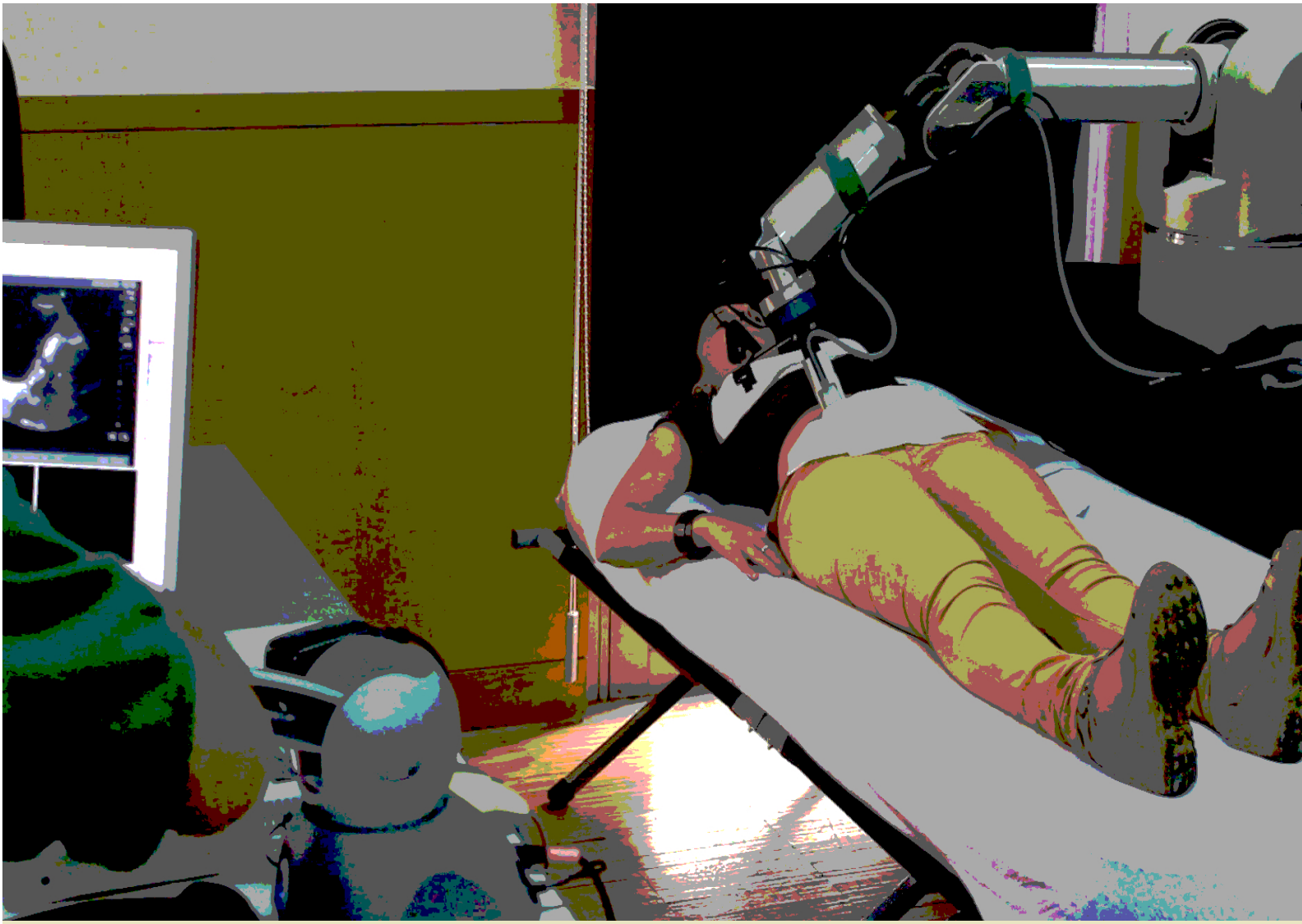


Luís Carlos Vitória dos Santos

Control and Estimation Strategies for Robotic-Assisted Tele-Echography

Tese de doutoramento em Engenharia Electrotécnica e de Computadores, Ramo de Especialização em Automação e Robótica,
orientada pelo Senhor Professor Doutor Rui Pedro Duarte Cortesão
e apresentada ao Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Cisências e Tecnologia da Universidade de Coimbra

Setembro de 2016

· U C ·

Universidade de Coimbra

UNIVERSITY OF COIMBRA

Control and Estimation Strategies for Robotic-Assisted Tele-Echography

Ph.D. Thesis

Luís Carlos Vitória dos Santos

September 2016

Advisor

Prof. Dr. Rui Pedro Duarte Cortesão, University of Coimbra

Control and Estimation Strategies for Robotic-Assisted Tele-Echography

Ph.D. Thesis

Luís Carlos Vitória dos Santos

September 2016

to Preta

ABSTRACT

Medical ultrasound examination is one the most common medical imaging procedures, being a crucial diagnostic tool for several medical specialties. This procedure requires skilled and experienced physicians, which might not always be available, especially in emergency scenarios or in geographically remote areas. Furthermore, some tasks require the physician to perform forceful and repetitive movements in uncomfortable body postures, leading to substantial prevalence of work-related musculoskeletal disorders among sonographers. A robotic-assisted tele-echography system pops up as a solution to these problems, where a haptic device can be used to tele-control a robot manipulator for echographic inspection. To accomplish this, probe motion must accurately replicate physician's movements, either in free space or under contact constraints, avoiding force peaks and jerky behaviors when moving from free-space to contact and vice-versa.

This thesis addresses strategies and methodologies to develop a teleoperation control architecture for robotic-assisted tele-echography, relying on computed-torque techniques, robot dynamics, contact stiffness estimation and task space and null space designs. The slave manipulator control architecture follows a hierarchical approach, where explicit Cartesian force control arises as the primary task while orientation control is designed in the null space. This solution allows accurate orientation control with compliant positioning, which resembles the physician behavior when handling the probe in a traditional ultrasound procedure. Based on a 3D time-of-flight camera and force data, contact stiffness is anticipated, allowing control adaptation before contact. This approach is adequate for tele-echographic tasks since it reduces robot dynamics before contact, enabling smooth transitions from free-space to contact and vice-versa. Additionally, free space inertial forces are also eliminated from the control loop, improving free space motion control. In contact, the environment stiffness is estimated online using robot inertial properties and force data. A strong correlation between the perceived stiffness and the effective mass exists, being this correlation used in the stiffness estimation algorithm to improve force control performance. In this way, the perceived stiffness can be seen, not as a well defined physical property, but as a force control optimization parameter. The teleoperation control architecture is validated in both clinical and non-clinical scenarios. In the clinical experiment, a gynecologist performs a robotic-assisted pelvic ultrasound examination in a healthy volunteer.

RESUMO

A ecografia é uma técnica de imagiologia médica comum e transversal a várias especialidades médicas. É um meio de diagnóstico que requer médicos especialistas para a sua execução, os quais nem sempre se encontram disponíveis, especialmente em locais com baixa densidade populacional ou em situações de emergência. Para além da falta de especialistas, alguns exames requerem que os radiologistas exerçam forças em posições não naturais durante largos períodos de tempo, o que leva a que haja uma elevada incidência de lesões músculo-esqueléticas entre a comunidade de radiologistas. Uma possível solução para estes problemas é a utilização de um sistema de tele-ecografia assistida por robô. Ao controlar a sonda através de um dispositivo háptico, o radiologista consegue efetuar o exame de forma similar ao procedimento tradicional, mantendo a perceção das forças que está a exercer no paciente. Porém, para esta solução ser adotada, a sonda tem que replicar fidedignamente os movimento que o médico efetua no dispositivo háptico, quer em aproximação ao paciente em espaço livre, quer em contacto com este. A dinâmica do robô deve ser adaptada ao tipo de exame, impondo limites nas forças aplicadas no paciente.

Ao longo desta tese vão ser apresentadas as estratégias e metodologias adotadas no desenvolvimento da arquitetura de controlo para tele-ecografia assistida por robô. A arquitetura de controlo apresenta uma estrutura hierárquica, sendo o controlo explicito de força a tarefa com maior prioridade, permitindo impor uma dinâmica de interação adequada entre sonda e paciente. O controlo de orientação é efetuado utilizando formalismo de controlo no espaço das juntas e no espaço da tarefa, o que permite ultrapassar limitações presentes em soluções de controlo efetuado apenas no espaço da tarefa. O controlo de orientação é considerado uma tarefa secundária, sendo projetado no espaço nulo. A dinâmica do manipulador é estabelecida através de uma câmara 3D tempo de voo e de um sensor de força. Utilizando dados de distância e de força, o contacto é antecipado, aumentando a rigidez do ambiente no modelo do controlador em situações de pre-contacto. Esta estratégia reduz a dinâmica do robô, permitindo uma transição suave entre o movimento em espaço livre e contacto (e vice-versa). Permite ainda eliminar as forças inerciais da malha de controlo, melhorando o desempenho do sistema em espaço livre. Em contacto, a rigidez é estimada em tempo real, usando as propriedades inerciais do manipulador e dados de força. Existe uma correlação entre a massa efetiva e a rigidez percecionada pelo

controlo, sendo esta correlação utilizada no algoritmo de estimação da rigidez, o que permite melhorar o desempenho do controlador. Desta forma, o parâmetro da rigidez deixa de ter um significado físico claro, passando a ser interpretado como um parâmetro de otimização do controlo de força. A arquitetura de tele-operação é validada clinicamente por um médico ginecologista, efetuando uma tele-ecografia pélvica num voluntário saudável.

ACKNOWLEDGMENTS

I have read somewhere, someone describing his motivation during his Ph.D. as a sinusoidal wave with a very high frequency. I find that description quite accurate for me too.

A lot of persons were important to me during these years of Ph.D. research. Some of them helped me to become a better researcher, others helped me just by being present in this seemingly endless journey. First of all, I want to express my greatest appreciation to my Ph.D. Advisor Prof. Rui Cortesão. Prof. Cortesão has always been present during my Ph.D., providing guidance and support every time I need it, but also giving me total freedom to develop my research. Undoubtedly this work owes a lot to him. His feedback was always wise and insightful, but I would like to specially acknowledge him for the comment: *"Isso parece que varia com a massa efetiva."* A lot of things started to get clear after that.

I would like to acknowledge the following institutions and programme: Portuguese Science Foundation (FCT) for my Ph.D. grant SFRH/BD/69557/2010 and the research grant under the project PTDC/EEA-ACR/72253/2006; the QREN programme "Mais Centro" for the research grant under the project A "Surgery and Diagnosis Assisted by Computer Using Images" (SCT_2011_02_027_4824); the Luso-American Foundation (FLAD) for supporting my participation in a scientific conference in the United States; the Institute of Systems and Robotics of University of Coimbra, for providing all the conditions to carry my research.

I would also like to acknowledge all the medical doctors that tested and helped to improve the robotic-assisted tele-echography procedure. Specially, I would like to express my gratitude to Dr. Paulo Cortesão for his relevant feedback and for always being available for another testing.

I want to thank my Lab mates. Cristóvão, Michel, Fernanda, Michel, Miguel, Rui, Francisco, Vitor, Pedro,... You all made this journey much more enjoyable. A special thanks to Cristóvão for his support every time I needed help.

To all my friends, thank you for being my friends. Fortunately, you are too many to mention you all.

A special thanks to my parents. As someone said: "Without you, I could never be here!" Thanks Inês. I never thought you would be useful.

For last, but not the least. Yes Preta, it is finally over. A long, long time ago, I promised you that, when I finish my Ph.D, we would... Unfortunately, I cannot recall what it was. Hopefully, it was something

unimportant... Love you!

Luís Carlos Vitória dos Santos
September 2016

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AMAXE	Absolute Maximum Error.
AME	Absolute Mean Error.
AOB	Active Observer.
DH	Denavit-Hartenberg.
DOF	degree of freedom.
EL	Euler-Lagrange.
GUI	Graphical User Interface.
HC	Hunt-Crossley model.
KV	Kelvin-Voight model.
MIMO	multiple input-multiple output.
MRAC	model reference adaptive control.
PD	proportional-derivative controller.
PI	proportional-integral controller.
PID	proportional-integral-derivative controller.
RCM	remote center of motion.
RLS	recursive least-squares.
RMSE	Root Mean Square Error.
SNR	signal-to-noise ratio.
WAM	Whole Arm Manipulator.

NOTATION

$\boldsymbol{A}$	Matrices are represented by bold capital letters	
$\hat{\boldsymbol{A}}$	Estimation of A	
$\boldsymbol{A}^{-1}$	Inverse matrix of A	
$\boldsymbol{A}^{T}$	Transpose matrix of A	
$\boldsymbol{I}$	Identity matrix	
$\det(\boldsymbol{A})$	Determinant of matrix A	
$\text{diag}[\dots \ \dots \ \dots]$	Diagonal matrix representation	
$\boldsymbol{R}$	Rotation matrix	
$\boldsymbol{S}(.)$	Skew symmetric matrix	
$\boldsymbol{a}$	Vectors are represented by bold lowercase letters	
$\dot{\boldsymbol{a}}$	First order time derivative	
$\ddot{\boldsymbol{a}}$	Second order time derivative	
a, A	Scalars are represented either by capital or lowercase regular letters	
a_{ij}, A_{ij}	i^{th} row, j^{th} column element of $\boldsymbol{A}$.	
$\boldsymbol{\alpha}$	Control variable	(3.71)
$\boldsymbol{f}^{*}$	Control variable	(3.79)
$\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$	Coriolis and centrifugal matrix	(3.43)
$\boldsymbol{c}_x(\boldsymbol{x}, \dot{\boldsymbol{x}})$	End-effector Coriolis and centrifugal forces	(3.54)
$\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}})$	Coriolis and centrifugal forces	
d_s	Environment damping	(6.4)
ζ	Damping ratio	(4.21)
$\boldsymbol{f}$	End-effector forces	(3.5)
f_n	Modeled contact force	(6.2)
ω_n	Natural frequency	(4.20)

$\boldsymbol{g}_x(\boldsymbol{x})$	End-effector gravity force	(3.55)
$\boldsymbol{g}(\boldsymbol{q})$	Gravity force	(3.43)
$\boldsymbol{D}$	Impedance control desired damping matrix	(4.3)
$\boldsymbol{M}$	Impedance control desired mass matrix	(4.3)
$\boldsymbol{K}$	Impedance control desired stiffness matrix	(4.3)
δ	Environment indentation	(6.3)
$\boldsymbol{B}(\boldsymbol{q})$	Generalized inertia matrix	(3.35)
$\boldsymbol{\Lambda}(\boldsymbol{q})$	End-effector pseudo-inertia matrix	(3.61)
$\boldsymbol{\Lambda}(\boldsymbol{x})$	End-effector inertia matrix	(3.53)
$\boldsymbol{J}(\boldsymbol{q})$	Jacobian matrix	(3.3)
$\bar{\boldsymbol{J}}(\boldsymbol{q})$	Dynamically consistent generalized inverse of the Jacobian matrix	(3.60)
$\boldsymbol{J}^{\#}(\boldsymbol{q})$	Right generalized inverse of the Jacobian matrix	(3.16)
$\boldsymbol{J}^{\dagger}(\boldsymbol{q})$	Moore-Penrose right pseudo-inverse of the Jacobian matrix	(3.17)
$\mathcal{K}$	Kinetic energy	(3.24, 3.34)
$\mathcal{L}$	Lagrangian	(3.21)
$\mathcal{N}(.)$	Null space	(3.7)
$\boldsymbol{N}(\boldsymbol{q})$	Jacobian matrix null space projector	(5.5)
$\boldsymbol{\omega}$	End-effector angular velocity in the task space	(3.28)
$\dot{\boldsymbol{p}}$	End-effector linear velocity in the task space	(3.28)
$\mathcal{P}$	Potential energy	(3.36)
$\boldsymbol{q}$	Joint position	(3.2)
$\dot{\boldsymbol{q}}$	Joint velocity	(3.4)
$\mathcal{Q}$	Unit Quaternion	(3.81)
η	Unit Quaternion scalar part	(3.82)
$\boldsymbol{\epsilon}$	Unit Quaternion vector part	(3.83)

$\mathcal{R}(.)$	Range space	(3.6)
σ_d	Sigmoid function	(6.8)
$\boldsymbol{A}_{ss}$	State space matrix	(5.39)
$\boldsymbol{B}_{ss}$	State space input matrix	(5.39)
$\boldsymbol{C}_{ss}$	State space output matrix	(5.39)
$\boldsymbol{u}$	State space input	(5.39)
$\boldsymbol{x}_r$	State space variables vector	(5.39)
$\boldsymbol{x}_{r,k}$	Discrete time state space variables vector	(5.40)
$\boldsymbol{\Gamma}_r$	Discrete time state space command matrix	(5.40)
$\boldsymbol{C}_r$	Discrete time state space measurement matrix	(5.40)
$\boldsymbol{\Phi}_r$	Discrete time state space transition matrix	(5.43)
$\boldsymbol{L}_r$	State feedback gain matrix	(5.69)
$\boldsymbol{K}_k$	Kalman gain matrix	(5.61)
$\boldsymbol{Q}_k$	System noise matrix	(5.65)
$\boldsymbol{R}_k$	Measurement noise matrix	(5.64)
K_{s_c}	Contact stiffness value	(6.7)
k_s	Environment stiffness	(6.2)
K_{s_f}	Free space stiffness value	(6.7)
$K_{s,n}$	Nominal stiffness value, used as the environment model in control design	(6.7)
$K_{s_{pc}}$	Pre-contact stiffness value	(6.7)
K_2	System damping	(5.35)
K_2^v	System damping feedback gain	(5.36)
K_2^s	System damping structural gain	(5.36)
$K_{2,n}$	Nominal system damping	(5.35)
$\boldsymbol{\tau}$	Generalized joint torque	(3.5)
$\boldsymbol{\tau}_0$	Arbitrary vector of joint torques	(3.57)
$\boldsymbol{\tau}_a$	Actuator torques	(3.47)
$\boldsymbol{\tau}_e$	External torques	(3.47)
$\boldsymbol{\tau}_f$	Friction torques	(3.47)
K_{virt}	Virtual coupling	(7.10)
$\boldsymbol{K}_{virt}$	Virtual coupling matrix	(5.34)

$\boldsymbol{x}$	End-effector position	(3.1)
$\dot{\boldsymbol{x}}$	End-effector velocity	(3.3)

CHAPTER 1

INTRODUCTION

Research motivation and goals are presented in this Chapter, as well as key contributions and the thesis structure.

1.1 Historical Perspective of Robotics

The idea of replacing humans by automated tools in mundane, dull or hazardous tasks has its roots in ancient times. Already in ancient Greece, Aristotle in his work Politics, wondered

" if every instrument could accomplish its own work, obeying or anticipating the will of others, like the statues of Daedalus, or the tripods of Hephaestus, which, says the poet, "of their own accord entered the assembly of the Gods; " if, in like manner, the shuttle would weave and the plectrum touch the lyre without a hand to guide them, chief workmen would not want servants, nor masters slaves."

Robotics is the result of pursuing this dream.

The etymology of the word "robot" comes from the Czech word "*robota*", meaning forced work, and was first used by the Czech playwright Karel Čapek in the play *Rossum's Universal Robots*, where Rossum, a fictional character, developed and mass-produced robots resembling humans to act as servants to their makers. Eventually, the robots rebelled and wiped out humanity (Clarke (1993)).

The word robotics was coined by Isaac Asimov, a biochemistry professor who was also a science fiction novelist, and defined it as: *"a science or art involving both artificial intelligence (to reason) and mechanical engineering (to perform physical acts suggested by reason)"* (Clarke (1993)). Asimov, together with fellow science fiction author John W. Campbell, developed in his books a series of principles to be used in robot conception, in order to avoid robots from hurting humans. Those principles are known as the "*Laws of Robotics*":

- **First Law:**

  A robot may not injure a human being, or through inaction, allow a human being to come to harm.

- **Second Law:**

  A robot must obey the orders given it by human beings, except where such orders would conflict with the First Law.

- **Third Law:**

  A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.

The laws were further extended, with the introduction of the zeroth law, that was devised to protect humanity as a whole (Clarke (1994))

- **Zeroth Law:**

  A robot may not injure humanity, or, through inaction, allow humanity to come to harm.

Inspired by Asimov novels, George Devol and Joseph Engelberger built the first industrial robot manipulator in the late 1950s, called "*Unimate*", which was first used in a General Motors plant for sequencing and stacking pieces of die-cast metal. Industrial robots are well suited to perform repetitive tasks that require high geometric accuracy in predefined motion trajectories and fixed time periods. Since the "*Unimate*" was first installed on an assembly line in a GM plant in Trenton, New Jersey, industrial robots have spread all over the world, being expected to reach a population of 2 million by 2017 (Brown (2015)).

1.2 Medical Robotics

Robots have changed society. The way robots couple different sources of information in physical actions, have positively influenced our society, either by increasing productivity at lower costs, or by executing tasks that otherwise would be infeasible or even impossible to be accomplished by humans. A robot can be defined as a device featuring the following characteristics (Clarke (1993)):

- programmability, which implies robots have computational capabilities;
- mechanical capability, enabling it to act on its surrounding environment;
- flexibility, since it can operate using several programs and act on objects in a variety of ways.

While being a mature technology in industrial contexts, the introduction of robots in other areas has been much slower, as is the case of the healthcare system. Medical robotics has the potential to revolutionize the healthcare system similarly to the way that robots have revolutionized industrial procedures. However, while the repetitive nature of most industrial tasks, such as welding or painting, allowed the replacement of the human worker by a robot in the assembly line, assigning the worker with a supervision role, in the medical context, rather than aiming to replace a physician by a robot, robots and physicians have to work together, side by side in a complementary fashion. Medical robots are designed to share its workspace with both patients and physicians, possibly even including the physician in the control loop, maximizing both human and robot strengths. While robots excel in geometric accuracy and in fine motions, humans have the ability to learn and make decisions in unpredictable scenarios. In this way, medical robots may be better understood as "smart" medical tools, enabling physicians to treat patients with improved safety, better precision, greater efficiency and less morbidity, simplifying and allowing to perform procedures otherwise infeasible, while minimizing patient trauma (Taylor (2006)).

With the evolution in control theory, the development of new mechanical designs and components (e.g., motors, materials), advances in medical imaging and a better understanding from physicians and patients that robots are only improved medical tools, new medical robotic-assisted procedures

are being regularly proposed. However, to be accepted by the medical community, real advantages must be obvious when comparing robotic-assisted procedures with traditional ones. Taylor (2006) summed up the three main advantages of medical robots:

1. **improvement of physicians capabilities** by exploiting both human and robotics strengths, in order to make procedures faster, accurate and less invasive or by allowing to perform procedures otherwise infeasible;

2. **improvement of procedure safety** either by improving procedure performance due to on-line monitoring or by preventing movements that may jeopardize delicate anatomical structures;

3. **procedure data recording**, being a valuable tool in medical training, skill assessment and medical certification.

Several works reporting the evolution, challenges and future directions of medical robotics exist in the literature (most focusing in robotic-assisted surgery), and some can be found in Beasley (2012), Fichtinger et al. (2008), Hoeckelmann et al. (2015), Howe and Matsuoka (1999), Kazanzides et al. (2008a,b), Najarian and Afshari (2012), and Taylor (2006).

1.3 Robotic-Assisted Ultrasound

The main area of research in medical robotics is undoubtedly surgery, being the surgical robotic system *da Vinci* the best example of a successful implementation of a robotic system in the medical field, (e.g., in 2008 in the United States, 80% of radical prostatectomies procedures where already performed using the da Vinci robot (Beasley (2012))). However, other medical areas also have potential to be favorably impacted with the development of new robotic tools. One of such areas is telemedicine. Telemedicine allows people living in remote locations to have access to medical services that, usually, are only available in areas with high density population. Among telemedicine applications, one that is particularly well suited for robotic implementation is tele-echography (Kontaxakis et al. (2000)).

1.3.1 A Brief History of Medical Ultrasound

The discovery of X-rays by Wilhelm Röntgen in 1895, was a key milestone for medical science, marking the beginning of a new era in medical diagnosis (Bradley (2008)). For the first time, physicians were able to "look" inside the human body without having to open or inserting an instrument on it. Since then, several other medical imaging techniques have emerged. Medical ultrasound is one of those techniques. It can be seen as a form of "medical sonar", where an ultrasound pulse is sent through the body and an image of traversed structures is built from reflected echoes. Nonetheless, the first medical use of ultrasounds was not as a diagnostic tool. Using the heating and disruptive effects of high intensity ultrasounds on animal tissue, ultrasounds were first employed as neurosurgical tool and as physical therapeutic tool (Woo (2006)). The first attempt to use ultrasounds as a diagnostic tool was performed by Karl Dussik in 1942, when he attempted to locate brain tumors and cerebral ventricles by measuring the attenuation of an ultrasound bean through the skull (Kane et al. (2004)). With the maturity of ultrasound as a physical therapeutic tool and the proliferation of ultrasound devices after the World War II, several researchers started to investigate its use as a diagnostic tool. In the earlier 1950's, John Wild and John Reid developed a linear hand-held B-mode ultrasound scanner and, by sweeping the scanner from side to side through breast lumps, were able to visualize breast tumors (Woo (2006)). This achievement was a key milestone, demonstrating the potential of ultrasounds for real-time imaging the interior of the human body in a safe, non-invasive and relatively inexpensive way. Ever since, ultrasound imaging has never stopped evolving, becoming a widespread and crucial diagnostic tool for a large number of medical specialties.

1.3.2 Robotic-Assisted Tele-Echography Concept

Ultrasound examination is a procedure that requires skilled and experienced physicians, which might not always be available, especially in emergency scenarios or in geographically remote areas. In traditional tele-echography procedures, a physician placed in a medical center, performs an ultrasound examination by coordinating a remote medical assistant to properly move an ultrasound probe, receiving the examination images in real-time. However, the task of properly moving the ultrasound probe is not a trivial one, requiring

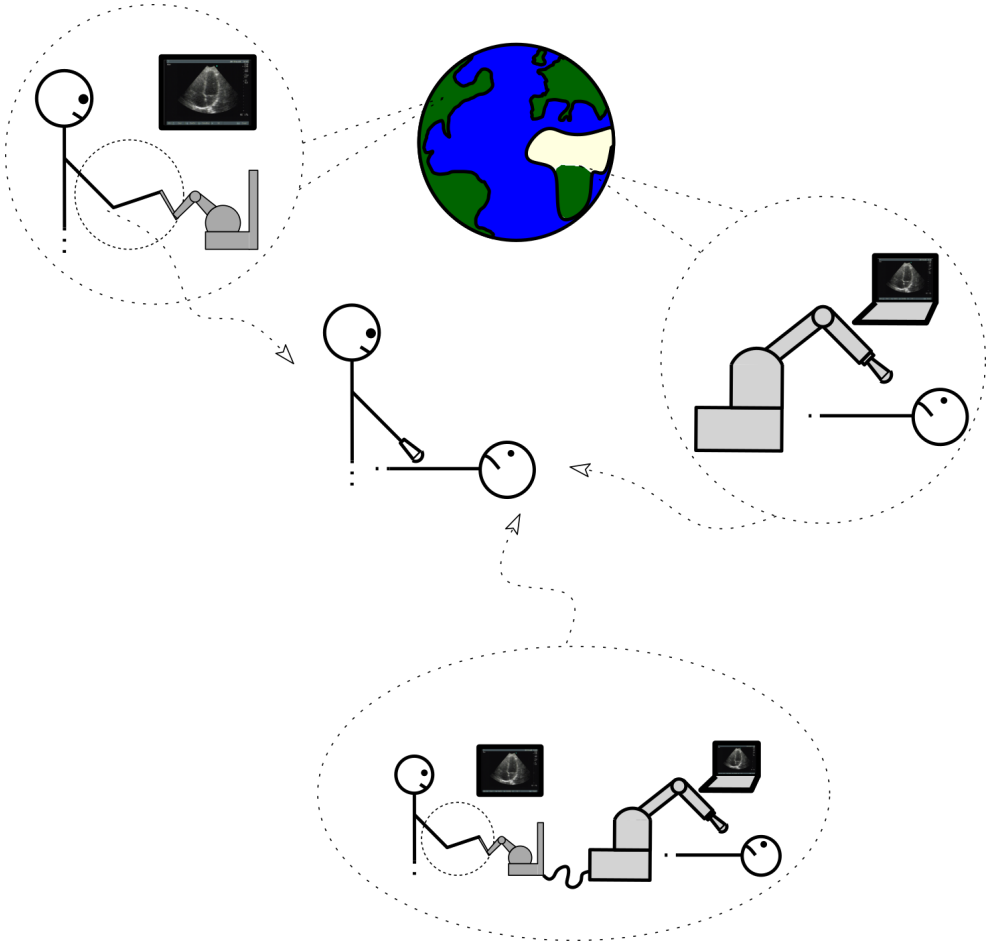


Figure 1.1— Robotic-assisted tele-echography concept. Robotic-assisted tele-echography concept. In a remote station, a robot holding an ultrasound probe is tele-operated by a physician in a master station. In the master station, the physician controls the ultrasound motion through a haptic device, enabling him with a perception of the touched environment. Real-time ultrasound and teleconference data are also sent to the physician.

a skilled assistant to acquire images with important diagnosis information, being very operator-dependent. Furthermore, some ultrasound procedures require the physician or the assistant to perform, for extended periods of time, forceful and repetitive movements in uncomfortable body postures, leading to a substantial prevalence of work-related musculoskeletal disorders among sonographers (Morton and Delf (2008)).

A robotic-assisted tele-echography system pops up as a solution to these problems. By tele-operating an ultrasound probe through a haptic device, the physician is able to perform the examination in a similar fashion to the standard procedure. Fig. 1.1 shows the robotic-assisted tele-echography concept. A slave manipulator holds an ultrasound probe that performs the examination, replicating the physicians movements performed through a haptic device in a master station. In the master station, the physician receives haptic feedback, enabling him with a perception of the touched environment. Usually the feedback force is a scaled value of contact forces, which reduces the strain felt by the physician. Real-time ultrasound and teleconference data are also sent to the master station, being both stations connected by a communication channel.

However, for a robotic-assisted ultrasound system to be adopted by the medical community, the probe motion must accurately replicate physician's movements, either in free space or under contact constraints, avoiding force peaks and jerky behaviors when moving from free-space to contact and vice-versa. Furthermore, the dynamic behavior has to be adapted according to the medical procedure, imposing interaction dynamics constraints, while the device that the physician is handling must clearly reflect the forces exerted in the patient, enabling the physician with the same tactile feeling as if he was directly performing the procedure. Patient safety should never be jeopardized by any kind of system malfunction.

1.4 Goals

The main objective of this thesis is the development of a control architecture for robotic-assisted tele-echography. This architecture has to be designed to cope with the following requirements:

- A remote center of motion (RCM) placed at the probe tip.

- A compliant contact behavior at position level.
- A stiff contact behavior at orientation level.
- High control performance in free space and in contact with soft and stiff environments, without control switching.
- Smooth transition from free space to contact and vice-versa.
- High control performance in contact when interacting with unstructured environments.

1.5 Key Contributions

This section summarizes the main contributions of this thesis.

A Dynamically Consistent Hierarchical Control Architecture

A dynamically consistent hierarchical control architecture is developed for robotic-assisted tele-echography, allowing to establish a compliant behavior at position level, while enforcing a stiff behavior at orientation level.

Motion and Contact Dynamics driven by a 3D Time-of-Flight Camera and a Force Sensor

The aim of this approach is twofold: first, it enables contact anticipation, reducing the robot dynamics before contact; second, it eliminates free space inertial forces from the control loop.

Online Stiffness Estimation using Force Data and the Robot Inertial Properties referred to the End-Effector

Due to the noisy nature of force measurements, force filtering is usually required. It is going to be shown in this thesis that when the force feedback signal is filtered, the system dynamics is no longer decoupled from the environment dynamics despite nonlinear feedback linearization. A correlation between the stiffness perceived by the controller and the manipulator inertial properties exists, and this correlation can be used in the stiffness estimation algorithm to improve force tracking performance.

1.6 Thesis Structure

The thesis is organized as follows. Chapter 1 motivates the research, establishing the main goals and key contributions. Chapter 2 reviews the solutions adopted for robotic-assisted ultrasound diagnosis and presents the experimental setup associated to this PhD thesis. Chapter 3 introduces fundamental concepts in robot manipulation and Chapter 4 reviews the fundamentals of interaction control architectures. In Chapter 5 the mathematical formulation of the hierarchical control architecture is presented, detailing the developed control strategies, while in Chapter 6 the relation between perceived stiffness and effective mass is demonstrated, being proposed a new stiffness estimation algorithm based on force measurements and manipulator inertial properties referred to the end-effector. In Chapter 7, the control architecture and the stiffness estimation algorithm are experimentally assessed, first in testbed scenarios and then validated in a clinical scenario. Chapter 8 concludes the thesis, presenting key contributions and future research directions.

CHAPTER 2

ROBOTIC-ASSISTED TELE-ECHOGRAPHY REVIEW AND SYSTEM SETUP

In the nineties, researches started to develop robotic-assisted ultrasound systems. According to Priester et al. (2013), a robotic-assisted ultrasound system can be defined as the integration of an ultrasound imaging system in a robotic platform, taking advantage of robots characteristics (e.g., accuracy, repeatability, etc...) to improve or create new medical procedures based on ultrasound imaging. Priester et al. (2013) performed a review of robotic-assisted ultrasound systems in different medical applications, dividing them in three large groups:

- Extracorporeal Diagnostic Systems (for ultrasound diagnostic procedures)
- Needle Guidance Systems
- Intraoperative Surgical Systems (ultrasound real-time imaging of hidden structures in surgical environments)

In this chapter, a literature review focusing on extracorporeal diagnostic systems is going to be performed. Additionally, our robotic-assisted tele-echography setup is also going to be described. This chapter is organized as follows. Section 2.1 provides the literature review, while Section 2.2 presents our setup.

2.1 Robotic-Assisted Tele-Echography Review

A few approaches detailing robotic-assisted ultrasound systems have already been discussed in the literature. Most solutions are based on custom made robotic systems, with the adopted solutions following one of two approaches:

- the first one, developed exclusively for tele-echography applications, addresses small and portable robots, requiring an assistant to hold and place them in the region under examination, being the physician able to control the probe orientation and perform small probe translations near the organ of interest.
- the second approach features independent robotic architectures to fully control the probe pose.

Portable Approaches

The OTELO system (Delgorge et al. (2005)) exemplifies an architecture that fall into the first approach. Taking advantage of the slave robot structure, an inverse kinematics position control with singularity avoidance is proposed by Courreges et al. (2005) for the OTELO slave robot. The TERESA robot (Vilchis et al. (2003), the ESTELE robot (Arbeille et al. (2008)) and more recently the commercially available MELODY system (Avgousti et al. (2016)) are also examples of architectures that fall into the first paradigm. When mounted in the patient body, these robotic structures are subjected to perturbation arising from unexpected assistant or patient motions, or due to patient physiological movements. Krupa et al. (2016) proposed to improve the tele-echography procedure by combining the teleoperation examination with an ultrasound image-based visual servoing, guaranteeing that the ultrasound image plane intersects the organ of interest, despite external perturbations.

Independent Robotic Structures

Solutions adopting an independent robotic architecture are more common and applied either to tele-echography as well as to autonomous ultrasound scans. One of the first approaches to introduce a robot in the echographic procedure was presented by Pierrot et al. (1999), where a force controlled manipulator was used to autonomously scan an artery for offline 3D reconstruction. The control architecture features an inner joint position loop, where joint references

are computed from its task space counterpart through inverse kinematics. Interaction forces are controlled by an outer force loop, whose output acts on the reference trajectory to cancel the force error. Zhu et al. (2000) devised a fully counter-balanced 6-degree of freedom (DOF) robot to be used as the probe holder. The control architecture presents a joint velocity control loop, with the joint velocity reference computed from its task space counterpart through differential inverse kinematics. By suitable design of a Cartesian velocity reference, in plane ultrasound visual servoing (Abolmaesumi et al. (2002)), force, and position control are all possible to be accomplished for each dimension individually. Switching between control modes is made offline. The TER system (Vilchis et al. (2003)) replaces the assistant by a parallel robotic structure. Probe orientation and the motion along the probe longitudinal axis are performed by a cable driven serial structure, with an architecture similar to the portable robots. The translation motions along the patient body are performed by a parallel robotic system. Koizumi et al. (2009) also developed a robotic-assisted tele-ecography system. The slave manipulator is impedance controlled at position level and a continuous path controller is used for orientation (Koizumi et al. (2008)). The continuous path control works by sampling and sending together, at a frequency f, n orientation samples from the master to the slave robot, where an orientation trajectory is generated by a cubic spline curve. The authors claim better orientation tracking performance when compared with the traditional point-to-point method, common in master-slave systems. However, it comes at a cost of an extra delay introduced by the orientation sampling. Monfaredi et al. (2015) presented a 6 DOFs parallel robot to control the probe motion. The robot is position controlled by a proportional-integral-derivative controller (PID) at joint level. A solution based on a commercially available robot is proposed by Mathiassen et al. (2016). They propose to control the slave manipulator by implementing a compliance control loop on top of an internal joint velocity controller. Mylonas et al. (2013) presented a different solution for remote robotic-assisted ultrasound diagnosis. Focusing in detecting intracavitary hemorrhage, they propose to perform an autonomous ultrasound scan via a learning-based control approach from expert demonstration. The manipulator is controlled by a Cartesian impedance control. Nadeau and Krupa (2013) presented an ultrasound intensity-based approach to control a robotic-assisted ultrasound system. The control architecture features a hybrid vision/force

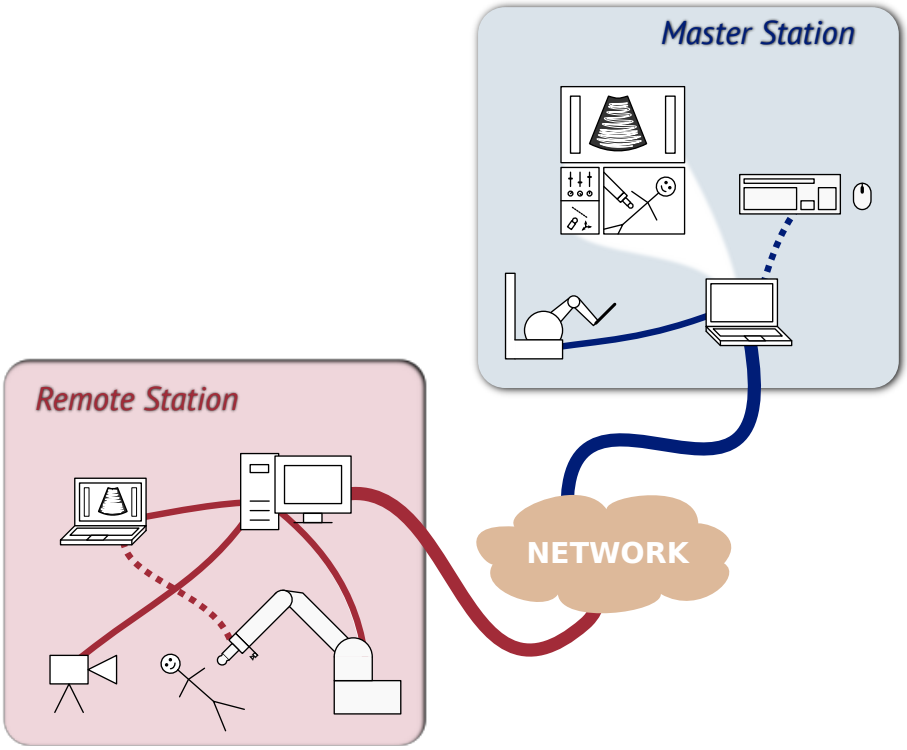


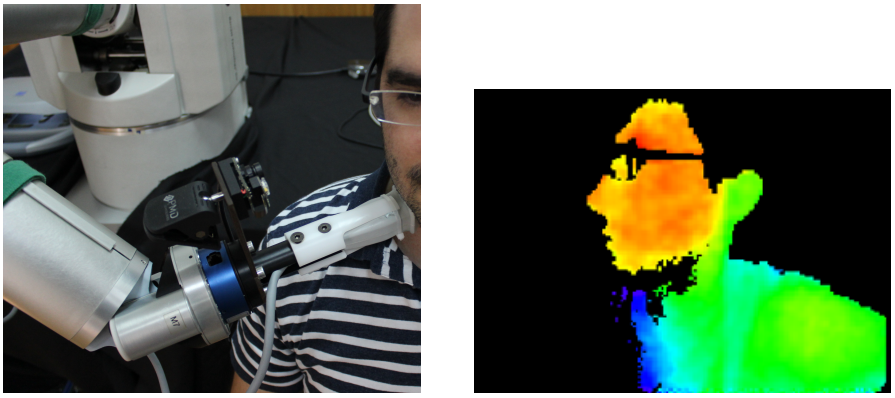
Figure 2.1— Robotic-assisted tele-echography system. The master station has a haptic device and a computer interface for ultrasound probe control. In the remote station a robotic platform interacts with the patient, being connected to the master station through a packet switched network like the Internet.

controller, with the motion normal to the contact surface controlled at force level while the remaining DOFs are controlled by ultrasound visual servoing. Nadeau et al. (2015) applied this technique for instrument and tissue tracking in 3D ultrasound guided beating heart surgery.

2.2 Robotic-Assisted Tele-Echography System and Experimental Setup

A robotic-assisted tele-echography system can be divided in three major parts:

- Master site, where the physician is located, controlling a remote ma-



(a) (b)

Figure 2.2— Slave Station End-Effector. (a) Ultrasound probe, depth camera and force sensor attached in the WAM manipulator end-effector. (b) Illustrative depth map example.

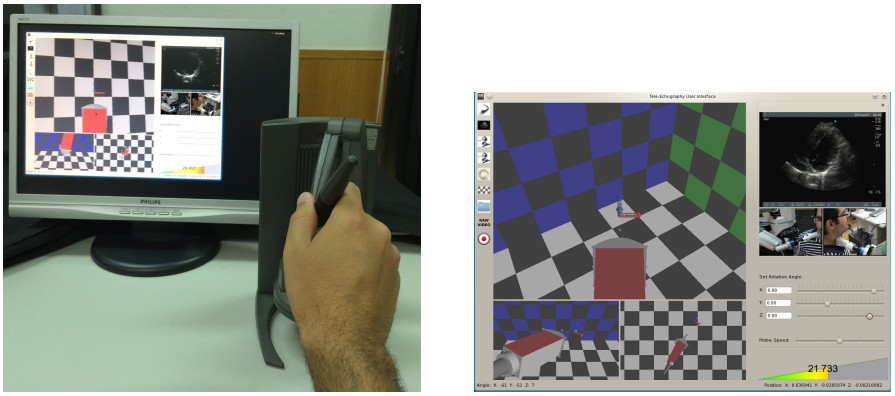
nipulator through a haptic device and receiving ultrasound and video conference images.

- Slave or Remote site, where the patient, robot and ultrasound device are placed.
- Communication link, used to exchange control, ultrasound and video conference data between master and slave station.

In Fig. 2.1 our robot-assisted tele-echography setup is depicted.

Remote Station

The remote station has an anthropomorphic robot, the Whole Arm Manipulator (WAM)^{TM} robot, with an ultrasound probe attached in its end-effector. The WAM is a torque controlled 7-DOF lightweight anthropomorphic robot from Barrett. The WAM presents high joint ranges, which provides high versatility. All joints but the last one, are cable driven (the last joint is gears driven), enabling low friction, no backlash and low opposition to external forces. Low motor-to-joint transmission ratios together with low frictions enables the *WAM* with high backdrivability. The robot controller is on an external PC with a *Linux-Xenomai* operating system, being connected to the robot through CAN bus. A JR3 67M25 force/torque sensor is attached to the end-effector. In



(a) (b)

Figure 2.3— Master Station. Through a haptic device, the physician controls the ultrasound probe position and orientation while receiving haptic feedback information. Real-time ultrasound images, video conference, real and virtual probe position are all displayed in a GUI. (a) Desktop Phantom device. (b) Graphical User Interface.

the force sensor, an ultrasound probe and a 3D time-of-flight camera are also attached (see Fig. 2.2). The ultrasound probe is a SonoSite Titan C15 curved transducer while the 3D camera is a PMD[vision]® CamBoard nano. The 3D camera is used in the stiffness estimation algorithm (see Chapter 6). Fig. 2.2a shows a depth camera and an ultrasound probe mounted in the end-effector, while Fig. 2.2b shows an illustrative depth map. Ultrasound images are acquired using a frame grabber in a separate computer, being compressed using H.264 video compression standard and sent to the master station through a packet switched network such as Internet. The streaming of compressed images is achieved using several transmission protocols: the *Real Time Streaming Protocol (RTSP)* is used to establish and control the session between server (master station) and client (remote station); video data is transmitted using *Real Time Protocol (RTP)* together with *User Datagram Protocol (UDP)*; and the *Real Time Control Protocol (RTCP)* is used to monitor the quality of service and to transport information about the members in the session. End-effector poses and interaction forces are also sent to the master station.

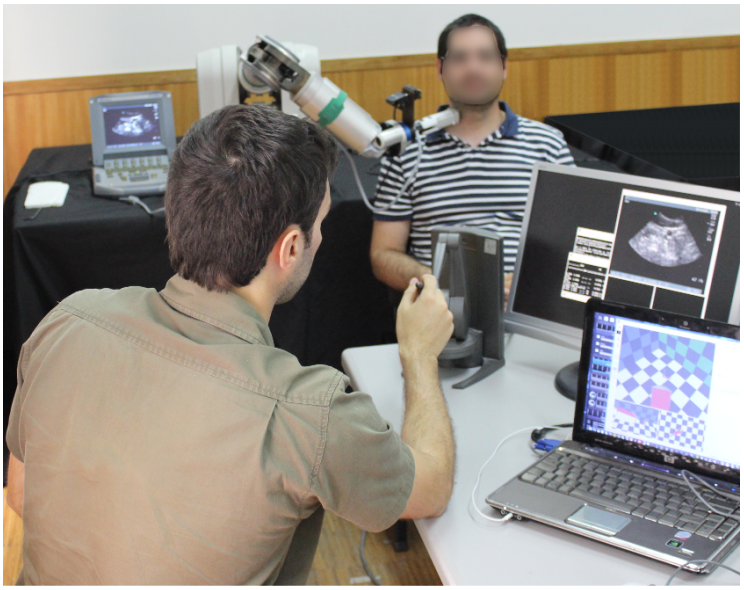


Figure 2.4— Robotic-assisted tele-echography system in a local setup. Illustrative example.

Master Station

Fig. 2.3 shows the master station. A haptic device and a Graphical User Interface (GUI) are presented for task monitoring, assessment and control. The haptic device sends 6-DOF (position and orientation) commands and receives 3-DOF force feedback. Two haptic devices are used in experiments, a Phantom Desktop and a Phantom Omni from SenSable. References can be generated by the haptic device or by moving a 3D virtual probe in a GUI, through keyboard/mouse inputs (Sousa et al. (2010)). Fig. 2.4 shows the robotic-assisted tele-echography system in a local network setup.

CHAPTER 3

BACKGROUND

A robot manipulator is as mechanical structure encompassing a sequence of rigid bodies, called links, connected by articulations that provide mobility to the structure – the joints – forming a kinematic chain. The manipulator structure can be divided in three major parts: an arm that ensures mobility, a wrist for dexterity and an end-effector to perform a task. One end of the kinematic chain is generally fixed – the base – while the other end – the end-effector – is free to move and interact with the environment. From a topological point of view, the chain can be named open, if there is only one sequence of links connecting the two ends (serial manipulator), or closed if a sequence of links forms a loop (parallel manipulator)[1].

This chapter introduces fundamental concepts and relations in robotics. The material presented in this chapter can be found in several classical robotics textbooks, e.g., Yoshikawa (1990), Murray et al. (1994), Khalil and Dombre (2004), Craig (2005), Spong et al. (2006), Siciliano et al. (2009), etc... The remainder of this chapter is organized as follows. Section 3.1 introduces the manipulator kinematics relations. Section 3.2 describes the robot dynamic model, while Section 3.3 explains how to achieve a linear and decoupled system through nonlinear feedback linearization techniques. Finally, Section 3.4 introduces the unit quaternion formulation as a rigid body orientation representation, since it is the representation used in orientation control.

[1]This work is only focused on serial manipulators.

3.1 Manipulator Kinematics

A serial-link manipulator is a set of $n+1$ links (including the base, link 0) connected by n joints. Each joint i, with $i = 1, \ldots, n$, typically has a DOF, being the manipulator characterized by the number of DOFs. The joint position vector $\boldsymbol{q}$, given by the ordered set of all joints positions $\boldsymbol{q} = \{q_1, \ldots, q_n\} \in \mathbb{R}^n$, uniquely defines the manipulator posture[2]. Knowing the manipulator geometry, it is possible to compute the end-effector configuration $\boldsymbol{x} \in \mathbb{R}^m$ in the operational[3] space as a function of joint variables $\boldsymbol{q} \in \mathbb{R}^n$. This transformation is known as *direct kinematics*[4] and is commonly performed resorting to the Denavit-Hartenberg (DH) convention (Denavit and Hartenberg (1955)). The direct kinematics is an unique and, in general, nonlinear mapping from the joint space $\boldsymbol{q}$ to the operational space $\boldsymbol{x}$,

$$\boldsymbol{x} = \mathbf{DirKinem}(\boldsymbol{q}) \ . \tag{3.1}$$

Motion specifications are usually performed in terms of end-effector positions and orientations in the operational space rather than in joint coordinates. This leads to the inverse problem that, given an end-effector pose, which joint configuration will accomplish the desired end-effector configuration? This problem is known as *inverse kinematics*

$$\boldsymbol{q} = \mathbf{InvKinem}(\boldsymbol{x}) \ , \tag{3.2}$$

and is a much more complex problem to solve, since inverse kinematics functions may result in single, multiple, infinite or even in no solutions, depending on the robot structure and if the desired end-effector pose is in the manipulator dexterous space (Siciliano et al. (2009)).

While direct and inverse kinematics establish a relationship between joint and operational space at position level, it is also possible to establish a mapping between both spaces at velocity level. The *differential kinematics* equation provides a linear mapping from joint velocities $\dot{\boldsymbol{q}} \in \mathbb{R}^n$ to end-effector velocities $\dot{\boldsymbol{x}} \in \mathbb{R}^m$, through a manipulator configuration dependent matrix called

[2]The term posture denotes the position and orientation of all rigid bodies in a kinematic chain, while the term pose denotes position and orientation of a rigid body.

[3]Throughout this document, end-effector, operational and task space will be used interchangeably.

[4]Some authors call it the direct geometric model, forward geometric model or forward kinematics.

Jacobian, $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{m \times n}$ (Spong et al. (2006)),

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \, . \tag{3.3}$$

If the manipulator is nonredundant, i.e., the number of joints and end-effector variables are equal $(n = m)$, the solution to the *inverse differential kinematics* problem is given by

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q})\dot{\boldsymbol{x}} \, , \tag{3.4}$$

where $\boldsymbol{J}^{-1}(\boldsymbol{q}) \in \mathbb{R}^{n \times n}$ is the Jacobian inverse matrix.

The Jacobian is a fundamental entity in robotics, arising in almost every aspect of robotic manipulation. It is also the Jacobian that establishes the relation between the end-effector forces $\boldsymbol{f} \in \mathbb{R}^m$ and the manipulator generalized forces (joint torques) $\boldsymbol{\tau} \in \mathbb{R}^n$. This mapping is performed through the Jacobian transpose $\boldsymbol{J}^T(\boldsymbol{q}) \in \mathbb{R}^{n \times m}$ by

$$\boldsymbol{\tau} = \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{f} \, . \tag{3.5}$$

3.1.1 Redundant Manipulators

A manipulator is said to be kinematically redundant when has more DOFs than the number of end-effector variables required to specify a given task. Regarding (3.3), the Jacobian $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{m \times n}$ performs a linear mapping from the joint velocity space $\dot{\boldsymbol{q}} \in \mathbb{R}^n$ to the end-effector velocity space $\dot{\boldsymbol{x}} \in \mathbb{R}^m$, operating between vector spaces $\mathbb{R}^n$ and $\mathbb{R}^m$ as shown in Fig. 3.1. If $m < n$, the structure is kinematically redundant with $(n - m)$ redundant DOFs, and (3.3) can be characterized by the Jacobian range and null space. The range space or column space $\mathcal{R}(\boldsymbol{J})$ is the subspace of end-effector velocities that, in the current manipulator posture, can be generated by joint velocities

$$\mathcal{R}(\boldsymbol{J}) = \{\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}, \ \dot{\boldsymbol{q}} \in \mathbb{R}^n\} \subseteq \mathbb{R}^m \, . \tag{3.6}$$

The null space $\mathcal{N}(\boldsymbol{J})$ is the subspace of joint velocities that, in the current manipulator posture, do not lead to any end-effector motion

$$\mathcal{N}(\boldsymbol{J}) = \{\boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{0}, \ \dot{\boldsymbol{q}} \in \mathbb{R}^n\} \subseteq \mathbb{R}^n \, . \tag{3.7}$$

If the Jacobian has full rank,

$$dim(\mathcal{R}(\boldsymbol{J})) = m \qquad dim(\mathcal{N}(\boldsymbol{J})) = n - m \, ,$$

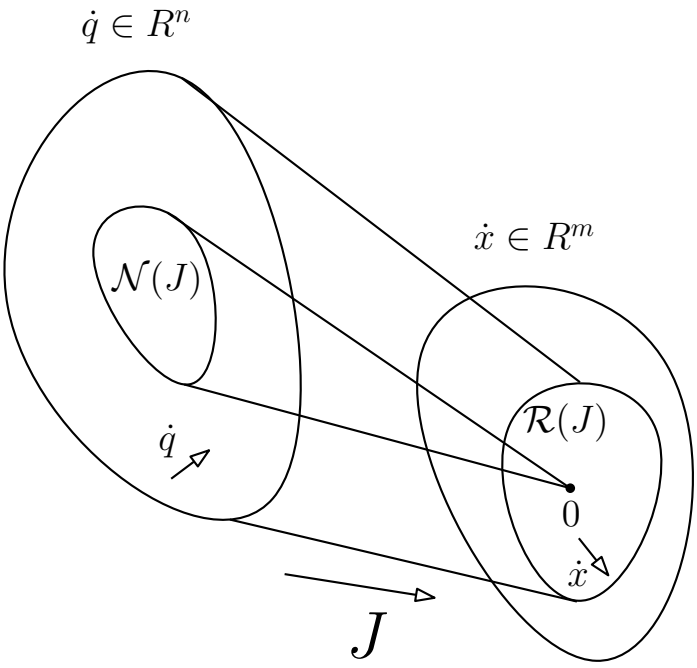


Figure 3.1— Mapping between joint and end-effector velocity spaces.

it spans the entire space $\mathbb{R}^m$. On the other hand, if the Jacobian degenerates at a singularity, the range space dimension decreases, increasing the null space dimension by the number of dimensions lost in the range space, such that

$$dim(\mathcal{R}(\boldsymbol{J})) + dim(\mathcal{N}(\boldsymbol{J})) = n$$

always hold.

Let $\boldsymbol{W} \in \mathbb{R}^{n \times n}$ be a projector to the null space of $\boldsymbol{J}(\boldsymbol{q})$, such that

$$\mathcal{R}(\boldsymbol{W}) \equiv \mathcal{N}(\boldsymbol{J}) \; .$$

If $\dot{\boldsymbol{q}}^*$ is the solution of (3.3), the joint velocity given by

$$\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}^* + \boldsymbol{W}\dot{\boldsymbol{q}}_0 \; , \tag{3.8}$$

is also a solution to (3.3), where $\dot{\boldsymbol{q}}_0$ is an arbitrary vector of joint velocities. It can be verified that (3.8) is in fact a solution of (3.3) by premultiplying both sides of (3.8) by $\boldsymbol{J}(\boldsymbol{q})$, which leads to

$$\boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}^* + \boldsymbol{J}(\boldsymbol{q})\boldsymbol{W}\dot{\boldsymbol{q}}_0 = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}^* = \dot{\boldsymbol{x}} \; , \tag{3.9}$$

since $\boldsymbol{J}(\boldsymbol{q})\boldsymbol{W}\dot{\boldsymbol{q}}_0 = \boldsymbol{0}$ for any $\dot{\boldsymbol{q}}_0$. This result shows that redundancy can be dealt in a systematic way through $\mathcal{N}(\boldsymbol{J})$. By properly setting $\dot{\boldsymbol{q}}_0$, the redundant DOFs can be used to reconfigure the manipulator posture when performing a given task, since $\boldsymbol{W}\dot{\boldsymbol{q}}_0$ will generate motions in the manipulator structure that will not be reflected in the task execution.

3.1.2 Inverse Differential Kinematics

When a manipulator is kinematically redundant, (3.3) is not uniquely defined and infinite solutions exist. A possible way to solve the inverse differential kinematics problem for a redundant manipulator is to consider it as a constrained linear optimization problem. For a given $\dot{\boldsymbol{x}}$, and knowing $\boldsymbol{J}(\boldsymbol{q})$ in a given configuration $\boldsymbol{q}$, one wants to find the solution that solves (3.3) and minimizes the quadratic cost function of joint velocities

$$r(\boldsymbol{q}) = \frac{1}{2}(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_0)^T \boldsymbol{A}(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_0) \,, \tag{3.10}$$

where $\dot{\boldsymbol{q}}$ should be close to $\dot{\boldsymbol{q}}_0$ as possible and $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix. Resorting to the Lagrange multipliers formulation, the cost function to incorporate the constraint (3.3) is given by

$$r(\boldsymbol{q}, \boldsymbol{\lambda}) = \frac{1}{2}(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_0)^T \boldsymbol{A}(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_0) + \boldsymbol{\lambda}^T (\dot{\boldsymbol{x}} - \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}) \,, \tag{3.11}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^m$ is a vector of unknown multipliers used to take into account the constraint (3.3) in the cost functional. The solution has to satisfy the necessary conditions:

$$\left(\frac{\partial r}{\partial \dot{\boldsymbol{q}}}\right)^T = \boldsymbol{0} \qquad \left(\frac{\partial r}{\partial \boldsymbol{\lambda}}\right)^T = \boldsymbol{0} \,.$$

The first condition leads to

$$\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}_0 + \boldsymbol{A}^{-1}\boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{\lambda} \,, \tag{3.12}$$

where $\boldsymbol{A}^{-1}$ exists and (3.12) is a minimum since

$$\frac{\partial^2 r}{\partial \dot{\boldsymbol{q}}^2} = \boldsymbol{A}$$

is positive definite. From the second condition, the constraint

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

is recovered. Pre-multiplying (3.12) by $\boldsymbol{J}(\boldsymbol{q})$, and then combining both conditions gives

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}_0 + \boldsymbol{J}(\boldsymbol{q})\boldsymbol{A}^{-1}\boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{\lambda} \ , \tag{3.13}$$

Solving for $\boldsymbol{\lambda}$ leads to

$$\boldsymbol{\lambda} = \left(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{A}^{-1}\boldsymbol{J}^T(\boldsymbol{q})\right)^{-1} \left(\dot{\boldsymbol{x}} - \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}_0\right) \ . \tag{3.14}$$

Replacing (3.14) in (3.12), the optimal solution is achieved

$$\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}_0 + \boldsymbol{J}^{\#}(\boldsymbol{q}) \left(\dot{\boldsymbol{x}} - \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}_0\right) \tag{3.15}$$

where

$$\boldsymbol{J}^{\#}(\boldsymbol{q}) = \boldsymbol{A}^{-1}\boldsymbol{J}^T(\boldsymbol{q}) \left(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{A}^{-1}\boldsymbol{J}^T(\boldsymbol{q})\right)^{-1} \tag{3.16}$$

is a right generalized inverse of $\boldsymbol{J}(\boldsymbol{q})$ such that

$$\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}^{\#}(\boldsymbol{q}) = \boldsymbol{I}_m$$

with $\boldsymbol{J}^{\#}(\boldsymbol{q}) \in \mathbb{R}^{n\times m}$ and $\boldsymbol{I}_m \in \mathbb{R}^{m\times m}$ as the identity matrix. If

$$\boldsymbol{A} = \boldsymbol{I}_n \ ,$$

$\boldsymbol{J}^{\#}(\boldsymbol{q})$ becomes the Moore-Penrose right pseudo-inverse of $\boldsymbol{J}(\boldsymbol{q})$ (or just right pseudo-inverse)

$$\boldsymbol{J}^{\dagger}(\boldsymbol{q}) = \left(\boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})\right)^{-1} \boldsymbol{J}^T(\boldsymbol{q}) \ , \tag{3.17}$$

which instantaneously minimizes the joint velocities norm (Chiaverini (1997)). Equation (3.15) can be rewritten as

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\dagger}(\boldsymbol{q})\dot{\boldsymbol{x}} + \left(\boldsymbol{I}_n - \boldsymbol{J}^{\dagger}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})\right)\dot{\boldsymbol{q}}_0 \ . \tag{3.18}$$

The matrix $\boldsymbol{I}_n - \boldsymbol{J}^{\dagger}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})$ has the same properties of $\boldsymbol{W}$, projecting $\dot{\boldsymbol{q}}_0$ in the null space of $\boldsymbol{J}(\boldsymbol{q})$. In this way, $\dot{\boldsymbol{q}}$ satisfies the constraint (3.3) while $\dot{\boldsymbol{q}}_0$ is a secondary objective that may or may not be fulfilled, an homogeneous term that does not induce any motion in the primary task execution. A common choice is to specify $\dot{\boldsymbol{q}}_0$ as

$$\dot{\boldsymbol{q}}_0 = \alpha \left(\frac{\partial \gamma(\boldsymbol{q})}{\partial \boldsymbol{q}}\right)^T \ , \tag{3.19}$$

where α is a positive constant and $\gamma(\boldsymbol{q})$ is a scalar objective function. Undesired manipulator configurations correspond to minima of $\gamma(\boldsymbol{q})$ while desired

configurations correspond to $\gamma(\boldsymbol{q})$ maxima. Since the gradient vector points away from minima in the direction of maxima, (3.19) points away from undesired $\gamma(\boldsymbol{q})$ minima. The manipulability measure proposed by Yoshikawa (1985), which vanishes at singular configurations, is a typical example of an objective function

$$\gamma(\boldsymbol{q}) = \sqrt{det(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q}))}\ , \tag{3.20}$$

where $det(.)$ is the matrix determinant of $\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q})$.

3.2 Manipulator Dynamic Model

The kinematic model describes the robot motion without accounting with the forces that generate the motion. Control algorithms based only on the manipulator kinematic model are able to perform low speed point-to-point motions, but unable to perform tasks that require high accelerations or to control interaction forces. To carry out such tasks it is necessary to know how motions and forces are related. The dynamic model establishes this relation.

3.2.1 Euler-Lagrange Formulation

The manipulator dynamic model establishes the relation between forces acting on the manipulator structure and its motion (position, velocities and accelerations). The Euler-Lagrange (EL) formulation allows to derive the equations of motion independently of the reference coordinate frame, provided that is possible to represent both kinetic $\mathcal{K}$ and potential energy $\mathcal{P}$ as a function of a set of *generalized coordinates*. Given a set of generalized coordinates that fully describe the configuration of a manipulator $\boldsymbol{q} = [q_1, \ldots, q_n]^T \in \mathbb{R}^n$, the Lagrangian is given by

$$\mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \mathcal{K}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \mathcal{P}(\boldsymbol{q})\ , \tag{3.21}$$

and the EL equations of motion are expressed as

$$\frac{d}{dt}\frac{\partial \mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\partial q_i} = \tau_i \qquad i = 1, \ldots, n\ , \tag{3.22}$$

where τ_i is the generalized force[5] associated with the generalized coordinate q_i. In compact form, (3.22) can be written as

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\partial \dot{\boldsymbol{q}}}\right)^T - \left(\frac{\partial \mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\partial \boldsymbol{q}}\right)^T = \boldsymbol{\tau} \ . \tag{3.23}$$

$\boldsymbol{\tau} \in \mathbb{R}^n$ is given by the forces doing work on the manipulator not derivable from a potential function, i.e., the actuators torques, the friction torques and the torques induced by the end-effector interaction with the environment.

Kinetic Energy Computation

The kinetic energy of a rigid body is given by the sum of its translational and rotational kinetic energy

$$\mathcal{K} = \frac{1}{2} m \dot{\boldsymbol{p}}_c^T \dot{\boldsymbol{p}}_c + \frac{1}{2} \boldsymbol{\omega}^T \boldsymbol{\mathcal{I}}_t \boldsymbol{\omega} \ . \tag{3.24}$$

$\dot{\boldsymbol{p}}_c \in \mathbb{R}^3$ is the linear velocity of the center of mass, $\boldsymbol{\omega} \in \mathbb{R}^3$ the angular velocity, and $\boldsymbol{\mathcal{I}}_t \in \mathbb{R}^{3\times 3}$ the inertia tensor, all expressed in the inertial frame[6]. m is the body mass. When expressed in the inertial frame, the inertia tensor is configuration dependent. However, if expressed in a frame located at its center of mass – the rigid body frame – the inertia tensor $\boldsymbol{I}_t \in \mathbb{R}^{3\times 3}$ is a constant matrix, independent of the body motion

$$\boldsymbol{I}_t = \begin{bmatrix} I_{t_{xx}} & I_{t_{xy}} & I_{t_{xz}} \\ I_{t_{yx}} & I_{t_{yy}} & I_{t_{yz}} \\ I_{t_{zx}} & I_{t_{zy}} & I_{t_{zz}} \end{bmatrix} \ , \tag{3.25}$$

with

$$\begin{aligned} I_{t_{xx}} &= \iiint_V (y^2 + z^2)\rho dV \ , \\ I_{t_{yy}} &= \iiint_V (x^2 + z^2)\rho dV \ , \\ I_{t_{zz}} &= \iiint_V (x^2 + x^2)\rho dV \ , \\ I_{t_{xy}} = I_{t_{xy}} &= -\iiint_V xy\rho dV \ , \\ I_{t_{xz}} = I_{t_{zx}} &= -\iiint_V xz\rho dV \ , \\ I_{t_{yz}} = I_{t_{zy}} &= -\iiint_V yz\rho dV \ . \end{aligned} \tag{3.26}$$

[5]The term torque is going to be used as a synonym for joint generalized force.

[6]In robotics, the inertial frame is commonly referred as the base frame. Throughout this work, both terms will be used interchangeably.

The diagonal elements $I_{t_{xx}}$, $I_{t_{yy}}$, $I_{t_{zz}}$ are the principal moments of inertia while the off diagonal elements are called cross product of inertia (Spong et al. (2006)). The integrals are computed over the volume of the rigid body and ρ is the density of the elementary particle of volume dV. The inertia tensor can be represented in the inertial frame via a similarity transformation

$$\mathcal{I}_t = \boldsymbol{R} \boldsymbol{I}_t \boldsymbol{R}^T \,, \tag{3.27}$$

where $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ is the rotation matrix from the rigid body frame to the inertial frame.

Manipulator Kinetic Energy

The kinetic energy of a n-link manipulator is given by the sum of the kinetic energy of all its n links. Equation (3.3) shows how to compute the end-effector velocity as a function of joint space velocity through the Jacobian matrix. Let the end-effector velocity be represented as $\dot{\boldsymbol{x}} = [\dot{\boldsymbol{p}} \;\; \boldsymbol{\omega}]^T$. Analogously, link i velocity is given by

$$\dot{\boldsymbol{x}}_i = \begin{bmatrix} \dot{\boldsymbol{p}}_i \\ \boldsymbol{\omega}_i \end{bmatrix} = \boldsymbol{J}_i(\boldsymbol{q})\dot{\boldsymbol{q}} = \begin{bmatrix} \boldsymbol{J}_{p_i}(\boldsymbol{q}) \\ \boldsymbol{J}_{o_i}(\boldsymbol{q}) \end{bmatrix} \dot{\boldsymbol{q}} \,, \tag{3.28}$$

with

$$\boldsymbol{J}_i(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{J}_{p_i}(\boldsymbol{q}) \\ \boldsymbol{J}_{o_i}(\boldsymbol{q}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{j}_{p_1} & \cdots & \boldsymbol{j}_{p_i} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{j}_{o_1} & \cdots & \boldsymbol{j}_{o_i} & \boldsymbol{0} & \cdots & \boldsymbol{0} \end{bmatrix} \,, \tag{3.29}$$

where $\boldsymbol{j}_{p_i} \in \mathbb{R}^3$ and $\boldsymbol{j}_{o_i} \in \mathbb{R}^3$ are, respectively, the i-th column of linear $\boldsymbol{J}_p(\boldsymbol{q}) \in \mathbb{R}^{3\times n}$ and orientation $\boldsymbol{J}_o(\boldsymbol{q}) \in \mathbb{R}^{3\times n}$ Jacobian matrices. The contribution of the Jacobians columns are taken up to link i, with the remainder $n - i$ columns a zero vector ($\boldsymbol{0} \in \mathbb{R}^3$). Similarly, the velocity of the link i center of mass expressed in the base frame, is given by

$$\begin{bmatrix} \dot{\boldsymbol{p}}_{c_i} \\ \boldsymbol{\omega}_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}^{c_i}_{p_i}(\boldsymbol{q}) \\ \boldsymbol{J}_{o_i}(\boldsymbol{q}) \end{bmatrix} \dot{\boldsymbol{q}} \,, \tag{3.30}$$

with

$$\boldsymbol{J}^{c_i}_{i}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{J}^{c_i}_{p_i}(\boldsymbol{q}) \\ \boldsymbol{J}_{o_i}(\boldsymbol{q}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{j}^{c_i}_{p_1} & \cdots & \boldsymbol{j}^{c_i}_{p_i} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{j}_{o_1} & \cdots & \boldsymbol{j}_{o_i} & \boldsymbol{0} & \cdots & \boldsymbol{0} \end{bmatrix} \,, \tag{3.31}$$

where the contribution of the Jacobians columns are taken up to the position of link i center of mass $\boldsymbol{p}_{c_i} \in \mathbb{R}^3$. For a revolute joint, the Jacobian is computed

as[7]

$$\boldsymbol{j}_{p_i}^{c_i} = \boldsymbol{z}_{j-1} \times \left(\boldsymbol{p}_{c_i} - \boldsymbol{p}_{j-1} \right) , \tag{3.32}$$

$$\boldsymbol{j}_{o_i} = \boldsymbol{z}_{j-1} , \tag{3.33}$$

where $\boldsymbol{p}_{j-1} \in \mathbb{R}^3$ and $\boldsymbol{z}_{j-1} \in \mathbb{R}^3$ are, respectively, the origin and z axis unit vector of frame $j-1$.

Taking into account the kinetic energy of all links, the manipulator kinetic energy is given by the quadratic form

$$\begin{aligned} \mathcal{K}(\boldsymbol{q}, \dot{\boldsymbol{q}}) &= \frac{1}{2} \dot{\boldsymbol{q}}^T \sum_{i=1}^{n} \left[m_i \boldsymbol{J}_{p_i}^{c_i T}(\boldsymbol{q}) \boldsymbol{J}_{p_i}^{c_i}(\boldsymbol{q}) + \boldsymbol{J}_{o_i}^T(\boldsymbol{q}) \boldsymbol{R}_i \boldsymbol{I}_{t_i} \boldsymbol{R}_i^T \boldsymbol{J}_{o_i}(\boldsymbol{q}) \right] \dot{\boldsymbol{q}} \\ &= \frac{1}{2} \dot{\boldsymbol{q}}^T \boldsymbol{B}(\boldsymbol{q}) \dot{\boldsymbol{q}} \\ &= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} b_{ij}(\boldsymbol{q}) \, \dot{q}_i \dot{q}_j \end{aligned} \tag{3.34}$$

where, for each link i, m_i is the mass, $\boldsymbol{I}_{t_i} \in \mathbb{R}^{3\times3}$ the inertia tensor computed in the coordinate frame parallel to the link reference frame, but whose origin is at its center of mass, and $\boldsymbol{R}_i \in \mathbb{R}^{3\times3}$ is the rotation matrix, expressing the link orientation in the base frame. The entity

$$\boldsymbol{B}(\boldsymbol{q}) = \sum_{i=1}^{n} \left[m_i \boldsymbol{J}_{p_i}^{c_i T}(\boldsymbol{q}) \boldsymbol{J}_{p_i}^{c_i}(\boldsymbol{q}) + \boldsymbol{J}_{o_i}^T(\boldsymbol{q}) \boldsymbol{R}_i \boldsymbol{I}_{t_i} \boldsymbol{R}_i^T \boldsymbol{J}_{o_i}(\boldsymbol{q}) \right] \tag{3.35}$$

is the manipulator *inertia matrix* ($\boldsymbol{B}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$), which is symmetric, positive definite and, in general, configuration dependent.

Potential Energy Computation

For a rigid body, gravity is the only source of potential energy[8]. The potential energy of a n-link manipulator is also given by the sum of the potential energy of all its links

$$\mathcal{P}(\boldsymbol{q}) = -\sum_{i=1}^{n} m_i \boldsymbol{g}_0^T \boldsymbol{p}_{ci} , \tag{3.36}$$

where $\boldsymbol{g}_0 \in \mathbb{R}^3$ is the gravity acceleration. Equation (3.36) shows that the potential energy is only function of the manipulator configuration $\boldsymbol{q}$.

[7] The symbol $\times$ stands for cross product.

[8] If the manipulator links exhibit some flexibility, the potential energy would include terms due to the energy stored in the elastic elements.

Equations of Motion

The equations of motions are given by (3.22). Computing the Lagrangian derivatives as

$$\frac{\partial \mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\partial \dot{q}_i} = \frac{\partial \mathcal{K}(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\partial \dot{q}_i} = \sum_{j=1}^{n} b_{ij}(\boldsymbol{q}) \dot{q}_j \ , \tag{3.37}$$

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial \mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\partial \dot{q}_i}\right) = \frac{d}{dt}\left(\frac{\partial \mathcal{K}(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\partial \dot{q}_i}\right) &= \sum_{j=1}^{n} b_{ij}(\boldsymbol{q}) \ddot{q}_j + \sum_{j=1}^{n} \frac{d b_{ij}(\boldsymbol{q})}{dt} \dot{q}_j \\ &= \sum_{j=1}^{n} b_{ij}(\boldsymbol{q}) \ddot{q}_j + \sum_{j=1}^{n} \sum_{k=1}^{n} \frac{\partial b_{ij}(\boldsymbol{q})}{\partial q_k} \dot{q}_k \dot{q}_j \ , \end{aligned} \tag{3.38}$$

$$\begin{aligned} \frac{\partial \mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\partial q_i} &= \frac{\partial \mathcal{K}(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\partial q_i} - \frac{\partial \mathcal{P}(\boldsymbol{q})}{\partial q_i} \\ &= \frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} \frac{\partial b_{ij}(\boldsymbol{q})}{\partial q_i} \dot{q}_k \dot{q}_j - \frac{\partial \mathcal{P}(\boldsymbol{q})}{\partial q_i} \ , \end{aligned} \tag{3.39}$$

and replacing (3.37), (3.38) and (3.39) in (3.22), the EL equations of motion are given by

$$\sum_{j=1}^{n} b_{ij}(\boldsymbol{q}) \ddot{q}_j + \sum_{j=1}^{n} \sum_{k=1}^{n} h_{ijk}(\boldsymbol{q}) \dot{q}_k \dot{q}_j + g_i(\boldsymbol{q}) = \tau_i \quad i = 1, \ldots, n \tag{3.40}$$

where

$$h_{ijk}(\boldsymbol{q}) = \frac{\partial b_{ij}(\boldsymbol{q})}{\partial q_k} - \frac{1}{2} \frac{\partial b_{ij}(\boldsymbol{q})}{\partial q_i} \tag{3.41}$$

and

$$\begin{aligned} g_i(\boldsymbol{q}) &= \frac{\partial \mathcal{P}(\boldsymbol{q})}{\partial q_i} \\ &= m_i \boldsymbol{g}_0^T \boldsymbol{J}_{p_i}(\boldsymbol{q}) \ . \end{aligned} \tag{3.42}$$

Three types of terms appear in (3.40). They have the following physical meaning (Siciliano et al. (2009)):

- The acceleration terms:
  - The coefficients b_{ii} are due to the moment of inertia of joint i, assuming that the other joints are blocked.
  - The coefficient b_{ij} accounts for joint j acceleration on joint i.
- The quadratic velocity terms:
  - The coefficients $h_{ijj}\dot{q}_j^2$ are due to the centrifugal effect induced on joint i by joint j velocity.

  - The terms $h_{ijk}\dot{q}_j\dot{q}_k$ are due to the Coriolis effect induced on joint i by the velocity of joints j and k .

- The configuration-dependent terms:

  - The term g_i is the torque induced on joint i due to gravity.

The manipulator equations of motion (3.40) can be written in a compact matrix representation, known as *joint space dynamic model*[9]:

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} \ . \tag{3.43}$$

$\boldsymbol{g}(\boldsymbol{q})$ is the gravity vector and $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times n}$ is the centrifugal and Coriolis matrix whose elements c_{ij} must satisfy the equation

$$\sum_{j=i}^{n} c_{ij}\dot{q}_j = \sum_{j=1}^{n}\sum_{k=1}^{n} h_{ijk}(\boldsymbol{q})\dot{q}_k\dot{q}_j \ . \tag{3.44}$$

$\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ is not unique. A common choice is to set

$$c_{ij} = \sum_{k=1}^{n} c_{ijk}\dot{q}_k \ , \tag{3.45}$$

where the coefficients

$$c_{ijk} = \frac{1}{2}\left(\frac{\partial b_{ij}}{\partial q_k} + \frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{ij}}{\partial q_i}\right) \tag{3.46}$$

are termed *Christoffel symbols of the first order*, which were obtained by interchanging the order of summation in (3.41) and by taking advantage of the symmetry of $\boldsymbol{B}(\boldsymbol{q})$.

The generalized torques $\boldsymbol{\tau}$ are given by the actuator torques $\boldsymbol{\tau}_a$, minus the friction torques $\boldsymbol{\tau}_f$ (mainly viscous and static friction) and the actuation torques used to balance torques $\boldsymbol{\tau}_e$ induced by contact forces

$$\boldsymbol{\tau} = \boldsymbol{\tau}_a - \boldsymbol{\tau}_e - \boldsymbol{\tau}_f \ . \tag{3.47}$$

[9] Along this work, the vector $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$ in the dynamic model will be represented as $\boldsymbol{c}(\boldsymbol{q},\dot{\boldsymbol{q}})$.

3.2.2 Operational Space Dynamic Model of Non-Redundant Manipulators

A task is typically specified in the end-effector configuration space[10]. Khatib (1987) presented the fundamentals of the operational space formulation, developing a unified approach for motion and force control. The idea behind this formulation is to control the end-effector motions and interaction forces by specifying end-effector control forces. For a nonredundant manipulator ($m = n$), the operational coordinates can be regarded as a set of generalized coordinates. Using the Lagrangian formulation,

$$\mathcal{L}(\boldsymbol{x}, \dot{\boldsymbol{x}}) = \mathcal{K}(\boldsymbol{x}, \dot{\boldsymbol{x}}) - \mathcal{U}(\boldsymbol{x}) \ , \tag{3.48}$$

the end-effector equations of motion are given by

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}(\boldsymbol{x}, \dot{\boldsymbol{x}})}{\partial \dot{\boldsymbol{x}}}\right)^T - \left(\frac{\partial \mathcal{L}(\boldsymbol{x}, \dot{\boldsymbol{x}})}{\partial \boldsymbol{x}}\right)^T = \boldsymbol{f} \ , \tag{3.49}$$

leading to *the operational space dynamic model*, formally analogous to its joint space counterpart (3.43)

$$\boldsymbol{\Lambda}(\boldsymbol{x})\ddot{\boldsymbol{x}} + \boldsymbol{c}_x(\boldsymbol{x}, \dot{\boldsymbol{x}}) + \boldsymbol{g}_x(\boldsymbol{x}) = \boldsymbol{f} \ . \tag{3.50}$$

$\boldsymbol{\Lambda}(\boldsymbol{x}) \in \mathbb{R}^{m \times m}$ is the operational space inertia matrix, $\boldsymbol{c}_x(\boldsymbol{x}, \dot{\boldsymbol{x}}) \in \mathbb{R}^m$ is the vector of end-effector centrifugal and Coriolis forces, $\boldsymbol{g}_x(\boldsymbol{x}) \in \mathbb{R}^m$ is the vector of gravity forces and $\boldsymbol{f} \in \mathbb{R}^m$ is the generalized operational force acting on the end-effector.

The relation between joint space and operational space acceleration is given by the second-order differential kinematics equation

$$\ddot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} \ . \tag{3.51}$$

Solving (3.51) for $\ddot{\boldsymbol{q}}$

$$\ddot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q})\ddot{\boldsymbol{x}} - \boldsymbol{J}^{-1}(\boldsymbol{q})\dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} \ , \tag{3.52}$$

and replacing (3.52) and (3.5) in (3.43), leads to

$$\boldsymbol{J}^{-T}(\boldsymbol{q})\left[\boldsymbol{B}(\boldsymbol{q})\boldsymbol{J}^{-1}(\boldsymbol{q})\ddot{\boldsymbol{x}} - \boldsymbol{B}(\boldsymbol{q})\boldsymbol{J}^{-1}(\boldsymbol{q})\dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q})\right] = \boldsymbol{f} \ , \tag{3.53}$$

[10]Throughout this document, end-effector, operational and task space will be used interchangeably.

being the relationship between joint space and operational space terms established by

$$\boldsymbol{\Lambda}(\boldsymbol{x}) = \boldsymbol{J}^{-T}(\boldsymbol{q})\boldsymbol{B}(\boldsymbol{q})\boldsymbol{J}^{-1}(\boldsymbol{q}) \,, \tag{3.54}$$

$$\begin{aligned} \boldsymbol{c}_x(\boldsymbol{x}, \dot{\boldsymbol{x}}) &= \boldsymbol{J}^{-T}(\boldsymbol{q})\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{J}^{-T}(\boldsymbol{q})\boldsymbol{B}(\boldsymbol{q})\boldsymbol{J}^{-1}(\boldsymbol{q})\dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} \\ &= \boldsymbol{J}^{-T}(\boldsymbol{q})\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{\Lambda}(\boldsymbol{x})\dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} \,, \end{aligned} \tag{3.55}$$

$$\boldsymbol{g}_x(\boldsymbol{x}) = \boldsymbol{J}^{-T}(\boldsymbol{q})\boldsymbol{g}(\boldsymbol{q}) \,. \tag{3.56}$$

3.2.3 Operational Space Dynamic Model of Redundant Manipulators

For a redundant manipulator, an independent set of end-effector configuration parameters does not constitute a generalized coordinate system ($n > m$), and its dynamics cannot be characterized by a dynamic model only in end-effector configuration coordinates (Khatib (1987)). However, the end-effector dynamics can still be described, and its equations of motion in the task space can still be found. Analogously to the inverse differential kinematics solution (3.15), the relationship between operational forces and joint torques is composed by a minimum norm term and a homogeneous term

$$\boldsymbol{\tau} = \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{f} + \left(\boldsymbol{I}_n - \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{J}^{\#^T}(\boldsymbol{q})\right)\boldsymbol{\tau}_0 \,, \tag{3.57}$$

where $\boldsymbol{J}^{\#}(\boldsymbol{q})$ is a right generalized inverse (3.16) and $\boldsymbol{\tau}_0 \in \mathbb{R}^n$ is an arbitrary vector of joint forces projected in $\mathcal{N}(\boldsymbol{J}^T)$, such that it can change the redundant joints configuration without inducing any end-effector forces. To infer how $\boldsymbol{\tau}_0$ affects the end-effector motion let's analyze the relationship between $\ddot{\boldsymbol{x}}$ and $\boldsymbol{f}$. Pre-multiplying (3.43) by $\boldsymbol{J}(\boldsymbol{q})\boldsymbol{B}^{-1}(\boldsymbol{q})$ and using both (3.51) and (3.57) in (3.43), the relation between $\ddot{\boldsymbol{x}}$ and $\boldsymbol{f}$ is given by

$$\begin{aligned} \ddot{\boldsymbol{x}} - \dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{J}(\boldsymbol{q})\boldsymbol{B}^{-1}(\boldsymbol{q})\left(\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q})\right) = \\ = \boldsymbol{J}(\boldsymbol{q})\boldsymbol{B}^{-1}(\boldsymbol{q})\left(\boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{f} + \left(\boldsymbol{I}_n - \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{J}^{\#^T}(\boldsymbol{q})\right)\boldsymbol{\tau}_0\right) \,. \end{aligned} \tag{3.58}$$

From the above expression, for $\boldsymbol{\tau}_0$ to not induce any acceleration in the end-effector motion, the second term in the right side of (3.58) must

$$\boldsymbol{J}(\boldsymbol{q})\boldsymbol{B}^{-1}(\boldsymbol{q})\left(\boldsymbol{I}_n - \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{J}^{\#^T}(\boldsymbol{q})\right)\boldsymbol{\tau}_0 = \boldsymbol{0} \,. \tag{3.59}$$

Solving for $\boldsymbol{J}^{\#}(\boldsymbol{q})$, the generalized inverse that is consistent with (3.59) is given by

$$\begin{aligned}\boldsymbol{J}^{\#}(\boldsymbol{q}) = \bar{\boldsymbol{J}}(\boldsymbol{q}) &= \boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{J}^{T}(\boldsymbol{q})\left(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{J}^{T}(\boldsymbol{q})\right)^{-1} \\ &= \boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{J}^{T}(\boldsymbol{q})\boldsymbol{\Lambda}(\boldsymbol{q}) ,\end{aligned} \tag{3.60}$$

where

$$\boldsymbol{\Lambda}(\boldsymbol{q}) = \left(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{J}^{T}(\boldsymbol{q})\right)^{-1} \tag{3.61}$$

should be seen as pseudo-kinetic energy matrix. $\bar{\boldsymbol{J}}(\boldsymbol{q})$ is also the solution of (3.3) that minimizes the manipulator's instantaneous kinetic energy (see (3.10)). Khatib (1995) called it the *dynamically consistent generalized inverse* of $\boldsymbol{J}(\boldsymbol{q})$.

Using $\bar{\boldsymbol{J}}(\boldsymbol{q})$ as the generalized inverse, the equations of motion of the end-effector can be obtained by pre-multiplying (3.58) by (3.61), leading to

$$\boldsymbol{\Lambda}(\boldsymbol{q})\ddot{\boldsymbol{x}} + \boldsymbol{c}_r(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{g}_r(\boldsymbol{q}) = \boldsymbol{f} , \tag{3.62}$$

where

$$\boldsymbol{c}_r(\boldsymbol{q},\dot{\boldsymbol{q}}) = \bar{\boldsymbol{J}}^{T}(\boldsymbol{q})\boldsymbol{c}(\boldsymbol{q},\dot{\boldsymbol{q}}) - \boldsymbol{\Lambda}(\boldsymbol{q})\dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} , \tag{3.63}$$

and

$$\boldsymbol{g}_r(\boldsymbol{q}) = \bar{\boldsymbol{J}}^{T}(\boldsymbol{q})\boldsymbol{g}(\boldsymbol{q}) . \tag{3.64}$$

$\boldsymbol{c}_r(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^m$ is a vector representing the Coriolis and centrifugal forces acting on the end-effector, while $\boldsymbol{g}_r(\boldsymbol{q}) \in \mathbb{R}^m$ is a vector representing the gravity effect on the end-effector. Equation (3.62) describes the dynamic behavior of the end-effector in the operational space, being simply the projection of the joint space dynamic model (3.43) by $\bar{\boldsymbol{J}}^{T}(\boldsymbol{q})$. The dynamically consistent relationship between the end-effector forces and the joint torques is then given by

$$\boldsymbol{\tau} = \boldsymbol{J}^{T}(\boldsymbol{q})\boldsymbol{f} + \left(\boldsymbol{I}_n - \boldsymbol{J}^{T}(\boldsymbol{q})(\boldsymbol{q})\bar{\boldsymbol{J}}^{T}(\boldsymbol{q})\right)\boldsymbol{\tau}_0 , \tag{3.65}$$

which allows the decomposition of the joint torque vector into two dynamically decoupled vectors

$$\boldsymbol{\tau} = \boldsymbol{\tau}_T + \boldsymbol{\tau}_N . \tag{3.66}$$

$\boldsymbol{\tau}_T$ are the torques due to the forces acting on the end-effector, while $\boldsymbol{\tau}_N$ are the joint torques acting on the manipulator structure, but without inducing any motion in the end-effector. In this way, the end-effector can be controlled by operational space forces, while redundancy can be handled by joint torques that are guaranteed to not change the end-effector dynamic behavior.

3.3 Nonlinear Feedback Linearization

The manipulator described by (3.43) is a nonlinear system, presenting nonlinear dependencies on posture and velocities ($\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$). With the identification of the manipulator dynamic model (Sousa (2014) and Sousa and Cortesão (2014)), it is possible to cancel the nonlinear terms through nonlinear feedback linearization, allowing to control each joint individually. Neglecting the effect of joint frictions ($\hat{\boldsymbol{\tau}}_f \approx 0$) and defining $\hat{\boldsymbol{\tau}}_e$ as an estimation of interaction torques

$$\hat{\boldsymbol{\tau}}_e = \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{f}_m \ , \tag{3.67}$$

where $\boldsymbol{f}_m \in \mathbb{R}^m$ are force and moments measurements, the commanded torque $\boldsymbol{\tau}_a$ can be specified as[11]

$$\boldsymbol{\tau}_a = \hat{\boldsymbol{c}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \hat{\boldsymbol{g}}(\boldsymbol{q}) + \hat{\boldsymbol{\tau}}_e + \boldsymbol{\tau}_c \ . \tag{3.68}$$

Replacing (3.68) in (3.43), the system reduces to

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} = \boldsymbol{\tau}_c \ , \tag{3.69}$$

where $\boldsymbol{\tau}_c \in \mathbb{R}^n$ is the torque computed by the control law. If $\boldsymbol{\tau}_c$ is set as

$$\boldsymbol{\tau}_c = \hat{\boldsymbol{B}}(\boldsymbol{q})\boldsymbol{\alpha} \ , \tag{3.70}$$

a linear and decoupled system arises

$$\ddot{\boldsymbol{q}} = \boldsymbol{\alpha} \ , \tag{3.71}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^m$ is the new control variable, *a resolved acceleration* in terms of joint variables. The new system plant is independent from the robot posture and velocity, being equivalent to a double integrator

$$\frac{Q(s)}{\alpha(s)} = \frac{1}{s^2} \ , \tag{3.72}$$

where $Q(s)$ is the joint position in the Laplace domain. In this context, $\boldsymbol{\tau}_a$ is known as *computed torque* and the control architectures developed within this work are all based on computed torque techniques.

[11] It is assumed throughout this work that robot kinematic and dynamic parameters are estimated with negligible errors. In this way, in the remainder of this thesis, the symbol $\hat{}$ is dropped for robot kinematic and dynamic parameter estimations.

3.3.1 Operational Space Nonlinear Feedback Linearization

The nonlinear feedback linearization can also be performed for the operational space formulation. Recalling that

$$\ddot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} \,, \tag{3.73}$$

$$\bar{\boldsymbol{J}}(\boldsymbol{q}) = \boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{\Lambda}(\boldsymbol{q}) \,, \tag{3.74}$$

$$\boldsymbol{\tau}_c = \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{f}_c \,. \tag{3.75}$$

where $\boldsymbol{\tau}_c \in \mathbb{R}^n$ and $\boldsymbol{f}_c \in \mathbb{R}^m$ are, respectively, torque and force vectors computed by the controller.

Pre-multiplying both sides of (3.69) by $\bar{\boldsymbol{J}}^T(\boldsymbol{q})$ and performing some simple algebraic manipulation (recall that $\boldsymbol{\Lambda}(\boldsymbol{q}) = \boldsymbol{\Lambda}^T(\boldsymbol{q})$ and $\boldsymbol{B}(\boldsymbol{q}) = \boldsymbol{B}^T(\boldsymbol{q})$ are both symmetric matrices), leads to

$$\boldsymbol{\Lambda}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}} = \boldsymbol{f}_c \,, \tag{3.76}$$

Replacing

$$\boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}} = \ddot{\boldsymbol{x}} - \dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} \,,$$

in (3.76), leads to

$$\boldsymbol{\Lambda}(\boldsymbol{q})\ddot{\boldsymbol{x}} - \boldsymbol{\Lambda}(\boldsymbol{q})\dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{f}_c \,. \tag{3.77}$$

Setting $\boldsymbol{f}_c$ in (3.77) as

$$\boldsymbol{f}_c = \boldsymbol{\Lambda}(\boldsymbol{q})\boldsymbol{f}^* - \boldsymbol{\Lambda}(\boldsymbol{q})\dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} \,, \tag{3.78}$$

a linear and decoupled system arises, representing the dynamics of a unitary mass for each Cartesian dimension

$$\ddot{\boldsymbol{x}} = \boldsymbol{f}^* \,, \tag{3.79}$$

where $\boldsymbol{f}^* \in \mathbb{R}^m$ is the new control variable. In Laplace domain

$$\frac{X(s)}{F^*(s)} = \frac{1}{s^2} \,. \tag{3.80}$$

3.4 Unit Quaternions

The orientation of a rigid body can be described using multiple representations. Typically, it is described by a rotation matrix $\boldsymbol{R} \in \mathbb{R}^{3\times3}$, although it can also be specified by other sets of three to nine parameters. For an arbitrary rotation,

four parameters representations are the representations with the least number of parameters that allow nonsingular mapping between the chosen orientation representation and their corresponding rotational matrix (Chou (1992)). Unit quaternions are a four parameter representation that comes with a complete quaternion algebra, enabling a better analysis and development of control algorithms (Caccavale et al. (1999)). The unit quaternion is defined as:

$$\mathcal{Q} = \{\eta, \boldsymbol{\epsilon}\} \ , \tag{3.81}$$

where η is called the scalar part

$$\eta = \cos\left(\frac{\vartheta}{2}\right) \ , \tag{3.82}$$

and $\boldsymbol{\epsilon} = [\epsilon_x \epsilon_y \epsilon_z]^T$ is the vector part

$$\boldsymbol{\epsilon} = \sin\left(\frac{\vartheta}{2}\right) \boldsymbol{r} \ . \tag{3.83}$$

ϑ is the angle of rotation and $\boldsymbol{r} \in \mathbb{R}^3$ is the unit vector of an equivalent angle/axis representation. Since unit quaternions are a four parameter representation for a sufficient three parameter representation, η and $\boldsymbol{\epsilon}$ are not independent, being related by the following constrain

$$\eta^2 + \boldsymbol{\epsilon}^T \boldsymbol{\epsilon} = 1 \ . \tag{3.84}$$

Given a unit quaternion, the corresponding rotation matrix is computed by

$$\boldsymbol{R} = \left(\eta^2 - \boldsymbol{\epsilon}^T \boldsymbol{\epsilon}\right) \boldsymbol{I} + 2\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T + 2\eta \boldsymbol{S}(\boldsymbol{\epsilon}) \ , \tag{3.85}$$

where $\boldsymbol{I} \in \mathbb{R}^{3\times 3}$ is the identity matrix and $\boldsymbol{S}(.)$ is the skew-symmetric matrix operator. The unit quaternion representation of a given rotation matrix

$$\boldsymbol{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \ , \tag{3.86}$$

can be obtained by

$$\eta = \frac{1}{2}\sqrt{r_{11} + r_{22} + r_{33} + 1} \, , \tag{3.87}$$

and

$$\boldsymbol{\epsilon} = \frac{1}{2} \begin{bmatrix} \operatorname{sgn}(r_{32} - r_{23})\sqrt{r_{11} - r_{22} - r_{33} + 1} \\ \operatorname{sgn}(r_{13} - r_{31})\sqrt{r_{22} - r_{33} - r_{11} + 1} \\ \operatorname{sgn}(r_{21} - r_{12})\sqrt{r_{33} - r_{11} - r_{22} + 1} \end{bmatrix} \ , \tag{3.88}$$

where

$$\text{sgn}(x) = \begin{cases} 1 & \text{if} \quad x \geq 0 \\ -1 & \text{if} \quad x < 0 \end{cases} . \tag{3.89}$$

From (3.87) and (3.82), $\eta \geq 0$ and $\vartheta \in [-\pi, \pi]$, which allows to represent any rotation without the ambiguity problem of the angle/axis representation (a rotation by ϑ about $\boldsymbol{r}$ gives the same unit quaternion as a rotation by $-\vartheta$ about $-\boldsymbol{r}$). $\boldsymbol{R}^{-1}$ is given by $\mathcal{Q}^{-1}$

$$\mathcal{Q}^{-1} = \{\eta, -\boldsymbol{\epsilon}\} \ , \tag{3.90}$$

where $\mathcal{Q}^{-1}$ is the conjugate of $\mathcal{Q}$.

Let $\mathcal{Q}_a = \{\eta_a, \boldsymbol{\epsilon}_a\}$ and $\mathcal{Q}_b = \{\eta_b, \boldsymbol{\epsilon}_b\}$ be the corresponding unit quaternion representation of $\boldsymbol{R}_a$ and $\boldsymbol{R}_b$ respectively. In the quaternion framework, the product $\boldsymbol{R}_a\boldsymbol{R}_b$ is given by

$$\mathcal{Q}_a * \mathcal{Q}_b = \left\{ \eta_a\eta_b - \boldsymbol{\epsilon}_a^T\boldsymbol{\epsilon}_b \ , \eta_a\boldsymbol{\epsilon}_b + \eta_b\boldsymbol{\epsilon}_a + \boldsymbol{\epsilon}_a \times \boldsymbol{\epsilon}_b \right\} \ , \tag{3.91}$$

where $*$ is the quaternion product operator. The product of a quaternion by its conjugate gives the identity element of the product of quaternions

$$\mathcal{Q} * \mathcal{Q}^{-1} = \{1, \boldsymbol{0}\} \ . \tag{3.92}$$

The relationship between the unit quaternion time derivative and the rigid body angular velocity $\boldsymbol{\omega}$ is given by

$$\dot{\eta} = -\frac{1}{2}\boldsymbol{\epsilon}^T\boldsymbol{\omega} \ , \tag{3.93}$$

$$\dot{\boldsymbol{\epsilon}} = \frac{1}{2}\boldsymbol{E}(\eta, \boldsymbol{\epsilon})\boldsymbol{\omega} \ , \tag{3.94}$$

with

$$\boldsymbol{E}(\eta, \boldsymbol{\epsilon}) = \eta\boldsymbol{I} - \boldsymbol{S}(\boldsymbol{\epsilon}) \ . \tag{3.95}$$

3.4.1 Mutual Orientation in the Quaternion Framework

Let $\boldsymbol{R}_a$ and $\boldsymbol{R}_b$ be the the orientation of two different frames in the same base frame. The mutual orientation between these two frames is given by

$${}^a\boldsymbol{R}_b = \boldsymbol{R}_a^T\boldsymbol{R}_b \ . \tag{3.96}$$

The unit quaternion representing the mutual orientation can be obtained from ${}^a\boldsymbol{R}_b$ using (3.87) and (3.88) or computed by composition of unit quaternions

$${}^a\mathcal{Q}_{ab} = \{\eta_{ab}, {}^a\boldsymbol{\epsilon}_{ab}\} = \mathcal{Q}_a^{-1} * \mathcal{Q}_b \ , \tag{3.97}$$

where the double subscript has been used to indicate a mutual orientation. η_{ab} and ${}^{a}\boldsymbol{\epsilon}_{ab}$ are given by

$$\eta_{ab} = \eta_a \eta_b + \boldsymbol{\epsilon}_a \boldsymbol{\epsilon}_b^T \,, \tag{3.98}$$

$${}^{a}\boldsymbol{\epsilon}_{ab} = \eta_a \boldsymbol{\epsilon}_b - \eta_b \boldsymbol{\epsilon}_a - \boldsymbol{S}(\boldsymbol{\epsilon}_a)\boldsymbol{\epsilon}_b \,. \tag{3.99}$$

The relationship between the unit quaternion time derivative and the angular velocity is given by (Caccavale et al. (1999))

$$\dot{\eta}_{ab} = -\frac{1}{2} {}^{a}\boldsymbol{\epsilon}_{ab}{}^{T} {}^{a}\boldsymbol{\omega}_{ab} \,, \tag{3.100}$$

$${}^{a}\dot{\boldsymbol{\epsilon}}_{ab} = \frac{1}{2}\boldsymbol{E}(\eta_{ab}, {}^{a}\boldsymbol{\epsilon}_{ab}) {}^{a}\boldsymbol{\omega}_{ab} \,, \tag{3.101}$$

where ${}^{a}\boldsymbol{\omega}_{ba}$ is the angular velocity of frame b relative to frame a, represented in frame a

$${}^{a}\boldsymbol{\omega}_{ba} = {}^{a}\boldsymbol{\omega}_b - {}^{a}\boldsymbol{\omega}_a \,. \tag{3.102}$$

and

$${}^{a}\boldsymbol{\omega}_{ba} = 2\, {}^{a}\dot{\mathcal{Q}}_{ab} {}^{a}\mathcal{Q}_{ab}^{-1} \,. \tag{3.103}$$

CHAPTER 4

INTERACTION CONTROL ARCHITECTURES

Controlling contact forces is critical for manipulation tasks that require interaction with the environment. To perform tasks such polishing, drilling, pushing or cutting, it is important to limit the range of interaction forces to improve task execution and safety. Motion controllers are well suited for positioning tasks but not for interaction. As a matter of fact, the better a motion controller performs, less suitable it is to interact in a constrained environment, especially with stiff surfaces, since a small position error in task planning might lead to prohibitively high contact forces, resulting either in joint actuators saturation or, in the worst case scenario, in breakage of contacting parts. The control of contact forces when the manipulator shares its workspace with humans is even a more critical issue, as is the case of a robotic-assisted tele-echography procedure, where an excessive contact force might result in patient trauma.

Starting with a task space impedance controller with inverse dynamics decoupling, this chapter and Chapter 5 describe the control architecture evolution, detailing adopted solutions to improve robotic-assisted tele-echography procedures. This chapter is organized as follows. In Section 4.1 literature review about interaction control is presented. Impedance control is presented in Section 4.2, while in Section 4.3 a new control approach using both joint and operational space formalism is discussed. Finally, Section 4.4 concludes the chapter.

4.1 Interaction Control Review

Control of interaction forces can be achieved either through *indirect force control*, when interaction forces are controlled through a motion controller, or by *explicit force control*, when contact forces are explicitly controlled to follow a desired force reference (Whitney (1985)).

4.1.1 Indirect Force Control

As its name suggests, indirect force control handles interaction forces without explicitly closing a force feedback loop. Stiffness (or compliance) and impedance control are two examples of control architectures that fall into this category. In stiffness control (Salisbury (1980)), the manipulator is controlled to exhibit a desired steady state behavior, acting as a generalized spring in contact with the environment. A compliant behavior can be achieved by designing the controller with a low stiffness value, although such designs lead to poor free space position tracking. High stiffness design improves free space control, however it might also lead to prohibitively high contact forces in stiff environments. If the environment is known, a trade-off between position tracking and compliant motion can be achieved, assigning high stiffness values in directions where contact is not expected and low values in kinematic constrained directions, ensuring in this way that contact forces are bounded.

Impedance control generalizes this approach, specifying not only the static behavior but also the manipulator dynamic behavior. In impedance control (Hogan (1985)), the manipulator dynamics matches the dynamics of a mechanical impedance, i.e., a mass-spring-damper system. Similarly to stiffness control, also impedance parameter design requires a compromise between free space tracking accuracy and compliant contact behavior. To increase disturbance rejection, motion control can be separated from impedance control. In admittance control (Chiaverini et al. (1999)), an outer impedance loop generates a compliant motion reference to the inner motion loop. The inner loop ensures high position tracking performance, while the outer loop is designed to provide the desired compliant behavior. Nevertheless, admittance control requires the availability of force measurements, and to limit contact forces, control parameters must also take into account environment dynamics (specially the environment stiffness).

4.1.2 Direct Force Control

Indirect force controllers limit interaction forces by specifying contact dynamics, suitably controlling the end-effector motion. In this way, if the environment is at least roughly known, high interaction forces can be avoided. However, if the task requires a precise control of interaction forces, explicit force control is required. To perform complex interaction tasks, in addition to control contact forces, it is also necessary to control the end-effector configuration. In contact, the environment imposes kinematic constraints on the manipulator, not allowing an arbitrary motion along constrained directions. On the other hand, along unconstrained directions it is not possible to apply an arbitrary force. However, it is possible (and desired) to control interaction forces in kinematic constrained directions and the end-effector posture along unconstrained ones. Following this reasoning, Mason (1981) states that the environment establishes either a motion or a force constraint, classifying the environment constraints along a DOF, as *natural constraint*. A controller is only able to control the *artificial constraints*, i.e., the environment unconstrained variable, not being possible to simultaneously enforce an arbitrary position and force along the same direction (Mason (1981)).

Due to this duality between motion and force control, most classical controllers follow a hybrid position/force control paradigm (Railbert and Craig (1981), Khatib (1987)). Motion and force controlled directions are selected through a selection matrix, being a number of DOFs force controlled while the remaining DOFs are controlled at motion level. This approach requires a precise knowledge of the environment geometry, otherwise, a small error in the environment geometric model or in task planning would introduce undesired behaviors in motion and force controlled directions. To overcome these limitations, solutions that combine force and motion control on a single direction have also been proposed (Chiaverini and Sciavicco (1993), De Schutter and Van Brussel (1988)). The idea behind parallel force/position control (Chiaverini and Sciavicco (1993)) is to combine the robustness of impedance/admittance control with the ability to achieve both force and position control of hybrid approaches, increasing the controller robustness to model uncertainties or planning errors. Force and position are both controlled in a single direction by superposition, with the force controller generating a suitable motion reference for a low level motion controller. Force control prevails over motion control in conflicting situations, i.e., force is regulated at expense of position errors

(De Schutter et al. (1998)), by introducing an integral action in the force loop (typically implemented by a proportional-integral controller (PI)), while at position level, a proportional-derivative controller (PD) is implemented.

Force Control Architectures

Force control architectures are usually designed taking in account a few idealizations. The environment is usually considered to behave as a linear spring and the manipulator is regarded to be infinitely stiff. A stabilizing damping action is also required. Since force measurements are usually corrupted with noise, the damping action is typically provided by velocity terms, meaning that a force controller implies, at least, force and motion measurements (velocity and/or position measurements). As De Schutter et al. (1998) states: *"every force control law adds position, velocity and/or force errors together in some way or another, and use the result to generate set-points for the joint actuators."* The way these errors from different physical nature are combined, is what distinguishes different force control approaches.

Early force control approaches followed, mostly, PI designs with static model-based compensation (gravity compensation) (De Schutter and Van Brussel (1988)). Volpe et al. (1993) performed a theoretic analysis and experimental testing on a set of basic controllers, e.g., proportional, integrative, PD, etc.., highlighting their strengths and weaknesses for force control. More complex architectures grounded on model-based control strategies have also been proposed. If the robot dynamic model is known, inverse dynamics control allows to achieve a linear and decoupled system, allowing to apply the rich and vast theory of linear systems in force control. Control designs that do not rely on perfect dynamics compensation have also been proposed. Robust force control techniques have been presented by Dawson et al. (1992) and Fraisse et al. (2007). Passivity base controllers also present enhanced robustness, since the controller does not rely on a perfect cancellation of robot nonlinearities (Sage et al. (1999) and Siciliano and Villani (2000)), being used to control either rigid robots (Siciliano and Villani (1996) and Villani et al. (1999)), as well as to control robots with flexible joints (Albu-Schäffer et al. (2007) and Schindlbeck and Haddadin (2015)). Adaptive model-based force control techniques have also been addressed (e.g., Whitcomb et al. (1997)).

4.2 Impedance Control

Impedance control (Hogan (1985)) is a natural starting point in the control architecture development. Due to the unstructured nature of the environment in echographic tasks, with free space motion interchanging with contact events, impedance control presents a suitable solution from a theoretical point view, providing motion control in free space, while enabling indirect force control (open-loop) in contact. Furthermore, while interaction forces must be limited and are procedure specific, to successfully perform an ultrasound examination it is not strictly necessary to have fine control over contact forces, making indirect force control approach a suitable solution. Impedance control relates position errors with contact forces through a mechanical impedance of adjustable parameters, i.e., the robot dynamic behavior is specified to match the dynamics of a mass-spring-damper system. In the Laplace domain, this relation (for one Cartesian dimension) is given by

$$F_e(s) = Z(s)X(s) \ , \tag{4.1}$$

with

$$Z(s) = Ms^2 + Ds + K \ , \tag{4.2}$$

where $F_e(s)$ and $X(s)$ are, respectively, contact force and position error in the Laplace domain. M, D and K are, respectively, mass, damping and stiffness parameters. In time domain, the desired impedance equation is given by

$$\boldsymbol{f}_m = \boldsymbol{M}\left(\ddot{\boldsymbol{x}}_d - \ddot{\boldsymbol{x}}_r\right) + \boldsymbol{D}\left(\dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}}_r\right) + \boldsymbol{K}\left(\boldsymbol{x}_d - \boldsymbol{x}_r\right) \tag{4.3}$$

where $\boldsymbol{M} \in \mathbb{R}^{m\times m}$, $\boldsymbol{D} \in \mathbb{R}^{m\times m}$ and $\boldsymbol{K} \in \mathbb{R}^{m\times m}$ are mass, damping and stiffness matrices, while $\boldsymbol{x}_r, \dot{\boldsymbol{x}}_r, \ddot{\boldsymbol{x}}_r \in \mathbb{R}^m$ and $\boldsymbol{x}_d, \dot{\boldsymbol{x}}_d, \ddot{\boldsymbol{x}}_d \in \mathbb{R}^m$ are, respectively, robot and reference kinematic quantities. The manipulator dynamics in contact is given by (3.43) ($\boldsymbol{\tau}_f \approx 0$)

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}_a - \boldsymbol{\tau}_e \ .$$

Computing $\boldsymbol{\tau}_a$ as in (3.68)

$$\boldsymbol{\tau}_a = \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{\tau}_c + \boldsymbol{\tau}_e \ , \tag{4.4}$$

leads to

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} = \boldsymbol{\tau}_c \ , \tag{4.5}$$

where $\boldsymbol{\tau}_c \in \mathbb{R}^n$ is the torque computed by the control law $\boldsymbol{f}_c$. Pre-multiplying both sides of (4.5) by $\boldsymbol{J}(\boldsymbol{q})\boldsymbol{B}(\boldsymbol{q})^{-1}$, one gets

$$\boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})\boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{\tau}_c \ . \tag{4.6}$$

Recall that

$$\begin{aligned} \boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}} &= \ddot{\boldsymbol{x}}_r - \dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} \ , \\ \boldsymbol{\tau}_c &= \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{f}_c \ , \\ \boldsymbol{\Lambda}^{-1}(\boldsymbol{q}) &= \boldsymbol{J}(\boldsymbol{q})\boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q}) \ . \end{aligned}$$

Replacing these expressions in (4.6) leads to

$$\ddot{\boldsymbol{x}}_r - \dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})\boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{f}_c \ , \tag{4.7}$$

and finally to

$$\boldsymbol{\Lambda}(\boldsymbol{q})\ddot{\boldsymbol{x}}_r - \boldsymbol{\Lambda}(\boldsymbol{q})\dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{f}_c \ . \tag{4.8}$$

Making $\boldsymbol{f}_c$ in (4.8) as

$$\boldsymbol{f}_c = \boldsymbol{\Lambda}(\boldsymbol{q})\boldsymbol{f}^* - \boldsymbol{\Lambda}(\boldsymbol{q})\dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} \ , \tag{4.9}$$

a linear and decoupled system arises, representing the dynamics of a unitary mass

$$\ddot{\boldsymbol{x}}_r = \boldsymbol{f}^* \ , \tag{4.10}$$

where $\boldsymbol{f}^* \in \mathbb{R}^m$ is the new control variable. Solving (4.3) for $\ddot{\boldsymbol{x}}_r$ and replacing it by $\boldsymbol{f}^*$ leads to

$$\boldsymbol{f}^* = \ddot{\boldsymbol{x}}_d + \boldsymbol{M}^{-1}\left(\boldsymbol{D}\left(\dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}}_r\right) + \boldsymbol{K}\left(\boldsymbol{x}_d - \boldsymbol{x}_r\right) - \boldsymbol{f}_m\right) \ . \tag{4.11}$$

Replacing (4.11) in (4.9), $\boldsymbol{f}_c$ becomes

$$\boldsymbol{f}_c = \boldsymbol{\Lambda}(\boldsymbol{q})\left(\ddot{\boldsymbol{x}}_d - \dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}}\right) + \boldsymbol{\Lambda}(\boldsymbol{q})\boldsymbol{M}^{-1}\left(\boldsymbol{D}\left(\dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}}_r\right) + \boldsymbol{K}\left(\boldsymbol{x}_d - \boldsymbol{x}_r\right) - \boldsymbol{f}_m\right) \tag{4.12}$$

The computed torque in (4.4) is then given by

$$\begin{aligned} \boldsymbol{\tau}_a =& \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{f}_m + \\ &+ \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{\Lambda}(\boldsymbol{q})\left(\ddot{\boldsymbol{x}}_d - \dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}}\right) + \\ &+ \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{\Lambda}(\boldsymbol{q})\boldsymbol{M}^{-1}\left(\boldsymbol{D}\left(\dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}}_r\right) + \boldsymbol{K}\left(\boldsymbol{x}_d - \boldsymbol{x}_r\right) - \boldsymbol{f}_m\right) \ , \end{aligned} \tag{4.13}$$

where the relation

$$\boldsymbol{\tau}_e = \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{f}_m \ ,$$

has been used. Fig. 4.1 shows a generic impedance control architecture.

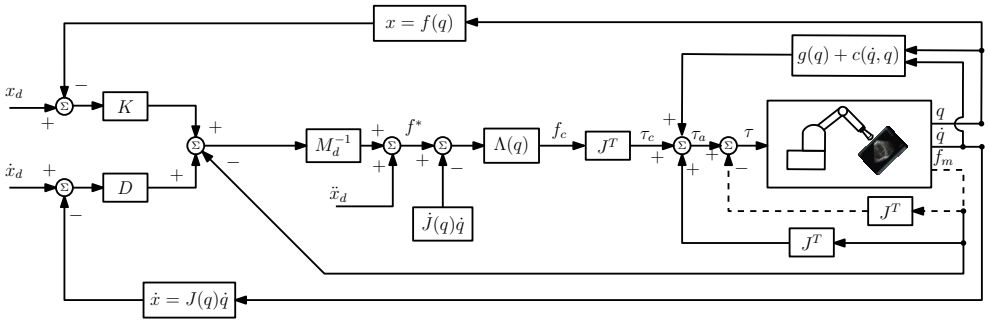


Figure 4.1— Impedance control. Task space impedance control with inverse dynamics nonlinear feedback linearization.

4.2.1 Linear and Rotational Impedance

Let $\boldsymbol{f}^* \in \mathbb{R}^6$ be a 6-DOF control variable

$$\boldsymbol{f}^* = \begin{bmatrix} \boldsymbol{f}_p^* \\ \boldsymbol{f}_o^* \end{bmatrix} \tag{4.14}$$

with $\boldsymbol{f}_p^* \in \mathbb{R}^3$ and $\boldsymbol{f}_o^* \in \mathbb{R}^3$ linear and rotational components, respectively.

Linear Impedance

$\boldsymbol{f}_p^*$ is given by

$$\boldsymbol{f}_p^* = \ddot{\boldsymbol{p}}_d + \boldsymbol{M}_{d_p}^{-1} \left(\boldsymbol{K}_{d_p} (\boldsymbol{p}_d - \boldsymbol{p}_r) + \boldsymbol{D}_{d_p} (\dot{\boldsymbol{p}}_d - \dot{\boldsymbol{p}}_r) - \boldsymbol{f}_{m_p} \right) , \tag{4.15}$$

where $\boldsymbol{M}_{d_p} \in \mathbb{R}^{3\times3}$, $\boldsymbol{K}_{d_p} \in \mathbb{R}^{3\times3}$ and $\boldsymbol{D}_{d_p} \in \mathbb{R}^{3\times3}$ are diagonal and positive definite matrices, representing the linear mass, stiffness and damping, respectively. $\boldsymbol{p}_d \in \mathbb{R}^3$ and $\boldsymbol{p}_r \in \mathbb{R}^3$ are, respectively, reference and robot position while $\dot{\boldsymbol{p}}_d \in \mathbb{R}^3$ and $\dot{\boldsymbol{p}}_r \in \mathbb{R}^3$ are reference and robot velocity, respectively. $\boldsymbol{f}_{m_p} \in \mathbb{R}^3$ is the contact force.

Rotational Impedance

Similarly, $\boldsymbol{f}_o^*$ is given by

$$\boldsymbol{f}_o^* = \dot{\boldsymbol{\omega}}_d + \boldsymbol{M}_{d_o}^{-1} \left(\boldsymbol{K}_{d_o} \boldsymbol{\epsilon}_{rd} + \boldsymbol{D}_{d_o} (\boldsymbol{\omega}_d - \boldsymbol{\omega}_r) - \boldsymbol{f}_{m_o} \right) , \tag{4.16}$$

where $\boldsymbol{M}_{d_o} \in \mathbb{R}^{3\times3}$, $\boldsymbol{K}_{d_o} \in \mathbb{R}^{3\times3}$ and $\boldsymbol{D}_{d_o} \in \mathbb{R}^{3\times3}$ are diagonal and positive definite matrices, representing the rotational mass, stiffness and damping,

respectively and $\boldsymbol{f}_{m_o} \in \mathbb{R}^3$ is the contact moment. $\boldsymbol{\omega}_d \in \mathbb{R}^3$ and $\boldsymbol{\omega}_r \in \mathbb{R}^3$ are, respectively, reference and robot angular velocity, given by (3.103). $\boldsymbol{\epsilon}_{rd} \in \mathbb{R}^3$ is the orientation error computed resorting to unit quaternion theory. Let $\mathcal{Q}_d = \{\eta_d, \boldsymbol{\epsilon}_d\}$ and $\mathcal{Q}_r = \{\eta_r, \boldsymbol{\epsilon}_r\}$ be, respectively, reference and robot orientation expressed in the unit quaternion framework (Caccavale et al. (1999)). η and $\boldsymbol{\epsilon}$ are, respectively, the scalar and vector part of the unit quaternion. The mutual orientation between $\mathcal{Q}_d$ and $\mathcal{Q}_r$, expressed in the robot frame ${}^r\mathcal{Q}_{dr} = \{\eta_{dr}, {}^r\boldsymbol{\epsilon}_{dr}\}$ is given as

$$ {}^r\mathcal{Q}_{dr} = \mathcal{Q}_r^{-1} * \mathcal{Q}_d \ , \tag{4.17} $$

Mapping (4.17) onto the base frame

$$ \begin{aligned} \mathcal{Q}_{dr} &= \mathcal{Q}_r * {}^r\mathcal{Q}_{dr} * \mathcal{Q}_r^{-1} \\ &= \mathcal{Q}_d * \mathcal{Q}_r^{-1} \ , \end{aligned} \tag{4.18} $$

while acknowledging that $\mathcal{Q}_{dr} = \{1, \mathbf{0}\}$ only if reference and robot frame are aligned, a minimal orientation error representation can be defined as (see Siciliano et al. (2009))

$$ \boldsymbol{\epsilon}_{dr} = \eta_r \boldsymbol{\epsilon}_d - \eta_d \boldsymbol{\epsilon}_r - \boldsymbol{S}(\boldsymbol{\epsilon}_d)\boldsymbol{\epsilon}_r \ , \tag{4.19} $$

where $\boldsymbol{S}(.)$ is the skew-symmetric operator.

4.2.2 Control Strategy

When performing an ultrasound examination, applied 3D Cartesian forces vary with medical procedure itself and patient body anatomy. On the other hand, at orientation level, to obtain a desired view of anatomical regions, the probe should be held in a desired orientation, presenting no compliant behavior, such that the ultrasound image plane intersects the organ of interest. To cope with these requirements, the impedance controller should be able to present a compliant behavior at position level, while at orientation level the controller must enforce stiff behaviors. In this way, the impedance controller should be designed with relatively low control gains for position dimensions, while high gains should be employed to present the desired stiff behavior at orientation level.

4.2.3 Experiments

When testing this controller, it became clear that a compliant position behavior with a stiff orientation would be unachievable with a satisfactory performance

both in free space and in contact, requiring a different control approach. It is well known that the selection of impedance parameters that leads to a good performance both in free space and in contact, specially with stiff objects, is difficult (Siciliano and Villani (2000). However, it was still important to assess the control performance in free space, since it is part of the medical procedure.

Impedance Control Design

In free space, without contact forces, the impedance controller becomes a PD task space controller with inverse dynamics decoupling if

$$\boldsymbol{M}_{d_p} = \boldsymbol{M}_{d_o} = \text{diag}\begin{bmatrix}1 & 1 & 1\end{bmatrix} .$$

Assuming a perfect linearization, the input/output relation for each task dimension is given by (in Laplace domain)

$$\frac{X_r(s)}{X_d(s)} = \frac{Ds + K}{s^2 + Ds + K} ,$$

with natural frequency

$$\omega_n = \sqrt{K} , \tag{4.20}$$

and damping ratio

$$\zeta = \frac{D}{2\omega_n} . \tag{4.21}$$

D is computed to assign a critically damped behavior ($\zeta = 1$). The feed-forward acceleration term in (4.15) and in (4.16) is usually neglected in tele-operation tasks, since its computation through velocity differentiation is very noisy (the reference trajectory is not known in advance). Two position impedance designs[1] were tested performing ultrasound examination motions in free space, a stiffer

$$\boldsymbol{K}_{d_{p_{stiff}}} = \text{diag}\begin{bmatrix}500 & 500 & 500\end{bmatrix} \quad \text{and} \quad \boldsymbol{D}_{d_{p_{stiff}}} = \text{diag}\begin{bmatrix}45 & 45 & 45\end{bmatrix} ,$$

and a more compliant one

$$\boldsymbol{K}_{d_{p_{compli}}} = \text{diag}\begin{bmatrix}300 & 300 & 300\end{bmatrix} \quad \text{and} \quad \boldsymbol{D}_{d_{p_{compli}}} = \text{diag}\begin{bmatrix}35 & 35 & 35\end{bmatrix} .$$

The orientation control is projected to present a stiff behavior

$$\boldsymbol{K}_{d_o} = \text{diag}\begin{bmatrix}500 & 500 & 500\end{bmatrix} \quad \text{and} \quad \boldsymbol{D}_{d_o} = \text{diag}\begin{bmatrix}45 & 45 & 45\end{bmatrix} .$$

Both position and velocity signals are filtered with a first order Butterworth low pass filter with cutoff frequencies of $10[Hz]$ and $5[Hz]$, respectively.

[1] $\boldsymbol{K}_{d_{p_{stiff}}}$, $\boldsymbol{K}_{d_{p_{compli}}}$ and $\boldsymbol{K}_{d_o}$ were tuned experimentally.

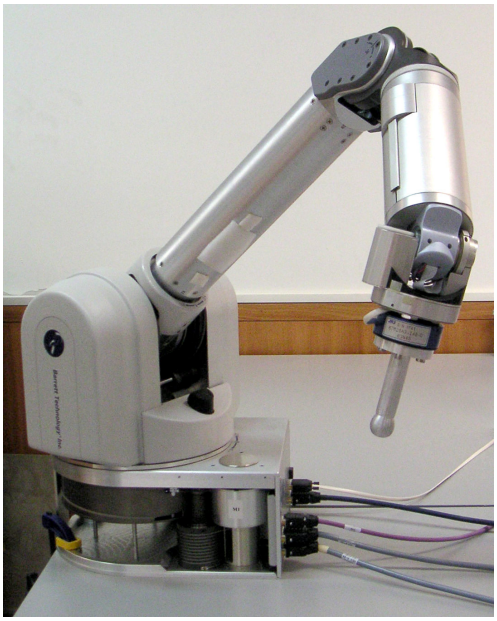


Figure 4.2— Slave manipulator with the tool used in Chapters 4 and 5 experiments. It is used a tool with a lower mass than the ultrasound probe to emphasize control issues of the last joint.

Control Issues

Controlling the last joint is particularly problematic (typically this joint performs roll motions). Due to its low inertia, friction effects are dominant (the last element in $\boldsymbol{B}(\boldsymbol{q})$, is two orders of magnitude smaller than the other wrist joints inertias). Since frictions are not taken into account in the dynamic model ($\boldsymbol{\tau}_f = 0$), to overcome friction effects, high control gains must be employed at orientation level, which leads the system close to instability. To cope with this issue, the tool mass taken into account in the dynamic model in this experiment was slightly increased ($+0.030[Kg]$). Further increasing the tool mass does not improve orientation performance. Fig. 4.2 shows the experimental setup reported in Chapters 4 and 5. It is used a tool with a lower mass than the ultrasound probe (see Section 2.2) to emphasize control issues.

Discussion

Fig. 4.3 shows position tracking performance, while orientation tracking results are shown in Fig. 4.4. With a compliant design, coupling effects from

Table 4.1: Absolute Mean Error (AME), Root Mean Square Error (RMSE) and Absolute Maximum Error (AMaxE) for Compliant vs. Stiff Impedance Design at Position Level. (Position Data).

	Compliant			Stiff		
$[m]$	AME	RMSE	AMaxE	AME	RMSE	AMaxE
x	0.0093	0.0107	0.0259	0.0056	0.0064	0.0144
y	0.0106	0.0125	0.0383	0.0069	0.0083	0.0207
z	0.0055	0.0076	0.0319	0.0026	0.0033	0.0112

Table 4.2: Absolute Mean Error (AME), Root Mean Square Error (RMSE) and Absolute Maximum Error (AMaxE) for for Compliant vs. Stiff Impedance Design at Position Level. (Orientation Data).

	Compliant			Stiff		
$[rad]$	AME	RMSE	AMaxE	AME	RMSE	AMaxE
Roll	0.1116	0.1321	0.3720	0.0814	0.1033	0.3920
Pitch	0.0775	0.0976	0.2944	0.0647	0.0828	0.2314
Yaw	0.1027	0.1241	0.2989	0.1079	0.1231	0.2686

the orientation control, model errors and dynamic unmodeled terms (such as frictions) act as perceptible disturbances in position control, leading to poor control performance, due to low control gains. Tables 4.1 and 4.2 present the Absolute Mean Error (AME), Root Mean Square Error (RMSE) and Absolute Maximum Error (AMAXE) for position and orientation tracking, respectively, showing that position tracking performance improves with stiffer design (as expected). Furthermore, the stiff design also leads to better orientation control performance, reducing AME, RMSE and AMAXE in all orientation dimensions, with the exception of roll AMAXE and AME yaw tracking.

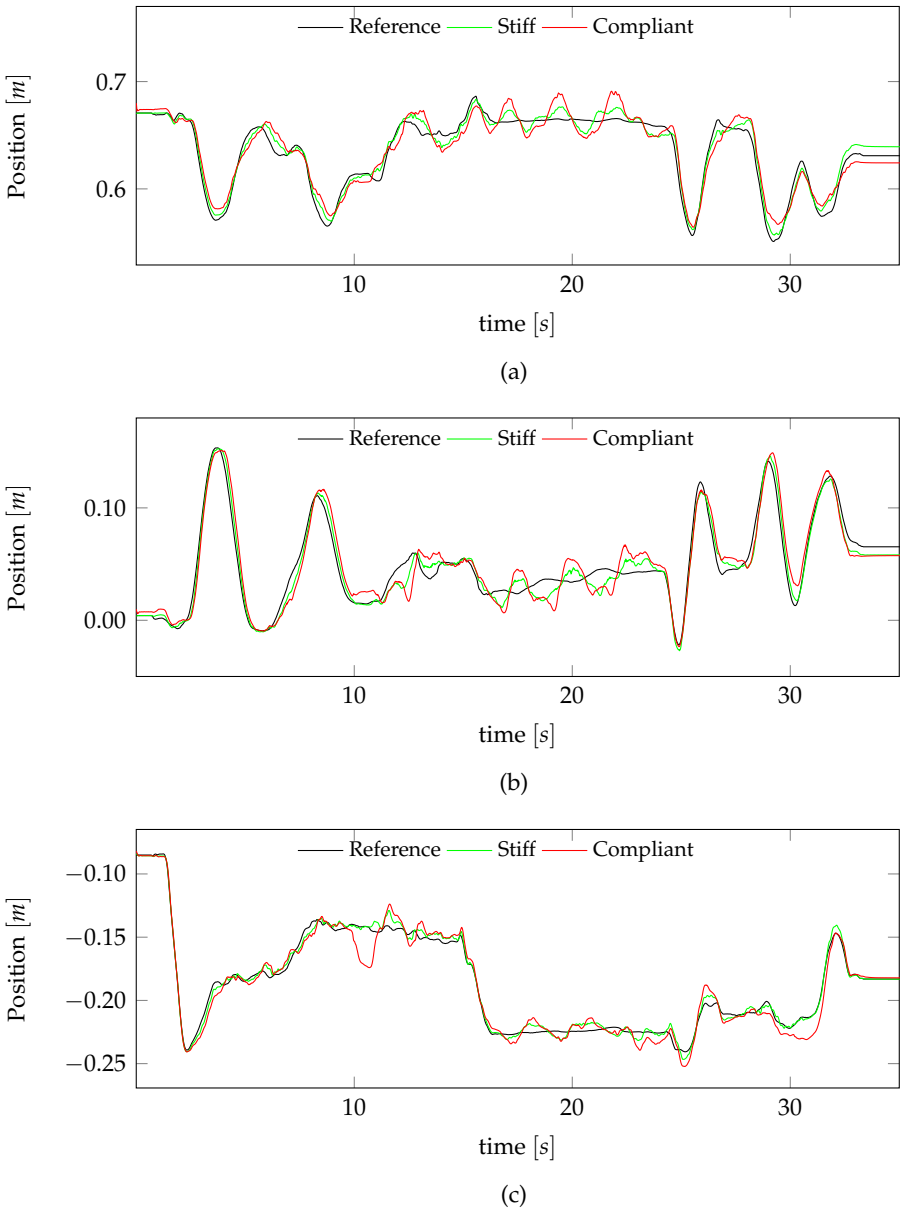
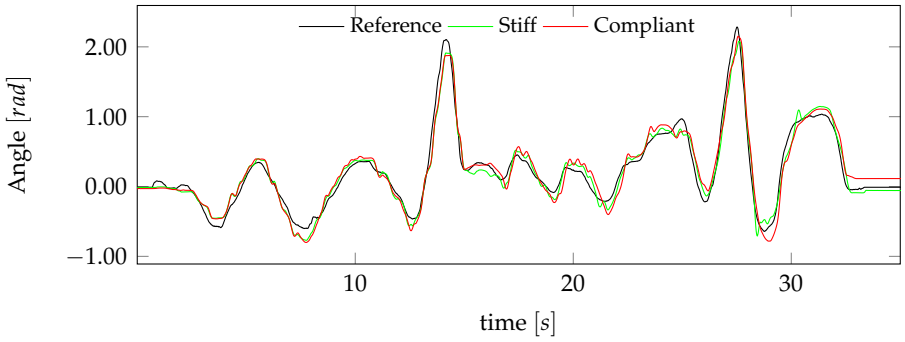
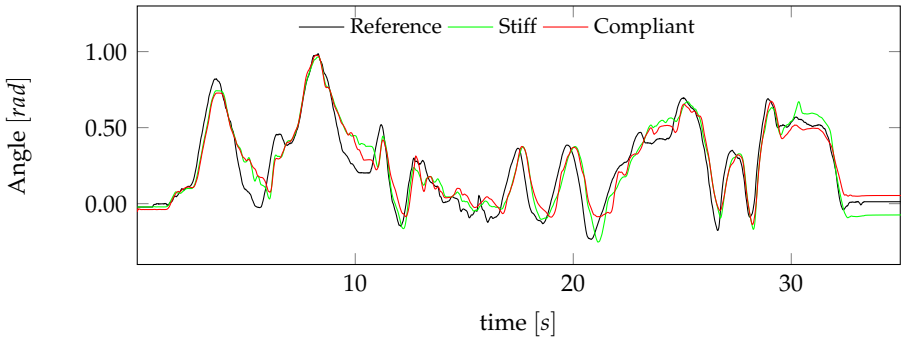


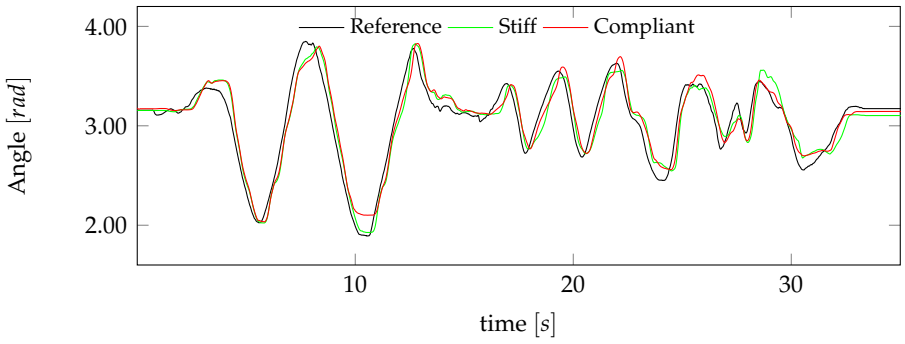
Figure 4.3— Cartesian position tracking performance. Impedance control with stiff and compliant impedance design. (a), (b) and (c) are respectively X, Y and Z tracking. Rotational impedance is designed for stiff behavior.



(a)



(b)



(c)

Figure 4.4— Orientation tracking performance. Impedance control with stiff and compliant impedance design. The orientation is represented by (a) Roll, (b) Pitch and (c) Yaw angles. Roll $\in [-\pi, \pi]$, Pitch $\in [-\pi, \pi]$ and Yaw angle $\in [0, 2\pi]$. Rotational impedance is designed for stiff behavior.

4.3 Joint Space Control with Task Space Pose References

Even with stiff position control and artificially increasing the tool mass, the orientation control performance is still not totally satisfactory, demanding a new control solution. Instead of controlling the end-effector pose in operational space, a strategy is devised to control each joint individually, still based on task space end-effector pose errors, but without explicitly solving the inverse kinematics problem. Task space errors are converted into joint velocity references, allowing to control each joint individually at velocity level, improving servo control performance.

4.3.1 Task Space Pose Control

Let $\dot{\boldsymbol{x}}_p \in \mathbb{R}^6$ be a task space velocity reference

$$\dot{\boldsymbol{x}}_p = \begin{bmatrix} \dot{\boldsymbol{p}}_p \\ \boldsymbol{\omega}_p \end{bmatrix} , \tag{4.22}$$

composed by linear and angular task space velocity references $\dot{\boldsymbol{p}}_p \in \mathbb{R}^3$ and $\boldsymbol{\omega}_p \in \mathbb{R}^3$, respectively. The task space velocity reference $\dot{\boldsymbol{x}}_p$ is computed from the task space error $\boldsymbol{x}_e \in \mathbb{R}^6$ as

$$\dot{\boldsymbol{x}}_p = \boldsymbol{I}\boldsymbol{x}_e , \tag{4.23}$$

with

$$\boldsymbol{x}_e = \begin{bmatrix} \Delta\boldsymbol{p}_{dr} \\ \boldsymbol{\epsilon}_{dr} \end{bmatrix} , \tag{4.24}$$

where $\boldsymbol{I} \in \mathbb{R}^{6\times 6}$ is the identity matrix and $\Delta\boldsymbol{p}_{dr} \in \mathbb{R}^3$ and $\boldsymbol{\epsilon}_{dr} \in \mathbb{R}^3$ are task space position and orientation errors (see (4.19)), respectively. The position error is given by

$$\Delta\boldsymbol{p}_{dr} = \boldsymbol{p}_d - \boldsymbol{p}_r , \tag{4.25}$$

where $\boldsymbol{p}_d \in \mathbb{R}^3$ and $\boldsymbol{p}_r \in \mathbb{R}^3$ are reference and robot end-effector task space positions, respectively. The velocity $\dot{\boldsymbol{x}}_p$ is mapped into the joint space through the differential kinematics equation

$$\dot{\boldsymbol{q}}_p = \boldsymbol{J}^{\dagger}(\boldsymbol{q})\dot{\boldsymbol{x}}_p , \tag{4.26}$$

where $\boldsymbol{J}^{\dagger}(\boldsymbol{q}) \in \mathcal{R}^{m\times n}$ is the Jacobian Moore-Penrose pseudo-inverse (see (3.17)) and $\dot{\boldsymbol{q}}_p \in \mathbb{R}^n$ is the joint space counterpart of $\dot{\boldsymbol{x}}_p$, used to compute inner joint velocity control references.

4.3.2 Joint Space Velocity Control

The joint velocity reference $\dot{\boldsymbol{q}}_d \in \mathbb{R}^n$ is computed by scaling $\dot{\boldsymbol{q}}_p$ by a proportional control gain $\boldsymbol{K}_1 \in \mathbb{R}^{n\times n}$,

$$\dot{\boldsymbol{q}}_d = \boldsymbol{K}_1 \dot{\boldsymbol{q}}_p \ , \tag{4.27}$$

where $\boldsymbol{K}_1$ is a diagonal positive definite matrix. A proportional gain $\boldsymbol{K}_d \in \mathbb{R}^{n\times n}$ acts on the joint velocity error

$$\Delta\dot{\boldsymbol{q}}_{dr} = \boldsymbol{K}_d \left(\dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}_r\right) \ . \tag{4.28}$$

The resolved acceleration $\boldsymbol{\alpha}$ (see (3.71)) is then given by

$$\boldsymbol{\alpha} = \boldsymbol{K}_d \left(\boldsymbol{K}_1 \dot{\boldsymbol{q}}_p - \dot{\boldsymbol{q}}_r\right) \ . \tag{4.29}$$

The torques $\boldsymbol{\tau}_a$ are given by

$$\boldsymbol{\tau}_a = \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{B}(\boldsymbol{q})\boldsymbol{\alpha} \ . \tag{4.30}$$

4.3.3 Control Design

The control gains are set neglecting the presence of $\boldsymbol{J}^{\dagger}(\boldsymbol{q})$ in the control law. Without taking into account $\boldsymbol{J}^{\dagger}(\boldsymbol{q})$ and assuming perfect linearization, the closed loop transfer function can be approximated by

$$\frac{Y(s)}{X(s)} = \frac{K_1 K_d}{s^2 + K_d s + K_1 K_d} \ .$$

To achieve a critically damped behavior ($\zeta = 1$), K_1 is computed as

$$K_1 = \frac{K_d}{2^2 \zeta^2} \ . \tag{4.31}$$

Fig. 4.6 shows the simplified control architecture. With the presence of $\boldsymbol{J}^{\dagger}(\boldsymbol{q})$ in the control loop, a decoupled control is no longer achievable. However in practice, the decoupled control architecture assumption provides good results, being used in control design.

4.3.4 Experiments

The joint controller is assessed by comparing its free space tracking performance against the performance of impedance control with stiff design, for the

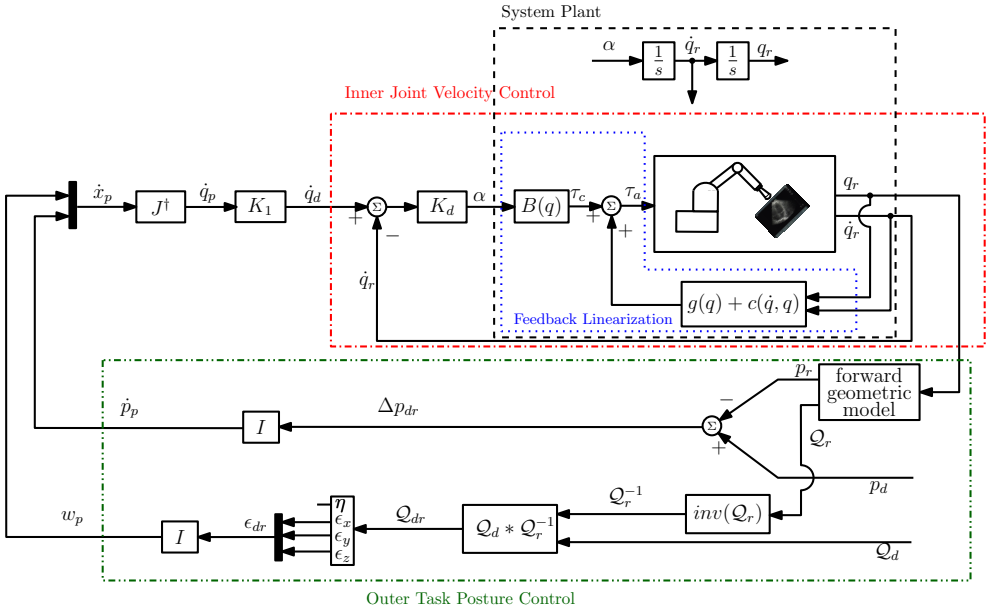


Figure 4.5— Joint velocity control architecture with task space pose references. The probe pose is controlled through a velocity control in the joint space, driven by task space orientation errors. Angular velocities from task space are mapped into the joint space following an inverse differential kinematics approach.

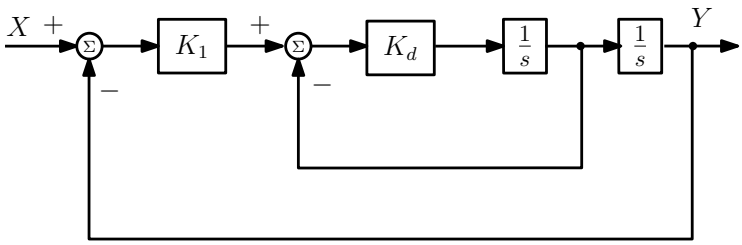


Figure 4.6— Simplified joint control architecture. The joint control gains are chosen based on this simplified model.

same motion of Section 4.2.3 experiments. Table 4.3 shows joint space control gains, which are individually tuned, allowing to overcome the effect of system disturbances. Higher gains are required to control wrist joints, specially the last one, where frictions are dominant.

Table 4.3: Joint Control Gains.

	K_d	K_1
q_1	42.5	10.625
q_2	45	11.25
q_3	45	11.25
q_4	50	12.5
q_5	75	18.75
q_6	75	18.75
q_7	110	27.5

Table 4.4: Absolute Mean Error (AME), Root Mean Square Error (RMSE) and Absolute Maximum Error (AMaxE) for Stiff Impedance Controller vs. Joint Space Controller (Position Data).

	Joint			Impedance		
$[m]$	AME	RMSE	AMaxE	AME	RMSE	AMaxE
x	0.0044	0.0054	0.0129	0.0056	0.0064	0.0144
y	0.0053	0.0068	0.0220	0.0069	0.0083	0.0207
z	0.0027	0.0042	0.0210	0.0026	0.0033	0.0112

Discussion

Figs. 4.7 and 4.8 show position and orientation tracking results, respectively. Table 4.4 summarizes position control results. Overall, position control performance of both controllers is similar, with the joint controller having marginally better results along X and Y directions (see Figs. 4.7a and 4.7b) and slightly worst results along Z (see Fig. 4.7c). Position tracking errors arise especially when performing orientation motions in a fixed position, highlighting coupling problems between orientation and position control (around $16[s] - [25s]$). Orientation control performance is greatly improved with explicit joint control. Table 4.5 summarizes orientation control results.

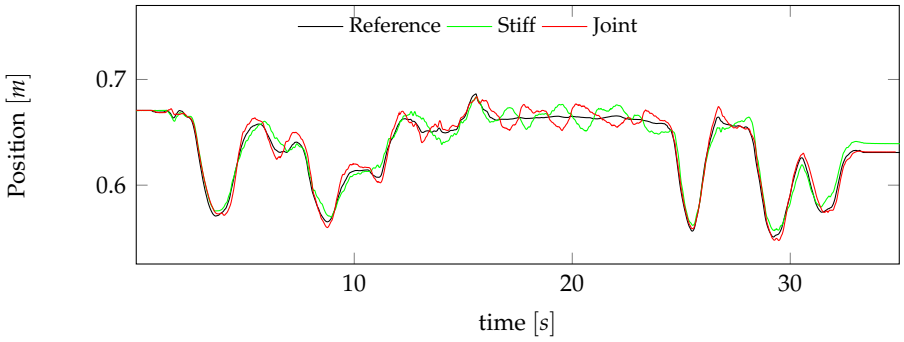
4.4 Conclusion

Two different control approaches are presented in this chapter. When performing an ultrasound examination, the physician controls interaction forces

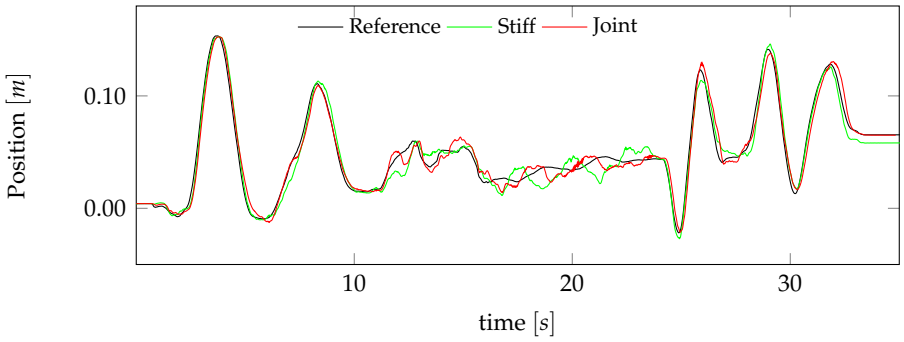
Table 4.5: Absolute Mean Error (AME), Root Mean Square Error (RMSE) and Absolute Maximum Error (AMaxE) for Stiff Impedance Controller vs. Joint Space Controller (Orientation Data).

	Joint			Impedance		
$[rad]$	AME	RMSE	AMaxE	AME	RMSE	AMaxE
Roll	0.0767	0.1093	0.3851	0.0814	0.1033	0.3920
Pitch	0.0468	0.0634	0.2071	0.0647	0.0828	0.2314
Yaw	0.0830	0.1042	0.2692	0.1079	0.1231	0.2686

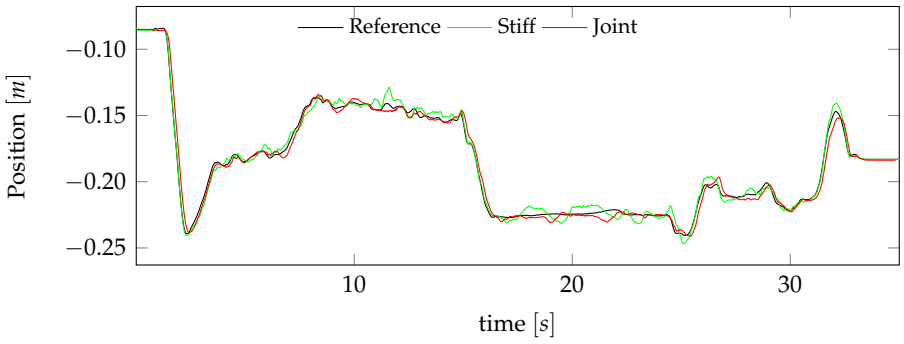
between probe and patient, enforcing stiff or compliant behaviors in accordance with the medical procedure. At orientation level, the probe is kept in a desired configuration, showing no compliant behavior. To address these requirements, a solution based on impedance control has been presented. The impedance control is designed to present a stiff behavior at orientation level and a compliant behavior at position level. Experimental results have shown that with this design, notorious coupling effects between position and orientation control exist, leading to undesired probe behaviors and poor motion tracking performance. Furthermore, even with stiff design for both position and orientation control, free space performance is still not satisfactory, specially at orientation level, which have led to follow a different control strategy, based on individual joint control. The idea behind joint space velocity control with task space pose references is to devise a control architecture that allows to design the control gains for each joint individually, reducing the detrimental effect of unmodeled friction terms in control performance. Experiments validate the joint control approach in free space, showing improved motion tracking performance, specially in orientation control. However, high control gains employed led to undesired stiff behavior also at position level. Therefore, a solution based on joint space velocity control with task space orientation references is going to be used as a building block for orientation control, while a different solution is required to control interaction forces.



(a)

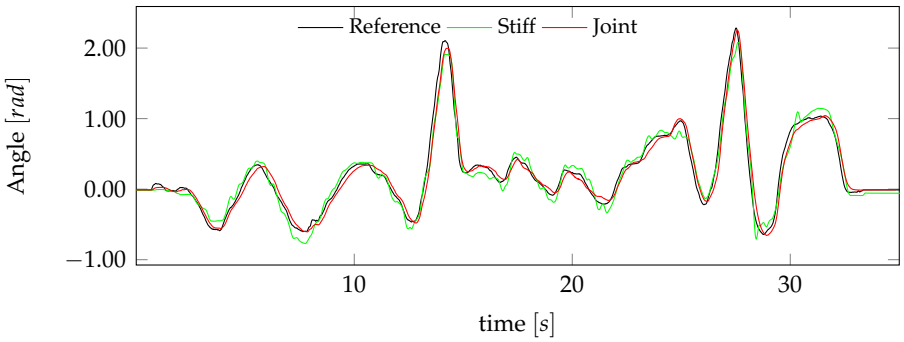


(b)

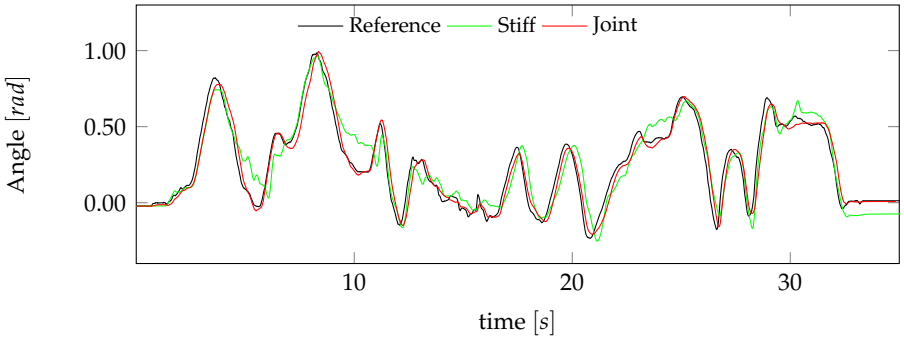


(c)

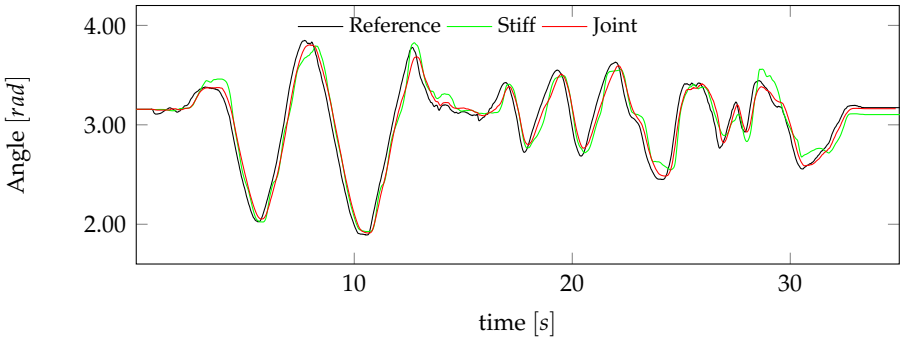
Figure 4.7— Cartesian position tracking performance. Impedance control with a stiff linear impedance design vs. joint space control with task space pose references. (a), (b) and (c) are respectively X, Y and Z tracking. Rotational impedance is designed to present a stiff behavior.



(a)



(b)



(c)

Figure 4.8— Orientation tracking performance. Impedance control with stiff impedance design vs. joint space control with task space pose references. The orientation is represented by (a) Roll, (b) Pitch and (c) Yaw angles. Roll $\in [-\pi, \pi]$, Pitch $\in [-\pi, \pi]$ and Yaw angle $\in [0, 2\pi]$. Rotational impedance is designed to present stiff behavior.

CHAPTER 5

HIERARCHICAL CONTROL ARCHITECTURE

For a robotic-assisted tele-echography system to be adopted by the medical community, free space and constrained probe motion must replicate the physician movements with high fidelity, while the dynamic behavior has to be adapted in accordance with the procedure nature, imposing constraints in interaction dynamics to improve both procedure execution and safety. In Chapter 4, it has been shown that an indirect force control approach falls short to deliver a satisfying interaction performance. Coupling effects between position and orientation control exist, leading to poor motion control performance when impedance control parameters are designed to present a compliant behavior. In this chapter, a new control architecture is proposed, featuring an explicit force control approach for compliant behavior at position level, while high performance motion control strategies are pursued to present stiff behaviors at orientation level. To avoid coupling issues between orientation and force control, a hierarchical control architecture is adopted. The hierarchical control architecture relies on:

- Explicit Cartesian force control driven by position errors as the high priority task. The force control establishes a suitable dynamic behavior between probe and patient, limiting the range of applied forces. In free space, position tracking is achieved without control switching.
- Probe orientation control in the null space, reducing coupling effects. The orientation control is driven by task space orientation errors con-

verted into joint velocity references, allowing to control each joint individually, boosting orientation tracking performance.

Both force and orientation are controlled through AOBs (Cortesão et al. (2006)). Furthermore, a strategy based on depth camera data is also developed to cope with inertial forces measurements. When coupling a load to a wrist force sensor, force measurements reflect not only contact forces but also inertial forces, which can become dominant in dynamic scenarios, deteriorating force control performance.

This chapter is organized as follows. In Section 5.1 a literature review on hierarchical priority control and inertial forces compensation is performed. Section 5.2 describes the hierarchical control architecture. Force and orientation control strategies are presented in Sections 5.3 and 5.4, respectively. Free space experiments are presented in Section 5.5 and Section 5.6 concludes the chapter.

5.1 Literature Review

This section provides literature review on hierarchical priority control and payload inertial force compensation.

5.1.1 Hierarchical Priority Control

Robots were initially designed to perform repetitive, dull and hazardous tasks in highly structured industrial scenarios. Therefore, robots were designed with minimal DOFs, adopting control architectures based on analytical inverse kinematics with high control gains to achieve high tracking accuracy at relatively low computational cost. With the increasing availability of computational power, roboticists focus started to shift from industrial applications in isolated environments, to applications where robots share their working space with humans (e.g., medical, assistive, and social robotics). To effectively perform in a human like environment (highly unstructured), where multiple and, sometimes, simultaneous tasks must be carried out, robots started to be designed with multiple redundant DOFs. Redundancy allows the execution of multiple tasks simultaneously in a hierarchical way, based on the Jacobian null-space (Siciliano (1990)). Under this general framework, redundancy can either be resolved at velocity (Liegeois (1977)), acceleration (Hsu et al. (1988))

or force level (Khatib (1987)). Hierarchical priority control is a well-established framework to perform multiple tasks simultaneously. By specifying multiple tasks, it is possible to assign different priorities to different tasks both at kinematic (Nakamura et al. (1987) and Siciliano and Slotine (1991)) and dynamic levels (Khatib et al. (2004), Sadeghian et al. (2011), and Sentis and Khatib (2006)). Compliant task space control presents a suitable approach to create an adequate motion of the end-effector in environments shared with humans. Nakanishi et al. (2008) present a study about different task space controllers for redundant manipulators. Null-space compliant control has been investigated by Sadeghian et al. (2014) while multi priority impedance control has been presented by Platt et al. (2010).

5.1.2 Payload Inertial Forces

When coupling a load in a wrist force sensor, force measurements reflect not only contact forces but also inertial forces, which can become dominant in dynamic scenarios, deteriorating force control performance. Garcia et al. (2006) present a sensor fusing approach (force, acceleration and position measurements) based on extended Kalman filter to be used to estimate non-contact forces, being this approach further extended by Kubus et al. (2008), where all inertial payload parameters are estimated online. On the other hand, Aghili (2010) proposes an impedance control scheme for manipulators carrying heavy payloads that overcomes non-contact forces without performing inertial forces estimation.

5.2 Hierarchical Priority Control

This section describes the hierarchical control architecture. Let $\boldsymbol{x}_p \in \mathbb{R}^m$ be a task to be performed. $\boldsymbol{J}_p(\boldsymbol{q}) \in \mathbb{R}^{m\times n}$ and $\boldsymbol{\Lambda}_p(\boldsymbol{q}) \in \mathbb{R}^{m\times m}$ are the Jacobian and inertia matrix associated with task $\boldsymbol{x}_p$, respectively. From Section 3.3.1, the torque vector computed by the control law to perform $\boldsymbol{x}_p$ is given by

$$\boldsymbol{\tau}_c = \boldsymbol{\tau}_p = \boldsymbol{J}_p(\boldsymbol{q})\boldsymbol{f}_p \,, \tag{5.1}$$

with $\boldsymbol{f}_p$ (similarly to (3.78)) as

$$\boldsymbol{f}_p = \boldsymbol{\Lambda}_p(\boldsymbol{q})\boldsymbol{f}^* - \boldsymbol{\Lambda}_p(\boldsymbol{q})\dot{\boldsymbol{J}}_p(\boldsymbol{q})\dot{\boldsymbol{q}} \,. \tag{5.2}$$

If $m < n$, the system is redundant when performing $\boldsymbol{x}_p$, and $\boldsymbol{\tau}_c$ is not unique. The generalized torque/force relationship is given by

$$\boldsymbol{\tau}_c = \boldsymbol{J}_p^T(\boldsymbol{q})\boldsymbol{f}_p + \boldsymbol{N}_p^T(\boldsymbol{q})\boldsymbol{\tau}_s \ . \tag{5.3}$$

This torque/force relationship allows the decomposition of $\boldsymbol{\tau}_c$ into two dynamically decoupled torque vectors

$$\boldsymbol{\tau}_c = \boldsymbol{\tau}_p + \boldsymbol{\tau}_n \ , \tag{5.4}$$

corresponding $\boldsymbol{\tau}_p$ to the primary task (see (3.75)) and

$$\boldsymbol{\tau}_n = \boldsymbol{N}_p^T(\boldsymbol{q})\boldsymbol{\tau}_s \tag{5.5}$$

to a secondary task that does not interfere with $\boldsymbol{\tau}_p$ and is performed in the null space ($\boldsymbol{N}(\boldsymbol{q})$) of $\boldsymbol{J}_p^T(\boldsymbol{q})$,

$$\boldsymbol{N}_p^T(\boldsymbol{q}) = \left[\boldsymbol{I}_n - \bar{\boldsymbol{J}}_p(\boldsymbol{q})\boldsymbol{J}_p(\boldsymbol{q})\right]^T = \left[\boldsymbol{I}_n - \boldsymbol{J}_p^T(\boldsymbol{q})\bar{\boldsymbol{J}}_p^T(\boldsymbol{q})\right] \ , \tag{5.6}$$

where $\boldsymbol{N}_p^T(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is the null space projector of task p and $\bar{\boldsymbol{J}}_p(\boldsymbol{q}) \in \mathbb{R}^{m\times n}$ is the dynamically consistent generalized inverse of $\boldsymbol{J}_p(\boldsymbol{q})$ (Khatib (1987)), given by

$$\bar{\boldsymbol{J}}_p(\boldsymbol{q}) = \boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{J}_p^T(\boldsymbol{q})\boldsymbol{\Lambda}_p(\boldsymbol{q}) \ . \tag{5.7}$$

Let $\boldsymbol{x}_s \in \mathbb{R}^l$ be a secondary task

$$\dot{\boldsymbol{x}}_s = \boldsymbol{J}_s(\boldsymbol{q})\dot{\boldsymbol{q}} \ , \tag{5.8}$$

where $\boldsymbol{J}_s(\boldsymbol{q}) \in \mathbb{R}^{l\times n}$ is its respective Jacobian matrix. The torque to perform the secondary task $\boldsymbol{\tau}_s \in \mathbb{R}^n$ can be defined similarly to $\boldsymbol{\tau}_p$ as

$$\boldsymbol{\tau}_s = \boldsymbol{J}_s^T(\boldsymbol{q})\boldsymbol{f}_s \ , \tag{5.9}$$

where $\boldsymbol{f}_s \in \mathbb{R}^l$ is the task space counterpart of $\boldsymbol{\tau}_s$. Substituting (5.9) in (5.3),

$$\begin{aligned} \boldsymbol{\tau}_c &= \boldsymbol{J}_p^T(\boldsymbol{q})\boldsymbol{f}_p + \boldsymbol{N}_p^T(\boldsymbol{q})\boldsymbol{J}_s^T(\boldsymbol{q})\boldsymbol{f}_s \\ &= \boldsymbol{J}_p^T(\boldsymbol{q})\boldsymbol{f}_p + \boldsymbol{J}_{s|p}^T(\boldsymbol{q})\boldsymbol{f}_s \ , \end{aligned} \tag{5.10}$$

where

$$\boldsymbol{J}_{s|p}^T(\boldsymbol{q}) = \left(\boldsymbol{J}_s(\boldsymbol{q})\boldsymbol{N}_p(\boldsymbol{q})\right)^T \tag{5.11}$$

appears as a new Jacobian matrix, combining the primary task null space operator with the secondary task Jacobian. Its range space is the instantaneous

space of the secondary task motion that is consistent with the primary task. Khatib et al. (2004) called it the *task-consistent posture Jacobian*. If $m + l < n$, the system is still redundant when performing both $\boldsymbol{x}_p$ and $\boldsymbol{x}_s$. The generalized $\boldsymbol{\tau}_c$ is then given by

$$\begin{aligned} \boldsymbol{\tau}_c &= \boldsymbol{J}_p^T(\boldsymbol{q})\boldsymbol{f}_p + \boldsymbol{J}_{s|p}^T(\boldsymbol{q})\boldsymbol{f}_s + \boldsymbol{N}_N^T(\boldsymbol{q})\boldsymbol{\tau}_0 \\ &= \boldsymbol{\tau}_p + \boldsymbol{\tau}_{s|p} + \boldsymbol{N}_N^T(\boldsymbol{q})\boldsymbol{\tau}_0 \,, \end{aligned} \tag{5.12}$$

where $\boldsymbol{\tau}_{s|p}$ is the torque to perform the secondary task constrained by the primary task, $\boldsymbol{\tau}_0 \in \mathbb{R}^n$ is a performance optimization torque and $\boldsymbol{N}_N(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is the projection matrix for the null space of all prioritized tasks, given by

$$\begin{aligned} \boldsymbol{N}_N(\boldsymbol{q}) &= \boldsymbol{N}_p(\boldsymbol{q})\boldsymbol{N}_{s|p}(\boldsymbol{q}) \\ &= \Big(\boldsymbol{I}_n - \bar{\boldsymbol{J}}_p(\boldsymbol{q})\boldsymbol{J}_p(\boldsymbol{q})\Big)\Big(\boldsymbol{I}_n - \bar{\boldsymbol{J}}_{s|p}(\boldsymbol{q})\boldsymbol{J}_{s|p}(\boldsymbol{q})\Big) \\ &= \boldsymbol{I}_n - \bar{\boldsymbol{J}}_p(\boldsymbol{q})\boldsymbol{J}_p(\boldsymbol{q}) - \bar{\boldsymbol{J}}_{s|p}(\boldsymbol{q})\boldsymbol{J}_{s|p}(\boldsymbol{q}) \,, \end{aligned} \tag{5.13}$$

where the property $\boldsymbol{J}_p(\boldsymbol{q})\boldsymbol{N}_p(\boldsymbol{q}) = \boldsymbol{0}$, with $\boldsymbol{0} \in \mathbb{R}^{n\times n}$ as the zero matrix, has been used.

5.2.1 Explicit Force Control with Null Space Orientation Control

In this section the control torque $\boldsymbol{\tau}_c$ is going to be computed for our architecture. Let $\boldsymbol{\tau}_p$ be the torque computed by the force controller and $\boldsymbol{\tau}_s$ be the torque computed by the joint space orientation controller, with $\boldsymbol{J}_p(\boldsymbol{q})$ and $\boldsymbol{J}_s(\boldsymbol{q})$ the position and the orientation Jacobian, respectively. Let also $\boldsymbol{f}^* \in \mathbb{R}^3$ be the control variable for the primary task and $\boldsymbol{\alpha} \in \mathbb{R}^n$ be the control variable for the secondary task. From (3.69) and (5.12), the manipulator dynamics with Coriolis, gravity and external force pre-compensation in the joint space (see (3.68)) is given by

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} = \boldsymbol{\tau}_p + \boldsymbol{\tau}_{s|p} + \boldsymbol{N}_N^T(\boldsymbol{q})\boldsymbol{\tau}_0 \,. \tag{5.14}$$

Left multiplying both sides of (5.14) by $\boldsymbol{J}_s^T(\boldsymbol{q})\bar{\boldsymbol{J}}_s^T(\boldsymbol{q})$ and setting $\boldsymbol{\tau}_{s|p}$ as

$$\boldsymbol{\tau}_{s|p} = \boldsymbol{N}_p^T(\boldsymbol{q})\boldsymbol{\tau}_s = \boldsymbol{N}_p^T(\boldsymbol{q})\boldsymbol{B}(\boldsymbol{q})\boldsymbol{\alpha} \,, \tag{5.15}$$

leads to

$$\begin{aligned} \boldsymbol{J}_s^T(\boldsymbol{q})\bar{\boldsymbol{J}}_s^T(\boldsymbol{q})\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} = \; & \boldsymbol{J}_s^T(\boldsymbol{q})\bar{\boldsymbol{J}}_s^T(\boldsymbol{q})\boldsymbol{\tau}_p + \\ & + \boldsymbol{J}_s^T(\boldsymbol{q})\bar{\boldsymbol{J}}_s^T(\boldsymbol{q})\boldsymbol{N}_p^T(\boldsymbol{q})\boldsymbol{B}(\boldsymbol{q})\boldsymbol{\alpha} + \\ & + \underbrace{\boldsymbol{J}_s^T(\boldsymbol{q})\bar{\boldsymbol{J}}_s^T(\boldsymbol{q})\boldsymbol{N}_p^T(\boldsymbol{q})\boldsymbol{N}_{s|p}^T(\boldsymbol{q})}_{=\boldsymbol{0}}\boldsymbol{\tau}_0 \,, \end{aligned} \tag{5.16}$$

where $\bar{\boldsymbol{J}}_s(\boldsymbol{q}) \in \mathbb{R}^{3\times n}$ is the dynamically consistent generalized inverse of $\boldsymbol{J}_s(\boldsymbol{q}) \in \mathbb{R}^{3\times n}$, computed similarly to (5.7). $\boldsymbol{J}_s^T(\boldsymbol{q})\bar{\boldsymbol{J}}_s^T(\boldsymbol{q})\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}}$ are the control torques acting on the secondary task space, composed by the orientation control torques (second term of (5.16)) and the coupling torques from the primary task (first term). The third term vanish due to the presence of $\boldsymbol{J}_{s|p}(\boldsymbol{q})\boldsymbol{N}_{s|p}(\boldsymbol{q}) = \boldsymbol{0}$,

$$\begin{aligned}\boldsymbol{J}_s^T(\boldsymbol{q})\bar{\boldsymbol{J}}_s^T(\boldsymbol{q})\boldsymbol{N}_p^T(\boldsymbol{q})\boldsymbol{N}_{s|p}^T(\boldsymbol{q})\boldsymbol{\tau}_0 = \\ = \boldsymbol{B}(\boldsymbol{q})\bar{\boldsymbol{J}}_s(\boldsymbol{q})\underbrace{\boldsymbol{J}_s(\boldsymbol{q})\boldsymbol{N}_p(\boldsymbol{q})}_{=\boldsymbol{J}_{s|p}(\boldsymbol{q})}\boldsymbol{N}_{s|p}(\boldsymbol{q})\ddot{\boldsymbol{q}}_0 \,,\end{aligned} \tag{5.17}$$

where the commutation properties have been used:

$$\boldsymbol{N}_p^T(\boldsymbol{q})\boldsymbol{B}(\boldsymbol{q}) = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{N}_p(\boldsymbol{q}) \,, \tag{5.18}$$

$$\boldsymbol{J}_s^T(\boldsymbol{q})\bar{\boldsymbol{J}}_s^T(\boldsymbol{q})\boldsymbol{B}(\boldsymbol{q}) = \boldsymbol{B}(\boldsymbol{q})\bar{\boldsymbol{J}}_s(\boldsymbol{q})\boldsymbol{J}_s(\boldsymbol{q}) \,. \tag{5.19}$$

Projecting (5.16) through $\boldsymbol{N}_p^T(\boldsymbol{q})$, the torques acting on the secondary task space without affecting the primary task execution are given by

$$\begin{aligned}\boldsymbol{N}_p^T(\boldsymbol{q})\boldsymbol{J}_s^T(\boldsymbol{q})\bar{\boldsymbol{J}}_s^T(\boldsymbol{q})\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} = \; & \boldsymbol{N}_p^T(\boldsymbol{q})\boldsymbol{J}_s^T(\boldsymbol{q})\bar{\boldsymbol{J}}_s^T(\boldsymbol{q})\boldsymbol{\tau}_p + \\ & + \boldsymbol{N}_p^T(\boldsymbol{q})\boldsymbol{J}_s^T(\boldsymbol{q})\bar{\boldsymbol{J}}_s^T(\boldsymbol{q})\boldsymbol{N}_p^T(\boldsymbol{q})\boldsymbol{B}(\boldsymbol{q})\boldsymbol{\alpha} \,.\end{aligned} \tag{5.20}$$

Left multiplying both sides by $\boldsymbol{B}^{-1}(\boldsymbol{q})$ and using the commutation properties

$$\boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{N}_p^T(\boldsymbol{q}) = \boldsymbol{N}_p(\boldsymbol{q})\boldsymbol{B}^{-1}(\boldsymbol{q}) \,, \tag{5.21}$$

$$\boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{J}_s^T(\boldsymbol{q})\bar{\boldsymbol{J}}_s^T(\boldsymbol{q}) \; = \bar{\boldsymbol{J}}_s(\boldsymbol{q})\boldsymbol{J}_s(\boldsymbol{q})\boldsymbol{B}^{-1}(\boldsymbol{q}) \,, \tag{5.22}$$

the corresponding secondary task joint acceleration is given by

$$\begin{aligned}\boldsymbol{N}_p(\boldsymbol{q})\bar{\boldsymbol{J}}_s(\boldsymbol{q})\boldsymbol{J}_s(\boldsymbol{q})\ddot{\boldsymbol{q}} = \boldsymbol{N}_p(\boldsymbol{q})\bar{\boldsymbol{J}}_s(\boldsymbol{q})\boldsymbol{J}_s(\boldsymbol{q})\boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{\tau}_p + \\ + \boldsymbol{N}_p(\boldsymbol{q})\bar{\boldsymbol{J}}_s(\boldsymbol{q})\boldsymbol{J}_{s|p}(\boldsymbol{q})\boldsymbol{\alpha} \,.\end{aligned} \tag{5.23}$$

Setting the control variable $\boldsymbol{\alpha}$ as

$$\boldsymbol{\alpha} = \bar{\boldsymbol{J}}_{s|p}(\boldsymbol{q})\boldsymbol{J}_s(\boldsymbol{q})\left(\boldsymbol{\nu} - \boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{\tau}_p\right) \tag{5.24}$$

and replacing (5.24) in (5.23), the secondary task closed loop dynamics becomes

$$\boldsymbol{N}_p(\boldsymbol{q})\bar{\boldsymbol{J}}_s(\boldsymbol{q})\boldsymbol{J}_s(\boldsymbol{q})\left(\ddot{\boldsymbol{q}} - \boldsymbol{\nu}\right) = \boldsymbol{0} \,. \tag{5.25}$$

where $\boldsymbol{\nu} \in \mathbb{R}^n$ arises as a new control variable. A linear and decoupled joint acceleration in the primary task null space shows up, performing the secondary task without inducing any motion in the primary task space.

Plugging (5.24) in (5.15) and replacing $\bar{\boldsymbol{J}}_{s|p}(\boldsymbol{q})$ by

$$\bar{\boldsymbol{J}}_{s|p}(\boldsymbol{q}) = \boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{J}_{s|p}^T(\boldsymbol{q})\boldsymbol{\Lambda}_{s|p}(\boldsymbol{q}) \ , \tag{5.26}$$

$\boldsymbol{\tau}_{s|p}$ becomes

$$\boldsymbol{\tau}_{s|p} = \boldsymbol{J}_{s|p}^T(\boldsymbol{q})\boldsymbol{\Lambda}_{s|p}(\boldsymbol{q})\boldsymbol{J}_s(\boldsymbol{q}) \left(\boldsymbol{\nu} - \boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{\tau}_p\right) \ , \tag{5.27}$$

where the null space idempotence property $\boldsymbol{N}_p^T(\boldsymbol{q}) = \left(\boldsymbol{N}_p^T(\boldsymbol{q})\right)^2$ has been used and $\boldsymbol{\Lambda}_{s|p}(\boldsymbol{q})$ is a prioritized inertia matrix given by (Khatib et al. (2004))

$$\boldsymbol{\Lambda}_{s|p}(\boldsymbol{q}) = \left(\boldsymbol{J}_s(\boldsymbol{q})\boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{J}_{s|p}^T(\boldsymbol{q})\right)^{-1} \ . \tag{5.28}$$

Substituting (5.27) and (5.2) in (5.12), the control torque to perform both tasks is given by

$$\begin{aligned} \boldsymbol{\tau}_c = \ & \boldsymbol{J}_p^T(\boldsymbol{q}) \left[\boldsymbol{\Lambda}_p(\boldsymbol{q})\boldsymbol{f}^* - \boldsymbol{\Lambda}_p(\boldsymbol{q})\dot{\boldsymbol{J}}_p(\boldsymbol{q})\dot{\boldsymbol{q}}\right] + \\ & + \boldsymbol{J}_{s|p}^T(\boldsymbol{q})\boldsymbol{\Lambda}_{s|p}(\boldsymbol{q})\boldsymbol{J}_s(\boldsymbol{q}) \left[\boldsymbol{\nu} - \boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{\tau}_p\right] + \\ & + \boldsymbol{N}_T^N(\boldsymbol{q})\boldsymbol{\tau}_0 \ . \end{aligned} \tag{5.29}$$

In the above formulation, algorithmic singularities arise when the primary and the secondary task are in conflict, causing $\boldsymbol{J}_{s|p}^T(\boldsymbol{q})$ to drop rank (Sentis and Khatib (2005)). To avoid issues with such singularities, $\boldsymbol{\alpha}$ is computed by

$$\boldsymbol{\alpha} = \bar{\boldsymbol{J}}_s(\boldsymbol{q})\boldsymbol{J}_s(\boldsymbol{q}) \left(\boldsymbol{\nu} - \boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{\tau}_p\right) \ , \tag{5.30}$$

where $\bar{\boldsymbol{J}}_s(\boldsymbol{q})$ has been used instead of $\bar{\boldsymbol{J}}_{s|p}(\boldsymbol{q})$. $\boldsymbol{J}_s(\boldsymbol{q})$ is the orientation Jacobian computed without any constraints from the force controller, with all DOFs available. Due to the manipulator structure, $\boldsymbol{J}_s(\boldsymbol{q})$ only loses rank at the external boundary of its reachable workspace, having no ill-condition issues near internal singular configurations, when both tasks are in conflict. Following this approach and using (5.18), (5.29) becomes

$$\begin{aligned} \boldsymbol{\tau}_c = & \boldsymbol{J}_p^T(\boldsymbol{q}) \left[\boldsymbol{\Lambda}_p(\boldsymbol{q})\boldsymbol{f}^* - \boldsymbol{\Lambda}_p(\boldsymbol{q})\dot{\boldsymbol{J}}_p(\boldsymbol{q})\dot{\boldsymbol{q}}\right] + \\ & + \boldsymbol{J}_{s|p}^T(\boldsymbol{q})\bar{\boldsymbol{J}}_s^T(\boldsymbol{q}) \left[\boldsymbol{B}(\boldsymbol{q})\boldsymbol{\nu} - \boldsymbol{\tau}_p\right] + \\ & + \boldsymbol{N}_T^N(\boldsymbol{q})\boldsymbol{\tau}_0 \ . \end{aligned} \tag{5.31}$$

The closed loop system in (5.25) is no longer achievable. As a consequence, tracking errors might arise in the orientation control, since the forces given by $\bar{\boldsymbol{J}}_s^T(\boldsymbol{q})\left[\boldsymbol{B}(\boldsymbol{q})\boldsymbol{\nu} - \boldsymbol{\tau}_p\right]$ might have components along the primary task space, which are canceled out by $\boldsymbol{N}_p^T(\boldsymbol{q})$. However, as it is going to be shown in Sections 5.5 and 7.4, this approach together with the orientation control strategy provides good orientation tracking results in the presence or absence of contact constraints.

5.2.2 Posture Optimization

The manipulator posture plays a key role in performance optimization. In our setup, performance is optimized by keeping redundant joints close to the middle of their range through gradient minimization, projecting $\boldsymbol{\tau}_0$ in the null space of all prioritized tasks (see (5.13)),

$$\boldsymbol{\tau}_0 = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{K}_N \left(\frac{\partial v(\boldsymbol{q})}{\partial \boldsymbol{q}}\right)^T . \tag{5.32}$$

$\boldsymbol{K}_N \in \mathbb{R}^{n\times n}$ is a diagonal gain matrix and $v(\boldsymbol{q})$ is an objective function given by

$$v(\boldsymbol{q}) = -\frac{1}{2}\sum_{i=1}^{n}\left(\frac{q_i - \bar{q}_i}{q_{i_M} - q_{i_m}}\right)^2 , \tag{5.33}$$

where $\bar{q}_i$, q_{i_M} and q_{i_m} are, respectively, the i^{th} joint middle, maximum and minimum range value.

Projecting $\boldsymbol{\tau}_0$ through (5.13) in (5.31), the total computed torque is then given by replacing (5.31) in (3.68). Fig. 5.1 shows the proposed control architecture scheme, where force and orientation control are the primary and secondary tasks, respectively, and posture optimization is performed in the null space of all prioritized tasks.

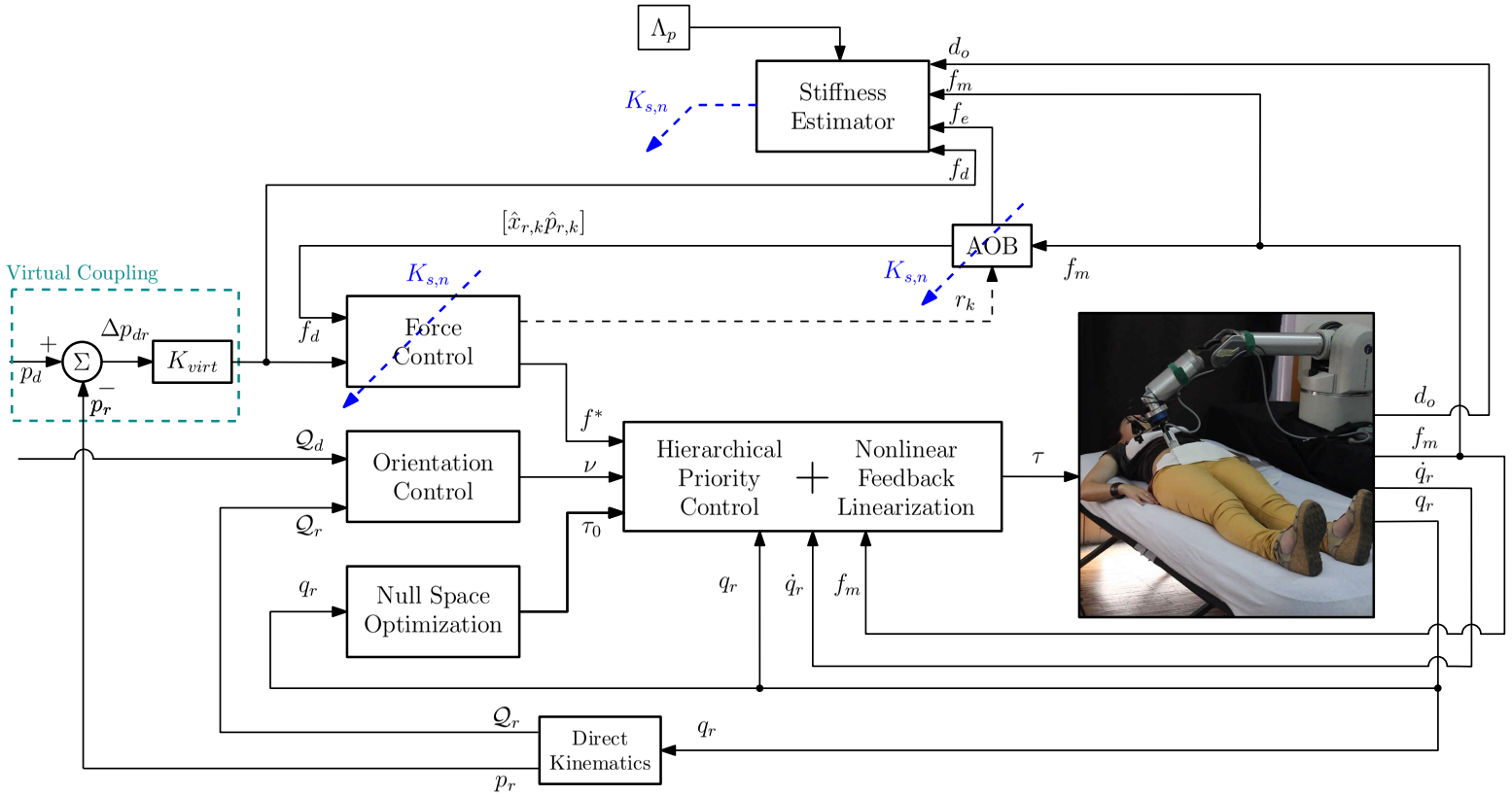


Figure 5.1— Hierarchical Control Architecture for Robotic-Assisted Tele-Echography. A two task hierarchy architecture with posture optimization is proposed, where Cartesian force control is the primary task, orientation control the secondary task and posture optimization is performed in the null space of all prioritized tasks. An explicit force control architecture controls the probe positioning and interaction dynamics through a model-reference adaptive control (AOB) in the task space. The control is adapted online based on the environment estimation ($K_{s,n}$). The environment is modeled as a linear spring and is estimated online using contact distance from a 3D camera (d_o), force data (f_d, f_{fs} and f_e) and the robot inertial properties referred to the end-effector (Λ_p). Force references are generated from position errors (Δp_{dr}) through virtual coupling (K_{virt}). The probe orientation is controlled by a joint space velocity control driven by task space orientation errors. Posture is optimized by keeping redundant joints close to the middle of their range value.

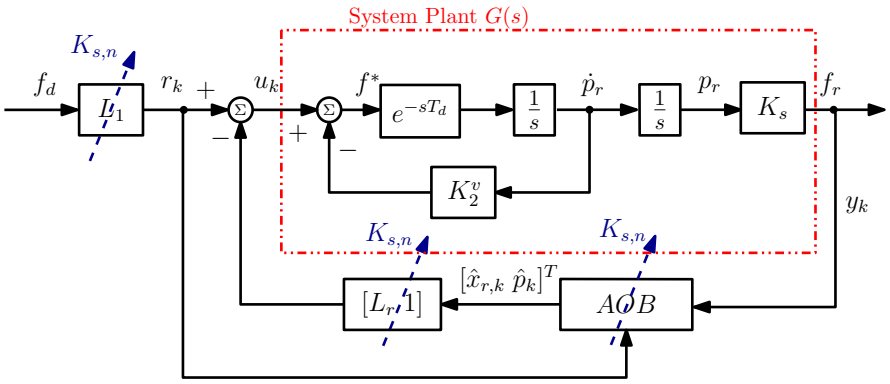


Figure 5.2— Force control with an AOB in the loop. Each Cartesian force dimension is controlled by an AOB. The dashed blue arrows show the control variables that change with the stiffness adaptation $K_{s,n}$.

5.3 Force Control

This section discusses the force control architecture. Following the approach presented in Cortesão et al. (2006), each Cartesian force dimension is controlled through state feedback from an AOB. $\boldsymbol{f}_d \in \mathbb{R}^3$ is the force reference, computed from the position error through virtual coupling as

$$\boldsymbol{f}_d = \boldsymbol{K}_{virt} \left(\boldsymbol{p}_d - \boldsymbol{p}_r \right) \ , \tag{5.34}$$

where $\boldsymbol{p}_d \in \mathbb{R}^3$ and $\boldsymbol{p}_r \in \mathbb{R}^3$ are, respectively, desired and end-effector Cartesian positions and $\boldsymbol{K}_{virt} \in \mathbb{R}^{3\times3}$ is a virtual coupling matrix (see Section 7.1). In the sequel, the control analysis is done for each Cartesian force dimension individually. Fig. 5.2 shows the force control scheme with the AOB in the loop.

5.3.1 System Plant

Equation (3.79) represents the dynamics of a unitary mass for each Cartesian dimension. Modeling the environment as a linear spring K_s[1], with nominal stiffness $K_{s,n}$, and accounting with system delay T_d due to signal processing,

[1] The system stiffness is given by $\frac{1}{K_{sys}} = \frac{1}{K_s} + \frac{1}{K_w}$, where K_s is the environment stiffness and K_w is the stiffness of the force sensor. It is assumed through this work that $K_w \gg K_s$, leading to $K_{sys} \approx K_s$.

the force controlled robot plant is given by

$$G(s) = \frac{K_s e^{-sT_d}}{s(s + K_2 e^{-sT_d})} , \tag{5.35}$$

where K_2 is the system damping, with nominal value $K_{2,n}$, which includes the feedback gain K_2^v (De Schutter et al. (1998)) and the robot structural damping K_2^s

$$K_2 = K_2^v + K_2^s . \tag{5.36}$$

Assuming a small time system delay, the force controlled robot can be approximated by

$$G(s) \approx \frac{K_s e^{-sT_d}}{s(s + K_2)} . \tag{5.37}$$

The equivalent temporal representation is given by

$$\ddot{y}(t) + K_2 \dot{y}(t) = K_s u(t - T_d) , \tag{5.38}$$

where $u(t)$ is an acceleration (plant input) and $y(t)$ is the force exerted (f_m) by the robot and measured at the end-effector (plant output). Defining $x_{r_1}(t) = y(t)$ and $x_{r_2}(t) = \dot{y}(t)$, respectively the output force and its time derivative, (5.38) can be written in a state-space representation as

$$\begin{cases} \dot{\boldsymbol{x}}_r(t) = \boldsymbol{A}_{ss} \boldsymbol{x}_r(t) + \boldsymbol{B}_{ss} \boldsymbol{u}(t - T_d) \\ y(t) = \boldsymbol{C}_{ss} \boldsymbol{x}_r(t) \end{cases} , \tag{5.39}$$

with

$$\boldsymbol{x}_r(t) = \begin{bmatrix} x_{r_1} \\ x_{r_2} \end{bmatrix} \quad \boldsymbol{A}_{ss} = \begin{bmatrix} 0 & 1 \\ 0 & -K_2 \end{bmatrix} \quad \boldsymbol{B}_{ss} = \begin{bmatrix} 0 \\ K_s \end{bmatrix} \text{ and } \boldsymbol{C}_{ss} = \begin{bmatrix} 1 & 0 \end{bmatrix} ,$$

where $\boldsymbol{A}_{ss} \in \mathbb{R}^{2\times 2}$, $\boldsymbol{B}_{ss} \in \mathbb{R}^2$ and $\boldsymbol{C}_{ss} \in \mathbb{R}^{1\times 2}$ are, respectively, state, input and output matrices.

5.3.2 Discrete System

Discretizing (5.39) with sampling time h (Aström and Wittenmark (1997)), the equivalent discrete time system becomes

$$\begin{cases} \boldsymbol{x}_{r,k} = \boldsymbol{\Phi}_r \boldsymbol{x}_{r,k-1} + \boldsymbol{\Gamma}_r u_{k-1} \\ y_k = \boldsymbol{C}_r \boldsymbol{x}_{r,k} \end{cases} , \tag{5.40}$$

where

$$\boldsymbol{x}_{r,k} = \begin{bmatrix} \boldsymbol{x}_k \\ u_{k-d} \\ \dots \\ u_{k-2} \\ u_{k-1} \end{bmatrix} , \quad \boldsymbol{\Gamma}_r = \begin{bmatrix} 0 \\ \dots \\ 0 \\ 1 \end{bmatrix} , \quad \boldsymbol{C}_r = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} . \tag{5.41}$$

$\boldsymbol{x}_k$ is the state vector of the core states (the discrete representation of $\boldsymbol{x}_r(t)$), being the remaining d states in $\boldsymbol{x}_{r,k}$ due to T_d. The term d is computed by

$$\begin{gathered} T_d = (d-1)h + \tau' \\ 0 < \tau' \leq h \end{gathered} . \tag{5.42}$$

$\boldsymbol{\Phi}_r$, $\boldsymbol{\Gamma}_r$ and $\boldsymbol{C}_r$ are, respectively, state transition, command and measurement state-space matrices in discrete time. $\boldsymbol{\Phi}_r$ is given by

$$\boldsymbol{\Phi}_r = \begin{bmatrix} \boldsymbol{\Phi}_1 & \boldsymbol{\Gamma}_1 & \boldsymbol{\Gamma}_0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} , \tag{5.43}$$

with $\boldsymbol{\Phi}_1$, $\boldsymbol{\Gamma}_1$ and $\boldsymbol{\Gamma}_0$ as

$$\boldsymbol{\Phi}_1 = e^{\boldsymbol{A}_{ss}h} , \tag{5.44}$$

$$\boldsymbol{\Gamma}_1 = e^{\boldsymbol{A}_{ss}(h-\tau')} \int_0^{\tau'} e^{\boldsymbol{A}_{ss}\lambda} \mathrm{d}\lambda \boldsymbol{B}_{ss} , \tag{5.45}$$

$$\boldsymbol{\Gamma}_0 = \int_0^{h-\tau'} e^{\boldsymbol{A}_{ss}\lambda} \mathrm{d}\lambda \boldsymbol{B}_{ss} , \tag{5.46}$$

where $e^{\boldsymbol{A}_{ss}t} = \boldsymbol{\phi}(t)$ is the continuous time state transition matrix. In the Laplace domain, $\boldsymbol{\phi}(t)$ is given by (Ogata (2010))

$$\boldsymbol{\Phi}(s) = (s\boldsymbol{I} - \boldsymbol{A}_{ss})^{-1} = \begin{bmatrix} s & -1 \\ 0 & s+K_2 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{s} & \frac{1}{s(s+K_2)} \\ 0 & \frac{1}{s+K_2} \end{bmatrix} . \tag{5.47}$$

Applying the inverse Laplace transformation to $\boldsymbol{\Phi}(s)$, $\boldsymbol{\phi}(t)$ arises as

$$\boldsymbol{\phi}(t) = \begin{bmatrix} 1 & \frac{1-e^{-K_2 t}}{K_2} \\ 0 & e^{-K_2 t} \end{bmatrix} . \tag{5.48}$$

Knowing $\boldsymbol{\phi}(t)$, the computation of $\boldsymbol{\Phi}_1$, $\boldsymbol{\Gamma}_1$ and $\boldsymbol{\Gamma}_0$ is straightforward. $\boldsymbol{\Phi}_1$ is given by

$$\boldsymbol{\Phi}_1 = \begin{bmatrix} 1 & \frac{1-e^{-K_2 h}}{K_2} \\ 0 & e^{-K_2 h} \end{bmatrix} . \tag{5.49}$$

For a system delay of $T_d = h$ (the delay present in our system), $d = 1$ and $\tau' = h$ (see (5.42)), which leads to

$$\boldsymbol{\Gamma}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{5.50}$$

and

$$\boldsymbol{\Gamma}_1 = \frac{K_s}{K_2} \begin{bmatrix} h + \frac{e^{-K_2 h}-1}{K_2} \\ \left(1 - e^{-K_2 h}\right) \end{bmatrix} . \tag{5.51}$$

As a result, $\boldsymbol{\Phi}_r$, $\boldsymbol{\Gamma}_r$ and $\boldsymbol{C}_r$ are given by

$$\boldsymbol{\Phi}_r = \begin{bmatrix} 1 & \frac{1-e^{-K_2 h}}{K_2} & \frac{K_s}{K_2}\left(h + \frac{e^{-K_2 h}-1}{K_2}\right) \\ 0 & e^{-K_2 h} & \frac{K_s}{K_2}\left(1 - e^{-K_2 h}\right) \\ 0 & 0 & 0 \end{bmatrix} , \boldsymbol{\Gamma}_r = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} , \quad \boldsymbol{C}_r = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} .$$

5.3.3 Active Observers

An AOB[2] is a state space observer that reformulates the Kalman filter framework in order to accomplish model-reference adaptive control (Cortesão (2003, 2007)). Resorting to state feedback control to achieve a desired closed loop behavior, the AOB performs state estimation based on the desired closed loop behavior, using Kalman filter stochastic design to reflect the uncertainties of system model and output measurements. An extra state (active state) p_k, is employed to estimate an equivalent disturbance referred to the system input due to unmodeled terms, such as higher order dynamics, parameter mismatches, unknown disturbances and noise, performing compensation actions. To be able to track functions with unknown dynamics, p_k is described by a stochastic equation

$$p_k - p_{k-1} = \xi_{p_k} , \tag{5.52}$$

[2]This section describes the AOB algorithm, its design and implementation in a force control architecture for a medical robotic application. It is out of the scope of this work to show AOB stability and robustness capabilities, since it has already been performed elsewhere. A thoroughly work detailing the AOB can be found in Cortesão (2003).

in which ξ_{p_k} is a zero-mean Gaussian random variable, meaning that the first-order evolution of p_k is randomly distributed (Cortesão (2007)). Introducing p_k and $\hat{p}_k$[3] in (5.40), and controlling the system by state feedback from an observer, the augmented state space representation is given by

$$\begin{bmatrix} \boldsymbol{x}_{r,k} \\ p_k \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_r & \boldsymbol{\Gamma}_r \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{r,k-1} \\ p_{k-1} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Gamma}_r \\ 0 \end{bmatrix} u'_{k-1} + \boldsymbol{\xi_k} \tag{5.53}$$

$$y_k = \boldsymbol{C}_a \begin{bmatrix} \boldsymbol{x}_k \\ p_k \end{bmatrix} + \kappa_k \tag{5.54}$$

where

$$u'_{k-1} = r_{k-1} - \begin{bmatrix} \boldsymbol{L}_r & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_{r,k-1} \\ \hat{p}_{k-1} \end{bmatrix} , \tag{5.55}$$

and

$$\boldsymbol{C}_a = \begin{bmatrix} \boldsymbol{C}_r & 0 \end{bmatrix} . \tag{5.56}$$

$\boldsymbol{r}_{k-1}$ is the input reference and $\boldsymbol{L}_r$ is the state feedback gain matrix used to set the closed loop system response. The stochastic inputs

$$\boldsymbol{\xi_k} = \begin{bmatrix} \xi_{x_{r,k}} \\ \xi_{p_k} \end{bmatrix} , \tag{5.57}$$

and κ_k represent model and measurement uncertainties, respectively. The desired closed loop system appears when $\hat{p}_k = p_k$ and $\hat{x}_{r,k} = x_k$, leading to

$$\begin{bmatrix} \boldsymbol{x}_{r,k} \\ p_k \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_r - \boldsymbol{\Gamma}_r \boldsymbol{L}_r & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{r,k-1} \\ p_{k-1} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Gamma}_r \\ 0 \end{bmatrix} r_{k-1} + \boldsymbol{\xi_k} . \tag{5.58}$$

The system state estimate is performed taking into account the desired closed loop response, the uncertainties $\boldsymbol{\xi}_{x_{r,k}}$, the deterministic term due to the system input and the augmented state[4]

$$\begin{bmatrix} \hat{\boldsymbol{x}}_{r,k} \\ \hat{p}_k \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_{r,n} - \boldsymbol{\Gamma}_{r,n} \boldsymbol{L}_r & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{x}}_{r,k-1} \\ \hat{p}_{k-1} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Gamma}_{r,n} \\ 0 \end{bmatrix} \boldsymbol{r}_{k-1} + \boldsymbol{K}_k \left(y_k - \hat{y}_k \right) \tag{5.59}$$

with

$$\hat{y}_k = \boldsymbol{C}_a \left(\begin{bmatrix} \boldsymbol{\Phi}_{r,n} - \boldsymbol{\Gamma}_{r,n} \boldsymbol{L}_r & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{x}}_{r,k-1} \\ \hat{p}_{k-1} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Gamma}_{r,n} \\ 0 \end{bmatrix} \boldsymbol{r}_{k-1} \right) . \tag{5.60}$$

[3] The subscript k indicates the iteration.

[4] $\boldsymbol{\Phi}_{r,n}$ and $\boldsymbol{\Gamma}_{r,n}$ are the nominal representation of $\boldsymbol{\Phi}_r$ and $\boldsymbol{\Gamma}_r$, respectively. These matrices are computed using $K_{s,n}$ and $K_{2,n}$, the stiffness and damping nominal value, respectively.

The Kalman gain $\boldsymbol{K}_k$ reflects the uncertainty of each state being a function of $\boldsymbol{\xi}_k$ and $\boldsymbol{\kappa}_k$ (Aström and Wittenmark (1997))

$$\boldsymbol{K}_k = \boldsymbol{P}_{1k}\boldsymbol{C}_a^T \left[\boldsymbol{C}_a\boldsymbol{P}_{1k}\boldsymbol{C}_a^T + \boldsymbol{R}_k\right]^{-1} \tag{5.61}$$

with

$$\boldsymbol{P}_{1k} = \boldsymbol{\Phi}_n\boldsymbol{P}_{k-1}\boldsymbol{\Phi}_n^T + \boldsymbol{Q}_k \tag{5.62}$$

$$\boldsymbol{P}_k = \boldsymbol{P}_{1k} - \boldsymbol{K}_k\boldsymbol{C}_a\boldsymbol{P}_{1k} \tag{5.63}$$

where $\boldsymbol{\Phi}_n$ is the open loop augmented state transition matrix (5.53)[5]

$$\boldsymbol{\Phi}_n = \begin{bmatrix} \boldsymbol{\Phi}_{r,n} & \boldsymbol{\Gamma}_{r,n} \\ 0 & 1 \end{bmatrix} . \tag{5.64}$$

$\boldsymbol{R}_k = E\left\{\boldsymbol{\kappa}_k\boldsymbol{\kappa}_k^T\right\}$ and $\boldsymbol{Q}_k = E\left\{\boldsymbol{\xi}_k\boldsymbol{\xi}_k^T\right\}$ are, respectively, measurement and system noise matrix

$$\boldsymbol{Q}_k = \begin{bmatrix} Q_{x_{r,k}} & \boldsymbol{0} \\ \boldsymbol{0} & Q_{p_k} \end{bmatrix} . \tag{5.65}$$

$\boldsymbol{\xi}_k$ and $\boldsymbol{\kappa}_k$ are the stochastic inputs due to model and measure uncertainties, respectively. $\boldsymbol{P}_k$ is the mean-square error matrix and its initial value reflects the uncertainty associated with the state estimation. Its initial value should not be lower than the initial value of $\boldsymbol{Q}_k$.

Stochastic Design Strategy

The observer gain $\boldsymbol{K}_k$ is defined by the relation between the stochastic matrices $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$, and not by their absolute value. Following a normalization strategy for $\boldsymbol{R}_k$ ($\boldsymbol{R}_k = 1$), the estimation strategy and the $\boldsymbol{K}_k$ dynamics is set by designing $\boldsymbol{Q}_k$ in order to reflect the uncertainty of the states (Cortesao et al. (2010)). Setting $\boldsymbol{Q}_k$ with a relatively small value with respect to $\boldsymbol{R}_k$, means that a model based strategy is followed, while a relatively high value of $\boldsymbol{Q}_k$ accounts for a sensor based approach.

State Feedback Control

The feedback gain $\boldsymbol{L}_r$ is designed based on desired close loop poles. The system is controllable and of dimension three, with two core states and one

[5] The desired closed loop system (5.58) is non-observable

extra state due to the delayed input ($d = 1$). In continuous time, the closed loop characteristic polynomial is given by (without taking into account dead-time)

$$s^2 + 2\zeta\omega_n s + \omega_n^2 \, .$$

Its discrete time representation (in the Z domain) is

$$z^2 + a_1 z + a_2 \, ,$$

with a_1 and a_2 computed as (Aström and Wittenmark (1997))

$$a_1 = -2e^{\zeta\omega_n h} \cos\left(\sqrt{1-\zeta^2}\omega_n h\right) \, , \tag{5.66}$$

$$a_2 = e^{-2\zeta\omega_n h} \, . \tag{5.67}$$

The state feedback gain $\boldsymbol{L}_r$ is set to achieve a critically damped behavior ($\zeta = 1$), with time constant τ_f

$$\tau_f = \frac{1}{\omega_n} \, , \tag{5.68}$$

and the extra pole due to T_d mapped to $z = 0$. Through the Ackermann's formula, $\boldsymbol{L}_r$ is computed by

$$\boldsymbol{L}_r = \begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix} \boldsymbol{W}_c^{-1} \boldsymbol{P}(\boldsymbol{\Phi}_{r,n}) \, . \tag{5.69}$$

$\boldsymbol{W}_c$ is the controllability matrix (Aström and Wittenmark (1997)),

$$\boldsymbol{W}_c = \begin{bmatrix} \boldsymbol{\Gamma}_r & \boldsymbol{\Phi}_{r,n}\boldsymbol{\Gamma}_{r,n} \dots \boldsymbol{\Phi}_{r,n}^{m-1}\boldsymbol{\Gamma}_{r,n} \end{bmatrix} \, , \tag{5.70}$$

where the superscript m represents the system order. $\boldsymbol{P}(\boldsymbol{\Phi}_{r,n})$ is the characteristic polynomial in $\boldsymbol{\Phi}_{r,n}$, and is given by

$$\boldsymbol{P}(\boldsymbol{\Phi}_{r,n}) = \boldsymbol{\Phi}_{r,n}^{d+2} + a_1 \boldsymbol{\Phi}_{r,n}^{d+1} + a_2 \boldsymbol{\Phi}_{r,n}^{d} \, . \tag{5.71}$$

AOB Contact Model Adaptation

Consistent force responses require that the AOB contact model adapts when the environment changes. If the environment stiffness changes from K_s to $K_s + \Delta K_s$, only Γ_1 needs to be updated in $\boldsymbol{\Phi}_r$ (see (5.43)) (Cortesão et al. (2003))

$$\Gamma_1^{K_s+\Delta K_s} = \Gamma_1^{K_s} + \Delta\Gamma_1 \, , \tag{5.72}$$

$$\Delta\Gamma_1 = \frac{\Delta K_s}{K_s}\Gamma_1^{K_s} \, . \tag{5.73}$$

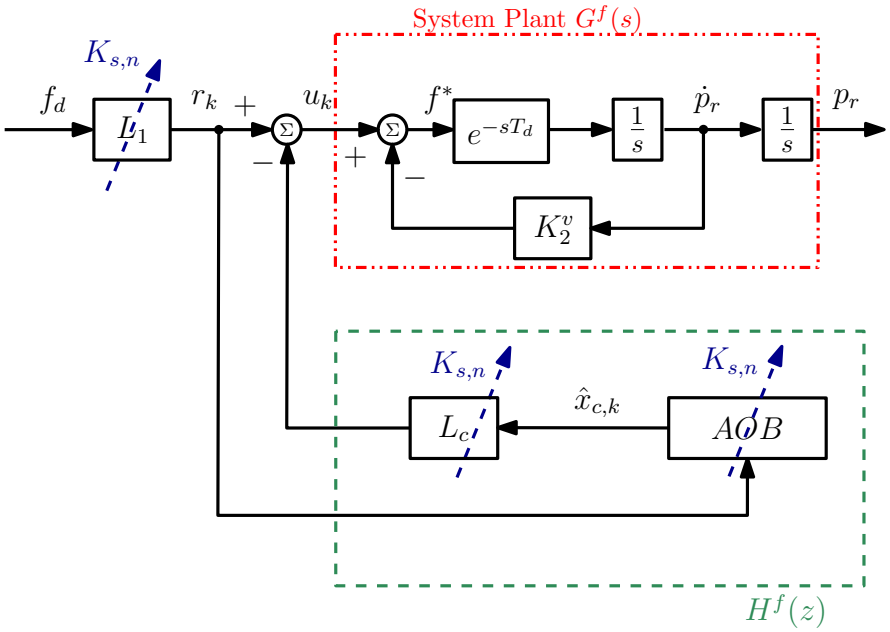


Figure 5.3— Force control with an AOB in free space. The AOB generates a "virtual state" into the system plant. This plant is not stable, having a continuous pole at $s = 0$ and a discrete pole at $z = 1$. H_z^f is the transfer function due to the AOB and $G^f(s)$ is robot plant in free space. The dashed blue arrows show the control variables that change with the stiffness adaptation $K_{s,n}$.

The system feedback gains also need to be updated (Cortesão and Koeppe (2001))

$$\boldsymbol{L}_r = \begin{bmatrix} \frac{L_1}{1+\Delta K_s/K_s} & \frac{L_2}{1+\Delta K_s/K_s} & L_3 & \dots & L_n \end{bmatrix} , \tag{5.74}$$

while the gains due to T_d remain the same. The core state feedback gains, L_1 and L_2, are proportional to $\frac{1}{K_s}$.

5.3.4 Free Space Position Control

The force control architecture is kept also in free space (no control switching). Let's rewrite (5.59) and (5.60) in a more compact form as

$$\hat{\boldsymbol{x}}_{c,k} = \boldsymbol{\Phi}_c \hat{\boldsymbol{x}}_{c,k-1} + \boldsymbol{\Gamma}_c r_{k-1} + \boldsymbol{K}_k \left(y_k - \hat{y}_k \right) , \tag{5.75}$$

$$\hat{y}_k = \boldsymbol{C}_a \left(\boldsymbol{\Phi}_c \hat{\boldsymbol{x}}_{c,k-1} + \boldsymbol{\Gamma}_c r_{k-1} \right) , \tag{5.76}$$

where

$$\boldsymbol{\Phi}_c = \begin{bmatrix} \boldsymbol{\Phi}_r - \boldsymbol{\Gamma}_r \boldsymbol{L}_r & 0 \\ 0 & 1 \end{bmatrix} , \qquad \boldsymbol{\Gamma}_c = \begin{bmatrix} \boldsymbol{\Gamma}_r \\ 0 \end{bmatrix} , \qquad \hat{\boldsymbol{x}}_{c,k} = \begin{bmatrix} \hat{\boldsymbol{x}}_{r,k} \\ \hat{p}_k \end{bmatrix} .$$

Assuming that inertial forces are effectively eliminated from the force control loop, force measurements in free space are zero ($f_m = 0$), and $\hat{\boldsymbol{x}}_{c,k}$ can be further simplified as (replacing (5.76) in (5.75))

$$\hat{\boldsymbol{x}}_{c,k} = \boldsymbol{\Phi}_c^f \hat{\boldsymbol{x}}_{c,k-1} + \boldsymbol{\Gamma}_c^f r_{k-1} \,, \tag{5.77}$$

with

$$\boldsymbol{\Phi}_c^f = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{C}_a)\, \boldsymbol{\Phi}_c \,, \tag{5.78}$$

and

$$\boldsymbol{\Gamma}_c^f = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{C}_a)\, \boldsymbol{\Gamma}_c \,. \tag{5.79}$$

The feedback control gain can also be represented in compact form as

$$\boldsymbol{L}_c = \begin{bmatrix} \boldsymbol{L}_r & 1 \end{bmatrix} \,. \tag{5.80}$$

Fig. 5.3 shows the force controlled robot in free space. The AOB state estimation generates a *virtual state* $\hat{\boldsymbol{x}}_{c,k}$ that enters in the system (Cortesão et al. (2006)). In free space, the robot transfer function is given by

$$G^f(s) \approx \frac{e^{-sT_d}}{s(s+K_2)} \,, \tag{5.81}$$

while the force control transfer function arises as (Z-transform domain)

$$H^f(z) = \boldsymbol{L}_c \left(z\boldsymbol{I} - \boldsymbol{\Phi}_c^f \right)^{-1} \boldsymbol{\Gamma}_c^f \,. \tag{5.82}$$

The overall system transfer function in free space (force controlled robot $G^f(z)$ plus the AOB) is given by

$$G_s^f(z) = L_1 \left[G^f(z) - H^f(z) G^f(z) \right] \,, \tag{5.83}$$

where $G^f(z)$ is the pulse-transfer function of $G^f(s)$, obtained by preceding $G^f(s)$ by a zero-order hold[6]

$$G^f(z) = \frac{1}{K_2} \frac{b_1 z + b_2}{z^2 + a_1 z + a_2} z^{-1} \,, \tag{5.84}$$

with

$$\begin{aligned} a_1 &= -\left(1 + e^{-K_2 h}\right) \\ a_2 &= e^{-K_2 h} \\ b_1 &= \frac{K_2 h - 1 + e^{-K_2 h}}{K_2} \\ b_2 &= \frac{1 - e^{-K_2 h} - K_2 h e^{-K_2 h}}{K_2} \end{aligned} \,. \tag{5.85}$$

[6] A table of the most common pulse-transfer functions can be found in Aström and Wittenmark (1997)

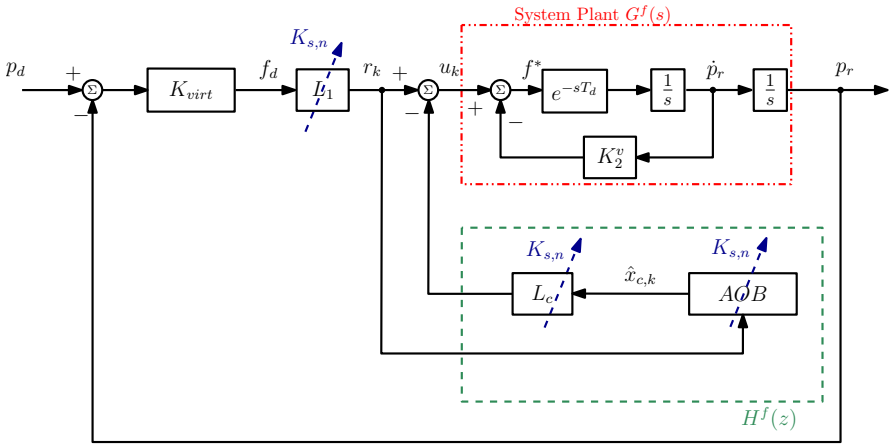


Figure 5.4— Position controlled robot with an AOB in free space. Control stability is achieved by computing force references f_d from position errors through virtual coupling K_{virt}, closing a position feedback loop around the force controlled robot. H_z^f is the transfer function due to the AOB and $G^f(s)$ is robot plant in free space. The dashed blue arrows show the control variables that change with the stiffness adaptation $K_{s,n}$.

This system is not stable, having a continuous pole at $s = 0$ and a discrete pole at $z = 1$. The continuous pole in $s = 0$ is due to the open loop plant, while the discrete pole at $z = 1$ is due to the active state (integral action).

Position Control

Position control is achieved by computing force references from position errors through virtual coupling K_{virt} (see (5.34)), closing a position feedback loop around the force controlled robot. With virtual coupling, the system transfer function is given by

$$G_{sp}^f(z) = \frac{K_{virt} G_s^f(z)}{1 + K_{virt} G_s^f(z)} \,. \tag{5.86}$$

Fig. 5.4 shows the force controlled robot in free space with an outer position control loop. Position tracking is achieved by the virtual stiffness and the force controller. In free space, the force control drives the robot towards the position reference, since force references are computed from position errors. In this way, as the robot gets closer to the position reference, both force references and position tracking errors approach zero.

5.3.5 Inertial Forces

Free space motion leads to non negligible inertial forces being captured by a wrist force sensor when a tool is attached to it. These forces enter in the AOB and in system linearization (see (3.67)), deteriorating control performance. Using the depth information from a 3D time-of-flight camera, measured forces can be taken into account only near contact, removing free space inertial forces from the control loop. Let $\boldsymbol{P} \in \mathbb{R}^{W \times H}$ be a depth image, where W and H are image width and height, respectively. ${}^{c}\boldsymbol{p}(i,j) \in \mathbb{R}^4$ is the 3D homogeneous coordinates of the i^{th}, j^{th} camera pixel projected in the camera frame

$$ {}^{c}\boldsymbol{p}(i,j) = \begin{bmatrix} {}^{c}p_x(i,j) & {}^{c}p_y(i,j) & {}^{c}p_z(i,j) & 1 \end{bmatrix}^T . $$

Knowing ${}^{p}\boldsymbol{T}_c \in \mathbb{R}^{4\times4}$, the homogeneous transformation matrix between camera and probe, ${}^{c}\boldsymbol{p} \in \mathbb{R}^4$ can be represented in probe frame as

$$ {}^{p}\boldsymbol{p}(i,j) = {}^{p}\boldsymbol{T}_c \, {}^{c}\boldsymbol{p}(i,j) . \tag{5.87} $$

The minimum distance of all 3D pixel coordinates to the probe frame is given by

$$ d_o = \min d_p(i,j) \quad i : 1,\ldots,W \; j : 1,\ldots,H , \tag{5.88} $$

where $d_p(i,j) = \|{}^{p}\boldsymbol{p}(i,j)\|$ is the Euclidean distance of ${}^{p}\boldsymbol{p}(i,j)$ to the probe frame.

Taking into account d_o, the new force variable $\boldsymbol{f}_m \in \mathbb{R}^3$, is given by

$$ \boldsymbol{f}_m = \begin{cases} \sigma_d\left(c, -\left(d_o - d_t\right)\right) \boldsymbol{f}_{fs} & \text{if } p_c = 0 \\ \boldsymbol{f}_{fs} & \text{if } p_c = 1 \end{cases} . \tag{5.89} $$

$\boldsymbol{f}_{fs} \in \mathbb{R}^3$ is the force sensor measurement and p_c is a boolean variable used to identify contact

$$ p_c = \begin{cases} 1 & \text{if } d_o < d_t \text{ and } f_{min} < \|\boldsymbol{f}_{fs}\| \\ 1 & \text{if } d_o > d_t \text{ and } f_{max} < \|\boldsymbol{f}_{fs}\| \\ 0 & \text{otherwise} \end{cases} , \tag{5.90} $$

where f_{min} is a threshold value for inertial forces near contact and f_{max} is a threshold value for contact not identified by the 3D camera. σ_d is a general sigmoid function, given by

$$ \sigma_d\left(c, x\right) = \frac{1}{1 + e^{-cx}} , \tag{5.91} $$

which acts as a smooth switch of d_o centered around d_t. The parameter c is a smoothing factor that defines the adaptation dynamics while d_t acts as a threshold for force measurements, anticipating contact.

5.4 Orientation Control

The orientation control is performed employing a joint space velocity control driven by task space orientation errors, following the strategy presented in Section 4.3.

Let $\boldsymbol{\omega}_s \in \mathbb{R}^3$ be the angular velocity reference, computed from the orientation error as

$$\boldsymbol{\omega}_s = \boldsymbol{I}\boldsymbol{\epsilon}_{dr} \ , \tag{5.92}$$

where $\boldsymbol{I} \in \mathbb{R}^{3\times3}$ is the identity matrix and $\boldsymbol{\epsilon}_{dr} \in \mathbb{R}^3$ is the orientation error computed by (4.19). The task space angular velocity reference is mapped to the joint space by the inverse differential kinematics equation

$$\dot{\boldsymbol{q}}_s = \boldsymbol{J}_s^{\dagger}(\boldsymbol{q})\boldsymbol{\omega}_s \ , \tag{5.93}$$

where $\boldsymbol{J}_s^{\dagger}(\boldsymbol{q}) \in \mathbb{R}^{3\times n}$ is the Moore-Penrose pseudo-inverse of the orientation Jacobian. $\dot{\boldsymbol{q}}_s$ is the joint space counterpart of $\boldsymbol{\omega}_s$, used to compute the reference for the joint velocity controller.

5.4.1 Joint Space Velocity Control

The joint velocity reference $\dot{\boldsymbol{q}}_d$ is computed by scaling $\dot{\boldsymbol{q}}_s$ by a proportional control gain $\boldsymbol{K}_1$,

$$\dot{\boldsymbol{q}}_d = \boldsymbol{K}_1\dot{\boldsymbol{q}}_s \ . \tag{5.94}$$

In the sequel, the control analysis will be done for each joint individually, being controlled by an AOB.

System Plant

The orientation controller is designed assuming the dynamics of a double integrator in the null space (see (5.25)). When controlling the system at velocity level, the system plant reduces to a single integrator, given by

$$G(s) = \frac{e^{-sT_d}}{s} \ . \tag{5.95}$$

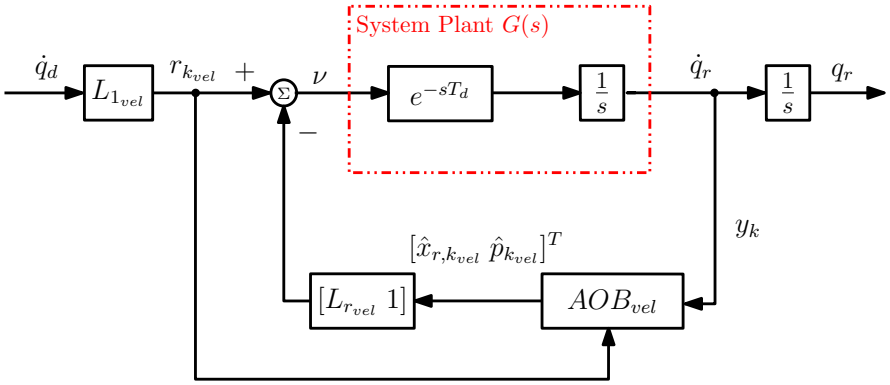


Figure 5.5— Joint velocity control with an AOB in the loop. Each joint is controlled individually.

The equivalent temporal representation is given by

$$\dot{y}(t) = \nu(t - T_d) \tag{5.96}$$

where ν is the plant input (joint acceleration) and $y(t)$ is the plant output (joint velocity $\dot{q}_r$). Defining $x_{r1}(t) = y(t)$, the state-space representation of (5.96) is given by (5.39), with

$$A = 0 \quad B = 1 \text{ and } C = 1\,.$$

The equivalent discrete system is achieved following the formalism presented in Sections 5.3.1 and 5.3.2 . The AOB algorithm has been discussed in Section 5.3.3 and the orientation control variable ν is given by

$$\nu = L_{1_{vel}}\dot{q}_d - \begin{bmatrix} \boldsymbol{L}_{r_{vel}} & 1 \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{x}}_{r,k_{vel}} \\ \hat{p}_{r,k_{vel}} \end{bmatrix} . \tag{5.97}$$

Fig. 5.5 shows the joint velocity control with the AOB in the loop.

Control Design

Motion control is designed to provide a high tracking performance, to enforce a stiff behavior at orientation level. $\boldsymbol{L}_{r_{vel}}$ is set through the Ackermann formula in accordance with the velocity loop time constant τ_v. A fast response is achieved by setting a small τ_v. K_1 is designed assuming that $\boldsymbol{J}_s^{\dagger}(\boldsymbol{q})$ and T_d have no influence in motion control dynamics. With this assumption, the pole

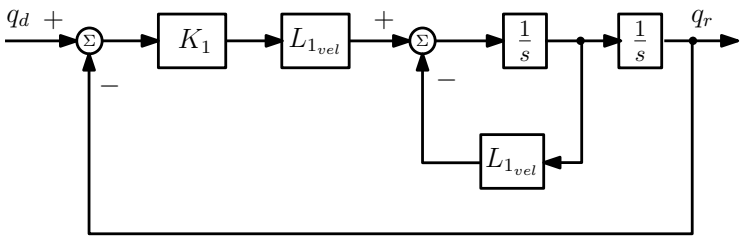


Figure 5.6— Simplified orientation control architecture at joint level. Similarly to what was performed in Section 4.3, the joint control gains are chosen based on this assumption.

placement gain reduces to $L_{1_{vel}}$. For the desired plant model ($p_{k_{vel}} = \hat{p}_{k_{vel}}$), the closed loop transfer function of the orientation control is equal to (see Fig. 5.6)

$$G(s) = \frac{Y(s)}{X(s)} = \frac{K_1 L_{1_{vel}}}{s^2 + L_{1_{vel}} s + K_1 L_{1_{vel}}} \,. \tag{5.98}$$

For a critically damped behavior ($\zeta = 1$), K_1 is given by

$$K_1 = \frac{L_{1_{vel}}}{2^2 \zeta^2} \,. \tag{5.99}$$

With the presence of $\boldsymbol{J}_s^{\dagger}(\boldsymbol{q})$ in the control law, the decoupled system is no longer achievable. Although being an approximation, this approach provides a “good guess" for the K_1 design, giving a good tracking performance without any notorious overshoots in the end-effector orientation.

5.4.2 Inverse Differential Kinematics

Equation (5.93) computes the joint velocity reference without any force control constraint. As a result, coupling effects with force control torques may arise, being canceled by the null space projector (see (5.11)). Thus, only redundant joints are used for orientation motion, which might introduce orientation errors. Since $\boldsymbol{J}_s^{\dagger}(\boldsymbol{q})$ is the Moore-Penrose pseudo-inverse, the solution of (5.93) will instantly minimize the joint velocity norm and, as long as $\dot{\boldsymbol{q}}_s$ is not parallel to $\boldsymbol{\tau}_p$, orientation motion is going to be performed by redundant joints. As a consequence of this approach, error convergence towards zero might be slower and, eventually, might converge to a non-zero error state if $\dot{\boldsymbol{q}}_s$ is parallel to $\boldsymbol{\tau}_p$. Slower convergence can be mitigated by setting higher control gains.

5.5 Experiments

This section assesses the hierarchical control architecture performance in free space. Motion tracking capabilities are evaluated against the joint space controller presented in Section 4.3. Control performance in contact is addressed in Chapters 6 and 7.

5.5.1 Control Design

Force and motion control design are going to be presented in this section. The control sampling time and the system time delay are $h = 1.2[ms]$ and $T_d = h$, respectively.

Force Control Design

Typically, the manipulator structural damping K_2^s is neglected in the AOB model ($K_2^s = 0$ and $K_{2,n} = K_2^v$). For this setup, setting $K_{2,n} = K_2^v = 16.6$ provides good control performance, however it introduces noticeable high frequency noise perturbation due to high K_2^v. Reducing both $K_{2,n}$ and K_2^v affects control performance, decreasing system responsiveness. Recognizing that the manipulator structure has some internal damping ($K_2^s > 0$), $K_{2,n}$ can be set in the AOB model higher than the velocity loop feedback gain K_2^v (see (5.36)), which reduces high frequency noise without any notorious drop off in control performance. Following this approach, the damping parameters are set to $K_{2,n} = 16.6$ and $K_2^v = 12.5$, being both tuned experimentally. The force control time constant is set to $\tau_f = 0.035[s]$, which leads to a desired closed loop behavior with a double pole ($\zeta = 1$) at $s_{1,2} = -28.57$. $K_{s,n}$ is set to 10 and K_{virt} is set to 40. Both values were experimentally tuned, with nominal low values to reduce the effects of modeling free space as a spring. Low $K_{s,n}$ value leads to higher feedback gains, which results in manipulator motion with fast dynamics. Pole placement gains in free space are

$$\boldsymbol{L}_r = \begin{bmatrix} 79.68 & 3.98 & 0.048 \end{bmatrix} ,$$

being given by (5.69). The null space gain $\boldsymbol{K}_N \in \mathbb{R}^{7\times 7}$ is set as

$$\boldsymbol{K}_N = diag \begin{bmatrix} 50 & 50 & \dots & 50 \end{bmatrix} .$$

Table 5.1: Motion Control Design.

	τ_v	K_1	$L_{1_{vel}}$	$L_{2_{vel}}$
q_1	0.008	29.02	116.08	0.14
q_2	0.008	29.02	116.08	0.14
q_3	0.0075	30.8	123.21	0.15
q_4	0.0075	30.8	123.21	0.15
q_5	0.005	44.45	177.81	0.21
q_6	0.005	44.45	177.81	0.21
q_7	0.005	44.45	177.81	0.21

Stochastic Design

For the AOB stochastic design, a sensor based strategy (relatively high $\boldsymbol{Q}_k$ when compared to $\boldsymbol{R}_k$) is pursued with

$$\boldsymbol{Q}_k = diag \begin{bmatrix} 10^{-1} & 10^{-12} & 10^{-12} & 3 \times 10^{-2} \end{bmatrix}$$

and $\boldsymbol{R}_k = 1$, leading to the following steady state Kalman gain in free space

$$\boldsymbol{K}_k = \begin{bmatrix} 0.2705 & 0.1133 & 0.148 & 0.148 \end{bmatrix}^T .$$

A sensor based design enables better free space position tracking, reducing model errors introduced by modeling free space as a spring. $\boldsymbol{f}_d$ is filtered with a first order low pass Butterworth filter with cutoff frequency of $25[Hz]$.

Motion Control Design

Table 5.1 summarizes motion control gains. For the AOB stochastic design, all joints have the same sensor based approach

$$\boldsymbol{Q}_{k_{vel}} = diag \begin{bmatrix} 10^{-1} & 10^{-12} & 10^{-12} \end{bmatrix} ,$$

entailing the following steady state Kalman gain

$$\boldsymbol{K}_{k_{vel}} = \begin{bmatrix} 0.2702 & 3.24 \times 10^{-10} & 3.24 \times 10^{-10} \end{bmatrix}^T .$$

High control gains and sensor based strategies tackle slow tracking control of the orientation.

Table 5.2: Absolute Mean Error (AME), Root Mean Square Error (RMSE) and Absolute Maximum Error (AMaxE) for Hierarchical Control Architecture vs. Joint Space Controller (Position Data).

	Joint			Hierarchical		
$[m]$	AME	RMSE	AMaxE	AME	RMSE	AMaxE
x	0.0044	0.0054	0.0129	0.0031	0.0037	0.0091
y	0.0053	0.0068	0.0220	0.0041	0.0050	0.0117
z	0.0027	0.0042	0.0210	0.0018	0.0024	0.0086

Table 5.3: Absolute Mean Error (AME), Root Mean Square Error (RMSE) and Absolute Maximum Error (AMaxE) for Hierarchical Control Architecture vs. Joint Space Controller (Orientation Data).

	Joint			Hierarchical		
$[rad]$	AME	RMSE	AMaxE	AME	RMSE	AMaxE
Roll	0.0767	0.1093	0.3851	0.0508	0.0820	0.3243
Pitch	0.0468	0.0634	0.2071	0.0695	0.0912	0.3146
Yaw	0.0830	0.1042	0.2692	0.0806	0.0966	0.2114

5.5.2 Discussion

Fig. 5.7 shows position tracking performance, while orientation tracking results are shown in Fig. 5.8. Tables 5.2 and 5.3 summarize both position and orientation control results. Position control results show improved tracking performance when compared to the joint control architecture, as can be seen in Table 5.2, validating the virtual coupling strategy for free space control. Orientation control performance is improved along Roll and Yaw dimensions and is worse along Pitch, which can be explained by the hierarchical approach (see Section 5.4.2). Overall, free space motion performance under hierarchical control is good, validating this solution to control the ultrasound probe motion in free space. Position control is improved due to a more advance control approach (AOB), and due to the hierarchical architecture which eliminates coupling effects from the orientation control in force/position tracking. On the other hand, orientation performance can be affected by null space projection. To cope with orientation tracking errors, better tracking performance can be achieved with higher control gains and/or with a more dynamic active state. In fact, the active state in orientation control is almost null to avoid

control issues (due to integral actions) at joint limits. Nevertheless, this design provides smooth probe motion with good tracking capabilities, being a conservative control design preferable due to the task nature (interaction with humans).

5.6 Conclusion

In this chapter the control architecture for the probe holder manipulator is presented. To cope with coupling errors between position and orientation control, a hierarchical approach is followed. Explicit Cartesian force control driven by position errors arises as the primary task, while a joint space orientation control is designed in the null space. Orientation is controlled by velocity control at joint level, being joint velocity references computed from task space orientation errors. Both controllers are boosted by Kalman active observers (AOBs). Free space performance shows promising results. Position tracking performance is improved when compared with the joint control approach, while orientation control results are also good, despite the null space projection.

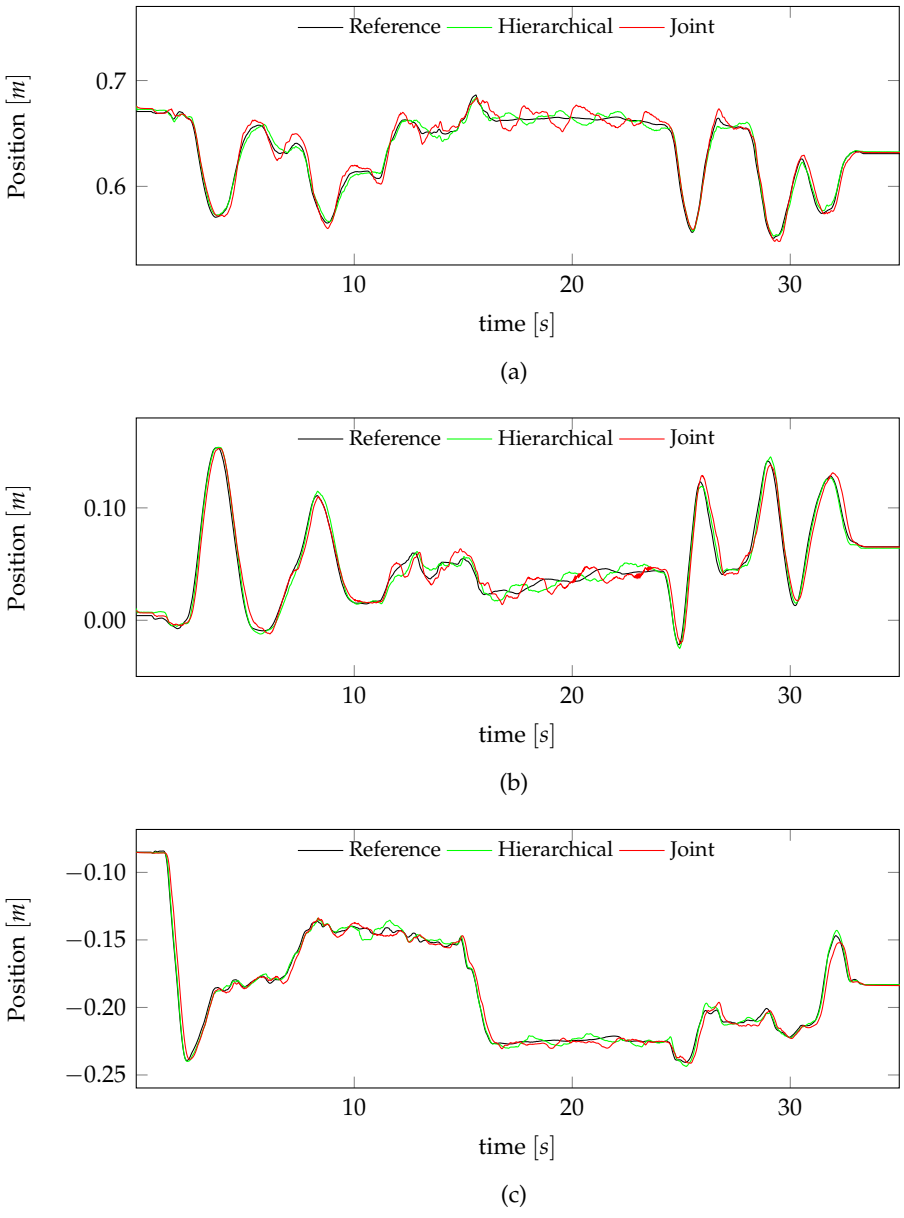
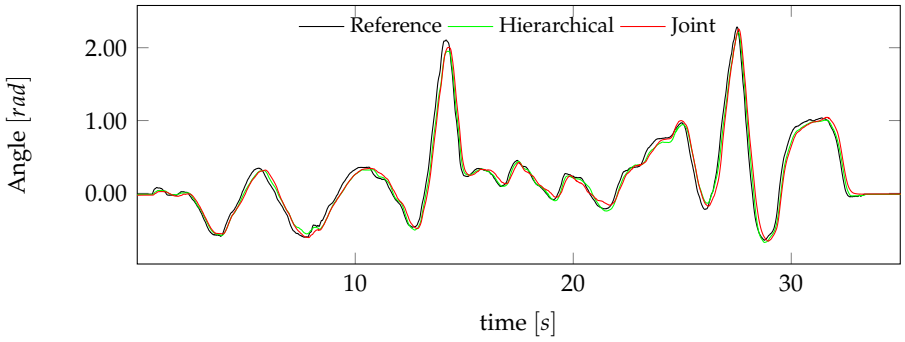
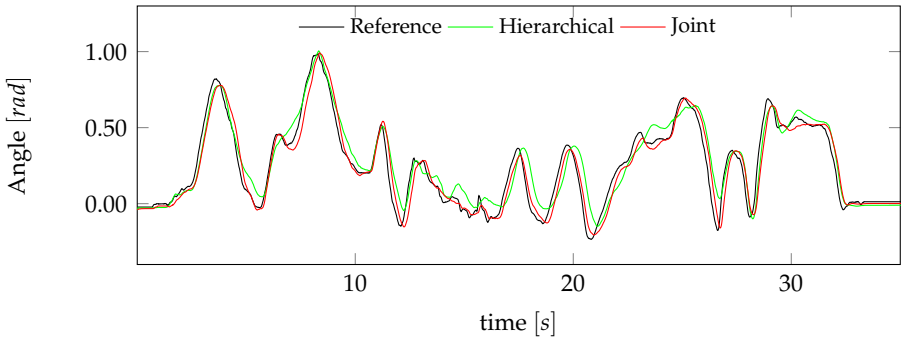


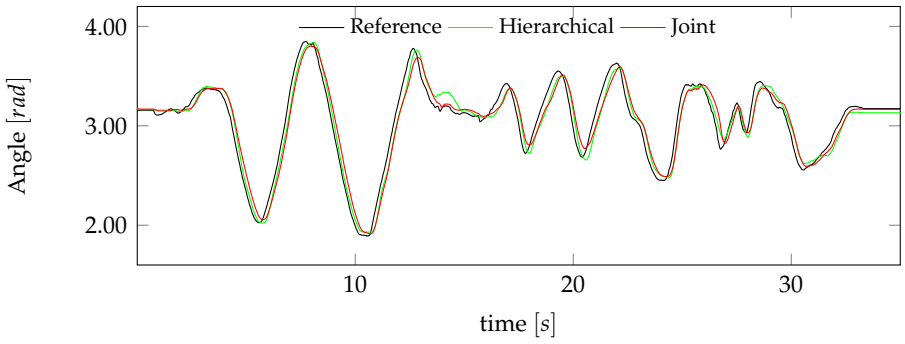
Figure 5.7— Cartesian position tracking performance. Hierarchical control architecture vs. joint space control with task space pose references. (a), (b) and (c) are respectively X, Y and Z tracking.



(a)



(b)



(c)

Figure 5.8— Orientation tracking performance. Hierarchical control architecture vs. joint space control with task space pose references. The orientation is represented by (a) Roll, (b) Pitch and (c) Yaw angles. Roll $\in [-\pi, \pi]$, Pitch $\in [-\pi, \pi]$ and Yaw angle $\in [0, 2\pi]$.

CHAPTER 6

ONLINE STIFFNESS ESTIMATION

Contact is a complex physical phenomenon. In the literature, different formulations have been followed to model it, ranging from a simple single parameter linear model to nonlinear multi parameter models, growing in model complexity when the desired estimation accuracy and/or the environment complexity increases. After a proper model selection, it is necessary to assign values to the contact model parameters, which is not a trivial task. The environment model[1] parameter estimation remains an open problem in robotics, playing a crucial role in interaction tasks, since proper estimation improves interaction control performance, enabling environment perception in tele-manipulation tasks and allowing computer simulation of contact scenarios. However, while an accurate environment description is required for contact computer simulation (Van Vliet (2000)) or to enable environment perception (Kikuuwe and Yoshikawa (2003)), from a control standpoint, rather than aiming to obtain a model that perfectly emulates contact dynamics, the critical issue is to ensure contact stability, boosting force tracking performance. In highly unstructured environments, as is the case in medical ultrasound tasks, where free space motion interchanges with soft and stiff contact scenarios, simpler but less accurate contact models might be preferable. Lumping the contact dynamics into a single parameter simplifies estimation algorithms, allowing for real-time contact parameter adaptation. In this way, the choice of a linear spring contact model is convenient.

[1] In this section, contact and environment model are used interchangeably.

Stiffness Estimation Overview

Following the idea proposed by Cortesão et al. (2006), the estimation is performed online resorting only to force data, relying on the transitory behavior of contact forces to reduce the mismatches between control model and the environment. In addition to force data, the estimation strategy also uses information about the robot inertial properties referred to the end-effector, i.e. its effective mass (Khatib (1995)). In force control tasks, due to the noisy profile of force sensor measurements, it is common to filter measured forces by low pass filtering (Volpe et al. (1993)). When the feedback force is filtered, robot dynamics is no longer decoupled from environment dynamics, despite performing inverse dynamics nonlinear feedback linearization, leading to different force responses for different contact orientations, since the contact mass is differently perceived in different robot configurations. Using the effective mass in the estimation algorithm, the stiffness perceived by the controller can be anticipated, improving force control performance in dynamic interactions, specially with stiff surfaces. Clearly, the elastic properties of the environment do not change with the interaction, however the controller perception of the environment does change, existing a strong correlation between the effective mass and the perceived stiffness. In this way, the perceived stiffness can be seen, not as a well defined physical property, but as an optimization parameter to improve force control performance. It is also important to notice that, since the control architecture is designed under the assumption that the robot is always in contact, to achieve a good position tracking performance, free space should be modeled as a very compliant environment. However, modeling free space with a low numerical stiffness value might introduce disruptive behaviors when changing from free space to constrained motion, specially when interacting with stiff environments, due to the abrupt change in control gains. To cope with this issue, when moving towards contact, the environment model is anticipated by 3D camera data, allowing pre-contact adaptation, providing a smooth transition from free space motion to contact interaction.

The remainder of this chapter is organized as follows. Section 6.1 reviews contact model formulations and contact parameter estimation strategies. Section 6.2 details the adaptation strategy from free space to contact, while Section 6.3 presents the stiffness estimation algorithm in contact. Section 6.4 explains why the stiffness perception change with the effective mass, presenting the new estimation approach. Perceived stiffness estimation experiments are

presented in Section 6.5 and Section 6.6 concludes the chapter.

6.1 Contact Modeling Overview

It is important to distinguish between contact and impact, different physical phenomenons but sometimes used interchangeably. According to Gilardi and Sharf (2002), *impact* occurs when two or more rigid bodies collide with each other, being the collision regarded to be almost instantaneously. It is characterized by large impulse-like reaction forces, large energy dissipation and large rate of change in the velocities of impacting bodies without significant change in their configuration. Impact analysis is performed assuming a discrete or impulse-momentum model, being typically divided in two intervals, before and after impact, with the relation between pre-impact and post-impact physical states established through coefficients, mainly the coefficient of restitution, defined along the normal direction, and coefficients along tangential directions (Gilardi and Sharf (2002)). Impact modeling aims to predict the post-impact system configuration given the pre-impact conditions.

On the other hand, *contact* is regarded as a continuous process in which two or more bodies come in touch with each other over a finite time period, with deformation on the contacting bodies surface. The continuous contact formulation aims to establish the contact force as a continuous function of the contacting bodies indentation and, possibly, indentation and force rate

$$f_n \equiv f_n \left(\delta, \dot{\delta}, \dot{f}_n\right) \ , \tag{6.1}$$

where f_n is the normal contact force, δ the bodies indentation, while $\dot{\delta}$ and $\dot{f}_n$ are the indentation and force rate of change.

6.1.1 Continuous Contact Models

Continuous contact models, also known as compliant contact models (Gilardi and Sharf (2002)) or contact force models (Lankarani and Nikravesh (1994)), are well suited to describe the contact dynamics between a robotic manipulator and the surrounding environment. Several continuous contact models have been described in the literature. Some of them are purely elastic, while others include dissipative terms. Most of them consider the robot to be much stiffer than the environment, with negligible indentation. This section reviews the most common contact models used in robotics.

The simplest approach to model contact is by assuming that the environment behaves as a linear spring, with the contact force proportional to the environment indentation. This approach is known as the Hooke's law and is given by

$$f_n = k_s \delta \,, \tag{6.2}$$

where k_s is the environment stiffness and δ is the indentation or deformation

$$\delta = p_0 - p_c \tag{6.3}$$

with p_0 and p_c as the environment rest and deformation position, respectively.

The Hooke's law is a simplistic approach for a very complex phenomenon. Contact modeling of soft tissues must account with energy dissipation that characterizes the interaction with vicoeslastic materials. The simplest approach to account for energy dissipation is by including a linear dashpot in the contact model. The most common approach to model a viscoelastic material is the Kelvin-Voight model (KV), which models the environment as the mechanical parallel of a damping term with a Hookean spring (Diolaiti et al. (2005))

$$f_n = k_s \delta + d_s \dot{\delta} \,, \tag{6.4}$$

where d_s is the damping term and $\dot{\delta}$ is the indentation rate. While being a popular model due to its relatively simplicity, KV model present some physical inconsistencies. At contact beginning, the damping term leads to non-null contact forces due to non-null contact velocities, while at separation, at the end of the restitution phase, the KV model give rise to an unnatural force holding the bodies together (sticking force), since the force due to the damping term becomes dominant regarding the elastic term. Fig. 6.1 shows a typical hysteresis loop of KV contact model, highlighting these issues. Furthermore, the equivalent of the coefficient of restitution for this model does not depend of the impact velocity whose dependence has been demonstrated experimentally (Gilardi and Sharf (2002)).

If the visco-elastic behavior of the material is noteworthy, more complex contact models can be obtained by alternative combinations of linear components. The Maxwell contact model is obtained by a linear spring and a linear damper in series, the Kelvin-Boltzmann or zener model (Moreira et al. (2012), Biagiotti and Melchiorri (2007)) is obtained as the series of a linear spring with

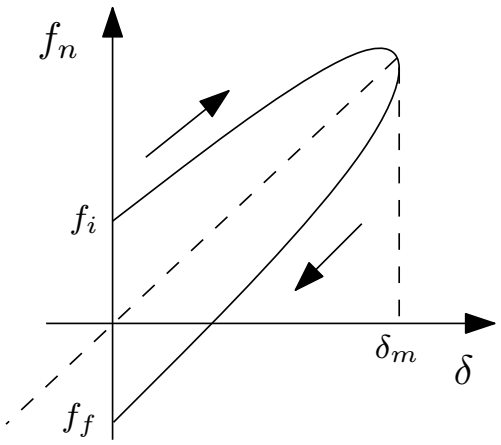


Figure 6.1— Typical behavior of a viscoelastic material modeled by the Kelvin-Voigt (KV) contact model. Hysteresis loop for linear damping. The compression phase corresponds to the upper arc, connecting the point f_i to the maximum indentation δ_m, while the restitution phase is represented by the lower arc, connecting δ_m to the point f_f.

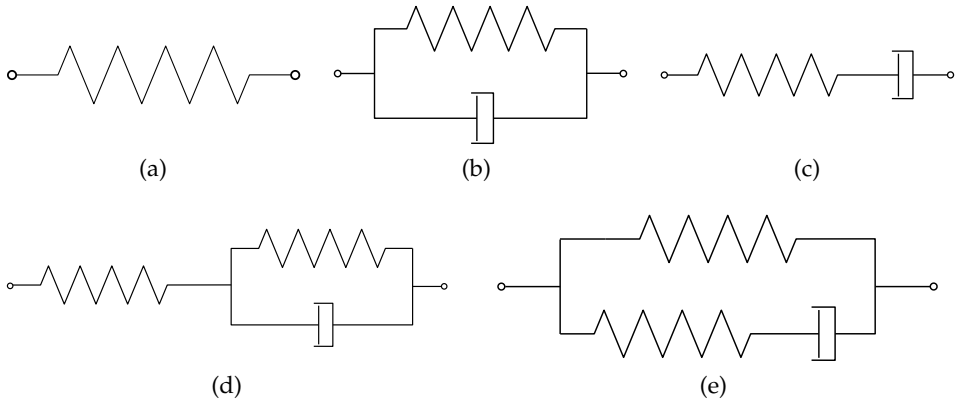


Figure 6.2— Linear contact models mechanical scheme. (a) Hook contact model. (b) Kelvin-Voigt contact model. (c) Maxwell contact model. (d) Kelvin-Boltzmann contact model. (e) Standard linear solid contact model.

a KV model while the standard linear solid model is a parallel composition of a Maxwell model with a linear spring (Khan et al. (2006)). Fig. 6.2 shows the schematics of these linear models.

Rather than being a linear phenomenon, contact presents an intrinsically nonlinear behavior. In this way, nonlinear models present a better physical consistency. Heinrich Hertz, in addition to his notable work with electro-

magnetic waves, was also a pioneer in contact modeling (Hertz et al. (1896)). While studying Newton's optical interference fringes in the gap between two glass lenses, Hertz wondered about the possible influence that the elastic deformation on the lenses surface might have in the optical interference fringes (Johnson (1985)). The Hertz contact model, also known as the power model, relates contact force with a nonlinear power function of local indentation as

$$f_n = k_s \delta^n \ , \tag{6.5}$$

where the exponent n is a real number, usually close to unity. Both k_s and n are determined from material and geometric properties of contacting bodies and can be computed analytically for central collision of axisymmetric bodies, such as sphere-sphere or sphere-half-space collisions.

Hunt and Crossley (1975) expanded the work of Hertz, proposing a nonlinear contact model for viscoelastic materials. In order to address the physical inconsistencies of KV model, the Hunt-Crossley model (HC) models contact as a Hertzian spring in parallel with a nonlinear damping term

$$f_n = k_s \delta^n + d_s \delta^n \dot{\delta} \ . \tag{6.6}$$

with the damping term as a function of contact indentation. This approach addresses the physical inconsistencies of KV model, eliminating the contact force discontinuities at contact beginning and at separation, with the modeled force beginning and finishing with an expected null value. Fig. 6.3 shows a typical hysteresis loop of a viscoelastic material modeled by the HC model.

6.1.2 Contact Modeling Parameter Identification - A literature review

Choosing the contact parameters that best describe the contact dynamics is not a trivial task. In the literature, different approaches have been followed to identify the contact parameters. Depending on the contact model followed and the estimation purpose, the contact parameter estimation can be more or less accurate.

Linear Models Parameter Identification

Linear models are typically used in force or impedance controllers, being the linear approximation enough to boost control performance and stability, while

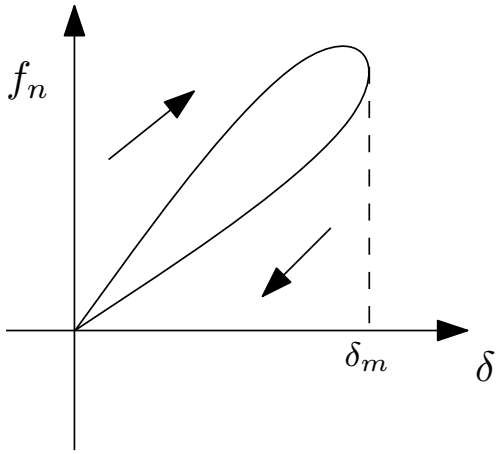


Figure 6.3— Typical behavior of a viscoelastic material modeled by the Hunt-Crossley contact model. Hysteresis loop for nonlinear damping. At the beginning of contact and at separation, the contact force is modeled with an expected zero value, addressing the physical inconsistencies of KV model.

keeping the estimation algorithm relatively simple to be performed online. Love and Book (1995) improved the performance of an impedance controller by including an estimation of the environment stiffness in the impedance control parameters. The environment stiffness is estimated offline by a multiple input-multiple output (MIMO) recursive least-squares (RLS) algorithm. Adaptive control techniques are used for stiffness estimation in Kim et al. (2000) and Seraji and Colbaugh (1997) and for stiffness and damping estimation in Singh and Popa (1995). In these approaches, stability is assessed by Lyapunov techniques, while parameter convergence can only be ensured for frequency-rich signals. Kikuuwe and Yoshikawa (2002) present a RLS algorithm to estimate the mechanical impedance perceived by a robot when interacting in a constrained environment. A speed-dependent forgetting factor together with a discontinuity detection approach allow to quickly detect environment transitions, improving online estimation performance. A similar approach is proposed by Wang et al. (2009). The authors use a time-varying forgetting factor in a RLS algorithm, claiming that the algorithm shows fast tracking performance in dynamic environments and robustness to noisy measurements. Moreira et al. (2014) assessed the performance of several contact models for biological soft tissue modeling, estimating the contact parameters by means of an offline linear least squares method. The authors conclude that

the Kelvin-Boltzmann and the Hunt-Crossley models are the ones that achieve the best results, choosing the Kelvin-Boltzmann model to be used in a model reference adaptive control (MRAC) controller due to its linearity. Erickson et al. (2003) perform a comparative study between the methods presented by Love and Book (1995), Seraji and Colbaugh (1997), and Singh and Popa (1995) and an offline approach proposed by them, in which the KV model parameters are estimated by an offline signal processing method.

Nonlinear Models Parameter Identification

When pursuing an accurate identification of environment contact dynamics, nonlinear contact models have a better physical consistency. Diolaiti et al. (2005) presented a double-stage RLS online estimation algorithm to identify the contact parameters of distinct materials modeled by the HC model. The authors show that the HC model is able to effectively describe the contact dynamics of both stiff and compliant materials. HC model parameter identification is improved using a single stage method proposed by Haddadi and Hashtrudi-Zaad (2012). Typically, the identified contact dynamics also contain the structural dynamics of the robotic system, which most of times is negligible. Verscheure et al. (2009) estimate the contact dynamic parameters of stiff robotic payloads modeled by a Hertzian spring, excluding the structural dynamics of the robotic system and the worksite from the estimation, since for stiff payloads the structural dynamics of the robotic system are no longer negligible. The estimation is performed by a total least squares estimator, taking into account nonidealities such as surface roughness. The estimation performance is assessed against finite element modeling and a material testing machine for payloads with different geometries.

6.1.3 Geometric Uncertainties and Stiffness Estimation based only on Force Data

Most of previous contact estimation algorithms require the availability of force, position and, sometimes, velocity measurements, in addition with a precise knowledge of the contact point. Geometrical uncertainty of contact position is the biggest obstacle to achieve a high-quality estimation, especially in contact with stiff environments, where a small error in contact location leads to large force errors. The great majority of previous estimators, deal with this issue

by assuming that the environment is static and the contact position is well known (e.g. Love and Book (1995)) or by starting the measurements with the robot at rest in uncompressed contact with the environment (e.g. Haddadi and Hashtrudi-Zaad (2012)). For most robotic applications, where the environment is unstructured and free space motion interchanges with soft and stiff contact scenarios, such data or setup are almost unpractical. Some authors attempt to overcome the geometrical uncertainty issue by combining contact parameter estimation with the estimation of the environment geometric parameters (e.g. Seraji and Colbaugh (1997) and Verscheure et al. (2010)), adding to an already difficult problem (contact parameter estimation) one that is equally or even more challenging.

On the other hand, some authors address the geometrical uncertainty problem by estimating the contact parameters resorting only to force data, avoiding using geometric information in the estimation algorithm. The offline signal processing technique of Erickson et al. (2003) was one of the first approaches to estimate the contact parameters (KV contact model) resorting only to force data. Aiming to improve the force tracking performance of a MRAC, Coutinho (2013) developed an extensive research about contact stiffness estimation based only on force sensing. Resorting to the AOB framework, Coutinho and Cortesão (2011) computed the environment stiffness by comparing the AOB active state with several theoretical predictions in different mismatch scenarios between model and environment stiffness. The stiffness estimation value is the one that minimizes, in the least squares sense, the error between the theoretical prediction and the active state. In Coutinho and Cortesão (2014), the contact stiffness is estimated based on the force error of two force observers tuned with different nominal stiffnesses, while in Coutinho and Cortesão (2013) the estimation is performed by an artificial neural network.

6.2 Stiffness Adaptation Towards Contact

Since the control architecture is designed assuming that the robot is always in contact (see Section 5.3.4), to achieve a good position tracking performance, free space must be modeled as a very compliant environment, with a very low nominal stiffness value. When moving towards contact, specially when interacting with a stiff surface, the abrupt change in control gains (from a very

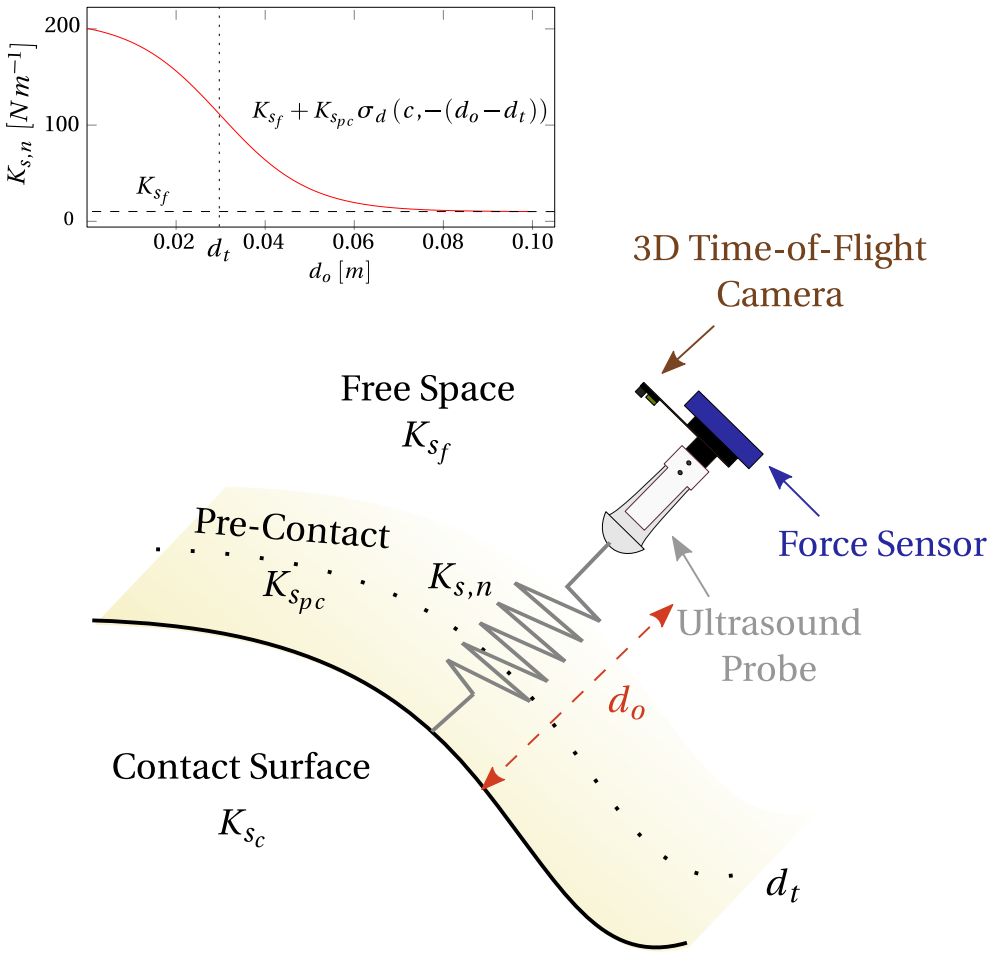


Figure 6.4— Stiffness adaptation $K_{s,n}$ towards contact. K_{s_f}, $K_{s_{pc}}$ and K_{s_c} are free space, pre-contact and contact stiffness parameters, respectively. d_o is the distance to contact given by the depth camera and d_t is a distance threshold.

compliant environment to a stiff one), introduce disruptive jerky behaviors that might turn the system unstable. Knowing the distance to contact d_o, the robot dynamic behavior can be adapted by updating $K_{s,n}$ as

$$K_{s,n} = \begin{cases} K_{s_f} + K_{s_{pc}}\sigma_d\,(c,-(d_o-d_t)) & \text{if } p_c = 0 \\ K_{s_c} & \text{if } p_c = 1 \end{cases}, \tag{6.7}$$

reducing or increasing the robot dynamics according to the distance to contact d_o. p_c is the boolean variable used to identify contact in (5.90) (and repeated

here for the reader convenience)

$$p_c = \begin{cases} 1 & \text{if } d_o < d_t \text{ and } f_{min} < \|\boldsymbol{f}_{fs}\| \\ 1 & \text{if } d_o > d_t \text{ and } f_{max} < \|\boldsymbol{f}_{fs}\| \\ 0 & \text{otherwise} \end{cases} ,$$

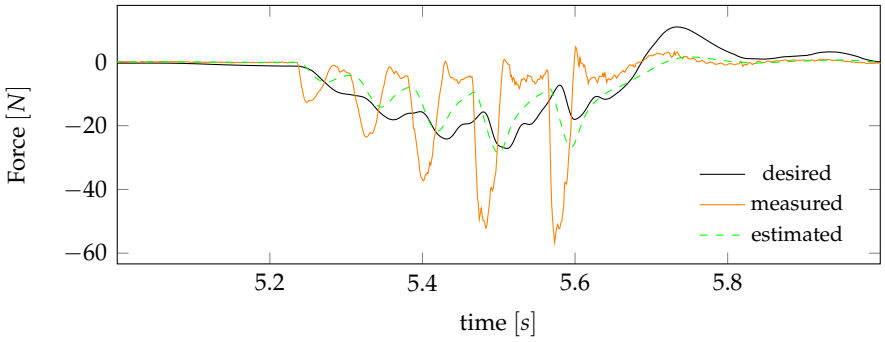
where f_{min} is a threshold value for inertial forces near contact and f_{max} is a threshold value for contact not identified by the 3D camera. σ_d is a general sigmoid function (similar to equation (5.91)) given by

$$\sigma_d\left(c, x-b\right) = \frac{1}{1+e^{-c(x-b)}} . \tag{6.8}$$

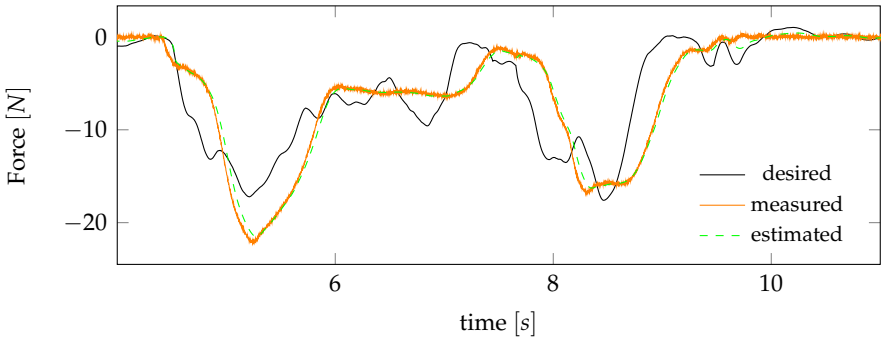
which acts as a smooth switch of x centered around b. The parameter c is a smoothing factor that defines the adaptation dynamics while d_t acts as a threshold. K_{s_f}, $K_{s_{pc}}$, and K_{s_c} are free space, pre-contact, and contact stiffness parameters, respectively. K_{s_f} drives the robot dynamics in free space, having a nominal low value so motion control performance is not affected by modeling free space as a spring. $K_{s_{pc}}$ is a pre-contact stiffness, used to reduce robot dynamics towards contact, when the inertial forces are introduced in the control loop (see Section 5.3.5). K_{s_c} is the contact stiffness. Fig. 6.4 shows the proposed approach for stiffness adaptation when moving from free space to contact.

6.3 Contact Stiffness Estimation

In unstructured environments, as the one in ultrasound examination, a precise geometrical description of the environment is unfeasible, even when d_o is available. Following the idea introduced in Cortesão et al. (2006) and in Park and Khatib (2006), the relation between desired, measured and estimated forces is used to estimate $\hat{K}_{s_c}$, reducing the mismatch between the environment stiffness K_s and the modeled one $K_{s,n}$. Let f_d, f_m and f_e be desired, measured and estimated forces along one Cartesian direction, with f_e given by the AOB first state estimate. When $K_s \gg K_{s,n}$, the controller anticipates the environment to be more compliant than it actually is. A low $K_{s,n}$ means high feedback gains, which leads to an overcompensation of force errors, leading to an underdamped response. In this situation, f_m presents an oscillatory behavior which f_e is unable to track, averaging out f_m oscillations. As a consequence,



(a)



(b)

Figure 6.5— Force control without stiffness adaptation. (a) Underestimated scenario. $K_{s,n} = 300\ N/m$ and K_s changes from free space to contact with a hard cover book ($K_s \approx 2000\ N/m$). (b) Overestimated scenario. $K_{s,n} = 6000\ N/m$ and K_s changes from free space to contact with a sponge ($K_s \approx 500\ N/m$).

the difference between f_d and f_e is relatively small when compared with the difference between f_m and f_e. Fig. 6.5a shows this behavior. On the other hand, when $K_s \ll K_{s,n}$ (see Fig. 6.5b), the environment is more compliant than expected by the controller, leading to a sluggish response since control gains are too low to overcome force errors. The force estimation is able to track measurements, however the low control gains lead to a poor f_d tracking, resulting in a small difference between f_m and f_e when compared with the difference between f_d and f_e. Based on this analysis, $\hat{K}_{s,1}$ is estimated by

$$\hat{K}^i_{s,1} = \hat{K}^{i-1}_{s_1} + \Delta\hat{K}^i_{s_1} \tag{6.9}$$

where

$$\Delta \hat{K}^i_{s,1} = \hat{K}_{s,u} + \hat{K}_{s,o} \ , \tag{6.10}$$

$$\hat{K}_{s,u} = k_u \left| f_m - f_e \right| \sigma_d \left(c_u, \ \left| f_m - f_e \right| - b_u \right) \ , \tag{6.11}$$

$$\hat{K}_{s,o} = -k_o \left| f_d - f_e \right| \sigma_d \left(c_o, \left| f_d - f_e \right| - b_o \right) \ , \tag{6.12}$$

and $\hat{K}^i_{s,1} \geq 0$. $\hat{K}_{s,u}$ corrects the errors in an underestimated stiffness scenario while $\hat{K}_{s,o}$ compensates for overestimated ones. System stability is guaranteed by the estimation algorithm design. Whenever the system presents an underdamped response, the stiffness is increased reducing the control feedback gains and stabilizing the system. k_u and k_o are algorithm gains. σ_d is a general sigmoid function, given by (6.8). b_u, b_o, c_u, c_o, k_u, and k_o are all positive parameters.

6.4 Stiffness Perception

In Cortesão et al. (2006), a second term increasing the stiffness estimation with the contact force is also included, however in all experiments the end-effector orientation is kept constant. Interestingly, we noted that, rather than changing with the contact force, the stiffness perceived by the controller changes with the end-effector posture.

Let m_j be the effective mass, the mass that is perceived at the end-effector along the j direction (Khatib (1995)))

$$\frac{1}{m_j} = \boldsymbol{J}_{p_j}(\boldsymbol{q}) \boldsymbol{B}^{-1}(\boldsymbol{q}) \boldsymbol{J}^T_{p_j}(\boldsymbol{q}) \tag{6.13}$$

where $\boldsymbol{J}_{p_j}(\boldsymbol{q}) \in \mathbb{R}^{1 \times n}$ is the linear Jacobian j-th row and $\boldsymbol{B}(\boldsymbol{q}) \in \mathbb{R}^{n \times n}$ the inertia matrix. Since both $\boldsymbol{J}_p(\boldsymbol{q})$ and $\boldsymbol{B}(\boldsymbol{q})$ change with the manipulator posture, when performing orientation motions, the inertial properties perceived at the end-effector will also change, changing the contact dynamics, which leads to different force responses from the environment. Fig. 6.6 illustrates this behavior. The WAM simulator is placed in gravity compensation in uncompressed contact with the environment, a virtual plane placed at $x_z = -0.15[m]$ with stiffness $K_{s_z} = 1000[N/m]$. A $f_{c_z} = -5[N]$ step force is applied against the virtual wall around $t \approx 1[s]$. This experiment is performed three times with the end-effector in three distinct poses, namely perpendicular to the wall (Figs. 6.6a, 6.6d and 6.6g), with 45° inclination (Figs. 6.6b, 6.6e and 6.6h) and

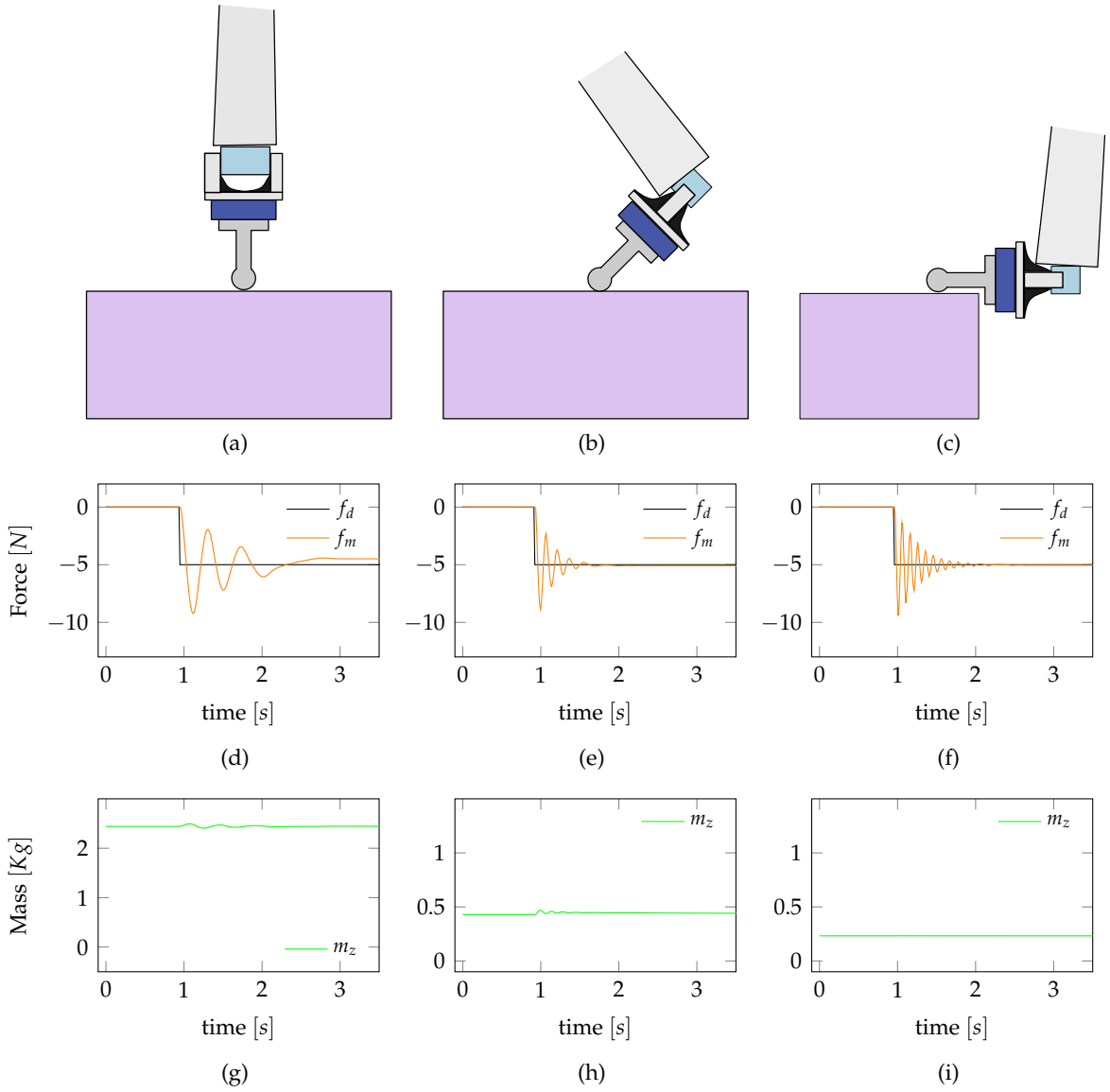


Figure 6.6— Contact force dynamics between WAM simulator and a virtual wall with the end-effector at different poses. A $f_{c_z} = -5[N]$ step force is applied around $t \approx 1[s]$ against a virtual wall plane placed at $x_z = -0.15[m]$ with stiffness $K_{s_z} = 1000[N/m]$. The interaction between robot and the virtual plane is performed with the end-effector in three different poses with respect to to the virtual wall. (a) Perpendicular contact. (b) 45° contact. (c) Parallel contact.(d), (e) and (f) Environment force response for each pose, respectively. (g), (h) and (i) Effective mass perception along the contact Cartesian direction for each pose, respectively.

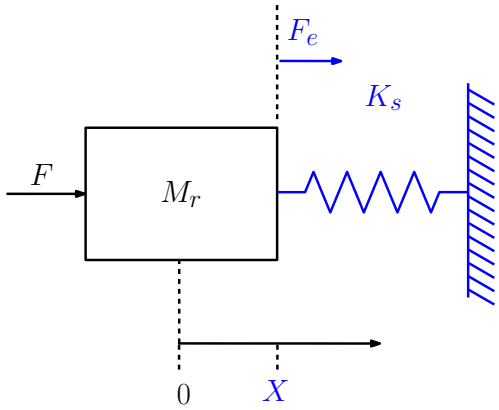


Figure 6.7— Mass-spring contact model of a 1 DoF robot.

parallel to the wall (Figs. 6.6c, 6.6f and 6.6i). The computed torque is given by

$$\tau_a = J_p^T(q) f_c + g(q) \ ,$$

with

$$f_c = \begin{bmatrix} 0 \\ 0 \\ -5 \end{bmatrix} N \ .$$

6.4.1 Non-Ideal Feedback Linearization

Under ideal conditions, when the system is linearized (see Section 3.3), the controller sees an unitary mass interacting with the environment, being the system response independent from the robot dynamic parameters. However, since a model is always an idealized, and often simplified view of a physical system, the feedback linearization may not perfectly compensate the manipulator dynamics, introducing a perturbation in the controller. Furthermore, force measurements are usually corrupted with noise, having a poor signal-to-noise ratio (SNR). A common strategy to enhance the SNR is to filter the measured force by a low pass filter (Volpe et al. (1993)). This section analyzes, through a 1-DOF example, how the introduction of a low pass filter in the feedback loop changes the force control performance and the contact dynamics.

Fig. 6.7 shows a 1-DOF robot interacting with an environment that is assumed to behave as a linear spring. According to Newton's 2^{nd} law, the equation of motion of the 1-DOF robot in contact with environment, is given

by

$$M_r\ddot{x} = f(t) - K_s x(t) \ , \tag{6.14}$$

where K_s is the environment stiffness, M_r the robot mass and $f(t)$ the applied force. $x(t)$ is the robot penetration in the environment[2]. In the Laplace domain, the contact dynamics is given by

$$\frac{X(s)}{F(s)} = \frac{1}{M_r s^2 + K_s} \ , \tag{6.15}$$

where $X(s)$ and $F(s)$ are, respectively, the Laplace domain representation of $x(t)$ and $f(t)$.

Let $F_e(s)$ be the contact force, given by

$$F_e(s) = K_s X(s) \ . \tag{6.16}$$

Solving (6.16) for $X(s)$ and replacing the result in (6.15), the open loop force response is given by

$$\frac{F_e(s)}{F(s)} = \frac{K_s}{M_r s^2 + K_s} \ . \tag{6.17}$$

Fig. 6.8 shows the 1-DOF robot interacting with the environment controlled by a force controller. Feeding back the contact force with positive feedback, the control variable $F^*(s)$ *"sees"* the robot dynamics decoupled from contact dynamics (see Fig. 6.8a and Fig. 6.8b)

$$\frac{X(s)}{F^*(s)} = \frac{1}{M_r s^2} \ . \tag{6.18}$$

If the dynamic model is known (M_r for the 1-DOF robot), it is possible to perform the system normalization, with the controller *"seeing"* the robot as an unitary mass interacting with the environment (see Fig. 6.8c)

$$\frac{X(s)}{F^*(s)} = \frac{1}{s^2} \ . \tag{6.19}$$

6.4.2 Force Feedback Filtering

Fig. 6.9a shows a generic force control with a low pass filter with ω_c cutoff frequency in the force feedback loop. When filtering the feedback force,

[2]In this section it is assumed that the robot is always in contact with the environment, i.e., no free space motion. The robot structure is regarded to be much stiffer than the environment, meaning that robot position is equal to the environment indentation.

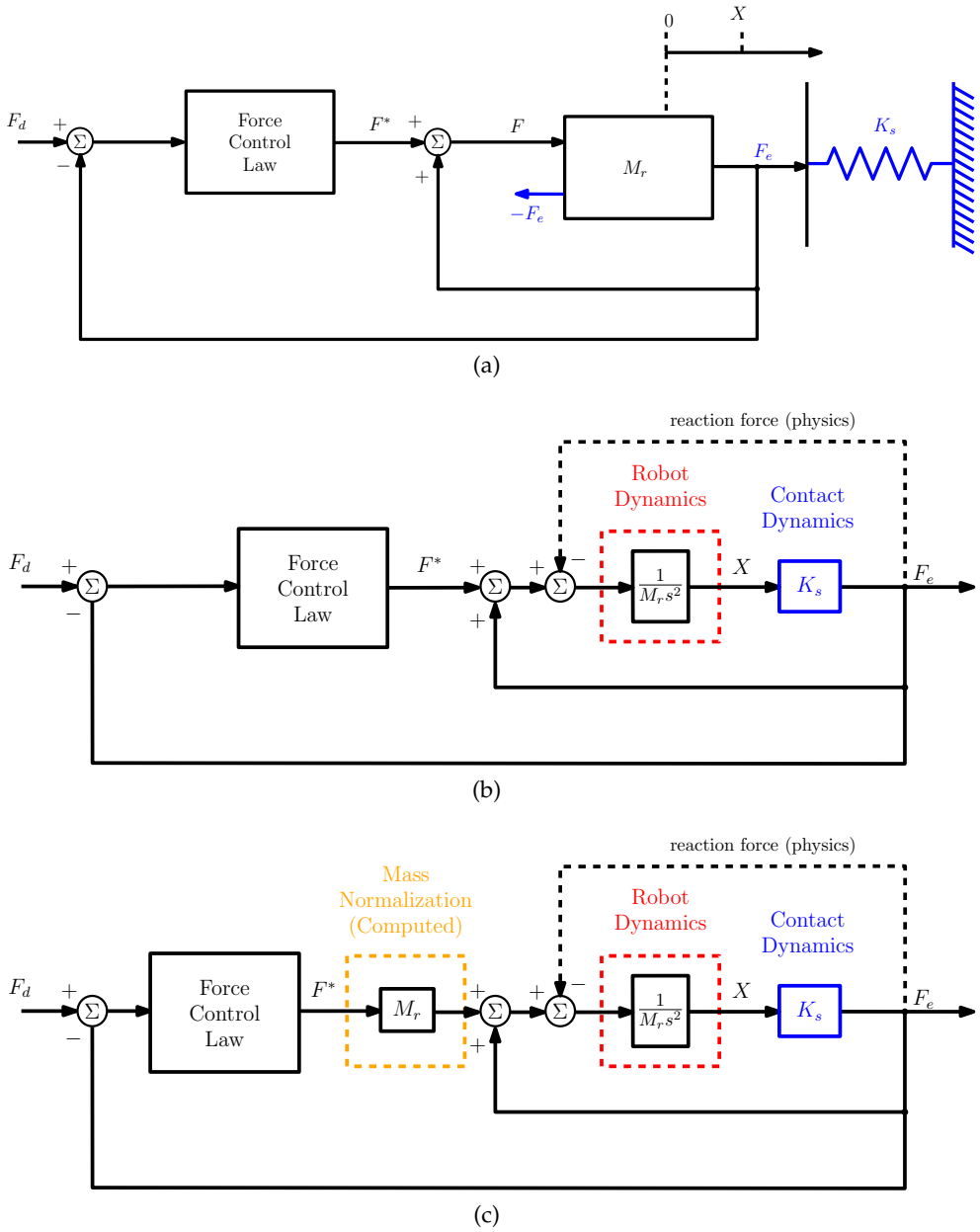
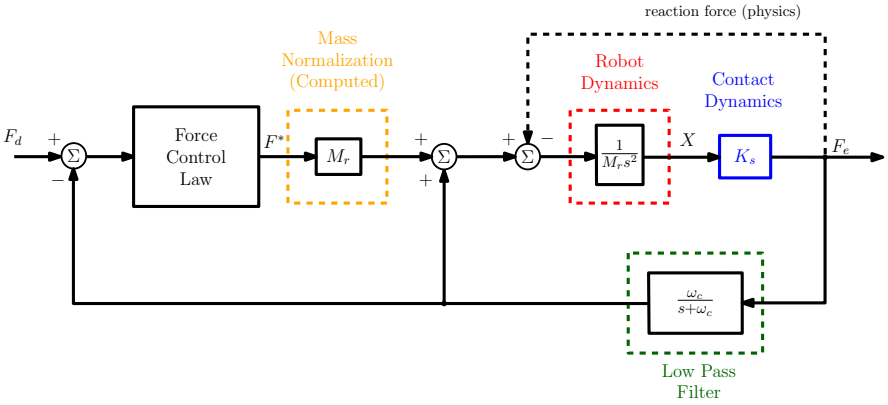
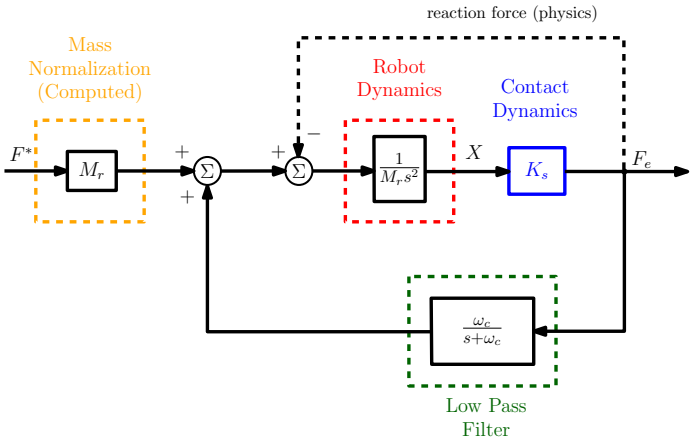


Figure 6.8— Force control with a positive feedback loop. Feeding back the contact force, the controller is able to see the robot dynamics decoupled from the contact dynamics. (a) Generic force control scheme with a positive feedback loop. (b) Simplified force control scheme. (c) Force controller with mass normalization.



(a)



(b)

Figure 6.9— Force controller with a low pass filter in the positive feedback loop. (a) Generic force control architecture with filtered feedback force. (b) System normalization with filtered feedback force.

the controller no longer *"sees"* the interaction with the environment as (see Fig. 6.8c)

$$\frac{F_e(s)}{F^*(s)} = \frac{K_s}{s^2}\,, \tag{6.20}$$

but as (see Fig. 6.9b)

$$\frac{F_e(s)}{F^*(s)} = \frac{K_s(s+\omega_c)}{s^3 + \omega_c s^2 + \frac{K_s}{M_r} s}\,, \tag{6.21}$$

highlighting that, when the contact force is filtered, the robot and the environment dynamics are no longer decoupled.

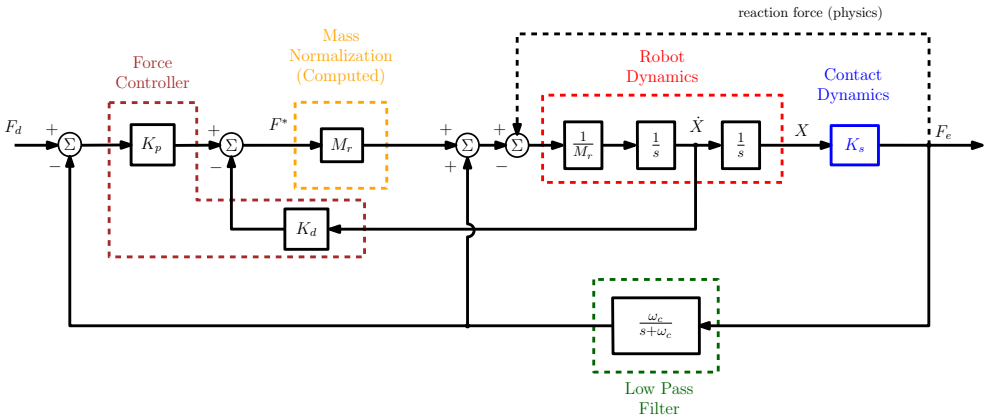


Figure 6.10— Force controller with damping action provided by velocity feedback and force feedback filtering. $F_d(s)$ is the reference force while $F_e(s)$ is the contact force.

6.4.3 Pole Placement Analysis for Different Masses

Let's assess the impact that the inertial (mass) parameters have in force control performance when filtering the force feedback (see (6.21)). The 1-DOF robot is going to be controlled in contact by a proportional force controller with damping action provided by velocity feedback (Fig. 6.10). The system transfer function is given by

$$\frac{F_e(s)}{F_d(s)} = \frac{(s+\omega_c)\,K_P K_s}{s^3 + (K_D+\omega_c)\,s^2 + \left(\frac{K_s}{M_r} + K_D\omega_c\right)s + K_P K_s \omega_c}\,. \tag{6.22}$$

It is hard to analyze how the pole placement behavior changes with the robot mass from the closed-form solution of (6.22). In this way, the system response is inferred by computing the root locus using the robot mass as the variable parameter. Figs. 6.11, 6.12 and 6.13 show the root locus when the robot mass is changed from $0.001[Kg]$ to $10[Kg]$ for three different control designs with $K_P = 1$, $K_P = 10$ and $K_P = 100$. K_D is set to assign a critically damped behavior, assuming that there is no filter in the feedback loop. The unfiltered transfer function is given by

$$\frac{F_e(s)}{F_d(s)} = \frac{K_P K_s}{s^2 + K_D s + K_P K_s}\,, \tag{6.23}$$

with K_D given by

$$K_D = 2\sqrt{K_P K_s}\,. \tag{6.24}$$

The remaining parameters are kept constant throughout the simulation. ω_c is given by

$$\omega_c = 2\pi f_c \ , \tag{6.25}$$

with the filter cutoff frequency set at $f_c = 5[Hz]$ and the environment is modeled to behave as an elastic wall with $K_s = 2000[Nm^{-1}]$.

Discussion

This experiment shows that, for systems with a low mass, increasing the control gains increases their responsiveness, while decreasing the gains for systems with a bigger mass, avoids the undesired oscillatory behavior of an underdamped response in force control tasks. For $K_P = 1$, when the mass is small, a pole near the complex plane origin dominates the system response, leading to sluggish response (see Figs. 6.11b and 6.11c). When the robot mass increases, the system dynamics becomes faster (Figs. 6.11d, 6.11e, 6.11f and 6.11g), eventually leading to an underdamped force response (Figs. 6.11h and 6.11i). For smaller masses, higher control gains improve system performance, as a higher K_P pulls the root locus to the left, improving system dynamics (Figs. 6.12b, 6.12c, 6.12d, 6.12e, 6.13b and 6.13c). For setups with large masses, increasing control gains, i.e., $K_P = 10$ and $K_P = 100$, leads to underdamped responses (Figs. 6.12f, 6.12g, 6.13d, 6.13e, 6.13f and 6.13g), while oscillatory responses from previously underdamped systems are further emphasized (Figs. 6.12h, 6.12i, 6.13h and 6.13i). As the mass increases, the system response converges to a similar result, e.g., for $K_p = 100$ the system response for $M_r = 1[Kg]$ and $M_r = 10[Kg]$ is almost identical (Figs. 6.13f, 6.13g, 6.13h and 6.13i).

To sum up, simulations show that, despite linearization, systems with different masses have different performances. For the same control design, setups with small masses display a sluggish response, while large mass setups present an overdamped behavior. It can be inferred when force feedback is filtered, to have similar force control performances with different contact postures, the robot effective mass needs to be taken into account in the perceived stiffness parameter, which in turn affects force control gains.

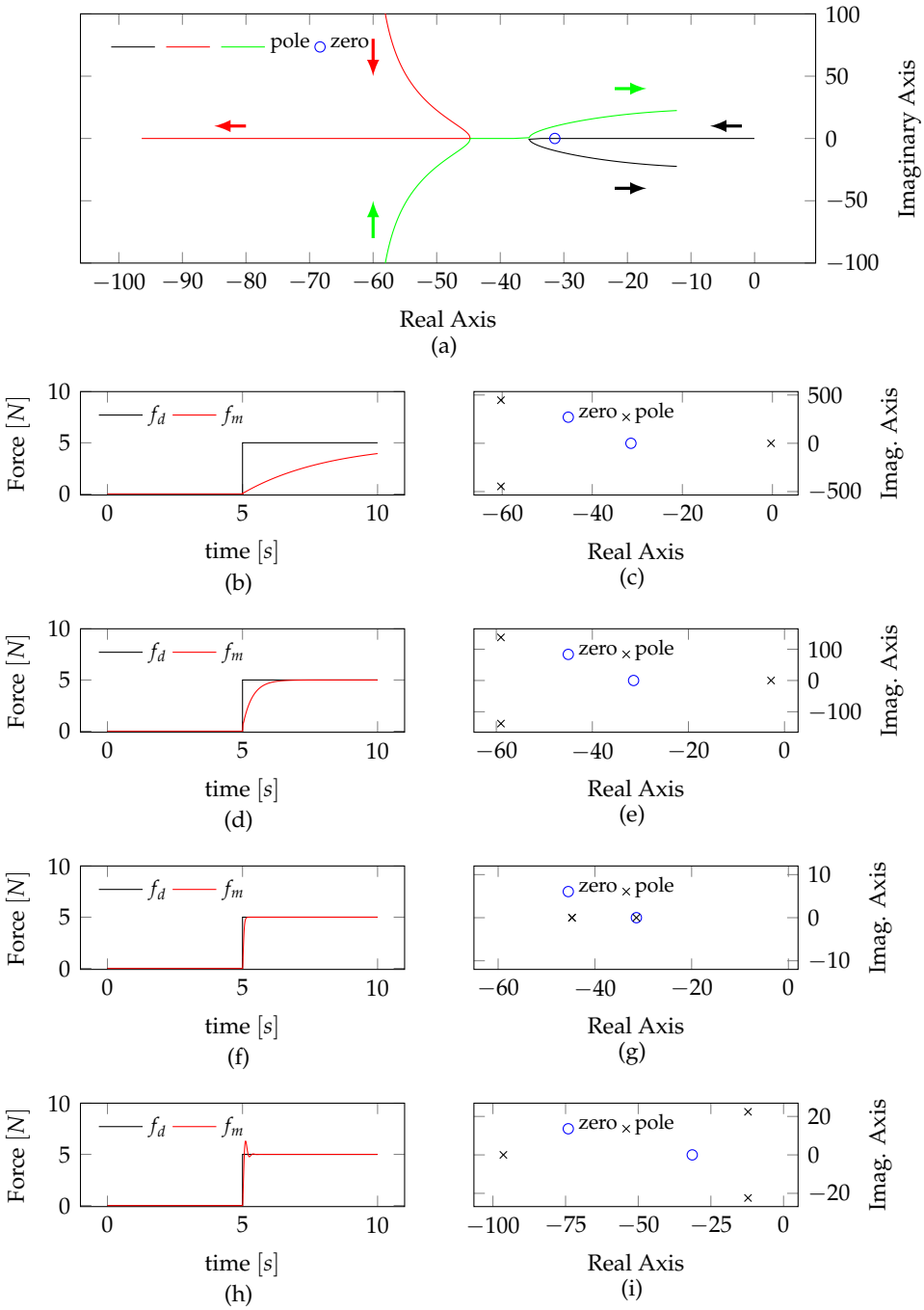


Figure 6.11— Root locus evolution for $0.001[Kg] \leq M_r \leq 10[Kg]$**,** $K_p = 1$ **and** $K_s = 2000[Nm^{-1}]$**.** (a) Root locus. The arrows point the root locus evolution as the mass increases from $0.001[Kg]$ to $10[Kg]$. For low M_r, the non-dominant complex pole pairs have a high imaginary component and are not represented in the plot. (b), (d), (f), and (h) show a $f_d = 5[N]$ step response for $M_r = 0.01, 0.1, 1$ and $10[Kg]$, respectively. f_d and f_m are reference and applied forces, respectively. (c), (e), (g) and (i) show poles and zero locations of the closed loop system for $M_r = 0.01, 0.1, 1$ and $10[Kg]$, respectively.

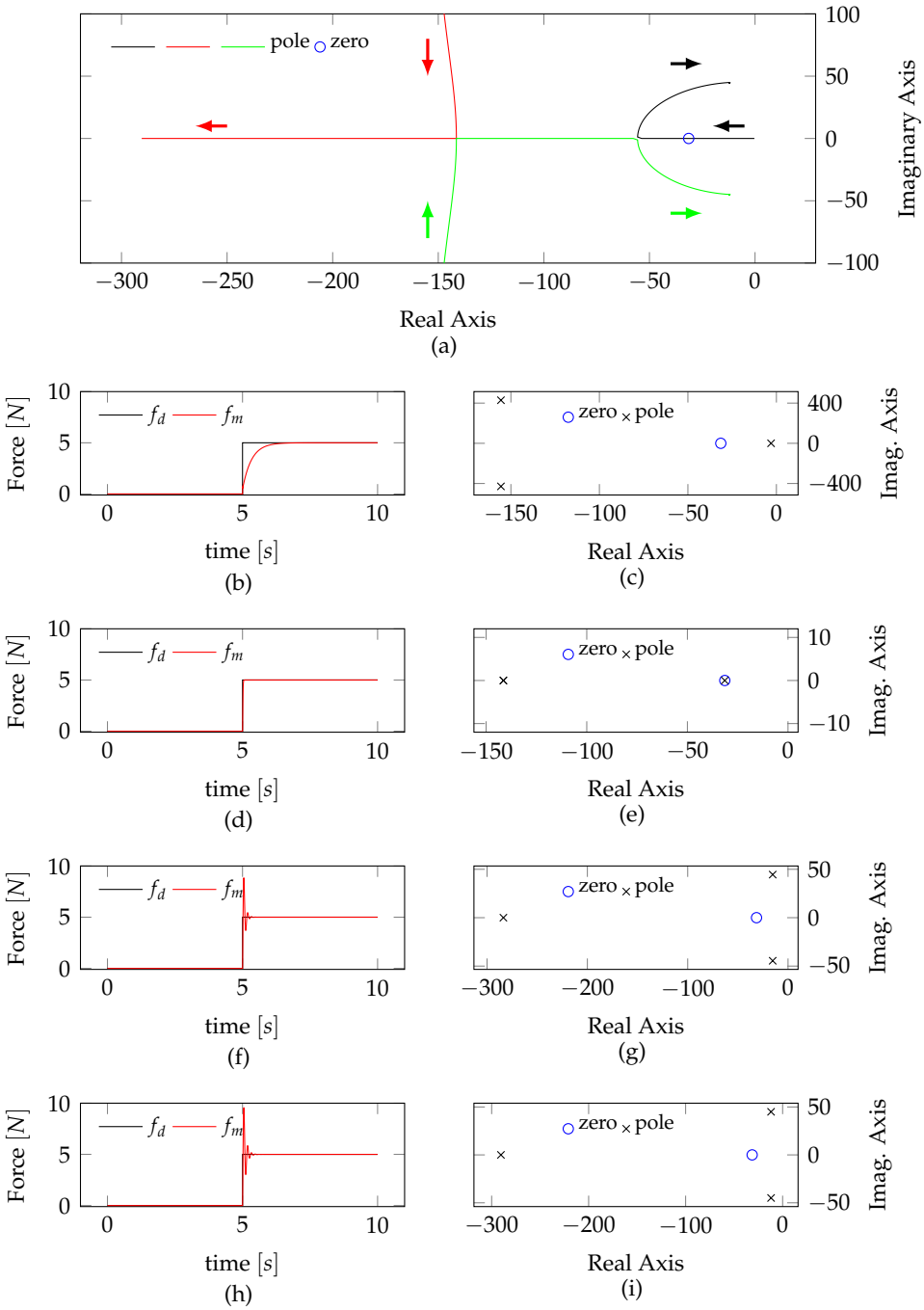


Figure 6.12— Root locus evolution for $0.001[Kg] \leq M_r \leq 10[Kg]$**,** $K_p = 10$ **and** $K_s = 2000[Nm^{-1}]$**.** (a) Root locus. The arrows point the root locus evolution as the mass increases from $0.001[Kg]$ to $10[Kg]$. For low M_r, the non-dominant complex pole pairs have a high imaginary component and are not represented in the plot. (b), (d), (f), and (h) show a $f_d = 5[N]$ step response for $M_r = 0.01, 0.1, 1$ and $10[Kg]$, respectively. f_d and f_m are reference and applied forces, respectively. (c), (e), (g) and (i) show poles and zero locations of the closed loop system for $M_r = 0.01, 0.1, 1$ and $10[Kg]$, respectively.

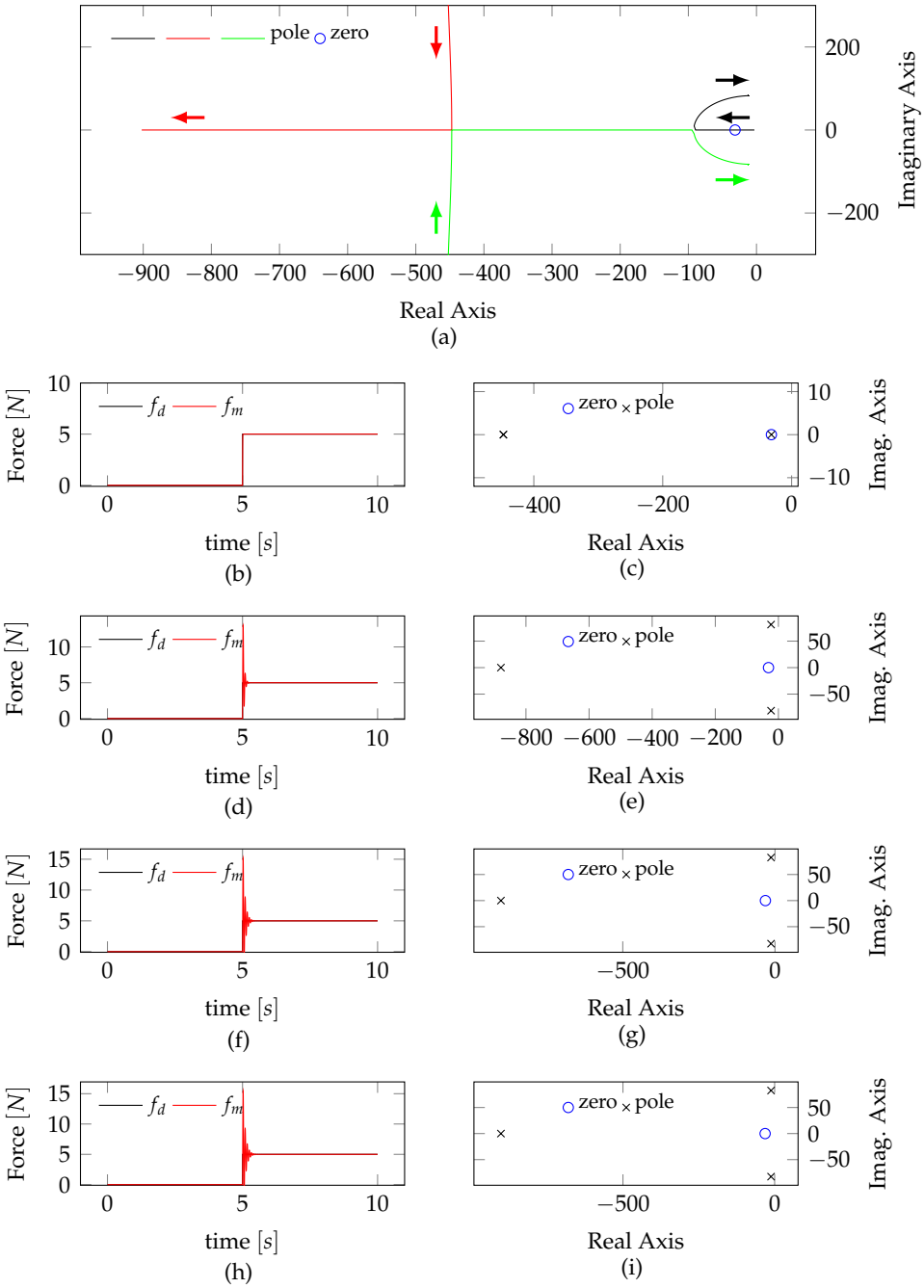


Figure 6.13— Root locus evolution for $0.001[Kg] \leq M_r \leq 10[Kg]$**,** $K_p = 100$ **and** $K_s = 2000[Nm^{-1}]$**.** (a) Root locus. The arrows point the root locus evolution as the mass increases from $0.001[Kg]$ to $10[Kg]$. For low M_r, the non-dominant complex pole pairs have a high imaginary component and are not represented in the plot. (b), (d), (f), and (h) show a $f_d = 5[N]$ step response for $M_r = 0.01, 0.1, 1$ and $10[Kg]$, respectively. f_d and f_m are reference and applied forces, respectively. (c), (e), (g) and (i) show poles and zero locations of the closed loop system for $M_r = 0.01, 0.1, 1$ and $10[Kg]$, respectively.

6.4.4 Effective Mass and Perceived Stiffness

Since the stiffness estimation strategy is based on the transitory behavior of force relations (see (6.11) and (6.12)), the estimation algorithm *"sees"* the drop off in the force tracking performance as a modification in contact parameters, changing the environment estimation accordingly, even though the environment physical properties remain unchanged. In this way, the perceived stiffness can be seen, not as a well defined physical property, but as an optimization parameter to improve force control performance. When the effective mass decreases, the pole near the origin due to the low pass filter, dominates the system response, leading to a sluggish response. The estimation algorithm reacts to this slow response, decreasing the stiffness estimation value, since the estimator sees the sluggish response as an interaction with a softer environment, which requires higher control gains to track the force reference. Similarly, when the effective mass increases, the system response is dominated by a complex pole pair, leading to an underdamped response. The stiffness estimator reacts to the oscillatory behavior, decreasing control gains by increasing the stiffness estimation. Using this information, the perceived stiffness can be anticipated by introducing a second term in the estimation algorithm that updates the estimation based on the end-effector effective mass

$$\hat{K}^i_{s,2} = k_m \, \lambda_j \, . \tag{6.26}$$

λ_j is the j^{th} main diagonal element of the end-effector inertia matrix $\boldsymbol{\Lambda}_p(\boldsymbol{q})$, given by

$$\boldsymbol{\Lambda}_p(\boldsymbol{q}) = \left(\boldsymbol{J}_p(\boldsymbol{q}) \boldsymbol{B}^{-1}(\boldsymbol{q}) \boldsymbol{J}_p^T(\boldsymbol{q}) \right)^{-1} \, , \tag{6.27}$$

with $\boldsymbol{J}_p(\boldsymbol{q}) \in \mathbb{R}^{3\times n}$ the linear Jacobian. λ_j is closely related to the effective mass m_j (6.13), presenting slower dynamics which can be seen as a "filtered version" of the effective mass. Therefore, we propose to estimate the contact stiffness as the contribution of two terms

$$\hat{K}_{s_c} = \hat{K}^i_{s,1} + \hat{K}^i_{s,2} \, . \tag{6.28}$$

The first one, given by (6.9) is computed taking into account the relation between desired, measured and estimated forces, while the second term, given by (6.26), takes into account the effective mass in the estimation. To

Table 6.1: Online Stiffness Estimation Design Parameters.

b_u	0.7	c_u	9.0	k_u	7.0	$[m^{-1}]$
b_o	1.0	c_o	7.0	k_o	3.0	$[m^{-1}]$
				k_m	500	$[NKg^{-1}m^{-1}]$

avoid jerky behaviors K_{s_c} in (6.7) is given by

$$K_{s_c} = \begin{cases} \hat{K}_{s_c} & \text{if } \hat{K}_{s_c} \geq K_{s_{c,min}} \\ K_{s_{c,min}} & \text{if } \hat{K}_{s_c} < K_{s_{c,min}} \end{cases} , \tag{6.29}$$

where $K_{s_{c,min}}$ is the minimum contact stiffness to limit control gains in contact. It is given by

$$K_{s_{c,min}} = K_{s_f} + K_{s_{pc}} . \tag{6.30}$$

Low pass filters are used in $K_{s,n}$ and in d_o computation, with the filter equations given by

$$K_{s,n}^{f,i} = K_{s,n}^{f,i-1} + l_{s,n} \left(K_{s,n}^{i} - K_{s,n}^{f,i-1} \right) , \tag{6.31}$$

$$d_o^{f,i} = d_o^{f,i-1} + l_o \left(d_o^{i} - d_o^{f,i-1} \right) , \tag{6.32}$$

where the superscript i indicates the iteration and f indicates the filtered value.

6.5 Experiments

Two experiments were conducted to assess the stiffness estimation algorithm. In the first experiment, the relation between the inertial properties at the end-effector and the contact stiffness K_{s_c} perceived by the controller is emphasized, while in the second experiment, the estimation algorithm performance is compared with the approach presented in Cortesão et al. (2006) in two distinct contact postures, where end-effector inertial properties are different. $\boldsymbol{f}_m$ is filtered with a first order low pass Butterworth filter with a $5[Hz]$ cutoff frequency.

6.5.1 Stiffness Estimation Algorithm Design

Table 6.1 gives the parameters for $\hat{K}_{s_c}$ estimation. Since the underdamped response is critical for system stability, the estimation algorithm is designed to

Table 6.2: Numerical Values of $K_{s,n}$ Design Parameters.

c	100	d_t	0.03	$[m]$	K_{s_f}	10	$[Nm^{-1}]$
$k_{v_{pc}}$	4.0	f_{min}	2.0	$[N]$	$K_{s_{pc}}$	200	$[Nm^{-1}]$
$l_{s,n}$	0.05	f_{max}	4.0	$[N]$			
l_o	0.05						

react faster in stiffness underestimated scenarios than in overestimated ones ($k_u > k_o$, $c_u > c_o$ and $b_u < b_o$). In underestimate scenarios, control gains are quickly reduced, keeping system stability, while in overestimated ones, the stiffness decreases more slowly, avoiding jerky behaviors. Table 6.2 gives $K_{s,n}$ parameters.

3D Camera Design

The camera integration time is set to $1000[\mu s]$ and the amplitude threshold is set to 4000. This setup provides environment description without probe detection. We have used manufacturer calibration data.

6.5.2 Perceived Stiffness Estimation Experimental Validation

The first experiment can be summarized as follows. The probe motion starts in free space with a vertical posture, reaching contact with a stiff surface (hard cover book, see Fig. 6.14a) around $12[s]$. While in contact, the probe rotates along a perpendicular axis, from the initial contact posture to an horizontal one ($20[s] - 30[s]$, see Fig. 6.14b) and then back to the initial orientation ($35[s] - 50[s]$). $\hat{K}_{s_c}$ is only estimated by force relations (see (6.9)), and the reference force is kept constant during the experiment. The overall motion is performed with slow dynamics to minimize disturbances in force control and, consequently, in the stiffness estimation, highlighting the relation between perceived stiffness and effective mass. This experimental procedure is repeated in interaction with a virtual wall along the Z plane ($p_z = -0.15[m]$, parallel to the robot base) with $K_{s_z} = 2500[Nm^{-1}]$ (from a vertical to a horizontal posture $7[s] - 25[s]$ and then back to the initial contact posture $30[s] - 50[s]$).

The second experiment can be described in the following way. The probe motion starts in free space, reaching contact with the same stiff surface for two times, first with a vertical posture (high effective mass) ($7[s] - 19[s]$, Fig. 6.14a)

and then with an horizontal one (low effective mass, $32[s] - 50[s]$, Fig. 6.14b). The probe posture and the reference force are kept constant in both contact scenarios. Between contacts, the orientation is changed in free space to the second contact posture. The experiment is performed two times, first using the stiffness estimation approach proposed in Cortesão et al. (2006) and then repeated using the estimation algorithm presented in this paper. The second term of (6.28) is computed in Cortesão et al. (2006) by

$$\hat{K}^i_{s,2} = k_3 \sigma_d \left(c_0, |f_m| - f_0 \right) \ , \tag{6.33}$$

increasing the stiffness estimation with the measured force f_m. $k_3 = 3000[Nm^{-1}]$, $c_0 = 0.2$ and $f_0 = 20[N]$ are set according to Cortesão et al. (2006). σ_d is given by (6.8). In both experiments the contact surface is placed parallel to the robot base.

6.5.3 Discussion

Fig. 6.14 shows the result of the first experiment in interaction with the hard cover book while Fig. 6.15 shows the experimental results of the interaction with a virtual wall. The reference force is well tracked in both runs, validating the stiffness estimation throughout the experiment (see Figs. 6.14c and 6.15a). From Fig. 6.14d, a connection between the effective mass and the perceived stiffness can be inferred. As can be seen, excluding the transitory behavior from free space to contact, the stiffness estimation and the effective mass curve have a similar shape ($13[s] - 54[s]$) in interaction with the book, while the contact force seems to have no effect in the stiffness estimation. Clearly, the physical properties of the book do not change throughout the experiment, however from a control standpoint, the environment is differently perceived when the end-effector posture changes. In Fig. 6.14d it is also depicted the evolution of λ_z, the last element of $\mathbf{\Lambda}_p(\boldsymbol{q})$ main diagonal. While the effective mass evolution closely matches the stiffness estimation evolution, λ_z displays slower dynamics (w.r.t. m_z), being more adequate for control adaptation. In this way, λ_z can be used to anticipate the perceived stiffness, improving system performance in dynamic interactions. Fig. 6.15b shows the connection between λ_z and the stiffness estimation ($4[s] - 50[s]$) in interaction with the virtual wall.

Fig. 6.16 shows the results of the second experiment. When the stiffness is estimated by Cortesão et al. (2006), the system displays two distinct behaviors,

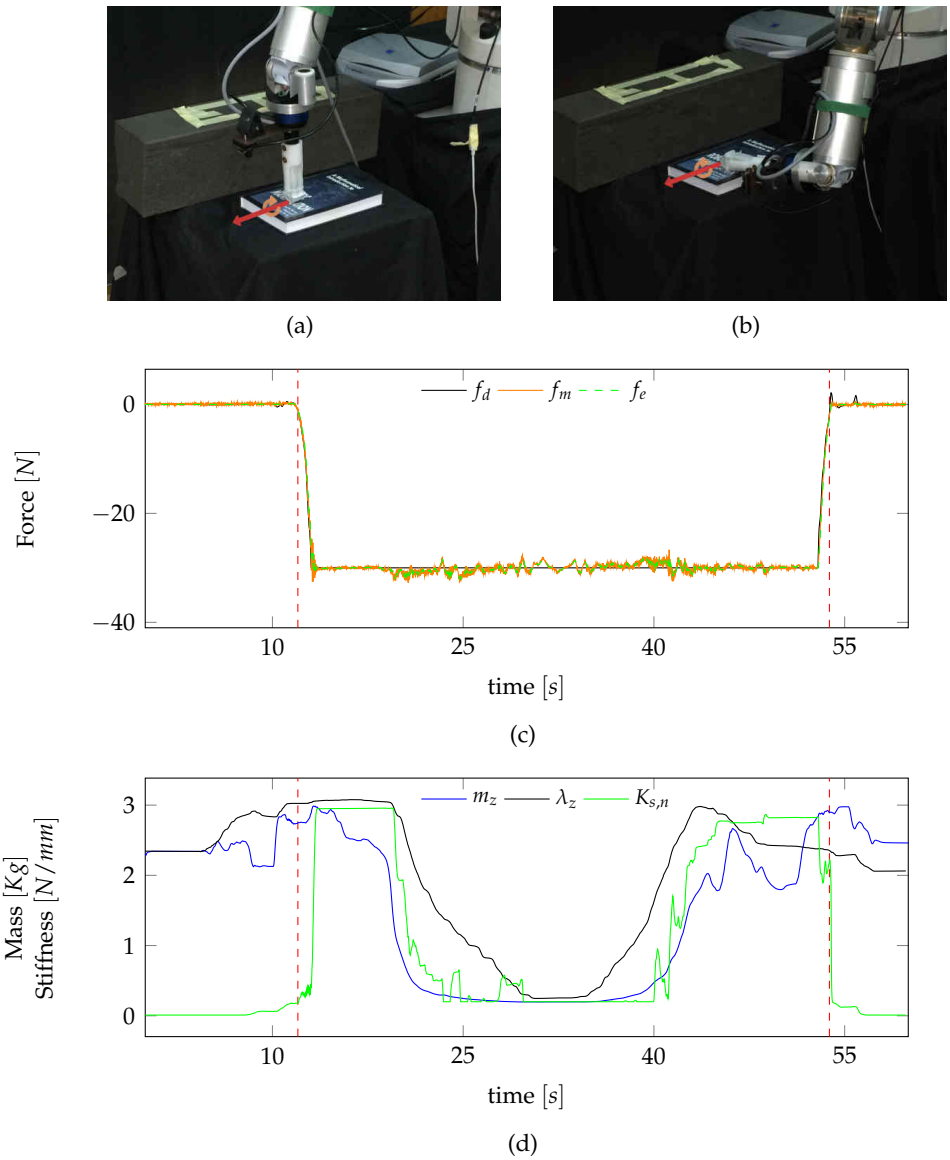


Figure 6.14— Stiffness perception by the controller when the end-effector posture changes constrained by a stiff surface (book). (a) and (b) Probe at vertical and horizontal posture, respectively. (c) Force tracking in Z Cartesian dimension with online stiffness estimation (only by (6.9)) and control adaptation. f_d, f_m and f_e are desired, measured and estimated forces, respectively. (d) Effective mass m_z, $\mathbf{\Lambda}_p(\boldsymbol{q})$ main diagonal element λ_z and stiffness estimation $K_{s,n}$ in Z Cartesian dimension. To highlight the connection between m_z, λ_z and $K_{s,n}$, $K_{s,n}$ is plotted in Newton per millimeter. Desired force and probe position are kept constant through the constrained orientation motion. Red dashed vertical lines identify free space to contact and vice-versa transitions.

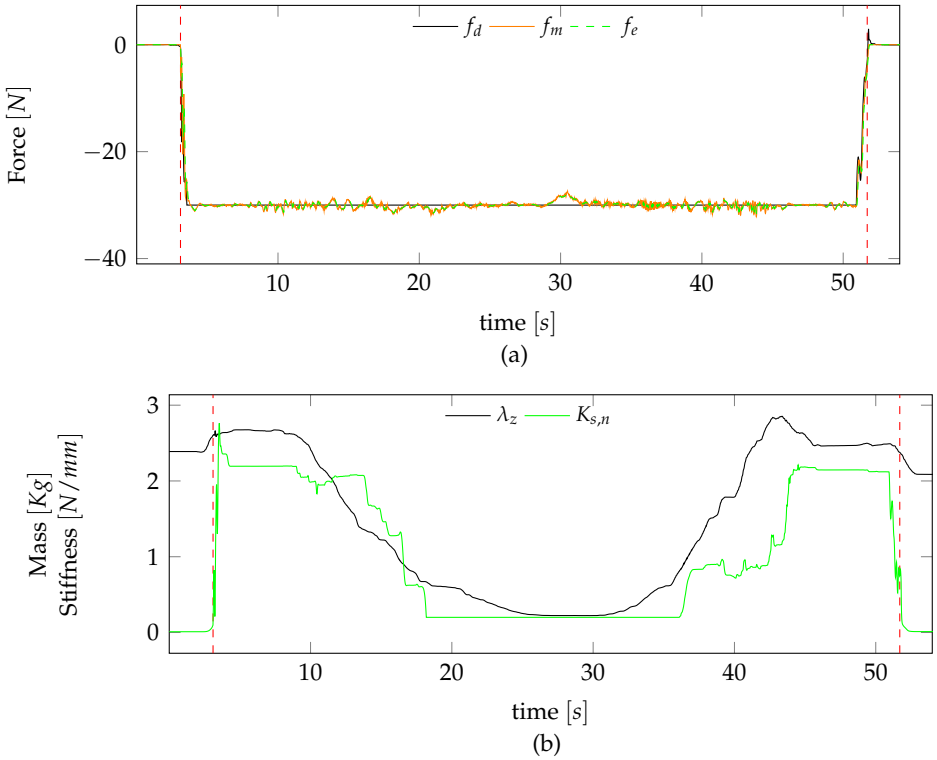
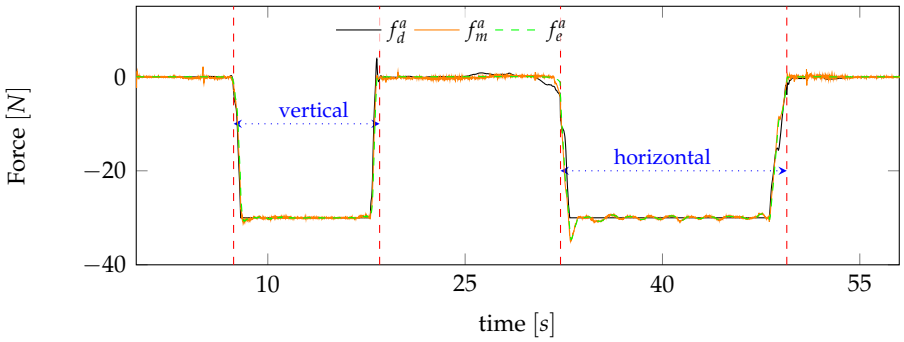


Figure 6.15— Stiffness perception by the controller when the end-effector posture changes constrained by a virtual wall along Z with stiffness $K_s = 2500Nm^{-1}$. (a) Force tracking with online stiffness estimation (only by (6.9)) and control adaptation. f_d, f_m and f_e are desired, measured and estimated forces, respectively. (b) $\mathbf{\Lambda}_p(\boldsymbol{q})$ main diagonal element λ_z and perceived stiffness estimation $K_{s,n}$. To highlight the connection between both variables, $K_{s,n}$ is plotted in Newton per millimeter. Desired force and probe position are kept constant through the constrained orientation motion. Red dashed vertical lines identify free space to contact and vice-versa transitions.

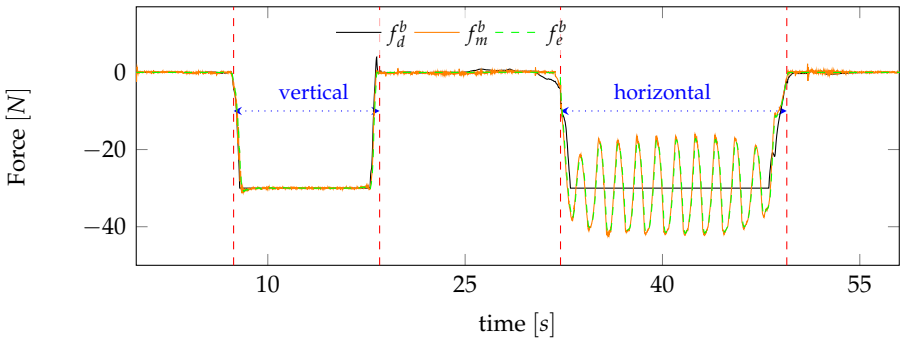
according to how the effective mass is perceived at end-effector. When the effective mass is high (Fig. 6.16e), f_d is effectively tracked, validating the stiffness estimation. However, when the effective mass is low, f_m oscillates around the reference force, leading to a poor f_d tracking performance. The relation between f_m, f_e and f_d in the second contact, shows a typical stiffness overestimation behavior (low control gains), with f_e accurately estimating f_m, but unable to track f_d (Fig. 6.16b). Due to low control gains (see Fig. 6.16f), most of the control effort comes from the AOB active state p_k (see Fig. 6.16d), which is not enough to tackle the overestimation disturbance, leading to f_m oscillatory behavior. Since $\hat{K}_{s_c}$ is driven by f_m (see (6.33)), its oscillatory behavior leads to the stiffness oscillations shown in Fig. 6.16c.

On the other hand, when using the effective mass in the estimation algorithm, the reference force is well tracked in both contact postures (Fig. 6.16a), validating the estimation approach. The environment is differently perceived by the controller (Fig. 6.16c), despite the constant contact force, which shows that f_m absolute value *per se* has no influence in $\hat{K}_{s_c}$ estimation (see (6.33)). In the second contact, excluding the transitory behavior from free space to contact and vice-versa, the estimation algorithm keeps the stiffness value constant at $K_{s_{c,min}}$, stressing that to track f_d when the effective mass is low, high control gains are required, i.e., when the effective mass changes, also the contact dynamics and the controller perception of the environment changes.

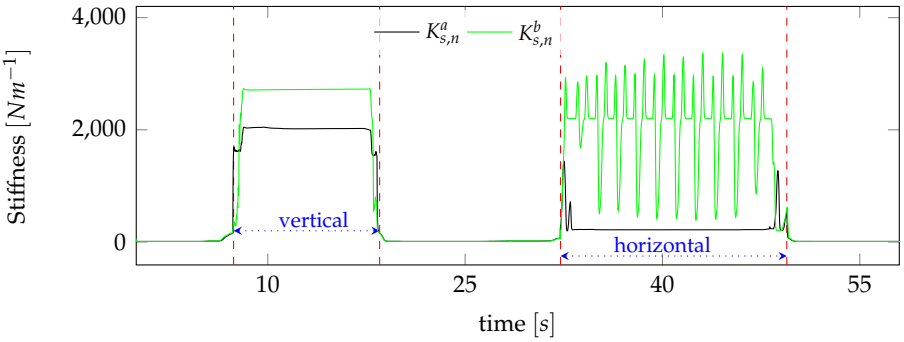
Furthermore, the AOB active state evolution (Fig. 6.16d) shows that, while in the first contact the active state has little influence in the control command, in the second contact, a significant amount of the control effort comes from p_k, meaning that the environment model is less precise and the model error is tackled by the AOB active state. Another important remark is that, from a control standpoint, a precise estimation is not critical for a good system performance, since in the first contact, f_d is effectively tracked with both estimation algorithms despite the estimation numerical value be somewhat different (around $500[Nm^{-1}]$).



(a)

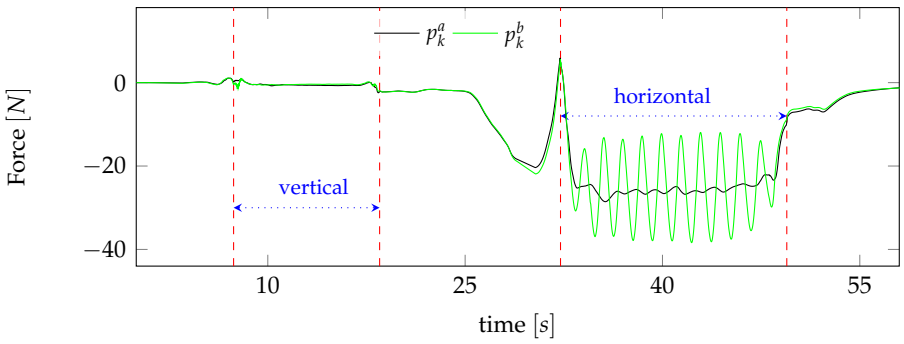


(b)

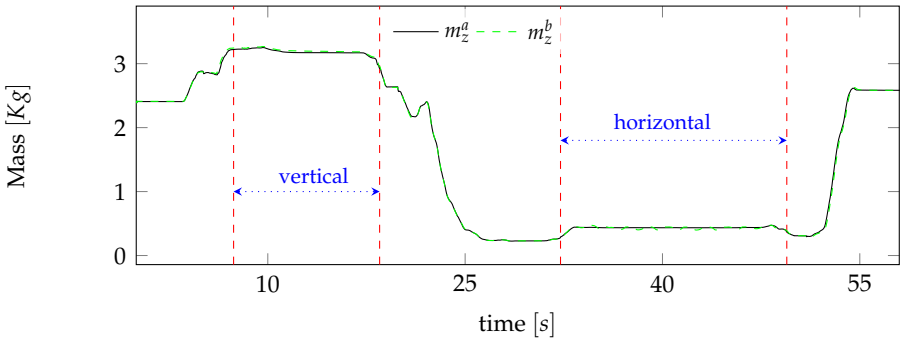


(c)

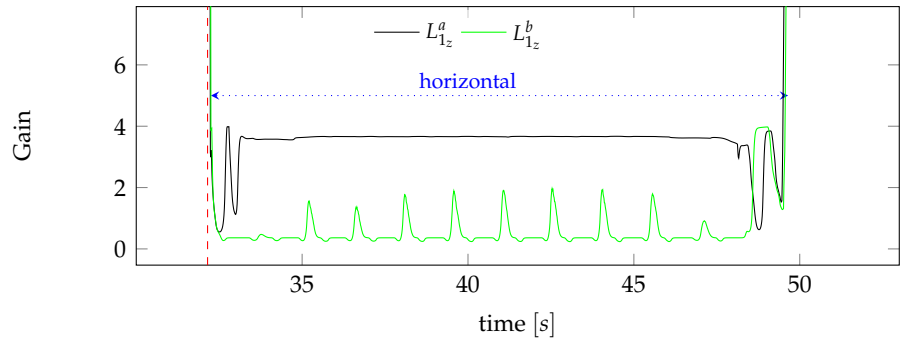
Figure 6.16— Experimental results of the second experiment. The performance of the stiffness estimator (identified by the superscript ^{a}) is compared with the estimator presented in Cortesão et al. (2006) (superscript ^{b}) in two distinct end-effector contact postures (vertical and horizontal contact posture). (a) and (b) Force tracking in Z Cartesian dimension with online stiffness estimation and control adaptation. f_d, f_m and f_e are desired, measured and estimated forces, respectively. (c) Stiffness estimation $K_{s,n}$ through the experiment.



(d)



(e)



(f)

Figure 6.16— Experimental results of the second experiment (continuation). The performance of the stiffness estimator (identified by the superscript ^{a}) is compared with the estimator presented in Cortesão et al. (2006) (superscript ^{b}) in two distinct end-effector contact postures (vertical and horizontal contact posture). (d) AOB active state p_k evolution. (e) Effective mass perception at end-effector along the Z Cartesian dimension. (f) First element of Z Cartesian dimension state feedback gain L_{1_z} in the second contact. Red dashed vertical lines identify free space to contact and vice-versa transitions, while blue dotted horizontal lines identify the end-effector contact posture.

6.6 Conclusion

This chapter presented the stiffness estimation strategy. Since the force control architecture is kept even in free space, the nominal stiffness value should be in accordance with the motion type: a low nominal value in free space to limit disturbances due to modeling free space as a spring; and an estimation of the surface stiffness when in contact. When moving from free space to constrained motion or vice-versa, specially when interacting with stiff surfaces, undesired behaviors arise due the abrupt change in control gains. To cope with this issue, motion and contact dynamics are driven by a 3D time-of-flight camera and a force sensing. Increasing nominal stiffness towards contact, allows to decrease robot motion dynamics, keeping the system stable when inertial forces disturb the control loop. In contact, the stiffness estimation is based on the transient relation between desired, measured and estimated force, and the manipulator inertial properties referred to the end-effector. It has been shown that when filtering contact force measurements, the inverse dynamics compensation with positive feedback loop no longer decouples the robot from environment dynamics. Simulation results show that force control performance changes with inertial parameters, requiring higher control gains for lower masses and lower control gains for setups with a larger mass. Using this information, the stiffness estimation is updated proportionally to the effective mass, improving force control performance. Two set of experiments validate the stiffness estimation algorithm, emphasizing the relation between effective mass and perceived stiffness.

CHAPTER 7

ROBOTIC-ASSISTED TELE-ECHOGRAPHY EXPERIMENTS

This chapter describes the teleoperation architecture and assesses the hierarchical control architecture in ultrasound teleoperation scenarios. Clinical and non-clinical experiments are going to be performed. In the non-clinical one, the control architecture is assessed in a testbed scenario where a hard cover book and a sponge are used for stiff (similar to bone contact) and soft contact assessment (similar to an abdominal ultrasound examination). In the clinical experiment, a physician performs a robotic-assisted tele-echography examination on a healthy volunteer to validate the system in a medical context. This chapter is organized as follows. The teleoperation architecture is described in Section 7.1. Section 7.2 presents the control design. Testbed experimental results are discussed in Section 7.3 while Section 7.4 presents the experimental results of a clinical examination in a healthy volunteer. Section 7.5 concludes the chapter.

7.1 Teleoperation Architecture

The teleoperation architecture (Fig. 7.1) follows a position-position approach similar to the teleoperation architectures proposed by Cortesão et al. (2006) and Park and Khatib (2006). Let $G_m(s)$ and $G_{sp}(s)$ be master and slave transfer functions respectively, with β_f and β_p force and position scaling factors. The input to both master and slave stations is the desired force f_d, computed from position errors through virtual coupling K_{virt} (see (5.34)). The slave station

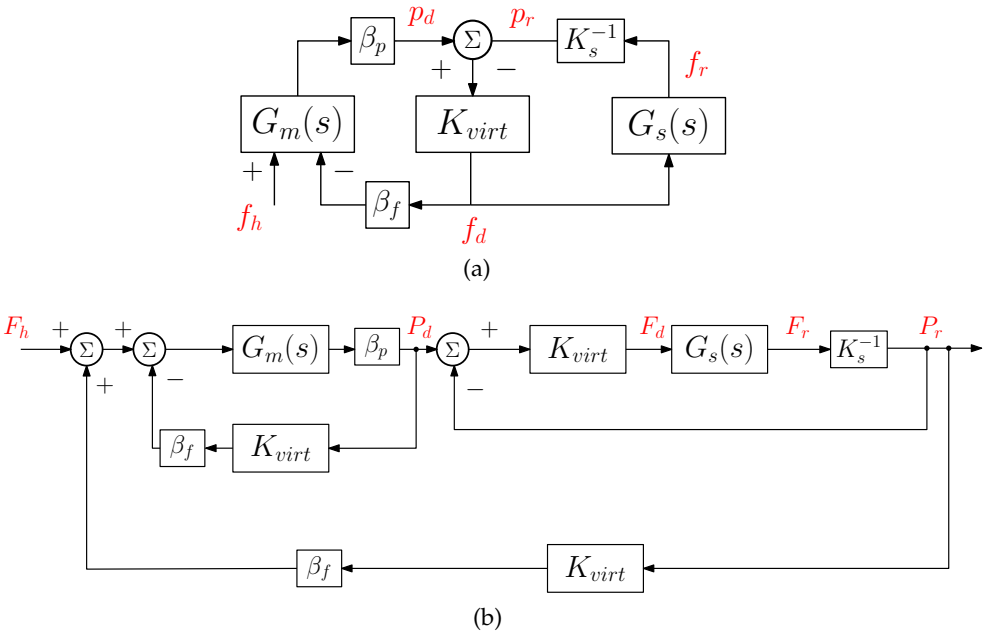


Figure 7.1— Teleoperation architecture. a) Teleoperation scheme. b) Block diagram rearranged to better highlight the local feedback in each system. β_f and β_p are force and position scaling factors, respectively. Throughout the experiments $\beta_f = 0.1$ and $\beta_p = 1$.

transfer function (including manipulator, force control and environment) from the force input f_d to the manipulator motion p_r is given by

$$G_{sp}(s) = \frac{P_r(s)}{F_d(s)} = \frac{1}{K_s} G_s(s) \ , \tag{7.1}$$

with

$$P_r(s) = \frac{F_r(s)}{K_s} \ . \tag{7.2}$$

$G_s(s)$ is the transfer function of the force controlled probe holder manipulator. In contact, the desired closed loop response is given by

$$G_s(s) = \frac{F_r(s)}{F_d(s)} \approx \frac{\left(\frac{1}{\tau_f}\right)^2}{\left(s + \frac{1}{\tau_f}\right)^2} \ , \tag{7.3}$$

with τ_f given by (5.68). The master station includes the haptic device and the human arm[1] and can be modeled as a mass-spring-damper system (Yokokohji

[1]There is no force controller at master station. This approach can only be applied to light-weight frictionless master stations.

and Yoshikawa (1994))

$$G_m(s) = \frac{1}{M_m s^2 + D_m s + K_m} \,, \tag{7.4}$$

where M_m, D_m and K_m are mass, damping and stiffness of the master station, respectively. K_m is mainly due to the human arm stiffness, being assumed that the operator presents a compliant behavior in contact (low K_m).

7.1.1 Telepresence

Lawrence (1993) states that transparency (i.e., telepresence) is achieved when the transfer function from a force input applied by the human operator ($F_h(s)$) to the master device position output ($P_d(s)$), matches the compliance of the slave system in contact

$$\frac{P_d(s)}{F_h(s)} = \frac{1}{M_m s^2 + D_m s + (K_m + \alpha K_s)} \,, \tag{7.5}$$

where α is a scaling factor to enhance environment perception ($\alpha > 0$). The transfer function of our teleoperation architecture is

$$\begin{aligned} \frac{P_d(s)}{F_h(s)} &= \frac{G_m(s)}{1 + \frac{G_m(s) K_{virt} \beta_p \beta_f}{1 + K_{virt} G_s(s) K_s^{-1}}} \\ &= \frac{\frac{1}{M_m s^2 + D_m s + K_m}}{1 + \frac{K_{virt} \beta_p \beta_f}{\left(1 + K_{virt} G_s(s) K_s^{-1}\right)(M_m s^2 + D_m s + K_m)}} \end{aligned} \,. \tag{7.6}$$

Fig. 7.1b shows a rearranged block diagram of Fig. 7.1a.

Low Frequency Motion

The analytical analysis of (7.6) is difficult since it depends on the human arm parameters, which are difficult to quantify and may vary during task execution (Cortesão et al. (2006)). However, for low frequency motions in contact ($s \to 0$, typically the motions present in ultrasound examinations), $G_s(s) \approx 1$ and (7.6) can be approximated by

$$\frac{P_d(s)}{F_h(s)} \approx \frac{1}{K_m \left(1 + \frac{K_{virt} \beta_p \beta_f}{K_m \left(1 + K_{virt} K_s^{-1}\right)}\right)} \tag{7.7}$$

If $K_{virt} \gg K_s$, (7.7) becomes

$$\frac{P_d(s)}{F_h(s)} \approx \frac{1}{K_m + K_s \beta_p \beta_f} , \tag{7.8}$$

which has the same form of (7.5) when $s \to 0$, meaning that telepresence is achieved. If $K_{virt} \ll K_s$

$$\frac{P_d(s)}{F_h(s)} \approx \frac{1}{K_m + K_{virt} \beta_p \beta_f} , \tag{7.9}$$

there is no telepresence, since the physician only feels K_{virt}.

Virtual Coupling Computation

Equations (7.8) and (7.9) show that the relation between K_{virt} and K_s is crucial to enable haptic perception. Keeping $K_{virt} \propto K_{s,n}$ improves telepresence in contact. Furthermore, K_{virt} design is also critical to enhance free space position tracking. We propose to design K_{virt} as

$$K_{virt} = \begin{cases} K_{virt_f} + k_{v_{pc}} K_{s_{pc}} \sigma_d \left(c, -\left(d_o - d_t\right)\right) & \text{if } p_c = 0 \\ K_{virt_f} + k_{v_{pc}} K_{s_{pc}} + \hat{K}_{s,2} & \text{if } p_c = 1 \end{cases} , \tag{7.10}$$

with p_c a boolean variable to identify contact given by (5.90), and σ_d given by (5.91). As before, the parameter c is a smoothing factor that defines adaptation dynamics, d_o is the distance to contact and d_t a threshold value. $K_{s_{pc}}$is the pre-contact stiffness and $\hat{K}_{s,2}$ is given by (6.26). The first term in (7.10) drives the probe motion in free space (see Section 5.3.4). The position tracking performance in free space motion is defined by the ratio between K_{virt_f} / K_{s_f}

$$k_{v_{pc}} = \frac{K_{virt_f}}{K_{s_f}} , \tag{7.11}$$

and by K_{s_f}. A higher ratio $k_{v_{pc}}$ increases system dynamics, being its maximum value limited by the manipulator motion bandwidth (Park and Khatib (2006)). K_{s_f} is set as low as possible to reduce errors due to modeling free space as a spring. A low K_{s_f} value leads to higher feedback gains (see Section 5.3.3), which entails better position tracking performance. From a haptic perspective, low K_{s_f} implies low K_{virt}, still enforcing position tracking (the position tracking capabilities are defined by the ratio $k_{v_{pc}}$), but without inducing "virtual" force feedback to the teleoperator, which appears when K_{virt} has a high value.

Table 7.1: Online Stiffness Estimation Design Parameters.

b_u	0.7	c_u	9.0	k_u	7.0	$[m^{-1}]$
b_o	1.0	c_o	7.0	k_o	3.0	$[m^{-1}]$
				k_m	500	$[NKg^{-1}m^{-1}]$

Table 7.2: Numerical Values of $K_{s,n}$ and K_{virt} Design Parameters.

c	100	d_t	0.03	$[m]$	K_{s_f}	10	$[Nm^{-1}]$
$k_{v_{pc}}$	4.0	f_{min}	2.0	$[N]$	$K_{s_{pc}}$	200	$[Nm^{-1}]$
$l_{s,n}$	0.05	f_{max}	4.0	$[N]$	K_{virt_f}	40	$[Nm^{-1}]$
l_o	0.05				K_{virt_M}	2000	$[Nm^{-1}]$

The second term in (7.10) increases K_{virt} proportionally to $K_{s,n}$ near contact. A higher $K_{s,n}$ reduces system dynamics when moving towards contact, allowing free space to contact smooth transitions. Increasing K_{virt} proportionally to $K_{s,n}$ prevents a substantial drop off in position tracking. From a haptic standpoint, the physician "feels" the increase in K_{virt}, experiencing a *dragging* effect when moving the probe in free space close to an object.

In contact, $\hat{K}_{s,2}$ drives K_{virt} (see Section 6.3) with $K_{virt} \leq K_{virt_M}$. K_{virt} is updated based on $\hat{K}_{s,2}$ instead of $\hat{K}_{s_c}$, preventing the physician from "feeling" the high dynamics of $\hat{K}_{s,1}$, which are particularly relevant in contact with stiff surfaces. In this way, K_{virt} changes with the effective mass, which might deteriorate telepresence specially when interacting with stiff objects, since the operator "feels" the perceived stiffness instead of the real object stiffness. However, for a task such as an ultrasound examination, the range of K_{virt} values is enough to provide realistic sensations to the physician, to distinguish soft tissue from surfaces like bones.

7.2 Control Design

This section summarizes the control design. Most design values have already been presented throughout the thesis, being repeated here for the reader's convenience. The control sampling time and system time delay are $h = 1.2[ms]$ and $T_d = h$, respectively.

7.2.1 Force Control Design

The damping parameters[2] are set to $K_2^t = 16.6$ and $K_2^v = 12.5$. The force control time constant is set to $\tau_f = 0.035[s]$. Table 7.1 gives the parameters for $\hat{K}_{s_c}$ estimation[3] and Table 7.2 gives the parameters for K_{virt} and $K_{s,n}$ computation. The null space gain $\boldsymbol{K}_N \in \mathbb{R}^{7\times 7}$ is

$$\boldsymbol{K}_N = diag \begin{bmatrix} 50 & 50 & \dots & 50 \end{bmatrix} .$$

In free space the pole placement gains are

$$\boldsymbol{L}_r = \begin{bmatrix} 79.68 & 3.98 & 0.048 \end{bmatrix} ,$$

being adapted according to (5.74) when $K_{s,n}$ changes. For the AOB stochastic design, a sensor based strategy (relatively high $\boldsymbol{Q}_k$ when compared to $\boldsymbol{R}_k$) is pursued with

$$\boldsymbol{Q}_k = diag \begin{bmatrix} 10^{-1} & 10^{-12} & 10^{-12} & 3\times 10^{-2} \end{bmatrix}$$

and $\boldsymbol{R}_k = 1$, leading to the following steady state Kalman gain in free space

$$\boldsymbol{K}_k = \begin{bmatrix} 0.2705 & 0.1133 & 0.148 & 0.148 \end{bmatrix}^T .$$

Again, when $K_{s,n}$ changes, $\boldsymbol{K}_k$ also changes. Due to force and position measurement noises, $\boldsymbol{f}_d$ and $\boldsymbol{f}_m$ are both filtered with a first order low pass Butterworth filter with cutoff frequencies of $25[Hz]$ and $5[Hz]$, respectively. Pre-filtering force signals allows to set higher control gains and a higher sensor based stochastic design, which is adequate for tele-echography tasks. A sensor based design enables better free space position tracking, reducing the effect of model errors introduced by modeling free space as a spring, which becomes notorious in pre-contact scenarios when the model stiffness is increased by (6.7). In contact, a sensor based design keeps the system responsive and, together with the contact anticipation strategy, avoids jerky behaviors when changing from free space to stiff contact scenarios.

7.2.2 Motion Control Design

Table 7.3 summarizes motion control gains. Attaching an ultrasound probe and a depth camera to the robot end-effector, the inertia of wrist joints increases, reducing the relative role of friction terms. In this way, a more conservative control design with lower control gains can be done for orientation

[2] Presented in Section 5.5

[3] Presented in Section 6.5.

Table 7.3: Motion Control Design.

	τ_v	K_1	$L_{1_{vel}}$	$L_{2_{vel}}$
q_1	0.008	29.02	116.08	0.14
q_2	0.008	29.02	116.08	0.14
q_3	0.0075	30.8	123.21	0.15
q_4	0.0075	30.8	123.21	0.15
q_5	0.00625	36.39	145.58	0.17
q_6	0.00625	36.39	145.58	0.17
q_7	0.01	23.56	94.23	0.11

control. All joints are designed with the same AOB stochastic strategy, following a sensor based approach,

$$\boldsymbol{Q}_{k_{vel}} = diag\begin{bmatrix}10^{-1} & 10^{-12} & 10^{-12}\end{bmatrix}$$

entailing the following steady state Kalman gain

$$\boldsymbol{K}_{k_{vel}} = \begin{bmatrix}0.2702 & 3.24 \times 10^{-10} & 3.24 \times 10^{-10}\end{bmatrix}^T .$$

High control gains and a sensor based strategy address the orientation control.

7.2.3 3D Camera Design

The camera integration time is set to $1000[\mu s]$ and the amplitude threshold is set to 4000. This setup provides environment description without probe detection. We have used the calibration data supplied by the manufacturer.

7.3 Testbed Experiments

In the testbed experiments, the probe holder manipulator is tele-operated through a Phantom Desktop haptic device, performing typical ultrasound motions in free space and in contact with soft (sponge) and stiff (hard cover book) environments. Fig. 7.2 shows the experimental setup. Throughout the experiments $\beta_f = 0.1$ and $\beta_p = 1$. The experiment in the testbed scenario can be summarized as follows. Probe motion starts in free space reaching contact with the book around $8[s]$. Constrained motion in the book happens from $10[s]$ to $19[s]$. From $20[s]$ to $52[s]$ typical echographic orientation motions are

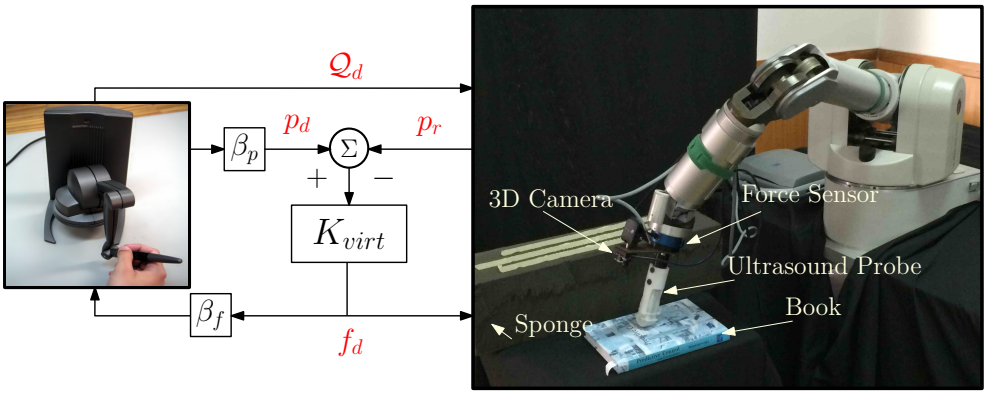


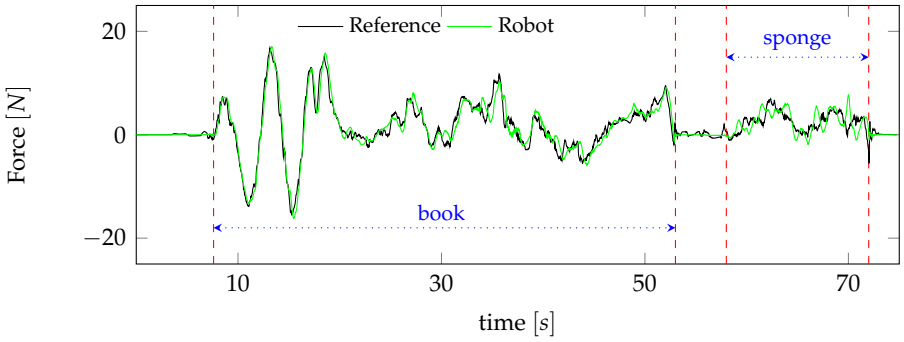
Figure 7.2— Experimental setup and the teleoperation architecture. Experimental setup used in the testbed experiments. Throughout the experiments $\beta_f = 0.1$ and $\beta_p = 1$.

performed. Contact is released around $53[s]$, being the probe moved towards the sponge in free space, reaching contact around $58[s]$. Typical echographic motions are then performed in the sponge. The book and the sponge are placed horizontally with respect to the robot base, $p_{book_z} \approx -0.33[m]$ and $p_{sponge_z} \approx -0.21[m]$ (see Fig. 7.2). While in contact, force is limited to $30[N]$ in any Cartesian direction.

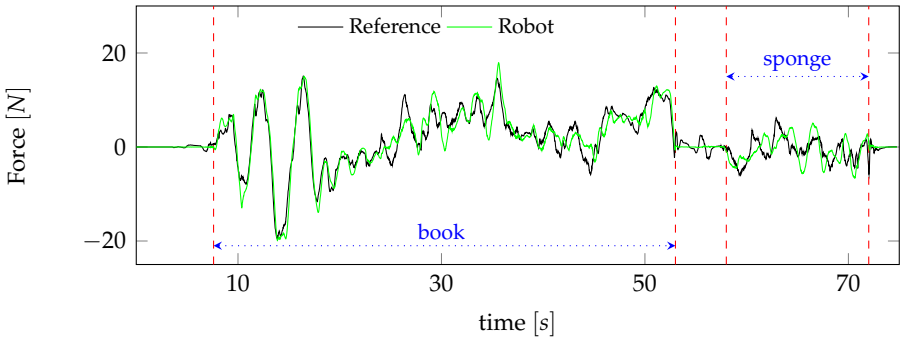
7.3.1 Discussion

Fig. 7.3 shows force control results. Good force tracking is achieved in both contact scenarios. Free space to contact transitions and vice-versa present no jerky behaviors and there are no noticeable steady state errors. Fig. 7.4 shows good position tracking in free space. From Figs. 7.3 and 7.4, consistent results can be observed while in constrained motion. Fig. 7.5 shows orientation control results. Good tracking performance can be inferred, both in contact and free space, validating the hierarchical control approach. However, Figs. 7.3 and 7.5 show that when performing orientation motions with contact constraints, specially when interacting with the stiff surface, force tracking errors arise. It can be further noticed in Fig. 7.3, that in some situations, robot force exceeds the $30[N]$ limit. This behavior results from orientation motion disturbances not being completely compensated by the null space, due to dynamic modeling errors. Fig. 7.6 shows online estimation results of $K_{s,n}$ and K_{virt} . In contact with the book, K_{s,n_z} ranges from $K_{s_{c,min}}$ when the

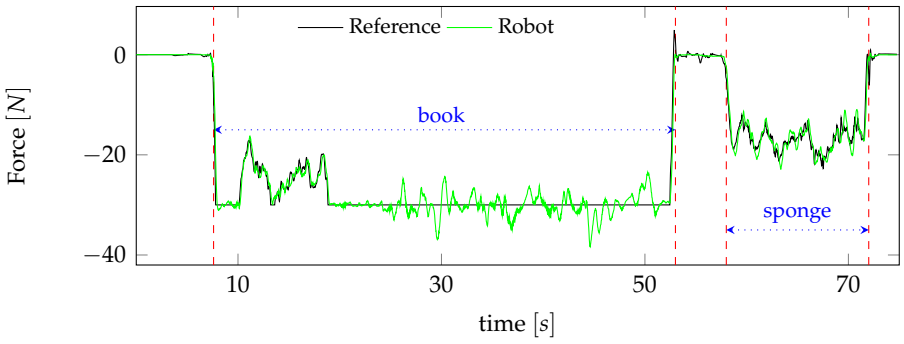
probe posture is parallel to the book (horizontal posture with low effective mass $47[s] - 49[s]$ see Fig. 6.14b) to $5200[Nm^{-1}]$ when performing orientation motions with contact constraints. Clearly $K_{s,n}$ does not reflect the real book stiffness. Since $K_{s,n}$ is estimated based on force relations, the estimation value reflects not only the environment stiffness but also the effect of unmodeled terms (e.g. damping), environment nonlinearities, dynamic modeling errors, and non uniform contact area, which explains $K_{s,n}$ high dynamics. From a control standpoint, since a stochastic sensor based strategy is followed, $K_{s,n}$ high dynamics does not pose a problem, enabling the controller to rapidly react to force mismatches. In this way, $K_{s,n}$ can be seen as an optimization parameter, reducing the system responsiveness whenever an underdamped response is imminent and increasing control gains when the system becomes sluggish. The AOB performance, despite being robust to environment model mismatches, deteriorates with model errors. From Fig. 7.3, a good estimation performance of $K_{s,n}$ can be inferred in both contact scenarios, since f_m tracks f_d with high performance.



(a)



(b)



(c)

Figure 7.3— Force tracking with stiffness adaptation in the testbed experiment. (a), (b) and (c) represent respectively, Cartesian force data in X, Y and Z. Red dashed vertical lines identify free space to contact and vice-versa transitions, while blue doted horizontal lines identify the contact surface.

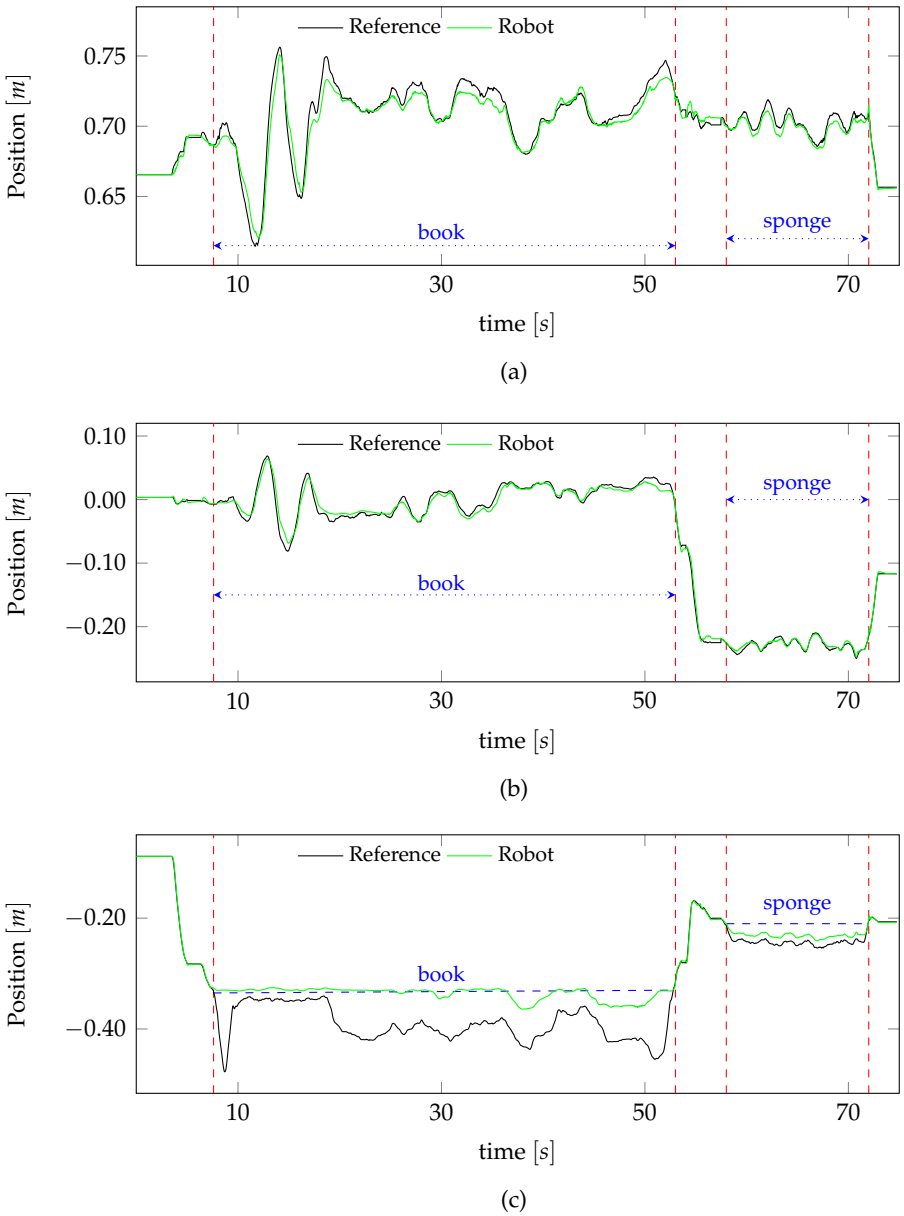
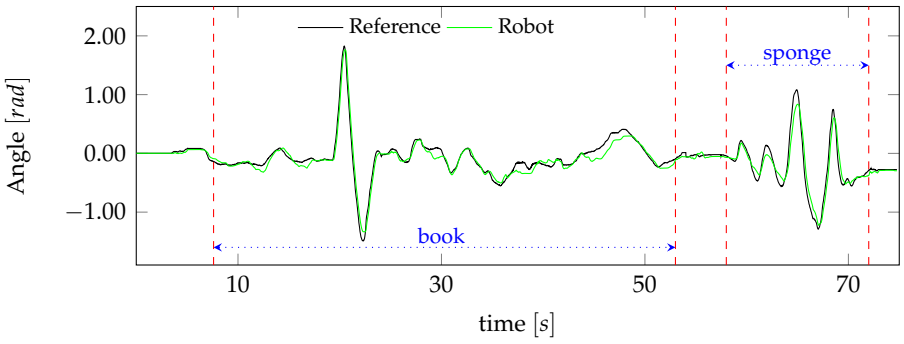
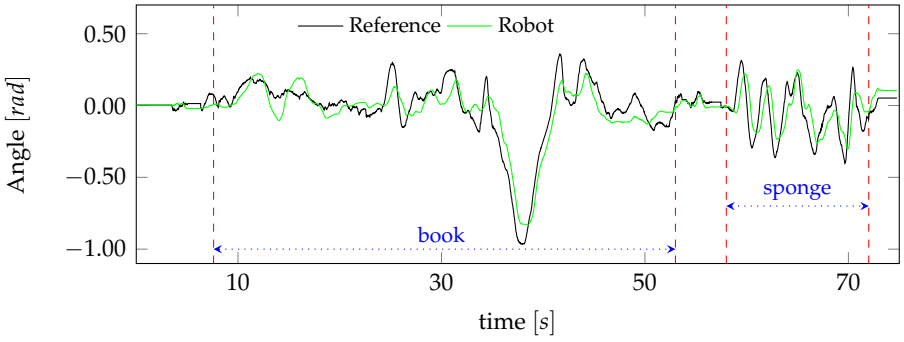


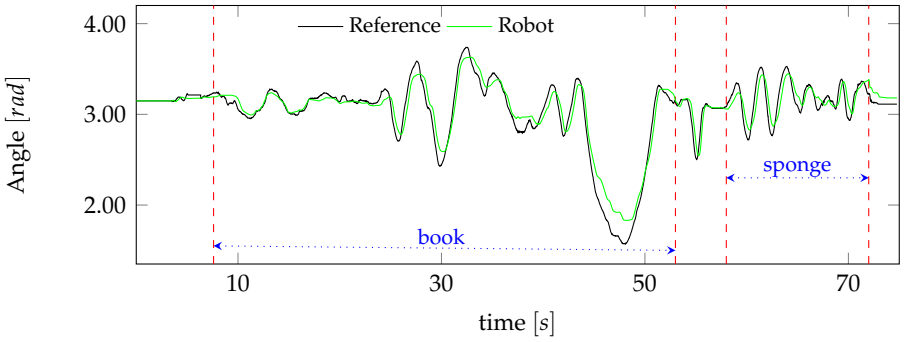
Figure 7.4— Cartesian position tracking in the testbed experiment. (a), (b) and (c) represent respectively, Cartesian position data in X, Y and Z. Red dashed vertical lines identify free space to contact and vice-versa transitions, while blue doted horizontal lines identify the contact surface. In (c), the blue dashed horizontal lines identify the surface location with respect to the robot base.



(a)

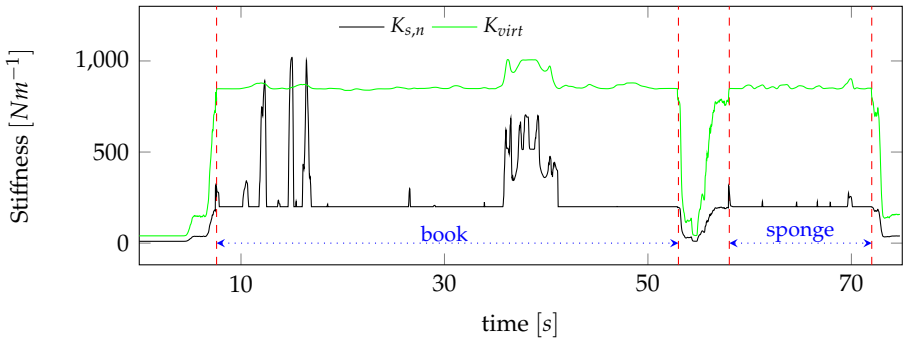


(b)

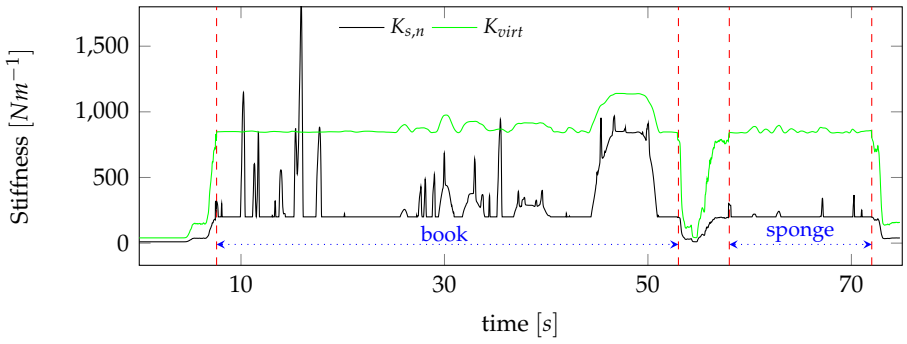


(c)

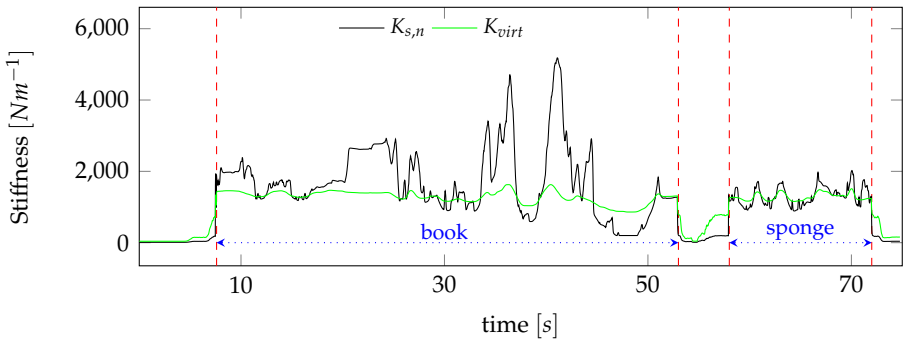
Figure 7.5— Orientation tracking in the testbed experiment. The orientation is represented by (a) Roll, (b) Pitch and (c) Yaw angles. Roll $\in \left[-\frac{\pi}{2}, \frac{3\pi}{2}\right]$, Pitch $\in [-\pi, \pi]$ and Yaw angle $\in [0, 2\pi]$. Red dashed vertical lines identify free space to contact and vice-versa transitions, while blue doted horizontal lines identify the contact surface.



(a)



(b)



(c)

Figure 7.6— $K_{s,n}$ and K_{virt} online computation in the testbed experiment. (a), (b) and (c) represent respectively, stiffness estimation and virtual coupling data in X, Y and Z. Red dashed vertical lines identify free space to contact and vice-versa transitions, while blue doted horizontal lines identify the contact surface.

7.4 Robotic-Assisted Tele-Echography Examination

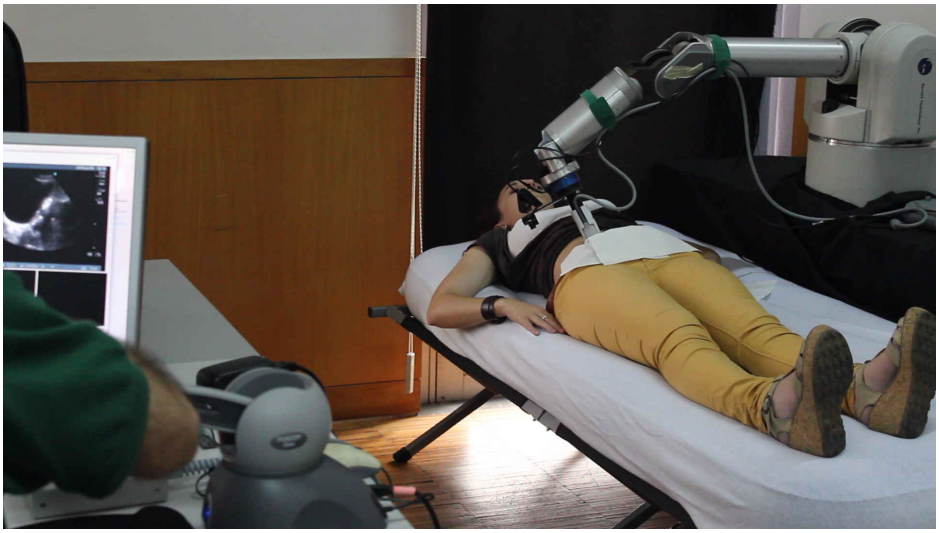
To validate the control architecture in a clinical context, a gynecologist performed a robotic-assisted pelvic ultrasound examination in a healthy volunteer. The physician tele-operates the probe holder manipulator through a Phantom Omni haptic device[4], sending 6-DOF (position and orientation) commands and receiving 3-DOF force feedback. The examination is performed with the patient urinary bladder full, which acts as an acoustic window for the pelvic organs, namely, the urinary bladder itself, the uterus and the ovaries. Fig. 7.7a shows the physician performing the ultrasound procedure while Figs. 7.7b–7.7e show ultrasound images of the pelvic examination. To increase the volunteer safety, the probe maximum velocity is limited to $|\dot{\boldsymbol{p}}_r| = 0.1[m/s]$ and $|\boldsymbol{\omega}_r| = 1[rad/s]$. The procedure starts with the probe in free space, reaching contact with the patient abdominal region around $12[s]$. Contact is kept during the examination ($12[s] - 111[s]$), except at $17[s]$, when the physician repositions the probe in free space and at the end of the procedure (around $109[s]$).

7.4.1 Discussion

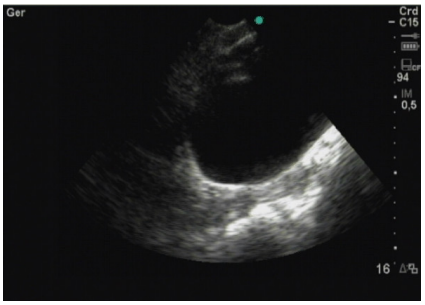
The pelvic ultrasound examination was chosen for two reasons. First, it is a dynamic procedure, requiring the probe to be placed in different locations with different postures, which highlights the system robustness and the control performance. In addition, the effectiveness of the robotic-assisted examination can be extrapolated to less dynamic abdominal procedures, validating the control architecture in those procedures. Second, it is easily performed in our lab without requiring to change the manipulator working space envelope (and hence its base) to a new position.

From a control standpoint, force and position are well tracked, as can be seen in Figs. 7.8 and 7.9, respectively. Free space to contact and vice-versa transitions present no jerky behaviors, there are no noticeable steady state errors in force tracking and the position tracking in contact is consistent with soft tissue interaction. Fig. 7.10 shows the orientation control performance. In general, the orientation is well tracked, despite a few tracking errors arising due to the hierarchical control strategy, as explained in Section 5.4.2. These errors can be tackled with higher control gains, however due to the task nature

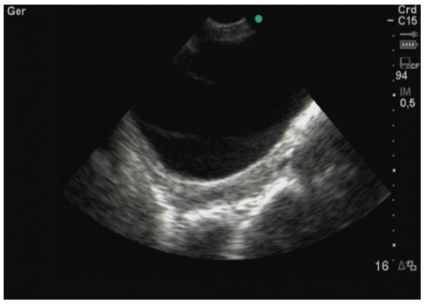
[4]When this experiment was performed, the Phantom Desktop was not operational.



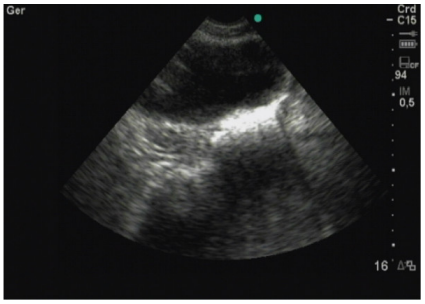
(a)



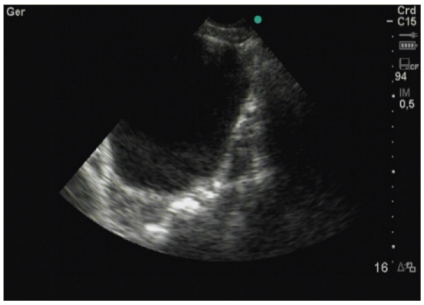
(b)



(c)



(d)



(e)

Figure 7.7— Robotic-Assisted Pelvic Ultrasound Examination. (a) The physician performing the robotic-assisted ultrasound examination in a healthy volunteer. (b)-(e) Ultrasound images of the procedure. (b) Uterus cross-section. (c) Uterus mid-sagittal section. (d) Left ovary. (e) Right ovary.

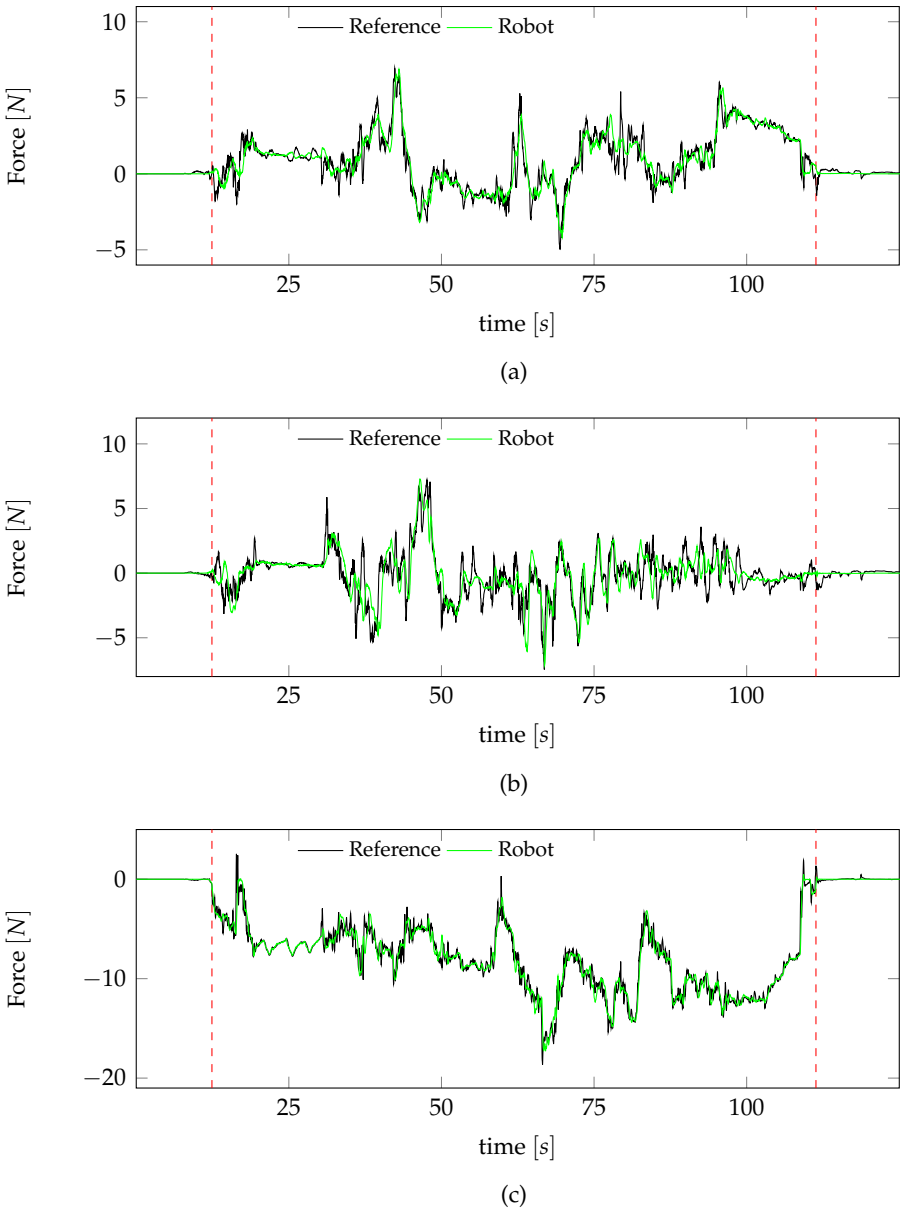


Figure 7.8— Force tracking with stiffness adaptation in the robotic-assisted ultrasound examination. (a), (b) and (c) represent respectively, Cartesian force data in X, Y and Z . Red dashed vertical lines identify free space to contact and vice-versa transitions.

(interaction with humans), a more conservative control design is followed. Nevertheless, the physician ability to perform the examination is not affected, validating the hierarchical approach to generate orientation motion. Finally, Fig. 7.11 shows $K_{s,n}$ and K_{virt} computation. A good estimation performance of the contact stiffness can be inferred, since f_m tracks f_d with high performance (see Fig. 7.8).

7.5 Conclusion

In this chapter the hierarchical architecture has been assessed in ultrasound tele-operation scenarios. Clinical and non-clinical experiments have been performed. In the non-clinical one, the control architecture is assessed in a testbed scenario, that simulates the interaction dynamics of a human torso. Testbed experiments have shown good results, either in free space and in contact with objects whose stiffnesses are similar to the human chest (bone contact) and abdomen, validating the proposed architecture to be tested in clinical robotic-assisted tele-echography scenarios. In the second experiment, the tele-operation architecture has been assessed in a medical ecographic diagnosis procedure, with a gynecologist performing a robotic-assisted pelvic ultrasound examination in a healthy volunteer. The physician effortlessly performed the procedure, validating the control architecture in a clinical scenario.

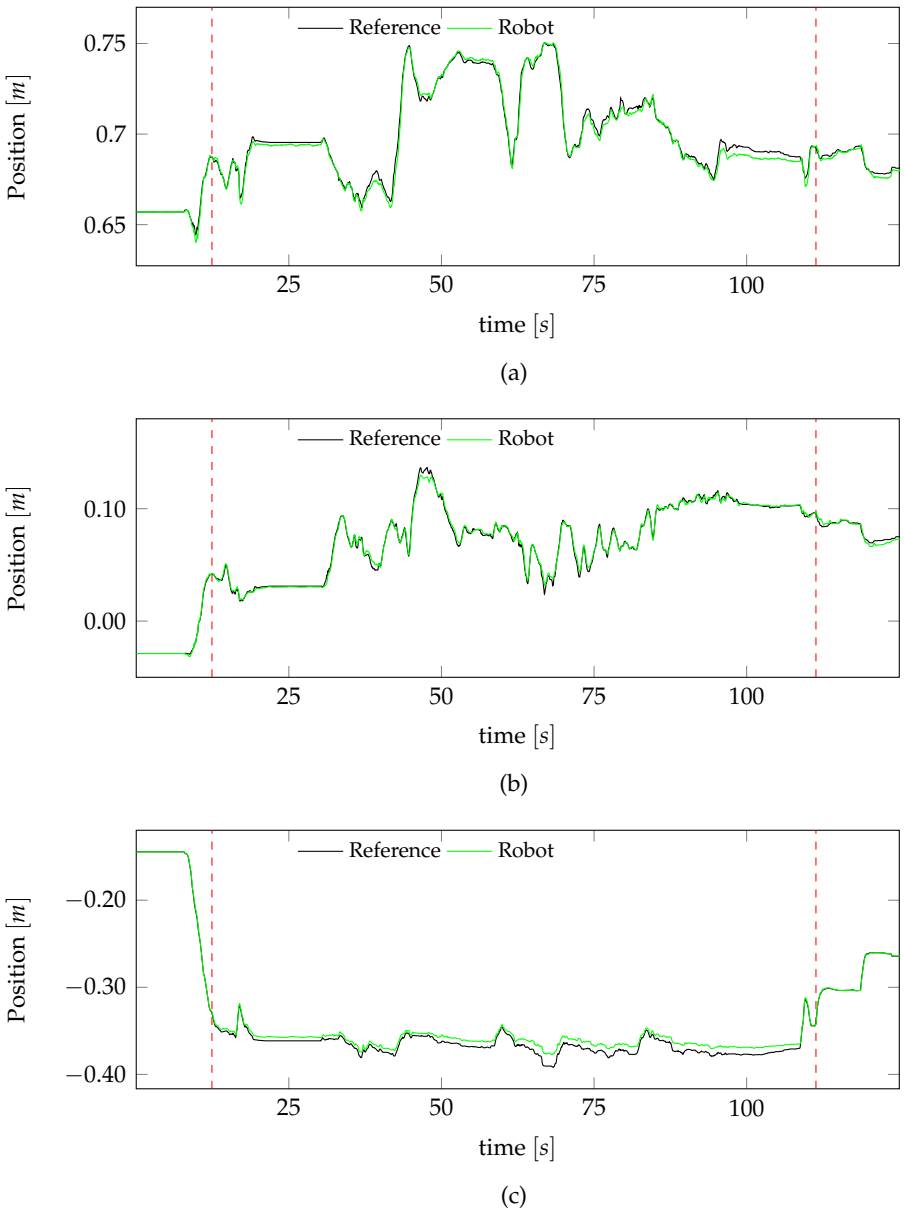


Figure 7.9— Cartesian position tracking in the robotic-assisted ultrasound examination. (a), (b) and (c) represent respectively, Cartesian position data in X, Y and Z. Red dashed vertical lines identify free space to contact and vice-versa transitions.

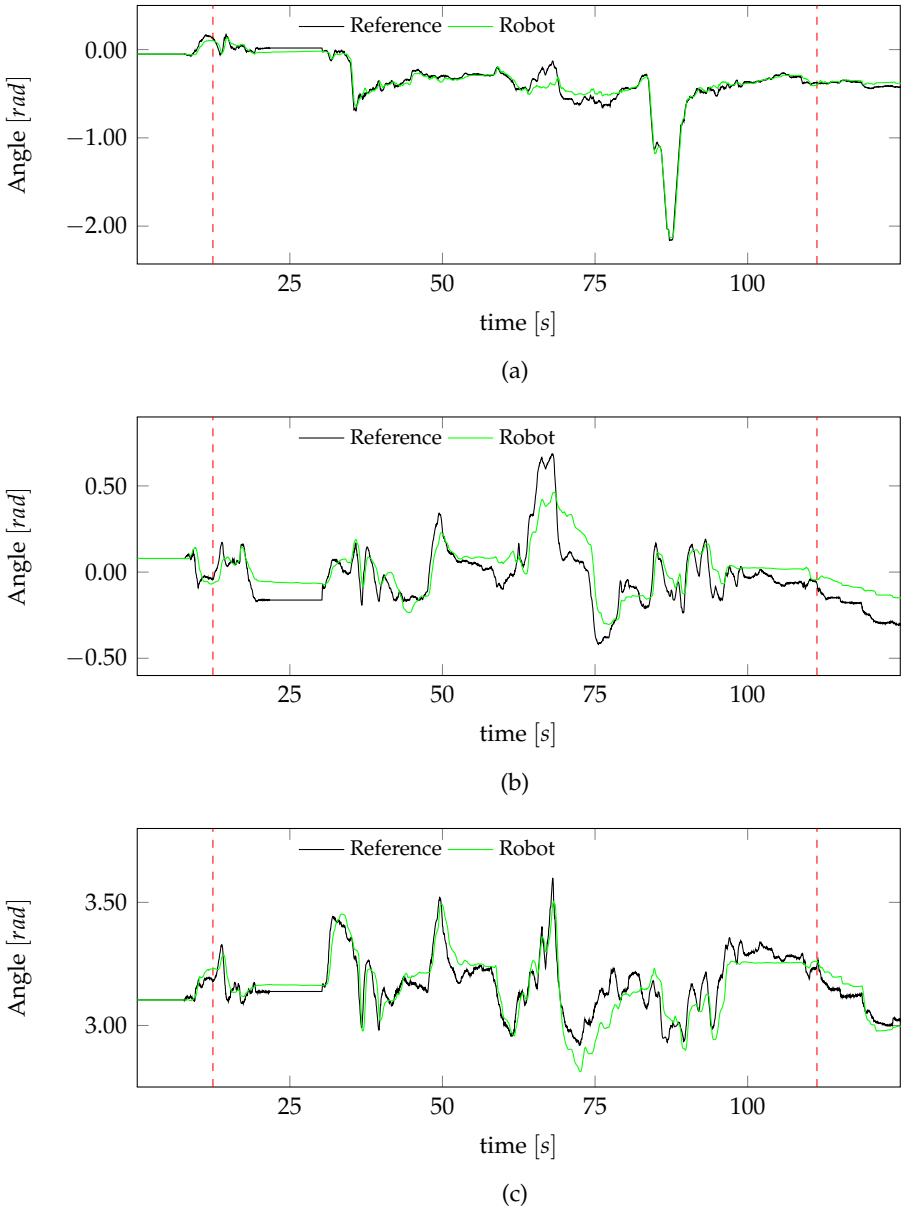


Figure 7.10— Orientation tracking in the robotic-assisted ultrasound examination. The orientation is represented by (a) Roll, (b) Pitch and (c) Yaw angles. Roll $\in [-\pi, \pi]$, Pitch $\in [-\pi, \pi]$ and Yaw angle $\in [0, 2\pi]$. Red dashed vertical lines identify free space to contact and vice-versa transitions.

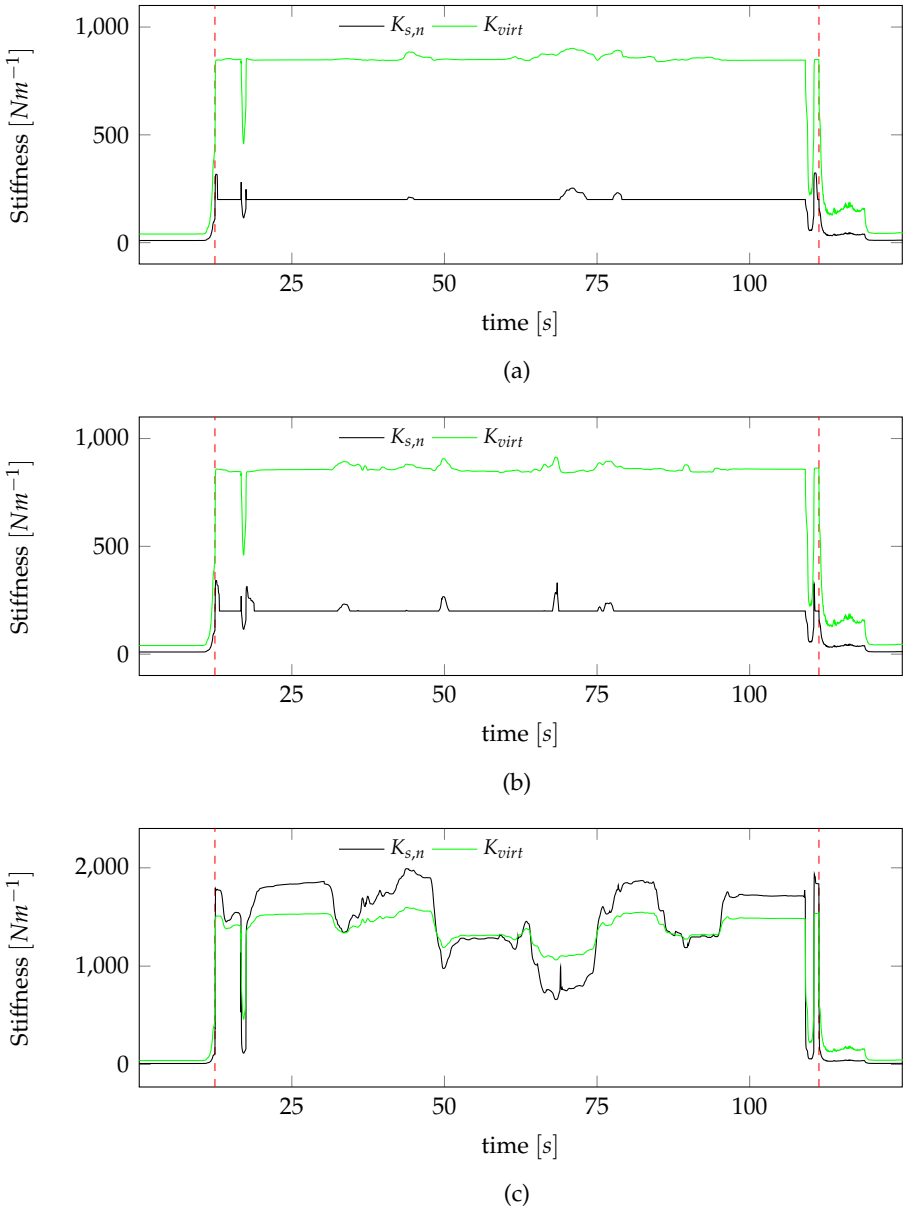


Figure 7.11— $K_{s,n}$ and K_{virt} online computation in the robotic-assisted ultrasound examination. (a), (b) and (c) represent respectively, stiffness estimation and virtual coupling data in X, Y and Z. Red dashed vertical lines identify free space to contact and vice-versa transitions.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

The ability to create visual representations of the interior of the human body through noninvasive imaging techniques was a major breakthrough in medical science. By pressing an ultrasound probe against the patient body, the physician is provided with real time access to the interior of the patient body in a safe (since there is no emission of ionizing radiation), noninvasive and relatively inexpensive way. Although being a common procedure, it is not a trivial one, requiring skilled and experienced physicians, which are not always available, specially in remote areas. A robotic-assisted tele-echography system comes up as a solution, allowing physicians to perform an ultrasound examination in a remote patient as if they were both at the same location. This chapter is organized as follows. Section 8.1 details the main contributions of this work, while Section 8.2 highlights future research directions.

8.1 Contributions

This section summarizes the main contributions of this thesis. A control architecture has been designed to cope with the following requirements:

- A RCM placed at the probe tip.
- A compliant contact behavior at position level.
- A stiff contact behavior at orientation level.
- High control performance in free space and in contact with soft and stiff environments, without control switching.

- Smooth transition from free space to contact and vice-versa.

- High control performance in contact when interacting with unstructured environments.

The main scientific contributions of this thesis are:

A Dynamically Consistent Hierarchical Control Architecture

The development of a robotic-assisted tele-control architecture for medical ecography is the backbone of this thesis. A dynamically consistent hierarchical control architecture is proposed, with explicit force control arising as the primary task, while orientation is performed in the null space, reducing coupling effects. An explicit Cartesian force control driven by position errors is regarded as the highest priority task, enforcing a compliant behavior at position level. Probe orientation is controlled at joint level, driven by task space orientation errors converted into joint velocity references, enforcing a stiff behavior by controlling the orientation at motion level. Controlling the orientation in the joint space allowed to achieve high tracking performance, addressing the limitations of task space orientation controllers.

Motion and Contact Dynamics driven by a 3D Time-of-Flight Camera and a Force Sensor

Since the force control architecture is kept in free space, to improve position tracking, the stiffness parameter that enters in the force control loop through the AOB plant model has to be designed with a low nominal value in free space, reducing the detrimental effect of modeling free space as a spring by increasing feedback control gains. When moving towards contact, the stiffness parameter is anticipated by 3D camera data, allowing adaptation before contact. This procedure decreases robot dynamics towards contact by reducing control gains, allowing a smooth transition from free space to contact. Free space inertial forces that deteriorate control performance are also eliminated from the control loop using depth camera data, improving free space control by providing reliable force measurements.

Online Stiffness Estimation using Force Data and the Robot Inertial Properties referred to the End-Effector

The unstructured nature of the echographic procedure requires a controller that has the ability to cope with different environments. In contact, the stiffness estimation is performed resorting only to force data, avoiding geometric uncertainty issues. Due to the noisy nature of force measurements, force filtering is usually required. It is shown in this thesis that, when the force signal is filtered, the inverse dynamics nonlinear feedback linearization no longer renders the manipulator dynamics decoupled from the contact dynamics, leading to different force responses for different end-effector contact orientations, since the effective mass is configuration dependent. Using the effective mass in the estimation algorithm, the stiffness perceived by the controller can be anticipated, improving force control performance in dynamic interactions, specially with stiff surfaces. Clearly, the elastic properties of an object do not change with probe interaction, however the controller perception of the object does change, existing a strong correlation between the effective mass and the perceived stiffness. In this way, it is shown that the perceived stiffness can be seen, not as a well defined physical property, but as a force control optimization parameter.

The control architecture has been validated in a clinical scenario, with a gynecologist performing a robotic-assisted pelvic ultrasound examination in a healthy volunteer. The physician effortlessly performed the procedure, validating the control architecture in a medical context.

8.2 Future Work

Robustness Analyses in the Presence of Random Communication Delays

The main focus of this work has been on the control of the slave manipulator. All teleoperation experiments were performed in a local area network with negligible time delays. When performing a robotic-assisted ultrasound examination between remote locations, delays are always present. If the communication channel is a packet-switched network, such as Internet, the delays can become randomly time-variable, and packets can be lost, which can lead the system to instability.

Estimation of Contact Forces

A force sensor is expensive and a source of noise. De Luca and Mattone (2005) presented a residual-based method to detect collisions without using force or acceleration measurements. The simple theoretical formulation of this method is attractive to estimate contact forces. Preliminary work shows potential to provide a reliable estimation of contact forces, allowing to remove explicit force sensing from the control loop.

REFERENCES

Abolmaesumi, P., S.E. E Salcudean, M.R. R Sirouspour, and S.P. P DiMaio (2002). "Image-guided control of a robot for medical ultrasound". In: *IEEE Trans. on Robotics and Automation* 18.1, pp. 11–23 (cit. on p. 13).

Aghili, Farhad (2010). "Robust impedance control of manipulators carrying a heavy payload". In: *Journal of Dynamic Systems, Measurement, and Control* 132.5 (cit. on p. 61).

Albu-Schäffer, Alin, Christian Ott, and Gerd Hirzinger (2007). "A Unified Passivity-based Control Framework for Position, Torque and Impedance Control of Flexible Joint Robots". In: *The International Journal of Robotics Research* 26.1, pp. 23–39 (cit. on p. 42).

Arbeille, Ph., J. Ayoub, V. Kieffer, P. Ruiz, B. Combes, A. Coitrieux, P. Herve, S. Garnier, B. Leportz, E. Lefbvre, and F. Perrotin (2008). "Realtime tele-operated abdominal and fetal echography in 4 medical centres, from one expert center, using a robotic arm & ISDN or satellite link". In: *Proc. IEEE Int. Conf. on Automation, Quality and Testing, Robotics (AQTR)*. Ieee, pp. 45–46 (cit. on p. 12).

Aström, Karl Johan and B. Wittenmark (1997). *Computer-controlled systems: theory and design*. Prentice-Hall International Editions (cit. on pp. 69, 73, 74, 76).

Avgousti, Sotiris, Eftychios G. Christoforou, Andreas S. Panayides, Sotos Voskarides, Cyril Novales, Laurence Nouaille, Constantinos S. Pattichis, and Pierre Vieyres (2016). "Medical telerobotic systems: current status and future trends". In: *BioMedical Engineering OnLine* 15.1, p. 96 (cit. on p. 12).

Beasley, Ryan a. (2012). "Medical Robots: Current Systems and Research Directions". In: *Journal of Robotics* 2012, pp. 1–14 (cit. on p. 4).

Biagiotti, Luigi and Claudio Melchiorri (2007). "Environment estimation in teleoperation systems". In: *Advances in Telerobotics*, pp. 211–231 (cit. on p. 92).

Bradley, William G. (2008). "History of Medical Imaging". In: *Proc. of the American Philosophical Society* 152.3, pp. 349–361 (cit. on p. 5).

Brown, Alan (2015). "Robots Grow in Numbers". In: *Mechanical Engineering Magazine* (cit. on p. 2).

Caccavale, Fabrizio, Ciro Natale, Bruno Siciliano, and Luigi Villani (1999). "Six-DOF impedance control based on angle/axis representations". In: *IEEE Trans. Robotics* 15.2, pp. 289–300 (cit. on pp. 36, 38, 46).

Chiaverini, Stefano (1997). "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators". In: *IEEE Trans. on Robotics and Automation* 13.3, pp. 398–410 (cit. on p. 24).

Chiaverini, Stefano and Lorenzo Sciavicco (1993). "The Parallel Approach to Force/Position Control of Robotic Manipulators". In: *IEEE Trans. on Robotics and Automation* 9.4, pp. 361–373 (cit. on p. 41).

Chiaverini, Stefano, Bruno Siciliano, and Luigi Villani (1999). "A survey of robot interaction control schemes with experimental comparison". In: *IEEE/ASME Trans. on Mechatronics* 4.3, pp. 273–285 (cit. on p. 40).

Chou, J.C.K. (1992). "Quaternion kinematic and dynamic differential equations". In: *IEEE Trans. on Robot. and Autom.* 8.1, pp. 53–64 (cit. on p. 36).

Clarke, R (1993). "Asimov's Laws of Robotics: Implications for Information Technology (Part I)". In: *Computer*, pp. 53–61 (cit. on pp. 1–3).

Clarke, Roger (1994). "Asimov's Laws of Robotics: Implications for Information Technology (Part II)". In: *Computer*, pp. 57–66 (cit. on p. 2).

Cortesao, R, Cristóvão Sousa, Pedro Queirós, Rui Cortesão, Cristóvão Sousa, Pedro Queirós, R Cortesao, Cristóvão Sousa, Pedro Queirós, Rui Cortesão, Cristóvão Sousa, and Pedro Queirós (2010). "Active impedance control design for human-robot comanipulation". In: *Proc. American Control Conference*, pp. 2805–2810 (cit. on p. 73).

Cortesão, Rui (2003). "Kalman techniques for intelligent control systems: theory and robotics experiments". PhD thesis (cit. on p. 71).

— (2007). "On Kalman Active Observers". In: *Journal of Intelligent and Robotic Systems* 48.2, pp. 131–155 (cit. on pp. 71, 72).

Cortesão, Rui and Ralf Koeppe (2001). "Compliant motion control with stochastic active observers". In: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. 4, pp. 1876–1881 (cit. on p. 75).

Cortesão, Rui, Jaeheung Park, and Oussama Khatib (2003). "Real-time adaptive control for haptic manipulation with active observers". In: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 2938–2943 (cit. on p. 74).

— (2006). "Real-time adaptive control for haptic telemanipulation with kalman active observers". In: *IEEE Trans. on Robotics* 22.5, pp. 987–999 (cit. on pp. 60, 68, 76, 90, 99, 101, 113, 115, 119, 120, 123, 125).

Courreges, F., P. Vieyres, G. Poisson, and C. Novales (2005). "Real-time singularity controller for a tele-operated medical echography robot". In: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, pp. 2222–2227 (cit. on p. 12).

Coutinho, Fernanda (2013). "Contact Stiffness Estimation Techniques for Robotic Manipulation". PhD thesis. University of Coimbra (cit. on p. 97).

Coutinho, Fernanda and Rui Cortesão (2011). "Adaptive stiffness estimation for compliant robotic manipulation using stochastic disturbance models". In: *International Journal of Systems Science* 42.8, pp. 1241–1252 (cit. on p. 97).

— (2013). "A neural-based approach for stiffness estimation in robotic tasks". In: *Proc. 16th Int. Conf. on Advanced Robotics (ICAR)*, pp. 1–7 (cit. on p. 97).

— (2014). "Online stiffness estimation for robotic tasks with force observers". In: *Control Engineering Practice* 24.1, pp. 92–105 (cit. on p. 97).

Craig, John J. (2005). *Introduction to Robotics: Mechanics and Control*. 3rd. Boston, MA, USA: Pearson/Prentice Hall (cit. on p. 19).

Dawson, D M, F L Lewis, and J F Dorsey (1992). "Robust Force Control of a Robot Manipulator". In: *International Journal of Robotics Research* 11, pp. 312–319 (cit. on p. 42).

De Luca, Alessandro and Raffaella Mattone (2005). "Sensorless robot collision detection and hybrid force/motion control". In: *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*. Vol. 2005. April, pp. 999–1004 (cit. on p. 146).

De Schutter, Joris, Herman Bruyninckx, WH Zhu, and Mark W. Spong (1998). "Force control: a bird's eye view". In: *Proc. IEEE CSS/RAS Workshop on: "Control Problems in Robotics and Automation: Future Directions"* December, pp. 1–14 (cit. on pp. 42, 69).

De Schutter, Joris and H. Van Brussel (1988). "Compliant Robot Motion II. A Control Approach Based on External Control Loops". In: *The International Journal of Robotics Research* 7.4, pp. 18–33 (cit. on pp. 41, 42).

Delgorge, Cécile, Fabien Courrèges, Lama Al Bassit, Cyril Novales, Christophe Rosenberger, Natalie Smith-Guerin, Concepció Brù, Rosa Gilabert, Maurizio Vannoni, Gérard Poisson, and Pierre Vieyres (2005). "A tele-operated mobile ultrasound scanner using a light-weight robot." In: *IEEE Trans. on Information Technology in Biomedicine* 9.1, pp. 50–58 (cit. on p. 12).

Denavit, J. and R. S. Hartenberg (1955). "A kinematic notation for lower-pair mechanisms based on matrices". In: *Trans. ASME Journal of Applied Mechanics* 22, pp. 215–221 (cit. on p. 20).

Diolaiti, Nicola, Claudio Melchiorri, and Stefano Stramigioli (2005). "Contact impedance estimation for robotic systems". In: *IEEE Trans. on Robotics* 21.5, pp. 925–935 (cit. on pp. 92, 96).

Erickson, D, M Weber, and I Sharf (2003). "Contact stiffness and damping estimation for robotic systems". In: *The International Journal of Robotics Research* (cit. on pp. 96, 97).

Fichtinger, Gabor, Peter Kazanzides, Allison M A.M. Okamura, Gregory D. Hager, Louis L Whitcomb, and Russell H R.H. Taylor (2008). "Surgical and

Interventional Robotics: Part II - Surgical CAD-CAM Systems". In: *IEEE Robotics & Automation Magazine* 15.September, pp. 94–102 (cit. on p. 4).

Fraisse, P., P. Dauchez, and F. Pierrot (2007). "Robust force control strategy based on the virtual environment concept". In: *Advanced Robotics* 21.3-4, pp. 485–498 (cit. on p. 42).

Garcia, J.G., A. Robertsson, J. Gomez Ortega, and R. Johansson (2006). "Generalized contact force estimator for a robot manipulator". In: *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*. May. Ieee, pp. 4019–4024 (cit. on p. 61).

Gilardi, G. and I. Sharf (2002). "Literature survey of contact dynamics modelling". In: *Mechanism and Machine Theory* 37.10, pp. 1213–1239 (cit. on pp. 91, 92).

Haddadi, Amir and K Hashtrudi-Zaad (2012). "Real-time identification of HuntCrossley dynamic models of contact environments". In: *IEEE Trans. on Robotics* 28.3, pp. 555–566 (cit. on pp. 96, 97).

Hertz, Heinrich Rudolf, D.E Jones, and G.a Schott (1896). *Miscellaneous papers*. Vol. 92 (cit. on p. 94).

Hoeckelmann, Mathias, Imre J. Rudas, Paolo Fiorini, Frank Kirchner, and Tamas Haidegger (2015). "Current Capabilities and Development Potential in Surgical Robotics". In: *International Journal of Advanced Robotic Systems* 12, p. 1 (cit. on p. 4).

Hogan, Neville (1985). "Impedance control: An approach to manipulation". In: *Journal of Dynamic Systems, Measurements and Control* March (cit. on pp. 40, 43).

Howe, Robert and Yoky Matsuoka (1999). "Robotics for Surgery". In: *Annual Review of Biomedical Engineering* 01, pp. 211–240 (cit. on p. 4).

Hsu, Ping, John Hauser, and Shankar Sastry (1988). "Dynamic control of redundant manipulators". In: *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*. 6. IEEE Comput. Soc. Press, pp. 183–187 (cit. on p. 60).

Hunt, K. H. and F. R. E. Crossley (1975). "Coefficient of Restitution Interpreted as Damping in Vibroimpact". In: *Journal of Applied Mechanics* 42.2, p. 440 (cit. on p. 94).

Johnson, K. L. (1985). *Contact Mechanics*. Cambridge: Cambridge University Press (cit. on p. 94).

Kane, D, W Grassi, R Sturrock, and P V Balint (2004). "A brief history of musculoskeletal ultrasound: 'From bats and ships to babies and hips'." In: *Rheumatology* 43.7, pp. 931–933 (cit. on p. 5).

Kazanzides, P, G Fichtinger, G D Hager, A M Okamura, L L Whitcomb, and R H Taylor (2008a). "Surgical and Interventional Robotics: Part I - Core Concepts, Technology, and Design". In: *IEEE Robotics & Automation Magazine* June (cit. on p. 4).

— (2008b). "Surgical and Interventional Robotics: Part III - Surgical Assistance Systems". In: *IEEE Robotics & Automation Magazine* 15.4, pp. 122–130 (cit. on p. 4).

Khalil, W. and E. Dombre (2004). *Modeling, identification & control of robots*. Butterworth-Heinemann (cit. on p. 19).

Khan, Akhtar S., Oscar Lopez-Pamies, and Rehan Kazmi (2006). "Thermo-mechanical large deformation response and constitutive modeling of viscoelastic polymers over a wide range of strain rates and temperatures". In: *International Journal of Plasticity* 22.4, pp. 581–601 (cit. on p. 93).

Khatib, Oussama (1987). "A unified approach for motion and force control of robot manipulators: The Operational Space Formulation". In: *IEEE Journal of Robotics and Automation* (cit. on pp. 31, 32, 41, 61, 62).

— (1995). "Inertial properties in robotic manipulation: An object-level framework". In: *The International Journal of Robotics Research* (cit. on pp. 33, 90, 101).

Khatib, Oussama, Luis Sentis, Jaeheung Park, and J. Warren (2004). "Whole-body dynamic behavior and control of human-Like robots". In: *International Journal of Humanoid Robotics* 01.01, pp. 29–43 (cit. on pp. 61, 63, 65).

Kikuuwe, R. and T. Yoshikawa (2002). "Robot perception of environment impedance". In: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. Vol. 2. May, pp. 1661–1666 (cit. on p. 95).

— (2003). "Recognizing Surface Properties using impedance perception". In: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. Vol. 1, pp. 1539–1544 (cit. on p. 89).

Kim, Byoung-Ho, Sang-Rok Oh, IL Hong Suh, and Byung-Ju Yi (2000). "A compliance control strategy for robots manipulators under unknown environment". In: *KSME International Journal* 14.10, pp. 1081–1088 (cit. on p. 95).

Koizumi, Norihiro, S. Warisawa, M Nagoshi, M Hashizume, and Mamoru Mitsuishi (2009). "Construction methodology for a remote ultrasound diagnostic system". In: *IEEE Trans. on Robotics* 25.3, pp. 522–538 (cit. on p. 13).

Koizumi, Norihiro, Shin'ichi Warisawa, Hiroyuki Hashizume, and Mamoru Mitsuishi (2008). "Continuous path controller for the remote ultrasound diagnostic system". In: *IEEE/ASME Trans. on Mechatronics* 13.2, pp. 206–218 (cit. on p. 13).

Kontaxakis, George, Stefan Walter, and Georgios Sakas (2000). "EU-TeleInViVo: an integrated portable telemedicine workstation featuring acquisition, processing and transmission over low-bandwidth lines of 3D ultrasound volume images". In: *Proc. IEEE Intern. Conf on Information Technology Applications in Biomedicine (ITAB)*. Vol. im. IEEE, pp. 158–163 (cit. on p. 4).

Krupa, Alexandre, David Folio, Cyril Novales, Pierre Vieyres, and Tao Li (2016). "Robotized Tele-Echography: An Assisting Visibility Tool to Support Expert Diagnostic". In: *IEEE Systems Journal* 10.3, pp. 974–983 (cit. on p. 12).

Kubus, Daniel, T Kroger, and FM Wahl (2008). "Improving force control performance by computational elimination of non-contact forces/torques". In: *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 2617–2622 (cit. on p. 61).

Lankarani, Hamid M. and Parviz.E. Nikravesh (1994). "Continuous Contact Force Models for Impact Analysis in Multibody Systems". In: *Nonlinear Dynamics* 5.2, pp. 193–207 (cit. on p. 91).

Lawrence, D.a. (1993). "Stability and transparency in bilateral teleoperation". In: *IEEE Trans. on Robotics and Automation* 9.5, pp. 624–637 (cit. on p. 125).

Liegeois, A (1977). "Automatic supervisory control of the configuration and behavior of multibody mechanisms". In: *IEEE Trans. on Systems, Man, and Cybernetics* 12, pp. 868–871 (cit. on p. 60).

Love, L.J. and W.J. Book (1995). "Environment estimation for enhanced impedance control". In: *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*. Vol. 2. 3, pp. 1854–1859 (cit. on pp. 95–97).

Mason, Matthew T. (1981). "Compliance and Force Control for Computer Controlled Manipulators". In: *IEEE Trans. on Systems, Man and Cybernetics* 11.6, pp. 418–432 (cit. on p. 41).

Mathiassen, Kim, Jørgen Enger Fjellin, Kyrre Glette, Per Kristian Hol, and Ole Jakob Elle (2016). "An ultrasound robotic system using the commercial robot UR5". In: *Frontiers in Robotics and AI* 3.February, pp. 1–16 (cit. on p. 13).

Monfaredi, Reza, Emmanuel Wilson, Bamshad Azizi Koutenaei, Brendan Labrecque, Kristen Leroy, James Goldie, Eric Louis, Daniel Swerdlow, and Kevin Cleary (2015). "Robot-assisted ultrasound imaging: overview and development of a parallel telerobotic system." In: *Minimally Invasive Therapy & Allied Technologies (MITAT)* 24.1, pp. 54–62 (cit. on p. 13).

Moreira, Pedro, Chao Liu, Nabil Zemiti, and Philippe Poignet (2012). "Soft Tissue Force Control Using Active Observers and Viscoelastic Interaction Model". In: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 4660–4666 (cit. on p. 92).

Moreira, Pedro, Nabil Zemiti, Chao Liu, and Philippe Poignet (2014). "Viscoelastic model based force control for soft tissue interaction and its application in physiological motion compensation". In: *Computer methods and programs in Biomedicine* 116.2, pp. 52–67 (cit. on p. 95).

Morton, Becky and Penny Delf (2008). "The prevalence and causes of MSI amongst sonographers". In: *Radiography* 14.3, pp. 195–200 (cit. on p. 7).

Murray, R M, Z Li, and S S Sastry (1994). *A mathematical introduction to robotic manipulation*. CRC Press (cit. on p. 19).

Mylonas, George P, Petros Giataganas, Muzzafer Chaudery, Valentina Vitiello, Ara Darzi, and Guang Zhong Yang (2013). "Autonomous eFAST ultrasound scanning by a robotic manipulator using learning from demonstrations". In: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 3251–3256 (cit. on p. 13).

Nadeau, Caroline and Alexandre Krupa (2013). "Intensity-Based Ultrasound Visual Servoing : Modeling and Validation With 2-D and 3-D Probes". In: *IEEE Trans. on Robotics* 29.4, pp. 1003–1015 (cit. on p. 13).

Nadeau, Caroline, Hongliang Ren, Alexandre Krupa, and Pierre Dupont (2015). "Intensity-based visual servoing for instrument and tissue tracking in 3D ultrasound volumes". In: *IEEE Trans. on Automation Science and Engineering* 12.1, pp. 367–371 (cit. on p. 14).

Najarian, Siamak and Elnaz Afshari (2012). "Evolutions and Future Directions of Surgical Robotics: A Review". In: *International Journal of Clinical Medicine* 03.02, pp. 75–82 (cit. on p. 4).

Nakamura, Yoshiihiko, Hideo Hanafusa, and Tsuneo Yoshikawa (1987). "Task-priority Based Redundancy Control of Robot Manipulators". In: *The International Journal of Robotics Research* 6.2, pp. 3–15 (cit. on p. 61).

Nakanishi, Jun, Rick Cory, Michael Mistry, Jan Peters, and Schaal Schaal (2008). "Operational space control: A theoretical and empirical comparison". In: *The International Journal of Robotics Research* 27.6, pp. 737–757 (cit. on p. 61).

Ogata, K. (2010). *Modern Control Engineering*. Prentice-Hall electrical engineering series. Instrumentation and controls series. Prentice Hall (cit. on p. 70).

Park, Jaeheung and Oussama Khatib (2006). "A Haptic Teleoperation Approach Based on Contact Force Control". In: *The International Journal of Robotics Research* 25.5-6, pp. 575–591 (cit. on pp. 99, 123, 126).

Pierrot, François, Etienne Dombre, Eric Dégoulange, Loïc Urbain, Pierre Caron, Sylvie Boudet, Jérome Gariépy, and Jean-louis Mégnien (1999). "Hippocrate: A safe robot arm for medical applications with force feedback". In: *Medical Image Analysis* 3.3, pp. 285–300 (cit. on p. 12).

Platt, Robert, Muhammad Abdallah, and Charles Wampler (2010). "Multi-Priority Cartesian Impedance Control". In: *Proc. Int. Conf. Robotics Science and Systems* (cit. on p. 61).

Priester, Alan M., Shyam Natarajan, and Martin Culjat (2013). "Robotic ultrasound systems in medicine". In: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 60.3, pp. 507–523 (cit. on p. 11).

Railbert, M. H. and J. J. Craig (1981). "Hybrid Position / Force Control of Manipulators". In: *Journal of Dynamic Systems, Measurement, and Control* 102.June 1981, pp. 126–133 (cit. on p. 41).

Sadeghian, Hamid, Luigi Villani, Mehdi Keshmiri, and Bruno Siciliano (2011). "Multi-priority control in redundant robotic systems". In: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. Ieee, pp. 3752–3757 (cit. on p. 61).

— (2014). "Task-space control of robot manipulators with null-space compliance". In: *IEEE Trans. on Robotics* 30.2, pp. 493–506 (cit. on p. 61).

Sage, H. G., M. F. De Mathelin, and E. Ostertag (1999). "Robust control of robot manipulators: A survey". In: *International Journal of Control* 72.16, pp. 1498–1522 (cit. on p. 42).

Salisbury, J. (1980). "Active stiffness control of a manipulator in cartesian coordinates". In: *Proc. IEEE Conf. on Decision and Control (CDC)*. Vol. 19, pp. 95–100 (cit. on p. 40).

Schindlbeck, Christopher and Sami Haddadin (2015). "Unified Passivity-Based Cartesian Force / Impedance Control for Rigid and Flexible Joint Robots via Task-Energy Tanks". In: *International Conference on Robotics and Automation*, pp. 440–447 (cit. on p. 42).

Sentis, Luis and Oussama Khatib (2005). "Synthesis of whole-body behaviors through hierarchical control of behavioral primitives". In: *International Journal of Humanoid Robotics* 02.04, pp. 505–518 (cit. on p. 65).

— (2006). "A whole-body control framework for humanoids operating in human environments". In: *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 2641–2648 (cit. on p. 61).

Seraji, Homayoun and Richard Colbaugh (1997). "Force Tracking in Impedance Control". In: *The International Journal of Robotics Research* 16.1, pp. 97–117. arXiv: `arXiv:1011.1669v3` (cit. on pp. 95–97).

Siciliano, B. and J. Slotine (1991). "A general framework for managing multiple tasks in highly redundant robotic systems". In: *Proc. 5th Int. Conf. on Advanced Robotics (ICAR)*, 1211–1216 vol.2 (cit. on p. 61).

Siciliano, Bruno (1990). "Kinematic control of redundant robot manipulators: A tutorial". In: *Journal of Intelligent and Robotic Systems* 3.3, pp. 201–212 (cit. on p. 60).

Siciliano, Bruno, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo (2009). *Robotics - Modelling, Planning and Control*. 1st. Advanced Textbooks in Control and Signal Processing. Springer London (cit. on pp. 19, 20, 29, 46).

Siciliano, Bruno and Luigi Villani (1996). "A Passivity-based Approach to Force Regulation and Motion Control of Robot Manipulators". In: *Automatica* 32.3, pp. 1–5 (cit. on p. 42).

— (2000). *Robot Force Control*. 1st. Norwell, MA, USA: Kluwer Academic Publishers (cit. on pp. 42, 47).

Singh, S. and D. Popa (1995). "An Analysis of Some Fundamental Problems in Adaptive Control of Force and Impedance Behavior: Theory and Experi-

ments". In: *IEEE Trans. on Robotics and Automation* 11.6, pp. 912–921 (cit. on pp. 95, 96).

Sousa, Cristovão (2014). "Dynamic model identification of robot manipulators: Solving the physical feasibility problem". PhD thesis (cit. on p. 34).

Sousa, Cristovao D. and Rui Cortesão (2014). "Physical feasibility of robot base inertial parameter identification: A linear matrix inequality approach". In: *The International Journal of Robotics Research* 33.6, pp. 931–944 (cit. on p. 34).

Sousa, Cristovão, Rui Cortesão, and Luís Santos (2010). "Computed torque posture control for robotic-assisted tele-echography". In: *Proc. IEEE 18th Mediterranean Conf. on Control & Automation (MED)*. 1. IEEE, pp. 1561–1566 (cit. on p. 17).

Spong, M.W. W, S. Hutchinson, and M. Vidyasagar (2006). *Robot modeling and control*. John Wiley & Sons Hoboken, NJ (cit. on pp. 19, 21, 27).

Taylor, RH H (2006). "A perspective on medical robotics". In: *Proceedings of the IEEE* 94.9 (cit. on pp. 3, 4).

Van Vliet, J. (2000). "Experimental Validation of Contact Dynamics Simulation of Constrained Robotic Tasks". In: *The International Journal of Robotics Research* 19.1993, pp. 1203–1217 (cit. on p. 89).

Verscheure, Diederik, Inna Sharf, Herman Bruyninckx, and Joris De Schutter (2009). "Identification of contact dynamics parameters for stiff robotic payloads". In: *IEEE Trans. on Robotics* 25.2, pp. 240–252 (cit. on p. 96).

Verscheure, Diederik, Inna Sharf, Herman Bruyninckx, Jan Swevers, and Joris De Schutter (2010). "Identification of Contact Parameters from Stiff Multi-point Contact Robotic Operations". In: *International Journal of Robotics Research* 29.4, pp. 367–385 (cit. on p. 97).

Vilchis, A, J Troccaz, P Cinquin, K Masuda, and F Pellissier (2003). "A new robot architecture for tele-echography". In: *IEEE Trans. on Robotics and Automation* 19.5, pp. 922–926 (cit. on pp. 12, 13).

Villani, Luigi, Carlos Canudas De Wit, and Bernard Brogliato (1999). "An exponentially stable adaptive control for force and position tracking of robot manipulators". In: *IEEE Trans. on Automatic Control* 44.4, pp. 798–802 (cit. on p. 42).

Volpe, Richard, Pradeep Khosla, and Senior Member (1993). "A Theoretical and Experimental Investigation of Explicit Force Control Strategies for Manipulators". In: *IEEE Trans. Automation and Control* 38, pp. 1634–1650 (cit. on pp. 42, 90, 103).

Wang, Zheng, Angelika Peer, and Martin Buss (2009). "Fast online impedance estimation for robot control". In: *Proc. IEEE Int. Conf. on Mechatronics (ICM)* (cit. on p. 95).

Whitcomb, Louis L., Suguru Arimoto, Tomohide Naniwa, and Fumio Ozaki (1997). "Adaptive model-based hybrid control of geometrically constrained

robot arms". In: *IEEE Transactions on Robotics and Automation* 13.1, pp. 105–116 (cit. on p. 42).

Whitney, Daniel (1985). "Historical perspective and state of the art in robot force control". In: *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)* 2, pp. 262–268 (cit. on p. 40).

Woo, Joseph (2006). *A short History of the development of Ultrasound in Obstetrics and Gynecology* (cit. on p. 5).

Yokokohji, Yasuyoshi and Tsuneo Yoshikawa (1994). "Bilateral control of master-slave manipulators for ideal kinesthetic coupling-formulation and experiment". In: *IEEE Trans. on Robotics and Automation* I.5 (cit. on p. 124).

Yoshikawa, T (1985). "Dynamic manipulability of robot manipulators". In: *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 1033–1038 (cit. on p. 25).

Yoshikawa, T. (1990). *Foundations of Robotics: Analysis and Control*. MIT Press (cit. on p. 19).

Zhu, W H, S E Salcudean, S Bachmann, and P Abolmaesumi (2000). "Motion/Force/Image Control of A Diagnostic Ultrasound Robot". In: *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 1580–1585 (cit. on p. 13).