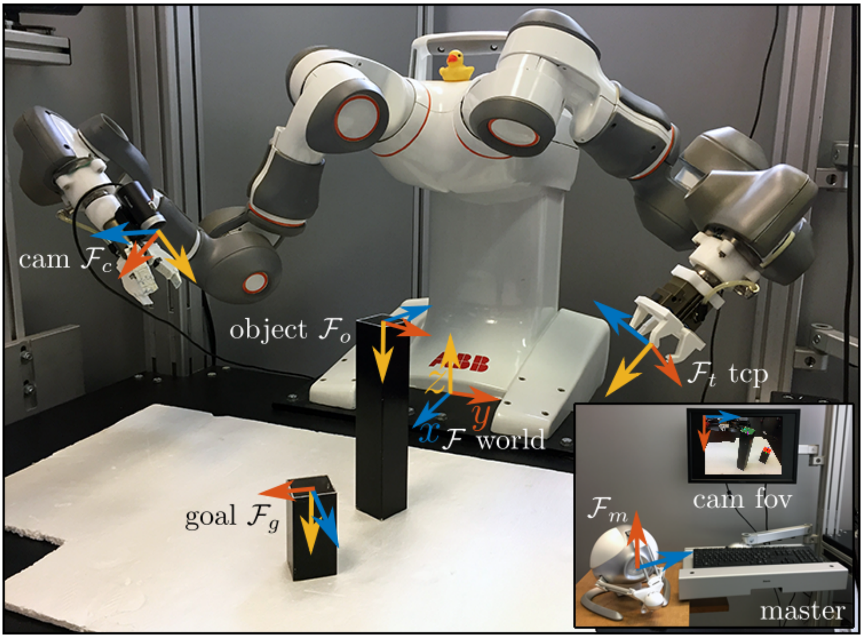


Figure 5.1: *The experimental setup for the proposed visual servoed dual-arm teleoperation scheme.*

stand what is happening at the remote scene, and how to influence the system behavior.

We define the following frames as shown in Fig. 5.1: $\mathcal{F}$ is the world frame, $\mathcal{F}_m$ is the master device base frame, $\mathcal{F}_c$ is the camera frame, $\mathcal{F}_t$ is the tool frame of the teleoperated arm, $\mathcal{F}_g$ is the frame of the goal that the user wishes to reach, and $\mathcal{F}_o$ is the frame of an object that can possibly occlude the tool or the goal.

For the camera we adopt the standard pinhole model. $\boldsymbol{\pi}$ identifies the image plane parallel to the (x_c, y_c) plane, and λ is the camera focal length. Given a 3D point $P(x^c, y^c, z^c)$ expressed in $\mathcal{F}_c$, the corresponding point $p(u, v)$ in $\boldsymbol{\pi}$ is given by the projection equations

$$u = \frac{\lambda x^c}{d_x z^c} + o_u, \quad v = \frac{\lambda y^c}{d_y z^c} + o_v \tag{5.1}$$

where (d_x, d_y) are the pixels dimensions in the (x_c, y_c) coordinates, and (o_u, o_v) offset the position of the first pixel.

For camera control and occlusion avoidance we employ IBVS. The regulation is done directly at the image feature level, for this reason it is necessary to relate the features movements in the image plane to the robot joint velocities. Given a vector of generic image features of an object $\boldsymbol{s}_o$, its time derivative depends on the joint velocities

$$\dot{\boldsymbol{s}}_o = \boldsymbol{L}_o \dot{\boldsymbol{x}}_c^c = \boldsymbol{L}_o \boldsymbol{J}_c^c \dot{\boldsymbol{q}}_c = \boldsymbol{P}_{c,o} \dot{\boldsymbol{q}}_c \tag{5.2}$$

where $\boldsymbol{L}_o$ is the interaction matrix, $\boldsymbol{J}_c^c$ the camera Jacobian expressed in $\mathcal{F}_c$ and $\dot{\boldsymbol{q}}_c$ the camera robot joint velocities. Note that if the considered features are those of a moving object, in (5.2) appears an additional term due to the object own movement. We will show later in Sec. 5.5 that our controller is robust with respect to this disturbance, even if the object dynamics cannot be estimated.

Nevertheless, in the particular case where the teleoperated TCP is considered, the corresponding features are the image point coordinates $\boldsymbol{p}_t = [u_t \;\; v_t]^T$, and its motion can be compensated since it depends on the arm joint velocities $\dot{\boldsymbol{q}}_t$.

$$\begin{aligned} \dot{\boldsymbol{p}}_t &= \boldsymbol{L}_t \begin{bmatrix} \boldsymbol{v}_c^c - \boldsymbol{v}_t^c \\ \boldsymbol{\omega}_c^c \end{bmatrix} = \boldsymbol{L}_t \boldsymbol{J}_c^c \dot{\boldsymbol{q}}_c + \boldsymbol{L}_t \begin{bmatrix} -\boldsymbol{R}^c & 0 \\ 0 & 0 \end{bmatrix} \boldsymbol{J}_t \dot{\boldsymbol{q}}_t = \\ &= \begin{bmatrix} \boldsymbol{P}_{c,t} & \boldsymbol{P}_{t,t} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{q}}_c \\ \dot{\boldsymbol{q}}_t \end{bmatrix} = \boldsymbol{P}_t \dot{\boldsymbol{q}} \end{aligned} \tag{5.3}$$

where $\boldsymbol{v}_t^c$ is the tool translational velocity expressed in $\mathcal{F}_c$, $\boldsymbol{J}_t$ the tool Jacobian in $\mathcal{F}$, and $\boldsymbol{R}^c$ the rotation matrix from world to camera frame.

Given the image features reference $\boldsymbol{p}_t^*$, and the error dynamics

$$\dot{\boldsymbol{p}}_t = \boldsymbol{P}_t \dot{\boldsymbol{q}} = k_p(\boldsymbol{p}_t^* - \boldsymbol{p}_t) \tag{5.4}$$

the classic IBVS controller produces the following control law

$$\dot{\boldsymbol{q}} = k_p \boldsymbol{P}_t^\dagger (\boldsymbol{p}_t^* - \boldsymbol{p}_t) + \boldsymbol{N}_t \dot{\boldsymbol{q}}_0 \tag{5.5}$$

where $\boldsymbol{P}_t^\dagger$ is the pseudo-inverse of $\boldsymbol{P}_t$ and $\boldsymbol{N}_t$ its null-space projector, k_p is the controller gain, and $\dot{\boldsymbol{q}}$ the joint reference vector for the low level controllers of the two robots. $\dot{\boldsymbol{q}}_0$ can be used to exploit the remaining degrees of freedom.

For the considered system, the second requirement is kinematic coordination between master and slave devices, which can be implemented via the following error dynamics

$$\dot{\boldsymbol{x}}_t = \begin{bmatrix} 0 & \boldsymbol{J}_t \end{bmatrix} \dot{\boldsymbol{q}} = k_p(\boldsymbol{x}_t^* - \boldsymbol{x}_t) + \dot{\boldsymbol{x}}_t^* \tag{5.6}$$

$\boldsymbol{x}_t$ and $\boldsymbol{x}_t^*$ are the teleoperated robot pose and its reference, respectively. Substituting $\dot{\boldsymbol{q}}$ in (5.6) with (5.5) yields $\dot{\boldsymbol{q}}_0$ by means of the pseudo-inverse, and the final joint velocities are obtained by substituting back in (5.5). The resulting control law will move the camera to regulate $\boldsymbol{p}_t$ in the FoV, while allowing the user to teleoperate the robot as desired.

Unfortunately, as seen in previous chapters, this standard approach does not consider specifications that do not require an explicit regulation, but just a generic limitation. This is the case for the considered system, where we want to ensure that the relevant image features remain inside the FoV (visibility) and are not occluded by other objects (occlusion avoidance), but we do not want to explicitly define a reference for these quantities. For this reason, we employ and adapt the optimization-based formalism detailed in Sec. 4.3 that allows the explicit inclusion of motion constraints via both equalities (visual servoing) and inequalities (visibility, occlusion avoidance). Therefore, in the following we will assume to have used the robust feedback linearization approach (Sec. 4.2), so that the system is simply described by the integrator chain (3.32), with $\boldsymbol{v}_0$ the auxiliary nominal control, i.e. the robot desired acceleration (for simplicity we drop the slave subscript s).

5.3 Occlusion avoidance constraint

In this section we present the formulation of the occlusion avoidance constraint in a form compatible with the presented optimization problem.

Let us consider the feature point $\boldsymbol{p}_t$ for which we want to avoid the occlusion, and the feature points characterizing a potentially occluding convex object. Depending on the object geometry, these points are connected to each other and they define a polytope in $\boldsymbol{\pi}$, that $\boldsymbol{p}_t$ is not allowed to enter if its depth is higher than that of the object points. What we need is a formulation of this requirement in terms of the two robots velocities.

In [117], it has been shown how a safety constraint avoiding collision between a robot rigid link and the human can be derived from geometric and kinematic arguments. The authors managed to obtain a minimum distance criterion in the form of an inequality on robot velocities. Here we translate this criterion to the image plane. If $\boldsymbol{p}_t$ starts from a visible position, we can avoid the occlusion by having the feature point never "collide" with the forbidden area.

5.3.1 Constraint definition

Without loss of generality, we consider a single pair of connected object feature points defining the occlusion area boundary, the final set of constraints satisfying the occlusion avoidance requirement will be straightfor-

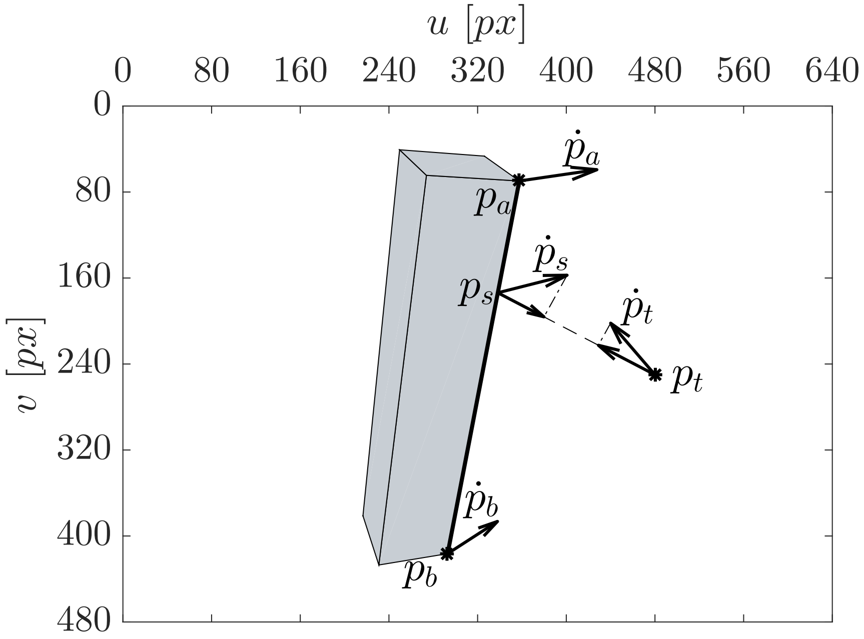


Figure 5.2: *Pictorial representation of the occlusion avoidance constraint in the image plane. The gray area identifies the projection of the occluding object.*

wardly obtained by applying the following computations for every adjacent pair.

Fig. 5.2 shows the segment connecting two object feature points $\boldsymbol{p}_a$ and $\boldsymbol{p}_b$ along with the point $\boldsymbol{p}_t$ that should avoid the occlusion. For simplicity, we assume the segment's depth to be less than that of $\boldsymbol{p}_t$, so that an occlusion can indeed occur. If this is not the case, we can just consider the sub-segment satisfying this condition.

The position and velocity of the generic point $\boldsymbol{p}_s$ on the segment can be expressed as follows

$$\boldsymbol{p}_s = \boldsymbol{p}_a + s(\boldsymbol{p}_b - \boldsymbol{p}_a), \quad \dot{\boldsymbol{p}}_s = \dot{\boldsymbol{p}}_a + s(\dot{\boldsymbol{p}}_b - \dot{\boldsymbol{p}}_a) \tag{5.7}$$

where $s \in [0, 1]$. By applying the minimum distance criterion we have

$$(\boldsymbol{p}_t - \boldsymbol{p}_s)^T(\boldsymbol{p}_t - \boldsymbol{p}_s) - t_b(\boldsymbol{p}_t - \boldsymbol{p}_s)^T(\dot{\boldsymbol{p}}_s - \dot{\boldsymbol{p}}_t) \geq 0, \forall s \in [0, 1] \tag{5.8}$$

The first term on the left represents the squared distance between the feature of interest and each of the points on the segment delimiting the occlusion area, while the second one is proportional to the projection of the relative velocity of the two points on the line connecting them. t_b is a design parameter that relates to the maximum time required by the robot to bring to a halt the features in the image. Substituting (5.7) in (5.8), for the first term we have

$$(\boldsymbol{p}_t - \boldsymbol{p}_s)^T(\boldsymbol{p}_t - \boldsymbol{p}_s) = \alpha s^2 + \beta s + \gamma \tag{5.9a}$$
$$\alpha = (\boldsymbol{p}_b - \boldsymbol{p}_a)^T(\boldsymbol{p}_b - \boldsymbol{p}_a) \tag{5.9b}$$
$$\beta = 2(\boldsymbol{p}_a - \boldsymbol{p}_t)^T(\boldsymbol{p}_b - \boldsymbol{p}_a) \tag{5.9c}$$
$$\gamma = (\boldsymbol{p}_a - \boldsymbol{p}_t)^T(\boldsymbol{p}_a - \boldsymbol{p}_t) \tag{5.9d}$$

Similarly, for the second term

$$- t_b(\boldsymbol{p}_t - \boldsymbol{p}_s)^T(\dot{\boldsymbol{p}}_s - \dot{\boldsymbol{p}}_t) = \alpha' s^2 + \beta' s + \gamma' \tag{5.10a}$$
$$\alpha' = t_b(\boldsymbol{p}_b - \boldsymbol{p}_a)^T(\dot{\boldsymbol{p}}_b - \dot{\boldsymbol{p}}_a) \tag{5.10b}$$
$$\beta' = t_b[(\boldsymbol{p}_a - \boldsymbol{p}_t)^T(\dot{\boldsymbol{p}}_b - \dot{\boldsymbol{p}}_a) + (\boldsymbol{p}_b - \boldsymbol{p}_a)^T(\dot{\boldsymbol{p}}_a - \dot{\boldsymbol{p}}_t)] \tag{5.10c}$$
$$\gamma' = t_b(\boldsymbol{p}_a - \boldsymbol{p}_t)^T(\dot{\boldsymbol{p}}_a - \dot{\boldsymbol{p}}_t) \tag{5.10d}$$

Then the constraint can be rewritten as

$$(\alpha + \alpha')s^2 + (\beta + \beta')s + (\gamma + \gamma') \geq 0, \ \forall s \in [0, 1] \tag{5.11}$$

For the previous equation to be valid $\forall s \in [0, 1]$ it is sufficient to check the minima $\overline{s}$ of the parabola in this interval, given the current feature coordinates and velocities.

Note that, since the segment is not a rigid body in the image plane, it is possible to have $\alpha' \neq 0$, thus we cannot simply evaluate (5.8) by computing $\min_s \|\boldsymbol{p}_t - \boldsymbol{p}_s\|$ and checking the term on the right side only in $s = 0, 1$ as done in [117].

If $\alpha + \alpha' > 0$ then the parabola is convex and the minimum is a point in the $[0, 1]$ interval, else if $\alpha + \alpha' < 0$ the parabola is concave and the minimum is necessarily one (or both) of the segment ends. Notice also that $\alpha + \alpha' = 0$ only if $\boldsymbol{p}_a$ and $\boldsymbol{p}_b$ are the same point or, given the current velocities, they will overlap in a time t_b.

By applying the relation between feature derivatives and joint velocities to the segment point features $\boldsymbol{p}_a$, $\boldsymbol{p}_b$, we have

$$\dot{\boldsymbol{p}}_a = \boldsymbol{P}_{c,a}\dot{\boldsymbol{q}}_c \tag{5.12}$$
$$\dot{\boldsymbol{p}}_b = \boldsymbol{P}_{c,b}\dot{\boldsymbol{q}}_c \tag{5.13}$$

For point $\boldsymbol{p}_t$ we can proceed similarly, but without loss of generality in the following we will consider the case where $\boldsymbol{p}_t$ is the teleoperated tool point feature. By substituting the minimum coordinate $\overline{s}$, (5.3), (5.12) and (5.13) in (5.11) we obtain the occlusion avoidance constraint in the form

$$\begin{bmatrix} \boldsymbol{A}_c & \boldsymbol{A}_t \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{q}}_c \\ \dot{\boldsymbol{q}}_t \end{bmatrix} \geq b \tag{5.14}$$

$$\begin{aligned} \boldsymbol{A}_c = t_b \big\{ &(\boldsymbol{p}_b - \boldsymbol{p}_a)^T (\boldsymbol{P}_{c,b} - \boldsymbol{P}_{c,a}) \overline{s}^2 + \\ &+ \left[(\boldsymbol{p}_b - \boldsymbol{p}_a)^T (\boldsymbol{P}_{c,a} - \boldsymbol{P}_{c,t}) + (\boldsymbol{p}_a - \boldsymbol{p}_t)^T (\boldsymbol{P}_{c,b} - \boldsymbol{P}_{c,a}) \right] \overline{s} + \\ &+ (\boldsymbol{p}_a - \boldsymbol{p}_t)^T (\boldsymbol{P}_{c,a} - \boldsymbol{P}_{c,t}) \big\} \end{aligned} \tag{5.15}$$

$$\boldsymbol{A}_t = -t_b \left[(\boldsymbol{p}_b - \boldsymbol{p}_a)^T \overline{s} + (\boldsymbol{p}_a - \boldsymbol{p}_t)^T \right] \boldsymbol{P}_{t,t} \tag{5.16}$$

$$b = -(\alpha \overline{s}^2 + \beta \overline{s} + \gamma) \tag{5.17}$$

Note that if $\boldsymbol{p}_t$ were the feature of a generic point, in the previous equation we would have $\boldsymbol{P}_{t,t} = 0$ and thus $\boldsymbol{A}_t = 0$, this shows that the formulation is general and applicable also to autonomous single-arm VS, without teleoperator. Furthermore, if the occluding object dynamics are known or otherwise estimated, they can be directly considered in (5.12), (5.13) by adding the terms $-\boldsymbol{L}_a [\boldsymbol{v}_a^{cT} \ 0^T]^T$, $-\boldsymbol{L}_b [\boldsymbol{v}_b^{cT} \ 0^T]^T$, where $\boldsymbol{v}_a^c$ and $\boldsymbol{v}_b^c$ are the velocities of the corresponding points on the 3D object. The same can be argued for the goal if it is the considered feature instead of the teleoperated TCP. Nonetheless, we will prove that this knowledge is, to a certain extent, unnecessary, thanks to the inherent robustness of the approach, which is discussed in the next section.

In order to obtain a constraint depending on the joint accelerations, we apply Grönwall's lemma by defining $f = \boldsymbol{A}_c \dot{\boldsymbol{q}}_c + \boldsymbol{A}_t \dot{\boldsymbol{q}}_t - b$, the inequality $\dot{f} + \delta f \geq 0$ yields an acceleration-based constraint that satisfies and exponentially converges to (5.14), where δ is the speed of convergence. The final acceleration-based constraint is then the following one

$$\begin{bmatrix} \boldsymbol{A}_c & \boldsymbol{A}_t \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_{0,c} \\ \boldsymbol{v}_{0,t} \end{bmatrix} \geq -\delta \left(\begin{bmatrix} \boldsymbol{A}_c & \boldsymbol{A}_t \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{q}}_c \\ \dot{\boldsymbol{q}}_t \end{bmatrix} - b \right) - \begin{bmatrix} \dot{\boldsymbol{A}}_c & \dot{\boldsymbol{A}}_t \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{q}}_c \\ \dot{\boldsymbol{q}}_t \end{bmatrix} + \dot{b} = b' \tag{5.18}$$

After repeating the procedure for all the segments defining the occlusion area, all the resulting inequality constraints can be immediately included in the optimization to prevent the occlusion event.

5.3.2 Robust reformulation

The obtained constraint ensures occlusion avoidance in a nominal scenario. In real applications, sources of uncertainty, such as camera calibration and measurement noise, might actually produce a violation of the constraint and possibly an occlusion event. Additionally, the numerical approximations of the optimization discrete implementation, often produce chattering of the control variables, due to intermittent constraint activation. In practice these events make the constraint unreliable, since some non-recoverable occlusions could occur: the feature of interest might enter the forbidden region and remain inside, due to the constraint now working on the wrong side of the segments defining the polytope.

To solve this problem it is necessary to take this uncertainty directly into account in the constraint formulation. Assuming bounded uncertainties, we can follow the approach in [47] to confer robustness to the constraint. Given the vectors $\Delta_p \in \mathbb{D}_p$ and $\Delta_{\dot{p}} \in \mathbb{D}_{\dot{p}}$, respectively modeling the point features relative position and velocity uncertainties, (5.8) can be rewritten as follows

$$(\boldsymbol{p}_t - \boldsymbol{p}_s + \Delta_p)^T(\boldsymbol{p}_t - \boldsymbol{p}_s + \Delta_p) - t_b(\boldsymbol{p}_t - \boldsymbol{p}_s + \Delta_p)^T(\dot{\boldsymbol{p}}_s - \dot{\boldsymbol{p}}_t + \Delta_{\dot{p}}) \geq 0, \quad \forall s \in [0,1], \Delta_p \in \mathbb{D}_p, \Delta_{\dot{p}} \in \mathbb{D}_{\dot{p}} \tag{5.19}$$

Then, what we are asking is that the constraint should be satisfied for a whole set of states and controls, an approach often used in robust MPC.

Note that this is not the same as simply adding a margin to the nominal constraint. In fact, uncertainty terms that depend on the state and control appear and cannot be easily bounded. Similarly to the nominal case, by finding the minimum of (5.19) to obtain the worst case scenario, and expanding the computations, we have

$$\begin{aligned}\boldsymbol{A}_c = t_b \big\{ &(\boldsymbol{p}_b - \boldsymbol{p}_a)^T(\boldsymbol{P}_{c,b} - \boldsymbol{P}_{c,a})\overline{s}^2 + \\ &+ \left[(\boldsymbol{p}_b - \boldsymbol{p}_a)^T(\boldsymbol{P}_{c,a} - \boldsymbol{P}_{c,t}) + (\boldsymbol{p}_a - \boldsymbol{p}_t + \Delta_p)^T(\boldsymbol{P}_{c,b} - \boldsymbol{P}_{c,a})\right]\overline{s} + \\ &+ (\boldsymbol{p}_a - \boldsymbol{p}_t + \Delta_p)^T(\boldsymbol{P}_{c,a} - \boldsymbol{P}_{c,t})\big\}\end{aligned} \tag{5.20}$$

$$\boldsymbol{A}_t = -t_b\left[(\boldsymbol{p}_b - \boldsymbol{p}_a)^T\overline{s} + (\boldsymbol{p}_a - \boldsymbol{p}_t + \Delta_p)^T\right]\boldsymbol{P}_{t,t} \tag{5.21}$$

$$\begin{aligned}b = -\{&(\boldsymbol{p}_b - \boldsymbol{p}_a)^T(\boldsymbol{p}_b - \boldsymbol{p}_a)\overline{s}^2 + 2[(\boldsymbol{p}_a - \boldsymbol{p}_t)^T(\boldsymbol{p}_b - \boldsymbol{p}_a) + t_b\Delta_{\dot{p}}^T(\boldsymbol{p}_b - \boldsymbol{p}_a)]\overline{s} + \\ &+ (\boldsymbol{p}_a - \boldsymbol{p}_t + \Delta_p)^T(\boldsymbol{p}_a - \boldsymbol{p}_t + \Delta_p) + t_b\Delta_{\dot{p}}^T(\boldsymbol{p}_a - \boldsymbol{p}_t + \Delta_p)\}\end{aligned} \tag{5.22}$$

Compared to the nominal matrices (5.15), (5.16) we have the following uncertainty-affected terms

$$(\boldsymbol{p}_a - \boldsymbol{p}_t + \Delta_p)^T(\boldsymbol{P}_{c,b} - \boldsymbol{P}_{c,a})\overline{s} \tag{5.23}$$

$$(\boldsymbol{p}_a - \boldsymbol{p}_t + \Delta_p)^T(\boldsymbol{P}_{c,a} - \boldsymbol{P}_{c,t}) \tag{5.24}$$

$$(\boldsymbol{p}_a - \boldsymbol{p}_t + \Delta_p)^T\boldsymbol{P}_{t,t} \tag{5.25}$$

Since these matrices in the end multiply the control variable, the feature relative position uncertainty essentially describes a distortion of the actuation channel each control variable is acting on. This affects both the segment itself (first term) and the point-segment relative velocity (second term). Clearly, describing this effect with a simple margin is not trivial. In (5.22), the following terms appear instead

$$\Delta_{\dot{p}}^T (\boldsymbol{p}_b - \boldsymbol{p}_a)\overline{s} \tag{5.26}$$

$$\Delta_{\dot{p}}^T (\boldsymbol{p}_a - \boldsymbol{p}_t + \Delta_p) \tag{5.27}$$

$$(\boldsymbol{p}_a - \boldsymbol{p}_t + \Delta_p)^T (\boldsymbol{p}_a - \boldsymbol{p}_t + \Delta_p) \tag{5.28}$$

which describe the influence of the feature relative velocity on the constraint. Also in this case, most of the uncertain terms depend on the state (segment or point-segment relative position), the usage of a constant margin would be, first, hard to compute, and second, overly conservative compared to the proposed approach, since it should bound all possible uncertainty occurrences.

Moreover, a simple margin does not help in mitigating the chattering phenomenon arising in optimization-based controllers, due to the fast activation and deactivation of constraints (e.g. caused by numerical rounding) [118]. In fact, a constant margin simply moves the constraint, but does not prevent the true state from crossing the boundary defined by the nominal constraint and generating chattering. This is instead alleviated with the presented set-robust formulation.

With this approach we are also implicitly making the constraint more robust against unmodeled movements of the considered 3D objects, since they turn out to be additional feature point motions unforeseen by the control. Additionally, interaction matrix uncertainties are partially considered. Indeed, for a generic feature point we can write the relation between its derivative and the robot joint velocities

$$\dot{\boldsymbol{p}} = \boldsymbol{P}\dot{\boldsymbol{q}} = (\boldsymbol{P}_n + \delta\boldsymbol{P})\dot{\boldsymbol{q}} = \boldsymbol{P}_n\dot{\boldsymbol{q}} + \delta\boldsymbol{P}\dot{\boldsymbol{q}} = \boldsymbol{P}_n\dot{\boldsymbol{q}} + \Delta_{\dot{p}} \tag{5.29}$$

where $\boldsymbol{P}_n$ is the nominal interaction matrix (e.g. the one obtained through calibration and pose estimation), while $\delta\boldsymbol{P}$ is the uncertainty on the interaction matrix. Our robust approach is based on the fact that the uncertainties are bounded, hence also $\delta\boldsymbol{P}$. Nonetheless, we must make some further considerations.

In some entries of $\boldsymbol{P}$ the point z coordinate in the camera frame and the focal length appear at the denominator of some terms. It is easy to see that while an overestimation will produce a bounded uncertainty on

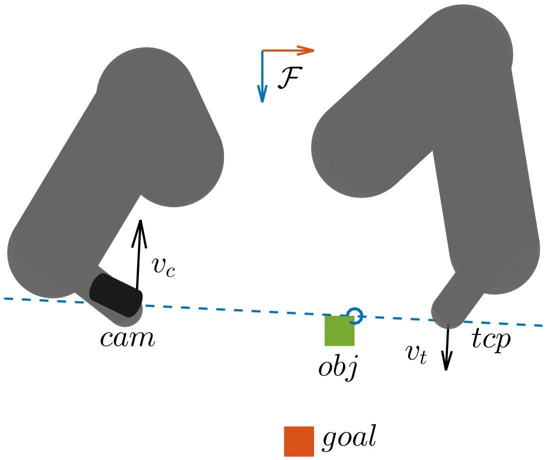


Figure 5.3: *The considered dual-arm system in the $x - y$ plane. To avoid the occlusion the camera pivots around the object.*

the affected elements of $\delta \boldsymbol{P}$, an underestimation of the pose coordinate will quickly make these terms diverge to infinity. In our scenario, this will mean that the controller will opt for a smaller camera movement to avoid the occlusion, since the object is thought to be near the camera, while in reality it is far away, thus the resulting motion will not be enough to prevent the occlusion. However, requiring robustness for all possible calibration and reconstruction errors is impractical and for reasonable calibrations the presented approach produces reliably robust results.

Finally, the application of Grönwall's lemma as in (5.18) gives the new robust constraint

$$\begin{bmatrix} \tilde{\boldsymbol{A}}_c & \tilde{\boldsymbol{A}}_t \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_{0,c} \\ \boldsymbol{v}_{0,t} \end{bmatrix} \geq \tilde{b}' \tag{5.30}$$

where the tilde indicates matrices accounting for the uncertainty.

5.4 Finite state machine and controllers

The set of constraints derived in the previous section guarantees that the robot system will not perform a trajectory leading to an occlusion. However, the simple inclusion of such constraints does not ensure a natural movement of the camera for a reaching task. In fact, imagine to control the robot TCP to a point in the image, and teleoperate the arm to induce an occlusion. Upon constraint activation, the controller will produce a camera motion around the corresponding object segment in the 3D space to avoid the occlusion (Fig. 5.3). Although effective, we have no control over this

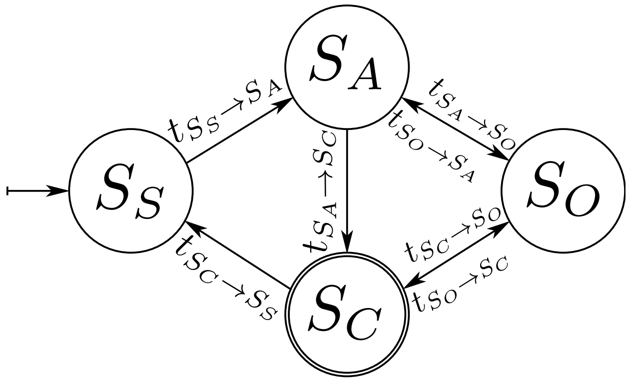


Figure 5.4: *The finite state machine for the teleoperation system.*

motion and no guarantees that it will be optimal from the user point of view (e.g. instead of going around the object, we could simply go above it, see Fig. 5.1).

For this reason we divide the reaching task in four different phases, where each one employs its own controller to regulate different sets of quantities and generate a user-friendly camera behavior. To model the phase transitions we employ a finite state machine, as sketched in Fig. 5.4. Since the control variables are fundamentally joint accelerations, we can guarantee the continuity of the velocity during the transitions. Therefore, it is not mandatory to employ task sequencing and smoothing techniques to avoid mechanical resonances excitation [119], although they can be integrated in the proposed approach if also continuous accelerations are required.
In the following we provide a description of each phase.

5.4.1 Setup state: S_S

The setup phase is the system starting state. We assume that the user has previously selected a goal from the image, and that at the start it is visible in the FoV. At the beginning it is not necessary for the teleoperated TCP to be in the FoV, since its position can be obtained from the robot joint values and then projected with (5.1) on the image plane.

In this state we autonomously regulate the tool and the goal to predefined positions in the image without any user input. We divide the image in four regions, one quadrant will be occupied by the TCP while the opposite one by the goal (Fig. 5.5). For the tool $\boldsymbol{p}_t$ we choose the nearest region in the plane, while the goal position is automatically chosen. To per-

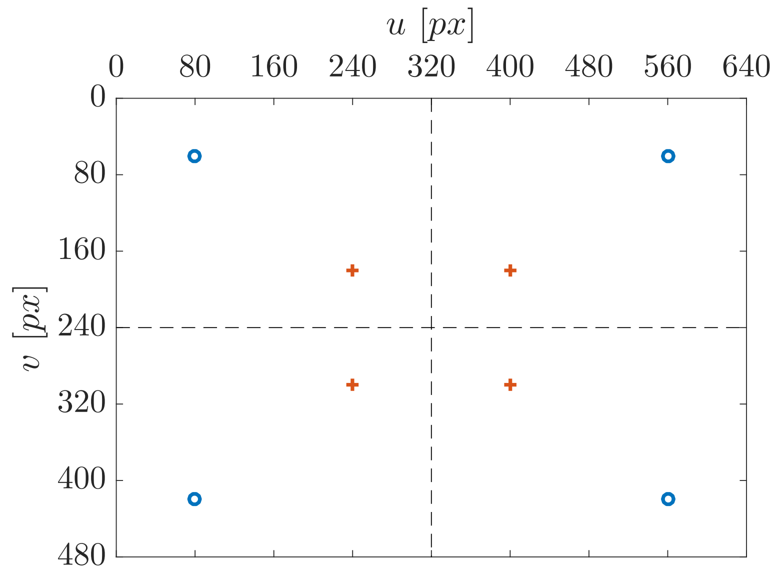


Figure 5.5: *Quadrant division of the camera FoV, TCP and goal references, blue circles and red crosses, respectively.*

form the goal regulation, we use the position of its image center of mass (CoM) $\boldsymbol{p}_{g,com}$ (see [120] for the computation of image moments and their interaction matrix).

Although such strict regulation is not mandatory, we think that restricting the features the user is interested in may be beneficial to better understand what is happening at the remote location, since it indirectly favors more predictable and less sudden camera movements. It also presents a view of the workspace that is always the same independently of the particular environment configuration, so that the user can clearly expect quantities of interest in precise regions of the FoV.

A formulation of these requirements compatible with the optimization problem can be obtained by defining second order dynamics for TCP and goal CoM feature regulation errors, $\boldsymbol{e}_t$ and $\boldsymbol{e}_{g,com}$ respectively. Via the appropriate interaction matrix and robot Jacobian it is possible to make the control variable dependence explicit within the optimization. This procedure will be used for all the following requirements. As for the constraints, here we include visibility for the goal $(\underline{\boldsymbol{p}}, \overline{\boldsymbol{p}})$ and robot joint limits $(\underline{\boldsymbol{q}}, \overline{\boldsymbol{q}})$, to ensure that the goal is kept at all times within the camera FoV. In compact notation, the optimization has the following cost function and constraints

$$\boldsymbol{v}_0 = \arg\min_{\boldsymbol{v}_0} \|\boldsymbol{e}_t\|^2_{\boldsymbol{Q}_t} + \|\boldsymbol{e}_{g,com}\|^2_{\boldsymbol{Q}_{g,com}} \tag{5.31a}$$

$$s.t. \quad \underline{\boldsymbol{q}} \leq \boldsymbol{q}(\boldsymbol{v}_0) \leq \overline{\boldsymbol{q}} \tag{5.31b}$$

$$\underline{\boldsymbol{p}} \leq \boldsymbol{p}_g(\boldsymbol{v}_0) \leq \overline{\boldsymbol{p}} \tag{5.31c}$$

The state machine switches to S_A when the regulation errors are below a certain threshold.

$$t_{S_S \to S_A} : \text{ if } \|\boldsymbol{e}_t\| \leq e_{thr} \wedge \|\boldsymbol{e}_{g,com}\| \leq e_{thr} \tag{5.32}$$

5.4.2 Approach state: S_A

After the setup, teleoperation is enabled and the user can start moving the robot arm. The inputs in the master device frame $\mathcal{F}_m$ are mapped into references for the TCP expressed in the camera frame $\mathcal{F}_c$ for an intuitive usage of the platform.

In this state, the objective is to have the camera zoom in on the scene by getting closer as the TCP approaches the goal, thus improving the image resolution for the user. A solution is to keep regulating these two points at the respective references used in S_S. Furthermore, we prevent any camera rotation around the $\mathcal{F}_c$ z-axis to avoid unexpected camera behavior, by regulating the angle ϕ to its starting value.

Therefore, if the user does not move the master device, the whole system remains still. Instead, when the user interacts with the master, the teleoperated robot executes the command while the camera moves mostly along its z-axis. Note that in the image the TCP projection never moves.

With respect to the previous controller, we add the camera angle ϕ regulation and the TCP pose tracking of the master, through the desired impedance relation $\boldsymbol{I}_{t,tele}$ (4.10), as well as the tool FoV constraint and occlusion avoidance (5.19) $(\boldsymbol{A}_{occ}, \boldsymbol{b}_{occ})$

$$\boldsymbol{v}_0 = \arg\min_{\boldsymbol{v}_0} \|\boldsymbol{e}_t\|^2_{\boldsymbol{Q}_t} + \|\boldsymbol{e}_{g,com}\|^2_{\boldsymbol{Q}_{g,com}} + \|\boldsymbol{I}_{t,tele}\|^2_{\boldsymbol{Q}_{t,tele}} + \|e_\phi\|^2_{Q_\phi} \tag{5.33a}$$

$$s.t. \quad \underline{\boldsymbol{q}} \leq \boldsymbol{q}(\boldsymbol{v}_0) \leq \overline{\boldsymbol{q}} \tag{5.33b}$$

$$\underline{\boldsymbol{p}} \leq \boldsymbol{p}_g(\boldsymbol{v}_0) \leq \overline{\boldsymbol{p}} \tag{5.33c}$$

$$\underline{\boldsymbol{p}} \leq \boldsymbol{p}_t(\boldsymbol{v}_0) \leq \overline{\boldsymbol{p}} \tag{5.33d}$$

$$\boldsymbol{A}_{occ}\boldsymbol{v}_0 \geq \boldsymbol{b}_{occ} \tag{5.33e}$$

The state machine switches to S_C when the regulation error is below a certain threshold and the goal area a_g is large enough, so that no further zoom in is necessary anymore.

$$t_{S_A \to S_C} : \text{ if } \|\boldsymbol{e}_t\| \leq e_{thr} \wedge a_g \geq a_{g,thr} \tag{5.34}$$

5.4.3 Conclusion state: S_C

Once the visible goal area is large enough, the camera should stop its approaching motion. Thus, we unlock the tool feature from its reference and allow it to move in the image plane while keeping the FoV constraints. As for the goal, we keep its CoM reference and also regulate its area. This allows the user to perform the final reaching motion, with a clearly visible goal, and, in absence of FoV and occlusion constraint activation, a stationary view of the scene.

The controller at this stage is then the following

$$\begin{aligned}
\boldsymbol{v}_0 = \arg\min_{\boldsymbol{v}_0} \ & \|\boldsymbol{e}_{g,com}\|^2_{\boldsymbol{Q}_{g,com}} + \|\boldsymbol{I}_{t,tele}\|^2_{\boldsymbol{Q}_{t,tele}} + \\
& + \|e_{g,area}\|^2_{Q_{g,area}} + \|e_{\phi}\|^2_{Q_{\phi}} && \text{(5.35a)} \\
s.t. \quad & \underline{\boldsymbol{q}} \leq \boldsymbol{q}(\boldsymbol{v}_0) \leq \overline{\boldsymbol{q}} && \text{(5.35b)} \\
& \underline{\boldsymbol{p}} \leq \boldsymbol{p}_g(\boldsymbol{v}_0) \leq \overline{\boldsymbol{p}} && \text{(5.35c)} \\
& \underline{\boldsymbol{p}} \leq \boldsymbol{p}_t(\boldsymbol{v}_0) \leq \overline{\boldsymbol{p}} && \text{(5.35d)} \\
& \boldsymbol{A}_{occ}\boldsymbol{v}_0 \geq \boldsymbol{b}_{occ} && \text{(5.35e)}
\end{aligned}$$

In case the user decides to abort the task by moving away from the goal and trying to leave the FoV, we go back to the setup state S_S and regulate again the tool point feature.

$$t_{S_C \to S_S} : \text{ if } \textit{FoV constraint is active} \tag{5.36}$$

5.4.4 Occlusion state: S_O

Upon activation of the occlusion constraints, the system should be properly controlled to ensure a smooth camera behavior. Although this is guaranteed by the state S_A and the occlusion constraint alone, the system should be also put in a configuration that makes the reactivation of the constraint unlikely, to avoid frequent camera positioning that may confuse the user. Indeed, while S_A guarantees no occlusion, the constraint alone does not influence that camera in a way that allows the system to explore the workspace and exploit the redundant degrees of freedom to minimize the likelihood of other occlusions, e.g. consider Fig. 5.3 or the first part of the video accompanying the experiments of Sec. 5.5 where S_O is absent. This motivates the addition of a specific state when potential occlusions are detected.

The FSM enters this state during the teleoperation, whenever an occlusion constraint holds with equality sign

$$t_{S_A \to S_O},\ t_{S_C \to S_O} : \text{ if } \textit{occlusion constraint is active} \tag{5.37}$$

The first action that we can take is to minimize the object area a_o. By reducing the forbidden area, it will be less likely for the TCP to perform a trajectory that passes through the occluding object projection, thus reducing the risk of occlusion. However, since this feature is mostly influenced by the camera motion in its z-axis, we simultaneously limit the camera velocity $v^c_{c,z}$, to avoid losing resolution by moving away. In fact, while retreating along the camera optical axis is a motion compatible with the occlusion constraint, it does not help in dealing with it, as it simply delays its occurrence, since the point-object relative positioning in the image remains the same. Indeed, in this case we want to let the camera exploit the other d.o.f. (e.g. rotation around the object), to reach a more favorable configuration.

At the same time, we also try to move the object projection away from the FoV. To ensure this behavior, we compute its CoM $\boldsymbol{p}_{o,com}$, and set its reference to a region unoccupied by the robot TCP and the goal. In absence of obstacles to the camera motion, either of the remaining free quadrants is a possible choice. Instead, if the camera motion is restricted in one way (e.g. by a table), the positioning is naturally forced to the remaining region. If the physical obstruction is more complex, the use of a planner based on the reconstructed environment becomes necessary. Clearly, the presented one is a greedy heuristic that solves the occlusion locally, disregarding the effect on objects outside the scene that might later induce a new occlusion.

Notice that by regulating at the same time $\boldsymbol{p}_t$, $\boldsymbol{p}_{g,com}$, and $\boldsymbol{p}_{o,com}$, we cannot further influence the camera motion to satisfy other requirements. In this case we should accept some error in the regulation of the TCP and goal CoM features to tackle the occlusion, and simply ask to minimize their velocities $\dot{\boldsymbol{p}}_t$, $\dot{\boldsymbol{p}}_{g,com}$.

Like in S_A and S_C, we still allow the teleoperation by tracking the master, however we remove the regulation of angle ϕ to allow more freedom to the camera. Occlusion constraints are kept in order to guarantee occlusion-free behavior also in this state. The optimization problem is thus the following one

$$\begin{aligned}
\boldsymbol{v}_0 = \arg\min_{\boldsymbol{v}_0}\ & \|\dot{\boldsymbol{p}}_t\|^2_{\boldsymbol{Q}_t} + \|\dot{\boldsymbol{p}}_{g,com}\|^2_{\boldsymbol{Q}_{g,com}} + \|\boldsymbol{I}_{t,tele}\|^2_{\boldsymbol{Q}_{t,tele}} + \\
& + \|\boldsymbol{e}_{o,com}\|^2_{\boldsymbol{Q}_{o,com}} + \|a_o\|^2_{Q_{o,area}} + \|v^c_{c,z}\|^2_{Q_{c,z}} && \text{(5.38a)} \\
s.t.\quad & \underline{\boldsymbol{q}} \le \boldsymbol{q}(\boldsymbol{v}_0) \le \overline{\boldsymbol{q}} && \text{(5.38b)} \\
& \underline{\boldsymbol{p}} \le \boldsymbol{p}_g(\boldsymbol{v}_0) \le \overline{\boldsymbol{p}} && \text{(5.38c)} \\
& \underline{\boldsymbol{p}} \le \boldsymbol{p}_t(\boldsymbol{v}_0) \le \overline{\boldsymbol{p}} && \text{(5.38d)} \\
& \boldsymbol{A}_{occ}\boldsymbol{v}_0 \ge \boldsymbol{b}_{occ} && \text{(5.38e)}
\end{aligned}$$

Table 5.1: *Occlusion-free teleoperation: normalized weight parameters and thresholds.*

Parameter	Value
$\boldsymbol{Q}_t$	$1 \cdot 40^{-2}\boldsymbol{I}_2 px^{-2}s^2$
$\boldsymbol{Q}_{g,com}$	$1 \cdot 40^{-2}\boldsymbol{I}_2 px^{-2}s^2$
$\boldsymbol{Q}_{t,tele}$	$1 \cdot 0.05\boldsymbol{I}_3 m^{-2}s^2$
$\boldsymbol{Q}_{o,com}$	$0.8 \cdot diag(640^{-2}, 480^{-2})px^{-2}$
$Q_{o,area}$	$1 \cdot 62500^{-2}px^{-4}$
$Q_{c,z}$	$0.1 \cdot 0.05^{-2}m^{-2}s^2$
e_{thr}	$5px$
$a_{g,thr}$	$14025px^2$
$a_{o,thr}$	$62500px^2$

$\boldsymbol{I}_n$ is the n-by-n identity matrix

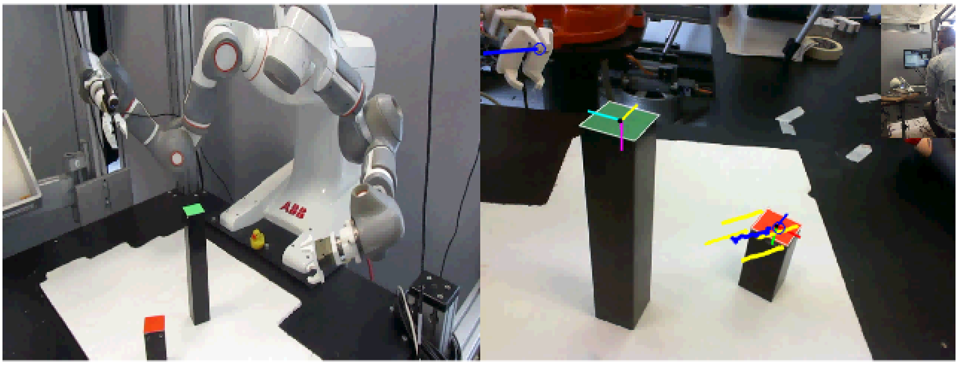
Since in this case we are regulating multiple quantities, it is necessary to accurately choose the weights $\boldsymbol{Q}_i$ to achieve the desired behavior. A possible tuning is provided in Sec. 5.5.
To determine when the camera has dealt with the occlusion, as a heuristic we check the existence of a straight path in the image plane from $\boldsymbol{p}_t$ to a point on the goal $\boldsymbol{p}_g$ and that the occluding area is small enough. From this state the system can switch either to S_A or S_C, depending on the goal area.

$$\begin{aligned} t_{S_O \to S_A} &: \text{ if } \exists \textit{ straight path from } \boldsymbol{p}_t \textit{ to } \boldsymbol{p}_g \wedge a_o < a_{o,thr} \wedge \\ &\quad \wedge \textit{ occlusion constraint is } \underline{\textit{not}} \textit{ active} \wedge a_g < a_{g,thr} \\ t_{S_O \to S_C} &: \text{ if } \exists \textit{ straight path from } \boldsymbol{p}_t \textit{ to } \boldsymbol{p}_g \wedge a_o < a_{o,thr} \wedge \\ &\quad \wedge \textit{ occlusion constraint is } \underline{\textit{not}} \textit{ active} \wedge a_g \geq a_{g,thr} \end{aligned} \tag{5.39}$$

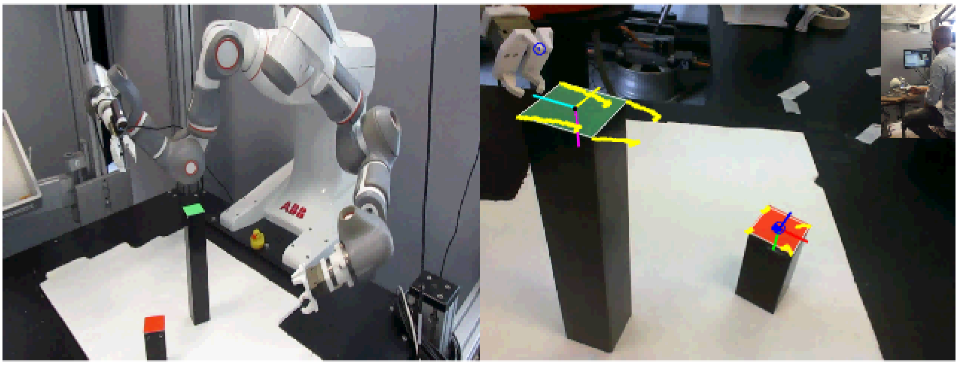
5.5 Experiments

We validated the presented approach on an experimental platform consisting of an ABB YuMi dual-arm robot with 7 d.o.f. for each arm as slave robot, and a Novint Falcon 3 d.o.f. master device. The right arm of YuMi is equipped with a $640px \times 480px$ resolution Microsoft LifeCam RGB webcam. All the image processing, as well as the camera calibration, is done with the OpenCV libraries, while for the optimization problem we employ the qpOASES solver [121].

Due to the limited number of degrees of freedom available on the master, we teleoperated only the position of the slave robot by mapping the master



(a) *Position at the end of S_S and at the start of S_A. The blue trails show the TCP and the goal CoM being brought to reference. The TCP starts from outside the FoV.*



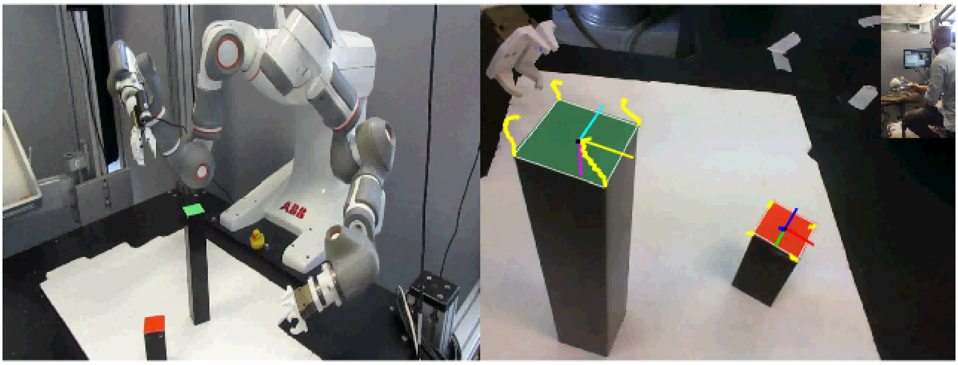
(b) *The user tries to go behind an object while in S_A. In the FoV, the object apparent motion towards the TCP activates the constraint and the transition to S_O, while the goal CoM is regulated to reference.*

Figure 5.6: *Occlusion-free teleoperation experiment*

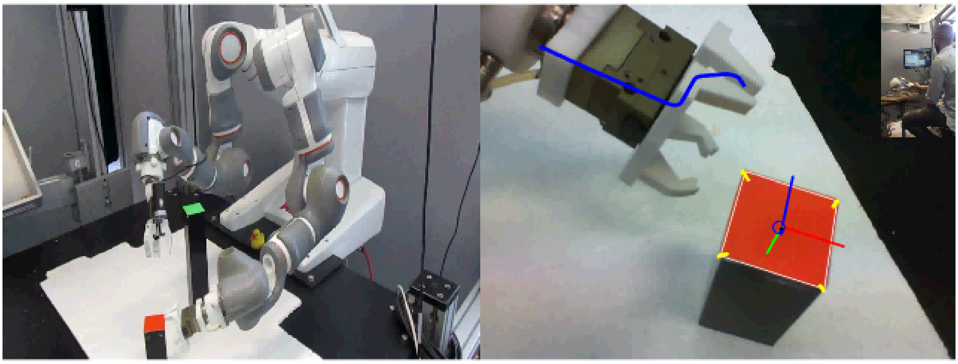
position to the slave TCP velocity expressed in the camera frame. The devices are connected to two real-time Linux PCs and communicate via TCP/IP. The controller runs at a frequency of $250Hz$, and produces the joint reference values for the ABB industrial low-level controller, while the image processing is performed at $20Hz$.

We placed two objects in the robot workspace, a small one acting as the reaching task goal, and a larger potentially occluding object. Since depth information is not provided, we assume to know the 3D model of the two objects to estimate their pose by solving the Perspective-n-Point problem from camera measurements and thus compute the interaction matrices.

In the image plane, FoV constraints were placed at distance of $32px$ and $24px$ from the true limits, while we set the TCP and the goal CoM reference respectively at a distance of $[80, 60]px$ and $[400, 300]px$ from the FoV boundary, depending on the starting configuration of the system (Fig.



(a) *During the occlusion the user can still teleoperate the robot, however the controller moves the camera to reduce the risk of occlusion and return to S_A, by minimizing the object area and moving its CoM out of the FoV. The tool and goal CoM displacement from their reference is minimal.*



(b) *When the TCP is close enough to the goal, the system enters S_C. Then the user can freely move the robot in the FoV while the camera provides a high resolution and constant view of the scene.*

Figure 5.7: *Occlusion-free teleoperation experiment*

5.5). For the goal reference area in S_C we set $a_g^* = 16500px^2$. $a_{g,thr}$ was chosen smaller than the reference to prevent the camera from pulling back after the transition to S_C. Table 5.1 shows weights and thresholds used.

Fig. 5.6 and 5.7 show the camera FoV with feature trails and the robot workspace during each phase of one of the experiments[1]. In Fig.5.6a the image features are regulated to the respective positions (S_S) and the user is ready to start the teleoperation to reach the orange goal (S_A). Notice in Fig. 5.6b how the TCP is moving while its projection is regulated in the FoV. The occlusion constraint activates due to the apparent motion of the object trying to occlude the tool, the system then switches to S_O. Fig. 5.7a shows the trail of the four vertices of the superior face of the object. The camera combines a motion around the occluding segment in the workspace

[1] Video available at `https://youtu.be/FUMXeglkqdo`.

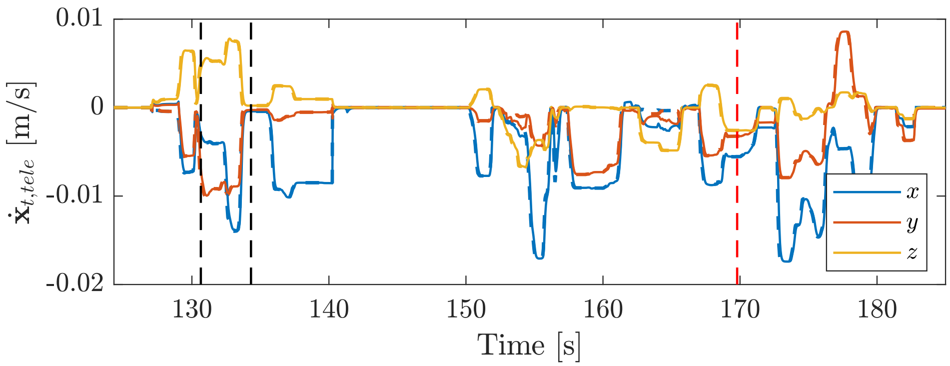


Figure 5.8: *Occlusion-free teleoperation experiment. Master (dashed) and slave (solid) velocities expressed in $\mathcal{F}_c$. The black dashes mark the start and the end of S_O, while the red ones the switch from S_A to S_C.*

and one upward to avoid the occlusion and minimize the object area while moving it towards the edge of the FoV. The designed behavior brings the camera in a more favorable position to complete the task and returns to state S_A. Finally, when the goal resolution is high enough, the final movement begins with the switch to S_C (Fig. 5.7b). Notice that the movement above the object performed by the camera is induced by the activation of state S_O, and, due to the state exit conditions, it stops when the path towards the goal becomes clear, therefore avoiding long camera readjustments.

Master and slave velocities are plotted in Fig. 5.8. We achieve kinematic coordination even during the occlusion phase, although some small mismatches are expected due to the weighing used in the optimization, compromising among the tasks when no more d.o.f. are available. The user experience with the system is smooth, since no teleoperation interruption due to the camera control is perceived.

As mentioned in Sec. 5.3.2, our formulation can implicitly deal with object motions not pre-compensated in (5.8). Thus, in another experiment we tested the robustness of the approach to unmodeled dynamics, moving the object by hand towards the TCP feature point, causing an artificial occlusion. Fig. 5.9 shows such experiment demonstrating the approach effectiveness even for perturbed and unexpected conditions.

In Fig. 5.10, 5.11, 5.12, additional experiments in other system configurations are reported, highlighting the behavior for different initializations.

In configuration A (Fig. 5.10) the occlusion constraint activates for the first time in the third figure, and the controller tries to push the occluding object in the lower right corner. The system returns to the Approach phase,

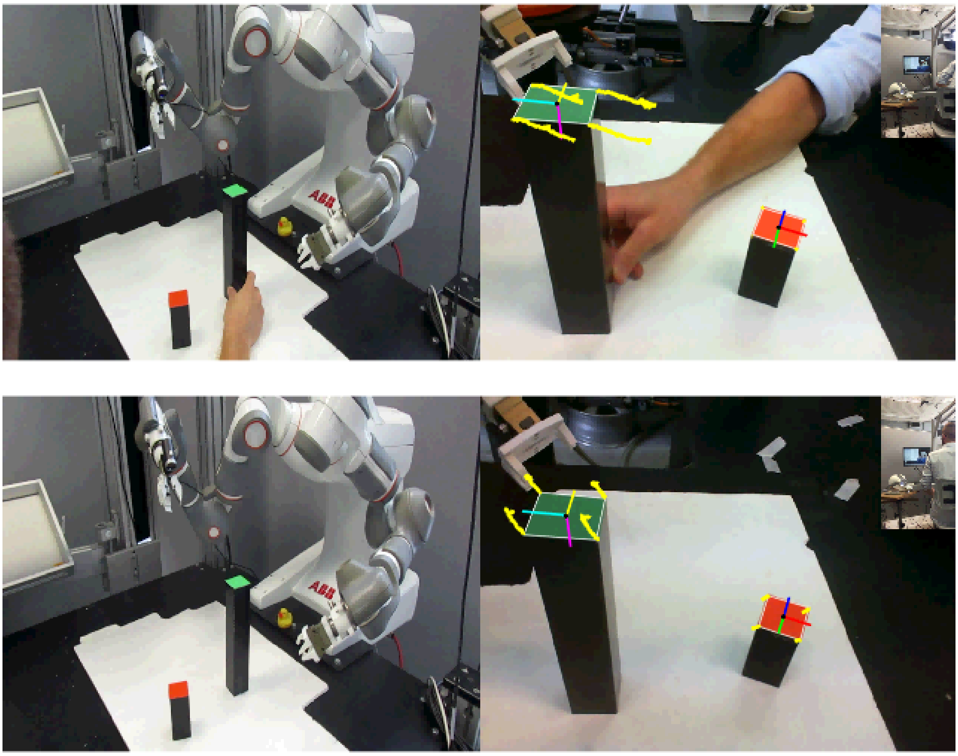


Figure 5.9: *The object is moved during S_A to induce an occlusion of the TCP. The controller is able to robustly deal with object dynamics not explicitly taken into account in the constraint derivation by moving the camera to bring the object away from the tool. The reaching task can then resume as in the nominal conditions.*

however the user tries again to bring the TCP behind the object (fourth figure), the controller solves the occlusion risk by pushing the object in the same quadrant as before, and then enters the Conclusion phase (fifth figure). Note that pushing the occluding object in the upper left corner would not have solved the Occlusion state, since the camera arm would have collided with the table. In configuration *B* (Fig. 5.11) the camera is mounted sideways. Similarly to configuration *A*, the user tries to bring the TCP behind the green object, which is however pushed in the lower right quadrant (second figure). The remaining path towards the goal is then occlusion free. Again, pushing the object to the top left would have resulted in a camera collision with the environment. Finally, in *C* (Fig. 5.12) the camera is upside down, and the object is pushed to the top right quadrant. Since the camera is closer to the object, the movement is faster to guarantee the occlusion prevention.

5.6 Closing comment

In this chapter we discussed the extension of the proposed teleoperation controller to a dual-arm system equipped with a camera. The shared control architecture with an autonomous camera and a teleoperated arm allows easy and intuitive navigation in cluttered environments with obstacles that can occlude the user's field of view. The occlusion avoidance requirement has been synthesized in terms of optimization constraints, with a high level state machine that governs the camera motion depending on constraint activation and system state. The robustness has been proven experimentally in presence of uncertain measurements and dynamic environments, as well as in a number of different system configurations.

The shared control approach presented here can be considered a reactive form of cooperation between user and system autonomy through the camera. The camera arm only reacts to the input provided by the user through the master device and has no proactive role in the completion of the task. The next chapter will deal with this aspect of shared control by trying to predict and infer the user intention to subsequently provide active assistance to the user to complete a task, possibly with full autonomy. To do so, the considered experimental domain will shift to a simpler human-robot cooperation setup, where the user is directly in contact with the robot and executing a *hands-on* task.

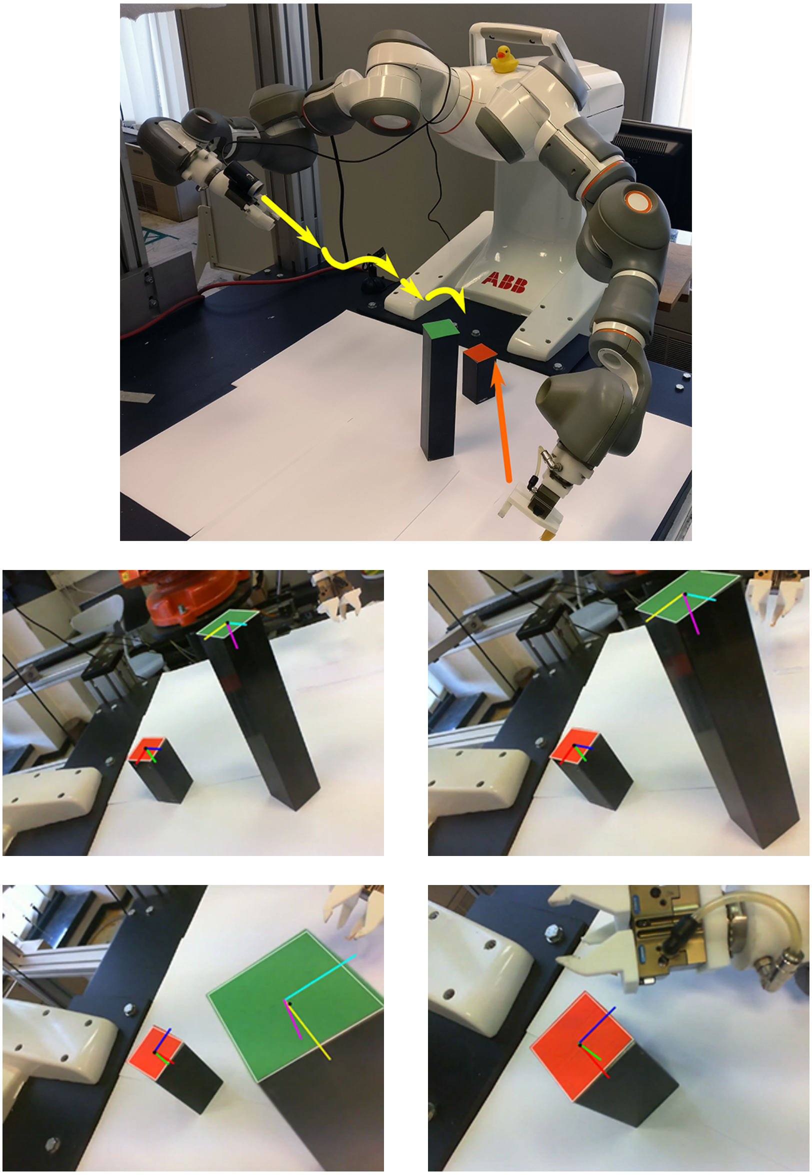


Figure 5.10: *Occlusion-free teleoperation experiment. Configuration A.*

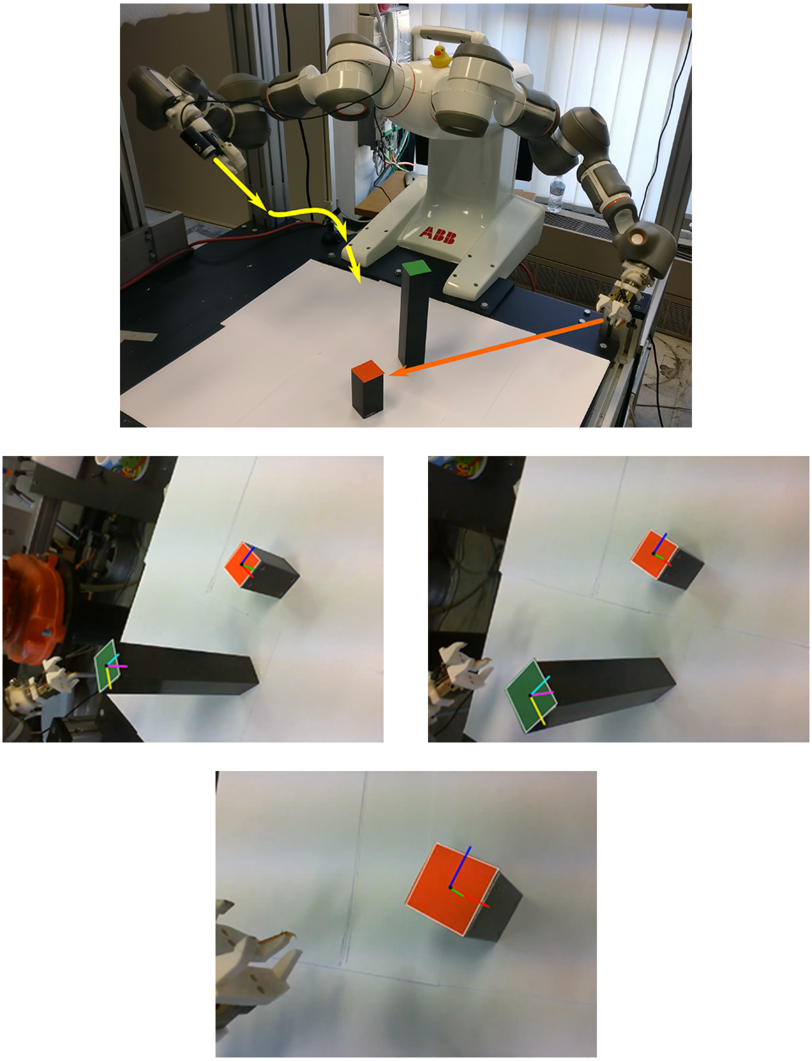


Figure 5.11: *Occlusion-free teleoperation experiment. Configuration B.*

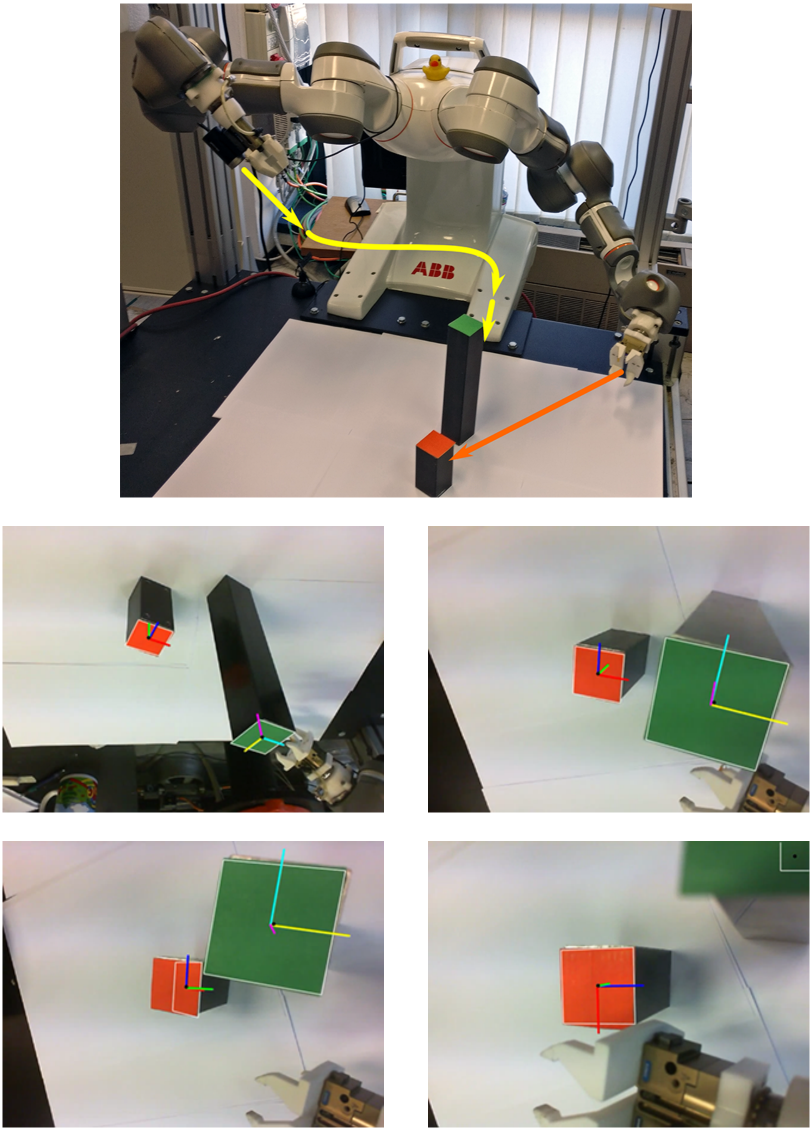


Figure 5.12: *Occlusion-free teleoperation experiment. Configuration C.*

CHAPTER 6

User intention estimation

In previous chapters we saw how a human-robot shared control architecture for teleoperation can be designed, first by means of virtual fixtures using a constrained controller (Chap. 2 and 4), and then by letting the slave be partly user-controlled and partly autonomous through the use of visual servoing (Chap. 5).

While we considered a reactive behavior of the autonomous components of the manipulators, it is meaningful to investigate if and how it is possible to predict the user plans, in order to seamlessly blend user commands with robot autonomy. In return, this may allow to better assist the operator in accomplishing his tasks, and reduce his cognitive and physical burden when using the system. In this sense, the teleoperation system can offer a better telepresence experience.

Here we present some preliminary results on user intention estimation and assistance. Instead of strictly considering teleoperation systems we give a broad overview of the approach by focusing on hands-on collaborative human-robot tasks.

We employ two Recurrent Neural Networks (RNNs) to predict the user motion in a manual guidance task (Fig. 6.1), and infer which is the most probable goal in the workspace that the user wants to reach. The neural network output is used in conjunction with a standard admittance controller to provide assistance to the user during task execution. This is achieved by helping the user drive the robot towards the intended goal, and relieve him/her from excessive physical efforts due to the stability requirements of the interface parameters. Instead of conducting an experimental campaign to collect training data, previous studies on human upper limb motion [122] are exploited to automatically generate it. This assumption is then validated on an experimental setup with a 6 d.o.f. ABB IRB140 industrial robot. Unlike previous attempts [123, 124], the approach is able to discern also when the user wants to reach a point in the workspace that is not among the predefined ones.

The chapter is arranged in the following way. Sec. 6.1 reviews some techniques present in the literature for intention estimation and assistance. In Sec. 6.2 the intention estimation scheme with RNNs and the reasoning behind training data generation are presented. In Sec. 6.3 the shared control scheme is defined, comprised of a standard admittance controller and an assistance control action obtained from the neural network outputs. Sec. 6.4 gives the experimental results.

6.1 Background

Variable impedance control has been used since the '90s to modify online the displayed damping to further reduce physical effort or allow both wide fast movements and accurate positioning during the interaction [125, 126]. In [127] the authors proposed a variable impedance controller for cooperative object lifting where the damping parameter is the result of an optimal control policy aimed at reproducing natural minimum-jerk trajectories. An estimation of human arm stiffness is used in [128] to scale the damping factor and allow a more accurate completion of a cooperative calligraphic task.

Nonetheless impedance control approaches do present some limitations when stability is concerned, due to the non-collocation of sensing and actuation produced by compliant robots, measurement filtering, operator response delays and human arm stiffening [129–131]. In [132] the authors developed an instability detection method to reactively adapt mass and damping in order to recover stability, and they also compared different adaptation techniques (i.e. varying one of the two parameters or their ratio). Similarly in [133] it was proposed to increase the apparent mass upon instability detection to avoid sluggish movements due to high damping, stability of the

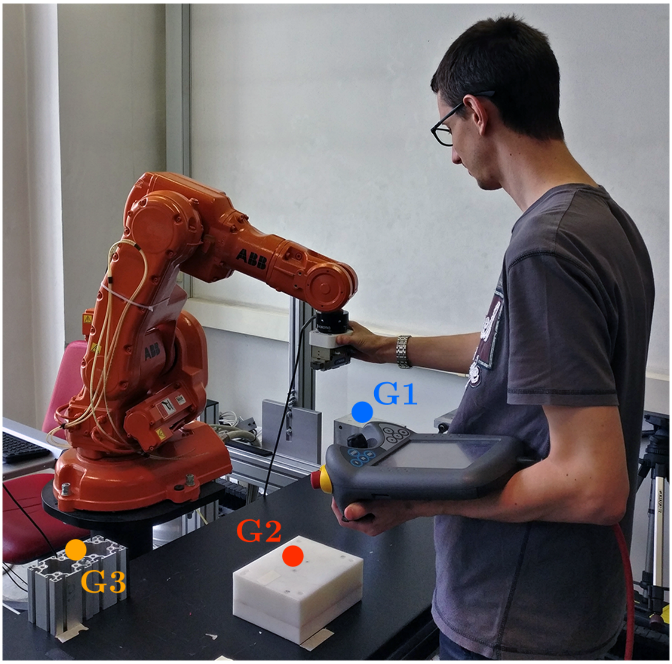


Figure 6.1: *Experimental setup for the manual guidance task. The dots indicate possible goals in the workspace.*

adaptation is assured with a tank-based passivity formulation.

Endowing the robot with proactive capabilities could improve the user interaction experience. It is however of paramount importance for the robot to correctly obtain an estimation of user intentions and act accordingly. In [134] the user force is monitored and the robot stiffness is decreased or increased whenever a sudden increment or reduction in the applied force is measured, in turn this amounts to a damping parameter proportional to the user interaction rate of change. Maeda et al. [135] estimated human motion with weighted least-squares to obtain in real-time the desired minimum jerk trajectory and subsequently feed the robot controller with human-like reference signals, making the robot actively participate in the operation. Other authors focused on mapping human intentions to robot motions by switching controllers whenever the user wants to perform a rotation or a translation of the held object, by relying on force measurements and kinematic constraints of the task [136].

In teleoperation scenarios, Dragan et al. [123] estimated the user intention probability of grasping an object among a set, using Bayesian inferencing. Given a cost function, a probability distribution is induced over the possible trajectories, and a most likely desired goal is obtained and then used to develop a shared control architecture. Similarly, in [124] the same approach is used in conjunction with Partially Observable Markov Decision Processes in shared workspace collaboration. Unfortunately the choice of the cost function is often non-trivial, although it might be learned via Inverse Optimal Control, and lacks modeling of those cases where the user actually does not intend to grasp any of the objects in the scene. Bayesian intention inference has also been used to predict the intended goal of an agent moving among static obstacles [137].

Machine learning has seen increasing popularity in teaching by demonstration scenarios, to encode prescribed behaviors in the robot motion. In [138] demonstrated motion trajectories are encoded using Dynamical Movement Primitives and locally weighted regression along with stiffness information for the autonomous execution of assembly tasks. Hersch et al. [139] presented a system for skill acquisition by combining dynamical systems with Gaussian mixture modeling and regression based on training data, for reaching and grasping tasks. The same methods have been subsequently used to learn cooperative behaviors for human-robot object carrying tasks [140]. Soft and hard virtual fixtures have been proposed by Raiola et al. [141] to guide the user towards desired locations and perform co-operative pick-and-place, by determining the most likely workspace spot based on a learned Gaussian mixture. These demonstration-based techniques have been successfully employed also in telerobotics to endow the system with shared control capabilities, relieving the user of some of the control burden, by recognizing his/her intention from data [142, 143]. Fuzzy learning has been exploited in variable admittance control to modify online the damping parameter and achieve natural minimum jerk profile movements based on the measured interaction force and velocity [144].

Among these approaches, neural networks have seen widespread use in robotics. Object handling behaviors have been generated in [145] using RNNs: the authors were able to train the network to switch between behaviors by continuously updating the input bias value depending on the object tracking error signal. Li et al. [146] made use of Radial Basis Neural Networks to compute the user equilibrium trajectory points and actively apply them as references to a robot impedance controller to achieve faster point-to-point motions while preserving stability. In [147] neural networks have been applied to the development of a virtual F/T sensor for lightweight sensor-less robots to detect user motion intent and act accordingly during human-robot physical interaction. Finally, feed-forward neural networks have been used for robot-human tool handover tasks to compute biologically inspired trajectories for the robot, and allow, on the user-side, easier understanding of the robot intentions [148].

6.2 Intention inference with neural networks

In this section we present a Recurrent Neural Network architecture that has been designed to predict user's motion and infer the user-desired goal region during a human-robot cooperative task. Specifically, we consider a hand guiding application, where the user interacts with the robot by moving its end effector or payload to areas in the workspace where some tasks have to be performed (e.g. assembly of heavy parts). We make the assumption that the user is always in contact with the robot, so that the robot tool and the user hand positions are constrained by each other.

The architecture consists of two RNNs. The first RNN objective is to provide a prediction of the future user movements given its history. The second one, instead, performs a trajectory classification, identifying the most probable goal or working area that the user wants to reach among a set of predefined ones (marked with $G1$, $G2$, $G3$ in the experimental setup of Fig. 6.1), while any other point is classified as a generic null goal $G0$. The inputs to this RNN are both the trajectory history and the prediction made by the first neural network. The outcome of the classification is then used to provide assistance to the operator.

6.2.1 Training data generation

Machine learning algorithms require the collection of a significant amount of data in order to be trained. Often, such as in the application considered in the following, data collection is costly in terms of time. We would need a certain number of users to perform multiple reaching trajectories to properly characterize the neural network, which seems impractical in real world application. To circumvent this problem we make use of previous studies on motion of the upper limbs. Viviani et al. [122] showed how reaching trajectories in humans follow the so-called two-thirds power law, concerning the relation between trajectory curvature and velocity, and a minimum-jerk profile. Given this result we argue that artificially generated trajectories that respect these rules should provide enough information to approximately model human hand motion and thus produce an accurate enough estimation of human intention. In the following we generate mini-

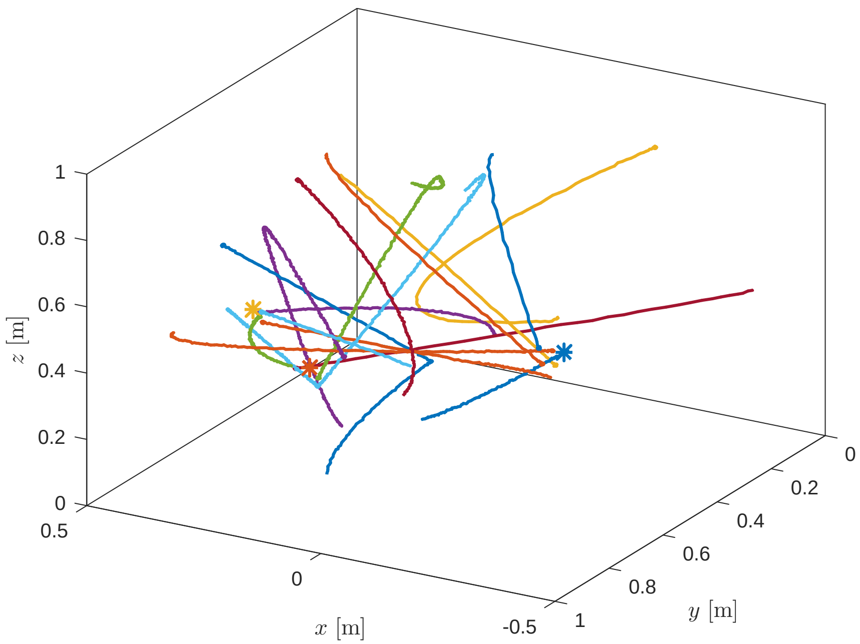


Figure 6.2: *Some of the generated human-like paths from the training set.*

mum jerk trajectories, thus minimizing the cost functional

$$C = \int_{t_0}^{t_f} \|\dddot{\boldsymbol{x}}(t)\|^2 dt \tag{6.1}$$

where t_0 and t_f are the start and final time of the trajectory, respectively, while $\dddot{\boldsymbol{x}}(t)$ is the jerk vector.

500 point to point trajectories were generated starting from random points in a cubic workspace of $1m^3$, sampled with uniform probability and ending in one of the defined goals ($G1$, $G2$, $G3$) or another random point ($G0$) with equal probability. The movement duration was sampled uniformly from a window of 1 to 10 seconds. Along with the trajectory, we defined for each goal a probability p_{Gi} that was set to 1 if that goal was the trajectory target and 0 otherwise. Since a real movement would not necessarily end exactly at the goal position, once a goal was selected we sampled the actual target location from a Gaussian distribution with mean μ_{Gi} equal to the goal position and standard deviation σ_{Gi} of $3cm$; whenever a goal $G0$ was selected within $3\sigma_{Gi}$ of one of the predefined goals, the value was discarded and re-sampled.

Additionally, to model a decision of the user to change target, we generated another 500 trajectories were the actual goal changed on-the-fly during

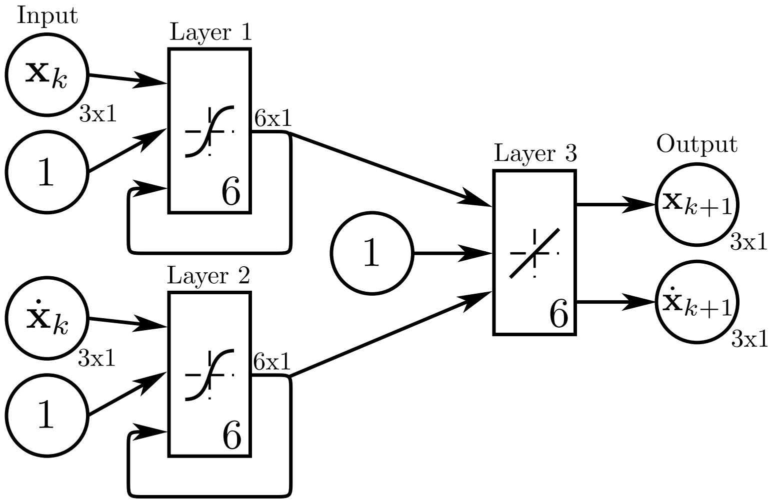


Figure 6.3: *The Simple Recurrent Network (SRN) trained for the 1-step ahead trajectory prediction.*

motion, essentially defining a trajectory with a via point. The goal probabilities were modified accordingly at the same time instant of the decision change. Furthermore, to simulate measurement noise, we added Gaussian noise with zero mean and $\sigma = 1mm$ for the position and $\sigma = 3mm/s$ for the velocity.

We repeated this training set generation procedure to acquire a second set for validation and testing purposes. In the end we obtained 2 sets of 1000 trajectories with the respective time histories of position, velocity, and goal probabilities spanning from 1 to 10 seconds in the whole workspace. Fig. 6.2 shows some of the generated paths.

6.2.2 Trajectory prediction

To capture the trajectory dynamics and take into account its time history while avoiding the definition of a large number of inputs, we trained a Simple Recurrent Network (SRN) to predict the user motion. The generated trajectories were re-sampled at a frequency of $5Hz$, while the network was trained to perform a 1-step ahead ($200ms$) prediction.

The network topology is shown in Fig. 6.3. We defined three layers of six neurons each, with the position entering the first layer and the velocity the second. The sigmoidal outputs have been combined and used as inputs to the third layer producing the prediction of both position and velocity. We trained the network using the Levenberg-Marquardt algorithm with

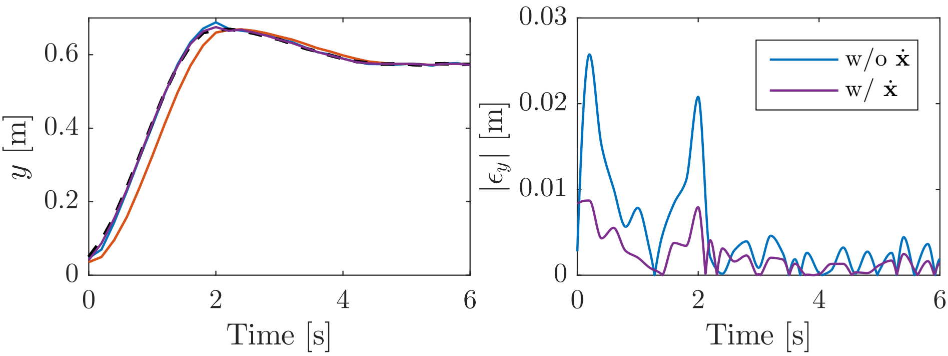


Figure 6.4: *Trajectory prediction comparison with and without velocity input for the y-axis of a sample trajectory from the validation set. Ideal prediction in black dashes.*

truncated back-propagation through time, by minimizing the mean squared error of the prediction over the N_s samples of the 1000 training trajectories

$$MSE = \min \sum_{i}^{N_s} \|\boldsymbol{\xi}_i - \hat{\boldsymbol{\xi}}_i\|^2 \tag{6.2}$$

where $\boldsymbol{\xi}_i$ is the normalized ideal target prediction, while $\hat{\boldsymbol{\xi}}_i$ the network prediction output. The second set of trajectories was equally split between validation and testing, and the training was stopped when no improvement of the validation MSE occurred for 6 consecutive epochs. We obtained a validation MSE value of $2.9 \cdot 10^{-6}$ for our best training attempt. Fig. 6.8 in the middle shows the position prediction compared with the target values for one of the trajectories.

Since the network should be able to generalize and achieve an accurate prediction based simply on the position data, the inclusion of the velocity as an input to the SRN may seem unnecessary. It was observed, however, that the use of velocity information provides on average slightly better results. Indeed, we tested the proposed architecture against a similar one with just the position as input and a hidden layer of 12 neurons. Over 100 training attempts an average of $3.1 \cdot 10^{-6}$ validation MSE was achieved for the network in Fig. 6.3, and $1.2 \cdot 10^{-5}$ for the network without velocity input.

Fig. 6.4 shows a comparison of the prediction given by the two networks for a sample trajectory. The absolute prediction error is consistently less for the proposed architecture, since the network does not have to implicitly reconstruct the velocity to perform the prediction, which is instead provided similarly to a feed-forward action, this is more critical when higher order derivatives become important (e.g. at $2s$ where motion stops). Moreover, in

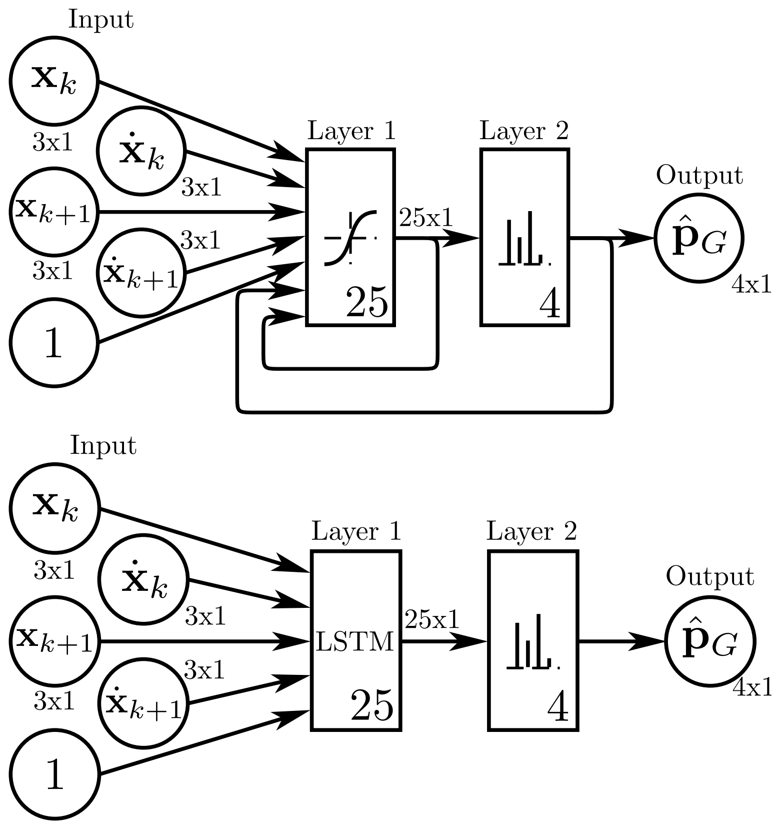


Figure 6.5: *The SRN (top) and LSTM (bottom) RNN architectures used to classify the trajectories and provide goal probability. Both inputs and outputs of the first network are used as inputs to these ones.*

the considered application knowledge of the velocity from joint measurements is a reasonable assumption.

6.2.3 Trajectory classification

A second RNN was trained to model a probability distribution over the trajectories and provide an estimation of the most probable goals up to that time instant. A SRN was first designed, characterized by a hidden layer of 25 neurons and an output layer with as many neurons as the number of goals with a softmax output. The inputs are the current position and velocity as well as the predicted state from the previous network. Both the output probability and the hidden layer state were fed back as inputs to mimic Bayesian inferencing (Fig. 6.5 top). In this framework, the posterior probability of goal Gi given the current trajectory is proportional to the one

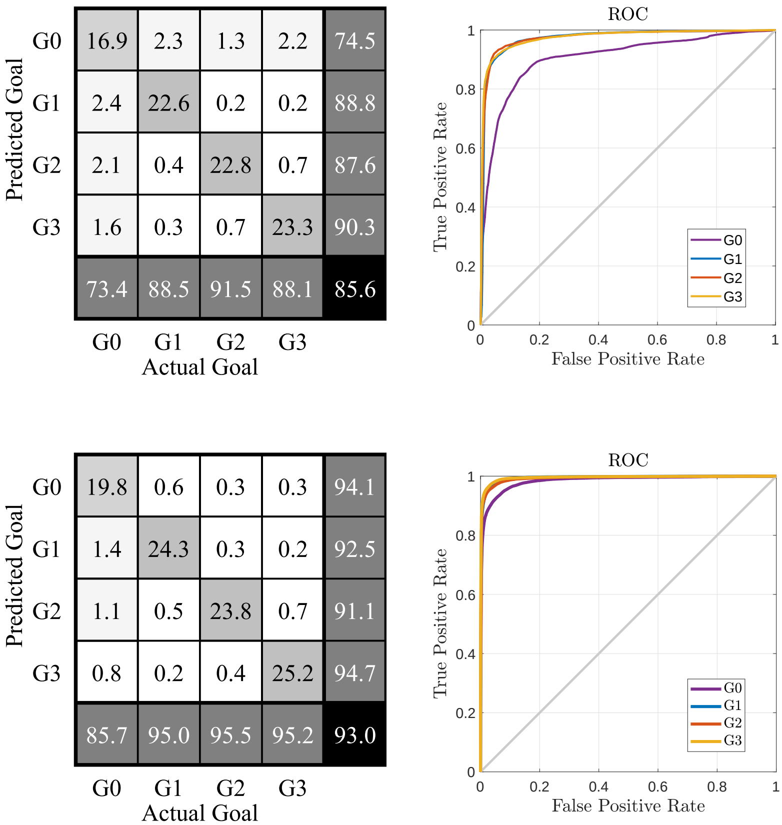


Figure 6.6: *Confusion matrix and ROC plot of the SRN (top) and LSTM network (bottom) for goal probability estimation. The last column of the confusion matrix shows the precision of each goal prediction, while the last row shows for each goal the number of correct identifications (recall), the last element gives the overall accuracy of the predictions.*

of observing the trajectory given Gi, times the prior probability of Gi being the intended goal

$$p(Gi|\boldsymbol{x}_{0:t}) \propto p(\boldsymbol{x}_t|Gi, \boldsymbol{x}_{0:t-1})p(Gi|\boldsymbol{x}_{0:t-1}) \tag{6.3}$$

While in the Bayesian framework Markov assumptions are often needed to simplify the problem [149], with a Recurrent Neural Network we can take into account the trajectory history. The network will adapt by weighing differently the various time instants, as it is clear that older samples may have no impact on the current user intention. Moreover we can also consider those cases when the user does not want to reach any of the default goals

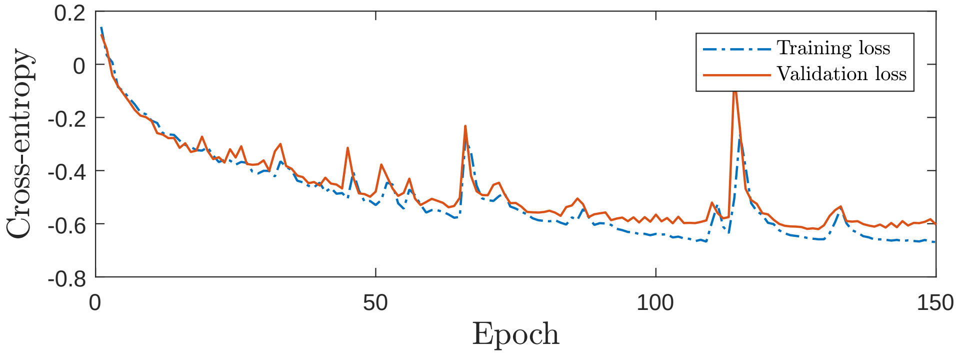


Figure 6.7: *Log-scale loss function history for the LSTM network during training.*

($G0$).

Since the problem is inherently a classification one, we trained the network by minimizing the cross entropy of the goal probability distributions

$$\mathcal{H}_c = \min - \sum_{i}^{N_s} \sum_{j}^{N_g} p_{i,Gj} \log(\hat{p}_{i,Gj}) \tag{6.4}$$

where $p_{i,Gj}$ is the probability for sample i to belong to class Gj as defined during data generation, and $\hat{p}_{i,Gj}$ is the network actual output goal probability. As before one set was used for training while the other for validation, and the network was trained by using Stochastic Gradient Descent with momentum in batches of 100 trajectories. The top of Fig. 6.6 presents the confusion matrix and the receiver operating characteristic for validation data. While we can correctly classify about 88% to 91% of the predefined goals ($G1$, $G1$, $G3$), it is more difficult to discern whether the user wants to reach another point in the workspace ($G0$).

In order to assess if a longer memory could improve the classification accuracy for trajectories heading towards null goals ($G0$), we substituted the recurrent layer with a Long Short-Term Memory (LSTM) one and removed the output feedback (Fig. 6.5 bottom). Fig. 6.6 shows the resulting confusion and ROC plots for the validation set, which exhibit a clear increase of the overall accuracy up to 93%, especially thanks to a 20% reduction of false positives and a 12% increase in sensitivity when predicting $G0$. Fig. 6.7 shows the loss function history during training.

To compare the performance obtained with the SRN and LSTM and evaluate the impact of the additional trajectory prediction input, we retrained each network 100 times, with and without this input. The results are shown in Table 6.1. We observed an increase of 3.63% to 5.92% in the

Table 6.1: *Classification accuracy comparison w/ and w/o prediction input for 100 trained networks each. Confidence intervals and p-values are obtained by performing a two-sample Welch's t-test* on the accuracy increase.*

	$\mu \pm \sigma$	C.I. at 95%	p-value
	Simple RNN		
Acc. w/o pred.	$75.35 \pm .05\%$	$[3.63 \quad 5.92]\ \%$	$4.3 \cdot 10^{-14}$
Acc. w/ pred.	$80.13 \pm .03\%$		
	LSTM RNN		
Acc. w/o pred.	$91.18 \pm .01\%$	$[0.26 \quad 1.02]\ \%$	$1.1 \cdot 10^{-3}$
Acc. w/ pred.	$91.82 \pm .01\%$		
	Simple RNN vs. LSTM RNN		
SRN acc. w/o	$75.35 \pm .05\%$	$[14.8 \quad 16.8]\ \%$	$7.2 \cdot 10^{-59}$
LSTM acc. w/o	$91.18 \pm .01\%$		
SRN acc. w/	$80.13 \pm .03\%$	$[11.0 \quad 12.4]\ \%$	$9.2 \cdot 10^{-6}$
LSTM acc. w/	$91.82 \pm .01\%$		

*For large sample sizes the test is robust against non-normality.

number of correctly classified trajectories when using the prediction with the SRN. This could be due to an anticipatory effect given by the prediction, allowing the network to understand that the system is about to reach a state where an intention change is likely to occur, thus starting to update the goal probabilities in advance. Compared to the SRN, the LSTM achieves an accuracy increase of over 10%, although the effect of the prediction input is reduced, as it is probably overshadowed by the whole past history that is now considered by the LSTM. Still, in 40.4% of the cases, the LSTM with prediction input detected a change of intended goal earlier than the one without, and at the same instant in 36.3%, with an overall 0.29 to 0.59-step average earlier detection (C.I. at 95%, p-value $= 1.9 \cdot 10^{-8}$). A multi-step prediction input might possibly further improve the result.

Fig. 6.8 at the top shows a trajectory from the test set going from the starting position at rest to $G2$ and then to $G1$. At the bottom, we can see the estimated probability for each goal compared with the true one. Note that there is a slight estimation inertia due to the LSTM memory, which, however, also avoids updating the probabilities too fast, as noise could produce abrupt changes of goal estimation despite unaltered user intention. It is also worth noticing how $\hat{p}_{G2}$ starts decreasing and $\hat{p}_{G1}$ increasing, one step before the actual change of intention, due to the trajectory prediction used in the classifier.

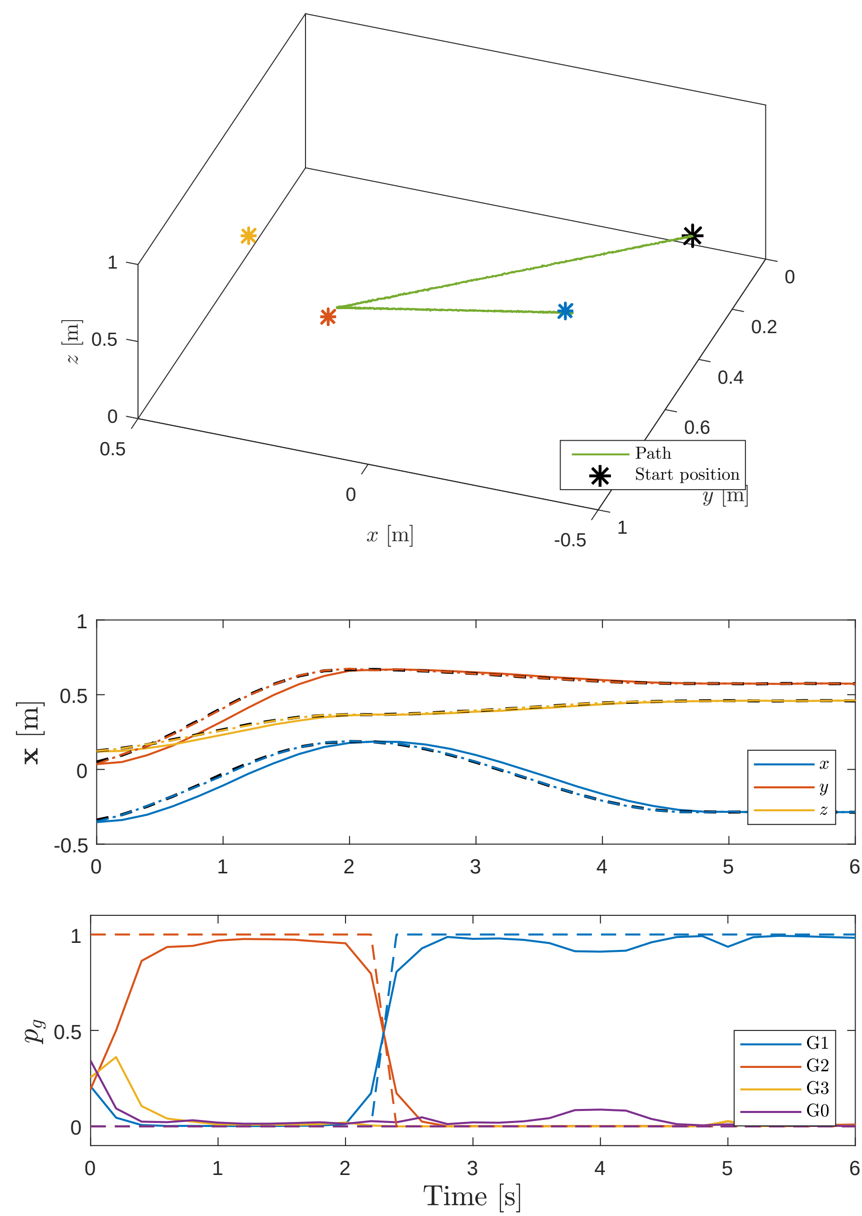


Figure 6.8: *Intention inference result for a sample trajectory. Top: sample trajectory going from a random point in the workspace to G2 and then to G1. Middle: neural network prediction (dashed colored) compared to the target signal (dashed black) for the position of the trajectory. Bottom: the network goal probability output history (solid) compared to the actual intended goal (dashed).*

6.3 Shared control scheme

The intention estimation obtained in Sec. 6.2 is used in conjunction with a classical admittance controller to provide assistance to the user in completing the desired reaching task. The objective is to define a controller that allows the robot and the human to share control of the robot end-effector.

Most methods in the literature [123] rely on defining an arbitration function α that depends on the estimation reliability and basically averages the control inputs from the human and the robot controller

$$u = \alpha u_a + (1 - \alpha) u_h \tag{6.5}$$

where u is the actual robot control input, while u_h and u_a are the user and the robot desired control, computed from the intention estimation algorithm, with $0 \leq \alpha \leq 1$. Although blending has been shown to be sub-optimal in some cases [150], due to the hands-on nature of the application, in the following we deem more appropriate to adopt a similar approach by letting the user be able to always intervene in full capacity, with the robot aiding in the completion of desired motions

$$u = \alpha u_a + u_h \tag{6.6}$$

6.3.1 Admittance controller

We assume the robot to be position-controlled with shared control at velocity level ($\boldsymbol{u} = \dot{\boldsymbol{x}}^r$), and equipped with a F/T sensor to measure the forces applied by the human on the robot. We will also consider only translations, while the robot orientation will be kept constant.

An admittance dynamic relation is defined by the following equation

$$\boldsymbol{M}\ddot{\boldsymbol{x}}_h^r + \boldsymbol{D}\dot{\boldsymbol{x}}_h^r = \boldsymbol{F}_h \tag{6.7}$$

where $\boldsymbol{M} > 0$ and $\boldsymbol{D} > 0$ are the admittance controller desired mass and damping matrices, $\boldsymbol{F}_h$ the vector of forces exerted by the human on the robot, and $\dot{\boldsymbol{x}}_h^r$ ($\boldsymbol{u}_h$) the resulting reference robot Cartesian velocity. At each control cycle k we can compute the desired accelerations $\ddot{\boldsymbol{x}}_{h,k}^r$ and velocities $\dot{\boldsymbol{x}}_{h,k}^r$ by integration.

Based on the results of Sec. 6.2, assume that we have computed a suitable velocity for the assistance control $\alpha \boldsymbol{u}_a = \dot{\boldsymbol{x}}_a^r$ (see Sec. 6.3.2). Therefore we can obtain the overall reference velocity and acceleration by summing them up (6.6)

$$\ddot{\boldsymbol{x}}^r = \ddot{\boldsymbol{x}}_h^r + \ddot{\boldsymbol{x}}_a^r \tag{6.8a}$$

$$\dot{\boldsymbol{x}}^r = \dot{\boldsymbol{x}}_h^r + \dot{\boldsymbol{x}}_a^r \tag{6.8b}$$

Since we want the robot to follow this reference in the Cartesian space, for simplicity we first compute the joint acceleration reference with a second order inverse kinematics and then integrate twice to get the reference joint position

$$\ddot{\boldsymbol{q}} = \boldsymbol{J}^{-1}\left(\begin{bmatrix}\ddot{\boldsymbol{x}}^r \\ 0\end{bmatrix} + \begin{bmatrix}\boldsymbol{K}_d(\dot{\boldsymbol{x}}^r - \dot{\boldsymbol{x}}) \\ -\boldsymbol{K}_d\dot{\boldsymbol{\phi}}\end{bmatrix} + \begin{bmatrix}0 \\ \boldsymbol{K}_p(\boldsymbol{\phi}^r - \boldsymbol{\phi})\end{bmatrix} - \dot{\boldsymbol{J}}\dot{\boldsymbol{q}}\right)$$

where $\boldsymbol{J}$ is the robot Jacobian, $\boldsymbol{K}_d$ and $\boldsymbol{K}_p$ are positive definite matrices, and $\boldsymbol{\phi}^r$ and $\boldsymbol{\phi}$ are the robot reference and current orientation. The resulting robot motion will track the Cartesian velocity reference while keeping constant orientation.

6.3.2 Assistance controller

To aid the user during cooperative tasks, at each iteration we compute an assistance control $\dot{\boldsymbol{x}}^r_a$. The idea is to exploit the information given by the neural network trajectory classifier, and generate a control action that is coherent with the one issued by the user (6.8), in such a way that less effort is required and the task can also be completed autonomously if the user stops providing any input.

For instance, let us consider an assistance of the following form

$$\dot{\boldsymbol{x}}^r_a = \alpha \dot{x}^{max*}_{a,k} \boldsymbol{n}_{\bar{G}i} \tag{6.9}$$

where α is a scaling factor, $\dot{x}^{max*}_{a,k}$ the assistance maximum intensity, and $\boldsymbol{n}_{\bar{G}i}$ a unit vector describing the assistance direction.

Given the network estimated $\hat{p}_{Gi}$, if the most probable goal is one of the predefined ones (e.g. $G1$—3) and its probability is above a certain threshold, we can infer what the user desired goal is. Thus, we take the maximum absolute value of the assistance control as follows

$$\dot{x}^{max}_{a,k} = \begin{cases} \max\{\dot{x}^{max}_{a,k-1},\ \dot{\boldsymbol{x}}^{r\,T}_{h,k}\boldsymbol{n}_{\bar{G}i}\} & \boldsymbol{F}^T_h \boldsymbol{n}_{\bar{G}i} \geq 0 \\ \max\{0,\ \dot{\boldsymbol{x}}^{r\,T}_{h,k}\boldsymbol{n}_{\bar{G}i}\} & \textit{otherwise} \end{cases} \tag{6.10}$$

where $\boldsymbol{n}_{\bar{G}i}$ is the unit vector going from the current position to the goal with the highest probability $\bar{G}i$. In this way we assist with an end effector velocity at most equal to the one given by the admittance filter in the goal direction. If the user applies a force that tries to pull the robot away from the predicted goal, the assistance is reduced and eventually vanishes, so that he/she always has control of the robot motion. Near the goal location $\boldsymbol{x}_{\bar{G}i}$ the maximum assistance is smoothened to zero depending on the distance from the goal, with the following logistic function

$$\dot{x}_{a,k}^{max*} = \frac{2\dot{x}_{a,k}^{max}}{1 + e^{\left(-\frac{2k}{\dot{x}_{a,k}^{max}}\|\boldsymbol{x}_{\bar{G}i} - \boldsymbol{x}\|\right)}} - \dot{x}_{a,k}^{max} \tag{6.11}$$

where k is the slope of the function in $\boldsymbol{x} = \boldsymbol{x}_{\bar{G}i}$. Note that this is equivalent to the definition of a potential field with minimum in the estimated goal and a saturation given by (6.10), more complex behaviors can be encoded for example with Dynamical Systems [139]. $\dot{x}_{a,k}^{max}$ is also reset to zero when the estimated goal $\bar{G}i$ changes.

If the null goal $G0$ is the most probable or the highest predefined goal probability is below the threshold, we simply take the RNN trajectory prediction $\boldsymbol{x}_{k+1}$ as a temporary goal and compute the maximum assistance as follows

$$\dot{x}_{a,k}^{max} = \begin{cases} \max\{\dot{x}_{a,k-1}^{max},\ \dot{\boldsymbol{x}}_{h,k}^{r\,T}\boldsymbol{n}_{\boldsymbol{x}_{k+1}}\} & \boldsymbol{F}_h^T\boldsymbol{n}_{\boldsymbol{x}_{k+1}} > 0 \\ \max\{0,\ \dot{\boldsymbol{x}}_{h,k}^{r\,T}\boldsymbol{n}_{\boldsymbol{x}_{k+1}}\} & \textit{otherwise} \end{cases} \tag{6.12}$$

where $\boldsymbol{n}_{\boldsymbol{x}_{k+1}}$ is now the unit vector from the current position to the predicted one $\boldsymbol{x}_{k+1}$.

Notice that, while in (6.10), if $\boldsymbol{F}_h^T\boldsymbol{n}_{\bar{G}i} = 0$ we can still apply an assistance control to reach the goal autonomously (e.g. the user has let go of the robot and $\boldsymbol{F}_h = 0$), in (6.12) if $\boldsymbol{F}_h^T\boldsymbol{n}_{\boldsymbol{x}_{k+1}} = 0$ the assistance quickly decays to zero and stops the robot if the user is not interacting anymore.

Other authors showed how the assistance aggressiveness influences the user preferences depending on the prediction correctness [123]. For this reason, the arbitration parameter α is selected as a function of the reliability of the goal estimation. To do so we use the entropy as a measure of uncertainty: a high entropy will suggest a less accurate estimate and low values of α, resulting in a less aggressive assistance, and vice versa for low entropy. The arbitration α is defined as

$$\alpha = 1 - \frac{\mathcal{H}}{\mathcal{H}_{max}} = 1 + \frac{\sum_i^{N_g} \hat{p}_{Gi} log(\hat{p}_{Gi})}{log(N_g)} \tag{6.13}$$

where $\mathcal{H}/\mathcal{H}_{max}$ is the normalized entropy.

Based on (6.10), (6.11), (6.12) and (6.13) we can compute the final assistance control action via (6.9). $\dot{\boldsymbol{x}}_a^r$ is then used together with the output of the admittance controller in (6.8), while $\ddot{\boldsymbol{x}}_a^r$ can be computed numerically to provide a feed-forward action. Fig. 6.9 shows the overall control scheme.

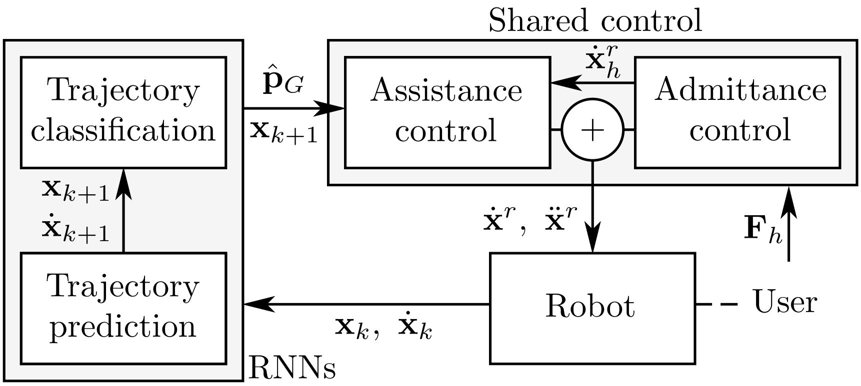


Figure 6.9: *Block diagram of the overall control scheme.*

6.4 Experiments

The proposed approach was tested on a system consisting of a 6 d.o.f. ABB IRB140 industrial robot equipped with a Robotiq FT300 F/T sensor. Three goal regions were set up in the robot workspace as shown in Fig. 6.1.

Fourteen subjects, with variable previous experience with robots, were enrolled to participate in the experimental validation of the system. The users were asked to perform a manual guidance task twice, moving the robot end effector towards the goals in a given order (Fig. 6.10), with and without the assistance control action. Time was given to gain confidence with the setup, but they were not informed in which occasion the assistance was enabled.

The admittance controller parameters were chosen conservatively to guarantee stability robustness during the interaction for all users. Lower values exhibited the onset of loss of passivity (oscillations) when some users stiffened their grip. We set

$$\boldsymbol{M} = 10\boldsymbol{I}_3, \qquad \boldsymbol{D} = 50\boldsymbol{I}_3$$

where $\boldsymbol{I}_3$ is the 3-by-3 identity matrix. The controller has been implemented with a cycle time $T_s = 4ms$.

Fig. 6.10 shows the path executed by one of the users during an experiment. The subjects performed a motion towards each goal in succession, starting from $G1$ and returning to a null goal $G0$ after each movement. During a final movement from $G0$ to $G1$ the user changed on-the-fly the intended goal and turned towards $G2$.

Fig. 6.11a shows the velocity of the robot during the experiment with assistance. The dashed lines indicate the robot velocity without the contri-

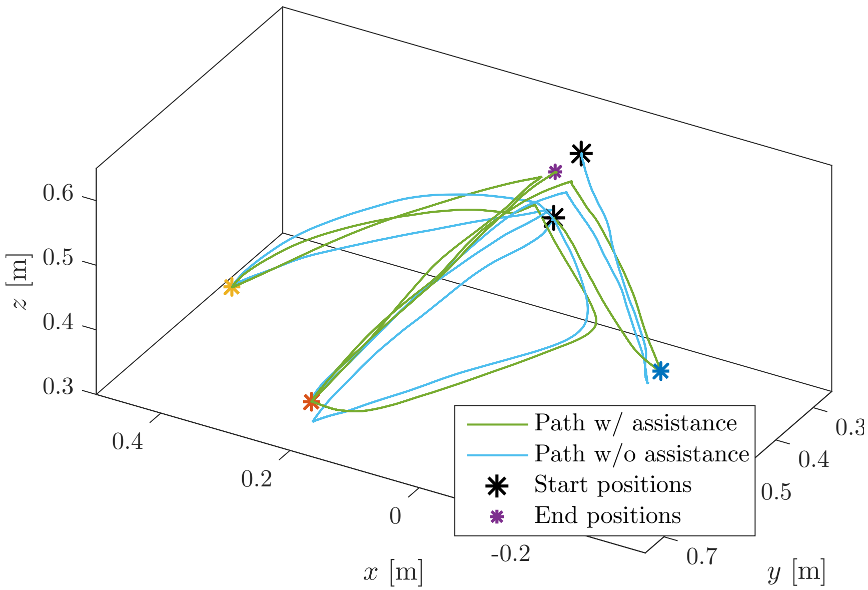
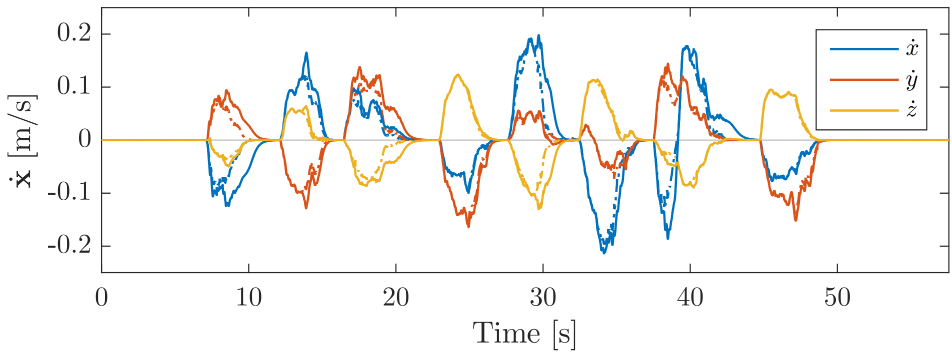


Figure 6.10: *Paths executed by one of the participants during the experiment, with and without assistance.*

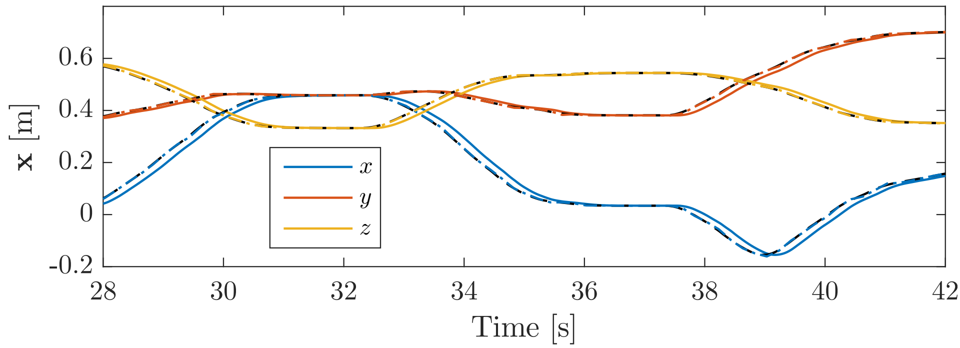
bution given by the assistance, thus considering only the user input via the admittance relation (6.7). It is clear that $\dot{\boldsymbol{x}}_a^r$ facilitates the reaching motion, as the user needs to apply less force to produce the same end-effector velocity. The correct position prediction in Fig. 6.11b confirms the assumption made in Sec. 6.2.1 about the minimum jerk nature of human arm motion. Note that while the controller works at a frequency of $250Hz$, the neural network performs a $200ms$ ahead prediction at the reduced rate of $5Hz$.

We compared our goal estimation approach with the Bayesian one used in [123], with the sum of squared velocities as cost function. Fig. 6.12 illustrates the estimated goal probabilities for the two algorithms. Although pinpointing the instant where the user changes objective is rather difficult compared to the synthetic data where accuracy could be computed exactly, the proposed approach correctly performs a sharp prediction before the goal is reached, as confirmed also by the arbitration parameter in the bottom figure. Furthermore, we can also understand when the user does not want to reach any of the defined goals, while it is often unclear with the standard Bayesian formulation.

Nonetheless, four of the participants stated that they perceived no major differences in robot behavior between the approach with assistance and the



(a) *Robot velocity during the manual guidance task with assistance. The dashed lines indicate the velocity due to the user input $\dot{\boldsymbol{x}}_h^r$ without the contribution given by the assistance.*



(b) *Robot end effector trajectory during the task with assistance, and trajectory prediction generated by the RNN.*

Figure 6.11: *Data for one of the subjects during an experiment with assistance.*

one without, while four others noticed a clear guidance. Two remarked that the difference was clearer in configurations near the goals, which is compatible with Fig. 6.11a, and with the fact that the arbitration parameter α inevitably grows larger when getting close to a goal, due to the greater confidence of the neural network (Fig. 6.12). Only two subjects reported no difference in effort, while the others described a reduction of fatigue with the assistance.

To quantitatively evaluate the difference between the proposed approach and the standard one without assistance, we monitored the energy used by the subjects while interacting with the system during the experiments. Note that in deceleration the energy extracted by the user is negligible since the damper dissipates most of it. By employing the assistance control action we should expect a decrease in expended energy to perform the same task, resulting in lower user fatigue levels.

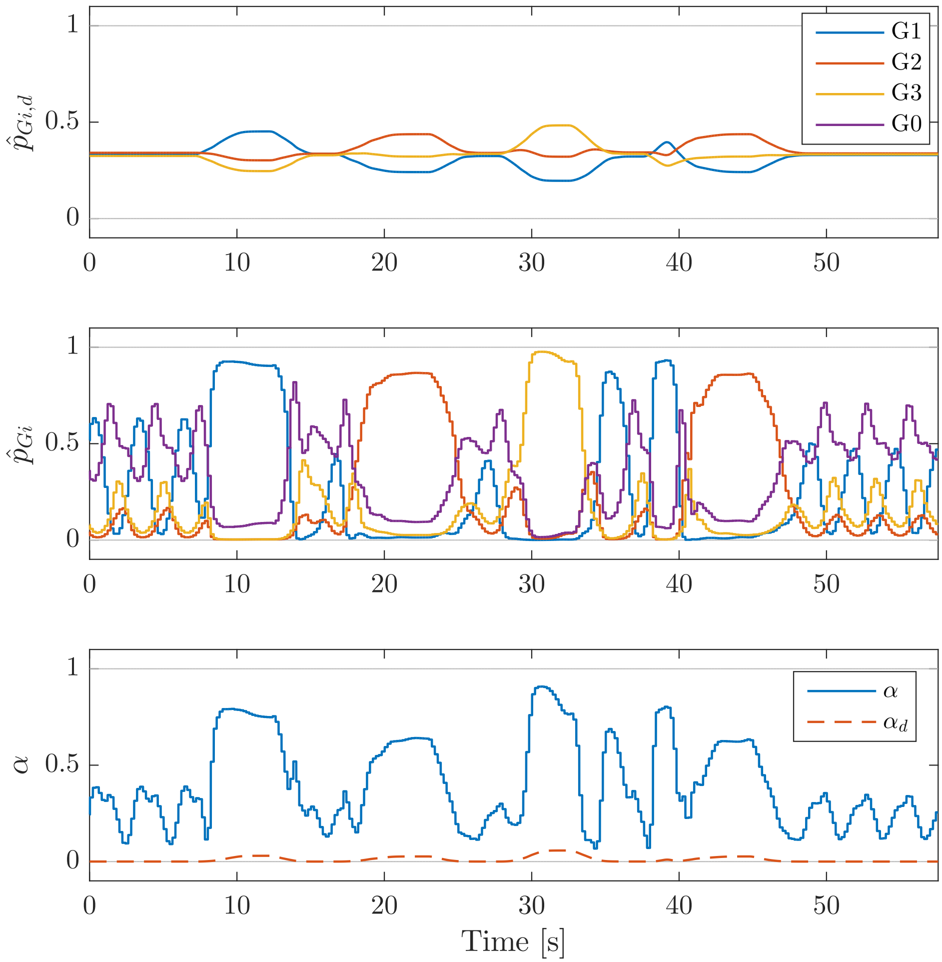


Figure 6.12: *Data for one of the subjects during an experiment with assistance. Top: estimated goal probability with the Bayesian approach used in [123]. Middle: estimated goal probability with the proposed algorithm. Bottom: arbitration parameter computed according to* (6.13) *for the two approaches.*

Fig. 6.13 shows the energy spent by the users normalized over the path length for the two cases, with a decrease in effort required for the proposed approach. The same figure on the right, shows a box plot of the percentage energy reduction, confirming that the robot is indeed helping the user during the task. By performing a Wilcoxon signed-rank test we obtain a

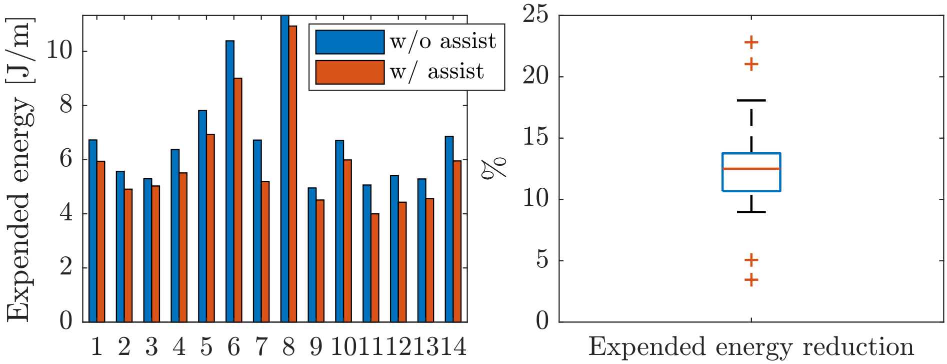


Figure 6.13: *Expended energy comparison with and without assistance. Left: energy per unit path length expended by the subjects during the experiments with and without the proposed approach. Right: box plot of the energy reduction percentage. The whiskers show the minimum and maximum values of the data, while the '+' represent the outliers.*

p-value $= 1.2 \cdot 10^{-4}$ with a confidence interval at 95% on the mean energy reduction of $[9.3;\ 15.3]\%$, rejecting the null hypothesis of no change in expended energy.

The reason why some subjects did not report any difference is probably due to the fact that they did not have time to interact with the system continuously for prolonged periods. Notice that the highest reduction of 22.8% was obtained by the user with more expertise with robotic systems, while the lowest of 3.4% by the one with no previous experience. The results agree with the ones obtainable with variable impedance techniques, however note that, with the proposed approach, once the robot understands the human intention it is able to autonomously complete the movement without any user intervention (i.e. $u_h = 0$).

6.5 Closing comment

In this chapter we discussed a possible integration of the shared control approach proposed in Chap. 5. While in the previous chapter the autonomous behavior was a reactive response to user inputs, here we tried to infer the user intention via neural networks and generate an assistance control that actively helps the user in completing a task. Although these results have been validated only preliminarily on a scenario involving direct cooperation, encouraging findings have been obtained in regards to prediction and inference of human intention, as well as reduction of user fatigue through active assistance, which may be easily extended to teleoperation systems.

CHAPTER 7

Conclusions

THE purpose of this work is to provide a contribution to the research area of robotic teleoperation. Although in many aspects robotics has been striving for fully autonomous systems, the demand for user presence, safety, and complex decision making of remote control applications in highly complex environments and scenarios, make the human-robot control loop of telerobotics a flourishing area of study.

This research provides contributions to a broad range of teleoperation aspects by presenting a complete control framework, spanning from the local low level controller of master and slave devices, to the high level human-robot shared control through visual and virtual force cues, passing through the stability analysis of the closed-loop delayed remote interaction. Moreover, preliminary results on human intention inference promise to further increase the synergy between operator and robotic platform.

In the first part (Chap. 2), we preliminarily focused on systems characterized by impedance-type master devices and position-controlled slaves. We first presented a teleoperation controller based on hierarchical optimization to obtain kinematic correspondence between master and slave devices, that has the advantage of allowing the simultaneous definition of hard and soft virtual fixtures. Differently from previous approaches, the controller structure is exploited, so that it naturally provides a method for haptic rendering and virtual kinesthetic feedback, based on a physical interpretation of the optimization problem. Control parameters tuning is based on their physical meaning, simplifying the procedure especially for the cost functions relative weightings.

Since the position control loop assumption reveals to be limiting when considering interaction with real environments and a desired dynamics profile, in Chap. 3 a novel robust controller is proposed for constrained redundant robots. The properties of sliding mode control theory are exploited to obtain a robust operational space formulation of impedance control. Sliding manifold projection, permits to completely reject disturbances and uncertainties for the whole kinematic chain, while preserving task hierarchy. The model predictive control component simultaneously satisfies inequality constraints and enforces task priority, minimizing the deteriorating effects of actuation delay and filtering via sliding manifold prediction. The controller properties have been analyzed through simulations, and their effectiveness experimentally validated on a real industrial platform.

In Chap. 4 these results are applied to achieve desired master and slave impedances, and combined with the optimization-based controller for virtual fixtures and task priority enforcement. The in-depth stability analysis considers the manipulators impedance tuning and highlights robust absolute stability conditions, that guarantee system reliability in presence of variable communication delay. Compared to time-domain passivity techniques the controller provides slightly better results, although under different assumptions on user and environment dynamics.

Visual feedback has been integrated in Chap. 5 on a dual-arm teleoperated slave. The shared control of the platform allows robust occlusion avoidance in cluttered environments by means of visual servoing, while maintaining the same operative complexity for the user. The designed constraints and finite state machine ensure a natural interaction between the user and the system, as well as occlusion-free task execution. The robust formulation of the optimization problem guarantees performance also in presence of measurement noise and unmodeled object dynamics.

A proactive shared control has been considered to further enhance human-robot cooperation, although presently only for hands-on applications. In the last chapter (Chap. 6) a neural network architecture for human intention estimation and assistance has been discussed. The exploitation of previous studies on human motion let us avoid conducting a data collection campaign and generate required training data offline. The output of the trained network provides the likelihood that the user wants to reach a particular goal, based on the trajectory history and prediction. This information is used to integrate an additional controller that assists the user in completing the task. Experimental data shows a reduction of the energy spent by the users to perform the operation, reducing fatigue with respect to standard admittance control techniques.

7.1 Further developments

Presently, a drawback in the proposed teleoperation architecture lies in the necessity of force/torque sensors both on slave and master devices, in order to discriminate between external forces and residual robot uncertain dynamics. Although estimation procedures exist [84], they are heavily affected by model accuracy. Finding suitable sliding mode control gains may also require some trial and error tuning, even though adaptive algorithms could be employed [63, 64]. Although experiments on a real platform showed the approach effectiveness of the local controllers, and of the overall teleoperation scheme, some obstacles regarding static friction compensation still have to be addressed to achieve accurate impedance control [85].

While the analysis of the delayed closed-loop system provided sufficient stability conditions to obtain a stable interaction, it did so by considering passive terminations of the two-port. Less stringent results may be obtained by removing these assumptions as proposed in [151]. This relaxation is already embedded in passivity-based techniques, however their implementation may impact negatively on transparency even when there is no reason to do so, depending on their tuning (Sec. 4.6). Moreover, future work may consider variable impedance parameters to improve the system performance and their impact on the user experience. Another possible line of research may investigate the estimation of the user end-point impedance and their grip strength, which plays a crucial role in limiting the choice of impedance parameters (Sec.2.4.1). In [152], model matching and EMG signals were used to open the control loop and avoid the stabilization problem, but at the price of completely sacrificing kinesthetic feedback. Nonlinear model based state observers may help overcome these difficulties.

In the visual servoing integration, we assumed to know the objects 3D models to compute the respective interaction matrices, however an accurate model is not necessary. Indeed, even in presence of unknown environment, one can generate a convex volume containing the object (e.g. by fitting a superquadric on a point cloud, as done in [153]), and then discretize it by selecting a set of connected points on its surface that will become point features in the image plane. To avoid constraint activation when the point of interest is closer to the camera than the potentially occluding object, a sensor providing depth information is necessary, if no a priori model is available. Future endeavors should also consider multiple objects and efficient solvers due to the growing computational time required by the optimization [115]. In the experiments only occlusions related to the TCP were considered for simplicity, although the whole approach can be applied also to the goal. Additional validation may include concurrent TCP and goal occlusions, even though some configurations may require the temporary relaxation of one of the constraints in order to complete the task. Moreover, simple reaching motions were considered, without particularly complex high level decision making on the user part, which may lead one to think that full automation could be possible. To better highlight the necessity of human intervention in such scenarios, further improvements may include grasping as well as manipulation or assembly of a priori unknown objects.

The shared control properties of the proposed approach have been analyzed in terms of reactive control to user inputs, where the system autonomous behavior is strictly a response to user decision making. These results have been extended to proactive autonomy in Chap. 6, where the robot tries to understand the user intention and anticipate him/her by providing an assistance that can ultimately also conclude the task.

Nonetheless, this has been tested only on a hands-on cooperative scenario, further work should try to apply the presented results to a full teleoperation system, possibly including feature information from the camera. Another aspect that should be considered is that an a priori definition of possible goal regions is required. Although the addition of new goals does not require new data collection and training can be immediately done offline, the estimation network should be modified to cope with goals that can be defined and modified in real-time and extend the approach to more challenging scenarios. More complex assistance controllers may also be considered, e.g. by employing Gaussian mixture models to teach the robot human-like assistance behaviors [154].

List of Figures

List of Tables

Bibliography

[1] R. C. Goertz and W. M. Thompson, "Electronically controlled manipulator," *Nucleonics (US) Ceased publication*, vol. 12, 1954.

[2] A. M. Okamura, "Haptic feedback in robot-assisted minimally invasive surgery," *Current opinion in urology*, vol. 19, no. 1, p. 102, 2009.

[3] D. Reintsema, K. Landzettel, and G. Hirzinger, "Dlr's advanced telerobotic concepts and experiments for on-orbit servicing," in *Advances in telerobotics*, pp. 323–345, Springer, 2007.

[4] A. Kron, G. Schmidt, B. Petzold, M. I. Zäh, P. Hinterseer, and E. Steinbach, "Disposal of explosive ordnances by use of a bimanual haptic telepresence system," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 2, pp. 1968–1973, IEEE, 2004.

[5] Shibuya Hoppman, "Shibuya hoppman webpage," 2012–2016.

[6] "Smartsurg." `http://www.smartsurg-project.eu/`, 2017.

[7] F. Abi-Farraj, N. Pedemonte, and P. R. Giordano, "A visual-based shared control architecture for remote telemanipulation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'16*, 2016.

[8] F. Negrello, A. Settimi, D. Caporale, G. Lentini, M. Poggiani, D. Kanoulas, L. Muratore, E. Luberto, G. Santaera, L. Ciarleglio, *et al.*, "Humanoids at work," *IEEE Robotics & Automation Magazine*, 2018.

[9] O. Khatib, X. Yeh, G. Brantner, B. Soe, B. Kim, S. Ganguly, H. Stuart, S. Wang, M. Cutkosky, A. Edsinger, *et al.*, "Ocean one: A robotic avatar for oceanic discovery," *IEEE Robotics & Automation Magazine*, vol. 23, no. 4, pp. 20–29, 2016.

[10] G. Hirzinger, B. Brunner, J. Dietrich, and J. Heindl, "Rotex-the first remotely controlled robot in space," in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pp. 2604–2611, IEEE, 1994.

[11] J. Artigas, R. Balachandran, C. Riecke, M. Stelzer, B. Weber, J.-H. Ryu, and A. Albu-Schaeffer, "Kontur-2: force-feedback teleoperation from the international space station," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pp. 1166–1173, IEEE, 2016.

[12] P. Schmaus, D. Leidner, T. Krüger, A. Schiele, B. Pleintinger, R. Bayer, and N. Y. Lii, "Preliminary insights from the meteron supvis justin space-robotics experiment," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3836–3843, 2018.

[13] S. Hirche, M. Ferre, J. Barrio, C. Melchiorri, and M. Buss, "Bilateral control architectures for telerobotics," in *Advances in Telerobotics*, pp. 163–176, Springer, 2007.

[14] C. Melchiorri, *Robot Teleoperation*, pp. 1–14. London: Springer London, 2013.

[15] P. Arcara and C. Melchiorri, "Control schemes for teleoperation with time delay: A comparative study," *Robotics and Autonomous systems*, vol. 38, no. 1, pp. 49–64, 2002.

[16] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, pp. 2035–2057, 2006.

[17] S. Hirche and M. Buss, "Human-oriented control for haptic teleoperation," *Proceedings of the IEEE*, 2012.

[18] R. J. Anderson and M. W. Spong, "Bilateral control of teleoperators with time delay," *IEEE Transactions on Automatic control*, vol. 34, no. 5, pp. 494–501, 1989.

[19] G. Niemeyer and J.-J. Slotine, "Stable adaptive teleoperation," *IEEE Journal of oceanic engineering*, vol. 16, no. 1, pp. 152–162, 1991.

[20] J. E. Colgate and G. G. Schenkel, "Passivity of a class of sampled-data systems: Application to haptic interfaces," *Journal of robotic systems*, vol. 14, no. 1, pp. 37–47, 1997.

[21] R. J. Adams and B. Hannaford, "Stable haptic interaction with virtual environments," *Robotics and Automation, IEEE Transactions on*, vol. 15, no. 3, pp. 465–474, 1999.

[22] D. A. Lawrence, "Stability and transparency in bilateral teleoperation," *IEEE transactions on robotics and automation*, vol. 9, no. 5, pp. 624–637, 1993.

[23] S. Y. Lee, K. Y. Lee, S. H. Lee, J. W. Kim, and C. S. Han, "Human-robot cooperation control for installing heavy construction materials," *Autonomous Robots*, vol. 22, no. 3, p. 305, 2007.

[24] S. Tafazoli, S. E. Salcudean, K. Hashtrudi-Zaad, and P. D. Lawrence, "Impedance control of a teleoperated excavator," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 3, pp. 355–367, 2002.

[25] K. Bussmann, A. Dietrich, and C. Ott, "Whole-body impedance control for a planetary rover with robotic arm: Theory, control design, and experimental validation," in *Robotics and Automation (ICRA), 2018 IEEE International Conference on*, IEEE, 2018.

[26] E. Beretta, E. De Momi, F. Rodriguez y Baena, and G. Ferrigno, "Adaptive hands-on control for reaching and targeting tasks in surgery," *International Journal of Advanced Robotic Systems*, vol. 12, no. 5, p. 50, 2015.

[27] A. Peer, S. Hirche, C. Weber, I. Krause, M. Buss, S. Miossec, P. Evrard, O. Stasse, E. S. Neo, A. Kheddar, *et al.*, "Intercontinental multimodal tele-cooperation using a humanoid robot," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pp. 405–411, IEEE, 2008.

[28] P. Chatelain, A. Krupa, and N. Navab, "Optimization of ultrasound image quality via visual servoing," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 5997–6002, IEEE, 2015.

[29] J. M. Ahola, J. Koskinen, T. Seppälä, and T. Heikkilä, "Development of impedance control for human/robot interactive handling of heavy parts and loads," in *Proceedings of the International Conference on ASME 2015 International Design Engineering Technical Conference & Computers and Information in Engineering Conference (IDETC/CIE), IEEE, Boston, MA*, p. 8, 2015.

[30] L. B. Rosenberg, "Virtual fixtures: Perceptual tools for telerobotic manipulation," in *Virtual Reality Annual International Symposium, 1993., 1993 IEEE*, pp. 76–82, IEEE, 1993.

[31] J. J. Abbott, P. Marayong, and A. M. Okamura, "Haptic virtual fixtures for robot-assisted manipulation," in *Robotics research*, pp. 49–64, Springer, 2007.

[32] J. J. Abbott and A. M. Okamura, "Virtual fixture architectures for telemanipulation," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 2, pp. 2798–2805, IEEE, 2003.

[33] A. Bettini, P. Marayong, S. Lang, A. M. Okamura, and G. D. Hager, "Vision-assisted control for manipulation using virtual fixtures," *Robotics, IEEE Transactions on*, vol. 20, no. 6, pp. 953–966, 2004.

[34] O. Kanoun, F. Lamiraux, and P.-B. Wieber, "Kinematic control of redundant manipulators: generalizing the task priority framework to inequality tasks," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 785–792, 2011.

[35] A. M. Zanchettin and P. Rocco, "Reactive motion planning and control for compliant and constraint-based task execution," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 2748–2753, IEEE, 2015.

[36] J. Funda, R. H. Taylor, B. Eldridge, S. Gomory, and K. G. Gruben, "Constrained cartesian motion control for teleoperated surgical robots," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 3, pp. 453–465, 1996.

[37] J. Sheng and M. Spong, "Model predictive control for bilateral teleoperation systems with time delays," in *Electrical and Computer Engineering, 2004. Canadian Conference on*, vol. 4, pp. 1877–1880, IEEE, 2004.

[38] M. Li, A. Kapoor, and R. H. Taylor, "A constrained optimization approach to virtual fixtures," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1408–1413, IEEE, 2005.

[39] M. Li, M. Ishii, and R. H. Taylor, "Spatial motion constraints using virtual fixtures generated by anatomy," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 4–19, 2007.

[40] T. Xia, A. Kapoor, P. Kazanzides, and R. Taylor, "A constrained optimization approach to virtual fixtures for multi-robot collaborative teleoperation," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 639–644, IEEE, 2011.

[41] S. A. Bowyer, B. L. Davies, and F. R. y Baena, "Active constraints/virtual fixtures: A survey," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 138–157, 2014.

[42] A. Kapoor, M. Li, and R. H. Taylor, "Constrained control for surgical assistant robots.," in *ICRA*, pp. 231–236, 2006.

[43] B. Siciliano and J.-J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Advanced Robotics, 1991.'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, pp. 1211–1216, IEEE, 1991.

[44] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 1006–1028, 2014.

[45] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[46] D. Lee and M. W. Spong, "Passive bilateral teleoperation with constant time delay," *IEEE transactions on robotics*, vol. 22, no. 2, pp. 269–281, 2006.

[47] A. M. Zanchettin and P. Rocco, "Robust constraint-based control of robot manipulators: an application to a visual aided grasping task," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'16*, IEEE, 2016.

[48] L. Dominjon, A. Lecuyer, J.-M. Burkhardt, G. Andrade-Barroso, and S. Richir, "The" bubble" technique: interacting with large virtual environments using haptic devices with limited workspace," in *Eurohaptics Conference, 2005 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. World Haptics 2005. First Joint*, pp. 639–640, IEEE, 2005.

[49] N. Hogan, "Impedance control: An approach to manipulation: Part ii - implementation," *Journal of dynamic systems, measurement, and control*, vol. 107, no. 1, pp. 8–16, 1985.

[50] M. S. Erden and A. Billard, "End-point impedance measurements at human hand during interactive manual welding with robot," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 126–133, IEEE, 2014.

[51] M. P. Polverini, A. M. Zanchettin, S. Castello, and P. Rocco, "Sensorless and constraint based peg-in-hole task execution with a dual-arm robot," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pp. 415–420, IEEE, 2016.

[52] D. Surdilovic, H. Zhao, G. Schreck, and J. Krueger, "Advanced methods for small batch robotic machining of hard materials," in *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on*, pp. 1–6, VDE, 2012.

[53] F. Ficuciello, A. Romano, L. Villani, and B. Siciliano, "Cartesian impedance control of redundant manipulators for human-robot co-manipulation," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pp. 2120–2125, IEEE, 2014.

[54] E. Colgate and N. Hogan, "An analysis of contact instability in terms of passive physical equivalents," in *1989 IEEE International Conference on Robotics and Automation*, pp. 404–409, IEEE, 1989.

[55] S. P. Buerger and N. Hogan, "Complementary stability and loop shaping for improved human–robot interaction," *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 232–244, 2007.

[56] C. Abdallah, D. M. Dawson, P. Dorato, and M. Jamshidi, "Survey of robust control for rigid robots," *IEEE Control Systems*, vol. 11, no. 2, pp. 24–30, 1991.

[57] M. W. Spong, "On the robust control of robot manipulators," *IEEE Transactions on Automatic Control*, vol. 37, no. 11, pp. 1782–1786, 1992.

[58] V. I. Utkin, *Sliding modes in control and optimization.* Springer Science & Business Media, 2013.

[59] Z. Lu, S. Kawamura, and A. A. Goldenberg, "Sliding mode impedance control and its application to grinding tasks," in *Intelligent Robots and Systems' 91.'Intelligence for Mechanical Systems, Proceedings IROS'91. IEEE/RSJ International Workshop on*, pp. 350–355, IEEE, 1991.

[60] B. Xiao, S. Yin, and O. Kaynak, "Tracking control of robotic manipulators with uncertain kinematics and dynamics," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 10, pp. 6439–6449, 2016.

[61] G. P. Incremona, G. De Felici, A. Ferrara, and E. Bassi, "A supervisory sliding mode control approach for cooperative robotic system of systems," *IEEE Systems Journal*, vol. 9, no. 1, pp. 263–272, 2015.

[62] G. Bartolini, A. Ferrara, A. Levant, and E. Usai, "On second order sliding mode controllers," in *Variable structure systems, sliding mode and nonlinear control*, pp. 329–350, Springer, 1999.

[63] V. I. Utkin and A. S. Poznyak, "Adaptive sliding mode control with application to super-twist algorithm: Equivalent control method," *Automatica*, vol. 49, no. 1, pp. 39–47, 2013.

[64] A. Goel and A. Swarup, "Mimo uncertain nonlinear system control via adaptive high-order super twisting sliding mode and its application to robotic manipulator," *Journal of Control, Automation and Electrical Systems*, vol. 28, no. 1, pp. 36–49, 2017.

[65] A. Ferrara and G. P. Incremona, "Design of an integral suboptimal second-order sliding mode controller for the robust motion control of robot manipulators," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 6, pp. 2316–2325, 2015.

[66] D. Brambilla, L. M. Capisani, A. Ferrara, and P. Pisu, "Fault detection for robot manipulators via second-order sliding modes," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 11, pp. 3954–3963, 2008.

[67] S. Habibi, "Sliding mode control of a hydraulic industrial robot," *Journal of Dynamic Systems, Measurement, and Control*, vol. 121, no. 2, pp. 312–318, 1999.

[68] Q. Ha, Q. Nguyen, D. Rye, and H. Durrant-Whyte, "Impedance control of a hydraulically actuated robotic excavator," *Automation in Construction*, vol. 9, no. 5-6, pp. 421–435, 2000.

[69] H. Alwi and C. Edwards, "Fault tolerant control using sliding modes with on-line control allocation," *Automatica*, vol. 44, no. 7, pp. 1859–1866, 2008.

[70] A. Dietrich, C. Ott, and A. Albu-Schaffer, "Multi-objective compliance control of redundant manipulators: Hierarchy, control, and stability," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 3043–3050, IEEE, 2013.

[71] J. Lee, H. Dallali, M. Jin, D. G. Caldwell, and N. G. Tsagarakis, "Robust and adaptive dynamic controller for fully-actuated robots in operational space under uncertainties," *Autonomous Robots*, pp. 1–18, 2018.

[72] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal, "Operational space control: A theoretical and empirical comparison," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 737–757, 2008.

[73] W. Garcia-Gabin, D. Zambrano, and E. F. Camacho, "Sliding mode predictive control of a solar air conditioning plant," *Control Engineering Practice*, vol. 17, no. 6, pp. 652–663, 2009.

[74] K. R. Muske, H. Ashrafiuon, and M. Nikkhah, "A predictive and sliding mode cascade controller," in *American Control Conference, 2007. ACC'07*, pp. 4540–4545, IEEE, 2007.

[75] G. P. Incremona, A. Ferrara, and L. Magni, "Mpc for robot manipulators with integral sliding modes generation," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 3, pp. 1299–1307, 2017.

[76] H. C. Cho and J. H. Park, "Stable bilateral teleoperation under a time delay using a robust impedance control," *Mechatronics*, vol. 15, no. 5, pp. 611–625, 2005.

[77] C. Edwards and S. Spurgeon, *Sliding mode control: theory and applications*. Crc Press, 1998.

[78] I. Spangelo, J. R. Sagli, and O. Egeland, "Bounds on the largest singular value of the manipulator jacobian," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 1, pp. 93–96, 1993.

[79] V. Utkin and J. Shi, "Integral sliding mode in systems operating under uncertainty conditions," in *Decision and Control, 1996., Proceedings of the 35th IEEE Conference on*, vol. 4, pp. 4591–4596, IEEE, 1996.

[80] A. Levant, "Sliding order and sliding accuracy in sliding mode control," *International Journal of Control*, vol. 58, no. 6, pp. 1247–1263, 1993.

[81] J. A. Moreno and M. Osorio, "Strict lyapunov functions for the super-twisting algorithm," *IEEE transactions on automatic control*, vol. 57, no. 4, pp. 1035–1040, 2012.

[82] A. De Luca, G. Oriolo, and B. Siciliano, “Robot redundancy resolution at the acceleration level,” *Laboratory Robotics and Automation*, vol. 4, pp. 97–97, 1992.

[83] D. Nicolis, A. M. Zanchettin, and P. Rocco, “A hierarchical optimization approach to robot teleoperation and virtual fixtures rendering,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5672–5679, 2017. 20th IFAC World Congress.

[84] S. Haddadin, A. De Luca, and A. Albu-Schäffer, “Robot collisions: A survey on detection, isolation, and identification,” *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1292–1312, 2017.

[85] G. Bartolini, A. Pisano, E. Punta, and E. Usai, “A survey of applications of second-order sliding mode control to mechanical systems,” *International Journal of control*, vol. 76, no. 9-10, pp. 875–892, 2003.

[86] J. Rollett, “Stability and power-gain invariants of linear twoports,” *IRE Transactions on Circuit Theory*, vol. 9, no. 1, pp. 29–32, 1962.

[87] T. A. Varkonyi, I. J. Rudas, P. Pausits, and T. Haidegger, “Survey on the control of time delay teleoperation systems,” in *Intelligent Engineering Systems (INES), 2014 18th International Conference on*, pp. 89–94, IEEE, 2014.

[88] J.-H. Ryu, J. Artigas, and C. Preusche, “A passive bilateral control scheme for a teleoperator with time-varying communication delay,” *Mechatronics*, vol. 20, no. 7, pp. 812–823, 2010.

[89] M. Franken, S. Stramigioli, S. Misra, C. Secchi, and A. Macchelli, “Bilateral telemanipulation with time delays: A two-layer approach combining passivity and transparency,” *IEEE transactions on robotics*, vol. 27, no. 4, pp. 741–756, 2011.

[90] K. Hashtrudi-Zaad and S. E. Salcudean, “Analysis of control architectures for teleoperation systems with impedance/admittance master and slave manipulators,” *The International Journal of Robotics Research*, vol. 20, no. 6, pp. 419–445, 2001.

[91] K. Hashtrudi-Zaad and S. E. Salcudean, “Transparency in time-delayed systems and the effect of local force feedback for transparent teleoperation,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 108–114, 2002.

[92] B. Willaert, B. Corteville, D. Reynaerts, H. Van Brussel, and E. B. Vander Poorten, “Transparency trade-offs for a 3-channel controller revealed by the bounded environment passivity method,” in *Advances in Computer-Human Interactions, 2010. ACHI'10. Third International Conference on*, pp. 66–72, IEEE, 2010.

[93] P. Buttolo, P. Braathen, and B. Hannaford, “Sliding control of force reflecting teleoperation: Preliminary studies,” *Presence: Teleoperators & Virtual Environments*, vol. 3, no. 2, pp. 158–172, 1994.

[94] L. G. García-Valdovinos, V. Parra-Vega, and M. A. Arteaga, “Observer-based sliding mode impedance control of bilateral teleoperation under constant unknown time delay,” *Robotics and Autonomous Systems*, vol. 55, no. 8, pp. 609–617, 2007.

[95] L. G. García-Valdovinos, A. López-Segovia, H. Santacruz-Reyes, T. Salgado-Jiménez, and L. A. García-Zarco, “Impedance-based sliding mode control for nonlinear teleoperators under constant time delay,” in *Control Applications (CCA), 2016 IEEE Conference on*, pp. 1537–1543, IEEE, 2016.

[96] A. Jafari and J.-H. Ryu, “Transparency improved sliding-mode control design for bilateral teleoperation systems by using virtual manipulator concept,” *IFAC Proceedings Volumes*, vol. 46, no. 29, pp. 21–26, 2013.

[97] A. Hace and K. Jezernik, “Bilateral teleoperation by sliding mode control and reaction force observer,” in *Industrial Electronics (ISIE), 2010 IEEE International Symposium on*, pp. 1809–1816, IEEE, 2010.

[98] D. T. McRuer and H. R. Jex, “A review of quasi-linear pilot models,” *IEEE transactions on human factors in electronics*, no. 3, pp. 231–249, 1967.

[99] M. Selvaggio, G. Notomista, F. Chen, B. Gao, F. Trapani, and D. Caldwell, “Enhancing bilateral teleoperation using camera-based online virtual fixtures generation,” in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pp. 1483–1488, IEEE, 2016.

[100] J. Kofman, X. Wu, T. J. Luu, and S. Verma, “Teleoperation of a robot manipulator using a vision-based human-robot interface,” *IEEE transactions on industrial electronics*, vol. 52, no. 5, pp. 1206–1219, 2005.

[101] P. Wunsch and G. Hirzinger, “Real-time visual tracking of 3d objects with dynamic handling of occlusion,” in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 4, pp. 2868–2873, IEEE, 1997.

[102] A. De Luca, G. Oriolo, and P. Robuffo Giordano, “Feature depth observation for image-based visual servoing: Theory and experiments,” *The International Journal of Robotics Research*, vol. 27, no. 10, pp. 1093–1116, 2008.

[103] R. Fleurmond and V. Cadenat, “Handling visual features losses during a coordinated vision-based task with a dual-arm robotic system,” in *Control Conference (ECC), 2016 European*, pp. 684–689, IEEE, 2016.

[104] M. Kazemi, K. Gupta, and M. Mehrandezh, “Path-planning for visual servoing: A review and issues,” *Visual Servoing via Advanced Numerical Methods*, pp. 189–207, 2010.

[105] Y. Mezouar and F. Chaumette, “Avoiding self-occlusions and preserving visibility by path planning in the image,” *Robotics and Autonomous Systems*, vol. 41, no. 2, pp. 77–87, 2002.

[106] O. Kermorgant and F. Chaumette, “Dealing with constraints in sensor-based robot control,” *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 244–257, 2014.

[107] Y. Koren and J. Borenstein, “Potential field methods and their inherent limitations for mobile robot navigation,” in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pp. 1398–1404, IEEE, 1991.

[108] M. Kazemi, K. K. Gupta, and M. Mehrandezh, “Randomized kinodynamic planning for robust visual servoing,” *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1197–1211, 2013.

[109] N. J. Cowan, J. D. Weingarten, and D. E. Koditschek, “Visual servoing via navigation functions,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, pp. 521–533, 2002.

[110] G. Chesi and Y. S. Hung, “Global path-planning for constrained and optimal visual servoing,” *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 1050–1060, 2007.

[111] G. Allibert, E. Courtial, and F. Chaumette, “Predictive control for constrained image-based visual servoing,” *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 933–939, 2010.

[112] A. A. Hafez, A. K. Nelakanti, and C. Jawahar, “Path planning approach to visual servoing with feature visibility constraints: a convex optimization based solution,” in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 1981–1986, IEEE, 2007.

[113] A. Remazeilles, N. Mansard, and F. Chaumette, “A qualitative visual servoing to ensure the visibility constraint,” in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp. 4297–4303, IEEE, 2006.

[114] D. Folio and V. Cadenat, “A controller to avoid both occlusions and obstacles during a vision-based navigation task in a cluttered environment,” in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC’05. 44th IEEE Conference on*, pp. 3898–3903, IEEE, 2005.

[115] B. Penin, P. R. Giordano, and F. Chaumette, "Vision-based reactive planning for aggressive target tracking while avoiding collisions and occlusions," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3725–3732, 2018.

[116] D. J. Agravante, G. Claudio, F. Spindler, and F. Chaumette, "Visual servoing in an optimization framework for the whole-body control of humanoid robots," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 608–615, 2017.

[117] A. M. Zanchettin and P. Rocco, "Path-consistent safety in mixed human-robot collaborative manufacturing environments," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1131–1136, IEEE, 2013.

[118] M. Kimmel, M. Lawitzky, and S. Hirche, "6d workspace constraints for physical human-robot interaction using invariance control with chattering reduction," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 3377–3383, IEEE, 2012.

[119] F. Keith, P.-B. Wieber, N. Mansard, and A. Kheddar, "Analysis of the discontinuities in prioritized tasks-space control under discreet task scheduling operations," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 3887–3892, IEEE, 2011.

[120] F. Chaumette, "Image moments: a general and useful set of features for visual servoing," *IEEE Transactions on Robotics*, vol. 20, no. 4, pp. 713–723, 2004.

[121] H. Ferreau, C. Kirches, A. Potschka, H. Bock, and M. Diehl, "qpOASES: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.

[122] P. Viviani and T. Flash, "Minimum-jerk, two-thirds power law, and isochrony: converging approaches to movement planning.," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 21, no. 1, p. 32, 1995.

[123] A. D. Dragan and S. S. Srinivasa, "A policy-blending formalism for shared control," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 790–805, 2013.

[124] S. Pellegrinelli, H. Admoni, S. Javdani, and S. Srinivasa, "Human-robot shared workspace collaboration via hindsight optimization," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pp. 831–838, IEEE, 2016.

[125] R. Ikeura and H. Inooka, "Variable impedance control of a robot for cooperation with a human," in *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, vol. 3, pp. 3097–3102, IEEE, 1995.

[126] T. Tsumugiwa, R. Yokogawa, and K. Hara, "Variable impedance control with regard to working process for man-machine cooperation-work system," in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 3, pp. 1564–1569, IEEE, 2001.

[127] R. Ikeura, T. Moriguchi, and K. Mizutani, "Optimal variable impedance control for a robot and its application to lifting an object with a human," in *Robot and Human Interactive Communication, 2002. Proceedings. 11th IEEE International Workshop on*, pp. 500–505, IEEE, 2002.

[128] T. Tsumugiwa, R. Yokogawa, and K. Hara, "Variable impedance control based on estimation of human arm stiffness for human-robot cooperative calligraphic task," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 1, pp. 644–650, IEEE, 2002.

[129] V. Duchaine and C. M. Gosfselin, "Investigation of human-robot interaction stability using lyapunov theory," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 2189–2194, IEEE, 2008.

[130] B. Lacevic and P. Rocco, "Closed-form solution to controller design for human-robot interaction," *Journal of Dynamic Systems, Measurement, and Control*, vol. 133, no. 2, p. 024501, 2011.

[131] T. Tsumugiwa, Y. Fuchikami, A. Kamiyoshi, R. Yokogawa, and K. Yoshida, "Stability analysis for impedance control of robot in human-robot cooperative task system," *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, vol. 1, no. 1, pp. 113–121, 2007.

[132] F. Dimeas and N. Aspragathos, "Online stability in human-robot cooperation with admittance control," *IEEE transactions on haptics*, vol. 9, no. 2, pp. 267–278, 2016.

[133] C. T. Landi, F. Ferraguti, L. Sabattini, C. Secchi, and C. Fantuzzi, "Admittance control parameter adaptation for physical human-robot interaction," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pp. 2911–2916, IEEE, 2017.

[134] V. Duchaine and C. M. Gosselin, "General model of human-robot cooperation using a novel velocity based variable impedance control," in *EuroHaptics Conference, 2007 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2007. Second Joint*, pp. 446–451, IEEE, 2007.

[135] Y. Maeda, T. Hara, and T. Arai, "Human-robot cooperative manipulation with motion estimation," in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 4, pp. 2240–2245, IEEE, 2001.

[136] Y. Karayiannidis, C. Smith, and D. Kragic, "Mapping human intentions to robot motions via physical interaction through a jointly-held object," in *Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on*, pp. 391–397, IEEE, 2014.

[137] G. Best and R. Fitch, "Bayesian intention inference for trajectory prediction with an unknown goal destination," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 5817–5823, IEEE, 2015.

[138] L. Peternel, T. Petrič, and J. Babič, "Human-in-the-loop approach for teaching robot assembly tasks using impedance control interface," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 1497–1502, IEEE, 2015.

[139] M. Hersch, F. Guenter, S. Calinon, and A. Billard, "Dynamical system modulation for robot learning via kinesthetic demonstrations," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1463–1467, 2008.

[140] L. Rozo, S. Calinon, D. G. Caldwell, P. Jimenez, and C. Torras, "Learning physical collaborative robot behaviors from human demonstrations," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 513–527, 2016.

[141] G. Raiola, X. Lamy, and F. Stulp, "Co-manipulation with multiple probabilistic virtual guides," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 7–13, IEEE, 2015.

[142] A. K. Tanwani and S. Calinon, "A generative model for intention recognition and manipulation assistance in teleoperation," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pp. 43–50, IEEE, 2017.

[143] F. Abi-Farraj, T. Osa, N. P. J. Peters, G. Neumann, and P. R. Giordano, "A learning-based shared control architecture for interactive task execution," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pp. 329–335, IEEE, 2017.

[144] F. Dimeas and N. Aspragathos, "Fuzzy learning variable admittance control for human-robot cooperation," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pp. 4770–4775, IEEE, 2014.

[145] M. Ito, K. Noda, Y. Hoshino, and J. Tani, "Dynamic and interactive generation of object handling behaviors by a small humanoid robot using a dynamic neural network model," *Neural Networks*, vol. 19, no. 3, pp. 323–337, 2006.

[146] Y. Li and S. S. Ge, "Human–robot collaboration based on motion intention estimation," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 3, pp. 1007–1014, 2014.

[147] E. Berger, D. Vogt, S. Grehl, B. Jung, and H. B. Amor, "Estimating perturbations from experience using neural networks and information transfer," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pp. 176–181, IEEE, 2016.

[148] E. De Momi, L. Kranendonk, M. Valenti, N. Enayati, and G. Ferrigno, "a neural network-based approach for trajectory planning in robot–human handover tasks," *Frontiers in Robotics and AI*, vol. 3, p. 34, 2016.

[149] A. M. Zanchettin and P. Rocco, "Probabilistic inference of human arm reaching target for effective human-robot collaboration," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, IEEE, 2017.

[150] P. Trautman, "Assistive planning in complex, dynamic environments: a probabilistic approach," in *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*, pp. 3072–3078, IEEE, 2015.

[151] A. Jazayeri and M. Tavakoli, "Revisiting llewellyn's absolute stability criterion for bilateral teleoperation systems under non-passive operator or environment," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 70–75, IEEE, 2012.

[152] A. Ajoudani, N. G. Tsagarakis, and A. Bicchi, "Tele-impedance: Teleoperation with impedance regulation using a body-machine interface," *The International Journal of Robotics Research*, p. 0278364912464668, 2012.

[153] G. Vezzani, U. Pattacini, and L. Natale, "A grasping approach based on superquadric models," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pp. 1579–1586, IEEE, 2017.

[154] M. J. Zeestraten, I. Havoutis, and S. Calinon, "Programming by demonstration for shared control with an application in teleoperation," *IEEE Robotics and Automation Letters*, vol. 3, pp. 1848–1855, 2018.