POLITECNICO
MILANO 1863

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

NOVEL CONTRIBUTIONS TO ROBOT FORCE CONTROL FOR INDUSTRIAL MANIPULATORS

Doctoral Dissertation of:
Matteo Parigi Polverini

Supervisor:
Prof. Paolo Rocco

Tutor:
Prof. Luca Bascetta

The Chair of the Doctoral Program:
Prof. Andrea Bonarini

Year 2018 – Cycle XXX

To my parents

Acknowledgements

I would like to express my deep gratitude to my supervisor Professor Paolo Rocco for giving me the opportunity to undertake a Ph.D. and enjoy academic research. His support and guidance together with the freedom he gave me to develop my research path, have created a healthy and productive work environment.

I want to thank Professor Ciro Natale and Professor Anders Robertsson for serving as reviewers of this thesis. Their comments and suggestions effectively improved the quality of this work.

Special thanks go to Roberto Rossi and Davide Nicolis with whom I collaborated in fruitful ways and to Professor Andrea Maria Zanchettin for the valuable discussions and support in part of this work.

I would also like to thank Giovanni Buizza Avanzini and Matteo Ragaglia for the stimulating conversations over the years.

Finally, my deepest gratitude and admiration goes to my parents Daniela and Luca for their exceptional love, support and confidence.

Abstract

CONTROL of robot interaction with the environment, generally referred to as *robot force control*, is required to face the inadequacy of pure motion control for the successful execution of those robot tasks involving contact with a surface. Widely popular since the early 1980s, research on force control algorithms employing a conventional single arm robot has gradually lost its appeal during the last decade, despite the growing employment of robots in finishing and machining operations would strongly benefit from increased controllers' performance. At the same time, the recent diffusion of new industrial robotic platforms, like dual-arm lightweight robots, has driven research on robot force control towards the execution of complex and dexterous robotic tasks, such as bimanual automated assembly.

This thesis provides contributions in two main areas of robot force control: performance improvement in *implicit force control* (an implementation of hybrid force-motion control for position controlled robots) for traditional industrial manipulators and force controlled bimanual assembly based on trajectory generation for lightweight dual-arm robots.

Force regulation with improved settling performance and absence of force overshoots is achieved by presenting a *constrained control* approach related to the ideas of *invariance control*, which is subsequently applied to the implicit robot force control problem. Controller robustness to compliance uncertainties is further addressed. Deterioration of force controller performance connected to environment modeling and identification is prevented by employing a *data-driven control design* approach. An on-line

implementation of the controller identification problem is presented, while an outer Model Predictive Controller (MPC) acting as *command governor* is introduced to further enhance the closed-loop performance.

When a dual-arm lightweight robot is used to perform parts assembly, force controlled bimanual assembly can be treated as a trajectory generation control problem fulfilling force control requirements. A *constraint-based trajectory generation* framework is exploited for this purpose, while estimation of the interaction force enables sensorless execution of the assembly operation. The presented approach is developed and experimentally validated in a peg-in-hole insertion task and in a cap rotation task.

Contents

Symbols and Abbreviations

In this thesis, all scalar quantities are described by plain letters (e.g. Δx, y, γ). Bold small and capital letters are used for vectors and matrices, respectively (e.g. $\boldsymbol{q}$, $\boldsymbol{x}$, $\boldsymbol{J}$).
Time derivatives of low order are denoted by a number of dots above the respective symbol: $\dot{y} := \frac{dy}{dt}$, $\ddot{y} := \frac{d^2y}{dt^2}$ and time derivatives of higher order are denoted by $y^{(i)} := \frac{d^i y}{dt^i}$.
The first order Lie-Derivative $\mathcal{L}_{\boldsymbol{f}} h(\boldsymbol{x})$ is the directional derivative of the scalar function $h(\boldsymbol{x})$ in the direction of $\boldsymbol{f}$:

$$\mathcal{L}_{\boldsymbol{f}} h(\boldsymbol{x}) = \frac{\partial h}{\partial \boldsymbol{x}} \boldsymbol{f}$$

Higher-order Lie-Derivatives $\mathcal{L}^i_{\boldsymbol{f}} h(\boldsymbol{x})$ are defined recursively.

Symbols

Several variables in the following list appear in this thesis with different subscripts, superscripts, additional symbols, and various dimensions. Here, the quantities are listed and generally described without further specification. The specific meaning becomes clear when the respective variable is introduced in the text.

Note that this list of variables is not complete, but it only contains quantities which appear at several places in the thesis or are of prominent importance.

$i,\ j$	Indices for numbering
$u,\ uc$	Subscripts for constrained and unconstrained directions
$\boldsymbol{n}$	Vector of constrained direction
$\boldsymbol{J}$	Jacobian matrix
$\boldsymbol{q}$	Vector of joint angles
$\boldsymbol{x}$	Vector of Cartesian coordinates
$\boldsymbol{\tau}$	Vector of joint torques
$\boldsymbol{\mu}$	Vector of forces/moments at the end effector
F	Force
K	Stiffness
C	Compliance
ω	Bandwidth
ζ	Damping ratio
$\boldsymbol{\xi}$	State vector of the output dynamics
Φ	Invariance function
r	Relative degree
γ	Controller parameter
p	Taylor polynomial
$\mathcal{G}$	Admissible set
$\partial\mathcal{G}$	Boundary of $\mathcal{G}$
$\mathcal{M}$	Reference model
ρ	Vector of controller parameters
$\mathcal{C}_\rho$	Controller transfer function

Abbreviations

DoFs	Degrees of Freedom
CNC	Computer Numerically Controlled
MPC	Model Predictive Control
I-O	Input-Output
SISO	Single-Input Single-Output
MIMO	Multi-Input Multi-Output
QP	Quadratic Programming
HQP	Hierarchical Quadratic Programming
VRFT	Virtual Reference Feedback Tuning
LTI	Linear Time-Invariant
TVP	Trapezoidal Velocity Profile
PID	Proportional-Integral-Derivative
TCP	Tool Center Point

CHAPTER 1

Introduction

Robot force control, i.e. control of the interaction between a robot manipulator and the environment, is required to face the inadequacy of pure motion control for the successful execution of a number of practical tasks where the robot end effector has to manipulate an object or perform some operation in contact with a surface.
The contact force, arising from the interaction between the robot and the environment, produces a deviation of the end effector motion from the desired trajectory. This deviation is counteracted by the position control system, eventually leading to a build-up of the contact force until saturation of the joint actuators or breakage of the parts in contact occurs.
Robot force control overcomes this unpleasant situation by ensuring a compliant behavior for the robot during the interaction task. It is therefore required in a variety of industrial tasks where traditional single-arm robots are employed, e.g. polishing, deburring, machining or assembly, and becomes mandatory for modern robotic platforms interacting with unstructured environments.
The crucial importance of robot force control for practical manipulation tasks, makes it one of the most deeply and widely addressed topics in robotic research community since the early 1980s.

This chapter introduces the reader to the overall topic of the present thesis. After an overview in Sec. 1.1, based on the classification in [1],

of the most effective control strategies, that have now become a paradigm when dealing with robot force control with traditional industrial manipulators, the state-of-the-art for modern robotic platforms is summarized in Sec. 1.2. Chapter 1 ends with the research contribution of this thesis and the overview of the content (Sec. 1.3).

1.1 Classical Control Approaches

Robot force control approaches aim at ensuring a compliant behavior for the robot during interaction with the environment. Robot compliant behavior at contact can be achieved either in a passive or in an active way.

In *passive* interaction control the trajectory of the robot end effector is modified by the interaction force/moment due to the inherent compliance of the robot. The most widely adopted mechanical device with passive compliance in industrial applications is the well known *Remote Center of Compliance* (RCC) [2]. The RCC, mounted on the robot end effector, is generally used in quasi-static assembly operations to prevent deadlocks during part insertion. Although simple and cheap, since force/torque sensors are not required, the passive approach to interaction control lacks on the other hand flexibility.

In *active* interaction control, the compliance of the robotic system is mainly ensured by a purposely designed control system. Active interaction control strategies can be grouped into two categories: those performing *indirect force control* and those performing *direct force control*.

The first category achieves force control via motion control, without an explicit closure of a force feedback loop. To this category belongs *impedance control*. The general concept of impedance control was introduced by Hogan in the seminal work [3–5] and can nowadays be considered a classical control approach in robotics. A desired dynamic behavior is imposed to the robot-environment interaction, e.g., a model with forces acting on a mass-spring-damper, so that the robot can mimic the human arm behavior moving in an unknown environment. While impedance control is developed for torque-controlled robots, the equivalent control approach for position-controlled robots, is *admittance control*. Based on measurements of the generalized environmental forces a desired set point for the inner position control loop is generated. Note that admittance control is well suited for industrial robots that are generally equipped with proprietary position or velocity controllers, while no joint torque or motor current interfaces are provided for the user. Special cases of impedance and admittance control are *stiffness control* and *compliance control* [6], respectively, together with

damping control [7].

Direct force control instead allows to control the contact force and moment to a desired value, thanks to the closure of a force feedback loop. The most widely adopted strategy belonging to this category is *hybrid force-motion control*, which aims at controlling the motion along the unconstrained task directions and force (and moment) along the constrained directions. Differently from indirect force control, an explicit model of the interaction task is now required. In fact, the user has to specify the desired motion and the desired contact force and moment in a consistent way with respect to the constraints imposed by the environment. The concepts of *natural* (i.e. imposed by a rigid environment) and *artificial constraints* (i.e. imposed by the controller) and of *compliance frame* were introduced in [8]. These ideas have been systematically developed in [9] within the *task frame* formalism and generalized with iTaSC (instantaneous Task Specification using Constraints) [10], consisting in a systematic constraint-based methodology to specify complex tasks. The original hybrid force-motion control concept was introduced in [11], relying on the natural and artificial constraint task formulation and on a purely kinematic robot model. Based on this task decomposition, hybrid force-motion control allows simultaneous control of both the contact force and the end effector motion in two mutually independent subspaces. Simple *selection matrices* acting on both the desired and feedback quantities serve this purpose for planar contact surfaces, whereas suitable *projection matrices*, derivable from the explicit constraint equations [12–14], must be used for general contact tasks. Several implementation of hybrid force-motion control schemes are available, e.g., based on inverse dynamics control in the operational space [15], passivity-based control [16], or outer force control loops closed around inner motion loops, typically available in industrial robots [9]. A detailed description of the control implementation in [9], hereafter referred to as *implicit force control*, will be given in Chapter 2. If an accurate model of the environment is not available, the force control action and the motion control action can be superimposed, resulting in a *parallel force/position control* scheme [17]. In this approach, the force controller is designed so as to dominate the motion controller; hence, a position error would be tolerated along the constrained task directions in order to ensure force regulation.

Finally, adaptive and robust control algorithms have been proposed for both indirect and direct force control methods to overcome uncertainties in the dynamic parameters of the robot manipulator and in the environment geometry, see e.g. [18–20] and [21–23].

1.2 Current Research Challenges

During the last decade robotic research has been mainly driven by the birth of new robotic platforms, e.g. light-weight robots and collaborative robots (Cobots), humanoids, mobile and aerial manipulators, which allow for the execution of a wide variety of complex tasks, that can not be performed by a traditional robot. As a consequence, the type of interactions requiring force control algorithms has considerably changed from traditional industrial applications.

Interaction with an unstructured environment is indeed the main challenge for legged humanoid robots, where force control is generally employed in grasping and object manipulation: think of a door opening task [24, 25], but (indirectly) also in locomotion [26, 27]. Similarly, Unmanned Aerial Vehicles (UAVs) require to actively perform tasks involving mechanical interactions with external environments/objects/humans, see [28, 29].

In recent years, indirect force control approaches, especially impedance control, have been extensively employed compared to direct force control methods. This is mainly due to their effectiveness in executing complex tasks where high accuracy is not a requirement. In this respect, when impedance control is applied to a manipulator, whose redundancy is exploited to simultaneously perform multiple task with a given order of priority, null space projections affect the passivity of the system. Approaches to restore the passivity for null space impedance and compliance control can be found in [30, 31]. It is also worth noticing that research on impedance control for physical Human-Robot Interaction (pHRI) has been recently motivated by the impedance modulation strategies typical of living beings. Adjustment of the impedance/admittance parameters on the basis of the inferred human intention has been extensively investigated [32–34]. Also in surgical and rehabilitation scenarios, stiffness regulation plays an important role to ensure accuracy and safety during interaction with unpredictable dynamic environments. An energy tank approach to reproduce time-varying parameters in impedance and admittance control for surgical robotics can be found in [35, 36]. These topics are also of interest in the applications where Cobots are employed.

A not comparable effort has been put instead in research on force control for typical industrial robots, i.e. manipulators neither redundant nor collaborative. In recent years, industrial attention (unlike research attention), although originally driven by automotive assembly, has mainly focused on machining and finishing applications, such as polishing [37]. As a matter of fact, an industrial manipulator can provide a more constant force

than a human being while preventing the risk of injuries inherent in a long day of hand polishing. Moreover, industrial manufactures now offer force control packages [38], e.g. ABB's Integrated Force Control [39], KUKA ForceTorqueControl [40], Fanuc force sensor [41]. On the other hand, it is generally accepted that robot arms, which inherently lack the stiffness of a machine tool, will not be able to perform precision machining operations comparable to Computer Numerically Controlled (CNC) machines. As a common trend instead, force controlled robots are progressively employed in rough pre-machining operations, leaving only a single pass on CNC machine for finish machining [37].

Finally, the commercial availability of redundant light-weight Cobots, e.g. Willow Garage PR2, Rethink Robotics Baxter, Kawada Nextage and the ABB YuMi (one of the experimental platforms in this thesis), has recently opened up the possibility to include robotic systems in automated industrial assembly processes, as a part of a highly efficient production line [42, 43]. The redundancy and the lower inertia combined with the compliant structure of this new type of robots provides an intrinsic degree of safety with respect to the environment, which is desirable in cooperative assembly operations. On the other hand, the consequent lower position accuracy makes the application of force control algorithms even more crucial for a successful task execution compared to traditional industrial manipulators. Examples of dual-arm collaborative assembly can be found in [44], while recent examples of robotic assembly using contact force control and estimation with the ABB Yumi are [45–47].

1.3 Research Contributions and Overview

Two main areas of control of robot interaction with the environment are covered in this thesis: direct robot force control for traditional industrial robots and force controlled robotic assembly for bimanual light-weight robots.

As outlined in the last section, comparatively few works have been published in the last decade in force control for traditional robots. Nevertheless, the growing employment of robots in finishing and machining operations, that would strongly benefit from increased controllers' performance, demands for further research on direct force control methods.

At the same time, the spread of new industrial robotic platforms, such as dual-arm light-weight robots, that allow to perform more dexterous robotic tasks, motivates research on automated assembly operations.

In this regard, the present thesis provides the following contributions:

I - Performance improvement in *implicit force control*, a state-of-the-art implementation of hybrid force-motion control for position controlled industrial robots, is addressed in the first Part of this work (Chapters 2-4).

  (i) A *constrained control* strategy is proposed related to the ideas of *invariance control*, where set invariance conditions are exploited to achieve output regulation with fast convergence speed and absence of output overshoots (Chapter 3). Its application and experimental validation to robot implicit force control are presented, while robustness to compliance uncertainties is provided in an adaptive fashion, and alternatively by ensuring robust set invariance.

  (ii) A *data-driven control design* approach is proposed to prevent deterioration of the force controller performance connected to *environment* modeling and identification (Chapter 4). An *online* implementation of the controller identification problem is additionally presented, while a *hierarchical* control architecture is introduced to enhance the closed-loop performance.

II - In the second Part of the thesis (Chapters 5-6), force controlled bimanual assembly for light-weight dual-arm robots is treated as an equivalent*constraint-based trajectory generation* control problem fulfilling force control requirements. The presented approach is developed and experimentally validated in a peg-in-hole insertion task and in a cap rotation task.

The organization of this thesis is summarized in the following.

Chapter 2 provides background knowledge on the implicit force control approach. The robot used in the experiments is presented in this chapter, together with the interface available for control. The inherent limitations of state-of-the-art implicit force control are analyzed and corresponding performance improvements are highlighted.

Chapter 3 addresses the problem of force regulation with improved settling performance and absence of force overshoots by means of constrained control and set invariance theory. After some details on the invariance control approach, a control strategy is introduced overcoming its main performance limitations. The proposed control approach is applied to implicit

force control and experimentally validated. Additionally, on the second part of this chapter, robustness with respect to compliance uncertainties is treated in more details, showing how the proposed controller can be properly modified to deal with this type of uncertainty.

Chapter 4 presents a data-driven control approach to be applied to implicit force control in order to prevent performance deterioration connected to a rough identification of the environment or to the presence of unmodeled dynamics. The proposed data-driven controller is based on the *Virtual Reference Feedback Tuning* (VRFT) approach. Advantages over standard model-based implicit force control are discussed and demonstrated. An on-line implementation of VRFT algorithm is further presented, based on the Recursive Least Squares (RLS) estimation method, while an outer Model Predictive Controller (MPC) acting as a *command governor* is applied to the VRFT closed-loop system aiming at enhancing closed-loop performance.

Chapter 5 provides details on the constraint-based trajectory generation control framework in [48], here exploited for bimanual robotic assembly. The dual-arm robot used for the experimental validation and its control interface are presented, together with the employed model-based observer of interaction forces.

Chapter 6 treats two bimanual force controlled assembly tasks: peg-in-hole insertion and cap rotation, as a constraint-based trajectory generation problem ensuring force control specifications. Admittance control is exploited within the presented control approach to the peg-in-hole insertion problem, while the introduced constraint on the interaction force, arising during the cap rotation task, allows for a successful task execution in presence of robot-environment contact model uncertainties, in addition to force measurement noise and surface uncertainties.

The research findings reported in this thesis resulted in the journal article

[J-1] 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, 2017

and in the following conference papers

[C-5] M. Parigi Polverini, D. Nicolis, A. M. Zanchettin, P. Rocco “Robust Set Invariance for Implicit Robot Force Control in Presence of Contact Model Uncertainty”, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6393-6399, 2017

[C-4] M. Parigi Polverini, A. M. Zanchettin, F. Incocciati, P. Rocco "Robust Constraint-Based Robot Control for Bimanual Cap Rotation", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4785-4790, 2017

[C-3] M. Parigi Polverini, S. Formentin, L. A. Dao, P. Rocco, "Data-Driven Design of Implicit Force Control for Industrial Robots", in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2322-2327, 2017. **Best Student Paper Award Finalist**

[C-2] M. Parigi Polverini, R. Rossi, G. Morandi, L. Bascetta, A. M. Zanchettin, P. Rocco, "Performance Improvement of Implicit Integral Robot Force Control through Constraint-Based Optimization", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3368-3373, 2016

[C-1] M. Parigi Polverini, A. M. Zanchettin, S. Castello, P. Rocco, "Sensorless and Constraint-Based Peg-in-Hole Task Execution with a Dual-Arm Robot", in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 415-420, 2016

Figure 1.1 maps these publications into the Chapters of the thesis.

The following publications, related to the author's contributions in Human-Robot Interaction, are not part of the present thesis.

- M. Parigi Polverini, A. M. Zanchettin, P. Rocco, "A Computationally Efficient Safety Assessment for Collaborative Robotics Applications", in *Robotics and Computer Integrated Manufacturing*, vol. 46, pp. 25-37, 2017

- R. Rossi, M. Parigi Polverini, A. M. Zanchettin, P. Rocco, "A Pre-Collision Control Strategy for Human-Robot Interaction Based on Dissipated Energy in Potential Inelastic Impacts", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 26-31, 2015. **Best Student Paper Award Finalist**

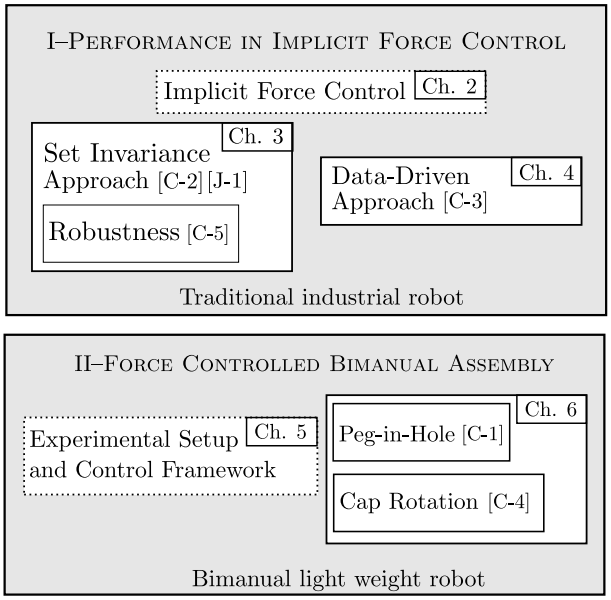


Figure 1.1: Graphical overview of the Chapters, the main topics, and the relation to the publications.

Part I

PERFORMANCE IN IMPLICIT FORCE CONTROL

CHAPTER 2

The Implicit Force Control

In usual industrial scenarios, robots are typically equipped with proprietary position (or velocity) controllers, while no joint torque sensing or direct motor current interfaces are provided for the user. Moreover, the interaction forces and torques are generally measured by means of 6 DoFs force/torque sensors mounted at the robot end effector. This kind of setting allows for the application of robot *direct force control* [1] algorithms, i.e. regulation of the contact force and moment to a desired value thanks to the closure of a force feedback loop, typically required in robotic polishing, deburring and machining. *Hybrid force-motion control* [11] is the most popular control approach amongst those performing direct force control. Its implementation for typical industrial position controlled robots [49], as the experimental platforms in this thesis, is hereafter referred to as *implicit force control*.

Implicit force control is described in Sec. 2.1. After the robotic platform used in the experiments is presented in Sec. 2.2, the considered interaction model, accounting for robot joint and link compliance together with environment compliance, is treated in Sec. 2.3. The complete robot controller structure is given in Sec. 2.4. Finally, Sec 2.5 underlines the limitations inherent in implicit force control and consequently defines corresponding controller performance improvements.

2.1 Control Approach

Implicit force control was originally proposed in [49]. It relies on the task specification introduced in [9] and is based on an external force control loop, closed around the inner robot positioning system.

The inner position control loop tends to decouple and linearize the robot dynamics, therefore it results in a system which is easy to model and easy to control for the external controller.

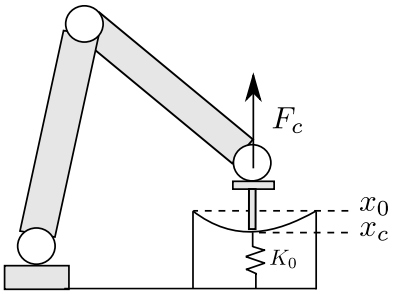


Figure 2.1: Compliant robot-environment contact situation.

Assuming a purely elastic contact situation, see Fig. 2.1, the robot displacement Δx_c along the direction elastically constrained by the environment (hereafter denoted with subscript c)

$$\Delta x_c = x_c - x_0 \tag{2.1}$$

i.e. the difference between the actual robot position x_c and the position of the environment x_0, produces a contact force via stiffness K_0

$$F_c = K_0 \, \Delta x_c \tag{2.2}$$

Although the contact force acts as a disturbance on the position control loop, this effect can be neglected assuming the environment stiffness to be small compared to the servo stiffness. The sensor dynamics can also be neglected assuming that the structural resonance frequency introduced by the force sensor lies well enough above the position loop bandwidth.

Fig. 2.2 shows the general 1 dimensional external force control scheme: the error between the desired force $\overline{F}_c$ and actual force F_c is fed into the force controller $g_c(s)$, which generates an end effector position command $\overline{x}_c$. Note that, the inner position control loop can be considered stiff, while the outer loop is responsible for the compliance. In the remainder of the chapter the end effector unconstrained directions are assumed to be position controlled according to the hybrid force-motion control paradigm; their representation will be therefore omitted in the reported control schemes.

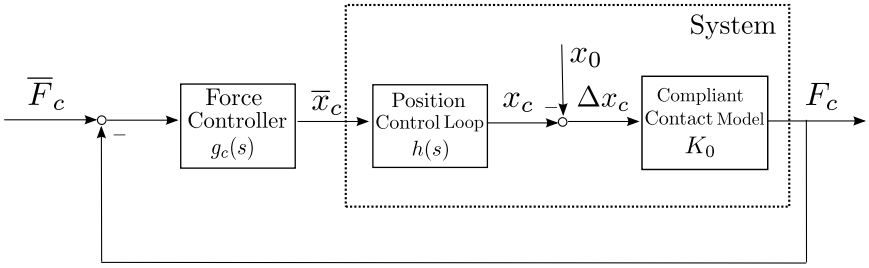


Figure 2.2: 1-dimensional external force control: simplified scheme.

If the closed loop transfer function of the internal position control system $h(s)$ is known, the external force control law can be designed with standard techniques. According to [49], we hereafter assume $h(s)$ to be linear and second order

$$h(s) = \frac{\omega_n^2}{s^2 + 2\zeta_n\omega_n s + \omega_n^2} \tag{2.3}$$

where ω_n and ζ_n are the position control bandwidth and damping ratio, respectively. It is highly desirable in all position controlled robot systems to have ζ_n to avoid overshoots. Finally note that, by denoting with $\Delta\overline{x}_c$ the desired end effector displacement with respect to the position of the environment

$$\Delta\overline{x}_c = \overline{x}_c - x_0 \tag{2.4}$$

the control scheme in Figure 2.2 is equivalent to the control scheme in Figure 2.3 where the force controller $g_c(s)$ generates the end effector displacement command $\Delta\overline{x}_c$.

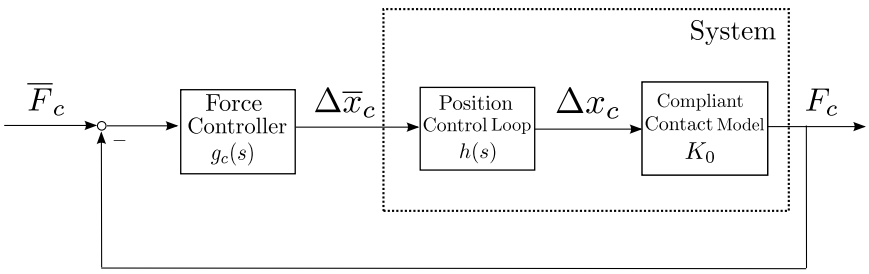


Figure 2.3: 1-dimensional external force control: equivalent scheme.

2.1.1 Integral Force Controller

Purely integral control [49, 50], the simplest control method yielding no steady state error and ensuring a robust behavior with respect to time de-

lays, e.g. due to filters on the force measurement [51, 52], will be considered as a baseline controller for this thesis, and described with some details hereafter. Hunting limit cycles induced by static friction can be prevented by employing an I-controller with dead-zone, see e.g. [53]. For the sake of fairness, note that the negative effects of time delay and uncertainties on contact stiffness can be counteracted also by means of PID-like controllers [54, 55].

In addition to the integral action, the force controller $g_c(s)$ needs to contain a compensation factor for the environment stiffness K_0, i.e.

$$g_c(s) = C_0 \, \frac{K_I}{s} \tag{2.5}$$

where $C_0 = K_0^{-1}$ is the environment compliance and K_I is the integral gain. The optimal gain can be therefore determined by means of standard control design techniques, leading to

$$K_I \approx 0.5 \, \omega_n \tag{2.6}$$

The resulting bandwidth of the closed force loop, which is directly set by the choice of the integral gain, is about half the position loop bandwidth. Improving the performance of this controller with the novel tools developed in this thesis is one of our targets.

2.2 Robot and Interface

The experimental setup consists of a 6 degrees of freedom (DoFs) COMAU Smart Six robot, see Fig. 2.4: a $6\ kg$ payload position controlled industrial manipulator manufactured by COMAU, equipped with the *open* version of the COMAU C4G controller, programmable in C environment.
In the open version, C4G (client) is linked to a real-time external PC through a real-time ethernet connection based on RTnet protocol. The real-time external PC (server) is based on RTAI Linux real-time extension. The industrial joint position controller sampling rate is $500\ Hz$ ($2\ ms$ sampling time T_s). A second order low-pass filter, with bandwidth $\omega_f = 75.4\ rad/s$ and unitary damping ratio ζ_f, is present on the joint reference signal to avoid the excitation of the mechanical resonance frequencies of the robot.

The COMAU Smart Six is equipped with a wrist mounted six DoFs ATI Gamma SI-130-10 force/torque sensor. The sensor communicates with the external PC through a DAQ board that is managed by the RTAI system, thanks to a real-time extension of the Comedi drivers.

Figure 2.4: The COMAU Smart Six, 6 kg payload position controlled industrial manipulator, used for experimental validation.

2.3 Compliant Contact Model

Assuming a compliant contact situation, when a contact force is exchanged between the environment and the robot end effector, the elastic deformations of robot joints and links, together with the environment deformation, can be reported to the contact point as an equivalent end effector displacement, see [50, 53]. Note that the approach in [50, 53] has been validated on the same experimental platform used in this thesis, described in the previous Section.

By modeling the system as an equivalent series of springs at the end effector, the total compliance $C_T(\boldsymbol{q}) \in \mathbb{R}$ along the direction constrained by the environment is given by the following expression

$$C_T(\boldsymbol{q}) = \boldsymbol{n}^T \Big(C_J(\boldsymbol{q}) + C_L(\boldsymbol{q}) \Big) \boldsymbol{n} + C_E \tag{2.7}$$

where $\boldsymbol{q} \in \mathbb{R}^6$ are the (link-side) robot joint coordinates, $\boldsymbol{n} \in \mathbb{R}^6$ is the vector representing the constrained direction, e.g. for the case of z constrained direction $\boldsymbol{n} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^T$. Finally, $C_E \in \mathbb{R}$ is the environment compliance, while $C_J(\boldsymbol{q})$, $C_L(\boldsymbol{q}) \in \mathbb{R}^{6\times 6}$ are the joint and link compliance matrices, respectively.
Since this modeling approach is not an original contribution of the present thesis, the reader can refer to [50] for details on the analytical expression and parameter identification for the joint and link compliance matrices.

The resulting (motor-side) end effector displacement Δx_c that corresponds to the contact force F_c is given by

$$\Delta x_c = x_c - x_0 = -C_T(\boldsymbol{q})\, F_c \tag{2.8}$$

where $x_c \in \mathbb{R}$ is the (motor-side) end effector constrained position, while $x_0 \in \mathbb{R}$ is the end effector position on the environment undeformed surface, which is hereafter assumed to be known with accuracy of approx. $1\ mm$.

Under the assumption of small displacements, the corresponding (motor-side) joint increment $\Delta \boldsymbol{q}_c$ can be computed through simple kinematic inversion

$$\Delta \boldsymbol{q}_c = \boldsymbol{J}(\boldsymbol{q})^{-1} \boldsymbol{n}\, \Delta x_c \tag{2.9}$$

where $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{6\times 6}$is the robot Jacobian matrix.

Finally, according to [53], the environment compliance C_E can be estimated based on the following expression

$$\hat{C}_E = -\Delta x_c\, F_c^{-1} - \boldsymbol{n}^T \Big(\hat{C}_J(\boldsymbol{q}) + \hat{C}_L(\boldsymbol{q})\Big)\boldsymbol{n} \tag{2.10}$$

where $\hat{C}_J(\boldsymbol{q})$ and $\hat{C}_L(\boldsymbol{q})$ are the off-line estimated robot joint and link compliance matrices [50]. Note that the estimation of the environment compliance $\hat{C}_E$ can be performed either off-line by means of Least Square Estimation (LSE) or on-line by means of an Exponentially Weighted Recursive Least Squares (EWRLS) algorithm.

2.4 Robot Force Controller

Based on Sec. 2.1 and on the contact model described in the previous Section, the robot force controller $g_c(s)$ contains an integral action combined with the compensation of the robot and environment stiffness

$$g_c(s) = \hat{C}_T(\boldsymbol{q})\, \frac{K_I}{s} \tag{2.11}$$

where $K_I \in \mathbb{R}$ is the integral gain and $\hat{C}_T(\boldsymbol{q})$ is the estimated total compliance, according to [53]. The force controller generates the end effector displacement command $\Delta \overline{x}_c \in \mathbb{R}$, which is in turn translated in an equivalent joint displacement command $\Delta \overline{\boldsymbol{q}}_c \in \mathbb{R}^6$ through kinematic inversion, see (2.9). Fig. 2.5 shows the resulting 1 dimensional robot force control scheme.

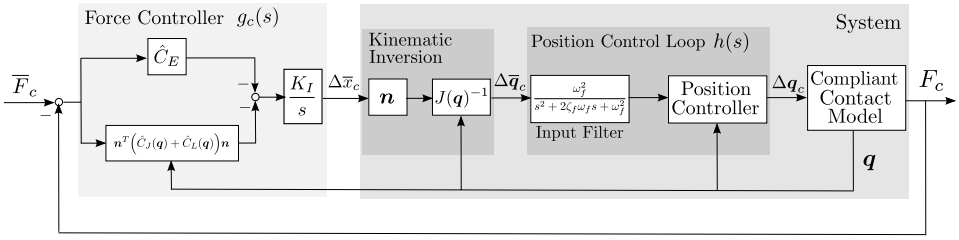


Figure 2.5: 1-dimensional external force control: complete scheme.

Although not represented in Fig. 2.5, the actual input of the robot position control loop $\overline{\boldsymbol{q}} \in \mathbb{R}^6$ is given by

$$\overline{\boldsymbol{q}} = \boldsymbol{q}_0 + \Delta\overline{\boldsymbol{q}}_c + \Delta\overline{\boldsymbol{q}}_{uc} \tag{2.12}$$

where $\boldsymbol{q}_0 \in \mathbb{R}^6$ is the initial robot configuration in contact with the environment, while, according to the hybrid force-motion control approach, $\Delta\overline{\boldsymbol{q}}_{uc} \in \mathbb{R}^6$ represents the joint displacement command responsible for the end effector motion in the unconstrained directions (hereafter denoted with subscript uc).

Furthermore, in the following the second order low-pass input filter on the joint reference signal will be considered as an approximation of the closed loop transfer function of the inner position control loop $h(s)$, see (2.3), therefore

$$\begin{aligned} \omega_n &\approx \omega_f \\ \zeta_n &\approx \zeta_f \end{aligned} \tag{2.13}$$

where $\omega_f = 75.4\ rad/s$ and $\zeta_f = 1$, according to Sec. 2.2.

2.5 Control Limitations and Performance Improvement

Although effective, implicit force control as described before shows some inherent limitations that will be highlighted in the following remarks.
This discussion is supported by experimental results from force regulation experiments in presence of a step reference signal $\overline{F}_c$ of $15\ N$ performed on the presented experimental platform (Sec. 2.2). The environment setup ($K_E \approx 10^5\ N/m$) consists of a set of commercial springs that sustain an aluminum plate, whose normal is parallel to the end effector z Cartesian direction, see Fig. 2.6. No additional displacement in the unconstrained directions, has been assigned, thus $\Delta\overline{\boldsymbol{q}}_{uc} = 0$. The control schemes analyzed in this thesis rely on the assumption that the manipulator end effector is in

Figure 2.6: Implicit force control experimental setup.

contact with the environment and that this contact is not lost, therefore every experiment shown hereafter is performed after having established contact between the robot end effector and the environment surface.

- **Remark 1:** As it can be noticed from Fig. 2.7, by increasing the integral gain a clear trade-off arises between rise time and occurrence of force overshoots. As a matter of fact, even for controller gains well below the optimal gain determined by standard control techniques ($K_I \approx 0.5\,\omega_n$), a decrease in the rise time occurs at the expense of oscillations in the force response. On the other hand, it is worth pointing out that typical industrial application of implicit force control, such as polishing, would benefit of both absence of force overshoots, in order to avoid damages to the workpiece, and low settling time, to speed up the task execution.

- **Remark 2:** An accurate description of the robot-environment interaction model is beneficial in implicit force control. On the other hand, while the model of the robot is known with enough precision, a detailed description of the environment is generally difficult to obtain; this eventually affects the force control performance, as it will be shown in the following.

  Even in presence of a purely elastic environment K_0, a rough environment model identification, leading to an under-estimate of the environment stiffness, i.e. $\hat{K}_0 < K_0$, results in an increased integral gain

$$\overline{K}_I = K_I\,\hat{C}_0\,K_0 > K_I \tag{2.14}$$

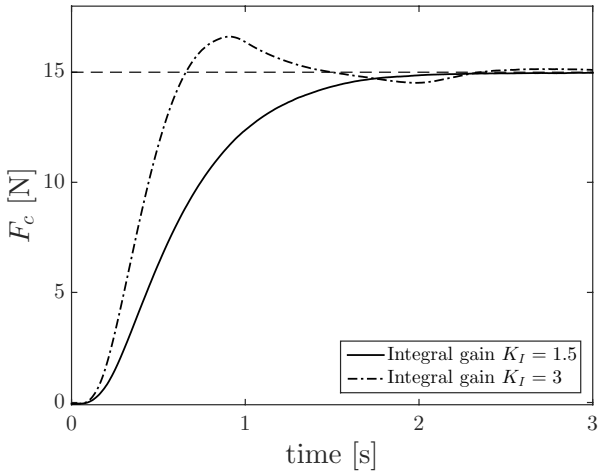


Figure 2.7: Experimental results: impact of the value of integral gain K_I on rise time and on the occurrence of force overshoots.

potentially inducing force overshoots, as shown in the previous remark, or even instability of the system, see [53]. Conversely, an overestimate of the environment stiffness, results in a lower integral gain that will consequently decrease the convergence speed.

Possible unmodeled dynamics, e.g. environment damping and/or filters on the force measurements, produce a mismatch between the ideal closed-loop behavior, i.e. without under-modeling, and the achieved closed-loop behavior. This can be clearly noticed in Fig. 2.8, where the measured force response, produced by an integral controller with $K_I = 1.5$, is compared with the expected step response, given by the first order closed-loop transfer function $\mathcal{M}_{K_I}(s)$ resulting from standard control design techniques, i.e.

$$\mathcal{M}_{K_I}(s) = \frac{K_I}{s + K_I}$$

The inherent limitations of implicit force control, highlighted in the previous remarks, correspond to control performance objectives that could have beneficial implications on typical industrial applications. Based on this discussion, two requirements for an improved control performance are introduced in the following and analyzed in the next Chapters.

- **Requirement 1:** Fast convergence speed with absence of force over-

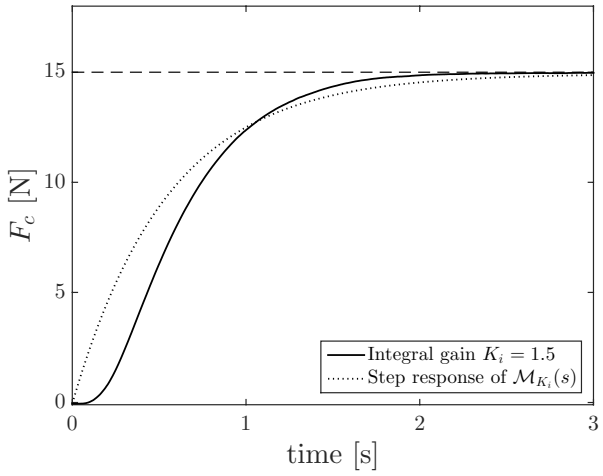


Figure 2.8: Experimental results: comparison between actual force response with integral controller $K_I = 1.5$ (solid line) and expected first order force response (dotted line).

shoots (addressed in Chapter 3);

- **Requirement 2:** Avoidance of environment modeling and identification (addressed in Chapter 4).

CHAPTER 3

Implicit Force Control Based on Set Invariance

The first performance requirement introduced in Sec. 2.5, i.e. fast convergence speed with absence of force overshoots, can be treated as a constrained control problem formulated with respect to the output, i.e. the interaction force, of the robot-environment system. The constrained control approach presented in this chapter relates to the ideas of *invariance control* and is based on [56–58]. The main objective is to develop a control law to be applied to the implicit force control problem, that guarantees the considered performance requirements.

After an overview on control methods for constrained systems and their application to robot control problems in Sec. 3.1, fundamentals on the invariance control approach are provided by Sec. 3.2. In Sec. 3.3, a control strategy is proposed related to the ideas of invariance control, where set invariance conditions are fully exploited to achieve output regulation with fast convergence speed and absence of output overshoots. Its application to robot implicit force control and its subsequent experimental validation are shown in Sec. 3.4. Controller robustness to compliance uncertainties, addressed in Sec. 3.5, is first ensured in an adaptive fashion in Sec. 3.6, while Sec. 3.7 proposes a modification of the constraint on the interaction force that can provide smooth convergence to the reference.

3.1 Introduction

A selection of the established methods to address constrained control problems spans: anti-windup/override control [59], theory of constraint admissible/invariant sets [60], optimal control/model predictive control [61, 62], the reference governor approach [63], backstepping using barrier functions [64] and invariance control [65, 66].

A first cathegory of control methods explicitly takes into account state constraints during the controller design procedure. For example, in [64] a backstepping approach is proposed that seeks to reshape control Lyapunov functions using barrier functions that grow to infinity on the constraints.

Alternatively, the command signal can be modified on-line to prevent constraint violations. A very successful example of such methods in the framework of optimal control is the well known Model Predictive Control (MPC), where an open loop optimal control problem is repeatedly solved to determine the control signal satisfying input, state and output constraints. Stability and robustness properties are usually guaranteed by a *terminal set* constraint: a sub-level set of a Lyapunov function, therefore control invariant. Note that invariant set theory [60] is connected to MPC stability and to the feasibility of the control problem at all time instants [67]. Although efficient explicit solutions of the optimal control problem exist for linear systems [62], the application of general MPC techniques to high-dimensional nonlinear systems with strict real-time requirements can often be computationally expensive, especially in presence of fast system dynamics, as in a robotic manipulator. Additionally, the behavior within the admissible set is difficult to predict in advance during the controller definition phase.

The *reference governor* approach is tightly related to MPC techniques and relies on the definition of an override supervisor (the governor) that promptly adapts the references, fed to an inner closed loop system, to prevent constraint violation in transient phases [63]. In [68] the authors introduce a more general formulation of this technique that does not require the definition of Lyapunov functions or set invariance. While its add-on nature makes it a popular choice for existing control schemes, the extensive numerical simulation often required to check for constraint admissibility might compromise its applicability [69].

Of particular interest in this thesis is the *invariance control* approach, proposed in [65,66], which has proved to be particularly suitable for robotic applications [70–72]. Here the constrained control problem is addressed by making a constraint admissible set *positively invariant* through Input-Output (I-O) Linearization and switching control. Note that a set is said to

be positively invariant with respect to a dynamical system if a trajectory, with initial condition in the set, remains therein for all future times [60]. Using this method, that will be described in details in Sec. 3.2, a nominal controller ensures stability of the system with respect to the tracking error neglecting the constraints, while the invariance controller monitors the state of the system and modifies the control signal on the boundary of an invariant set. For single input systems, this results in a switching action between the *nominal control* mode and the *corrective control* mode.

Concerning robot control, constrained control has been mainly applied to navigation scenarios requiring collision avoidance. Robot motion in a constrained environment is achieved in [73] by defining overlapping collision free sets. The related control laws enable to drive the robot towards the goal, if a suitable control switching occurs in the overlapped regions. A similar approach is used in [74] where a potential function in each set drives the robot in the direction of an outlet face that guarantees progress towards the goal. Both approaches avoid the occurrence of local minima, however the second one requires to recompute the local control laws each time the goal changes.

Although originally employed in path tracking problems, examples of constrained control approaches can be also found in robot force control. In [75], the authors apply the so called *prescribed performance* approach, introduced in [76]. A state feedback control law is derived ensuring the force tracking error to be bounded between two exponential functions, shaped in order to achieve desired transient and steady state performance specifications. The tracking performance of the developed scheme is fully decoupled by both the control gains selection and the robot dynamic model. This technique offers relevant robustness results, however the tuning of the parameters is not always straightforward. In [77] the authors present a particular application of the constraint based trajectory generation control framework proposed in [78,79] to a force-motion tasks performed by a dual arm redundant robot. Similarly to the invariance control approach, here the nominal control action is constrained by upper and lower bounds on the measured force, although resulting in a smaller invariant set compared to the one derivable from [66]. Finally, in [80, 81] an optimization based control strategy is proposed to track a reference force signal, that relies on the indirect force controller available in the used experimental platform. The force tracking requirement is represented as a constraint for an optimization problem expressed on the reference joint coordinates. Although simple and intuitive, this method is yet unsuitable for high precision task.

3.2 The Invariance Control Approach

In the following we summarize the main results of [65, 66, 70] to provide an introduction on invariance control.

Consider the nonlinear control affine system:

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})u \\ y_{out} = h_{out}(\boldsymbol{x}) \end{cases} \tag{3.1}$$

with $\boldsymbol{x} \in \mathbb{R}^n$, $u \in \mathbb{R}$, $y_{out} \in \mathbb{R}$, while $\boldsymbol{f}, \boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^n$, and $h_{out} : \mathbb{R}^n \to \mathbb{R}$ are sufficiently smooth functions to apply Input-Output (I-O) Linearization. A *nominal control* u_{nom} is chosen to ensure stability of the system with respect to the tracking error $e = y_{out} - y_d$, based on the assumption that there exists a positive definite Lyapunov function guaranteeing stability.

The considered m state constraints are defined by means of an output function for each constraint

$$y_i = h_i(\boldsymbol{x}) \leq 0, \;\; 1 \leq i \leq m \tag{3.2}$$

which is equal to zero exactly on the constraint and negative within the admissible set.

3.2.1 Input-Output Linearization

By denoting with r_i the relative degree of the output function y_i, the first $r_i - 1$ time derivatives of y_i do not algebraically depend on the control input, while $y_i^{(r_i)}$ depends on u through

$$y_i^{(r_i)} = \mathcal{L}_{\boldsymbol{f}}^{r_i} h_i(\boldsymbol{x}) + \mathcal{L}_{\boldsymbol{g}} \mathcal{L}_{\boldsymbol{f}}^{r_i - 1} h_i(\boldsymbol{x}) u \tag{3.3}$$

As a matter of fact, rather then being calculated by the function h_i from the state, the output y_i can be alternatively interpreted as generated by a chain of r_i integrators, illustrated in Fig.3.1, that is fed by a nonlinear function of the state and the control input.
Equation (3.3) allows to compute the input to this integrator chain for a given control input. Conversely, introduction of the *pseudo control input* $\overline{y}_i^{(r_i)}$ and inversion of (3.3) leads to a feedback control law that allows to set $y_i^{(r_i)}$ to the desired value $\overline{y}_i^{(r_i)}$ by proper choice of the control input. The described procedure is well known from Input-Output Linearization, but in invariance control, the linear description of the output dynamics is not directly used for control design but to construct an admissible set that can be made invariant by proper choice of the control input on its boundary.

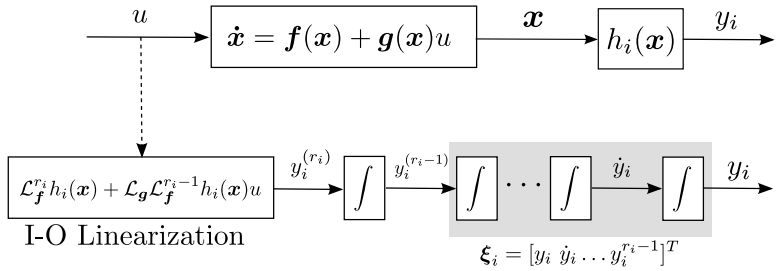


Figure 3.1: Structure of Input-Output Linearization.

3.2.2 The Invariance Function

Let's denote with $\boldsymbol{\xi}_i$ the state space of the output dynamics

$$\boldsymbol{\xi}_i = \begin{bmatrix} y_i & \dot{y}_i & \dots & y_i^{(r_i-1)} \end{bmatrix}^T \tag{3.4}$$

and assume that the output derivative $y_i^{(r_i)}$ is upper bounded by a constant controller parameter $\gamma_i < 0$ for some open time interval $\tau \in (0, \Delta)$

$$y_i^{(r_i)}(t+\tau) \leq \gamma_i \tag{3.5}$$

The future output trajectory is upper bounded by the Taylor series $p_i(\tau, \boldsymbol{\xi}_i, \gamma_i)$ affine in $\boldsymbol{\xi}_i$

$$y_i(t+\tau) \leq p_i(\tau, \boldsymbol{\xi}_i, \gamma_i) := \frac{\tau^{r_i}}{r_i!}\gamma_i + \sum_{k=0}^{r_i-1} \frac{\tau^k}{k!} y_i^{(k)} \tag{3.6}$$

The admissible region for the i-th constraint can be now defined by means of the so-called *invariance function* $\Phi_i(\boldsymbol{\xi}_i)$.

The invariance function for the output y_i is obtained by taking the maximum of (3.6) for $\tau \geq 0$, i.e.

$$\Phi_i(\boldsymbol{\xi}_i) := \max_{\tau \geq 0}\ p_i(\tau, \boldsymbol{\xi}_i, \gamma_i) \tag{3.7}$$

This maximum gives a worst case estimation for the future output trajectory as long as the pseudo control input $\overline{y}_i^{(r_i)} = \gamma_i$ is applied and for $\Phi_i(\boldsymbol{\xi}_i) \leq 0$ defines the admissible region for the i-th constraint.

For low relative degrees, i.e. $r_i \leq 3$, the invariance function can be

analytically computed as follows

$$
\begin{aligned}
r_i = 1 &: \Phi_i(\boldsymbol{\xi}_i) = y_i \\
r_i = 2 &: \Phi_i(\boldsymbol{\xi}_i) = \begin{cases} y_i & \dot{y}_i \leq 0 \\ -\frac{1}{2\gamma_i}\dot{y}_i^2 + y_i & \dot{y}_i > 0 \end{cases} \\
r_i = 3 &: \Phi_i(\boldsymbol{\xi}_i) = \begin{cases} y_i & \dot{y}_i \leq 0 \ \wedge \ddot{y}_i \leq 0 \\ y_i & \ddot{y}_i - 2\dot{y}_i\gamma_i \leq 0 \\ p_i(\tau_{max}, \boldsymbol{\xi}_i, \gamma_i) & \text{otherwise} \end{cases} \\
&\tau_{max} = -\frac{\ddot{y}_i}{\gamma_i} + \sqrt{\frac{\ddot{y}_i^2}{\gamma_i^2} - 2\frac{\dot{y}_i}{\gamma_i}}
\end{aligned}
\tag{3.8}
$$

On the other hand, for higher relative degrees the invariance function needs to be evaluated by means of a numerical procedure.

In this respect, a numerical approach is here proposed to compute the invariance function for constraints with high relative degrees, based on [58]. The proposed approach directly relies on the output of a root-finding method to determine the real roots of polynomial

$$
\dot{p}_i(\tau, \boldsymbol{\xi}_i, \gamma_i) := \frac{dp_i(\tau, \boldsymbol{\xi}_i, \gamma_i)}{d\tau} \tag{3.9}
$$

required by (3.7).

In order to show an example of such method, we here consider an output function with relative degree $r_i = 4$. The corresponding polynomial $p_i(\tau, \boldsymbol{\xi}_i, \gamma_i)$ is a quartic function, while $\dot{p}_i(\tau, \boldsymbol{\xi}_i, \gamma_i)$ is a cubic function. Let's now denote with $\boldsymbol{\tau}_{roots}$ the vector of real roots of the cubic function $\dot{p}_i(\tau, \boldsymbol{\xi}_i, \gamma_i)$, provided at time instant t by the numerical root-finding method. Note that for a cubic function the cardinality $n(\boldsymbol{\tau}_{roots})$, i.e. the number of real roots, can be either 1 or 3. Therefore, in case $n(\boldsymbol{\tau}_{roots}) = 1$, the only real root is also the maximum of p_i, since $\gamma_i < 0$, and it will be denoted with τ_{max}. Instead, in case $n(\boldsymbol{\tau}_{roots}) = 3$, let's denote with τ_{max1} and τ_{max2} the real roots corresponding to the two maxima of the polynomial function, with $\tau_{max1} < \tau_{max2}$.

Based on the vector of real roots $\boldsymbol{\tau}_{roots}$, together with the state vector $\boldsymbol{\xi}_i$ and the design parameter γ_i, it is now possible to determine the invariance function $\Phi_i(\boldsymbol{\xi}_i)$ according to Algorithm 1.
Fig.3.2 shows an example of the output of Algorithm 1 when $n(\boldsymbol{\tau}_{roots}) = 3$, with $\tau_{max1} < 0$ and $\tau_{max2} > 0$.

Algorithm 1 Output constraint with relative degree $r_i = 4$

Input: $\boldsymbol{\tau}_{roots}, \boldsymbol{\xi}_i, \gamma_i < 0$
Output: $\Phi_i(\boldsymbol{\xi}_i)$

if $\tau_{max} \leq 0$ **then** ▷ $n(\boldsymbol{\tau}_{roots}) = 1$
    $\Phi_i(\boldsymbol{\xi}_i) = y_i$
else
    $\Phi_i(\boldsymbol{\xi}_i) = p_i(\tau_{max}, \boldsymbol{\xi}_i, \gamma_i)$
end if

if $\tau_{max2} \leq 0$ **then** ▷ $n(\boldsymbol{\tau}_{roots}) = 3$
    $\Phi_i(\boldsymbol{\xi}_i) = y_i$
else if $\tau_{max1} \leq 0$ **then**
    $\Phi_i(\boldsymbol{\xi}_i) = max\big(p_i(\tau_{max2}, \boldsymbol{\xi}_i, \gamma_i), y_i\big)$
else
    if $p_i(\tau_{max1}, \boldsymbol{\xi}_i, \gamma_i) \geq p_i(\tau_{max2}, \boldsymbol{\xi}_i, \gamma_i)$ **then**
        $\Phi_i(\boldsymbol{\xi}_i) = p_i(\tau_{max1}, \boldsymbol{\xi}_i, \gamma_i)$
    else
        $\Phi_i(\boldsymbol{\xi}_i) = p_i(\tau_{max2}, \boldsymbol{\xi}_i, \gamma_i)$
    end if
end if

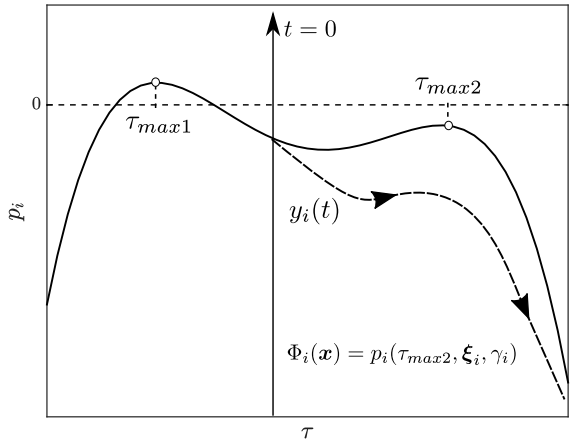


Figure 3.2: Relation of $y(t)$, the quartic polynomial $p_i(\tau, \boldsymbol{\xi}_i, \gamma_i)$ with $\gamma_i < 0$, and the invariance function $\Phi_i(\boldsymbol{\xi}_i)$ returned by Algorithm 1.

It is worth pointing out that the proposed approach can be applied to ar-

bitrary relative degrees by conveniently modifying the numerical procedure in Algorithm 1.

3.2.3 Switching Invariance Control Law

The *admissible set* for the constrained system is the set of m state vectors, for which every invariance function $\Phi_i(\boldsymbol{\xi}_i)$, related to the considered state constraint, takes a non-positive value

$$\mathcal{G} = \{\boldsymbol{x} \in \mathbb{R}^n \mid \Phi_i(\boldsymbol{\xi}_i) \leq 0, 1 \leq i \leq m\} \tag{3.10}$$

The *boundary* $\partial\mathcal{G}$ of set $\mathcal{G}$ is given by all state vectors, for which at least one invariance function is equal to zero and the others take a non-positive value. By defining with $\mathcal{I}(\boldsymbol{x})$ the set of active invariance functions

$$\mathcal{I}(\boldsymbol{x}) = \{i \in \{1, 2, \ldots, m\} \mid \Phi_i(\boldsymbol{\xi_i}) = 0\}. \tag{3.11}$$

the *boundary* $\partial\mathcal{G}$ is given by

$$\partial\mathcal{G} = \{\boldsymbol{x} \in \mathcal{I}(\boldsymbol{x})\} \tag{3.12}$$

As shown in [65, 66], when the state is on the boundary $\partial\mathcal{G}$, the set (3.10) is positively invariant for the controlled system, if each invariance function is decreasing on the boundary, thus

$$\dot{\Phi}_i(\boldsymbol{\xi_i}) := \frac{d\Phi_i(\boldsymbol{\xi_i})}{dt} \leq 0, \quad \forall i \in \mathcal{I}(\boldsymbol{x}) \tag{3.13}$$

This is fulfilled if one of these two conditions holds

$$y_i^{(r)}(\boldsymbol{x}) < 0, \quad 1 \leq r \leq r_i - 1 \tag{3.14}$$

$$\boxed{y_i^{(r_i)}(\boldsymbol{x}, u) \leq \gamma_i} \tag{3.15}$$

By applying I-O Linearization through (3.3) and denoting

$$a_i(\boldsymbol{x}) = \mathcal{L}_{\boldsymbol{g}}\mathcal{L}_{\boldsymbol{f}}^{r_i-1}h_i(\boldsymbol{x}), \; b_i(\boldsymbol{x}) = \mathcal{L}_{\boldsymbol{f}}^{r_i}h_i(\boldsymbol{x}) \tag{3.16}$$

condition (3.15) can be rewritten as

$$a_i(\boldsymbol{x})u + b_i(\boldsymbol{x}) \leq \gamma_i \tag{3.17}$$

Finally, according to [70], compliance with the state constraints is achieved by a switching control action determined by the following constrained minimization problem

$$\begin{aligned} &\min_u \|u - u_{nom}\|_2^2 \\ &\text{s.t. } \boldsymbol{A}_{\mathcal{K}}(\boldsymbol{x})u + \boldsymbol{b}_{\mathcal{K}}(\boldsymbol{x}) \leq \boldsymbol{\gamma} \end{aligned} \tag{3.18}$$

where $\mathcal{K}$ is the set of active constraints

$$\mathcal{K} = \{i \in \{1, 2, \dots, m\} \mid \Phi_i(\boldsymbol{\xi}_i) \geq 0\} \tag{3.19}$$

while $\boldsymbol{A}_{\mathcal{K}}(\boldsymbol{x}) = [a_i(\boldsymbol{x})]$, $\boldsymbol{b}_{\mathcal{K}}(\boldsymbol{x}) = [b_i(\boldsymbol{x})]$ and $\boldsymbol{\gamma} = [\gamma_i]$ with $i \in \mathcal{K}$.
This ensures that the corrective control equals the nominal control u_{nom}, whenever no constraint is active and that the corrective control is as close as possible to the nominal control in the sense of the Euclidean distance.

3.3 Proposed Invariance Control Law

The controller we present here, based on [56, 58], extends the results of invariance control by exploiting set invariance conditions for output regulation purposes. According to the performance requirement addressed in this chapter, see Sec. 2.5, a first control objective is to avoid the definition of the *nominal controller* u_{nom} which, in addition to demanding a control design effort, is not explicitly related with settling performance. Furthermore, output overshoots with respect to the reference signal should be simultaneously prevented.

3.3.1 Regulation Error Constraint

Consider a number of state constraints of the general form (3.2) and their related invariance functions (3.8). The admissible set $\mathcal{G}$ to be made positively invariant is defined according to (3.10).
As proposed in [56], an additional constraint on the regulation error is introduced by means of the output function

$$y_e = h_e(\boldsymbol{x}) \leq 0 \tag{3.20}$$

and its associated invariance function $\Phi_e(\boldsymbol{\xi}_e)$, where $h_e(\boldsymbol{x})$ represents the regulation error for the output y_{out}, i.e.

$$h_e(\boldsymbol{x}) = y_{out} - y_d = h_{out}(\boldsymbol{x}) - y_d \tag{3.21}$$

while

$$\boldsymbol{\xi}_e = \begin{bmatrix} y_e & \dot{y}_e & \dots & y_e^{(r_e-1)} \end{bmatrix}^T \tag{3.22}$$

is the state space of the output dynamics

The aim of the regulation error constraint, which contributes to the definition of the set $\mathcal{G}$, is to prevent output overshoots with respect to the reference signal y_d. It is also worth pointing out that in [56] this constraint has been employed within the original invariance control approach, thus in presence of a nominal controller.

3.3.2 Invariance Function for Output Regulation

Assuming that the output y_e has relative degree $r_e = 2$, the state $\boldsymbol{\xi}_e = 0$ belongs to the boundary $\partial\mathcal{G}$ of the admissible set, see Fig. 3.3. At the same time, since regulation to zero of the error $e = y_{out} - y_d$ is required, $\boldsymbol{\xi}_e = 0$ is also an equilibrium state for the controlled system.

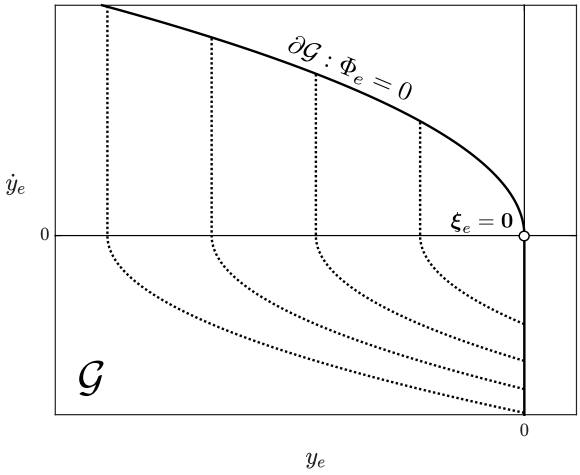


Figure 3.3: Subdomain of the set $\mathcal{G}$ bounded by $\Phi_e(\boldsymbol{\xi}_e) = 0$ and level sets of Lyapunov function $V_e(\boldsymbol{x})$ introduced in (3.23).

Based on these considerations, the main idea behind the proposed approach is to entirely exploit set invariance conditions to drive the system state to the boundary $\partial\mathcal{G}$ and then to the equilibrium $\boldsymbol{\xi}_e = 0$ without leaving $\partial\mathcal{G}$.

In order to obtain a control law with guarantees of stability, the invariance function $\Phi_e(-\boldsymbol{\xi}_e)$ can be considered as a candidate Lyapunov function

$$
\begin{aligned}
&r_e = 1 : V_e(\boldsymbol{x}) := \Phi_e(-\boldsymbol{\xi}_e) = -y_e \\
&r_e = 2 : V_e(\boldsymbol{x}) := \Phi_e(-\boldsymbol{\xi}_e) = \begin{cases} -y_e & \dot{y}_e \geq 0 \\ -\frac{1}{2\gamma_e}\dot{y}_e^2 - y_e & \dot{y}_e < 0 \end{cases}
\end{aligned}
\tag{3.23}
$$

where $\gamma_e < 0$ is the related controller parameter.
Although (3.23) is not positive definite on the whole state space, by limiting the analysis to set $\mathcal{G}$ and assuming that the control action satisfies the set invariance property, the Lyapunov function is positive definite, with $V_e(\boldsymbol{x}) = 0$ in the equilibrium $\boldsymbol{\xi}_e = 0$, see Fig. 3.3.

Its time derivative $\dot{V}_e(\boldsymbol{x})$ is given by

$$\begin{aligned} r_e = 1 &: \dot{V}_e(\boldsymbol{x}) = -\dot{y}_e \\ r_e = 2 &: \dot{V}_e(\boldsymbol{x}) = \begin{cases} -\dot{y}_e & \dot{y}_e \geq 0 \\ -\left(\frac{\ddot{y}_e}{\gamma_e} + 1\right)\dot{y}_e & \dot{y}_e < 0 \end{cases} \end{aligned} \tag{3.24}$$

By requiring $\dot{V}_e(\boldsymbol{x}) \leq 0$ simple Lyapunov stability yields

$$r_e = 1 : \dot{y}_e \geq 0 \tag{3.25a}$$

$$r_e = 2 : \begin{cases} \ddot{y}_e \mid \underline{u} \leq u \leq \overline{u} & \dot{y}_e \geq 0 \\ \ddot{y}_e \geq -\gamma_e & \dot{y}_e < 0 \end{cases} \tag{3.25b}$$

where the control constraints have been additionally considered.

$$\underline{u} \leq u \leq \overline{u} \tag{3.26}$$

For $r_e = 1$, asymptotic stability can be achieved whenever (3.25a) holds as a strict inequality, while imposing $\dot{y}_e = 0$ in the equilibrium.
For $r_e = 2$ however, see (3.25b), it can be noticed that

$$\dot{V}_e(\boldsymbol{x}) = 0, \ \forall y_e \mid \dot{y}_e = 0 \tag{3.27}$$

In order to apply LaSalle's principle the following must be ensured

$$\mathcal{S} \triangleq \{\boldsymbol{\xi}_e(t) \mid \dot{V}_e = 0\} = \{\boldsymbol{\xi}_e(t) = 0, t \geq 0\} \tag{3.28}$$

i.e. $\mathcal{S}$ contains no perturbed trajectory except for the desired equilibrium $\boldsymbol{\xi}_e = 0$. This in turn requires a non-zero $\ddot{y}_e$ whenever $\dot{y}_e = 0$ and $y_e < 0$. For the second inequality of (3.25b), the equality sign can be accepted because LaSalle's principle is again satisfied.
Lyapunov asymptotic stability can therefore be ensured by the following choice of $y_e^{(r_e)}$

$$r_e = 1 : \begin{cases} \dot{y}_e = 0 & y_e = 0 \\ \dot{y}_e > 0 & \text{else} \end{cases} \tag{3.29a}$$

$$r_e = 2 : \begin{cases} \ddot{y}_e \mid \underline{u} \leq u \leq \overline{u} & \dot{y}_e > 0 \\ \ddot{y}_e \geq -\gamma_e & \dot{y}_e < 0 \vee (\dot{y}_e = 0 \wedge y_e < 0) \\ \ddot{y}_e = 0 & \dot{y}_e = 0 \wedge y_e = 0 \end{cases} \tag{3.29b}$$

3.3.3 Selection of Control Actions

As previously stated, the choice of control actions satisfying (3.29) must also ensure positive invariance of $\mathcal{G}$.
Define the set of active invariance functions

$$\mathcal{I}(\boldsymbol{x}) = \{i \in \{1, 2, \dots, m, e\} \mid \Phi_i(\boldsymbol{\xi_i}) = 0\}. \tag{3.30}$$

If both (3.29) and (3.13) hold, asymptotic stability of the desired equilibrium can be achieved along with control invariance of $\mathcal{G}$.

Define the set of pseudo inputs $y_i^{(r_i)}$ with $i \in \{1, 2, \dots, m, e\}$ fulfilling (3.29) and (3.13)

$$\mathcal{Y}(\boldsymbol{x}) = \{y_i^{(r_i)}(u) \mid \dot{V}_e(\boldsymbol{x}) \leq 0 \wedge \Phi_i(\boldsymbol{\xi_i}) \leq 0,\ \forall i \in \mathcal{I}(\boldsymbol{x})\} \tag{3.31}$$

and the corresponding set of feasible stabilizing control actions

$$\mathcal{U}(\boldsymbol{x}) = \{u \mid y_i^{(r_i)} \in \mathcal{Y}(\boldsymbol{x}) \wedge \underline{u} \leq u \leq \overline{u}\} \tag{3.32}$$

Note that $\mathcal{U}(\boldsymbol{x})$ is not empty if the defined constraints are not conflicting and the parameters γ_e and γ_i are chosen so that the control constraints are not violated, i.e. by calculating a priori the control action in the whole set $\mathcal{G}$ and on its boundary.

Finally, according to the settling performance requirement addressed in this chapter, the control action can be chosen in order to maximize convergence speed as follows

$$u \mid \{\overline{y}_i^{(r_i)} = \max \mathcal{Y}(\boldsymbol{x})\} \wedge \underline{u} \leq u \leq \overline{u} \tag{3.33}$$

which is given by either $\max \mathcal{U}(\boldsymbol{x})$ or $\min \mathcal{U}(\boldsymbol{x})$ depending on the sign of $\mathcal{L}_{\boldsymbol{g}} \mathcal{L}_{\boldsymbol{f}}^{r_e - 1} h_e(\boldsymbol{x})$.

Again, it is worth pointing out that, compared to the original invariance control, the controller presented here does not require anymore the definition of the nominal controller u_{nom}. Fast convergence speed and absence of output overshoots with respect to the reference are in fact achieved by entirely exploiting set invariance conditions (3.13) and Lyapunov stability conditions (3.29), where the only parameters that require tuning are γ_i, γ_e.

Finally note that the resulting control law produces a bang-bang switching of the pseudo control input $\overline{y}_e^{(r_e)}$ to the I-O linearized system with state $\boldsymbol{\xi}_e$. Nonetheless, bang-bang control is a well known solution to the time-optimal control problem for a linear system with bounded control by applying Pontryagin's maximum principle (PMP) [82] or Bellman's principle of

optimality [83]. Consequently, if the number of switches is no larger than the system order, the proposed controller can further achieve minimum-time output tracking.

3.3.4 Extension to MIMO Systems

A formal extension of the proposed controller to Multi-Input Multi-Output (MIMO) systems under multiple state and input constraints is provided in the following.

Consider the nonlinear control affine system of order n with m inputs

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \sum_{k=1}^{m} \boldsymbol{g}_k(\boldsymbol{x}) u_k \tag{3.34}$$

where $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{u} \in \mathbb{R}^m$, smooth vector fields $\boldsymbol{f}, \boldsymbol{g}_k : \mathbb{R}^n \to \mathbb{R}^n$, and $\boldsymbol{y}_{out} \in \mathbb{R}^p$ are outputs that need to be regulated to a reference $\boldsymbol{y}_d \in \mathbb{R}^p$.

For each of these outputs their respective regulation error constraint (3.20), invariance function and associated Lyapunov function (3.23) are defined

$$\begin{aligned} &y_{e,j} = h_{e,j}(\boldsymbol{x}) \leq 0 \\ &\Phi_{e,j}(\boldsymbol{\xi}_{e,j}) \leq 0 \qquad\qquad 1 \leq j \leq p \\ &V_{e,j}(\boldsymbol{x}) = \Phi_{e,j}(-\boldsymbol{\xi}_{e,j}) \end{aligned} \tag{3.35}$$

According to (3.30), a set $\mathcal{Y}_{e,j}$ can be computed for each output, satisfying the control constraints, the Lyapunov conditions, and the invariance condition for that specific output.

For each set, the input of the linearized system, that maximizes settling performance, is chosen as follows

$$\overline{y}_{e,j}^{(r_{e,j})} = \max \mathcal{Y}_{e,j}(\boldsymbol{x}), \;\; 1 \leq j \leq p \tag{3.36}$$

Therefore, the control vector $\boldsymbol{u}$ can be determined by solving the following constrained minimization problem

$$\begin{aligned} \min_{\boldsymbol{u}} \; & \sum_{j=1}^{p} \| y_{ej}^{(r_{ej})} - \overline{y}_{ej}^{(r_{ej})} \|_2^2 \\ \text{s.t.} \quad & y_{ej}^{(r_{ej})} \in \mathcal{Y}_{ej}(\boldsymbol{x}), \;\; 1 \leq j \leq p \\ & \dot{\Phi}_i(\boldsymbol{\xi}_i) \leq 0, \;\; \forall i \in \mathcal{I}(\boldsymbol{x}) \end{aligned} \tag{3.37}$$

where the second row of constraints takes into account the invariance conditions that arise from the generic state and output constraints (3.2).
Note that the system is control affine as a result of I-O Linearization

$$\begin{aligned} & y_i^{(r_i)} = \boldsymbol{a}_i(\boldsymbol{x})^T \boldsymbol{u} + b_i(\boldsymbol{x}) \\ & \text{with } \boldsymbol{a}_i(\boldsymbol{x})^T = \left[\mathcal{L}_{\boldsymbol{g}_1}\mathcal{L}_{\boldsymbol{f}}^{r_i-1} h_i(\boldsymbol{x}) \ \dots \ \mathcal{L}_{\boldsymbol{g}_m}\mathcal{L}_{\boldsymbol{f}}^{r_i-1} h_i(\boldsymbol{x})\right] \\ & \text{and } b_i(\boldsymbol{x}) = \mathcal{L}_{\boldsymbol{f}}^{r_i} h_i(\boldsymbol{x}) \end{aligned} \tag{3.38}$$

Therefore, the optimization problem is linear in the control variables and can be efficiently handled by state-of-the-art Quadratic Programming (QP) solvers, e.g. [84].

3.4 Application to Robot Implicit Force Control

In order to apply the controller presented above to robot implicit force control, the following assumptions are made:

Assumption 1: Since the proposed control law results in a switching control action, in order to avoid feeding the inner robot position controller with a switching reference, the selected control action $\boldsymbol{u}$ corresponds to the reference joint acceleration vector $\ddot{\overline{\boldsymbol{q}}} \in \mathbb{R}^6$, instead of the reference joint positions $\overline{\boldsymbol{q}} \in \mathbb{R}^6$.

Assumption 2: The force control bandwidth is well inside the robot joint position control bandwidth. Therefore, the position control dynamics and the second order low-pass input filter on the joint reference signal, see Sec. 2.2, will be neglected, i.e.

$$\overline{\boldsymbol{q}} \approx \boldsymbol{q} \tag{3.39}$$

where $\boldsymbol{q} \in \mathbb{R}^6$ are the actual joint positions.

Based on these assumptions the system reduces to a double integrator

$$\begin{cases} \frac{d\overline{\boldsymbol{q}}}{dt} = \dot{\overline{\boldsymbol{q}}} \\ \frac{d\dot{\overline{\boldsymbol{q}}}}{dt} = \ddot{\overline{\boldsymbol{q}}} = \boldsymbol{u} \\ \boldsymbol{y}_{out} = \begin{bmatrix} F_c & \boldsymbol{x}_{uc} \end{bmatrix}^T \end{cases} \tag{3.40}$$

where $F_c \in \mathbb{R}$ is the interaction force in the direction elastically constrained by the environment, while $\boldsymbol{x}_{uc} \in \mathbb{R}^5$ is the end effector position and orientation in the unconstrained directions.

Force regulation error constraint: In order to fulfill the performance requirement addressed in this chapter, force overshoots with respect to the reference signal $\overline{F}_c$ can be prevented by introducing a regulation error constraint described by the output function

$$y_{e,c} = F_c - \overline{F}_c \leq 0 \tag{3.41}$$

According to Assumption 2 the following relation exists between the reference and the actual end effector displacements in the constrained direction $\Delta\overline{x}_c$ and Δx_c, respectively

$$\Delta\overline{x}_c = x_c(\overline{\boldsymbol{q}}) - x_0 \approx \Delta x_c = x_c(\boldsymbol{q}) - x_0 \tag{3.42}$$

Thus, employing the compliant contact model in (2.8)

$$F_c = -\frac{\Delta x_c}{C_T(\boldsymbol{q})} \approx -\frac{\Delta\overline{x}_c}{C_T(\boldsymbol{q})} \tag{3.43}$$

where $C_T(\boldsymbol{q})$ is the total compliance introduced in Sec. 2.3.

According to [56], the output constraint in (3.20) can be alternatively expressed with respect to the reference end effector displacement $\Delta\overline{x}_c$, as follows

$$\boxed{y_{e,c} = -\Delta\overline{x}_c - C_T(\boldsymbol{q})\overline{F}_c \leq 0} \tag{3.44}$$

As a consequence, since $\Delta\overline{x}_c$ depends on $\overline{\boldsymbol{q}}$, the output relative degree $r_{e,c}$ is equal to 2 and the corresponding state of the output dynamics is given by

$$\boldsymbol{\xi}_{e,c} = \begin{bmatrix} y_{e,c} & \dot{y}_{e,c} \end{bmatrix}^T \tag{3.45}$$

Considering a typical industrial force control task, for which change in the environment stiffness and big modifications in the robot configuration are unlikely, $C_T(\boldsymbol{q})$ is hereafter assumed to be constant. Therefore, the corresponding invariance function $\Phi_{e,c}(\boldsymbol{\xi}_{e,c})$ and its time derivative are given by

$$\Phi_{e,c}(\boldsymbol{\xi}_{e,c}) = \begin{cases} -\Delta\overline{x}_c - C_T(\boldsymbol{q})\overline{F}_c & \dot{\overline{x}}_c \geq 0 \\ -\frac{1}{2\gamma_{e,c}}\dot{\overline{x}}_c^2 - \Delta\overline{x}_c - C_T(\boldsymbol{q})\overline{F}_c & \dot{\overline{x}}_c < 0 \end{cases} \tag{3.46}$$

$$\dot{\Phi}_{e,c}(\boldsymbol{\xi}_{e,c}) = \begin{cases} -\dot{\overline{x}}_c & \dot{\overline{x}}_c \geq 0 \\ -\left(\frac{\ddot{\overline{x}}_c}{\gamma_{e,c}} + 1\right)\dot{\overline{x}}_c & \dot{\overline{x}}_c < 0 \end{cases} \tag{3.47}$$

where $\gamma_{e,c}$ is the corresponding control parameter, while

$$\dot{\overline{x}}_c = \boldsymbol{n}^T \boldsymbol{J}(\overline{\boldsymbol{q}})\dot{\overline{\boldsymbol{q}}} \tag{3.48}$$

Note that in this application, the selection of parameter $\gamma_{e,c}$ is simplified by its kinematical meaning, i.e. the minimum value of acceleration that the robot end effector should have in the constrained direction to avoid constraint violation ($F_c > \overline{F}_c$) when $\Phi_{e,c}(\boldsymbol{\xi}_{e,c}) = 0$ and $\dot{\overline{x}}_c < 0$.

The pseudo control input $\ddot{\overline{y}}_{e,c}$ of the double integrator chain with state $\boldsymbol{\xi}_{e,c}$ can be computed according to (3.36)

$$\ddot{\overline{y}}_{e,c} = \max \mathcal{Y}_{e,c} \tag{3.49}$$

It is worth pointing out that the I-O Linearization equation (3.38), relating the control variable $\boldsymbol{u} = \ddot{\overline{\boldsymbol{q}}}$ to the pseudo input $\ddot{y}_{e,c}$, is here represented by the well known robot kinematics relationship

$$\ddot{y}_{e,c} = -\ddot{\overline{x}}_c = -\boldsymbol{n}^T \left(\dot{\boldsymbol{J}}(\overline{\boldsymbol{q}})\dot{\overline{\boldsymbol{q}}} + \boldsymbol{J}(\overline{\boldsymbol{q}})\boldsymbol{u} \right) \tag{3.50}$$

According to the hybrid force-motion control paradigm, position control of the end effector unconstrained directions is achieved by means of the following closed-loop inverse kinematic constraint

$$\boldsymbol{y}_{e,uc} = \ddot{\boldsymbol{e}} + \boldsymbol{K}_D\, \dot{\boldsymbol{e}} + \boldsymbol{K}_P\, \boldsymbol{e} = 0, \;\; \boldsymbol{e} = \overline{\boldsymbol{x}}_{uc} - \boldsymbol{x}_{uc} \tag{3.51}$$

where $\boldsymbol{y}_{e,uc} \in \mathbb{R}^5$, $\boldsymbol{K}_P, \boldsymbol{K}_D \in \mathbb{R}^{5\times 5}$ are positive definite gain matrices, while $\overline{\boldsymbol{x}}_{uc} \in \mathbb{R}^5$ is the desired end effector position in the unconstrained directions. Because constraint (3.51) is expressed at acceleration level, $r_{e,uc} = 0$ and therefore $\overline{\boldsymbol{y}}_{e,uc} = 0$.

Position, velocity and acceleration constraints in both the joint and the Cartesian space, together with joint torque limits [79], can be considered as additional state constraints, with their related invariance functions $\Phi_i(\boldsymbol{\xi}_i)$. By denoting with $\mathcal{I}(\boldsymbol{\xi}_i)$ the set of active invariance functions

$$\mathcal{I}(\boldsymbol{\xi}_i) = \{ i \in \{1, 2, \ldots, m, e_c\} \mid \Phi_i(\boldsymbol{\xi}_i) = 0 \} \tag{3.52}$$

the resulting constrained optimization problem is given by

$$\begin{aligned} &\min_{\boldsymbol{u}=\ddot{\overline{\boldsymbol{q}}}} \|\ddot{y}_{e,c} - \ddot{\overline{y}}_{e,c}\|_2^2 + \|\boldsymbol{y}_{e,uc} - \overline{\boldsymbol{y}}_{e,uc}\|_2^2 \\ &\text{s.t.} \qquad\qquad \ddot{y}_{e,c} \in \mathcal{Y}_{e,c} \\ &\qquad\qquad \dot{\Phi}_i(\boldsymbol{\xi}_i) \leq 0, \;\; \forall i \in \mathcal{I}(\boldsymbol{x}) \end{aligned} \tag{3.53}$$

Once the optimal control variables $\ddot{\overline{\boldsymbol{q}}}$ have been determined, the desired robot joint positions $\overline{\boldsymbol{q}}$ are computed via double integration and fed to the inner robot position control loop.

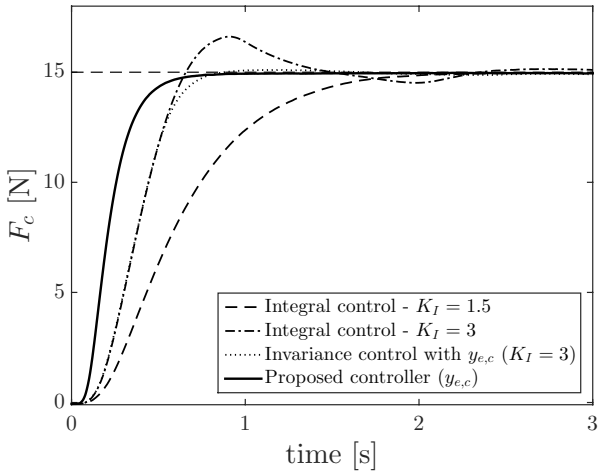


Figure 3.4: Experimental results with the force regulation error constraint in (3.44): comparison among the integral controller [50] (dashed and dashed-dotted lines), invariance control with force regulation error constraint [56] (dotted line) and the presented controller [57] (solid line).

3.4.1 Experimental Results

Here, an experimental comparison is presented among the original invariance control approach combined with the force regulation error constraint in (3.44), as proposed in [56], the presented controller [57] and the integral controller introduced in Sec. 2.4, based on [50].

The force regulation experiments, see Fig. 3.4, have been performed in presence of a step reference signal $\overline{F}_c$ of $15\ N$ on the same environment setup described in Sec. 2.5. The C QP library qpOASES [84] has been exploited to carry out the optimization problem in (3.53), while the environment compliance has been estimated off-line by means of Least Square Estimation (LSE), see Sec. 2.3.

The nominal control law for the invariance control approach in [56], corresponds to the integral controller in [50] with integral gain $K_I = 3$. As it can be noticed from Fig. 3.4, while the force initial transient is equal to the one produced by the pure integral control with $K_I = 3$, the occurrence of overshoots with respect to the reference signal is effectively prevented by the regulation error constraint $y_{e,c}$. On the other hand, the settling performance with [56] is inherently limited by the selection of the nominal controller. As a matter of fact, by employing the proposed control law a considerable decrease in the settling time can be achieved, again prevent-

ing the occurrence of force overshoots.

3.5 Robustness to Compliance Uncertainty

The dependency of the output function $y_{e,c}$ on the estimated system compliance $\hat{C}_T(\boldsymbol{q})$, according to the formulation in (3.44), is the main source of fragility of the proposed controller.

The relation between the equilibrium state $\boldsymbol{\xi}_{e,c} = 0$ and the estimated compliance is given by

$$\boldsymbol{\xi}_{e,c} = \begin{bmatrix} y_{e,c} \\ \dot{y}_{e,c} \end{bmatrix} = 0 \implies \begin{cases} \Delta\overline{x}_c = -\hat{C}_T(\boldsymbol{q})\ \overline{F}_c \\ \dot{\overline{x}}_c = 0 \end{cases} \tag{3.54}$$

Therefore, by employing the compliant contact model in (2.8), the corresponding value of the interaction force at the equilibrium, i.e.

$$F_c|_{\boldsymbol{\xi}_{e,c}=0} = \frac{\hat{C}_T(\boldsymbol{q})}{C_T(\boldsymbol{q})}\ \overline{F}_c \tag{3.55}$$

is directly affected by uncertainties in the total compliance. The integral controller [50] conversely, ensures zero steady-state regulation error.

As a consequence, assuming an accurate estimation of the robot joint and link compliance matrices $\hat{C}_J(\boldsymbol{q})$ and $\hat{C}_L(\boldsymbol{q})$, respectively, a rough estimation of the environment compliance $\hat{C}_E$ will eventually produce a mismatch between the steady state value of the measured force and the desired force $\overline{F}_c$. An under-estimate in the environment stiffness results in an increased system compliance, i.e. $\hat{C}_T(\boldsymbol{q}) > C_T(\boldsymbol{q})$, and thus in a higher value for the steady state force. Conversely, an over-estimate results in a lower value for the steady state force with respect to the reference.

In other to circumvent this limitation an adaptive approach is first presented in the following section. Alternatively, the formulation of the force regulation error constraint given in (3.41) is employed in Sec. 3.7 and conveniently modified in order to improve controller robustness.

3.6 Adaptive Force Constraint

An adaptive specification of the force regulation error constraint in (3.44) can be easily obtained by employing an on-line estimation of the system compliance $\hat{C}_T(\boldsymbol{q})$ within the definition of the corresponding output function, i.e.

$$y_{e,c} = -\Delta\overline{x}_c - \hat{C}_T(\boldsymbol{q})\overline{F}_c \leq 0 \tag{3.56}$$

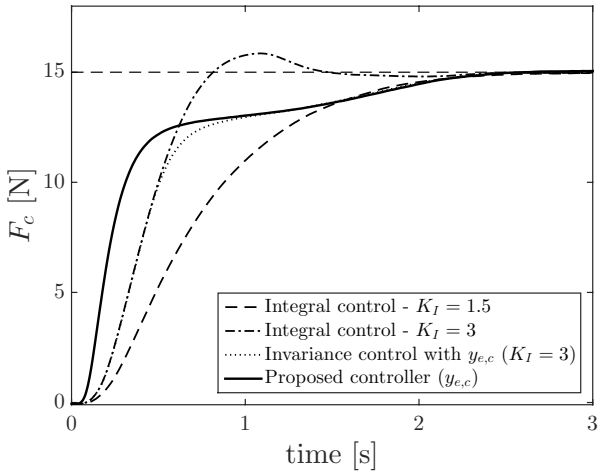


Figure 3.5: Experimental results with the adaptive constraint specification in (3.56): comparison among the integral controller [53] (dashed and dashed-dotted lines), invariance control with force regulation error constraint [56] (dotted line) and the presented controller [57] (solid line).

According to Sec. 2.3, it is hereafter assumed the robot joint and link compliance matrices $\hat{C}_J(\boldsymbol{q})$ and $\hat{C}_L(\boldsymbol{q})$, respectively, to be estimated off-line, while an Exponentially Weighted Recursive Least Square (EWRLS) algorithm is exploited, as proposed in [53], for the real-time update of the environment compliance $\hat{C}_E$, see Equation (2.10).

3.6.1 Experimental Results

The same experimental comparison described in Sec. 3.4.1 has been performed employing the adaptive output constraint in (3.56). A forgetting factor μ equal to 0.995 has been selected for the EWRLS algorithm, while the environment stiffness has been initialized to an approximate 20% overestimate of the real stiffness.

As it can be noticed from Fig. 3.5, although the introduced adaptive specification now guarantees zero-error regulation, settling performance is instead inherently affected by the estimator dynamics, eventually leading to a decreased convergence speed compared to the one achieved in Sec. 3.4. As a matter of fact, both [56] and [57] effectively prevent force overshoots but result in a higher settling time, comparable to the one achieved by an integral controller with $K_I = 1.5$. Nevertheless note that, if a new force step reference was subsequently applied, the controller performance shown

in Fig. 3.4 would be successfully recovered.

A relevant issue, related to the adaptive formulation in (3.56), is that sudden variations on the estimated compliance, mainly due to surface irregularities and noise on the measured force, directly produce expansions and/or contractions of the portion of the set $\mathcal{G}$ bounded by $\Phi_{e,c}(\boldsymbol{\xi}_{ec}) = 0$. As a consequence of the contraction of $\mathcal{G}$, the system state may end up outside the set, thus resulting in a constraint violation and compromising control invariance. Therefore, a possible strategy to handle these situations, occurring when $\Phi_{e,c}(\boldsymbol{\xi}_{ec}) > 0$, is to employ a recovery procedure by redefining the force regulation error constraint in (3.56) as follows

$$y_{e,c} = \Delta\overline{x}_c + \hat{C}_T(\boldsymbol{q})\overline{F}_c \leq 0. \tag{3.57}$$

This way the state will be driven back towards the interior of the set $\mathcal{G}$ and then to the desired equilibrium.

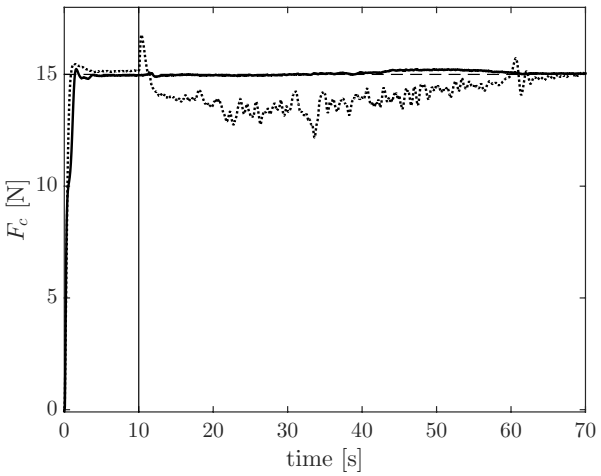


Figure 3.6: Force responses from the experiments performed with assigned robot motion on a plastic (solid line) and plywood (dotted line) surface. During the initial $10\ s$ the contact force is stabilized and then a $50\ s$ movement of $0.1\ m$ is performed on the $x-y$ plane.

An additional experiment has been performed involving the execution of a linear path on a soft plastic box ($K_E \approx 10^4 N/m$) and on a stiff and rough plywood surface ($K_E \approx 10^6 N/m$). In the first part of the experiment, the contact force along the z Cartesian direction is stabilized to a reference value of $15\ N$, afterwards a linear motion of $0.1\ m$ is executed on the $x-y$

plane, while regulating the contact force. Again, the EWRLS algorithm has been initialized with an over-estimate of the environment stiffness for both materials.
Besides highlighting the validity of the proposed controller during a motion task in contact with a surface, Fig. 3.6 shows the effectiveness of the introduced recovery procedure in handling constraint violations. This is particularly evident in the experiment performed on the plywood surface (solid line) where force peaks occur, corresponding to grooves on the surface.

3.7 Force Constraint Modification

An alternative approach, based on [58], consists in relaxing Assumption 2 in Sec. 3.4 and employing the force regulation error constraint in (3.41), i.e.

$$\boxed{y_{e,c} = F_c - \overline{F}_c \leq 0}$$

which directly depends on the measured force F_c.
As a consequence of the relaxation of Assumption 2, the relation between the actual joint position $\boldsymbol{q}$ and the reference joint position $\overline{\boldsymbol{q}}$ is given by

$$\boldsymbol{q} = h(s)\, \overline{\boldsymbol{q}} \tag{3.58}$$

where $h(s)$ is the closed-loop transfer function of the inner position control loop, introduced in Sec. 2.1

$$h(s) = \frac{\omega_n^2}{s^2 + 2\zeta_n\omega_n s + \omega_n^2}.$$

being ω_n and ζ_n the position control bandwidth and damping ratio, respectively. Subsequently, the following relation exists between the reference and the actual end effector displacements in the constrained direction $\Delta\overline{x}_c = x_c(\overline{\boldsymbol{q}}) - x_0$ and $\Delta x_c = x_c(\boldsymbol{q}) - x_0$, respectively

$$\Delta\overline{x}_c = h(s)\, \Delta x_c \tag{3.59}$$

Note that (3.58) and (3.59) can be expressed in the time domain as follows

$$\begin{aligned} \omega_n^2\overline{\boldsymbol{q}} &= \ddot{\boldsymbol{q}} + 2\zeta_n\omega_n\dot{\boldsymbol{q}} + \omega_n^2\boldsymbol{q} \\ \omega_n^2\Delta\overline{x}_c &= \ddot{x}_c + 2\zeta_n\omega_n\dot{x}_c + \omega_n^2\Delta x_c \end{aligned} \tag{3.60}$$

where $\dot{x}_c = \boldsymbol{n}^T\boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$ and $\ddot{x}_c = \boldsymbol{n}^T\big(\dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}}\big)$.
Based on (3.60) and considering as control actions $\boldsymbol{u}$ the reference joint

accelerations $\ddot{\overline{\boldsymbol{q}}}$, according to Assumption 1, the following 4-th order system is obtained

$$\begin{cases} \frac{d\boldsymbol{q}}{dt} = \dot{\boldsymbol{q}} \\ \frac{d\dot{\boldsymbol{q}}}{dt} = \omega_n^2 \overline{\boldsymbol{q}} - \omega_n^2 \boldsymbol{q} - 2\zeta_n \omega_n \dot{\boldsymbol{q}} \\ \frac{d\ddot{\boldsymbol{q}}}{dt} = \omega_n^2 \dot{\overline{\boldsymbol{q}}} - \omega_n^2 \dot{\boldsymbol{q}} - 2\zeta_n \omega_n \frac{d\dot{\boldsymbol{q}}}{dt} \\ \frac{d}{dt}\boldsymbol{q}^{(3)} = \omega_n^2 \overline{\boldsymbol{u}} - \omega_n^2 \frac{d\dot{\boldsymbol{q}}}{dt} - 2\zeta_n \omega_n \frac{d\ddot{\boldsymbol{q}}}{dt} \\ \boldsymbol{y}_{out} = \begin{bmatrix} F_c & \boldsymbol{x}_{uc} \end{bmatrix}^T \end{cases} \tag{3.61}$$

Since F_c depends on $\boldsymbol{q}$ through the compliant contact model in (2.8), the relative degree $r_{e,c}$ of the output function $y_{e,c}$ in (3.41) is now equal to 4. Accordingly the corresponding state of the output dynamics $\boldsymbol{\xi}_{e,c}$ is described by the vector of the 4 integrators chain, whose terms can be obtained by exploiting the contact model in (2.8) and (3.60), leading to

$$\boldsymbol{\xi}_{e,c} = \begin{bmatrix} y_{e,c} \\ \dot{y}_{e,c} \\ \ddot{y}_{e,c} \\ y_{e,c}^{(3)} \end{bmatrix} = \begin{bmatrix} F_c - \overline{F}_c \\ -\frac{\dot{x}_c}{C_T(\boldsymbol{q})} \\ -\frac{\omega_n^2 \Delta\overline{x}_c - \omega_n^2 \Delta x_c - 2\zeta_n \omega_n \dot{x}_c}{C_T(\boldsymbol{q})} \\ -\frac{\omega_n^2 \dot{\overline{x}}_c - \omega_n^2 \dot{x}_c - 2\zeta_n \omega_n (\omega_n^2 \Delta\overline{x}_c - \omega_n^2 \Delta x_c - 2\zeta_n \omega_n \dot{x}_c)}{C_T(\boldsymbol{q})} \end{bmatrix} \tag{3.62}$$

Furthermore, since $r_{e,c} = 4$, the numerical procedure proposed in Sec. 3.2.2, see Algorithm 1, needs to be exploited in order to compute the corresponding invariance function $\Phi_{e,c}(\boldsymbol{\xi}_{e,c})$.

Unlike the force regulation constraint in (3.44), see Sec. 3.5, now the equilibrium state $\boldsymbol{\xi}_{e,c} = 0$ is independent of $C_T(\boldsymbol{q})$

$$\boldsymbol{\xi}_{e,c} = \begin{bmatrix} y_{e,c} \\ \dot{y}_{e,c} \\ \ddot{y}_{e,c} \\ y_{e,c}^{(3)} \end{bmatrix} = 0 \implies \begin{cases} F_c = \overline{F}_c \\ \dot{x}_c = 0 \\ \Delta\overline{x}_c = \Delta x_c \\ \dot{\overline{x}}_c = 0 \end{cases} \tag{3.63}$$

This ensures that the value of the interaction of force at the equilibrium $F_c|_{\boldsymbol{\xi}_{e,c}=0}$ will be equal to the desired value $\overline{F}_c$ despite uncertainties in the system compliance.

The pseudo control input $\overline{y}_{eF}^{(4)}$ of the 4 integrators chain with state $\boldsymbol{\xi}_{e,c}$ can be computed according to (3.30)

$$\overline{y}_{e,c}^{(4)} = \max \mathcal{Y}_{e,c} \tag{3.64}$$

while, following from (3.62), the I-O Linearization equation is given by

$$y_{e,c}^{(4)} = -\frac{\omega_n^2}{C_T(\boldsymbol{q})}\ddot{\overline{x}}_c - \omega_n^2 \ddot{y}_{e,c} - 2\zeta_n \omega_n y_{e,c}^{(3)} \tag{3.65}$$

where the complete expression of $\ddot{y}_{e,c}$ and $y_{e,c}^{(3)}$ can be found in (3.62), while $\ddot{\overline{x}}_c = \boldsymbol{n}^T\big(\dot{\boldsymbol{J}}(\overline{\boldsymbol{q}})\dot{\overline{\boldsymbol{q}}} + \boldsymbol{J}(\overline{\boldsymbol{q}})\boldsymbol{u}\big)$.

The resulting constrained optimization problem

$$\begin{aligned} &\min_{\boldsymbol{u}=\ddot{\overline{\boldsymbol{q}}}} \|y_{e,c}^{(4)} - \overline{y}_{e,c}^{(4)}\|_2^2 + \|\boldsymbol{y}_{e,uc} - \overline{\boldsymbol{y}}_{e,uc}\|_2^2 \\ &\text{s.t.} \qquad\quad \ddot{y}_{e,c} \in \mathcal{Y}_{e,c} \\ &\qquad\qquad \dot{\Phi}_i(\boldsymbol{\xi}_i) \leq 0, \;\; \forall i \in \mathcal{I}(\boldsymbol{x}) \end{aligned} \tag{3.66}$$

returns the optimal control variables $\ddot{\overline{\boldsymbol{q}}}$, from which the reference robot positions are obtained via double integration and fed to the inner robot joint position controller.

Finally note that, the controller parameter $\gamma_{e,c}$ can be selected as follows

$$\gamma_{e,c} \geq y_{e,c}^{(4)}, \;\; \forall y_{e,c}^{(4)} \in Y_{e,c}^{(4)} \tag{3.67}$$

where $Y_{e,c}^{(4)}$ is the set of $y_{e,c}^{(4)}$ computed through (3.65) for all the admissible values of $\Delta x_c, \Delta \dot{x}_c, \Delta \overline{x}_c, \Delta \dot{\overline{x}}_c, C_T(\boldsymbol{q}), \ddot{\overline{x}}_c$.

3.7.1 Bouncing Reduction

Let's assume that for some time instant $\tau = \tau_0$ the system state lies on the boundary of the set $\mathcal{G}$, so that the following equality holds

$$p_{e,c}\big(\tau_0, \boldsymbol{\xi}_{e,c}, \gamma_{e,c}\big) = \Phi_{e,c}(\boldsymbol{\xi}_{e,c}) = 0 \tag{3.68}$$

where $p_{e,c}\big(\tau_0, \boldsymbol{\xi}_{e,c}, \gamma_{e,c}\big)$ is the Taylor series that upper bounds the output function $y_{e,c}$, see Sec. 3.2.1.
Based on the definition of invariance function in (3.7) and further assuming $y_{e,c}^{(4)} = \gamma_{e,c}$, the equality in (3.68) yields

$$\begin{aligned} y_{e,c}(t+\tau_0) &= p_{e,c}\big(\tau_0, \boldsymbol{\xi}_{e,c}, \gamma_{e,c}\big) = 0 \\ \dot{y}_{e,c}(t+\tau_0) &= \dot{p}_{e,c}\big(\tau_0, \boldsymbol{\xi}_{e,c}, \gamma_{e,c}\big) = 0 \end{aligned} \tag{3.69}$$

but also

$$\begin{aligned} \ddot{y}_{e,c}(t+\tau_0) &= \ddot{p}_{e,c}\big(\tau_0, \boldsymbol{\xi}_{e,c}, \gamma_{e,c}\big) \neq 0 \\ y_{e,c}^{(3)}(t+\tau_0) &= p_{e,c}^{(3)}\big(\tau_0, \boldsymbol{\xi}_{e,c}, \gamma_{ec,}\big) \neq 0 \end{aligned} \tag{3.70}$$

Subsequently, the following holds for the state space of the output dynamics

$$\boldsymbol{\xi}_{e,c}(t+\tau_0) = \begin{bmatrix} y_{e,c}(t+\tau_0) \\ \dot{y}_{e,c}(t+\tau_0) \\ \ddot{y}_{e,c}(t+\tau_0) \\ y^{(3)}_{e,c}(t+\tau_0) \end{bmatrix} \neq 0 \tag{3.71}$$

As a matter of fact, instead of reaching the equilibrium $\boldsymbol{\xi}_{e,c} = 0$, the system state is driven inside the interior of the set $\mathcal{G}$ immediately after touching the boundary $\partial\mathcal{G}$, coherently with the set positive invariance (3.15).

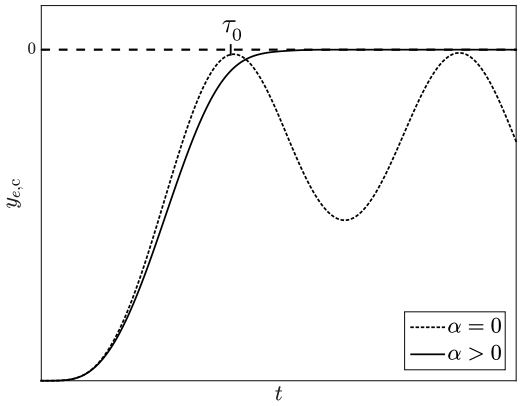


Figure 3.7: Simulated time evolution of the output function $y_{e,c}$: bouncing effect (dotted line) and bouncing reduction within the subset $\hat{\mathcal{G}}$ (solid line).

As shown by the dotted line in Fig. 3.7, this eventually causes the output function $y_{e,c}$ to repeatedly bounce off the zero-level curve. This is a well known result in polynomial trajectory generation [85]: assuming τ_0 to be the final duration of a trajectory that drives the state of the output dynamics to $\boldsymbol{\xi}_{e,c}(t+\tau_0) = 0$, a polynomial $p_{e,c}\big(\tau_0, \boldsymbol{\xi}_{e,c}, \gamma_{e,c}\big)$ with relative degree equal to 7 (instead of 4) would in fact be required. On the other hand, this behavior does not occur with the force regulation constraint in (3.44) since $\ddot{y}_{e,c}$, which directly relates to the control variable through equation (3.50), can be instantly set to 0.

Aiming at reducing this undesired effect arising when the set invariance condition in [65, 66] is employed for output regulation with high relative degrees, a subset $\hat{\mathcal{G}}$ of the constraint set $\mathcal{G}$ is introduced in order to bound the higher order derivatives of $y_{e,c}$ in (3.70) that cause the bouncing behavior.

This approach resembles the one proposed in [86] to replace the switching of the invariance control modes by a continuous transition.

The boundary layer $\hat{\mathcal{G}}$ on the inside of $\partial\mathcal{G}$ can be specified by introducing a *virtual* state space of the output dynamics $\hat{\boldsymbol{\xi}}_{e,c}$, derived from the actual state $\boldsymbol{\xi}_{e,c}$.

$$\hat{\mathcal{G}} = \{\hat{\boldsymbol{\xi}}_{e,c} := \alpha(\boldsymbol{\xi}_{e,c}) \mid \Phi_{e,c}(\boldsymbol{\xi}_{e,c}) < \Phi_{e,c}(\hat{\boldsymbol{\xi}}_{e,c}) = 0\} \tag{3.72}$$

In order to determine $\hat{\boldsymbol{\xi}}_{e,c}$ so that (3.72) holds, useful upper bounds on the polynomial roots of $\dot{p}_{e,c}\big(\tau, \boldsymbol{\xi}_{e,c}, \gamma_{e,c}\big)$ can be exploited, such as the Fujiwara bound [87], briefly recalled in the following.

Fujiwara bound: Given a polynomial in the generic form

$$p(x) = a_0 + a_1 x + \cdots + a_n x^n \tag{3.73}$$

where $a_0, a_1, \ldots, a_n$ are complex numbers, the Fujiwara upper bound $\overline{x}$ for the magnitudes of all polynomial's roots is given by

$$\overline{x} = 2\max\left\{\left|\frac{a_{n-1}}{a_n}\right|, \left|\frac{a_{n-2}}{a_n}\right|^{\frac{1}{2}}, \ldots, \left|\frac{a_1}{a_n}\right|^{\frac{1}{n-1}}, \left|\frac{a_0}{a_n}\right|^{\frac{1}{n}}\right\} \tag{3.74}$$

Let's now consider the time derivative of the polynomial function related to the output function $y_{e,c}$ required for the compuation of $\Phi_{e,c}(\boldsymbol{\xi}_{e,c})$

$$\dot{p}_{e,c}\big(\tau, \boldsymbol{\xi}_{e,c}, \gamma_{e,c}\big) = \dot{y}_{e,c} + \ddot{y}_{e,c}\tau + \frac{y^{(3)}_{e,c}}{2}\tau^2 + \frac{\gamma_{e,c}}{6}\tau^3 \tag{3.75}$$

Exploiting (3.74), the Fujiwara upper bound $\overline{\tau}$ of the roots of $\dot{p}_{e,c}\big(\tau, \boldsymbol{\xi}_{e,c}, \gamma_{e,c}\big)$ is given by

$$\overline{\tau} = 2\max\left\{\left|\frac{3y^{(3)}_{e,c}}{\gamma_{e,c}}\right|, \left|\frac{6\ddot{y}_{e,c}}{\gamma_{e,c}}\right|^{\frac{1}{2}}, \left|\frac{6\dot{y}_{e,c}}{\gamma_{e,c}}\right|^{\frac{1}{3}}\right\} \tag{3.76}$$

Note that in order to fulfill (3.72), the choice of the *virtual* state space $\hat{\boldsymbol{\xi}}_{e,c}$ needs to ensure the Fujiwara bound $\hat{\tau}$ of the roots of the polynomial $\dot{p}_{e,c}\big(\tau, \hat{\boldsymbol{\xi}}_{e,c}, \gamma_{e,c}\big)$ to be greater than $\overline{\tau}$, so that

$$\Phi_{e,c}(\boldsymbol{\xi}_{e,c}) = p(\overline{\tau}, \boldsymbol{\xi}_{e,c}, \gamma_{e,c}) < \Phi_{e,c}(\hat{\boldsymbol{\xi}}_{e,c}) = p_{e,c}(\hat{\tau}, \hat{\boldsymbol{\xi}}_{e,c}, \gamma_{e,c}) \tag{3.77}$$

This can be achieved by specifying $\hat{\boldsymbol{\xi}}_{e,c}$ as proposed in [58], i.e.

$$\hat{\boldsymbol{\xi}}_{eF} := \alpha(\boldsymbol{\xi}_{eF}) = \begin{bmatrix} y_{e,c} \\ (1+\alpha)\dot{y}_{e,c} \\ (1+\alpha)\ddot{y}_{e,c} \\ (1+\alpha)y^{(3)}_{e,c} \end{bmatrix} \tag{3.78}$$

with $\alpha > 0$, hence leading to the following relation between $\hat{\tau}$ and $\overline{\tau}$

$$(1+\alpha)^{\frac{1}{3}}\overline{\tau} \leq \hat{\tau} \leq (1+\alpha)\overline{\tau} \tag{3.79}$$

that eventually fulfills (3.72).

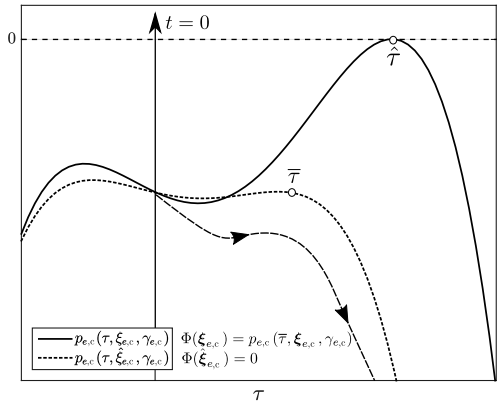


Figure 3.8: Impact of the *virtual* state space $\hat{\boldsymbol{\xi}}_{eF}$ on the shape of the corresponding polynomial function $p_{e,c}(\hat{\tau}, \hat{\boldsymbol{\xi}}_{e,c}, \gamma_{e,c})$ and on the value of the corresponding invariance function $\Phi_{e,c}(\hat{\boldsymbol{\xi}}_{e,c})$.

Fig. 3.8 graphically shows the impact of the introduced *virtual* state $\hat{\boldsymbol{\xi}}_{eF}$ on the shape of the corresponding polynomial function, while the solid line in Fig. 3.7 shows the time evolution of the output function $y_{e,c}$ within the introduced subset $\hat{\mathcal{G}}$. The effect of $\hat{\mathcal{G}}$ is to bound the higher derivatives of $y_{e,c}$, thus resulting in a slower convergence speed, yet effectively reducing the bouncing effect.

Furthermore, the bounding layer $\hat{\mathcal{G}}$ can be exploited in order to prevent constraint violations when additional constraints on the intermediate states of the integrator chain, i.e. $\dot{y}_{e,c},\ \ddot{y}_{e,c},\ y^{(3)}_{e,c}$, are simultaneously considered.

To this end, a state-of-the-art on-line trajectory generation algorithm [88], [89] can be adopted to provide a state of the output dynamics $\tilde{\boldsymbol{\xi}}_{e,c}$ which is consistent with all the constraints. Consequently, α can be selected so that $\tilde{\boldsymbol{\xi}}_{e,c}$ belongs to the bounding layer $\hat{\mathcal{G}}$, i.e.

$$\alpha \big| \tilde{\boldsymbol{\xi}}_{e,c} \in \hat{\mathcal{G}} \tag{3.80}$$

This way the subset $\hat{\mathcal{G}}$ is specified, acting on α, according to the output of a trajectory generator, so that constraint violation is prevented.
Note that for an output function $y_{e,c}$ with relative degree $r_{e,c} = 3$, the parameter α can be analytically obtained based on the following expression

$$\alpha \geq \frac{\ddot{F}_{c,max} \left(\frac{3\tilde{F}_c}{\ddot{F}_{c,max}} \right)^{\frac{2}{3}}}{2\dot{F}_{c,max}} - 1 \tag{3.81}$$

where $\dot{F}_{c,max},\ \ddot{F}_{c,max}$ are the maximum first and second derivative of the interaction force, respectively, considered within their related output functions $\dot{y}_{e,c} = \dot{F}_c - \dot{F}_{c,max} \leq 0$ and $\ddot{y}_{e,c} = \dot{F}_c - \ddot{F}_{c,max} \leq 0$, while $\tilde{F}_c$ is the value of the force provided by the trajectory generator through $\tilde{\boldsymbol{\xi}}_{e,c}$.

3.7.2 Robustness

Robot joint and link compliance matrices $\hat{C}_J(\boldsymbol{q})$ and $\hat{C}_L(\boldsymbol{q})$, respectively, are here assumed to be estimated off-line with good accuracy, although due to a deterministic uncertainty in the environment stiffness, only a partial knowledge of the total compliance is available, described by

$$C_T(\boldsymbol{q}) \in \begin{bmatrix} C_{T,min} & C_{T,max} \end{bmatrix} \tag{3.82}$$

where $C_{T,min}$ and $C_{T,max}$ represent a lower and upper bound, respectively.

Based on [58], the problem of guaranteeing robustness to compliance uncertainties can be traced back to the problem of properly selecting a value of α that can prevent the bouncing effect in presence of a deterministic model mismatch. Following from (3.78), this can be achieved if the following inequality holds

$$\frac{1+\alpha}{\hat{C}_T(\boldsymbol{q})} > \frac{1}{C_T(\boldsymbol{q})} \tag{3.83}$$

The worst case value of $\overline{\alpha}$ satisfying (4.42) is thus given by

$$\overline{\alpha} = \frac{C_{T,max}}{C_{T,min}} - 1 \tag{3.84}$$

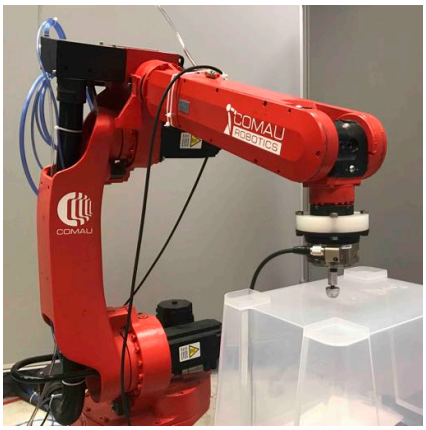


Figure 3.9: Experimental setup: plastic environment ($K_E \approx 10^4\ N/m$)

Therefore, by choosing $\alpha \geq \overline{\alpha}$, the same behavior shown in Fig. 3.7 can be ensured also in presence of compliance uncertainties.

Furthermore, as previously addressed in Sec. 3.6.1, constraint violations can occur due to force measurement noise. In this case, the recovery procedure described in Sec. 3.6.1 can be employed by simply redefining the force regulation error constraint as follows

$$y_{e,c} = \overline{F}_c - F_c \leq 0 \tag{3.85}$$

in order to drive back the state to the interior of the set $\mathcal{G}$.

3.7.3 Experimental Results

Force regulation experiments to a step reference of $15\ N$ have been performed on an environment setup now consisting of a plastic box ($K_E \approx 10^4\ N/m$), see Fig. 3.9, in order to compare the performance of the force regulation error constraint with bouncing reduction considered in this section with the one achieved by the adaptive constraint specification proposed in Sec. 3.6 and the integral controller with EWRLS estimation of the environment compliance [53].

A forgetting factor μ equal to 0.995 has been selected for the EWRLS algorithm, while the environment stiffness has been initialized to an approximate 50% over-estimate of the real stiffness, regarded as the considered deterministic uncertainty. The invariance function related to the force regulation constraint considered in this section, has been computed by exploiting the root-finding method provided by the GNU Scientific Library [90]

to return vector $\boldsymbol{\tau}_{roots}$ in Algorithm 1. Finally, the selected values for the controller parameters are: $\alpha = 3$ and $\gamma_{e,c} = -1 \cdot 10^5$.

Due to the presence of a further filter on the force measurement, the identified relation between the reference end effector displacement in the constrained direction $\Delta\overline{x}_c$ and the measured force in the same direction, can be expressed in the time domain by means of a second order low-pass filter with bandwidth $\hat{w}_n = 12.6\ rad/s$ and unitary damping.

$$-\frac{\hat{\omega}_n^2 \Delta\overline{x}_c}{C_T(\boldsymbol{q})} = \ddot{F}_c + 2\hat{\omega}_n \dot{F}_c + \hat{\omega}_n^2 F_c \tag{3.86}$$

As a drawback, a numerical computation of the state space of the output dynamics needs to be performed, instead of exploiting (3.62)

$$\boldsymbol{\xi}_{e,c_{(k)}} = \begin{bmatrix} y_{e,c} \\ \dot{y}_{e,c} \\ \ddot{y}_{e,c} \\ y_{e,c}^{(3)} \end{bmatrix}_{(k)} = \begin{bmatrix} F_{c(k)} - \overline{F}_c \\ \frac{1}{T_s}\left(y_{e,c_{(k)}} - y_{e,c_{(k-1)}}\right) \\ \frac{1}{T_s}\left(\dot{y}_{e,c_{(k)}} - \dot{y}_{e,c_{(k-1)}}\right) \\ -\frac{\hat{\omega}_n^2 \overline{x}_c}{C_T(\boldsymbol{q})} - \hat{\omega}_n^2 \dot{y}_{e,c_{(k)}} - 2\hat{\omega}_n \ddot{y}_{e,c_{(k)}} \end{bmatrix} \tag{3.87}$$

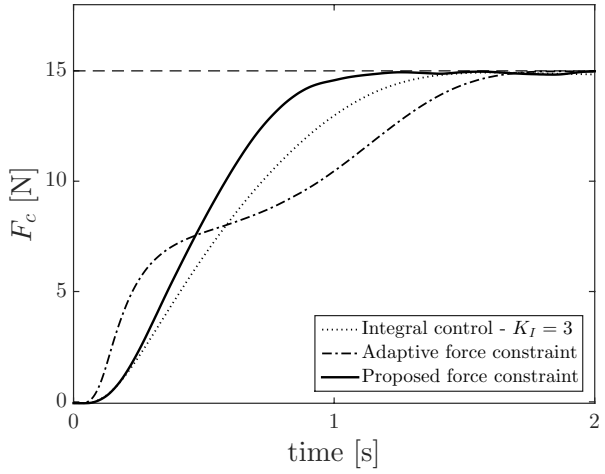


Figure 3.10: Experimental results on plastic environment: comparison among the integral controller [53] (dotted line), the adaptive constraint specification in (3.56) (dashed dotted line) and the force regulation constraint in (3.41) addressed in this section (solid line).

Following from the numerical differentiation of $\dot{y}_{e,c}$ and $\ddot{y}_{e,c}$, stiff contact situations need to be avoided in order to prevent deterioration of the con-

troller performance due to numerical noise.

As shown Fig. 3.10, by adopting the force regulation error constraint in (3.41) with the proposed bouncing reduction, a smoother convergence to the desired force value is achieved, compared to the adaptive constraint specification in [57]. As a result, an improved settling performance is now obtained over [57] and the integral controller in [53] despite compliance uncertainties.

CHAPTER 4

Data-Driven Design of Implicit Force Control

In order to avoid the deterioration of the force controller performance related to a rough identification of the environment model and to possible under-modeling, as described in Sec. 2.5, a *data-driven control design* approach can be exploited to directly tune the force controller using on-line or off-line Input-Output (I-O) data of the controlled system, without any need of parametric models of the system. Conversely, *model-based control* requires to model the plant, or identify the plant model, and then design the controller based on the plant model.

In Sec. 4.1, examples of modeling and identification of robot/environment interaction for robot force control are provided, together with an introduction on data-driven control methods. The *Virtual Reference Feedback Tuning* (VRFT) algorithm is described in Sec. 4.2, while its application to robot implicit force control and subsequent experimental validation are addressed in Sec. 4.3. In order to overcome the main limitations inherent in a data-driven approach to the robot force control problem, described in Sec. 4.4, an on-line implementation of VRFT is presented in Sec. 4.5, while a hierarchical control architecture is proposed in Sec. 4.6 to enhance the closed-loop performance.

4.1 Introduction

Robot force control has been traditionally addressed in a model-based fashion. As such, modeling of the robot-environment interaction and identifi-

cation of the model parameters are necessarily required. Assuming that the robot and the environment are not perfectly rigid, the implicit force control approach [49,50], extensively addressed in Chapter 2, belongs to those control strategies based on compliant interaction models. Robust control [91] can be applied when prior information on model parameters is missing. Alternatively, compliance parameters can be estimated and compensated within an adaptive control strategy, see e.g. [92–94] and the adaptive implementation of implicit force control in [53]. In other works [95–99] estimation and control algorithms are instead strictly interdependent.

The mentioned control approaches rely on simple interaction models, as the one described in Sec. 2.3 based on [50, 53], accounting for the robot joints compliance [100, 101] and link flexibility [102, 103] together with the environment elasticity. In this respect, the identification of compliance parameters can be performed not only for control but also for simulation, detection and perception purposes. The identification methods with the latter goals are generally more accurate and derive from nonlinear complex models [104–106], while the identification algorithms employed in control depend on linear models, e.g. Recursive Least Square (RLS) [107, 108], Extended Kalman Filter (EKF) [109, 110], algorithms based on an active observer [99], or Model Reference Control (MRC) methods, as the Model Reference Adaptive Control (MRAC) approach in [95, 97].

Note that in Model Reference Control, a model-based stabilizing controller is combined with an additional controller that minimizes the difference between the output of the system and a desired output response from a reference model. As a matter of fact, this relates to the second performance requirement introduced in Sec. 2.5 that will be addressed in this chapter. In this regard, various controller design approaches have been proposed throughout the years for the design of a high performance controller from the available model, see [111]. However, any model-based controller is not necessarily optimal when connected to the plant, as the control performance is limited by modeling errors. Hence, several data-driven controller tuning techniques have been proposed to avoid the problem of under-modeling and to facilitate the design of fixed-order controllers (e.g. PIDs) both iteratively and non-iteratively, e.g. [112], [113]. Among the other methods, *Virtual Reference Feedback Tuning* (VRFT) [114], that will be described in the following Section, allows the designer to tune a controller in one-shot, by means of simple least squares-like formulas, without any prior parameterization of the plant under control. Moreover, it is robust to measurement errors and easily applicable to fast PID tuning [115]. For the above reasons, it was used and showed satisfactory performance on a large number of ex-

perimental applications, see e.g. [113, 116–119]. A thorough comparison between direct and model-based approaches for model reference control problem can be found in [120].

4.2 The VRFT method

Virtual Reference Feedback Tuning (VRFT) is a control design method in which experimental data are directly mapped onto a fixed-structure controller parameters (e.g., PID parameters), without the need of a mathematical model of the system. The idea of Virtual Reference Feedback Tuning was first proposed in [121] with the name of Virtual Reference Direct Design (VRD^2) and subsequently extended in [114], [122] and [123] respectively for linear time-invariant (LTI), nonlinear and linear parameter-varying (LPV) systems. In this section, the method for the LTI case will be reviewed.

Consider the unknown LTI SISO stable plant $G(s)$. In the VRFT approach, the objective is to design a linear, fixed-order controller $\mathcal{C}_\rho(s)$, parameterized through ρ, for which the closed-loop system matches a given stable strictly proper *reference model* $\mathcal{M}(s)$. More specifically, let the controller parameterization be

$$\mathcal{C}_\rho(s) = \beta^T(s)\rho,$$

where $\beta(s)$ is a vector of n linear continuous-time basis functions

$$\beta^T(s) = \begin{bmatrix}\beta_1(s) & \beta_2(s) & \dots & \beta_n(s)\end{bmatrix}$$

for a controller of n^{th} order. More formally, the aim is to find the vector of parameters ρ that minimizes the (filtered) $\mathcal{H}_2$-norm of the difference between the reference model and the achieved closed-loop system

$$J_{mr}(\rho) = \left\| \left(\frac{G\mathcal{C}_\rho}{1 + G\mathcal{C}_\rho} - \mathcal{M} \right) W \right\|_2^2, \tag{4.1}$$

where $W(s)$ is a user-defined frequency-weighting filter.

Consider now that an open-loop collection of input-output (I-O) data

$$\mathcal{D}_N = \{u(t), y(t)\}_{t=1,\dots,N}$$

is available and let the output $y(t)$ be affected by the additive noise

$$v(t) = H(s)d(t),$$

where $H(s)$ is an unknown stable LTI system and $d(t)$ is a zero mean white Gaussian noise with variance of σ^2, uncorrelated with $u(t)$.

In standard *indirect data-driven* or *model-based* approaches, the above objective can be achieved by identifying from data a model $\hat{G}$ of the plant and designing a model-based controller $K(\hat{G})$ as

$$K(\hat{G}) = \frac{\mathcal{M}}{\hat{G}(1-\mathcal{M})}. \tag{4.2}$$

However, $K(\hat{G})$ might be non-causal and unstable. Moreover, it usually does not belong to the considered controller set, thus non-trivial controller reduction is needed.

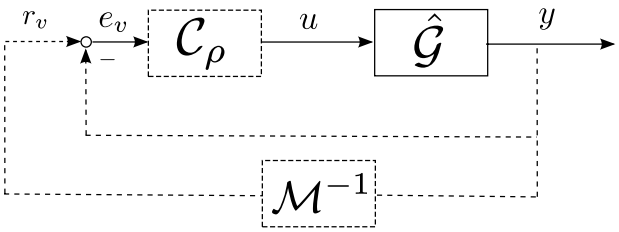


Figure 4.1: The *virtual* loop and the real plant.

The main idea to minimize (4.1) without identifying $G(s)$ is to build a *virtual* closed-loop system, shown in Fig. 4.1, where the input and output signals are equal to $u(t)$ and $y(t)$ and the closed-loop transfer function corresponds to $\mathcal{M}(s)$. From such a loop, the so-called *virtual reference* $r_v(t)$ and *virtual error* $e_v(t)$ signals are respectively computed as

$$\begin{aligned} r_v(t) &= \mathcal{M}^{-1}(s)\, y(t) \\ e_v(t) &= r_v(t) - y(t) \end{aligned} \tag{4.3}$$

The control design is then reduced to an identification problem, where the optimal controller generates $u(t)$ when fed by $e_v(t)$.
The cost function to be minimized is then

$$J_{vr}^N(\rho) = \frac{1}{N}\sum_{t=1}^{N}\Big(u_L(t) - \mathcal{C}_\rho(s)e_L(t)\Big)^2 \tag{4.4}$$

where $u_L(t) = L(s)u(t)$, $e_L(t) = L(s)e_v(t)$ and $L(s)$ is a suitable prefilter such that (4.4) is equal to the second-order Taylor expansion of (4.1) in the neighborhood of the minimum point [114]. A stability constraint [112] based on the small-gain theorem can additionally be considered in (4.4), representing a sufficient condition for closed-loop stability.

4.3 Application to Robot Implicit Force Control

In order to achieve the second performance requirement described in Sec. 2.5, i.e. avoidance of environment identification and under-modeling, a mixed model-based/VRFT-based controller for robot implicit force control is proposed in this Section, based on [124].

4.3.1 Plant Description

Consider as control variable $u \in \mathbb{R}$, the (motor-side) end effector displacement Δx_c along the direction elastically constrained by the environment, see Sec. 2.3

$$u := \Delta x_c = x_c - x_0 \tag{4.5}$$

where x_c is the (motor-side) end effector constrained position and x_0 is the end effector position on the environment undeformed surface, hereafter assumed to be known with good accuracy. The output variable $y \in \mathbb{R}$ in instead represented by the measured force at the end effector along the constrained direction

$$y := F_c \tag{4.6}$$

Denoting with $C_J(\boldsymbol{q}), C_L(\boldsymbol{q}) \in \mathbb{R}^{6\times 6}$ the robot joint and link compliance matrices, respectively, and with $G_E(s) \in \mathbb{R}$ the unknown transfer function, accounting for the environment behavior and possible unmodeled dynamics, the plant to be controlled $\mathcal{G}$ is hence represented by the transfer function

$$G(s) := \frac{Y(s)}{U(s)} = -\frac{1}{\boldsymbol{n}^T\big(C_J(\boldsymbol{q}) + C_L(\boldsymbol{q})\big)\,\boldsymbol{n} + G_E(s)} \tag{4.7}$$

where $\boldsymbol{n} \in \mathbb{R}^6$ is the vector representing the constrained direction. Note that, in case of a purely elastic environment, the same expression in (2.8) would be obtained.

Since the environment frequency response $G_E(s)$ is now assumed to be unavailable, the plant frequency response $G(s)$ is in turn partially available, except for the estimate of joint and link compliances $\hat{C}_J(\boldsymbol{q})$ and $\hat{C}_L(\boldsymbol{q})$, respectively. To compensate for such a lack of knowledge, based on Section 4.2, the VRFT approach would require to run experiments on the plant $\mathcal{G}$ and subsequently use the collected I-O trajectories u and y to solve the problem of finding a controller minimizing the model reference cost function.

4.3.2 Open-loop Experiment

It is worth pointing out that, since the plant's $\mathcal{G}$ transfer function $G(s)$ is robot configuration dependent, see (4.7), a number of open-loop experiments comparable to the number of possible robot configurations would be required, in order to apply the VRFT approach. On the other hand, the unknown environment frequency response $G_E(s)$ is inherently robot configuration independent, since it reflects the behavior of the environment only.

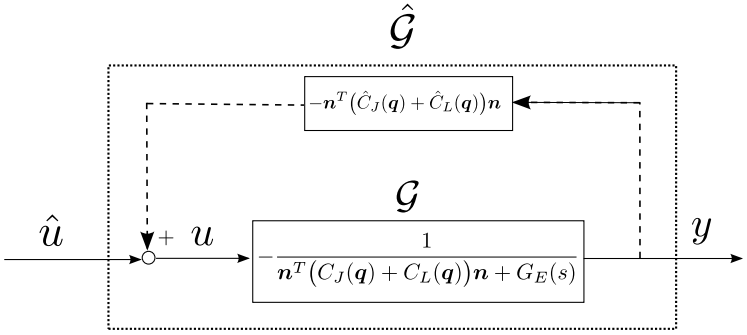


Figure 4.2: VRFT open-loop modified control scheme, with the introduced feedback loop.

Based on this consideration and assuming an accurate off-line estimation of the robot joints and links compliances, i.e.

$$\begin{aligned}\hat{C}_L(\boldsymbol{q}) &\approx C_L(\boldsymbol{q})\\ \hat{C}_J(\boldsymbol{q}) &\approx C_J(\boldsymbol{q})\end{aligned} \tag{4.8}$$

a feedback loop is introduced on the plant $\mathcal{G}$, shown in Fig. 4.2, with the following transfer function

$$H(s) = -\boldsymbol{n}^T\big(\hat{C}_J(\boldsymbol{q}) + \hat{C}_L(\boldsymbol{q})\big)\boldsymbol{n}, \tag{4.9}$$

The modified plant with the introduced feedback loop, see Fig. 4.2, will be hereafter denoted with $\hat{\mathcal{G}}$, being $\hat{u} \in \mathbb{R}$ its control input. The plant $\hat{\mathcal{G}}$ transfer function $\hat{G}(s)$ is now entirely dependent on the unknown environment frequency response.

$$\hat{G}(s) := \frac{G(s)}{1 - G(s)H(s)} = \frac{-\frac{1}{\boldsymbol{n}^T\left(C_J(\boldsymbol{q})+C_L(\boldsymbol{q})\right)\boldsymbol{n}+G_E(s)}}{1 - \frac{\boldsymbol{n}^T\left(\hat{C}_J(\boldsymbol{q})+\hat{C}_L(\boldsymbol{q})\right)\boldsymbol{n}}{\boldsymbol{n}^T\left(C_J(\boldsymbol{q})+C_L(\boldsymbol{q})\right)\boldsymbol{n}+G_E(s)}} \tag{4.10}$$

yielding from (4.8)

$$\hat{G}(s) \approx -\frac{1}{G_E(s)} \tag{4.11}$$

It is worth pointing out that, apart from making the dependence on the environment transfer function explicit, the introduced feedback loop on the plant $\mathcal{G}$ has the notable effect that the open-loop experiment on the resulting plant $\hat{\mathcal{G}}$ is now robot configuration independent. Therefore, a single robot configuration needs to be selected in order to run the open-loop experiment. Furthermore, note that $\hat{\mathcal{G}}$ can be considered open-loop stable, due to the presence of the (stable) inner robot joint position controller.

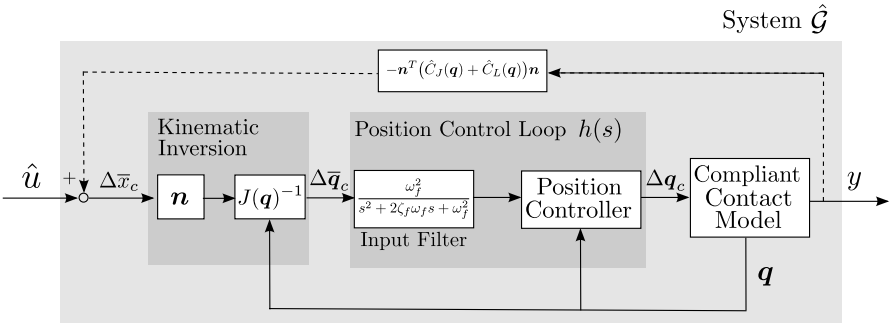


Figure 4.3: VRFT open-loop robot control structure.

As depicted in Fig. 4.3, the end effector displacement command $\Delta\overline{x}_c \in \mathbb{R}$, which is in turn translated in an equivalent joint displacement command $\Delta\overline{\boldsymbol{q}}_c \in \mathbb{R}^6$ through kinematic inversion (2.9), is computed as follows

$$\Delta\overline{x}_c = \hat{u} - \boldsymbol{n}^T\left(\hat{C}_J(\boldsymbol{q}) + \hat{C}_L(\boldsymbol{q})\right)\boldsymbol{n}\, y \tag{4.12}$$

No additional displacement has been applied in the unconstrained directions. Finally, the input of the robot position control loop $\overline{\boldsymbol{q}} \in \mathbb{R}^6$ is given by (2.12).

4.3.3 Closed-loop Experiment

As proposed in [124], the VRFT algorithm is employed to tune the parameters of a PID controller.
Consider now the frequency response of the PID controller

$$\mathcal{C}_\rho(s) = \beta(s)^T\rho$$

where

$$\rho = \begin{bmatrix} K_P & K_I & K_D \end{bmatrix}^T$$

are the parameters to tune and

$$\beta(s) = \begin{bmatrix} 1 & \frac{1}{s} & \frac{s}{1+sT_D} \end{bmatrix}^T$$

is the corresponding vector of linear continuous-time transfer functions. The time constant of the derivative part T_D is selected as twice the sampling time.
According to [114], the VRFT algorithm provides the parameter vector ρ starting from the open-loop experiments data set on the system $\hat{\mathcal{G}}$ and the desired frequency behavior for the closed-loop system described by the *reference model* $\mathcal{M}(s)$, where $\mathcal{M}(s)$ is given, or, alternatively, it is derived from user requirements. The VRFT algorithm allows the resulting PID controller to achieve the desired closed-loop dynamics when it is possible and, in any case, minimizes a suitable frequency-wise model matching error, also when the output signal is corrupted by some measurement noise.

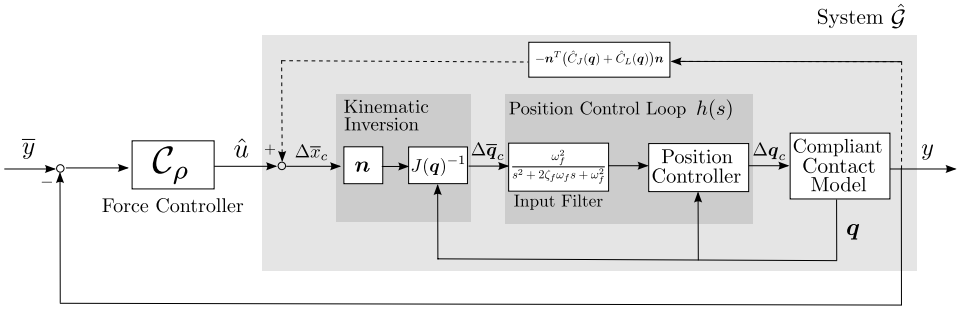


Figure 4.4: VRFT closed-loop robot control structure.

The PID controller $\mathcal{C}_\rho(s)$ tuned by VRFT is finally employed within the closed-loop system shown in Fig. 4.4, where the end effector displacement command $\Delta\overline{x}_c$ is given by

$$\Delta\overline{x}_c = \mathcal{C}_\rho(s)(\overline{y}-y) - \boldsymbol{n}^T\left(\hat{C}_J(\boldsymbol{q})+\hat{C}_L(\boldsymbol{q})\right)\boldsymbol{n}\,y \tag{4.13}$$

while the reference signal $\overline{y}\in\mathbb{R}$ represents the desired value for the interaction force in the constrained direction.

4.3.4 Experimental Results

The open-loop experiments have been performed on the same environment setup described in Sec. 2.5. The chosen open-loop control action $\hat{u}$ consists in a sequence of end effector displacements with Trapezoidal Velocity Profile (TVP), along the force controlled direction, i.e. the z Cartesian direction. Time histories of I-O data are shown in Fig. 4.5. The adequacy of the chosen open-loop profile of $\hat{u}$ for the Least Squares Estimation problem in (4.4) has been confirmed by checking that the prefiltered *virtual error* e_L, obtained from the collected output (4.3), results in a persistently

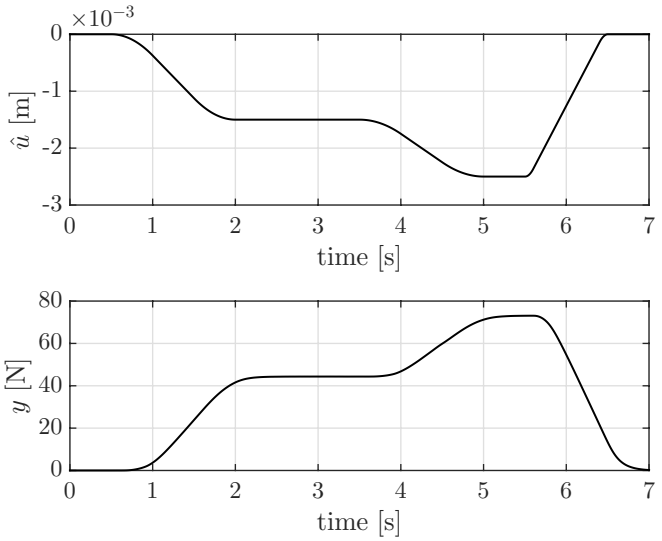


Figure 4.5: Open-loop experiment: sequence of applied end effector displacements $\hat{u}$ along the direction constrained by the environment (upper plot) and measured force y.

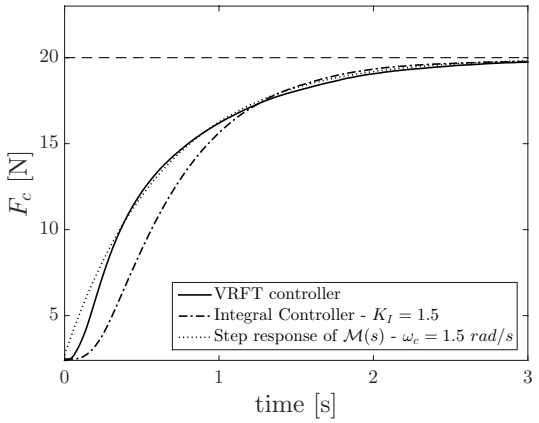
exciting signal [125, 126] with order greater than 3 [1] and by computing the correlation coefficient of the least squares fitting.
The VRFT Toolbox for MATLAB [127] has been exploited to provide the parameters vector ρ from open-loop experiments. Taking into account the low-pass filter ($\omega_f = 75.4\ rad/s$ and $\zeta_f = 1$) applied to the joint reference, previously described in Sec. 2.2, the reference model $\mathcal{M}(s)$ has been selected as the following first order low-pass filter

$$\mathcal{M}(s) = \frac{\omega_c}{s + \omega_c} \tag{4.14}$$

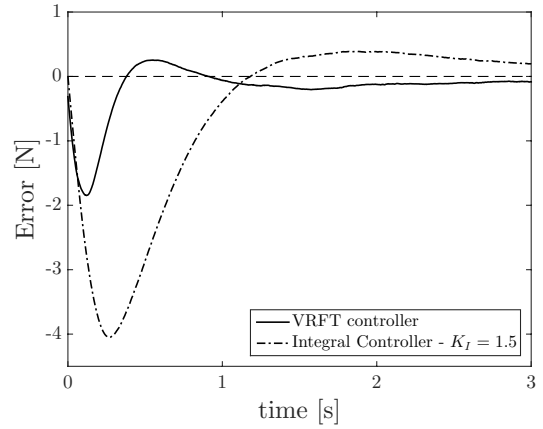
where $\omega_c = 1.5\ rad/s$ provides a suitable frequency separation with the robot position control loop. The PID controller gains identified by VRFT are: $K_P = -5.37 \cdot 10^{-5}$, $K_I = -2.53 \cdot 10^{-4}$, $K_D = 1.62 \cdot 10^{-8}$.
The force regulation experiments, performed in presence of a step reference signal $\overline{F}_c$ of 20 N on the same environment setup described in Sec. 2.5, compare the proposed control approach with the purely integral controller in [50]. The chosen value for the integral gain: $K_I = 1.5$ has been selected so that the related first-order closed-loop transfer function is equal to the desired frequency behavior $\mathcal{M}(s)$ in (4.14). The environment compliance $\hat{C}_E$ required by [50], has been estimated off-line by means of Least

[1] MATLAB function `pexcit` has been used: `https://it.mathworks.com/help/ident/ref/pexcit.html`

Square Estimation (LSE), see Sec. 2.3.



(a) Interaction force time history



(b) Error time history

Figure 4.6: Closed-loop experiments. Upper plot: time histories of the measured interaction force obtained with the VRFT controller (solid line) and with the integral controller in [50] (dashed-dotted line), compared with the step response of the reference model $\mathcal{M}(s)$ in (4.14) (dotted line). Lower plot: time histories of the error between the measured interaction forces and the step response of $\mathcal{M}(s)$.

Fig. 4.6(a) shows the measured force time histories obtained with the proposed controller and the integral controller in [50], compared with the first-order step response given by the reference model $\mathcal{M}(s)$ in (4.14). The corresponding error time history is shown in Fig. 4.6(b), whereas Table 4.1 provides the maximum absolute error together with the error standard

Table 4.1: Maximum absolute value (Max Abs. Error [N]) and standard deviation (Error SD [N]) of the error between achieved and desired force step response.

Controller	Max Abs. Error [N]	Error SD [N]
Integral controller [50]	4.05	1.24
VRFT controller	1.58	0.32

deviation (SD).

As a matter of fact, thanks to the proposed data-driven controller [124], force overshoots and possible instabilities of the controlled system related to environment modeling and identification (see Sec. 2.5) are inherently prevented. Furthermore, as shown in Fig. 4.6, an increased matching between the desired and the achieved closed-loop dynamics can be obtained compared to the purely integral controller, providing a significant reduction of the maximum absolute value and standard deviation (SD) of the error between the measured and the desired force step response, see Table 4.1.

4.4 Discussion

The main limitations inherent in a data-driven approach to the robot force control problem are summarized in the following remarks.

- **Remark 1:** The required open-loop collection of input-output data represents a first issue in the implementation of data-driven methods on typical industrial robot controller (as the one described in Sec. 2.2) where storage and manipulation of large amount of sensory information can prevent real-time execution of control algorithms. Furthermore, the MATLAB functions of VRFT Toolbox [127], developed for *off-line* identification of the controller parameters, are clearly unsuitable for real-time applications. Consequently, the open-loop and closed-loop experiments must be performed separately, in order to run the VRFT Toolbox in MATLAB environment. Clearly enough, this time consuming procedure restricts the applicability of the proposed control strategy in industrial settings. In order to address this issue an *on-line* implementation of the VRFT algorithm will be proposed in Sec. 4.5.

- **Remark 2:** In direct data-driven control methods, the desired closed-loop performance (i.e. the bandwidth of the closed-loop system) is set by the choice of the reference model. Note that if a data-driven con-

troller has been designed based on a tentative low-performance reference model, re-tuning of the controller will be necessarily required in order to increase the closed-loop performance, by changing the reference model. Moreover, in case a high-performance reference model is chosen, force overshoots can arise as shown in Fig. 4.7, where the adopted first-order reference model $\mathcal{M}(s)$ with $\omega_c = 4.5\ rad/s$ reduces the step response settling time to approximately $1\ s$ compared to Fig. 4.6. As previously described in Sec. 2.5, output overshoots in a force control scenario are often connected to potential damages to the workpiece, and should be therefore avoided. Based on these considerations and following the work of Piga et al. [128], in Sec. 4.6 an outer model predictive controller (MPC) acting as a *reference governor* [63, 129], is applied to the inner VRFT closed-loop system, aiming at enhancing the closed-loop performance.

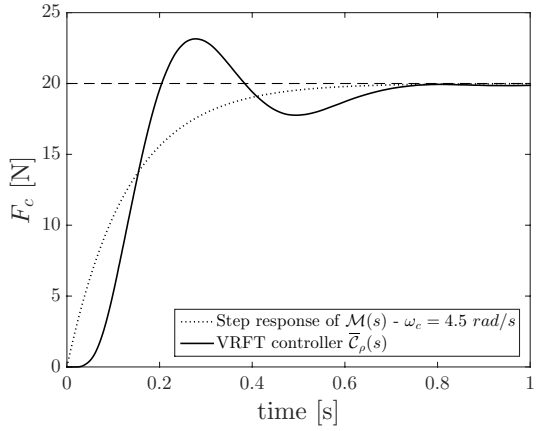


Figure 4.7: Closed-loop experiments choosing an aggressive first-order reference model $\mathcal{M}(s)$ with $\omega_c = 4.5\ rad/s$: time history of the interaction force produced by the VRFT controller (solid line) and the step response of the reference model $\mathcal{M}(s)$(dotted line).

4.5 On-line VRFT

This section proposes an *on-line* implementation of the VRFT algorithm based on the Recursive Least Squares (RLS) estimation method [125, 126] and shows its application to robot implicit force control.

4.5.1 RLS Implementation

Consider the *discrete time* controller parametrization

$$C_\rho(z) = \beta(z)^T \rho$$

where $\beta(z) = \begin{bmatrix}\beta_1(z) & \beta_2(z) & \dots & \beta_n(z)\end{bmatrix}^T$ is a vector of linear discrete-time transfer functions and $\rho = \begin{bmatrix}\rho_1 & \rho_2 & \dots & \rho_n\end{bmatrix}^T$ is the n-dimensional vector of controller parameters. According to the VRFT method (see Sec. 4.2), the relation between the prefiltered control input $u_L \in \mathbb{R}$ and the prefiltered *virtual error* $e_L \in \mathbb{R}$ can be written as

$$u_{L,k} = C_{\rho_k}(z)\, e_{L,k} = \beta_z^T\, \rho_k\, e_{L,k} \tag{4.15}$$

where the subscript k denotes the sample index.
By denoting the regressor vector φ as

$$\varphi_k = \beta(z)\, e_{L,k} \tag{4.16}$$

and considering as parameter vector θ the vector of controller parameters ρ

$$\theta_k = \rho_k \tag{4.17}$$

equation (4.15) can be expressed in regressor form as

$$u_k = \varphi_k^T \theta_k \tag{4.18}$$

The RLS update equations for the VRFT algorithm are hence given by

$$\begin{cases} \hat{\theta}_k = \hat{\theta}_{k-1} + P_k \varphi_k \varepsilon_k \\ \varepsilon_k = u_{L,k} - \varphi_k^T \hat{\theta}_{k-1} \\ P_k = P_{k-1} - \frac{P_{k-1}\varphi_k \varphi_k^T P_{k-1}}{1+\varphi_k^T P_{k-1} \varphi_k} \end{cases} \tag{4.19}$$

The update of the parameter estimate $\hat{\theta}$ is calculated based on the previous estimate and the residual term ε, while the update of the adaptation gain matrix P is given by the last equation. Initialization is required for both the parameter vector $\hat{\theta}_0$ and the adaptation gain matrix P_0. Note that a large norm of P_0, meaning that $\hat{\theta}_0$ is not reliable, eventually results in a fast convergence.

According to Sec. 4.3.3, consider now a discrete time PID controller with transfer function

$$C_\rho(z) = K_P + \frac{K_I}{1 - z^{-1}} + K_D(1 - z^{-1}) = \beta_z^T \rho \tag{4.20}$$

where

$$\beta(z) = \begin{bmatrix} 1 & \frac{1}{1-z^{-1}} & 1 - z^{-1} \end{bmatrix}^T \tag{4.21}$$

and

$$\rho = \begin{bmatrix} K_P & K_I & K_D \end{bmatrix}^T \tag{4.22}$$

The discrete time reference model $\mathcal{M}(z)$ corresponding to the first order low-pass filter in (4.14) is derived as follows

$$\mathcal{M}(z) = \frac{(1-A)z^{-1}}{1 - Az^{-1}} \tag{4.23}$$

where

$$A = 1 - T_s\, \omega_c$$

A similar choice of a PID controller combined with a first order reference model can be found in the AVRFT approach [130], an adaptive version of the VRFT applied to PID controller design.

In order to apply the RLS algorithm in (4.19), it is necessary to specify the parameter vector $\hat{\theta}$, the residual term ε and the regressor vector φ.
The parameter vector is given by

$$\hat{\theta} = \hat{\rho} = \begin{bmatrix} \hat{K}_P & \hat{K}_I & \hat{K}_D \end{bmatrix}^T$$

while the residual term ε can be written as follows (neglecting the prefilter, i.e. $u_L = u$ and $e_L = e_v$)

$$\varepsilon_k = u_k - \varphi_k^T \hat{\theta}_{k-1}$$

The regressor vector φ can be finally computed by means of (4.3) and (4.16), leading to

$$\varphi_k^T = e_{v,k}\, \beta_z^T = \left(\mathcal{M}(z)^{-1} - 1\right) y_k\, \beta_z^T \tag{4.24}$$

By plugging (4.21) and (4.23) in the previous equation, one obtains

$$\varphi_k^T = \left(\frac{1 - Az^{-1}}{(1-A)z^{-1}} - 1 \right) \begin{bmatrix} 1 & \frac{1}{1-z^{-1}} & 1 - z^{-1} \end{bmatrix} y_k \tag{4.25}$$

and therefore

$$\varphi_k^T = \begin{bmatrix} \varphi_{p,k} & \varphi_{i,k} & \varphi_{d,k} \end{bmatrix} \tag{4.26}$$

where

$$\begin{aligned} \varphi_{p,k} &= \frac{1}{1-A} \left(y_k - y_{k-1} \right) \\ \varphi_{i,k} &= \frac{1}{1-A}\, y_k \\ \varphi_{d,k} &= \frac{1}{1-A} \left(y_k - 2y_{k-1} + y_{k-2} \right) \end{aligned} \tag{4.27}$$

Compared to conventional VRFT, where a database collected from an open-loop experiment is required, now the amount of past I-O data reduces to: y_{k-1} and y_{k-2} according to (4.27). This, combined with the computationally inexpensive operations in (4.19), clearly allows for real-time execution of the VRFT algorithm.
Finally note that the proposed RLS implementation of VRFT relies on the assumption that the (unknown) environment model does not change during the performed experiments, i.e. the forgetting factor is equal to 1. Nevertheless, in case of a time varying environment behavior: think of a multi-material surface, an Exponentially Weighted Recursive Least Squares (EWRLS) implementation of VRFT can be easily obtained from (4.19), thus allowing for an adaptive update of the controller parameters, see [130, 131].

4.5.2 Experimental Results

Thanks to the proposed RLS implementation of VRFT, it is now possible to setup a single force control experiment on the robot system, divided in the following steps:

Step 1 A sequence of end effector displacements along the force controlled direction $\hat{u}$ is applied on the open-loop system according to Sec. 4.3.2, while PID parameters are simultaneously computed by the RLS algorithm in (4.19).

Step 2 The final parameter estimate (at the end of Step 1) is used as candidate PID controller within the closed-loop system in Sec. 4.3.3 and a force regulation task is subsequently performed.

In analogy to Sec. 4.3, the z Cartesian direction represents the force controlled direction. The adequacy of the chosen open-loop profile $\hat{u}$ (*Step 1*) for the RLS estimation problem has been previously addressed in Sec. 4.3.2. Finally, a step reference signal $\overline{F}_c$ of $20\ N$ has been applied to the closed-loop system in *Step 2*.

A set of three experiments has been performed on different environment materials: aluminium, wood, plastic.
Fig. 4.8, Fig. 4.9 and Fig. 4.10 show the time histories of I-O data (upper plots) and the estimated PID controller parameters (lower plot), obtained during *Step 1* for the aluminium, wood and plastic environment, respectively. The bandwidth of the reference model $\mathcal{M}(z)$ in (4.23) has been set by choosing $A = 0.997$ ($\omega_c = 1.5\ rad/s$), thus corresponding to a step

response settling time of approximately $3\ s$ as in Sec. 4.3.3.
The following initial estimate of the parameter vector has been chosen: $\hat{\theta}_0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$, while the approximate initialization: $P_0 = \delta \cdot \boldsymbol{I}_3$ with $\delta = 100$ has been employed for the adaptation matrix.

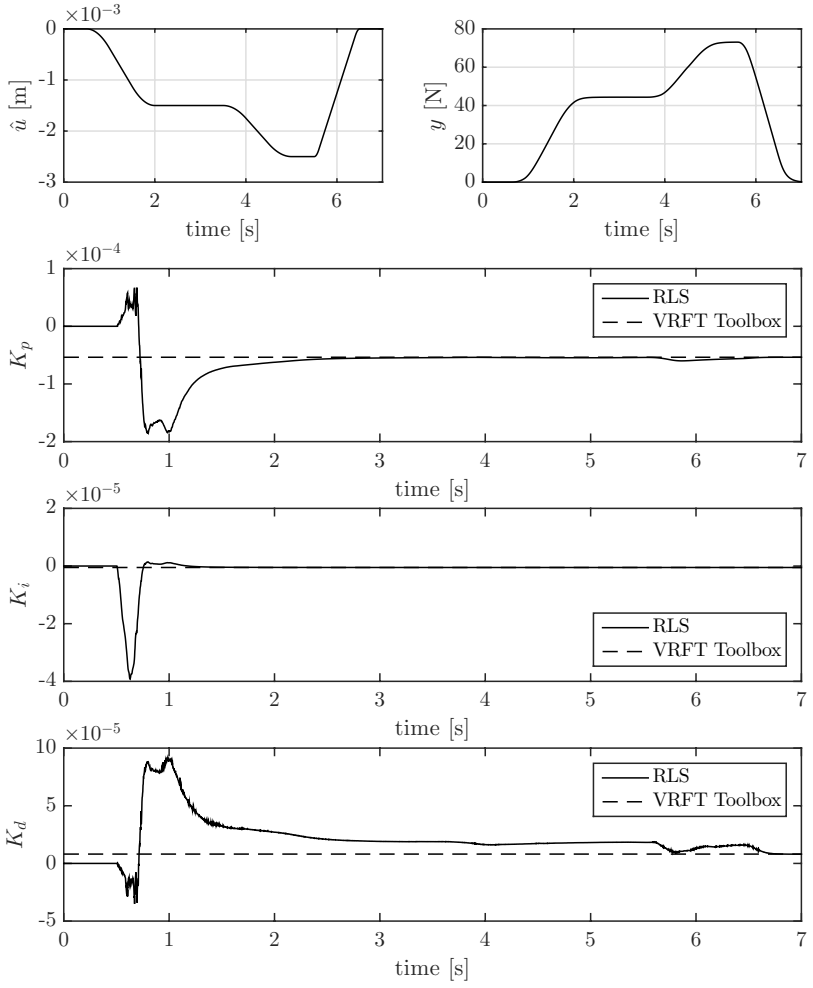


Figure 4.8: *Step 1*: aluminium. Upper plots: sequence of applied end effector displacements $\hat{u}$ along the direction constrained by the environment (left) and measured force y (right). Lower plots: RLS estimated controller parameters (solid lines) and corresponding values returned by the VRFT Toolbox for MATLAB (dashed lines).

The time history of the RLS estimated controller parameters (solid lines in

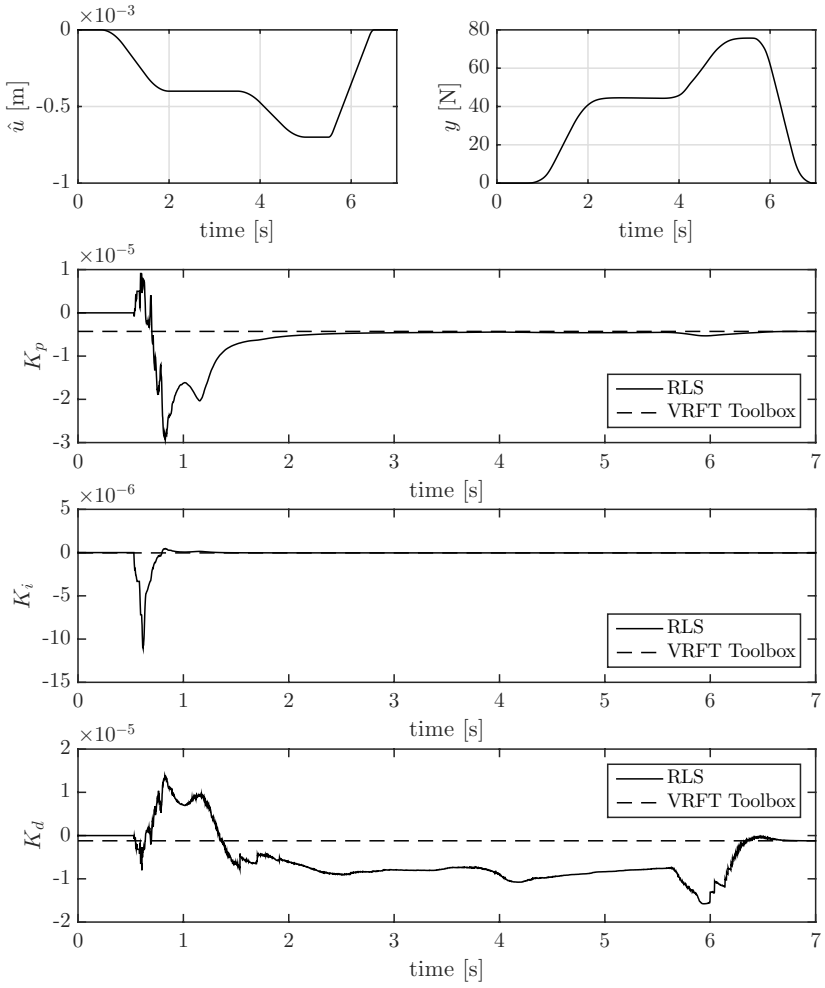


Figure 4.9: *Step 1*: wood. Upper plots: sequence of applied end effector displacements $\hat{u}$ along the direction constrained by the environment (left) and measured force y (right). Lower plots: RLS estimated controller parameters (solid lines) and corresponding values returned by the VRFT Toolbox for MATLAB (dashed lines).

the lower plots) is compared to the values returned off-line by the VRFT Toolbox for MATLAB (dashed lines) based on the complete collection of I-O data. As expected, the values returned by RLS at the end of *Step 1*, match with high accuracy (up to 6 significant digits) those returned by the VRFT Toolbox.

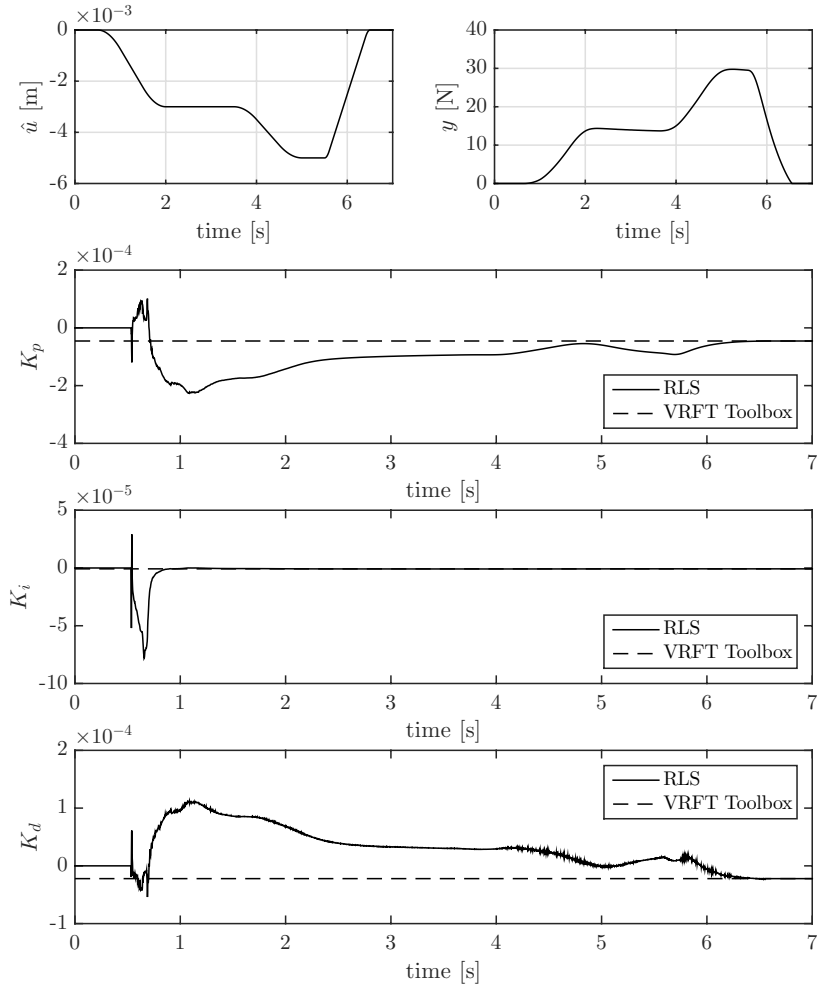
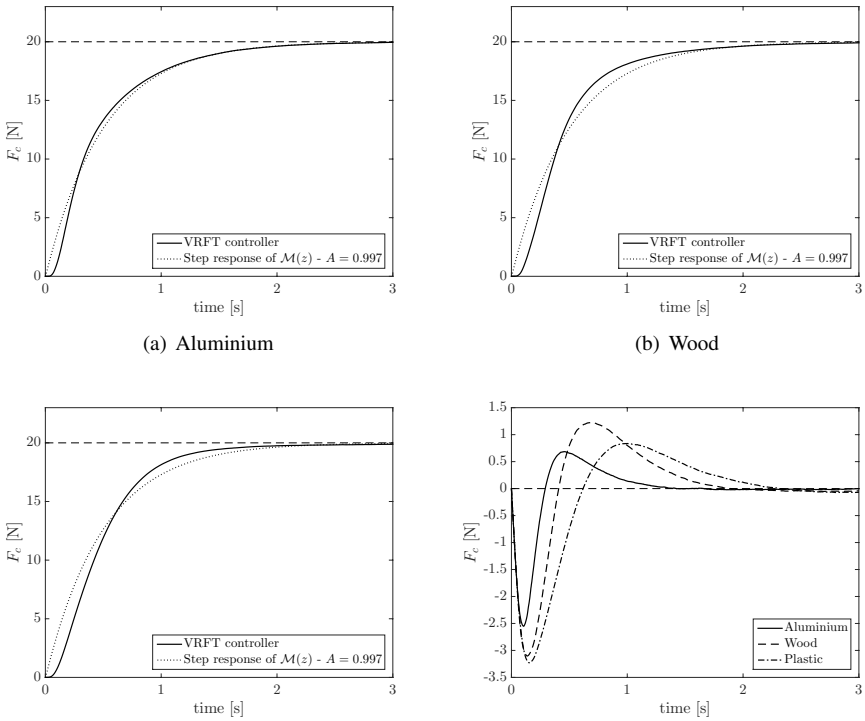


Figure 4.10: *Step 1*: plastic. Upper plots: sequence of applied end effector displacements $\hat{u}$ along the direction constrained by the environment (left) and measured force y (right). Lower plots: RLS estimated controller parameters (solid lines) and corresponding values returned by the VRFT Toolbox for MATLAB (dashed lines).

Finally Fig. 4.11 shows the time histories of the measured interaction force during the force regulation task in *Step 2*, for the three experiments performed on the different environments. The corresponding time history of the errors between the achieved force responses and the step response of

$\mathcal{M}(z)$ is shown in Fig. 4.11(d). Again, note that a significant matching between desired and achieved closed-loop dynamics is achieved, see Table 4.2.



(a) Aluminium (b) Wood

(c) Plastic (d) Error

Figure 4.11: *Step 2*: time history of the measured interaction force (solid line) during the force regulation task employing the VRFT controller estimated in *Step 1*, compared to the step response of the reference model $\mathcal{M}(z)$ in (4.23) with $A = 0.997$ ($\omega_c = 1.5\ rad/s$) (dotted line). The corresponding time history of the error between the achieved force responses and the step response of $\mathcal{M}(z)$ is shown in the last plot.

Table 4.2: Maximum absolute value (Max Abs. Error [N]) and standard deviation (Error SD [N]) of the error between achieved and desired force step response.

Material	Max Abs. Error $[N]$	Error SD $[N]$
Aluminium	2.55	0.35
Wood	3.10	0.57
Plastic	3.23	0.62

4.6 Hierarchical Approach

This section aims at further enhancing the closed-loop performance (i.e. the bandwidth of the closed-loop system) without any need of retuning the VRFT controller, while simultaneously preventing the occurrence of force overshoots connected to the choice of a high-performance reference model, as previously discussed in Sec. 4.4. In this respect a hierarchical control architecture is proposed in the following, which relates to the ideas in [128].

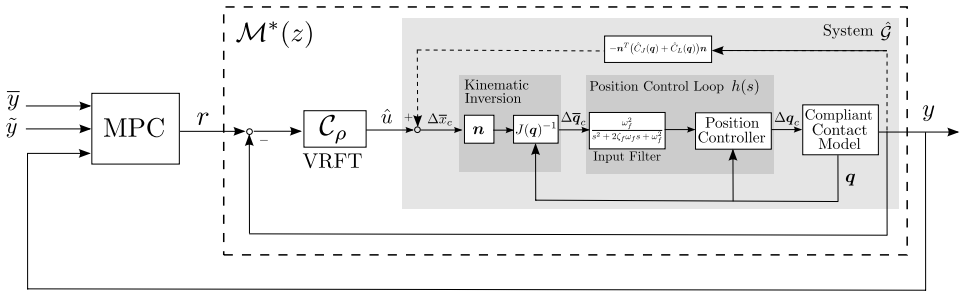


Figure 4.12: Hierarchical control architecture: the inner VRFT controller is designed from data, so that the inner loop matches as much as possible $\mathcal{M}(s)$, whereas the outer MPC controller enhances the performance.

As shown in Fig. 4.12, the controller is split into two components organized in a hierarchical fashion. An inner VRFT controller $C_\rho(z)$ is first designed, as described in Sec. 4.3 and Sec. 4.5, to match an achievable reference model $\mathcal{M}(z)$. Then, an outer linear model predictive controller (MPC) acting as a *reference governor* [63, 129] selects on-line, and, according to a *receding horizon* strategy, the optimal reference supplied to the inner closed-loop system in order to improve the closed-loop performance of the inner loop, without complicating the data-driven design procedure. By merging together the two controllers in the above hierarchical fashion, one can choose a low-performance, i.e. with a moderate bandwidth, inner closed-loop behavior $\mathcal{M}(z)$, while the tasks of optimizing the closed-loop performance is left to the outer MPC, which can be designed based on the (known) closed-loop dynamics $\mathcal{M}(z)$.

4.6.1 Outer MPC Design

In the following it is assumed that the inner data-driven controller $C_\rho(z)$ is designed to achieve a low-performance (i.e. with low bandwidth) first order reference model $\mathcal{M}(z)$, see (4.23), whose bandwidth is set by the parameter A with $0 < A < 1$. In order to account for an additional time

delay, due to robot joint friction effects, the introduced transfer function $\mathcal{M}^*(z)$, describing the stable dynamics of the inner closed-loop system, will be considered as a *prediction model* for the outer MPC-based controller

$$\mathcal{M}^*(z) = \frac{(1-A)z^{-1}}{1-Az^{-1}} \cdot z^{-N} \tag{4.28}$$

where N is the number of sampling periods corresponding to the considered time delay T_d given by

$$N = \frac{T_d}{T_s}$$

T_s being the sampling time.
Based on the *prediction model* in (4.28), the outer MPC controller is designed in order to achieve a high-performance (i.e. with high bandwidth) closed-loop dynamics, represented by the following *high-performance reference model*

$$\tilde{\mathcal{M}}(z) = \frac{(1-\tilde{A})z^{-1}}{1-\tilde{A}z^{-1}} \tag{4.29}$$

with

$$0 < \tilde{A} < A < 1 \tag{4.30}$$

In order to allow for real-time implementation, the following horizon-1 MPC problem can be formulated at each time instant t

$$\min_{r(t)} \mathcal{Q}_y \left(\tilde{y}(t+N_p) - y(t+N_p)\right)^2 + \mathcal{Q}_r \left(\overline{y} - r(t)\right)^2 \tag{4.31a}$$

subject to

$$y(t+N_p) = A^{N_p} y(t) + (1-A)\sum_{i=0}^{N_p-2} A^{N_p-i-1} r(t-N+i) + (1-A)r(t) \tag{4.31b}$$

$$\tilde{y}(t+N_p) = A^{N_p} y(t) + (1-A)\sum_{i=0}^{N_p-1} A^{N_p-i-1} g(t-N+i) = \tilde{A}^{N_p}\tilde{y}(t) + (1-\tilde{A}^{N_p})\overline{y} \tag{4.31c}$$

where $\overline{y}$ is the output reference (hereafter assumed to be constant) and $\tilde{y}$ is the output of the *high-performance reference model* in (4.29), $\mathcal{Q}_y \in \mathbb{R}$ and $\mathcal{Q}_r \in \mathbb{R}$ are nonnegative weights, while N_p is the *minimum prediction horizon* due to the considered delay, given by

$$N_p = N + 1$$

In the MPC formulation the following terms are penalized in (4.31a):

1. the tracking error between the output of the *high-performance reference model* $\tilde{y}$ and the output y.

2. the error between the reference signal $\overline{y}$ and the MPC output r.

Due to the considered time delay, the N_p-step ahead predictions $y(t+N_p)$ and $\tilde{y}(t+N_p)$ can be computed as in (4.31b) and (4.31c), representing the Lagrange equations for the *prediction model* $\mathcal{M}^*(z)$ in (4.28) and for the *high-performance reference model* $\tilde{\mathcal{M}}(z)$ in (4.29), respectively.
Note that from a practical perspective, while the first penalty term improves the bandwidth of the closed-loop system, the goal of the penalty on $\overline{y}-r$ is to guarantee the absence of output (i.e. force) overshoots with respect to the reference value.

4.6.2 Stability

Let $\varepsilon_N := N-\overline{N}$ be the time delay estimation error between the actual time delay ($\overline{N}$) of the inner closed-loop system $\mathcal{M}^*(z,\overline{N})$ and the estimated time delay (N) considered in the MPC problem.
Denoting

$$\mathcal{Q} := \frac{\mathcal{Q}_r}{\mathcal{Q}_y} \in [0,+\infty)$$

and

$$\alpha := \frac{(1-A)^2}{(1-A)^2+\mathcal{Q}} = \begin{cases} 1 & \text{if } \mathcal{Q}_r = 0 \\ 0 & \text{if } \mathcal{Q}_y = 0 \end{cases} \tag{4.32}$$

the following Theorem holds.

Theorem 1. *There exists a set of values of the weigths $\mathcal{Q}_y \geq 0$ and $\mathcal{Q}_r > 0$ ensuring the asymptotic stability of the closed-loop system, for any ε_N.*

Proof. The optimal output of the LTI-MPC in (4.31) can be computed explicitly, unlike the LPV-MPC in [128], leading to the following expression

$$\begin{aligned} r(t) =& \frac{(1-A)\tilde{y}(t+N_p) - (1-A)A^{N_p}y(t) + \mathcal{Q}\overline{y}}{(1-A)^2+\mathcal{Q}} + \\ & - \frac{(1-A)^2\sum_{i=0}^{Np-2} A^{N_p-i-1}g(t-N+i)}{(1-A)^2+\mathcal{Q}} \end{aligned} \tag{4.33}$$

which can be rewritten in the discrete-time frequency domain through $\mathcal{Z}$-transform as

$$r(z) = \frac{\left(\frac{\alpha}{1-A}F_1(z) + \frac{\alpha\mathcal{Q}}{(1-A)^2}\right)\overline{y} - \frac{\alpha A^{N_p}}{1-A}y(z)}{1+\alpha F_2(z)} \tag{4.34}$$

where $F_1(z)$ is a stable transfer function for $\tilde{A} < 1$

$$F_1(z) := \frac{1-\tilde{A}^{N_p}}{z^{N_p} - \tilde{A}^{N_p}} \tag{4.35}$$

while $F_2(z)$ is given by

$$\begin{aligned} F_2(z) :&= \sum_{i=0}^{Np-2} A^{N_p-i-1} z^{-N+i} = \frac{A}{z}\sum_{i=0}^{Np-2}\frac{A}{z}^{N_p-i-2} = \\ &= \frac{A(z^N - A^N)}{z^N(z-A)} \end{aligned} \tag{4.36}$$

Accordingly, the proposed hierarchical control architecture results in the control scheme shown in Fig. 4.13 that will be employed in the following proving the stability of the closed-loop.

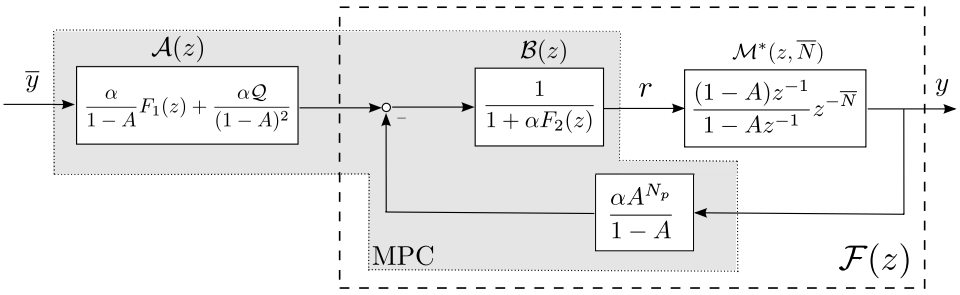


Figure 4.13: Hierarchical control architecture: equivalent control scheme for stability analysis.

Denote with

$$\mathcal{A}(z) := \frac{\alpha}{1-A}F_1(z) + \frac{\alpha\mathcal{Q}}{(1-A)^2} = \begin{cases} \frac{F_1(z)}{1-A} & \text{if } \mathcal{Q}_r = 0 \\ 1 & \text{if } \mathcal{Q}_y = 0 \end{cases} \tag{4.37}$$

and

$$\mathcal{B}(z) := \frac{1}{1+\alpha F_2(z)} \tag{4.38}$$

Since $\mathcal{A}(z)$ represents a stable transfer function with off-set, the stability of the closed-loop system $\mathcal{F}(z)$

$$\mathcal{F}(z) = \frac{\mathcal{B}(z)\mathcal{M}^*(z,\overline{N})}{1 + \frac{\alpha A^{N_p}}{1-A}\mathcal{B}(z)\mathcal{M}^*(z,\overline{N})} \tag{4.39}$$

can be verified by computing the roots of the related characteristic polynomial, given by

$$\Phi(z) = z^N\big(z + (\alpha - 1)A\big) + \alpha A^{N+1}\big(z^{\varepsilon_N} - 1\big) \tag{4.40}$$

- For $\varepsilon_N = 0$ the resulting polynomial

  $$\Phi(z) = z^N\big(z + (\alpha - 1)A\big)$$

  has N roots equal to zero and a stable real root $\lambda \in [0, A)$

  $$\lambda = \frac{A\mathcal{Q}}{(1-A)^2 + \mathcal{Q}}$$

  Since $A < 1$, see (5.24), asymptotic stability is therefore ensured for any choice of $\mathcal{Q}_y$ and $\mathcal{Q}_r$. For $\varepsilon_N \neq 0$, by applying Jury stability criterion, it can be shown that the characteristic polynomial in (4.40) corresponds to a stable closed-loop dynamics if the following sufficient condition is satisfied

  $$0 \leq \alpha < \min(\overline{\alpha}, 1) \tag{4.41}$$

  where $\overline{\alpha}$, for $\varepsilon_N > 0$, is given by

  $$\overline{\alpha} = \frac{A^{N+1} - A^N + \sqrt{A^N(A^{N+2} - 2A^{N+1} + 5A^N + 4)}}{2A^{N+1}(A^N + 1)} \tag{4.42}$$

  while, for $\varepsilon_N > 0$, $\overline{\alpha}$ is obtained by replacing N with $\overline{N}$ in (4.42).

□

As a matter of fact, if an exact knowledge of the time delay is available ($\varepsilon_N = 0$) the proposed hierarchical control architecture guarantees the asymptotic stability of the closed-loop system for any choice of the weights $\mathcal{Q}_y$ and $\mathcal{Q}_r$. Conversely, in presence of time delay uncertainty ($\varepsilon_N \neq 0$), asymptotic stability can be guaranteed if $\mathcal{Q}_y$ and $\mathcal{Q}_r$ satisfy condition (4.41). For the sake of fairness, note that deterioration of the closed-loop behavior can be expected for increasing values of $|\varepsilon_N|$.

4.6.3 Experimental Results

The effectiveness of the proposed hierarchical control approach has been evaluated on a set of three force regulation experiments performed on different environment materials, i.e. aluminium, wood and plastic, according to the following task execution structure:

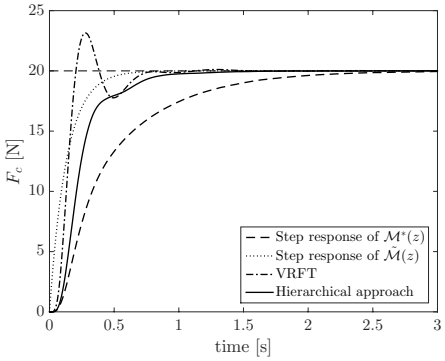
Step 1 A sequence of end effector displacements along the force controlled direction $\hat{u}$ is applied on the open-loop system according to Sec. 4.3.2, while PID parameters of the VRFT controller are simultaneously computed by RLS (4.19) based on a low-performance reference model $\mathcal{M}(z)$.

Step 2 The final parameter estimate (at the end of Step 1) is used as candidate PID controller for the inner data driven control loop, whose reference r is set by the outer MPC controller, based on the *prediction model* $\mathcal{M}^*(z)$ and the *high-performance reference model* $\tilde{\mathcal{M}}(z)$.

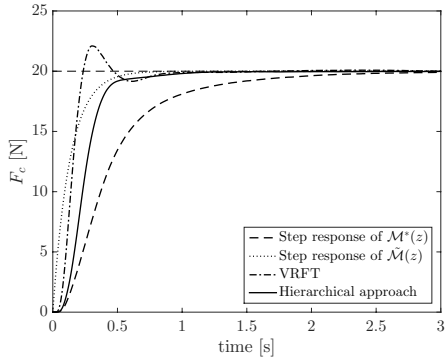
In analogy to Sec. 4.3, the z Cartesian direction represents the force controlled direction, while a step reference signal $\overline{F}_c$ of $20\ N$ has been applied in *Step 2*. RLS initialization has been performed according to Sec. 4.5.2. The bandwidth of the low-performance reference model $\mathcal{M}(z)$ in (4.23) has been set by choosing $A = 0.997$ ($\omega_c = 1.5\ rad/s$), thus corresponding to a step response settling time of approx. $3s$, as in Sec. 4.3.3. A time delay $T_d = 0.08s$ (i.e. $N = 40$) has been considered within the *prediction model* $\mathcal{M}^*(z)$ in (4.28). The bandwidth of the *high-performance reference model* $\tilde{\mathcal{M}}(z)$ in (4.29) corresponds to a step response settling time of approximately $1s$, by setting $\tilde{A} = 0.985$ ($\omega_c = 4.5\ rad/s$). Finally, the following values of the MPC weights are used: $\mathcal{Q}_y = 1$ and $\mathcal{Q}_r = 0.001$ based on the performed stability analysis.

Fig. 4.14 shows the time histories of the measured interaction force during the force regulation task (*Step 2*) for the three experiments performed on the different environments. The step responses of the *prediction model* $\mathcal{M}^*(z)$ (dashed line) and of the *high-performance reference model* $\tilde{\mathcal{M}}(z)$ (dotted line) are shown together with the force time history produced by a VRFT controller tuned on the *high-performance reference model* (dashed-dotted line) and by the proposed hierarchical control approach (solid line).

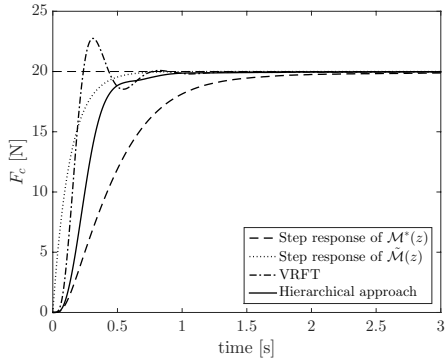
As it can be noticed, while force overshoots arise employing the standard VRFT as described in Sec. 4.3, closed-loop performance is effectively enhanced by the proposed hierarchical control approach, which allows to achieve a high settling performance simultaneously preventing the occurrence of output overshoots.



(a) Aluminium



(b) Wood



(c) Plastic

Figure 4.14: *Step 2*: comparison between the time history of the measured interaction force during the force regulation task employing a VRFT controller tuned on the *high-performance reference model* $\tilde{\mathcal{M}}(z)$ (dashed-dotted line) and the proposed hierarchical control approach (solid line), together with the step responses of $\tilde{\mathcal{M}}(z)$ (dotted line) and of the *prediction model* $\mathcal{M}^*(z)$ (dashed line).

Part II

FORCE CONTROLLED BIMANUAL ASSEMBLY

CHAPTER 5

Experimental Setup and Control Framework

The recent industrial development of bimanual, or dual-arm, robots (e.g. Motoman SDA-series, DLR Justin, ABB YuMi, Willow Garage PR2, Rethink Robotics Baxter, Kawada Nextage) has rapidly captured the attention of researchers and users on this new class of robotic systems. Besides cost saving and space saving advantages over single arm systems, the capability of independently or synchronously handling objects enables dual-arm robots to fully employ their increased dexterity in assembly operations. On the other hand, the growing complexity of robotic tasks, combined with the inherent redundancy of bimanual robots, makes real-time motion planning more challenging for this type of platforms.

This chapter provides details on the hardware used for experimental validation and on the constraint-based programming approach responsible for real-time motion planning. The dual-arm robotic platform used in the experiments and its interface are presented in Sec. 5.1, together with the chosen method for estimation of the interaction force/torque. Finally, Sec 5.2 describes the constraint-based trajectory generation framework that will be adopted in Chapter 6 to develop force control algorithms for bimanual assembly tasks.

5.1 Hardware and Interfaces

This section gives a brief overview of the hardware and interface available for experimental validation.

5.1.1 Bimanual Robot

The second robot used in this thesis is the ABB YuMi [132, 133] formerly known as FRIDA and shown in Fig. 5.1.

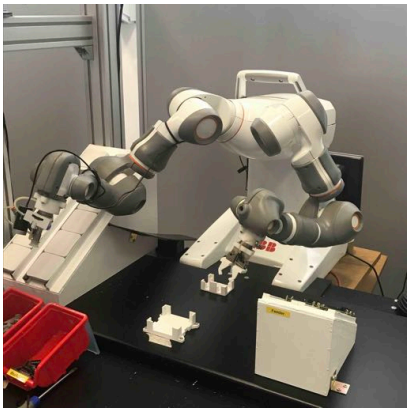


Figure 5.1: The ABB YuMi robot used in this thesis inside a collaborative assembly workcell.

YuMi is a dual-arm robot specifically designed for robotic assembly. Each of the two arms is redundant with 7 degrees of freedom. Its light-weight skeleton is covered with a floating plastic casing wrapped in soft padding. This, combined with power and speed limitations, provides an inherent degree of safety that allows for safe Human-Robot collaboration [134].

Aiming at performing the assembly operations addressed in the following chapter, the YuMi robot available at the Merlin Lab of Politecnico di Milano has been equipped with pneumatically actuated grippers on both end effectors and 3D printed tools designed for the specific assembly operation.

5.1.2 Interface to the Robot

The robot described in the previous section is controlled by the ABB IRC5 control system: each joint is individually position controlled, while a main computer calculates the joint position reference. Joint velocity reference and torque feedforward can be additionally sent to the inner joint position controller [135]. The available measurements include joint positions and velocities obtained by numerical differentiation and filtering of the joint positions.
The low-level joint position control loops run at a sampling frequency of 2 kHz, while a *research interface* for setting the references and reading

measurements runs at $250\ Hz$. The external controller using the research interface is executed on an external PC with Linux and Xenomai for real-time performance. The external PC communicates with the robot controller through LabComm protocol [136]. Executable code is generated via the Simulink Coder, and the compiled program runs as an external controller using the research interface.

5.1.3 Force Sensing

In its current setup the YuMi robot is not equipped with any wrist mounted force/torque sensor or joint torque sensors, therefore estimation of the interaction force/torque is required in order to perform assembly tasks, see [47]. Sensorless force estimation is hereafter performed based on the well known generalized momentum method [137, 138]. The benefit of using this method is that (noisy) joint acceleration measurements are not needed, although knowledge of the robot dynamical model is required.

In the following, details on the YuMi dynamical model and on the estimation method in [137, 138] are provided.

Consider the dynamic model of a rigid robot with n DoFs

$$\boldsymbol{B}\left(\boldsymbol{q}\right)\ddot{\boldsymbol{q}} + \boldsymbol{C}\left(\boldsymbol{q},\dot{\boldsymbol{q}}\right)\dot{\boldsymbol{q}} + \boldsymbol{\gamma}\left(\boldsymbol{q},\dot{\boldsymbol{q}}\right) = \boldsymbol{\tau} + \boldsymbol{\tau}_{ext} \tag{5.1}$$

where $\boldsymbol{B}\left(\boldsymbol{q}\right) \in \mathbb{R}^{n\times n}$ is the symmetric positive definite inertia matrix, depending on the joint configuration $\boldsymbol{q} \in \mathbb{R}^n$. Coriolis/centrifugal forces and torques are represented by $\boldsymbol{C}\left(\boldsymbol{q},\dot{\boldsymbol{q}}\right)\dot{\boldsymbol{q}} \in \mathbb{R}^n$, while gravity and friction effects are contained in $\boldsymbol{\gamma}\left(\boldsymbol{q},\dot{\boldsymbol{q}}\right) \in \mathbb{R}^n$. The control torques applied to the robot are given by $\boldsymbol{\tau} \in \mathbb{R}^n$. Finally, $\boldsymbol{\tau}_{ext} \in \mathbb{R}^n$ represents the external torques.
The inertial, centrifugal and gravitational terms for the YuMi used in the experiments have been assembled on the basis of the dynamic parameters provided by the manufacturer. The following model [139] has been used to describe the joint friction torque for the i-th joint τ_i^F which take part in the definition of $\boldsymbol{\gamma}\left(\boldsymbol{q},\dot{\boldsymbol{q}}\right)$

$$\tau_i^F = \tau_i^O + \tau_i^C \tanh\left(130\,\dot{q}_i\right) + \begin{cases} D_i^+\dot{q}_i & \text{if } \dot{q}_i > 0 \\ D_i^-\dot{q}_i & \text{if } \dot{q}_i < 0 \end{cases} \tag{5.2}$$

where τ_i^C and τ_i^O represent the Coulomb friction and a torque offset, respectively, whilst D_i^+ and D_i^- are the viscous coefficients for positive and negative velocities, respectively. Identification of friction coefficients has been

performed in [139] via Least Squares (LS) estimation with non-negativity constraints. In order to account for possible uncertainties, the following threshold has been additionally applied to the module of τ_i^F

$$\tau^{thr} = \tau_i^{still} + \tau_i^{move} \text{sech}\left(30\dot{q}_i\right) \tag{5.3}$$

where τ_i^{still} and τ_i^{move} are static and viscous terms, respectively.
The identified friction model and the considered threshold for a single joint are shown in Fig. 5.2.

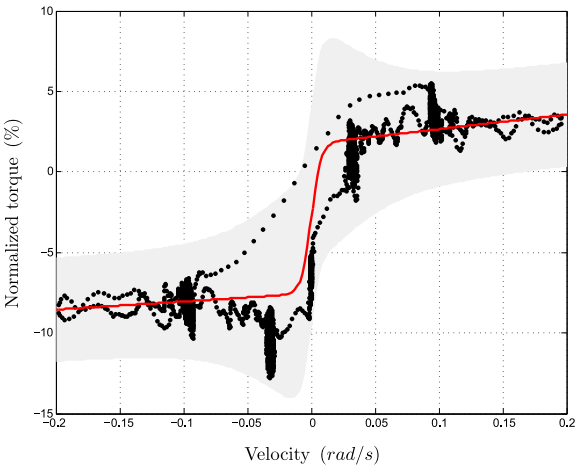


Figure 5.2: Joint friction torque for the i-th joint: joint friction model (red), considered threshold (gray shadow) and experimental data (black dots).

Defining $\boldsymbol{p} \in \mathbb{R}^n$ as the generalized momentum vector

$$\boldsymbol{p} = \boldsymbol{B}\left(\boldsymbol{q}\right)\dot{\boldsymbol{q}} \tag{5.4}$$

its time derivative is given by

$$\dot{\boldsymbol{p}} = \boldsymbol{C}\left(\boldsymbol{q}, \dot{\boldsymbol{q}}\right)^T \dot{\boldsymbol{q}} - \boldsymbol{\gamma}\left(\boldsymbol{q}, \dot{\boldsymbol{q}}\right) + \boldsymbol{\tau} + \boldsymbol{\tau}_{ext} \tag{5.5}$$

According to [137], estimation of the external torques $\boldsymbol{\tau}_{ext}$ can be performed by computing the residual vector $\boldsymbol{r} \in \mathbb{R}^n$

$$\boldsymbol{r} = \boldsymbol{K}_R \left[\boldsymbol{p} - \int_0^{t_k} \left(\boldsymbol{\tau} + \boldsymbol{C}^T\left(\boldsymbol{q}, \dot{\boldsymbol{q}}\right)\dot{\boldsymbol{q}} - \boldsymbol{\gamma}\left(\boldsymbol{q}, \dot{\boldsymbol{q}}\right) + \boldsymbol{r}\right) dt\right] \tag{5.6}$$

where $\boldsymbol{K}_R$ is a symmetric positive definite matrix.
From (5.6), one obtains a first order stable linear relationship between the external torques $\boldsymbol{\tau}_{ext}$ and the residual

$$\dot{\boldsymbol{r}} = \boldsymbol{K}_R\left(\boldsymbol{\tau}_{ext} - \boldsymbol{r}\right) \tag{5.7}$$

Accounting for robot redundancy and assuming the interaction force/torque $\boldsymbol{\mu} \in \mathbb{R}^m$ (with $m < n$) to be applied to the Tool Center Point (TCP), the following equation can be finally adopted to estimate $\boldsymbol{\mu}$

$$\boldsymbol{\mu} = \left(\boldsymbol{J}\left(\boldsymbol{q}\right)^T \right)^{\dagger} \boldsymbol{r} \tag{5.8}$$

where $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{m \times n}$ represents the Jacobian of the TCP frame, while † stands for the Moore-Penrose pseudo-inverse.

5.2 Real-Time Motion Planning

The dexterity of a dual-arm system clearly allows for the execution of tasks with an increased level of complexity compared to traditional positioning tasks. Furthermore, kinematic redundancy can be exploited to simultaneously perform multiple task with a given order of priority. On the other hand, these capabilities introduce additional constraints on the planning problem of a robot trajectory. Moreover, the robot should be able to modify its motion in real-time to cope with unpredictable sensory events.
Based on these considerations, the following section provides details on the constraint-based motion planning algorithm described in [48, 140, 141], adopted as control framework in the following of this thesis.

5.2.1 Trajectory Generation Using Constrained Control

The control architecture in [48], shown in Fig. 5.3, is based on a real-time *trajectory generation* algorithm combined with an optimization based *reactive controller*, feeding the inner industrial joint position/velocity controller. Subscript k refers to the value of the related vector at discrete time instant t_k.

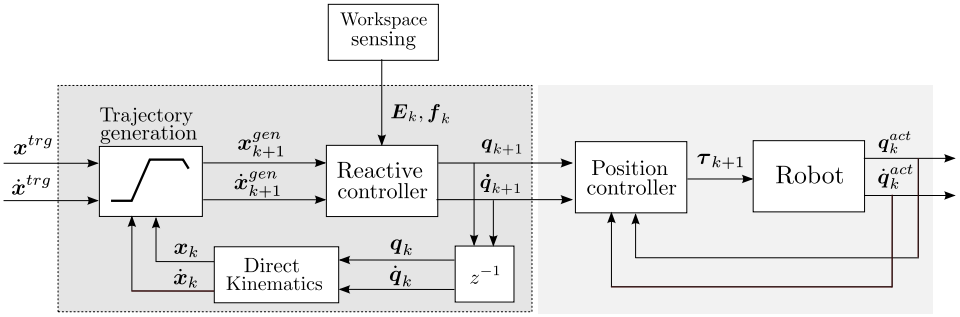


Figure 5.3: Block diagram of the control framework.

The *reactive controller*, responsible for kinematic inversion in presence of redundancy and control constraints, generates the joint position and velocity *reference* to the robot position/velocity controller $\left(\boldsymbol{q}_{k+1},\ \dot{\boldsymbol{q}}_{k+1} \in \mathbb{R}^n\right)$ based on the state of motion in task coordinates provided by a *trajectory generation* algorithm [142] $\left(\boldsymbol{x}^{gen}_{k+1},\ \dot{\boldsymbol{x}}^{gen}_{k+1} \in \mathbb{R}^m\right)$, the current state of motion in task coordinates $\left(\boldsymbol{x}_k,\ \dot{\boldsymbol{x}}_k \in \mathbb{R}^m\right)$ and workspace sensing. If this entails a deviation from the planned trajectory, a new trajectory from the current state of motion to the target state of motion $\left(\boldsymbol{x}^{trg},\ \dot{\boldsymbol{x}}^{trg} \in \mathbb{R}^m\right)$ is (reactively) generated by the *trajectory generation* algorithm.

The following assumption is made:

Assumption: The reference joint acceleration vector represents the control input $\boldsymbol{u} \in \mathbb{R}^n$

$$\ddot{\boldsymbol{q}}_k = \boldsymbol{u}_k$$

Based on this, the *joint space* process model reduces to a (discrete-time) double integrator.

$$\begin{aligned} \boldsymbol{q}_{k+1} &= \boldsymbol{q}_k + T_s \dot{\boldsymbol{q}}_k + 0.5\, T_s^2 \boldsymbol{u}_k \\ \dot{\boldsymbol{q}}_{k+1} &= \dot{\boldsymbol{q}}_k + T_s \boldsymbol{u}_k \end{aligned} \tag{5.9}$$

where T_s is the controller sampling time.
By considering the well-known forward kinematic relation between joint coordinates $\boldsymbol{q} \in \mathbb{R}^n$ and task variables $\boldsymbol{x} \in \mathbb{R}^m$

$$\boldsymbol{x} = \boldsymbol{f}\left(\boldsymbol{q}\right) \quad \dot{\boldsymbol{x}} = \boldsymbol{J}\left(\boldsymbol{q}\right)\dot{\boldsymbol{q}} \quad \ddot{\boldsymbol{x}} = \dot{\boldsymbol{J}}\left(\boldsymbol{q}\right)\dot{\boldsymbol{q}} + \boldsymbol{J}\left(\boldsymbol{q}\right)\ddot{\boldsymbol{q}} \tag{5.10}$$

the second order *task space* process model is given by

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + T_s \boldsymbol{J}_k \dot{\boldsymbol{q}}_k + 0.5\, T_s^2 \left(\dot{\boldsymbol{J}}_k \dot{\boldsymbol{q}}_k + \boldsymbol{J}_k \boldsymbol{u}_k\right) \\ \dot{\boldsymbol{x}}_{k+1} &= \boldsymbol{J}_k \dot{\boldsymbol{q}}_k + T_s \left(\dot{\boldsymbol{J}}_k \dot{\boldsymbol{q}}_k + \boldsymbol{J}_k \boldsymbol{u}_k\right) \end{aligned} \tag{5.11}$$

where $\boldsymbol{J}_k = \boldsymbol{J}\left(\boldsymbol{q}_k\right)$ and $\dot{\boldsymbol{J}}_k = \dot{\boldsymbol{J}}\left(\boldsymbol{q}_k\right)$.

A generic constraint, considered within the *reactive controller*, has the following expression, which is linear in the control input

$$\boldsymbol{E}_k \boldsymbol{u}_k \leq \boldsymbol{f}_k \tag{5.12}$$

where $\boldsymbol{E}_k$ and $\boldsymbol{f}_k$ can be constant or time-varying vectors/matrices. Possible constraints can account for typical motion planning limitations (i.e.

joint limits, bounds on the maximum velocities and accelerations in joint and task space) but also for sensor-related events occurring at time instant k, see [48, 141].
The considered constraints have been specified according to the *invariance control* approach, previously described in Sec. 3.2, assuming a relative degree not greater than 2. It is worth recalling that constraints' satisfaction is ensured if set invariance condition (3.15) holds for all active constraints. This in turn can be expressed consistently with (5.12) by exploiting I-O Linearization (3.38), leading to

$$\mathcal{A}_{\mathcal{I}}\boldsymbol{u}_k \preceq \boldsymbol{b}_{\mathcal{I}} \tag{5.13}$$

where $\mathcal{I}$ is the set of active constraints at time instant k, $\mathcal{A}_{\mathcal{I}} = [\boldsymbol{a}_i^T]$ and $\boldsymbol{b}_{\mathcal{I}} = [\gamma_i - b_i]$ with $i \in \mathcal{I}$. Finally $\boldsymbol{a}_i(\boldsymbol{x})^T$ and b_i are defined in (3.38). The operator $\preceq$ denotes the inequality $\leq$ for all elements of the vectors.

The complete control architecture is described in Algorithm 2.

Algorithm 2

Input: $\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k, \boldsymbol{x}^{trg}, \dot{\boldsymbol{x}}^{trg}, \dot{\boldsymbol{x}}^{max}, \ddot{\boldsymbol{x}}^{max}, \boldsymbol{E}_k, \boldsymbol{f}_k$
Output: $\boldsymbol{q}_{k+1}, \dot{\boldsymbol{q}}_{k+1}, \boldsymbol{\tau}_k$

1: compute relevant kinematic quantities (Jacobians, task variables, etc.)
2: generate a trajectory, consistent with $\dot{\boldsymbol{x}}^{max}, \ddot{\boldsymbol{x}}^{max}$ to connect $\boldsymbol{x}_k, \dot{\boldsymbol{x}}_k$ with $\boldsymbol{x}^{trg}, \dot{\boldsymbol{x}}^{trg}$
3: evaluate the next desired state of motion $\boldsymbol{x}_{k+1}^{gen}, \dot{\boldsymbol{x}}_{k+1}^{gen}$
4: solve the following QP problem

$$\min_{\boldsymbol{u}_k} \mathcal{L}\left(\dot{\boldsymbol{x}}_{k+1} - \dot{\boldsymbol{x}}_{k+1}^{gen}, \boldsymbol{x}_{k+1} - \boldsymbol{x}_{k+1}^{gen}\right) \tag{5.14a}$$

subject to

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + T_s\boldsymbol{J}_k\dot{\boldsymbol{q}}_k + 0.5T_s^2\left(\dot{\boldsymbol{J}}_k\dot{\boldsymbol{q}}_k + \boldsymbol{J}_k\boldsymbol{u}_k\right) \tag{5.14b}$$

$$\dot{\boldsymbol{x}}_{k+1} = \boldsymbol{J}_k\dot{\boldsymbol{q}}_k + T_s\left(\dot{\boldsymbol{J}}_k\dot{\boldsymbol{q}}_k + \boldsymbol{J}_k\boldsymbol{u}_k\right) \tag{5.14c}$$

$$\mathcal{A}_{\mathcal{I}}\boldsymbol{u}_k \preceq \boldsymbol{b}_{\mathcal{I}} \tag{5.14d}$$

5: update the state of motion as in (5.9)

$$\begin{aligned} \boldsymbol{q}_{k+1} &= \boldsymbol{q}_k + T_s\dot{\boldsymbol{q}}_k + 0.5T_s^2\boldsymbol{u}_k \\ \dot{\boldsymbol{q}}_{k+1} &= \dot{\boldsymbol{q}}_k + T_s\boldsymbol{u}_k \end{aligned} \tag{5.15}$$

The cost function in (5.14a) weighs the difference between the next state of motion $(\boldsymbol{x}_{k+1}, \dot{\boldsymbol{x}}_{k+1})$ and its reference values $\left(\boldsymbol{x}_{k+1}^{gen}, \dot{\boldsymbol{x}}_{k+1}^{gen}\right)$ obtained as an

output of the *trajectory generation* algorithm.
A quadratic cost function can be used to this purpose

$$\begin{aligned}\mathcal{L}\left(\boldsymbol{e}, \dot{\boldsymbol{e}}\right) = 0.5\boldsymbol{e}^T\boldsymbol{Q}_p\boldsymbol{e} + 0.5\dot{\boldsymbol{e}}^T\boldsymbol{Q}_v\dot{\boldsymbol{e}} + \boldsymbol{e}^T\boldsymbol{Q}_{p,v}\dot{\boldsymbol{e}} + \\ + \boldsymbol{g}_p^T\boldsymbol{e} + \boldsymbol{g}_v^T\dot{\boldsymbol{e}}\end{aligned} \tag{5.16}$$

where $\boldsymbol{Q}_p, \boldsymbol{Q}_v, \boldsymbol{Q}_{p,v}$ are positive definite matrices, while $\boldsymbol{g}_p$ and $\boldsymbol{g}_v$ are vectors of suitable dimension.
Constraints (5.14b) and (5.14c) are required to map joint space velocities and accelerations into their task space counterparts. Finally, (5.14d) represents the set of all considered constraints accounting for motion planning limitations and sensor-related events, as previously discussed.
The output of the Quadratic Programming (QP) problem consists in the reference joint acceleration $\ddot{\boldsymbol{q}}_k$, i.e. the control input $\boldsymbol{u}_k$, according to Assumption 2, from which the corresponding reference position $\boldsymbol{q}_{k+1}$ and velocity $\dot{\boldsymbol{q}}_{k+1}$ can be computed through (5.9) and fed to the inner position/velocity controller.

In presence of redundant degrees of freedom, as with the dual-arm robotic platform described in Sec. 5.1.1, or in case of task redundancy, the QP problem in (5.14) has infinite, and equally optimal, solutions.
Being $\boldsymbol{u}_k^0$ any of those optimal solutions, robot redundancy can be exploited to perform a second task having lower priority with respect to the main task. This can be obtained only if the alternative solution $\boldsymbol{u}_k$ differs from $\boldsymbol{u}_k^0$ in the null space of the task Jacobian [143, 144], i.e. by satisfying the following constraint

$$\boldsymbol{J}_k\left(\boldsymbol{u}_k - \boldsymbol{u}_k^0\right) = 0 \tag{5.17}$$

Within an optimization based framework this can be handled by introducing a second optimization stage acting in the task null space, which inherits all the constraints in (5.14) together with the optimality criterion in (5.17)

$$\begin{gathered}\min_{\boldsymbol{u}_k} \frac{1}{2}\boldsymbol{u}_k^T\boldsymbol{Q}_u\boldsymbol{u}_k + \boldsymbol{g}_u^T\boldsymbol{u}_k \\ \text{subject to} \\ \boldsymbol{J}_k\boldsymbol{u}_k = \boldsymbol{J}_k\boldsymbol{u}_k^0 \\ \mathcal{A}_{\mathcal{I}}\boldsymbol{u}_k \preceq \boldsymbol{b}_{\mathcal{I}}\end{gathered} \tag{5.18}$$

where $\boldsymbol{Q}_u$ is a positive definite matrix and $\boldsymbol{g}_u$ is a vector of suitable dimension.

Several formulations of the cost function in (5.18) can be considered. By setting

$$\begin{aligned} \boldsymbol{Q}_u &= 2T_s^2 \boldsymbol{I}_n \\ \boldsymbol{g}_u &= 2T_s \dot{\boldsymbol{q}}_k \end{aligned} \tag{5.19}$$

joint space velocities $\dot{\boldsymbol{q}}_{k+1}$ can be minimized, as in the following of this thesis. Alternatively, kinematic redundancy can be exploited for collision-avoidance purposes, including self-collision avoidance. Being $\dot{\boldsymbol{q}}_k^0$ the vector of evasive joint displacements, defined as in [145], the candidate cost function in (5.18) is given by

$$\begin{aligned} \boldsymbol{Q}_u &= 2T_s^2 \boldsymbol{I}_n \\ \boldsymbol{g}_u &= 2T_s \left(\dot{\boldsymbol{q}}_k - \dot{\boldsymbol{q}}_k^0 \right) \end{aligned} \tag{5.20}$$

As another option, the robot reflected mass could be minimized, according to [146], in order to decrease the dissipated energy in potential inelastic impacts.

The sequence of QP problems representing the *reactive controller* module, i.e. (5.14) and (5.18), has been solved with QPOASES [84], or alternatively with the state-of-the-art hierarchical solver presented in [144]. Finally, Reflexxes Motion Libraries [88] have been chosen as real-time *trajectory generation* algorithm.

5.2.2 Robust Constraints Specification

As proposed in [48, 147], possible uncertainties in the description of the system kinematics/dynamics or in sensor measurement can be considered within the constraint specification

Consider a scalar task function $p = p(\boldsymbol{q}) \in \mathbb{R}$ with relative degree equal to 2. The time derivatives of p are given by

$$\dot{p} = \boldsymbol{J}_p(\boldsymbol{q})\, \dot{\boldsymbol{q}}, \quad \ddot{p} = \dot{\boldsymbol{J}}_p(\boldsymbol{q})\, \dot{\boldsymbol{q}} + \boldsymbol{J}_p(\boldsymbol{q})\, \boldsymbol{u} \tag{5.21}$$

while its dynamics can be represented by the state vector $\boldsymbol{\pi} \in \mathbb{R}^2$

$$\boldsymbol{\pi} = \begin{bmatrix} p & \dot{p} \end{bmatrix}^T \tag{5.22}$$

Defining $\Delta \in \mathbb{R}^2$ and $\boldsymbol{v} \in \mathbb{R}^2$ as two vectors accounting for state measurement uncertainty and modeling errors, respectively, the (discrete time) state vector $\boldsymbol{\pi}_{k+1}$ is given by the following double-integrator relation

$$\boldsymbol{\pi}_{k+1} = \boldsymbol{A}(\boldsymbol{\pi}_k + \Delta_k) + \boldsymbol{B}\ddot{p}_k + \boldsymbol{v}_k \tag{5.23}$$

where

$$\boldsymbol{A} = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \quad \boldsymbol{B} = \begin{bmatrix} 0.5T_s^2 \\ T_s \end{bmatrix} \tag{5.24}$$

T_s being the sampling time.
Consequently, by assuming Δ_k and $\boldsymbol{v}_k$ to be bounded and such that $\Delta_k \in \mathbb{D}$ and $\boldsymbol{v}_k \in \mathbb{V}$, where $\mathbb{D}$ and $\mathbb{V}$ are generic polytopes, the following *reachable set* can be obtained, representing the set of all possible values for the state vector $\boldsymbol{\pi}_{k+1}^F$ in (5.23)

$$\boldsymbol{\pi}_{k+1} \in \boldsymbol{A}\boldsymbol{\pi}_k \oplus \boldsymbol{B}\ddot{\mathbb{P}} \oplus \boldsymbol{A}\mathbb{D} \oplus \mathbb{V} \tag{5.25}$$

where $\oplus$ represents the Minkowski sum and

$$\ddot{\mathbb{P}} = \{\ddot{p} : |\ddot{p}| \leq \gamma_p,\ \gamma_p > 0\}$$

The relation between the different sets, accounting for the considered uncertainty, is depicted in Fig. 5.4.

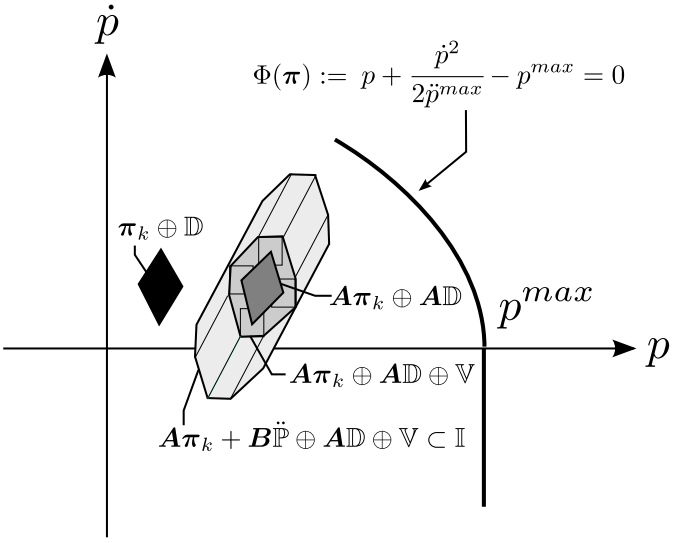


Figure 5.4: Geometric interpretation of (5.25)

In case an upper bound on p represents a control requirement

$$p_k \leq p^{max}, \forall k$$

the corresponding invariance function $\Phi(\boldsymbol{\pi})$ can be analytically computed according to (3.8) leading to

$$\Phi(\boldsymbol{\pi}) = \begin{cases} p - p^{max} & \dot{p} \leq 0 \\ p + \frac{\dot{p}^2}{2\gamma_p} - p^{max} & \dot{p} > 0 \end{cases}$$

The sub-domain $\mathbb{I}$ of the space $\boldsymbol{\pi}$ bounded by $\Phi(\boldsymbol{\pi}) = 0$ is shown in Fig. 5.4. To make the set $\mathbb{I}$ robustly positive invariant with respect to the given uncertainties, the controller must select a value $\ddot{p}_k$ satisfying the set invariance condition in (3.15), such that $\boldsymbol{\pi}_{k+1} \in \mathbb{I}$.
This can be achieved by monitoring the value of the invariance function $\Phi(\boldsymbol{\pi})$ computed with respect to the future state vector $\boldsymbol{\pi}_{k+1}$, i.e.

$$\Phi(\boldsymbol{\pi}_{k+1}), \forall\, \boldsymbol{\pi}_{k+1} \in \boldsymbol{A}\boldsymbol{\pi}_k \oplus \boldsymbol{B}\ddot{\mathbb{P}} \oplus \boldsymbol{A}\mathbb{D} \oplus \mathbb{V} \tag{5.26}$$

Since the reachable set in (5.25) is convex, see Fig. 5.4, it is sufficient to evaluate the invariance function on the vertexes of the reachable state.
Finally, according to *invariance control*, whenever $\Phi(\boldsymbol{\pi}_{k+1}) = 0$, positive invariance of the set $\mathbb{I}$ is ensured by condition (3.15), i.e.

$$\ddot{p}_k \leq -\gamma_p,\ \forall k | \Phi(\boldsymbol{\pi}_{k+1}) = 0$$

which can be translated in the following constraint on the optimization variable through (5.21)

$$\boldsymbol{J}_p\left(\boldsymbol{q}_k\right)\boldsymbol{u}_k \leq -\gamma_p - \dot{\boldsymbol{J}}_p\left(\boldsymbol{q}_k\right)\dot{\boldsymbol{q}}_k \tag{5.27}$$

Since this constraint is linear in the optimization variable, it is consistent with the generic constraint formulation in (5.12) and can be therefore plugged into (5.13).

CHAPTER 6

Force Controlled Bimanual Robotic Assembly Based on Trajectory Generation

Parts assembly represents the natural application in industrial settings of bimanual and light-weight robotic systems, like the experimental platform considered in this chapter (see Sec. 5.1). The inherent compliance of this type of manipulators provides an intrinsic degree of safety towards manipulated objects, yet negatively affecting position accuracy, especially during tasks in which interaction between the robot and the environment is involved. For these reasons, while dual-arm robots seem particularly suited for assembly tasks, the application of force control algorithms is even more crucial for a successful task execution compared to traditional industrial robots.

In this chapter a bimanual robotic assembly task is translated in an equivalent trajectory generation control problem fulfilling force control requirements. The constraint-based trajectory generation control framework described in Sec. 5.2 will be exploited for this purpose, while estimation of the interaction force/torque (see Sec. 5.1.3) will enable force sensorless execution of the assembly operation.
After some state-of-the-art examples of bimanual robotic assembly provided in Sec. 6.1, together with a description of the considered assembly use case, the proposed trajectory generation approach to force controlled bimanual assembly is developed and experimentally validated in a peg-in-

hole insertion problem (Sec 6.2) and in a cap rotation problem (Sec. 6.3).

6.1 Bimanual Robotic Assembly

Robotic assembly [148] involves a variety of autonomous operations, e.g. insertion [44,149], folding [150], capping/tapping [151,152], where the capabilities of a bimanual robotic system can be fully exploited. For example, by employing a dual-arm robot in a work-cell, the design of task specific fixtures is no longer required, compared to a traditional single arm system. Yet, not so many examples of bimanual parts assembly can be found in the literature.
Bimanual robots have been first employed for automation of the assembly of automotive parts. A pre-programmed gearbox assembly is described in [153], while [154] describes programming by demonstration for similar tasks. Another example from the automotive industry is given in [155] for power-train assembly, whereas an example from the construction industry is given in [156]. In [157], the authors propose a motion planning method for dual-arm robots specifically developed for bimanual assembly. The experimental validation on a peg-in-hole insertion problem has been implemented in a dual robot setting. A bimanual folding task is performed in [150] by means of a switched controller combined with a Kalman filter estimator, which tracks the contact point between the sliding parts. In [158] a dexterous origami-folding task is performed by a dual-arm manipulator equipped with multi-fingered robot hands. Additionally, manipulation of highly deformable objects [159] or material reshaping [160] can be performed with a bimanual platform.
Of particular interest is the work of Stolt et al. [45, 46], where force controlled assembly of an emergency stop button has been performed on an ABB YuMi robot, the experimental platforms used in this chapter. In [47], the authors propose a sensorless estimation of the interaction forces at the end effector from the joint position control errors. The method has been experimentally verified in a small part assembly task.

The assembly use case considered in this chapter consists in the bimanual assembly of an Eppendorf combitip® plastic pipette, generally used in laboratory automation. An assembly graph for the assembly task is displayed in Fig. 6.1.

The assembly can be divided in two parts:

- The plunger should be initially inserted into the body of the pipette.

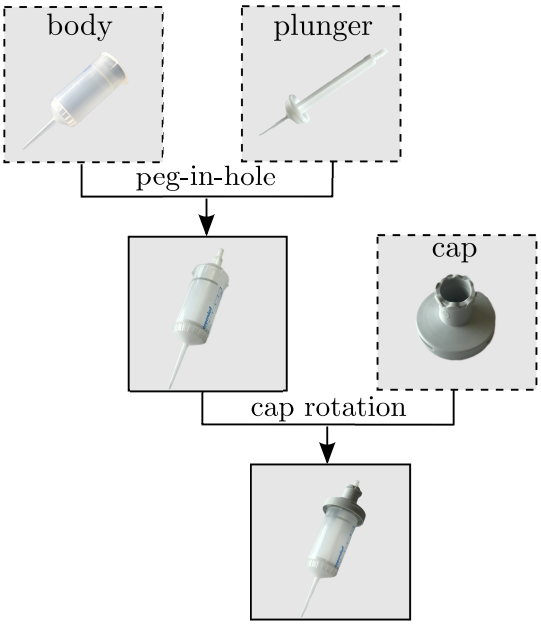


Figure 6.1: Assembly graph for the Eppendorf combitip® pipette assembly scenario.

This is a typical peg-in-hole insertion problem that will be addressed in Sec. 6.2.

- Once the plunger has been inserted, a cap rotation task, treated in Sec. 6.3, should be performed in order to screw the cap on top of the pipette body. A bayonet mount ensures the fastening between the cap and the body.

6.2 Bimanual Peg-in-Hole Insertion

This section addresses the peg-in-hole control problem for a dual-arm robot by merging *admittance control* (i.e. the implementation of *impedance control* for position-controlled robots, see Sec. 1.1) with constraint-based trajectory generation. After a brief introduction on typical control approaches to peg-in-hole insertion, the proposed control method [161] is presented in details and experimentally validated on the ABB YuMi robot.

6.2.1 Introduction

Peg-in-hole insertion is a traditional control problem that requires the adoption of force control methods. Its clear relevance in many industrial assem-

bly tasks makes it one of the most addressed topics in robot force control since the birth of this research field. The main approaches to the peg-in-hole problem are traditionally divided in two categories.

The first category includes the so called *passive* approaches relying on the adoption of a Remote Center of Compliance (RCC) device [2] located on the robot gripper. In this respect [162] Kim et al. stress the importance of the location of the compliance center and the coupling stiffness element between the translational and the rotational direction for the case of a multi-fingered robot hand. Alternatively, Yun [163] shows the advantage of passive compliance within a reinforcement learning approach for a 2D peg-in-hole problem.

The second category of methods accomplish the peg-in-hole insertion task in an *active* way. Wook et al. [164] use a force/torque sensor within a control algorithm that enables the robot to continue the task even with large directional errors. In [165], the authors exploit a geometric description of the environment, including uncertainty, for assembly operations as a constraint to shape compliant motion strategies. Bruyninckx et al. [166] specify the alignment motion for arbitrarily large alignment errors, based on a model of the peg-in-hole contact situation.
Search strategies for peg-in-hole assemblies with position uncertainty are proposed in [167], whereas in [168] the peg-in-hole task is approached by employing hybrid control. In [169] an active method using force feedback is proposed for a fixtureless peg-in-hole assembly in highly uncertain environment using two manipulators. Assembly states are described by extended contact relations and qualitative models for event evolution are derived from the marking of the Petri net model. Tsumugiwa et al. [170] propose an admittance control in a carrying task, while torque control is used in a fitting task. In [171] the authors propose a method to insert a long peg into a tandem shallow hole using search trajectory generation without force feedback.
Alternative approaches combine force/torque and *visual* sensors. Su et al. [172], present a vision-based and sensorless eccentric peg-in-hole insertion strategy, based on an attractive region constructed in the configuration space. More recently, Huang et al. [173], inspired by the so-called *visual compliance control*, propose a visual servoing approach with a single eye-in-hand high-speed camera and a visual compliance strategy to deal with the problem of fast peg-in-hole alignment with large position and attitude uncertainty [174].

6.2.2 Trajectory Generation of Compliant Motion

In order to accomplish a peg-in-hole insertion task, a compliant behavior for the robot is required due to the interaction force/torque arising from the misalignment between the peg and the hole. In this respect and assuming the robotic system to be position-controlled (as the YuMi experimental platform), constraint-based trajectory generation can ensure a compliant robot motion during a peg insertion by exploiting *admittance control*.

Consider a 1-DOF peg-in-hole task as sketched in Figure 6.2, where θ represents the actual orientation of the peg, θ_{hole} is the actual orientation of the hole, whilst θ^{trg} represents an available estimate of the hole orientation, in order to consider additional uncertainty in the insertion problem.

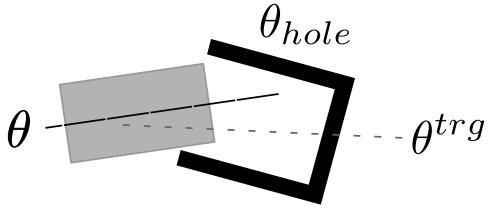


Figure 6.2: Schematic representation of a 1-DOF peg-in-hole task

Due to the misalignment between the peg and the hole, a torque μ arises, according to the Kelvin-Voigt linear model

$$\mu = -K^{env}\left(\theta - \theta_{hole}\right) - D^{env}\dot{\theta} \tag{6.1}$$

where K^{env} and D^{env} are the environment rotational stiffness and damping, respectively.
The aim of the proposed control approach [161] is to ensure a compliant alignment between the peg orientation θ and the actual hole orientation θ_{hole} by combining *trajectory generation* and *admittance control* within the constraint-based trajectory generation framework described in Sec. 5.2.
Two implementations for the *trajectory generation* module, hereafter referred to as *position based* and *velocity based* trajectory generation, are considered in the following:

Position based trajectory generation: the next state of motion for the peg, in terms of peg orientation and angular velocity reference, θ^{gen}_{k+1} and $\dot{\theta}^{gen}_{k+1}$, respectively, is produced by the trajectory generation algorithm selecting as target state of motion the estimated hole orientation θ^{trg} and zero angular velocity $\dot{\theta}^{trg} = 0$, see Fig 6.3.

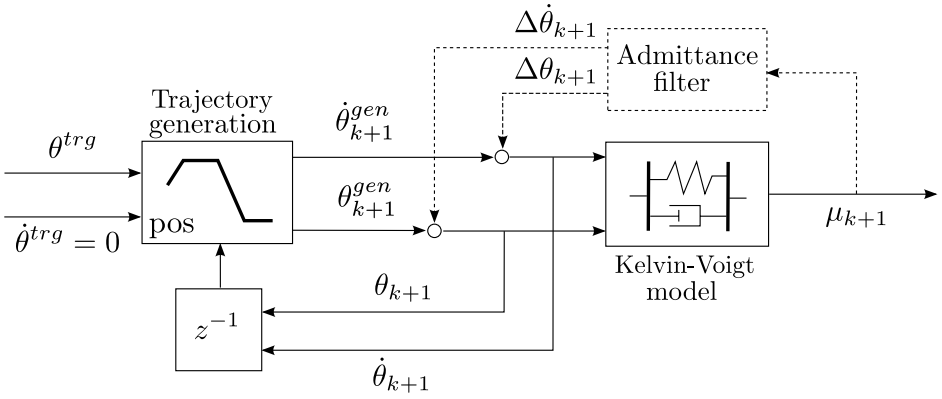


Figure 6.3: 1-DoF proposed control approach: admittance control combined with a *position based* trajectory generation algorithm.

Velocity based trajectory generation: as shown in Fig 6.4, the next state of motion for the peg, θ^{gen}_{k+1} and $\dot{\theta}^{gen}_{k+1}$, is produced by the trajectory generation algorithm selecting as target state of motion zero angular velocity $\dot{\theta}^{trg} = 0$.

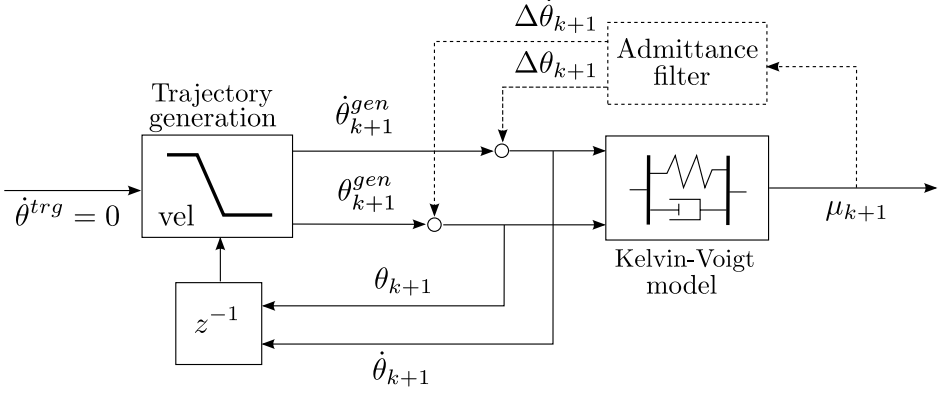


Figure 6.4: 1-DoF proposed control approach: admittance control combined with a *velocity based* trajectory generation algorithm.

In order to combine *trajectory generation* and *admittance control*, the peg orientation and angular velocity references provided by the trajectory generation module, θ^{gen}_{k+1} and $\dot{\theta}^{gen}_{k+1}$, respectively, are added to the orientation and angular velocity increments produced by an *admittance filter*, $\Delta\theta_{k+1}$

and $\Delta\dot{\theta}_{k+1}$, respectively.

$$\begin{aligned} \theta_{k+1} &= \theta_{k+1}^{gen} + \Delta\theta_{k+1} \\ \dot{\theta}_{k+1} &= \dot{\theta}_{k+1}^{gen} + \Delta\dot{\theta}_{k+1} \end{aligned} \tag{6.2}$$

Denoting with H and D the filter mass and damping coefficients, respectively, the corresponding mechanical impedance equation is given by

$$\mu_k = H\Delta\ddot{\theta}_k + D\Delta\dot{\theta}_k \tag{6.3}$$

The filter state can be described by the following discrete-time double integrator dynamics

$$\begin{aligned} \Delta\dot{\theta}_{k+1} &= \Delta\dot{\theta}_k + T_s\Delta\ddot{\theta}_k \\ \Delta\theta_{k+1} &= \Delta\theta_k + T_s\Delta\dot{\theta}_k + \frac{T_s^2}{2}\Delta\ddot{\theta}_k \end{aligned} \tag{6.4}$$

where T_s is the controller sampling time.
By retrieving $\Delta\ddot{\theta}_k$ from (6.3) and plugging it into (6.4), the orientation and angular velocity outputs of the admittance filter are finally given by

$$\begin{aligned} \Delta\dot{\theta}_{k+1} &= \Delta\dot{\theta}_k + T_s H^{-1}\left(\mu_k - D\Delta\dot{\theta}_k\right) \\ \Delta\theta_{k+1} &= \Delta\theta_k + T_s\Delta\dot{\theta}_k + \frac{T_s^2}{2}H^{-1}\left(\mu_k - D\Delta\dot{\theta}_k\right) \end{aligned} \tag{6.5}$$

Due to the nonlinear nature of the *position based* trajectory generation algorithm, simulation analysis has been performed in order to evaluate the closed-loop performance of the control schemes in Fig. 6.3 and Fig. 6.4. As shown in the top plot of Fig. 6.5, steady-state error arises between the peg and the hole orientation when *position based* trajectory generator is employed. On the other hand, with the *velocity based* trajectory generation algorithm, zero steady-state error can be achieved, making this control implementation the most suitable option. The corresponding (simulated) time evolution of the peg orientation and angular velocity, together with the time history of the interaction torque are shown in the lower plots of Fig. 6.5.

Selection of the admittance parameters, H and D, ensuring closed-loop system stability can be performed by considering the following *velocity based* trajectory generation algorithm, where saturation in the angular ac-

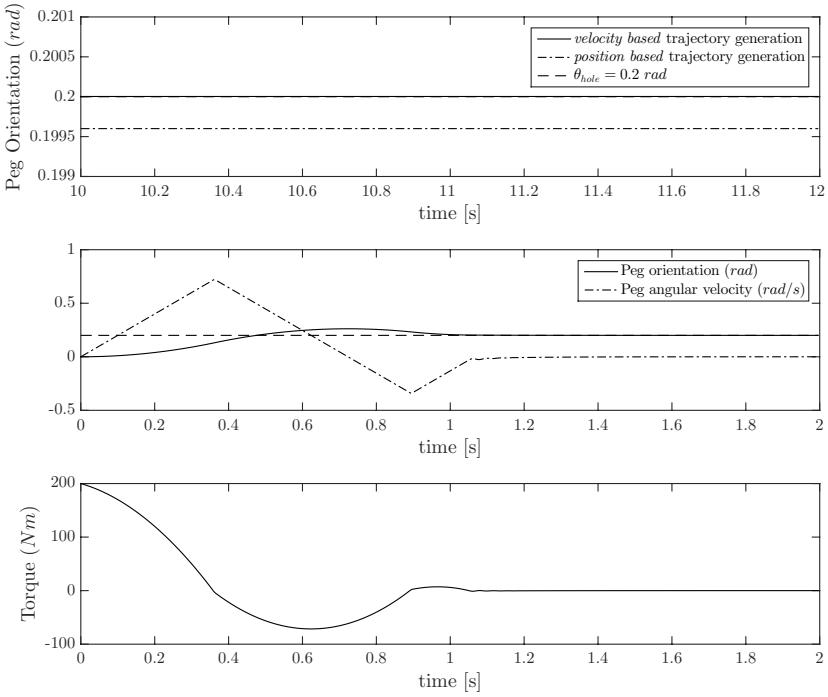


Figure 6.5: Top plot: simulated time evolution of peg orientation at steady-state with *velocity based* trajectory generation (solid line) and *position based* trajectory generation (dashed-dotted line). Middle plot: simulated time evolution of peg orientation (solid line) and angular velocity (dashed-dotted line) with respect to hole orientation (dashed line) with *velocity based* trajectory generation. Bottom plot: simulated time evolution of the interaction torque μ with *velocity based* trajectory generation. Simulation parameters: $K^{env} = 500\ Nm/rad$, $D^{env} = 100\ Nms/rad$, $H = 0.5\ kgm^2/rad$, $D = 50\ Nms/rad$, $\theta^{trg} = 0\ rad$, $\theta_{hole} = 0.2\ rad$, $T_s = 0.004\ s$.

celeration $\ddot{\theta}^{gen}$ has been neglected

$$\begin{aligned} \ddot{\theta}^{gen}_{k+1} &= T_s^{-1}(\dot{\theta}^{trg} - \dot{\theta}_k) = -T_s^{-1}\dot{\theta}_k \\ \dot{\theta}^{gen}_{k+1} &= \dot{\theta}_k + T_s\ddot{\theta}^{gen}_{k+1} = 0 \\ \theta^{gen}_{k+1} &= \theta_k + T_s\dot{\theta}_k + 0.5\,T_s^2\,\ddot{\theta}^{gen}_{k+1} = \theta_k \end{aligned} \tag{6.6}$$

Consider as state vector $\boldsymbol{x}_k = \begin{bmatrix}\theta_k & \dot{\theta}_k & \Delta\theta_k & \Delta\dot{\theta}_k\end{bmatrix}$. A state-space representation of the closed-loop system can be obtained by employing equations (6.1),(6.2), (6.5), (6.6). Consequently, Jury stability criterion can be applied in order to obtain necessary and sufficient conditions for the sta-

bility of the closed-loop system expressed with respect to the admittance parameters, leading to

$$\begin{aligned} D + D^{env} &> K^{env} T_s \\ H &> T_s \frac{4\,(D + D^{env}) - K^{env} T_s}{8} \end{aligned} \tag{6.7}$$

6.2.3 Task Specification and Assembly Sequence

The kinematic representation adopted for the peg-in-hole insertion is detailed in the following, together with a finite state machine model [175] of the task execution.

A *leader-follower approach* [176] has been employed so that a relative motion command is generated for the *follower* robot arm (the one holding the peg) with respect to the *leader*'s arm state of motion (the one holding the hole). The following vector of task variables $\boldsymbol{x} \in \mathbb{R}^5$ has been chosen as state vector

$$\boldsymbol{x} = \begin{bmatrix} x & y & z & \phi & \theta \end{bmatrix}^T$$

representing the linear and angular displacements (XY Euler angles) of the *follower* frame with respect to the *leader* frame, see Fig. 6.13.

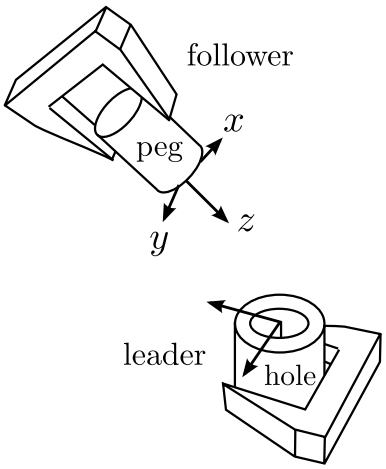


Figure 6.6: Frames for the peg-in-hole bimanual task

Due to the cylindrical symmetry of the two workpieces, the Z Euler angle does not participate in the task description, thus resulting in a redundant degree of freedom.

The complete task, shown in Fig. 6.7, is described in the following.
The peg longitudinal axis is hereafter assumed to be aligned with the *follower* z direction.

1. *Approach with alignment* - the *follower* end effector reaches the initial pose $\boldsymbol{x}^{trg}$, corresponding to an approximate alignement between the peg and the hole;

2. *Compliant motion* - the *follower* insertion motion along the z direction is produced by the *velocity based* trajectory generation module. Simultaneously, a compliant behavior along the other directions (x, y, ϕ, θ) is ensured by the combination of trajectory generation and *admittance control* described in the previous section;

3. Whenever the force in z direction exceeds a given bound ($F_z^{thr,up}$), meaning that the peg has been completely inserted, the robot enters a *Stopping motion* state and starts decelerating towards zero target velocity in z direction;

4. In order to prevent false positives, the robot continues its stopping motion unless the force in z direction drops under a certain threshold ($F_z^{thr,down}$);

5. The task terminates in the state *Opening gripper*.

6.2.4 Control Implementation

During the *Approach with alignment* phase, Reflexxes Motion Libraries [88] have been exploited as real-time trajectory generation algorithm selecting the initial pose $\boldsymbol{x}^{trg}$ as target position and $\dot{\boldsymbol{x}}^{trg} = 0$ as target velocity.

In order to merge the proposed peg-in-hole control approach within the constraint-based trajectory generation control framework described in Sec. 5.2, the following cost function is employed in Algorithm 2 during the *Compliant motion* phase

$$\min_{\boldsymbol{u}_k} \mathcal{L}\left(\dot{\boldsymbol{x}}_{k+1} - \dot{\tilde{\boldsymbol{x}}}_{k+1}^{gen}, \boldsymbol{x}_{k+1} - \tilde{\boldsymbol{x}}_{k+1}^{gen}\right) \tag{6.8}$$

where, according to (6.2)

$$\begin{aligned} \tilde{\boldsymbol{x}}_{k+1}^{gen} &= \boldsymbol{x}_{k+1}^{gen} + \Delta\boldsymbol{x}_{k+1} \\ \dot{\tilde{\boldsymbol{x}}}_{k+1}^{gen} &= \boldsymbol{x}_{k+1}^{gen} + \Delta\dot{\boldsymbol{x}}_{k+1} \end{aligned} \tag{6.9}$$

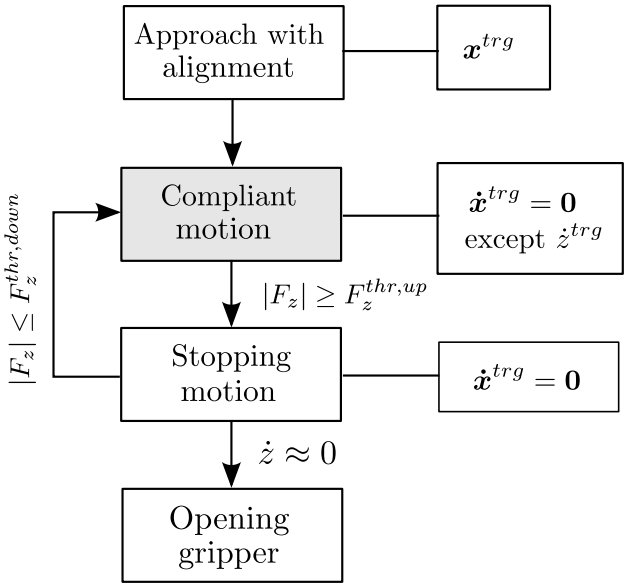


Figure 6.7: State machine governing the peg-in-hole task execution

being $\boldsymbol{x}_{k+1}^{gen},\ \boldsymbol{x}_{k+1}^{gen} \in \mathbb{R}^5$ the output of the *velocity based* trajectory generation module computed by Algorithm 3, while $\Delta\boldsymbol{x}_{k+1},\ \Delta\dot{\boldsymbol{x}}_{k+1} \in \mathbb{R}^5$ are the position and velocity increments, respectively, produced by the *admittance filter*.

Algorithm 3 *Velocity based* trajectory generation

Input: $\boldsymbol{x}_k, \dot{\boldsymbol{x}}_k, \ddot{\boldsymbol{x}}^{max}, \dot{\boldsymbol{x}}^{trg}$
Output: $\boldsymbol{x}_{k+1}^{gen}, \dot{\boldsymbol{x}}_{k+1}^{gen}$

if $\left|\dot{\boldsymbol{x}}^{trg} - \dot{\boldsymbol{x}}_k\right| / \ddot{\boldsymbol{x}}^{max} \leq T_s$ **then**
    $\ddot{\boldsymbol{x}}_k = \left(\dot{\boldsymbol{x}}^{trg} - \dot{\boldsymbol{x}}_k\right) / T_s$
else
    $\ddot{\boldsymbol{x}}_k = \text{sign}\left(\dot{\boldsymbol{x}}^{trg} - \dot{\boldsymbol{x}}_k\right) \ddot{\boldsymbol{x}}^{max}$
end if
$\dot{\boldsymbol{x}}_{k+1}^{gen} = \dot{\boldsymbol{x}}_k + T_s \ddot{\boldsymbol{x}}_k$
$\boldsymbol{x}_{k+1}^{gen} = \boldsymbol{x}_k + T_s \dot{\boldsymbol{x}}_k + 0.5 T_s^2 \ddot{\boldsymbol{x}}_k$

Following from (6.5), $\Delta\boldsymbol{x}_{k+1},\ \Delta\dot{\boldsymbol{x}}_{k+1}$ are hence given by

$$\begin{aligned} \Delta\dot{\boldsymbol{x}}_{k+1} &= \Delta\dot{\boldsymbol{x}}_k + T_s \boldsymbol{M}^{-1} \left(\boldsymbol{\mu}_k - \boldsymbol{D}\Delta\dot{\boldsymbol{x}}_k\right) \\ \Delta\boldsymbol{x}_{k+1} &= \Delta\boldsymbol{x}_k + T_s \Delta\dot{\boldsymbol{x}}_k + 0.5 T_s^2 \boldsymbol{M}^{-1} \left(\boldsymbol{\mu}_k - \boldsymbol{D}\Delta\dot{\boldsymbol{x}}_k\right) \end{aligned} \tag{6.10}$$

where $\boldsymbol{M}$ and $\boldsymbol{D}$ are the positive definite design matrices that set the desired mechanical impedance. The state of the admittance filter is then updated as

$$\Delta \boldsymbol{x}_{k+1} = \tilde{\boldsymbol{x}}_{k+1}^{ref} - \boldsymbol{x}_{k+1}$$
$$\Delta \dot{\boldsymbol{x}}_{k+1} = \dot{\tilde{\boldsymbol{x}}}_{k+1}^{ref} - \dot{\boldsymbol{x}}_{k+1}$$

Finally, minimization of joint space velocities $\dot{\boldsymbol{q}}_{k+1}$ has been chosen as candidate cost function in (5.18) to handle kinematic redundancy, see (5.19). The corresponding sequence of QP problems within the *reactive controller* module has been solved by QPOASES software [84].

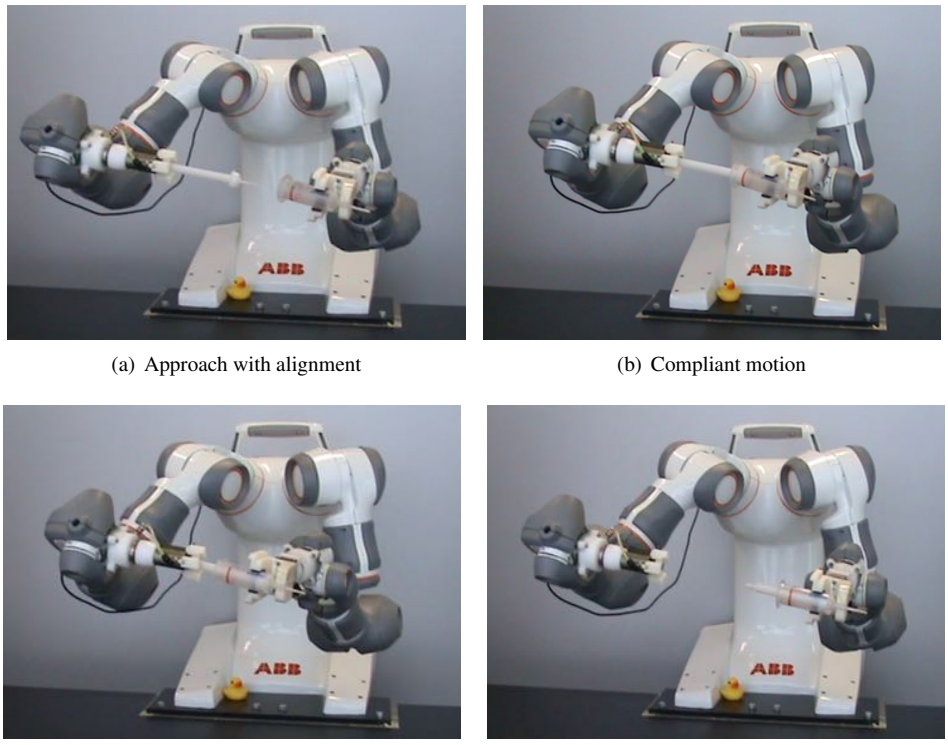
6.2.5 Experimental Results

The peg-in-hole task for the considered assembly use case consists in the insertion of a plunger into the body of a plastic pipette, see Fig. 6.8.

Figure 6.8: Experimental setup: the ABB YuMi robot used in this thesis equipped with Schunk pneumatically actuated grippers and 3D printed tools to perform the peg-in-hole insertion of an Eppendorf combitip® plastic pipette.

As no force sensor is available, force estimation has been used as previously described in Sec. 5.1.3.
Controller robustness to peg/hole misalignment has been evaluated by adding a position offset (x direction) and an orientation offset (ϕ Euler angle) calculated with respect to the target end effector pose $\boldsymbol{x}^{trg}$ in the *Approach with alignment* state.
An insertion velocity $\dot{z}^{trg}$ of 75 mm/s (see Fig. 6.7) has been chosen in

order to achieve a task execution speed comparable to the execution speed of a human operator.
Selection of the admittance parameters has been performed according to (6.7), based on approximate knowledge of the environment (i.e. plastic) stiffness and damping. The following values have been used: $H = 0.2$ kg, $H^{rot} = 0.1\ kgm^2/rad$ for the mass and rotational inertia, respectively, $D = 15\ Ns/m$, $D^{rot} = 15\ Nms/rad$ for the linear and rotational damping, respectively.
Snapshots of a complete experiment and corresponding time histories of the *follower*-side estimated interaction force and end effector velocity are shown in Fig. 6.9 and Fig. 6.10, respectively.



(a) Approach with alignment

(b) Compliant motion

(c) Stopping motion

(d) Opening the gripper

Figure 6.9: Snapshots from a peg-in-hole experiment.

As shown in Fig. 6.10, the robot successfully performs the compliant peg-in-hole insertion in approximately $1.25\ s$ (red vertical line), achieving a considerable task completion speed. At time instant $t = 1.2\ s$ the absolute value of the force in the z direction exceeds the threshold $F_z^{thr,up} = 1\ N$

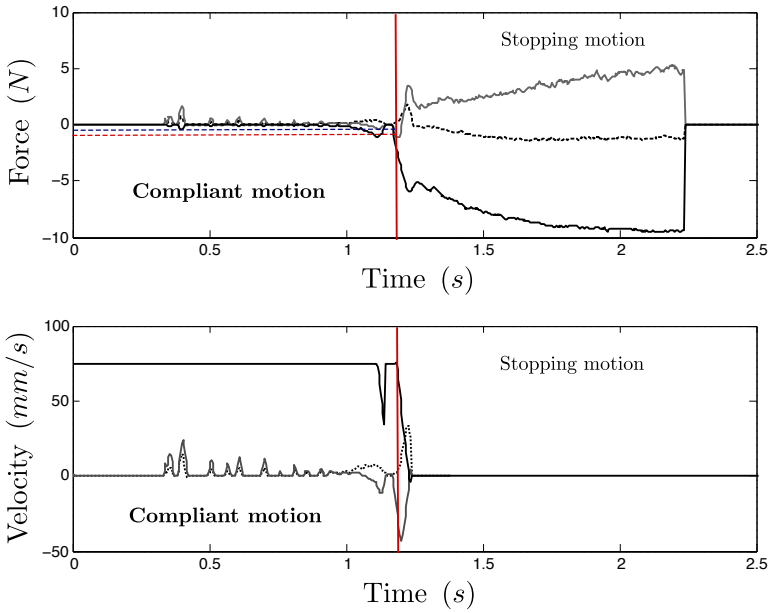


Figure 6.10: Time histories of the *follower* estimated interaction force and corresponding end effector linear velocity during the *Compliant motion* and *Stopping motion* phase (x: solid gray line, y: dotted black line, z: solid black line).

(red dashed horizontal line) and the task execution enters the *Stopping motion* state, consistently with the state machine sequence in Fig. 6.7. Nevertheless, as soon as the force drops below the threshold $F_z^{thr,down} = 0.5\ N$(blue dashed horizontal line), the robot resumes the *Compliant motion* state, until at time instant $t = 1.25\ s$ the peg-in-hole insertion terminates and the robot starts the *Stopping motion*.

		Position offset		
		3 mm	6 mm	9 mm
Orientation offset	10 deg	5/5	5/5	4/5
	15 deg	-	5/5	5/5
	20 deg	4/6	4/5	5/5

Table 6.1: Success rate of the performed peg-in-hole experiments for different combinations of position and orientation offset.

Additional experiments have been performed varying the initial peg/hole misalignment, see Table 6.1. A significant success rate has been obtained even in presence of a severe misalignment, i.e. $9\ mm$ position offset (equal to the 35% of the pipette body inner diameter) and $20\ deg$ angular displacement.

6.3 Bimanual Cap Rotation

Amongst the bimanual manipulation control problems, screwing/unscrewing and capping/tapping tasks [151, 152], or generally cap rotation tasks [177], are considered challenging topics, only recently addressed in robotics research.
In [178] a sensor-based controller allows a dual-arm robot system to unscrew the cap of a target object. The unscrew primitive is divided in three phases: grasp, unscrew, cap removal. The unscrew phase consists in applying a twisting motion to the grasping fingers, once a certain grasping force has been established based on the tactile sensors of each finger.
In [179], the cap unscrewing problem with a two-fingered dual-arm robot has been addressed. The control strategy maximizes the set of object poses that can be handled by the system based on observed human arm movements, while a learning approach extracts from human training data a manifold of finger synergies to carry out the task.
Although the mentioned control approaches provide effective solutions to the cap rotation problem, the amount of sensor information required on one side [178] and the complexity of the overall control process on the other [179], clearly limit their applicability on industrial scale.

In this section, constraint-based trajectory generation, see Sec. 5.2, is exploited in order perform a bimanual cap rotation task, where contact loss can be prevented by adding a constraint on the interaction force (arising during the screwing phase) to the motion planning problem. The proposed control method [180] is presented in detail and experimentally validated on the considered assembly use case employing the ABB YuMi robot.

6.3.1 Robust Force Constraint Specification

In order to perform a cap rotation task, loss of contact between the *cap* and the corresponding *target object*, see Fig. 6.11, is a relevant control issue that needs to be addressed during the screwing phase. In this respect, a successful task execution can be ensured by conveniently bounding the interaction force F_c arising from the contact along the screwing rotation axis by means of an equivalent constraint to be considered within the set of constraints (5.13) in Algorithm 2. Extending the results in Sec. 5.2.2, force measurement noise and surface geometric uncertainties can be considered within the constraint specification, together with bounded uncertainty in the environment stiffness.

It is hereafter assumed that the screwing phase begins once a contact with alignment has been established between the cap and the corresponding target object. In addition, the cap grasped by the robot end effector is considered rigid with respect to the target object, modeled as a purely elastic environment.

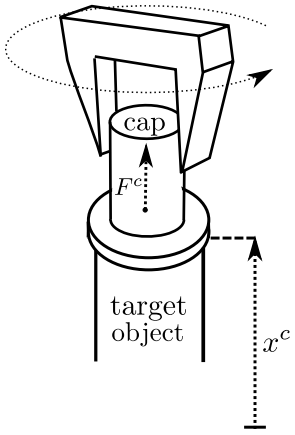


Figure 6.11: Example of a cap rotation task.

With reference to Fig. 6.11, the relation between the robot end effector position (accounting for the tool and cap sizes) $x_c \in \mathbb{R}$ and the contact force $F_c \in \mathbb{R}$, along the direction constrained by the environment, i.e. the screwing rotation axis, is modeled as an equivalent spring at the end effector and is given by

$$F_c = -K(x_c - x_0) \tag{6.11}$$

where $x_0 \in \mathbb{R}$ is the nominal undeformed pose on the target object surface in the constrained direction, while $K \in \mathbb{R}$ is the environment stiffness depending on the object material.

In order to prevent contact loss between the parts during the cap screwing phase, the controller must satisfy an upper and lower bound on the interaction force F_c.
According to Sec. 5.2.2 and assuming that (6.11) has relative degree 2, the state vector $\boldsymbol{\pi}^F \in \mathbb{R}^2$ related to (6.11) is given by

$$\boldsymbol{\pi}^F = \begin{bmatrix} F_c & \dot{F}_c \end{bmatrix}^T \tag{6.12}$$

Following from (6.11)

$$\boldsymbol{\pi}^F = \begin{bmatrix} F_c & -K\,\dot{x}_c \end{bmatrix}^T = \begin{bmatrix} 1 & 0 \\ 0 & -K \end{bmatrix} \begin{bmatrix} F_c & \dot{x}_c \end{bmatrix}^T \tag{6.13}$$

where $\dot{x}_c = \boldsymbol{J}_c(\boldsymbol{q})\,\dot{\boldsymbol{q}}$.
By introducing the new state vector $\hat{\boldsymbol{\pi}}^F \in \mathbb{R}^2$

$$\hat{\boldsymbol{\pi}}^F = \begin{bmatrix} F_c & \dot{x}_c \end{bmatrix}^T \tag{6.14}$$

equation (6.13) can be written as

$$\boldsymbol{\pi}^F = \begin{bmatrix} 1 & 0 \\ 0 & -K \end{bmatrix} \hat{\boldsymbol{\pi}}^F \tag{6.15}$$

The corresponding discrete-time state vector $\boldsymbol{\pi}^F_{k+1}$ can be obtained employing the following double-integrator relation

$$\boldsymbol{\pi}^F_{k+1} = \boldsymbol{A}^F \hat{\boldsymbol{\pi}}_k + \boldsymbol{B}^F \ddot{x}_{c,k} \tag{6.16}$$

where the state matrices $\boldsymbol{A}^F$ and $\boldsymbol{B}^F$ now depend on the environment stiffness

$$\boldsymbol{A}^F(K) = \begin{bmatrix} 1 & K\,T_s \\ 0 & -K \end{bmatrix} \quad \boldsymbol{B}^F(K) = \begin{bmatrix} 0.5\,K\,T_s^2 \\ -K\,T_s \end{bmatrix} \tag{6.17}$$

T_s being the sampling time and

$$\ddot{x}_c = \dot{\boldsymbol{J}}_c(\boldsymbol{q})\,\dot{\boldsymbol{q}} + \boldsymbol{J}_c(\boldsymbol{q})\,\boldsymbol{u} \tag{6.18}$$

Assume now that only a rough knowledge of the interaction model in (6.11) is available, here represented by a bounded uncertainty in the environment stiffness

$$K \in \mathbb{K} := [K_{min}, K_{max}]$$

where K_{min} and K_{max} represent a lower and an upper bound, respectively. Consequently, the state matrices in (6.17) belong to the following sets

$$\mathbb{A}^F : \{\boldsymbol{A}^F(K) \in \mathbb{R}^{2\times 2} \mid K \in \mathbb{K}\},\ \ \mathbb{B}^F : \{\boldsymbol{B}^F(K) \in \mathbb{R}^2 \mid K \in \mathbb{K}\}$$

The terms $\Delta^F \in \mathbb{D}^F$ and $\boldsymbol{v}^F \in \mathbb{V}^F$ can be additionally considered, in order to account for force measurement noise and possible uncertainties on the surface position x_0, respectively. The following *reachable set* can be therefore computed for the state vector $\boldsymbol{\pi}^F_{k+1}$

$$\boldsymbol{\pi}^F_{k+1} \in \mathbb{A}^F \hat{\boldsymbol{\pi}}^F_k \oplus \mathbb{B}^F \ddot{\mathbb{X}}_c \oplus \mathbb{A}^F \mathbb{D}^F \oplus \mathbb{V}^F \tag{6.19}$$

being

$$\ddot{\mathbb{X}}_c = \{\ddot{x}_c : |\ddot{x}_c| \leq \gamma_x,\ \gamma_x > 0\}$$

The relation between the different sets, accounting for the considered uncertainties, is depicted in Fig. 6.12 and results in a butterfly shaped reachable set.

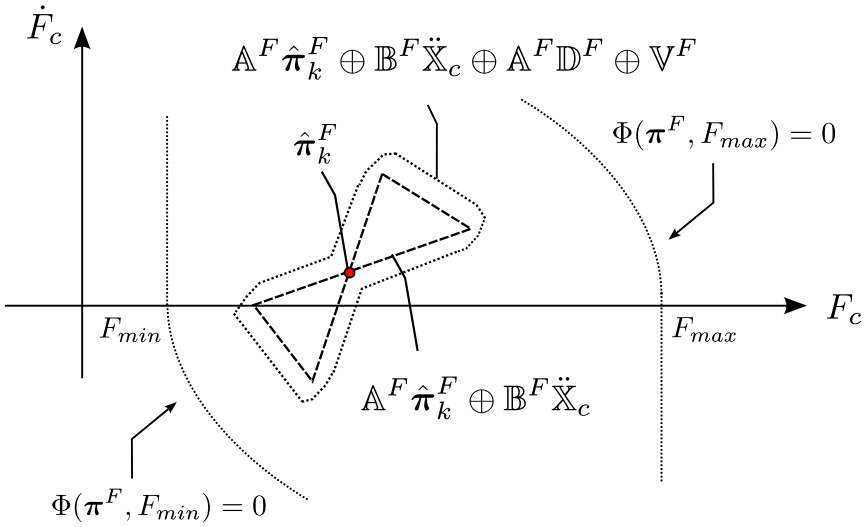


Figure 6.12: Geometric interpretation of condition (6.19).

Assume now that the controller must ensure a lower and upper bound on the interaction force

$$F_{min} \leq F_{c,k} \leq F_{max}, \forall k \tag{6.20}$$

The corresponding invariance functions $\Phi(\boldsymbol{\pi}^F, F_{max})$ and $\Phi(\boldsymbol{\pi}^F, F_{min})$ related to the constraints $F_{c,k} \leq F_{max}$ and $F_{c,k} \geq F_{min}$, respectively, can be analytically computed according to (3.8) for relative degree equal to 2, leading to

$$\Phi(\boldsymbol{\pi}^F, F_{max}) = \begin{cases} F_c - F_{max} & \dot{F}_c \leq 0 \\ F_c + \frac{\dot{F}_c^2}{2\ddot{F}_c^{max}} - F_{max} & \dot{F}_c > 0 \end{cases}$$

and

$$\Phi(\boldsymbol{\pi}^F, F_{min}) = \begin{cases} F_{min} - F_c & \dot{F}_c \geq 0 \\ -F_c + \frac{\dot{F}_c^2}{2\ddot{F}_c^{max}} + F_{min} & \dot{F}_c < 0 \end{cases}$$

The sub-domain $\mathbb{I}^F$ of the space $\boldsymbol{\pi}^F$ bounded by $\Phi(\boldsymbol{\pi}^F, F_{max}) = 0$ and $\Phi(\boldsymbol{\pi}^F, F_{min}) = 0$ is shown in Fig. 6.12.
To make the set $\mathbb{I}^F$ robustly positive invariant with respect to the given uncertainties, the controller must select a value $\ddot{x}_{c,k}$ satisfying the set invariance condition in (3.15), such that $\boldsymbol{\pi}_{k+1}^F \in \mathbb{I}^F$.
This can be achieved by monitoring the value of the invariance functions

$\Phi(\boldsymbol{\pi}^F, F_{max})$ and $\Phi(\boldsymbol{\pi}^F, F_{min})$ computed with respect to the future force state vector $\boldsymbol{\pi}^F_{k+1}$, i.e.

$$\Phi(\boldsymbol{\pi}^F_{k+1}, F_{max}), \forall\, \boldsymbol{\pi}^F_{k+1} \in \mathbb{A}^F \hat{\boldsymbol{\pi}}^F_k \oplus \mathbb{B}^F \ddot{\mathbb{X}}_c \oplus \mathbb{A}^F \mathbb{D}^F \oplus \mathbb{V}^F$$

and

$$\Phi(\boldsymbol{\pi}^F_{k+1}, F_{min}), \forall\, \boldsymbol{\pi}^F_{k+1} \in \mathbb{A}^F \hat{\boldsymbol{\pi}}^F_k \oplus \mathbb{B}^F \ddot{\mathbb{X}}_c \oplus \mathbb{A}^F \mathbb{D}^F \oplus \mathbb{V}^F$$

Despite the non-convexity of the reachable set in (5.25) (see Fig.6.12), by considering the convex hull of the reachable set [181], it is again sufficient to evaluate the invariance functions on the vertexes of the convex hull.
Finally, according to *invariance control*, positive invariance of the set $\mathbb{I}$ is ensured by the following conditions derived from (3.15)

$$\ddot{x}_{c,k} \leq -\gamma_c,\ \forall k | \Phi(\boldsymbol{\pi}^F_{k+1}, F_{max}) = 0$$

$$\ddot{x}_{c,k} \geq \gamma_c,\ \forall k | \Phi(\boldsymbol{\pi}^F_{k+1}, F_{min}) = 0$$

which can be translated in equivalent constraints on the optimization variable through (6.18)

$$\begin{aligned} \boldsymbol{J}_c(\boldsymbol{q}_k)\, \boldsymbol{u}_k &\leq -\gamma_x - \dot{\boldsymbol{J}}_c(\boldsymbol{q}_k)\, \dot{\boldsymbol{q}}_k \\ -\boldsymbol{J}_c(\boldsymbol{q}_k)\, \boldsymbol{u}_k &\leq -\gamma_x + \dot{\boldsymbol{J}}_c(\boldsymbol{q}_k)\, \dot{\boldsymbol{q}}_k \end{aligned} \tag{6.21}$$

Again, these constraints, linear in the optimization variable, are consistent with the generic constraint formulation in (5.12) and can be therefore plugged into (5.13).

6.3.2 Task Specification and Assembly Sequence

The kinematic representation adopted for the bimanual cap rotation task is detailed in the following, together with the finite state machine model [175] of the task execution.

Similarly to the peg-in-hole insertion, see Sec. 6.3.1, a *leader-follower approach* [176] has been employed to generate the motion commands for *follower* robotic arm (the one holding the cap) based on position and velocity of the *leader*'s arm (the one holding the target object).
The following vector of task variables $\boldsymbol{x} \in \mathbb{R}^6$ has been chosen as state vector

$$\boldsymbol{x} = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^T$$

representing the linear and angular displacements (in terms of XYZ Euler angles) of the *follower* frame with respect to the *leader* frame, see Fig.

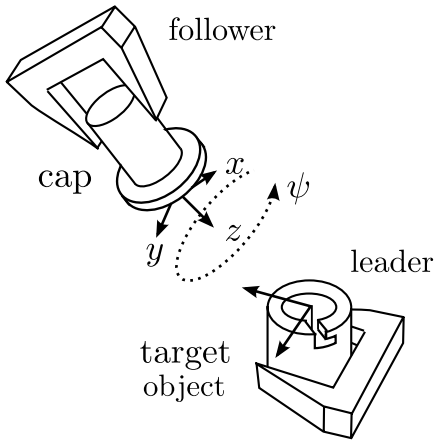


Figure 6.13: Frames for the bimanual cap rotation task. A bayonet mount ensures the fastening between the cap and the target object.

6.13.

The complete task, shown in Fig. 6.14, is described in the following. Thanks to the control approach presented in Section 6.3.1, a compliant alignment of the cap with respect to the target object is performed in order to establish contact between the parts. Consequently, the force controlled screwing operation takes place. The screwing axis is assumed to be aligned with the *follower* z direction, considered in the following as the force controlled direction. A bayonet mount ensures the fastening between the cap and the target object.

1. *Approach with alignment* - the *follower* end effector reaches the initial pose $\boldsymbol{x}^{trg}$, corresponding to an approximate alignment between the cap and the target object;

2. *Compliant motion* - the *follower* motion along the z direction is produced by a *velocity based* trajectory generation module. Simultaneously, a compliant behavior along the other directions $(x, y, \phi, \theta, \psi)$ is ensured by combining trajectory generation and admittance control, as described in Sec. 6.2.2;

3. *Force constrained rotation* - whenever the force in the z direction exceeds a given bound ($F_z^{thr,up}$), meaning that a steady contact has been established between the parts, the robot starts a velocity controlled rotation $\dot{\psi}$, while simultaneously constraining the force F_z within a lower and an upper bound F_{min} and F_{max}, respectively, as described

in Section 6.3.1.

4. When the norm of the screwing torque τ_z exceeds the bound τ_z^{thr} required for the fastening of the bayonet mount, the robot enters the *Stopping motion* state and starts decelerating towards zero angular velocity $\dot{\psi}$;

5. The task terminates in *Opening leader gripper* state.

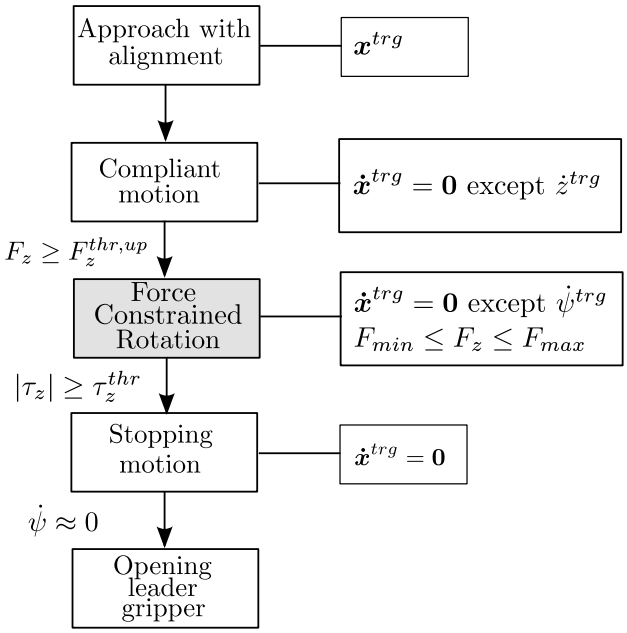


Figure 6.14: State machine governing the task execution.

6.3.3 Control Implementation

During the *Approach with alignment* phase, Reflexxes Motion Libraries [88] have been exploited as real-time trajectory generation algorithm, selecting the initial pose $\boldsymbol{x}^{trg}$ as target position and $\dot{\boldsymbol{x}}^{trg}$ as target velocity.

The cost function (6.8) has been employed in Algorithm 2 during the *Compliant motion* phase.

$$\min_{\boldsymbol{u}_k} \mathcal{L}\left(\dot{\boldsymbol{x}}_{k+1} - \tilde{\dot{\boldsymbol{x}}}_{k+1}^{gen}, \boldsymbol{x}_{k+1} - \tilde{\boldsymbol{x}}_{k+1}^{gen}\right)$$

where, according to (6.2)

$$\tilde{\boldsymbol{x}}_{k+1}^{gen} = \boldsymbol{x}_{k+1}^{gen} + \Delta\boldsymbol{x}_{k+1}$$
$$\tilde{\dot{\boldsymbol{x}}}_{k+1}^{gen} = \boldsymbol{x}_{k+1}^{gen} + \Delta\dot{\boldsymbol{x}}_{k+1}$$

being $\boldsymbol{x}_{k+1}^{gen},\ \boldsymbol{x}_{k+1}^{gen} \in \mathbb{R}^6$ the output of the *velocity based* trajectory generation module, computed by Algorithm 3, while $\Delta\boldsymbol{x}_{k+1},\ \Delta\dot{\boldsymbol{x}}_{k+1} \in \mathbb{R}^6$ are the position and velocity modifications, respectively, produced by the admittance filter and computed as in (6.10).

In order to account for the lower and upper bound on the interaction force during the *Force constrained rotation* phase, the corresponding constraints (6.21) have been considered within the set of constraints (5.13) in Algorithm 2.

Minimization of joint space velocities $\dot{\boldsymbol{q}}_{k+1}$ has been chosen as candidate cost function in (5.18) to handle kinematic redundancy, see (5.19). The corresponding sequence of QP problems within the *reactive controller* module has been solved employing the state-of-the-art hierarchical solver in [144].

6.3.4 Experimental Results

The bimanual cap rotation task for the considered assembly use case consists in performing the compliant alignment and subsequent force constrained rotation of the cap of a plastic pipette on top of the pipette body, see Fig. 6.15.
Sensorless execution of the assembly task is enabled by estimation of the interaction force, according to Sec. 5.1.3. During the *Approach with alignment* phase, an orientation offset of $5\,\mathrm{deg}$ has been purposely introduced on the X and Y Euler angles of the *follower* end effector pose $\boldsymbol{x}^{trg}$, in order to evaluate controller robustness to possible misalignments. An approaching velocity $\dot{z}^{trg}$ (see Fig. 6.14) of $1\ mm/s$ has been adopted to limit the impact force when the contact is established. Selection of the admittance parameters has been performed according to (6.7), based on approximate knowledge of the environment (i.e. plastic) stiffness and damping. The following values have been used: $M = 0.5\ kg$, $M^{rot} = 0.2\ kgm^2/rad$ for the mass and rotational inertia, respectively, $D = 40\ Ns/m$, $D^{rot} = 50\ Nms/rad$ for the linear and rotational damping, respectively. A threshold $F_z^{thr,up} = 0.5\ N$ has been used to switch to the subsequent *Force constrained rotation* phase. The lower and upper bounds on the interaction force are $F_{min} = 0.2\ N$ and $F_{max} = 0.5\ N$, respectively, while the con-

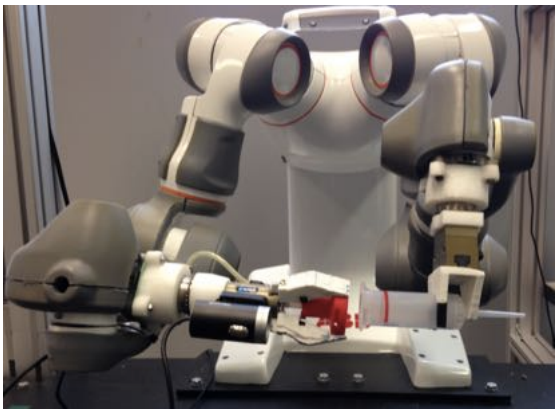
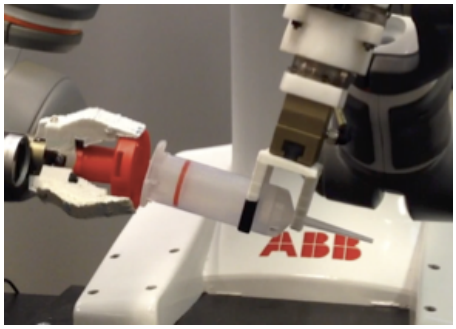
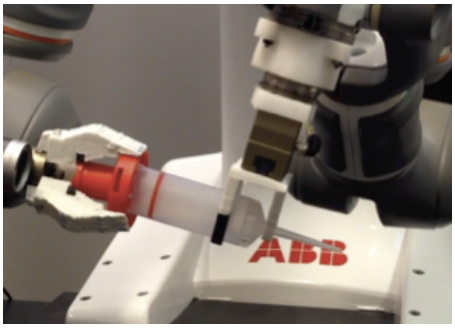


Figure 6.15: Experimental setup: the ABB YuMi robot used in this thesis has been equipped with Schunk pneumatically actuated grippers and 3D printed tools, specifically designed to perform a cap rotation task using an Eppendorf combitip® plastic pipette.

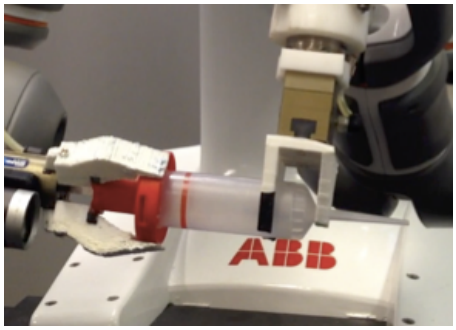
sidered uncertainty in the environment stiffness is: $K_{min} = 1300\ N/m$ and $K_{max} = 1600\ N/m$. An angular velocity $\dot{\psi}^{trg} = 0.1\ rad/s$ is applied during the screwing phase (see Fig. 6.14) . Finally, a threshold value $|\tau_z^{thr,up}| = 0.05\ Nm$ has been chosen to complete the task execution by stopping the cap rotation and opening the leader gripper.
Snapshots of the complete experiment are shown in Fig. 6.16, corresponding to the different phases of the cap assembly operation. Time history of the estimated contact force along the screwing axis F_z during the complete assembly task is reported in Fig. 6.17, together with the additional time history of F_x, F_y and τ_x, τ_y during the *Compliant motion* phase. After approx. $t = 28\ s$ the interaction force exceeds the threshold $F_z^{thr,up} = 0.5\ N$ and the control execution enters the *Force constrained rotation* phase, during which contact loss is effectively prevented.



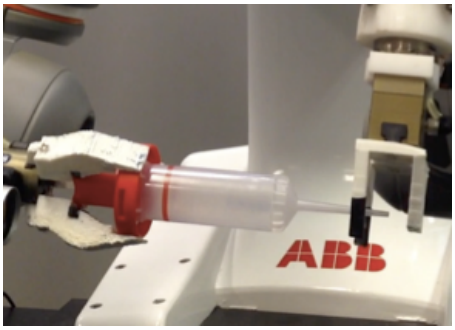
(a) Approach phase



(b) Compliant motion



(c) Force constrained cap rotation



(d) Opening *leader* gripper

Figure 6.16: Snapshots taken from an experiment of bimanual cap assembly.

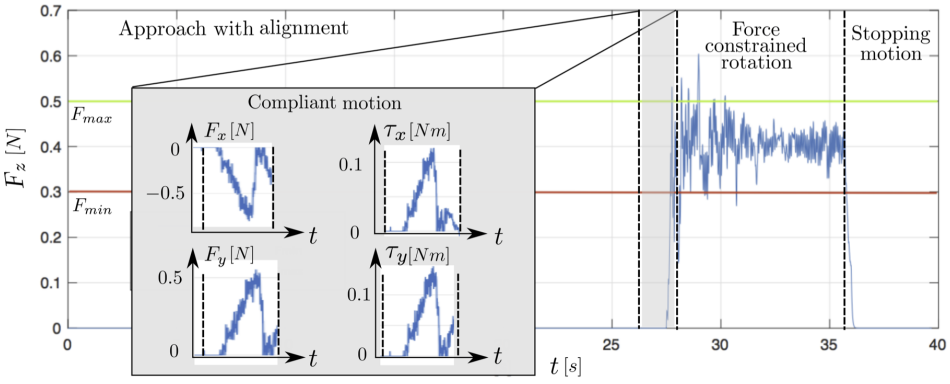


Figure 6.17: Time history of the estimated interaction force along the screwing axis F_z and considered lower bound F_{min} (solid red line) and upper bound F_{max} (solid green line). Additional time history window of F_x, F_y and τ_x, τ_y during the *Compliant motion* phase.

CHAPTER 7

Conclusions

THIS thesis provides research contributions in the context of control of robot interaction with the environment, commonly known as *robot force control*, with a specific focus on the types of interaction enabled by industrial robotic platforms, as traditional (non-redundant) industrial robots and the recently commercially available light-weight dual-arm robots.

As described in Chapter 1, *direct robot force control*, i.e. regulation of the contact force/moment through the closure of a force feedback loop, is required in finishing and machining operations and it is generally implemented on the first category of manipulators, due to the inherent stiffness and the high position accuracy. In this respect, performance improvement of *implicit force control* (a state-of-the-art implementation of hybrid force-motion control for position controlled industrial robots) has been addressed in Chapter 2 by introducing suitable control requirements.

A constrained control approach based on *invariance control* has been presented in Chapter 3. Its application to robot implicit force control and subsequent experimental validation enables force regulation with improved settling performance, compared to standard implicit force control, and absence of force overshoots. Controller robustness to compliance uncertainties has been addressed first in an adaptive fashion, and later by exploiting robust set invariance for a smooth convergence to the force reference.

It remains as future work to preserve the high controller performance during robot motion and in presence of stiff contact conditions.

A *data-driven control design* approach based on *Virtual Reference Feedback Tuning* (VRFT) has been proposed in Chapter 4 to prevent the deterioration of controller performance related to environment modeling and identification. Thanks to the proposed data-driven controller, force overshoots and possible instabilities of the controlled system connected to a rough environment identification or system under-modeling are inherently prevented. Furthermore, an increased matching between the desired and the achieved closed-loop dynamics can be obtained. In order to overcome the main limitations inherent in a data-driven approach, a Recursive Least Squares (RLS) implementation of VRFT has been further presented, while an outer Model Predictive Controller (MPC) has been introduced to effectively enhance the closed-loop performance.

Controller extension to stiff contact situations and torque controlled manipulators represents a future improvement. Furthermore, real-time adaptation of VRFT controller parameters based on varying environment conditions: think of a multi-material surface, is a promising future research direction.

The redundancy and inherent compliance of bimanual light-weight robotic systems clearly motivate their employment in assembly tasks. On the other hand, due to the lower position accuracy, the application of force control algorithms (mainly *indirect force control* methods) becomes even more crucial for a successful task execution, compared to traditional industrial robots. In this regard, bimanual robotic assembly has been treated in Chapter 6 as an equivalent trajectory generation control problem fulfilling force control requirements. As described in Chapter 5, the presented control approach relies on a *constraint-based trajectory generation* framework, while estimation of the interaction force/torque allows for sensorless execution of the assembly operation. Two methods have been developed and successfully validated in a peg-in-hole insertion task and in a cap rotation task, required for the bimanual assembly of a plastic pipette.

The presented methods have been evaluated independently. It remains as future work to merge them together and additionally exploit vision sensors in order to achieve a fully autonomous assembly process.

List of Figures

List of Tables

Bibliography

[1] L. Villani and J. De Schutter. Force control. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, chapter 7, pages 161–185. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[2] T. L. De Fazio, D. S. Seltzer, and D. E. Whitney. The instrumented remote center of compliance. *Ind. Robot*, 11(4):238–242, 1984.

[3] N. Hogan. Impedance control: An approach to manipulation, part I - theory. *ASME Journal of Dynamic Systems, Measurement, and Control*, 107:1–7, 1985.

[4] N. Hogan. Impedance control: An approach to manipulation, part II - implementation. *ASME Journal of Dynamic Systems, Measurement, and Control*, 107:8–16, 1985.

[5] N. Hogan. Impedance control: An approach to manipulation, part III - applications. *ASME Journal of Dynamic Systems, Measurement, and Control*, 107:17–24, 1985.

[6] J. K. Salisbury. Active stiffness control of a manipulator in cartesian coordinates. In *IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, pages 95–100, 1980.

[7] D. E. Whitney. Force feedback control of manipulator fine motions. *ASME Journal of Dynamic Systems, Measurement, and Control*, 99:91–97, 1977.

[8] M. T. Mason. Compliance and force control for computer controlled manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(6):418–432, 1981.

[9] J. De Schutter and H. Van Brussel. Compliant robot motion I. A formalism for specifying compliant motion tasks. *The International Journal of Robotics Research*, 7(4):3–17, 1988.

[10] J. De Schutter, T. De Laet, J. Rutgeerts, W. Decré, R. Smits, E. Aertbeliën, K. Claes, and H. Bruyninckx. Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty. *The International Journal of Robotics Research*, 26(5):433–455, 2007.

[11] M. H. Raibert and J. J. Craig. Hybrid position/force control of manipulators. *ASME Journal of Dynamic Systems, Measurement, and Control*, 103(2):126–133, 1981.

[12] T. Yoshikawa, T. Sugie, and M. Tanaka. Dynamic hybrid position/force control of robot manipulators-controller design and experiment. *IEEE Journal on Robotics and Automation*, 4(6):699–705, 1988.

[13] N. H. McClamroch and D. Wang. Feedback stabilization and tracking of constrained robots. *IEEE Transactions on Automatic Control*, 33(5):419–426, 1988.

[14] J. K. Mills and A. A. Goldenberg. Force and position control of manipulators during constrained motion tasks. *IEEE Transactions on Robotics and Automation*, 5(1):30–46, 1989.

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

[16] L. Villani, C. C. De Wit, and B. Brogliato. An exponentially stable adaptive control for force and position tracking of robot manipulators. *IEEE Transactions on Automatic Control*, 44(4):798–802, 1999.

[17] S. Chiaverini and L. Sciavicco. The parallel approach to force/position control of robotic manipulators. *IEEE Transactions on Robotics and Automation*, 9(4):361–373, 1993.

[18] R. Kelly, R. Carelli, M. Amestegui, and R. Ortega. On adaptive impedance control of robot manipulators. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 572–577, 1989.

[19] R. Colbaugh, H. Seraji, and K. Glass. Direct adaptive impedance control of manipulators. In *IEEE Conference on Decision and Control*, volume 3, pages 2410–2415, 1991.

[20] Z. Lu and A. A. Goldenberg. Robust impedance control and force regulation: Theory and experiments. *The International Journal of Robotics Research*, 14(3):225–254, 1995.

[21] J.-H. Yang, F.-L. Lian, and L.-C. Fu. Adaptive hybrid position/force control for robotic manipulators with compliant links. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 603–608, 1995.

[22] L. L. Whitcomb, S. Arimoto, T. Naniwa, and F. Ozaki. Adaptive model-based hybrid control of geometrically constrained robot arms. *IEEE Transactions on Robotics and Automation*, 13(1):105–116, 1997.

[23] B. Yao, S. P. Chan, and D. Wang. Unified formulation of variable structure control schemes for robot manipulators. *IEEE Transactions on Automatic Control*, 39(2):371–376, 1994.

[24] S. Nozawa, I. Kumagai, Y. Kakiuchi, K. Okada, and M. Inaba. Humanoid full-body controller adapting constraints in structured objects through updating task-level reference force. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3417–3424, 2012.

[25] H. Arisumi, N. Kwak, and K. Yokoi. Systematic touch scheme for a humanoid robot to grasp a door knob. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3324–3331, 2011.

[26] L. Saab, O. E. Ramos, F. Keith, N. Mansard, P. Souères, and J. Y. Fourquet. Dynamic whole-body motion generation under rigid contacts and other unilateral constraints. *IEEE Transactions on Robotics*, 29(2):346–362, 2013.

[27] A. Del Prete, F. Nori, G. Metta, and L. Natale. Prioritized motion–force control of constrained fully-actuated robots: "task space inverse dynamics". *Robotics and Autonomous Systems*, 63, Part 1:150 – 157, 2015.

[28] M. Fumagalli, R. Naldi, A. Macchelli, R. Carloni, S. Stramigioli, and L. Marconi. Modeling and control of a flying robot for contact inspection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3532–3537, 2012.

[29] H. N. Nguyen and D. Lee. Hybrid force/motion control and internal dynamics of quadrotors for tool operation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3458–3464, 2013.

[30] A. Dietrich, C. Ott, and S. Stramigioli. Passivation of projection-based null space compliance control via energy tanks. *IEEE Robotics and Automation Letters*, 1(1):184–191, 2016.

[31] A. Dietrich, X. Wu, K. Bussmann, C. Ott, A. Albu-Schäffer, and S. Stramigioli. Passive hierarchical impedance control via energy tanks. *IEEE Robotics and Automation Letters*, 2(2):522–529, 2017.

[32] R. Ikeura and H. Inooka. Variable impedance control of a robot for cooperation with a human. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 3097–3102, 1995.

[33] C. Yang, G. Ganesh, S. Haddadin, S. Parusel, A. Albu-Schaeffer, and E. Burdet. Human-like adaptation of force and impedance in stable and unstable interactions. *IEEE Transactions on Robotics*, 27(5):918–930, 2011.

[34] F. Ficuciello, L. Villani, and B. Siciliano. Variable impedance control of redundant manipulators for intuitive human-robot physical interaction. *IEEE Transactions on Robotics*, 31(4):850–863, 2015.

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

[36] F. Ferraguti, N. Preda, A. Manurung, M. Bonfè, O. Lambercy, R. Gassert, R. Muradore, P. Fiorini, and C. Secchi. An energy tank-based interactive control architecture for autonomous and teleoperated robotic surgery. *IEEE Transactions on Robotics*, 31(5):1073–1088, 2015.

[37] DesignNews. Force control and machine vision guide robots. [DesignNews webpage] `https://www.designnews.com/automation-motion-control/force-control-and-machine-vision-guide-robots/145196803951421/page/1/0`, 2007.

[38] M. Bélanger-Barrette. Some industrial robot manufacturers offer force control packages. [ROBOTIQ webpage] `http://blog.robotiq.com/some-robot-manufacturers-offer-force-control-packages`, 2015.

[39] ABB Robotics. Integrated force control. [ABB webpage] `http://new.abb.com/products/robotics/application-equipment-and-accessories/integrated-force-control`.

[40] KUKA Robotics. KUKA.ForceTorqueControl. [KUKA webpage] `https://www.kuka.com/en-cn/products/robotics-systems/software/application-software/kuka_forcetorquecontrol`.

[41] FANUC America Corporation. Utilizing force sensors in your robotic applications. [FANUC webpage] `http://robot.fanucamerica.com/robotics-articles/force-sensors-in-robot-applications.aspx`.

[42] J. Krüger, T.K. Lien, and A. Verl. Cooperation of human and machines in assembly lines. *CIRP - Annals Manufacturing Technology*, 58(2):628–646, 2009.

[43] A. Cherubini, R. Passama, A. Crosnier, A. Lasnier, and P. Fraisse. Collaborative manufacturing with physical human–robot interaction. *Robotics and Computer-Integrated Manufacturing*, 40:1–13, 2016.

[44] J. Krüger, G. Schreck, and D. Surdilovic. Dual arm robot for flexible and cooperative assembly. *CIRP Annals - Manufacturing Technology*, 60(1):5 – 8, 2011.

[45] A. Stolt, M. Linderoth, A. Robertsson, and R. Johansson. Force controlled assembly of emergency stop button. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3751–3756, 2011.

[46] A. Stolt, M. Linderoth, A. Robertsson, and R. Johansson. Force controlled robotic assembly without a force sensor. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1538–1543, 2012.

[47] A. Stolt, M. Linderoth, A. Robertsson, and R. Johansson. Robotic assembly of emergency stop buttons. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2081–2081, 2013.

[48] A. M. Zanchettin and P. Rocco. Motion planning for robotic manipulators using robust constrained control. *Control Engineering Practice*, 59:127–136, 2017.

[49] J. De Schutter and H. Van Brussel. Compliant robot motion II. A control approach based on external control loops. *The International Journal of Robotics Research*, 7(4):18–33, 1988.

[50] R. Rossi, L. Bascetta, and P. Rocco. Implicit force control for an industrial robot with flexible joints and flexible links. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4742–4749, 2014.

[51] R. Volpe and P. Khosla. A theoretical and experimental investigation of explicit force control strategies for manipulators. *IEEE Transactions on Automatic Control*, 38(11):1634–1650, 1993.

[52] L. S. Wilfinger, J. T. Wen, and S. H. Murphy. Integral force control with robustness enhancement. *IEEE Control Systems*, 14(1):31–40, 1994.

[53] R. Rossi, L. Fossali, A. Novazzi, L. Bascetta, and P. Rocco. Implicit force control for an industrial robot based on stiffness estimation and compensation during motion. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1138–1145, 2016.

[54] Ciro Natale, Ralf Koeppe, and Gerd Hirzinger. A systematic design procedure of force controllers for industrial robots. *IEEE/ASME Transactions on Mechatronics*, 5(2):122–131, 2000.

[55] Rui Cortesao, Ralf Koeppe, Urbano Nunes, and Gerd Hirzinger. Explicit force control for manipulators with active observers. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2, pages 1075–1080, 2000.

[56] M. Parigi Polverini, R. Rossi, G. Morandi, L. Bascetta, A.M. Zanchettin, and P. Rocco. Performance improvement of implicit integral robot force control through constraint-based optimization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3368–3373, 2016.

[57] M. Parigi Polverini, D. Nicolis, A. M. Zanchettin, and P. Rocco. Implicit robot force control based on set invariance. *IEEE Robotics and Automation Letters*, 2(3):1288–1295, 2017.

[58] M. Parigi Polverini, D. Nicolis, A.M. Zanchettin, and P. Rocco. Robust set invariance for implicit robot force control in presence of contact model uncertainty. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6393–6399, 2017.

[59] A. Glattfelder and W. Schaufelberger. *Control Systems with Input and Output Constraints*. Advanced Textbooks in Control and Signal Processing, Springer London, 2003.

[60] F. Blanchini. Set invariance in control. *Automatica*, 35(11):1747–1767, 1999.

[61] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[62] A. Bemporad, F. Borrelli, and M. Morari. Model predictive control based on linear programming - the explicit solution. *IEEE Transactions on Automatic Control*, 47(12):1974–1985, 2002.

[63] A. Bemporad. Reference governor for constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 43(3):415–419, Mar 1998.

[64] K. B. Ngo, R. Mahony, and Z.-P. Jiang. Integrator backstepping using barrier functions for systems with multiple state constraints. In *IEEE Conference on Decision and Control*, pages 8306–8312, 2005.

[65] J. Wolff and M. Buss. Invariance control design for nonlinear control affine systems under hard state constraints. *IFAC Symposium on Nonlinear Control Systems (NOLCOS)*, 37(1):555–560, 2004.

[66] J. Wolff and M. Buss. Invariance control design for constrained nonlinear systems. *IFAC Proceedings Volumes*, 38(1):37–42, 2005.

[67] E. C. Kerrigan and J. M. Maciejowski. Invariant sets for constrained nonlinear discrete-time systems with application to feasibility in model predictive control. In *IEEE Conference on Decision and Control*, volume 5, pages 4951–4956, 2000.

[68] E. Gilbert and I. Kolmanovsky. Nonlinear tracking control in the presence of state and control constraints: a generalized reference governor. *Automatica*, 38(12):2063–2073, 2002.

[69] I. Kolmanovsky, E. Garone, and S. Di Cairano. Reference and command governors: A tutorial on their theory and automotive applications. In *American Control Conference*, pages 226–241, 2014.

[70] M. Scheint, J. Wolff, and M. Buss. Invariance control in robotic applications: Trajectory supervision and haptic rendering. In *American Control Conference*, pages 1436–1442, June 2008.

[71] M. Kimmel, M. Lawitzky, and S. Hirche. 6d workspace constraints for physical human-robot interaction using invariance control with chattering reduction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3377–3383, 2012.

[72] M. Kimmel and S. Hirche. Active safety control for dynamic human-robot interaction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4685–4691, 2015.

[73] F. Blanchini, S. Miani, F. A. Pellegrino, and B. Van Arkel. Enhancing controller performance for robot positioning in a constrained environment. *IEEE Transactions on Control Systems Technology*, 16(5):1066–1074, 2008.

[74] D. C. Conner, A. A. Rizzi, and H. Choset. Composition of local potential functions for global robot control and navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 4, pages 3546–3551, 2003.

[75] C. P. Bechlioulis, M. V. Liarokapis, and K. J. Kyriakopoulos. Robust model free control of robotic manipulators with prescribed transient and steady state performance. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 41–46, 2014.

[76] C. P. Bechlioulis and G. A. Rovithakis. Robust adaptive control of feedback linearizable mimo nonlinear systems with prescribed performance. *IEEE Transactions on Automatic Control*, 53(9):2090–2099, 2008.

[77] D. Nicolis, A. M. Zanchettin, and P. Rocco. Constraint-based and sensorless force control with an application to a lightweight dual-arm robot. *IEEE Robotics and Automation Letters*, 1(1):340–347, 2016.

[78] A. M. Zanchettin and P. Rocco. Reactive motion planning and control for compliant and constraint-based task execution. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2748–2753, 2015.

[79] A. M. Zanchettin and P. Rocco. Motion planning for robotic manipulators using robust constrained control. *Control Engineering Practice*, 59:127 – 136, 2017.

[80] E. Lutscher and G. Cheng. Hierarchical inequality task specification for indirect force controlled robots using quadratic programming. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4722–4727, 2014.

[81] E. Lutscher and G. Cheng. A practical approach to generalized hierarchical task specification for indirect force controlled robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1854–1859, 2013.

[82] A. E. Bryson and Y.-C. Ho. *Applied Optimal Control: Optimization, Estimation, and Control*. 1975.

[83] J. Wing and C. Desoer. The multiple-input minimal time regulator problem (general theory). *IEEE Transactions on Automatic Control*, 8(2):125–136, 1963.

[84] H. J. Ferreau, A. Potschka, and C. Kirches. [qpOASES webpage]. `http://www.qpOASES.org/`, 2007–2015.

[85] S. Chand and K. L. Doty. On-line polynomial trajectories for robot manipulators. *The International Journal of Robotics Research*, 4(2):38–48, 1985.

[86] J. Wolff, C. Weber, and M. Buss. Continuous control mode transitions for invariance control of constrained nonlinear systems. In *IEEE Conference on Decision and Control*, pages 542–547, 2007.

[87] M. Fujiwara. Über die obere schranke des absoluten betrages der wurzeln einer algebraischen gleichung. *Tôhoku Mathematical Journal*, 10:167–171, 1916.

[88] T. Kröger and F. M. Wahl. Online trajectory generation: Basic concepts for instantaneous reactions to unforeseen events. *IEEE Transactions on Robotics*, 26(1):94–111, 2010.

[89] O. Gerelli and C. Guarino Lo Bianco. A discrete-time filter for the on-line generation of trajectories with bounded velocity, acceleration, and jerk. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3989–3994, 2010.

[90] M. Galassi. GNU Scientific Library Reference Manual (3rd Ed.). [GNU webpage] `http://www.gnu.org/software/gsl/`.

[91] D. Surdilovic. Robust control design of impedance control for industrial robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3572–3579, 2007.

[92] G. Mester. Adaptive force and position control of rigid-link flexible-joint scara robots. In *International Conference on Industrial Electronics, Control and Instrumentation (IECON)*, volume 3, pages 1639–1644, Sep 1994.

[93] D. Surdilovic and Z. Cojbasic. Robust robot compliant motion control using intelligent adaptive impedance approach. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 2128–2133, 1999.

[94] T. Kroger, B. Finkemeyer, M. Heuck, and F.M. Wahl. Adaptive implicit hybrid force/pose control of industrial manipulators: compliant motion experiments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 816–821, 2004.

[95] S. K. Singh and D.O. Popa. An analysis of some fundamental problems in adaptive control of force and impedance behavior: theory and experiments. *IEEE Transactions on Robotics and Automation*, 11(6):912–921, 1995.

[96] H. Seraji and R. Colbaugh. Force tracking in impedance control. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 499–506, 1993.

[97] J. Seul, T.C. Hsia, and R.G. Bonitz. Force tracking impedance control of robot manipulators under unknown environment. *IEEE Transactions on Control Systems Technology*, 12(3):474–483, 2004.

[98] J. Prak, R. Cortesao, and O. Khatib. Robust and adaptive teleoperation for compliant motion task. In *International Conference on Advanced Robotics*, pages 513–519, 2003.

[99] R. Cortesao, Jaeheung Park, and O. Khatib. Real-time adaptive control for haptic telemanipulation with kalman active observers. *IEEE Transactions on Robotics,*, 22(5):987–999, 2006.

[100] G. Alici and B. Shirinzadeh. Enhanced stiffness modeling, identification and characterization for robot manipulators. *IEEE Transactions on Robotics*, 21(4):554–564, 2005.

[101] C. Dumas, S. Caro, M. Cherif, S. Garnier, and B. Furet. A methodology for joint stiffness identification of serial robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 464–469, 2010.

[102] S. Tang and C.-C. Wang. Computation of the effects of link deflections and joint compliance on robot positioning. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 4, pages 910–915, 1987.

[103] J.-H Yang and L.-C. Fu. Analysis and control for manipulators with both joint and link flexibility. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 984–989, 1993.

[104] D. Erickson, M. Weber, and I. Sharf. Contact stiffness and damping estimation for robotic systems. *The International Journal of Robotics Research*, 22(1):41–57, 2003.

[105] D. Verscheure, I. Sharf, H. Bruyninckx, J. Swevers, and J. De Schutter. Identification of contact dynamics parameters for stiff robotic payloads. *IEEE Transactions on Robotics*, 25(2):240–252, 2009.

[106] N. Diolaiti, Claudio Melchiorri, and S. Stramigioli. Contact impedance estimation for robotic systems. *IEEE Transactions on Robotics*, 21(5):925–935, 2005.

[107] L.J. Love and W.J. Book. Environment estimation for enhanced impedance control. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1854–.1859, 1995.

[108] M. Weber, K. Patel, O. Ma, and I. Sharf. Identification of contact dynamics model parameters from constrained robotic operations. *Journal of dynamic systems, measurement, and control*, 128(2):307–318, 2006.

[109] S.-T. Lin and K. H. Yae. Identification of unknown payload and environmental parameters for robot compliant motion. In *American Control Conference*, pages 2952–2956, 1992.

[110] R. Kikuuwe and T. Yoshikawa. Robot perception of environment impedance. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1661–1666, 2002.

[111] J.C. Doyle, B.A. Francis, and A. Tannenbaum. *Feedback control theory*, volume 134. Macmillan New York, 1992.

[112] K. van Heusden, A. Karimi, and D. Bonvin. Data-driven model reference control with asymptotically guaranteed stability. *International Journal of Adaptive Control and Signal Processing*, 25(4):331–351, 2011.

[113] S. Formentin and A. Karimi. A data-driven approach to mixed-sensitivity control with application to an active suspension system. *IEEE Transactions on Industrial Informatics*, 9(4):2293–2300, 2013.

[114] M.C. Campi, A. Lecchini, and S.M. Savaresi. Virtual reference feedback tuning: a direct method for the design of feedback controllers. *Automatica*, 38(8):1337–1346, 2002.

[115] S. Formentin, M.C. Campi, and S.M. Savaresi. Virtual reference feedback tuning for industrial PID controllers. In *World Congress The International Federation of Automatic Control (IFAC)*, volume 47, pages 11275–11280, 2014.

[116] S. Formentin, A. L. Cologni, F. Previdi, and S. M. Savaresi. A data-driven approach to control of batch processes with an application to a gravimetric blender. *IEEE Transactions on Industrial Electronics*, 61(11):6383–6390, 2014.

[117] S. Formentin, S. M. Savaresi, and L. Del Re. Non-iterative direct data-driven controller tuning for multivariable systems: theory and application. *IET control theory & applications*, 6(9):1250–1257, 2012.

[118] T. E. Passenbrunner, S. Formentin, S. M. Savaresi, and L. Del Re. Direct multivariable controller tuning for internal combustion engine test benches. *Control Engineering Practice*, 29:115–122, 2014.

[119] J. D. Rojas, X. Flores-Alsina, U. Jeppsson, and R. Vilanova. Application of multivariate virtual reference feedback tuning for wastewater treatment plant control. *Control Engineering Practice*, 20(5):499–510, 2012.

[120] S. Formentin, K. Heusden, and A. Karimi. A comparison of model-based and data-driven controller tuning. *International Journal of Adaptive Control and Signal Processing*, 28(10):882–897, 2014.

[121] G.O. Guardabassi and S.M. Savaresi. Approximate feedback linearization of discrete-time non-linear systems using virtual input direct design. *Systems & Control Letters*, 32(2):63–74, 1997.

[122] M.C. Campi and S.M. Savaresi. Direct nonlinear control design: The virtual reference feedback tuning (vrft) approach. *IEEE Transactions on Automatic Control*, 51(1):14–27, 2006.

[123] S. Formentin, D. Piga, R. Tóth, and S. M. Savaresi. Direct learning of LPV controllers from data. *Automatica*, 65:98–110, 2016.

[124] M. Parigi Polverini, S. Formentin, L. A. Dao, and P. Rocco. Data-driven design of implicit force control for industrial robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2322–2327, 2017.

[125] R. Johansson. *System Modeling and Identification*. Prentice-Hall, Englewood Cliffs, New Jersey, 1993.

[126] S. Bittanti. *Identificazione dei Modelli e Sistemi Adattativi*. Pitagora Editrice, Bologna, 2004.

[127] M. C. Campi. VRFT toolbox for MATLAB. [webpage] `http://marco-campi.unibs.it/VRFTwebsite/index.html`.

[128] D. Piga, S. Formentin, and A. Bemporad. Direct data-driven control of constrained systems. *IEEE Transactions on Control Systems Technology*, PP(99):1–8, 2017.

[129] U. V. Kalabić, I. V. Kolmanovsky, and E. G. Gilbert. Reduced order extended command governor. *Automatica*, 50(5):1466 – 1472, 2014.

[130] Y. Kansha, Y. Hashimoto, and M.-S. Chiu. New results on VRFT design of PID controller. *Chemical Engineering Research and Design*, 86(8):925 – 931, 2008.

[131] X. Yang, Y. Li, Y. Kansha, and M.-S. Chiu. Enhanced VRFT design of adaptive PID controller. *Chemical Engineering Research and Design*, 76:66 – 72, 2012.

[132] ABB IRB 14000 YuMi. [ABB webpage] `http://new.abb.com/products/robotics/industrial-robots/yumi`, 2015.

[133] S. Kock, T. Vittor, B. Matthias, H. Jerregard, M. Källman, I. Lundberg, R. Mellander, and M. Hedelind. Robot concept for scalable, flexible assembly automation: A technology study on a harmless dual-armed robot. In *IEEE International Symposium on Assembly and Manufacturing (ISAM)*, pages 1–5, 2011.

[134] B. Matthias, S. Kock, H. Jerregard, M. Källman, and I. Lundberg. Safety of collaborative industrial robots: Certification possibilities for a collaborative assembly robot concept. In *IEEE International Symposium on Assembly and Manufacturing (ISAM)*, pages 1–6, 2011.

[135] A. Blomdell, G. Bolmsjo, T. Brogardh, P. Cederberg, M. Isaksson, R. Johansson, M. Haage, K. Nilsson, M. Olsson, T. Olsson, A. Robertsson, and Jianjun Wang. Extending an industrial robot controller: implementation and applications of a fast open sensor interface. *IEEE Robotics Automation Magazine*, 12(3):85–94, 2005.

[136] LabComm protocol. [Lund University - Tools and Softwares] `http://www.control.lth.se/Research/tools.html`.

[137] A. De Luca and R. Mattone. Sensorless robot collision detection and hybrid force/motion control. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 999–1004, 2005.

[138] A. De Luca, A. Albu-Schaffer, S. Haddadin, and G. Hirzinger. Collision detection and safe reaction with the dlr-iii lightweight manipulator arm. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, page 1623–1630, 2006.

[139] M. Ragaglia, A. M. Zanchettin, L. Bascetta, and P. Rocco. Accurate sensorless lead-through programming for lightweight robots in structured environments. *Robotics and Computer-Integrated Manufacturing*, 39:9 – 21, 2016.

[140] A. M. Zanchettin and P. Rocco. Near time-optimal and sensor-based motion planning for robotic manipulators. In *IEEE Conference on Decision and Control*, pages 965–970, 2013.

[141] A. M. Zanchettin and P. Rocco. Reactive motion planning and control for compliant and constraint-based task execution. In *IEEE Conference on Robotics and Automation (ICRA)*, pages 2748–2753, 2015.

[142] L. Biagiotti and C. Melchiorri. *Trajectory planning for automatic machines and robots*. Springer Science & Business Media, 2008.

[143] O. Kanoun, F. Lamiraux, and P. B Wieber. Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task. *IEEE Transactions on Robotics*, 27(4):785–792, 2011.

[144] A. Escande, N. Mansard, and P.-B. Wieber. Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *International Journal of Robotics Research*, 3(7):1006 – 1028, 2014.

[145] M. Parigi Polverini, A. M. Zanchettin, and P. Rocco. A computationally efficient safety assessment for collaborative robotics applications. *Robotics and Computer-Integrated Manufacturing*, 46:25–37, 2017.

[146] Roberto Rossi, Matteo Parigi Polverini, Andrea Maria Zanchettin, and Paolo Rocco. A pre-collision control strategy for human-robot interaction based on dissipated energy in potential inelastic impacts. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 26–31, 2015.

[147] A. M. Zanchettin and P. Rocco. Robust constraint-based control of robot manipulators: An application to a visual aided grasping task. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3634–3639, 2016.

[148] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas, and D. Kragic. Dual arm manipulation—a survey. *Robotics and Autonomous Systems*, 60(10):1340 – 1353, 2012.

[149] D. Surdilovic, Y. Yakut, T. M. Nguyen, X. B. Pham, A. Vick, and R. Martin-Martin. Compliance control with dual-arm humanoid robots: Design, planning and programming. In *IEEE-RAS International Conference on Humanoid Robots*, pages 275–281, 2010.

[150] D. Almeida, F. E. Viña, and Y. Karayiannidis. Bimanual folding assembly: Switched control and contact point estimation. In *IEEE-RAS International Conference on Humanoid Robots*, pages 210–216, 2016.

[151] N. T. Dantam, H. Ben Amor, H. I. Christensen, and M. Stilman. Online multi-camera registration for bimanual workspace trajectories. In *IEEE-RAS International Conference on Humanoid Robots*, pages 588–593, 2014.

[152] P. K. Kim, J. H. Bae, H. Park, D. H. Lee, J. H. Park, M. H. Baeg, and J. Park. Dual-arm robot box taping with kinesthetic teaching. In *International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 555–557, 2016.

[153] Y. Yamada, S. Nagamatsu, and Y. Sato. Development of multi-arm robots for automobile assembly. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 2224–2229, 1995.

[154] C. Park, K. Park, D. I. Park, and J. H. Kyung. Dual arm robot manipulator and its easy teaching system. In *IEEE International Symposium on Assembly and Manufacturing*, pages 242–247, 2009.

[155] D. Gravel, F. Maslar, G. Zhang, S. Nidamarthi, H. Chen, and T. Fuhlbrigge. Toward robotizing powertrain assembly. In *World Congress on Intelligent Control and Automation*, pages 541–546, 2008.

[156] E. Gambao, C. Balaguer, and F. Gebhart. Robot assembly system for computer-integrated construction. *Automation in Construction*, 9(5):479–487, 2000.

[157] M. J. Hwang, D. Y. Lee, and S. Y. Chung. Motion planning of bimanual robot for assembly. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 240–245, 2007.

[158] A. Nakashima, Y. Iwanaga, and Y. Hayakawa. A motion planning of dual arm-hand manipulators for origami-folding based on a probabilistic model of constraint transitions within human behavior. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 562–569, 2016.

[159] C. Bersch, B. Pitzer, and S. Kammel. Bimanual robotic cloth manipulation for laundry folding. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1413–1419, 2011.

[160] Y. F. Zheng and M. Z. Chen. Trajectory planning for two manipulators to deform flexible beams. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 1019–1024, 1993.

[161] M. Parigi Polverini, A. M. Zanchettin, S. Castello, and P. Rocco. Sensorless and constraint based peg-in-hole task execution with a dual-arm robot. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 415–420, 2016.

[162] B.-H. Kim, B.-J. Yi, I. H. Suh, and S.-R. Oh. Stiffness analysis for effective peg-in/out-hole tasks using multi-fingered robot hands. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2, pages 1229–1236, 2000.

[163] S.-K. Yun. Compliant manipulation for peg-in-hole: Is passive compliance a key to learn contact motion? In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1647–1652, 2008.

[164] I.-W. Kim, D.-J. Lim, and K.-I. Kim. Active peg-in-hole of chamferless parts using force/-moment sensor. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2, pages 948–953, 1999.

[165] T. Lozano-Perez, M. T. Mason, and R. H Taylor. Automatic synthesis of fine-motion strategies for robots. *The International Journal of Robotics Research*, 3(1):3–24, 1984.

[166] H. Bruyninckx, S. Dutre, and J. De Schutter. Peg-on-hole: a model based solution to peg and hole alignment. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1919–1924, 1995.

[167] S.R. Chhatpar and M.S. Branicky. Search strategies for peg-in-hole assemblies with position uncertainty. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 1465–1470, 2001.

[168] Y. Li. Hybrid control approach to the peg-in-hole problem. *IEEE Robotics & Automation Magazine*, 4(2):52–60, 1997.

[169] S.-Y. Chung and Lee. D. Y. Discrete event systems approach to fixtureless peg-in-hole assembly. In *American Control Conference*, volume 6, pages 4962–4967, 2001.

[170] T. Tsumugiwa, A. Sakamoto, R. Yokogawa, and K. Hara. Switching control of position/torque control for human-robot cooperative task - human-robot cooperative carrying and peg-in-hole task. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1933–1939,, 2003.

[171] T. Matsuno, T. Fukuda, and Y. Hasegawa. Insertion of long peg into tandem shallow hole using search trajectory generation without force feedback. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1123–1128, 2004.

[172] J. Su, H. Qiao, Z. Ou, and Y. Zhang. Sensor-less insertion strategy for an eccentric peg in a hole of the crankshaft and bearing assembly. In *Assembly Automation*, volume 32.1, pages 86–99, 2012.

[173] H. Shouren, K. Murakami, Y. Yamakawa, T. Senoo, and M. Ishikawa. Fast peg-and-hole alignment using visual compliance. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 286–292, 2013.

[174] H. Shouren, Y. Yamakawa, T. Senoo, and M. Ishikawa. Realizing peg-and-hole alignment with one eye-in-hand high-speed camera. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1127–1132, 2013.

[175] A. Gill. *Introduction to the Theory of Finite-state Machines*. McGraw-Hill, New York, NY, USA, 1962.

[176] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor. A vision-based formation control framework. *IEEE Transactions on Robotics and Automation*, 18(5):813–825, 2002.

[177] U. Thomas, B. Finkemeyer, T. Kroger, and F. M. Wahl. Error-tolerant execution of complex robot tasks based on skill primitives. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 3069–3075, 2003.

[178] J. Felip and A. Morales. A solution for the cap unscrewing task with a dual arm sensor-based system. In *IEEE-RAS International Conference on Humanoid Robots*, pages 823–828, 2014.

[179] J. Steffen, C. Elbrechter, R. Haschke, and H. Ritter. Bio-inspired motion strategies for a bimanual manipulation task. In *IEEE-RAS International Conference on Humanoid Robots*, pages 625–630, 2010.

[180] M. Parigi Polverini, A. M. Zanchettin, F. Incocciati, and P. Rocco. Robust constraint-based robot control for bimanual cap rotation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4785–4790, 2017.

[181] B. Barmish and J. Sankaran. The propagation of parametric uncertainty via polytopes. *IEEE Transactions on Automatic Control*, 24(2):346–349, 1979.