Control of Contact Forces using Whole-Body Force and Tactile Sensors: Theory and Implementation on the iCub Humanoid Robot

BY
ANDREA DEL PRETE

A thesis submitted in partial fulfillment
of the requirements for the degree of
DOCTOR OF PHILOSOPHY

Doctoral Course on
ROBOTICS, COGNITION AND INTERACTION TECHNOLOGIES

Doctoral School on
LIFE AND HUMANOID TECHNOLOGIES

SUPERVISORS: Dott. Lorenzo Natale and Prof. Giorgio Metta
Università di Genova
Istituto Italiano di Tecnologia

GENOA, 2010–2013

If you can’t explain it simply, you don’t understand it well enough.

— Albert Einstein

Try not to become a man of success, but rather try to become a man of value.

— Albert Einstein

ABSTRACT

The new generation of service robots requires force control to safely cooperate with humans in unstructured dynamic environments. The objective of this thesis was to improve the abilities of humanoid robots to interact with their surrounding. In particular, the idea of integrating force and touch feedback was investigated, so as to control the robot in presence of external forces acting on any part of its body. To this aim, this work exploited the humanoid robot iCub as test platform, and it tackled three main issues: i) spatial calibration of tactile sensors, ii) estimation of contact forces using tactile sensors and force/torque sensors, and iii) prioritized position and force control.

The main contribution of this thesis is a new framework for prioritized position and force control of floating-base robots. The framework was compared to other state-of-the-art similar frameworks, both analytically and in simulation, and it proved preferable in terms of optimality and computational efficiency — twice as efficient, while preserving the optimality of the solution. Moreover, a method for estimating the 3D positions of tactile sensors was proposed. The method relies on force/torque measurements and it was exploited to calibrate the 1500 tactile sensors mounted on the arms of the iCub robot, with an average error of approximately 7 mm. Another method was introduced, which makes use of the calibrated tactile sensors, together with the distributed force/torque sensors, to estimate an arbitrary number of contact forces acting on any part of the robot's body. The method is based on the Recursive Newton-Euler Algorithm, and it was implemented as part of the open-source C++ library *iDyn*. Furthermore, a theoretical and empirical analysis investigated how incorrect estimation of contact points may affect the resulting contact forces and induce undesired joint accelerations. Tests on the iCub robot demonstrated a significant improvement in the performance of the force controller when the tactile system was used. All things considered, this work advanced the state of the art of force control of humanoid robots, providing estimation methods and control strategies that can be applied to make robots safely work side-by-side with humans.

To friends and family,
because where you come from,
you should never forget.

ACKNOWLEDGEMENTS

I want to start by saying thanks to the people that have been supporting me the longest, namely my family: my parents, Tania and Mauro, have given me the opportunity to study, and I have never taken it for granted, so the first big *thank* goes to them.

My girlfriend Roberta is the person who gave me more than anybody else in these three years; she made me a better person — or at least she tried — living with me every single day of my PhD. We have been together for 8 years now, in which she has taught me so many things, such as the value of sacrifice and organization, two qualities that helped me a lot during my PhD. She was there to cheer me up when things went bad, and to celebrate when things started going better. In a way, this PhD is as mine as hers, and I cannot thank her enough for this. And just to mention it, she is also the most beautiful, intelligent, fun and trustworthy woman I have ever met, and I am not overstating!

I think I have been really lucky to find a tutor as good as Lorenzo Natale: he has always been open, honest and available, qualities you do not find so easily in a person. At the same time he has also let me direct my research towards what was most interesting to me and he has always taken into account my ideas and opinions. A big thank goes to Francesco Nori who was my "second tutor" (even if he does not appear formally), especially in the last part of my PhD. His scientific support and his enthusiasm for my research topics have been a great help and stimulus in my research on inverse dynamics control. My supervisor Giorgio Metta definitely deserves a *thank*: he proved wise enough to known the right time to give responsibilities to people, an essential quality to be a good boss.

Not only I have found good bosses, but I have also been lucky enough to find myself in a laboratory full of wonderful people: an atmosphere of cooperation, rather than competition, has always ensured a pleasant working environment. All my colleagues have constantly been available to help me when I was struggling with topics that were over my head: Ugo, Randaz, Vadim, Carlo (and many others I cannot remember now) have taught me more than they probably realize. I have to thank also the *oldest* students who already left the lab: Matteo, Serena, Arjan, Alex; and how to forget the *younger* students who arrived after me: Ale, Luca, Dalia, Sean, Ilaria, Nick, Jor. All these people have been great buddies in and outside the lab, sharing with me most of the (rare) moments of relax away from work. A special thank goes to Carlo, who has been a great travel companion in our two trips to IROS in San Francisco and Algarve: his presence has made the work much more fun than it would have been without him.

I met a lot of wonderful people also outside my lab. Thanks to Alice, Eleò and Alessandra, who shared with me endless car trips from home to work, making the traffic jams almost fun. Massi, Ste, Jacob, Dalia, Moni, Albe, Albi and Sara have had lunch with me almost every day, helping me to take a nice and fun break from work. These same people and I have also shared many nights out in the “vicoli”, or dinners at someone’s home: in few words, they have been friends of mine.

My oldest friends Ale, Ruben, Bena, Zemi, Danilo, Jack, Pilons definitely deserve to be thanked. We have been friends since I was 12 (or even younger), and after so long, we still hang up every time I go back home: these are Friends, with the capital F.

Thanks to Nick for teaching me English. Thanks to Simone Denei and to Houman for collaborating with me. Thanks to Marco Frigerio for the nice chats in the hall about robot dynamics. Even if I have never met him, thanks to Professor Stephen Boyd from Stanford University for his wonderful online video-courses on linear dynamical systems and convex optimization.

I should also thank the (few) people who did not believe in me and tried to put me off: their lack of confidence in me has been a great stimulus to give more so as to prove myself.

To all the people I mentioned here, I would like to say that, even if sometimes I may seem not to care, I really do care about them, and I appreciate immensely their presence in my life. So, sincerely, thanks for having been here.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

This thesis deals with the position and force control of a humanoid robot interacting with the environment. The robot detects whole-body contacts with its distributed tactile sensor network and it measures contact forces using 6-axis force/torque sensors. To improve the abilities of the robot to interact with the surroundings, this work tackles three main issues: i) spatial calibration of tactile sensors, ii) estimation of contact forces, and iii) multi-task position/force control.

1.1 MOTIVATIONS: THE NEED FOR FORCE CONTROL

In the last half century we have been witnessing a major shift in robotic applications. While 50 years ago robots belonged only in industries and research laboratories, through the years some robotic platforms have entered our homes. Predictions for the future go so far as to compare the current emergence of the robotics industry with the PC-revolution started in the '70s: "*a robot in every home*" [Gates 2007]. Industrial manipulators today represent only a fragment of a wide range of robotics applications: caregiver robots help the elderly and disabled people, service robots [Kim et al. 2005; Roy et al. 2000] sweep the floor and mow the lawn, and therapy robots ease rehabilitation processes. The field of service robotics [Siegwart et al. 2003; Wyrobek et al. 2008] is trying to achieve a particularly challenging goal: building general-purpose robots that can work side by side with humans to do a variety of different chores.

Future predictions: "a robot in every home".

This shift in robotics applications calls for new capabilities. First, robots that share the workspace with humans have to be *safe*, meaning that they need to control their physical interactions with humans and environment [Haddadin et al. 2009; Heinzmann and Zelinsky 2003; Hirzinger and Albu-Schäffer 2001]. Physical interaction may be either intentional or due to uncertainties — typical of unstructured dynamic environments such as homes. Besides being safe, service robots should perform tasks in cooperation with other active agents (humans or other robots), hence they need to regulate contact forces on their whole body. Moreover, general-purpose robots can not be programmed as industrial manipulators that perform only few repetitive tasks. Non-expert users should be able to program robots by demonstrating a task, either visually or *kinesthetically* (i.e. physically moving the robot) [Argall et al. 2009; Calinon and Billard 2007; Hersch et al. 2008; Kormushev et al. 2011]. The paradigms of "programming by demonstration" and "kinesthetic teaching" are appealing also to the industry [Brooks 2008], because they simplify the programming phase and increase the adaptability of robots.

Robots need to interact with humans and the environment.

To meet these requirements, the new generation of robots demands *force control*, that is the ability to regulate the interaction forces exchanged between robot and environment.

1.2 FORCE CONTROL

The issue of providing robots with force control gave birth to several research trends, both on the hardware side and on the control/software side. This section briefly reviews the major efforts on both sides, in order to contextualize the presented work inside the field of force control.

1.2.1 *Hardware*

The field of Whole-Arm Manipulation (WAM) focuses on building robotic arms specifically designed to make contact on their whole surface. In 1988 Salisbury and Townsend wrote:

> Manipulator should be designed to contact and interact with the environment with any of its surfaces. [...] Since the system is intended to contact objects [...] it is important that forces anywhere along the links be controllable.

One approach to fulfill these requirements is *passive force control*, which relies on mechanical compliant elements. In 1995, Pratt and Williamson suggested that the introduction of passive springs at the joints could improve shock tolerance, lower reflected inertia, enhance accuracy and stability of force control, lessen environment damages, and allow for energy storage; research on *series elastic actuators* had then started. Along the same line of thoughts, people built *variable stiffness/impedance actuators*, which can control both position and mechanical impedance of the moving parts, in such a way to optimize performance while intrinsically guaranteeing safety [Laurin-Kovitz 1991; Tonietti et al. 2005].

Two approaches: active force control and passive force control.

An alternative approach is *active force control*, which, rather than exploiting elastic mechanical elements, commands the robot's motors so as to regulate the interaction forces. In this approach, *force/torque* sensors and *tactile* sensors are paramount, because they allow to detect, localize and measure contact forces. Joint position sensors are relatively easy to build an to integrate into a robotic system. On the other hand the problem of designing compact, robust and accurate joint torque sensors is far from solved. Most difficulties arise from the high sensitivity of the sensing elements, which easily measure unwanted force/torque components, specially in small and compact designs. Hirzinger and Albu-Schäffer [2001] presented advances towards a new generation of service robots that are characterized by joint torque sensors, light weight, high weight-to-payload ratio, and low power consumption. Tactile sensing has always been widespread in manipulation [Tegin and Wikander 2005], but in recent years we have seen more and more large-scale tactile networks that cover large parts of the robot's body [Minato et al. 2007; Schmitz et al. 2011]. Tactile sensors and force/torque sensors proved to be complementary technologies [Gordon and Townsend 1989], which give best results when used together to measure magnitude and location of multiple contact forces [Fumagalli et al. 2012].

1.2.2 *Control/Software*

Since the 1970s, force control has heavily impacted the field of robotics control [Groome 1972; Nevins and Whitney 1973; Whitney 1976]. Ever since, a few basic control paradigms have established the foundations of this research area (see [Chiaverini et al. 1999; De Schutter and Bruyninckx 1998; Whitney 1987] for a thorough review). The next subsection quickly describes the following well-known control paradigms: explicit force control, implicit force control, impedance control, hybrid control and parallel control. After that, we review the state of the art of the so-called "control frameworks", which allow robots to perform multiple tasks at the same time, combining together different control paradigms. Finally, we summarize the main challenges of force control, with a particular focus on humanoid robots.

1.2.2.1 *Control paradigms*

Explicit force control exploits force feedback to adjust the contact force to a desired value [Volpe and Khosla 1993; Whitney 1976]. On the other hand *implicit force control*, without any force feedback, regulates the position of the contact point, while tuning the joint servo gains so as to give a particular stiffness/admittance to the contact point [Borrel 1979; Rocco et al. 1997].

Well-known control paradigms: explicit/implicit force control, impedance control, hybrid control and parallel control.

In 1985, Hogan proposed *impedance control*, consisting in the control of the motion of the manipulator and, in addition, its dynamic behavior in response to external forces. Based on the observation that all environments can be modeled as *admittances* (i.e. physical systems that accept force inputs and yield motion outputs), Hogan suggested to control the manipulator as an *impedance* (i.e. a physical system that accepts motion inputs and yields force outputs). This guarantees that the two dynamically interacting systems physically complement each other. Differently from previous approaches, impedance control does not regulate motion or force directly, but instead it regulates the ratio of force to motion, that is the mechanical impedance [De Schutter and Bruyninckx 1998].

In 1979, Craig and Raibert introduced *hybrid control*, which regulates force and position independently, along orthogonal directions. Any assembly task involving rigid frictionless contact defines two sets of *natural constraints*: position constraints prevent the robot from moving through the environment, whereas force constraints prevent the application of forces along the tangent directions. These two sets of constraints partition the space into two orthogonal subspaces, which are then controlled according to different criteria [Khatib 1987].

More recently, Chiaverini and Sciavicco [1993] proposed to superimpose a proportional-derivative position control on a proportional-integral force control, giving birth to the so-called *parallel control*. The integral action in the force regulator ensures dominance of the force control over the position control, which is typically desired for safety reasons.

1.2.2.2 *Control frameworks*

Redundancy resolution and task prioritization.

In order to meet the increasing demand of flexibility and versatility, robot manipulators with more than six degrees of freedom appeared. These manipulators introduced the problem of *redundancy resolution*, namely to select a unique solution among the infinite control actions that generate the desired end-effector behavior. At first, redundancy was exploited to minimize the kinetic/potential energy of the manipulator [Whitney 1969], avoid obstacles in the workspace [Khatib and Maitre 1980], avoid singularities [Yoshikawa 1984], or keep joint angles and torques within limits [Liegeois 1977]. In 1987, Nakamura et al. were the first to discuss the concept of *task prioritization*: a task was divided into subtasks with different priorities and the joint motion was resolved so that lower priority subtasks used only redundancy not committed to higher priority subtasks. Nakamura et al. [1987] formulated the problem both in terms of joint velocity control [Whitney 1969], and joint acceleration control [Luh et al. 1980].

The problem of *redundancy resolution* drew increasingly more attention as research started to devote more attention to complex robots, such as humanoids, opening the possibility to perform many tasks at the same time. New control frameworks [Baerlocher and Boulic 1998; Siciliano and Slotine 1991] generalized the concept of task prioritization to the case of an arbitrary number of tasks. In 2004, Khatib et al. presented the first control framework working on the dynamic equations of the robot. Differently from the previous approaches, this framework allowed for explicit force control and hybrid control.

Many control frameworks have then been formulated in the last decade, each of them presenting particular features, such as: specification of inequality constraints [Mansard et al. 2009; Saab et al. 2011a], control of underactuated systems [De Lasa and Hertzmann 2009; Mistry and Righetti 2011], low computational cost [Escande et al. 2010; Mansard 2012].

1.2.2.3 *Main challenges*

The position and force control of humanoid robots has been an active research subject in the last decade, in part because of the many challenges that it exhibits.

Contact dynamics are difficult to model.

Physical interaction with the environment definitely represents a crucial issue: while we can estimate a dynamical model of the robot, it is much harder to retrieve a model of the *contact dynamics*. Researchers often resort to simplified models such as "perfectly-rigid frictionless contacts", or "spring-damper contacts", because they have proved sufficient to roughly capture the dynamics of the interaction. However, even when using these simple models, the online estimation of their parameters remains problematic.

Sensing capabilities represent another limiting factor: force and torque sensors are expensive and difficult to integrate on robots; moreover, tactile sensors are rarely found and often cover only limited parts of the robot's body with low resolution. As this thesis demonstrates, the latter is a crucial problem for the estimation of contact forces.

Most robots cannot sense contact forces due to the lack of adequate sensors.

Redundancy resolution, as we already mentioned, is another well-known challenge, and it has been the subject of a great body of work. Redundancy can either be seen as a problem, or as an opportunity to address other issues, such as singularities or joint torque/angle limits. Even if some control frameworks allows for inequality constraints, which can model joint torque/angle limits, this leads in turn to another problem, that is the *computational cost*. Similarly, to avoid singularities, one could resort either to local approaches, which do not guarantee optimality of the solution, or to global approaches, which require high computational cost. Since active force control benefits from fast control loop, control strategies should be as computationally efficient as possible (rate of force control loops is typically at least 1 kHz). A trade-off between efficiency and optimality seems to exist, but it is still unclear where it lies.

Redundancy is both a challenge and an opportunity.

Finally, underactuation is probably the most recent ingredient that fell into this recipe. Underactuated systems have less actuators than degrees of freedom, hence they can control only a portion of their state at each instant of time. Nonetheless, most of these systems are *controllable*, namely they can be driven from any initial state to any final state in finite time. Feedback linearization greatly simplifies the control of fully-actuated robots, reducing them to decoupled linear systems. Differently from fully-actuated systems, underactuated robots are not feedback linearizable, namely their dynamics cannot be linearized by a nonlinear feedback. Floating-base robots are a class of underactuated systems, which has been largely studied in the field of space robotics. Floating-base robots suffer from *dynamic singularities*, that are configurations in which the robot can not accelerate in certain directions due to the dynamic coupling between its active and passive degrees of freedom. This dynamic coupling makes unfeasible to address control and planning from a kinematic-only standpoint, as it is often done with fully-actuated systems. Dynamic models are typically less precise and more complex than kinematic models, hence their inclusion decreases the reliability of the planning, while increasing the computation time.

The control of underactuated robots is challenging because these systems are not feedback linearizable.

Moreover, humanoid robots differ from space robots because they are almost always in rigid contact with the environment, typically with the ground. They are then called "constrained systems", meaning that their motion is *constrained* by the rigid contacts; the contact constraints need to be taken into account in control and planning, increasing even further the complexity of these processes.

1.3 CONTRIBUTIONS OF THE THESIS

The work presented in this thesis belongs in the control/software side of force control.

The first part of the thesis treats problems that are strictly related to the specific sensor arrangement of the iCub robot, i.e. the distributed tactile sensors and the 6-axis force/torque sensors. Before implementing any force control paradigm we needed a way to measure and localize contact forces on the body of the robot; this part of the work was then essential for testing the second part. We devised a method for the spatial calibration of the tactile sensors and we tested it on the robot. The method exploits the 6-axis force/torque sensor to estimate the positions of the tactile sensors on the arm of the robot. Then we tackled the problem of fusing the information coming from the different sensors - tactile and force/torque - to estimate position and magnitude of the contact forces. Next we implemented some control strategies (explicit force control, position control, parallel control) on the iCub robot, and we studied how the tactile sensors improved the performance of the controllers.

The second part of the thesis deals with control frameworks: we present a new control framework for position and force control and we compare it with other state-of-the-art control frameworks. Both an analytical analysis and tests in simulation show the advantages of the presented framework, which concern its optimality and efficiency. Then we extend the presented analysis and the control framework to floating-base systems, which exhibit particular challenges because of their underactuation.
To summarize, we list here the main contributions of this thesis.

1. A method for spatial calibration of tactile sensors, presented in chapter 2 and published in [Del Prete et al. 2011].

2. A method for estimation of contact forces using tactile sensors and force/torque sensors, presented in chapter 3 and published in [Del Prete et al. 2012a].

3. A theoretical and empirical analysis of how errors in contact localization affect the control of contact forces, presented in chapter 4 and published in [Del Prete et al. 2012b].

4. A framework for position and force control of floating-base robots, presented in chapters 5 and 6, and submitted for publication in [Del Prete et al. 2013].

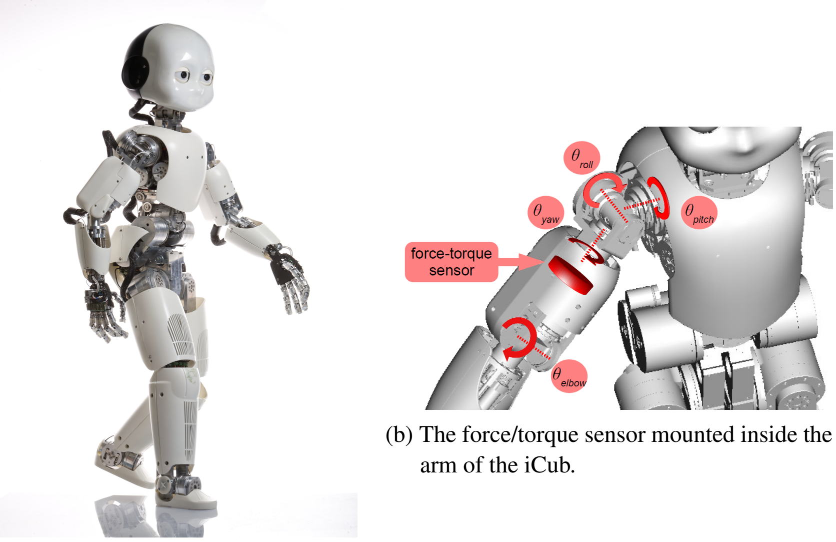
1.3.1 *Basic assumptions*

This work is based on a few assumptions, meaning that it does not tackle some of the aspects related to force control of humanoid robots. Even though we had to implement some simple planners for testing our control strategies, this thesis does not deal with the problem of computing reference trajectories (i.e. *planning*). Since we are interested in *model-based* control, we always assume to have a precise kinematic/dynamic model of the robot that we control. We extracted the kinematic parameters from the CAD model of the robot, whereas the *inertial parameters* were estimated, but we do not enter into the details of the process. Also, we do not consider the problem of errors in the robot parameters and the application of *adaptive/robust* control techniques. While we explain a technique for the spatial calibration of the tactile sensors, we assume that the force/torque sensors are already calibrated and they provide reliable measurements. We do not deal with the control of robots with passive springs at the joints (i.e. SEA or VIA). Although at the moment our test platform is not equipped with passive springs (apart from the fingers), some elasticity is surely present in the structure; however, we always assume to work with a perfectly rigid robot. Another assumption is the perfect tracking of the *joint torque control* of the DC motors. Finally, even if two cameras are mounted on the iCub robot, we do not exploit *vision* in the presented controllers.

This thesis does not deal with planning, model identification and series elastic actuators.

1.4 TEST PLATFORM — ICUB HUMANOID ROBOT & SOFTWARE ARCHITECTURE

This section presents the humanoid robot iCub (see Fig. 1.1a), which is the test platform for all the methods and controllers presented in this thesis. The iCub is an open robotic platform, designed for studying embodied cognition [Metta et al. 2008]. Its high number of degrees of freedom (53 in total, 30 in the upper body of which 9 just in each hand) and its small height of 104 cm, distinguish this robot from the other humanoid platforms worldwide. A large variety of sensors are mounted on the robot: digital cameras, gyroscopes, accelerometers, microphones, joint encoders, 6-axis force/torque sensors and a distributed sensorized skin.



(b) The force/torque sensor mounted inside the arm of the iCub.

(a) The iCub humanoid robot.

1.4.1 *Force/torque sensors*

Currently, the iCub is equipped with four custom made force/torque sensors [Fumagalli et al. 2010], one in each limb. The position of the sensors differs from the usual distal configuration at the end-effector. Indeed the sensors are located in between the shoulder and the elbow in the arm (see Fig. 1.1b), and in between the knee and the hip in the legs. The solution was dictated by practical reasons, but it has also some advantages: while the distal configuration can measure only end-effector forces, the proximal configuration allows the measurement of both internal dynamics and external forces exerted on the whole limb.

1.4.2 *Tactile sensors*

A compliant skin [Schmitz et al. 2011] incorporating a distributed pressure sensor, based on capacitive technology, is currently mounted on the torso, arms, palms and fingertips of the iCub (see Fig. 4.7). The skin is organized in triangular modules, except for the fingertips, where a particular solution has been designed for complying with the limited size and round shape. Each module, composed by 12 taxels (see Fig. 1.1), is able to scan locally 12 measurements of capacitance and send them through a serial bus.

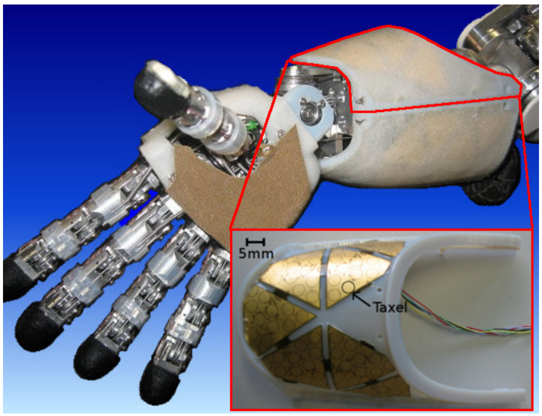


Figure 1.1: Right arm of the iCub robot, with skin mounted on the forearm (not covered with the lycra yet), palm and fingertips. In the red box, a detailed view of the forearm cover and the placement of the sensors [Schmitz et al. 2011]. This part is not covered with the silicone foam yet.

The basis of the sensor is a flexible printed circuit board (PCB). A 2/3 mm thick layer of silicone foam is placed above the PCB, covering the 12 taxels. The role of this layer is two-fold: (i) it acts as a deformable dielectric for the capacitive pressure sensor and (ii) it makes the skin compliant. A second conductive layer of lycra is placed on top of the silicon foam, in order to make the sensors sensitive to every material, regardless of its electrical properties. The capacitance of all the taxels is read with an 8 bit resolution, since any higher resolution is covered by noise. In every skin triangle the 12 taxels can be read either independently at 50 Hz, or as an average of them at about 500 Hz.

1.4.3 *Software*

1.4.3.1 *YARP*

YARP [Metta et al. 2006] (Yet Another Robot Platform) is the software middleware that we exploited for the implementation of all the software reported in this thesis. YARP is a set of C++ libraries, protocols, and tools to keep modules and devices cleanly decoupled. It supports building a robot control system as a collection of programs communicating in a peer-to-peer way, using different connection types (tcp, udp, multicast, local, MPI, mjpg-over-http, XML/RPC, tcpros). It also supports similarly flexible interfacing with hardware devices. YARP allows software modules to communicate using ports and it supports both synchronous and asynchronous communication. YARP is free and open, under the LGPL.

1.4.3.2 *iDyn*

The software library *iDyn* [Ivaldi et al. 2011] is at the core of the implementation of force control on the iCub. Using the dynamic parameters of the robot (extracted from the CAD model) and the data of F/T sensors and tactile sensors, *iDyn* can:

- estimate the joint torques to simulate joint torque sensors and so implement joint torque control in the control boards of the motors;
- estimate the contact forces [Del Prete et al. 2012a].

The information relative to the contact points comes from a software module called *SkinManager*, which uses the 3D positions of the tactile sensors to compute the center of pressure of each contact area. To obtain this information the tactile sensors were previously calibrated using the technique presented in [Del Prete et al. 2011].

Part I

CALIBRATION AND ESTIMATION

2

SKIN SPATIAL CALIBRATION USING FORCE/TORQUE MEASUREMENTS

This chapter deals with the problem of estimating the position of tactile elements (i.e. *taxels*) that are mounted on a robot body part. This problem arises with the adoption of tactile systems with a large number of sensors, and it is particularly critical in those cases in which the system is made of flexible material that is deployed on a curved surface. In this scenario the location of each taxel is partially unknown and difficult to determine manually. Placing the device is in fact an inaccurate procedure which is affected by displacements in both position and orientation. Our approach is based on the idea that it is possible to automatically infer the position of the taxels by measuring the interaction forces exchanged between the sensorized part and the environment. The location of the contact is estimated through force/torque (F/T) measures gathered by a sensor mounted on the kinematic chain of the robot. Our method requires few hypotheses and it can be effectively implemented on a real platform, as demonstrated by the experiments with the iCub humanoid robot.

2.1 INTRODUCTION

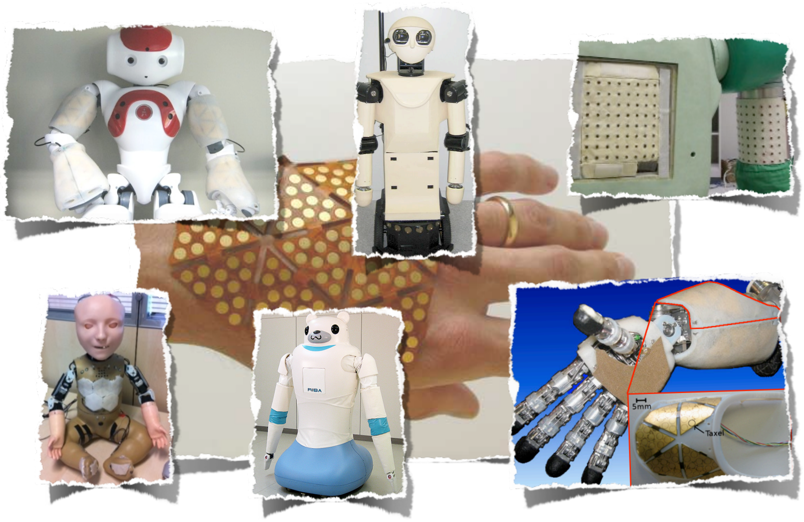


Figure 2.1: Examples of large scale tactile systems mounted on different robotic platforms. From the top left corner there are Nao, Robovie, Riman, Kaspar, Riba and iCub.

Nowadays, robots are expected to exhibit advanced features and complex interaction capabilities. For these reasons, recent research work has been focused on the design of robot structures that either take inspiration from the human functional structure [Marques et al. 2010; Metta et al. 2008] or are specifically designed for particular tasks [Hoffman and Weinberg 2010]. Furthermore, the design of deeply embedded and functional sensor systems that extend the perception of the surrounding environment has been enforced. In this field, particular attention has been devoted to *tactile sensing*: the key feature for enabling safe human-robot interaction and autonomous tasks in unstructured environments [Lumelsky 2006; Taichi et al. 2006].

The development of large-scale tactile sensors requires to deal with several issues that are well beyond the realization of single prototypes. Scalability, conformability [Ohmura et al. 2006], wiring [Cannata et al. 2008] and networking [Baglini et al. 2010] are just some of the issues that must be taken into account in the definition of a large scale tactile system that can cover huge parts of a robot body. Ohmura et al. [2006] presented a possible attempt to solve these issues, with a network of tactile elements that can be applied to arbitrarily curved surfaces. In spite of the low spatial resolution and high power consumption, the proposed solution allows a simple mechanical integration of transducers on the robot body. Cannata et al. [2008] presented another example of a modular skin, in which the triangular shape of the sensing modules and the adoption of flexible PCB ensure good conformal properties. Moreover, the modular design allows the skin to be adapted to robot platforms with varying shape and morphology [Schmitz et al. 2011].

In spite of the issues discussed above, today we can count on a number of prototypes providing robots with *large-scale*, *skin-like* tactile systems [Cannata et al. 2008]. These devices are characterized by a large number of sensors, which are essential for measuring contacts at high resolution. Dealing with such a large number of tactile units that can be freely placed on a robot has revealed a new class of problems, like folding the skin on the surface to best cover the robot, or the *spatial calibration of the tactile elements*. The latter has been defined [Cannata et al. 2010] as *the process of finding the location of each tactile element with respect to a known reference frame, after the skin sensor has been actually fixed on a robot body part*. Knowing the location of the tactile elements (i.e., *taxels*) is fundamental to develop complex autonomous behaviors, such as quick response to sudden stimuli or compliant human-robot interaction. Unfortunately the precise location of the taxels is unavailable because the deployment of the tactile device is an imprecise procedure that is affected by unpredictable displacements in both position and orientation. Yet, due to the large number of sensing units, the manual measure of the position of each taxel is tedious and error prone. It is fundamental to realize an automatic or semi-automatic procedure for calibrating the skin once it is placed on the robot.

The design of robots that are able to learn their own sensory space is not a new problem. Pierce and Kuipers [1997] introduced the paradigm of a learning agent that is able to infer the structure of its sensorimotor system through the interaction with the surrounding environment. This work defines a set of primitives that can be used by a robotic platform to build cognitive maps of the sensor system. A different approach is followed by Kuniyoshi et al. [2004]

and Noda et al. [2008], who reconstructed a topological map (rather than a geometric map of taxel locations). Kuniyoshi et al. [2004] presented a method for building an artificial somatosensory map that reflects the spatio-temporal correlations between incoming signals from 250 tactile sensors distributed on the skin of a simulated baby. The result is a map describing the structure of the robot body rather than a set of positions in the robot space. The work presented by Noda et al. [2008] is somewhat similar to the work of Kuniyoshi et al. [2004], but it introduces a different approach based on the extraction of features from correlated sensors and it has been tested with real tactile elements on the robotic platform Robovie-IIF. Also Modayil [2010] used the correlation between sensors to perform an embedding of sensors, that is, in this paper, a mapping between the sensors space and a Euclidean space. The author shows that, after defining a sensor correlation distance, it is possible to build a representation of sensors that preserve the computed distances, with the technique of Maximum Variance Unfolding. Although the results show a reconstruction of the geometry of the sensors placed on a 3D model with different curvatures and holes, the constructed map does not correspond to real sensor positions, but it is a scaled version resembling the original shape. Cannata et al. [2010] proposed an automatic procedure for calibrating a robot skin, which is based on the compliant motion between a robot and an external object with known shape. Although the results reported in the paper show the feasibility of the approach in simulation, this work makes some simplifying hypothesis and assumes a certain prior knowledge about the environment and, overall, it seems difficult to replicate on a real robotic platform.

With the sole exception of this last work [Cannata et al. 2010], all the available calibration techniques produce maps that are topological. Unfortunately for such tactile maps to be useful for control (at least considering the control systems that are commonly implemented in robotics) it is necessary to provide robots with metric information, rather than only topological.

The major contribution of this chapter is a method for the geometrical calibration of robot skin in 3D space. Our method can be implemented on any robot platform that is equipped with a sensor that can measure forces and torques on the part of the body that requires calibration. The experimental section shows that this technique allows the estimation of the positions of the taxels with good precision (average error of about 7 mm). This can be further improved if knowledge of the shape of the robot is available.

The chapter is structured as follows: section 2.2 describes the proposed method for estimating the taxel positions. Section 2.3 and section 2.4 report and discuss the results, respectively.

2.2 DETERMINING TAXEL POSITIONS THROUGH FORCE/TORQUE SENSOR MEASUREMENTS

The proposed approach is based on the idea that it is possible to determine where the taxels are placed by applying forces on the sensorized part. In fact, under some assumptions, by measuring the resulting forces and torques on the body part it is possible to determine the point of application of the forces (i.e., the contact point). If we apply forces that activate a small set of sensors near the point of application of the force, the measured point of contact provides an estimation of the position of the activated taxels.

The problem of determining the contact location, given force and torque measurements, has been already investigated in the literature [Bicchi et al. 1993a; Eberman and Salisbury 1990]. When arbitrary forces and moments are applied to a body and only the measures of a single sensor attached to the body are available, it is hard to derive the contact geometry and, as a consequence, to determine each contact point. Let us consider for instance a sensor measuring forces and torques on a rigid body, on which n wrenches are applied at different locations $c_u \in \mathbb{R}^3$, $u = 1, \dots, n$ (expressed w.r.t. the sensor reference frame). The equations relating the forces f_{c_u} and the moments m_{c_u} applied to the body, with the force and the moment measured by the F/T sensor, respectively f_M and m_M, are:

$$f_M = f_b + \sum_{u=1}^{n} f_{c_u},$$

$$m_M = m_b + \sum_{u=1}^{n} (m_{c_u} + c_u \times f_{c_u}).$$

where f_b and m_b are the internal force and moment due to the weight of the body (in case the body is moving, also its velocity and acceleration affect the internal force and moment). In the following we assume that f_b and m_b are known, and we define the part of the F/T sensor measure that is due to external forces as:

$$f_s = f_M - f_b = \sum_{u=1}^{n} f_{c_u},$$
$$m_s = m_M - m_b = \sum_{u=1}^{n} (m_{c_u} + c_u \times f_{c_u}).$$

In the hypothesis that only a single contact point c exists, no torque can be applied to the body (i.e. $m_c = 0$). In this case, the number of unknowns of the problem decreases, and some information about the contact point can be retrieved [Bicchi et al. 1993a]:

$$f_s = f_c, \tag{2.1}$$

$$m_s = c \times f_c. \tag{2.2}$$

Notice that in practice the assumption $m_c = 0$ can be easily verified. In fact, from (2.1) and (2.2) it follows that:

$$f_s m_s = 0. \tag{2.3}$$

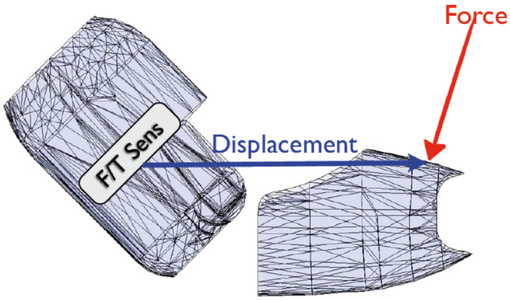


Figure 2.2: The basic principle on which the calibration is based: the moment at the F/T sensor is equal to the cross product between the displacement (i.e. the distance between the F/T sensor and the contact point) and the contact force.

Now, substituting (2.1) in (2.2) leads to:

$$m_s = c \times f_s, \tag{2.4}$$

which is a three equation system, where the only unknowns are the three coordinates of the contact point c. The system can not be completely solved, because the three equations are not independent. However, we can determine the axis of the force r:

$$c \in \left\{ \frac{f_s \times m_s}{||f_s||^2} + \lambda f_s \right\} = r, \qquad \forall \lambda \in \mathbb{R},$$

where λ is a free variable and $||.||$ is the Euclidean norm.

Suppose that we are applying a sequence of pure forces to the sensorized robot body part; we can define the set of all the h measured wrenches $w_{s_i} \in \mathbb{R}^6$, $i = 1, \ldots, h$ as:

$$W_s = \{w_{s_1}, w_{s_2}, \ldots, w_{s_h}\},$$

$$w_{s_i} = \begin{bmatrix} f_{s_i} \\ m_{s_i} \end{bmatrix}.$$

For estimating the position p_j of the j-th taxel we can use only the subset of W_s defined as:

$$\tilde{W}_j = \{w_{s_i} \in W_s : {}^i t_j > \delta\}, \qquad j \in \{1, \ldots, d\}$$

where d is the number of taxels, ${}^i t_j$ is the output of the j-th taxel with respect to the i-th applied force, and δ is an experimentally determined constant parameter (i.e., a threshold) that is used for discriminating a contact response from the noise of the sensor.

At this point we have two options for computing the taxel locations starting from the force axes. The first method, described in Section 2.2.1, does not require the knowledge of the robot surface. The second method, described in Section 2.2.2, assumes that we have a mathematical description of the robot body part, or, in other words, that we only need to determine the position of each taxel on the surface.

2.2.1 *Contact point estimation without knowing the robot shape*

Let us consider an ideal scenario in which a sequence of single point contacts, occurring at different locations c_i, activate one taxel at a time. In this case it is possible to identify the location of a taxel, if more than a single measure – originating from the same taxel – is available. In fact as soon as two force axes are determined, their intersection locates the taxel. However, the F/T measures are subject to noise and this may cause errors in the position and orientation of the force axis. Moreover, the hypothesis of single contact point does not hold in practice, and an undesired moment $m_c \neq 0$ may introduce further errors in axis estimation. Furthermore, given the resolution of the skin, around 3 taxels per cm^2 in our tests (see Fig. 1.1), different contact locations can activate the same taxel and provide different contributions to the estimation of the taxel position.

One way to tackle these issues, is to collect a large number of measures (i.e., wrenches) and determine the point that best approximates the intersection of the related force axes, to finally get an estimate of the real location of the taxel. We solve this problem using a least squares procedure [Bjorck 1996]. Formally, starting from (2.4) and considering the n_j wrenches $\tilde{w}_{s_r} \in \tilde{W}_j$, $r = 1, \ldots, n_j$ corresponding to the same taxel j, we can define the following system of equations:

$$\underbrace{\begin{bmatrix} S(\frac{\tilde{f}_{s_1}}{||\tilde{f}_{s_1}||}) \\ S(\frac{\tilde{f}_{s_2}}{||\tilde{f}_{s_2}||}) \\ \vdots \\ S(\frac{\tilde{f}_{s_{n_j}}}{||\tilde{f}_{s_{n_j}}||}) \end{bmatrix}}_{[F\times]_j \in \mathbb{R}^{3n_j \times 3}} p_j = - \underbrace{\begin{bmatrix} \frac{\tilde{m}_{s_1}}{||\tilde{f}_{s_1}||} \\ \frac{\tilde{m}_{s_2}}{||\tilde{f}_{s_2}||} \\ \vdots \\ \frac{\tilde{m}_{s_{n_j}}}{||\tilde{f}_{s_{n_j}}||} \end{bmatrix}}_{M_j \in \mathbb{R}^{3n_j \times 1}}, \tag{2.5}$$

where $S(v) \in \mathbb{R}^{3\times3}$ is the operator performing the cross product $v\times$. We divided both sides of (2.4) by the module of the force, so as to avoid weighting the contribution of each axis proportionally to the module of the relative force. In the end the sum of the squares of the residual is:

$$\begin{aligned} R_j(p_j) = & \, || \, [F\times]_j \, p_j + M_j||^2 = \\ & = \sum_{r=1}^{n_j} \left(\frac{||\tilde{f}_{s_r} \times p_j + \tilde{m}_{s_r}||}{||\tilde{f}_{s_r}||} \right)^2 = \\ & = \sum_{r=1}^{n_j} \left(\frac{||\tilde{f}_{s_r} \times ({}^r\hat{p}_j + {}^r e_j) + \tilde{m}_{s_r}||}{||\tilde{f}_{s_r}||} \right)^2 = \\ & = \sum_{r=1}^{n_j} \left(\frac{|| - \tilde{m}_{s_r} + \tilde{f}_{s_r} \times {}^r e_j + \tilde{m}_{s_r}||}{||\tilde{f}_{s_r}||} \right)^2 = \\ & = \sum_{r=1}^{n_j} \left(\frac{||\tilde{f}_{s_r} \times {}^r e_j||}{||\tilde{f}_{s_r}||} \right)^2 = \sum_{r=1}^{n_j} ||{}^r e_j||^2, \end{aligned}$$

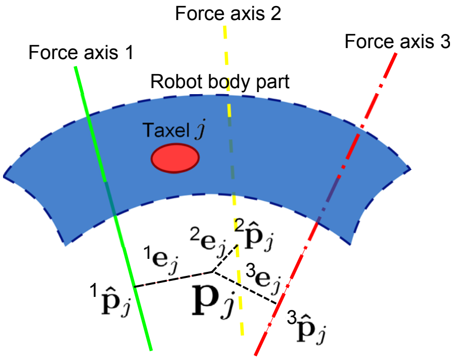


Figure 2.3: The j-th taxel estimated position p_j obtained with the Least Square Technique. In this case, the estimate is not constrained to be on the surface.

where we decomposed p_j into the sum of its projection on the r-th force axis ${}^{r}\hat{p}_j$, and its distance from the r-th force axis ${}^{r}e_j$ (see Fig. 2.3). The least squares estimate of the j-th taxel position p_j^* is the point that minimizes the sum of the squares of the distances from the force axes:

$$p_j^* = \operatorname*{argmin}_{p_j \in \mathbb{R}^3} R_j(p_j) = -\left([F\times]_j^T \, [F\times]_j \right)^{-1} [F\times]_j^T \, M_j,$$

where we assume that $[F\times]_j$ has full rank (i.e. $rank([F\times]_j) = 3$). This assumption is reasonable, given the large number of rows of the matrix, and besides it is easy to verify.

2.2.1.1 *Multi-taxel activation*

Consider now a more realistic scenario, in which each contact activates more than a single taxel at the same time. Since the taxels are quite close to each other (about 5.5 mm from the two centers) this happens most of the time. Obviously enough, certain taxels are closer than others to the actual contact point, hence they should not be treated in the same way. This aspect can be modelled by introducing a weight for each active taxel, that is proportional to its distance from the contact location. Of course the real distance between the taxels and the contact point is not known, but we do know that the closer a taxel is to the contact point, the greater its response. Formally, we can define the weight associated to the taxel j for the r-th wrench measure in $\tilde{W}_j$ as:

$${}^{r}k_j = \frac{{}^{r}t_j}{\max\limits_{y \in \{1 \ldots d\}} \left({}^{r}t_y \right)}, \tag{2.6}$$

where ${}^{r}t_y$ is the output of the y-th taxel with respect to the r-th wrench in $\tilde{W}_j$, and $\max\limits_{y \in \{1 \ldots d\}} ({}^{r}t_y)$ is the maximum of all the activated taxel outputs with respect to the r-th wrench. We construct a diagonal weighting matrix $K_j \in \mathbb{R}^{3n_j \times 3n_j}$ whose diagonal elements are ${}^r k_j$ (each appearing three times):

$$K_j = \begin{bmatrix} {}^1k_j & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & {}^1k_j & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & {}^1k_j & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & {}^{n_j}k_j & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & {}^{n_j}k_j & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & {}^{n_j}k_j \end{bmatrix}$$

Then we compute the solution of a Weighted Least Squares problem [Bjorck 1996] as:

$$p_j^* = \underset{p_j \in \mathbb{R}^3}{\operatorname{argmin}} ||K_j([F\times]_j p_j + M_j)||^2 = -\left([F\times]_j^T K_j^2 [F\times]_j\right)^{-1} [F\times]_j^T K_j^2 M_j. \tag{2.7}$$

2.2.1.2 *Reference frames*

The points p_j^* so computed represent the distances between the F/T sensor and the taxel positions, expressed w.r.t. the F/T sensor reference frame <s>, which is depicted in Fig. 2.4. If the distance between the taxels and the F/T sensor is not constant we can not use (2.7) as it is. However, if we know the rototranslation between <s> and a reference frame that is fixed to the taxels,

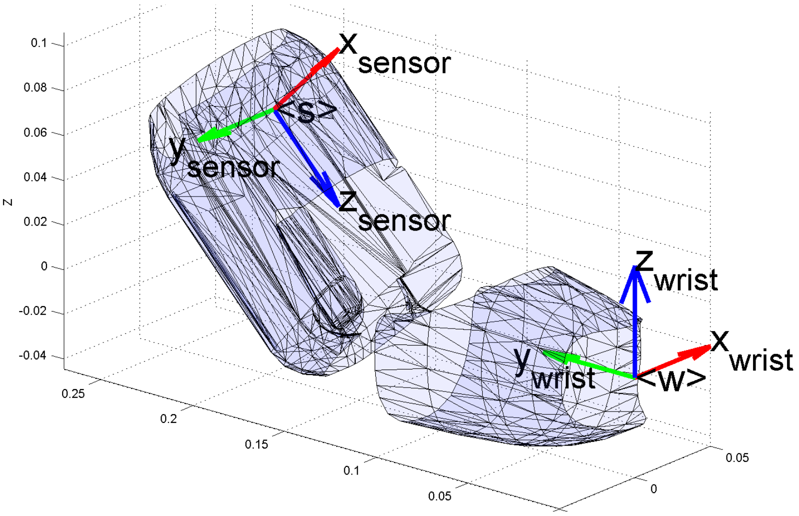


Figure 2.4: F/T sensor reference frame <s> and wrist reference frame <w>. The F/T sensor is located inside the upper part of the iCub arm, hence the two degrees of freedom of the iCub elbow affect the position of <w> w.r.t. <s>. In this figure the two joint angles of the elbow (j_3, j_4) are set to 45° and 0°. The remaining joints j_0, j_1, j_2 are located before the F/T sensor in the arm kinematic chain, hence they affect the position of the whole arm.

we can modify (2.7) to take into account this rototranslation. By doing so, wrenches collected with different arm configurations can be used at the same time in (2.5). Since we want to calibrate the tactile sensors on the forearm of the robot, we chose to express the taxel position w.r.t. the wrist reference frame <w>. Denoting with ${}^{a}v$ a vector $v \in \mathbb{R}^3$ expressed in the reference frame <a>, we represent the points on the wrist reference frame <w>:

$$ {}^{s}p = {}^{s}o_w + {}^{s}_{w}R\,{}^{w}p, \tag{2.8} $$

where ${}^{s}o_w$ is the origin of <w> expressed w.r.t. <s>, and ${}^{s}_{w}R \in \mathbb{R}^{3\times3}$ is the rotation matrix from <s> to <w>. Starting from (2.4) and using (2.8) we can write:

$$ \begin{aligned} {}^{s}f \times {}^{s}p &= -{}^{s}m \\ {}^{s}f \times {}^{s}_{w}R\,{}^{w}p &= -{}^{s}m - {}^{s}f \times {}^{s}o_w, \end{aligned} $$

where the only unknown is the taxel position ${}^{w}p$. Finally the least square problem in (2.5) can be easily reformulated starting from this new equation.

2.2.2 *Contact point estimation knowing the robot shape*

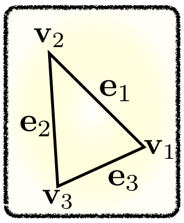


Figure 2.5: A triangle composing the surface on which the network of tactile sensors is mounted.

If we know the shape and the position of the surface where the skin is placed, we can exploit this information to constrain the taxel position estimates on that surface. A least squares problem in which the unknowns are required to satisfy a system of equality and inequality constraints may be solved using a Constrained Least Squares procedure [Bjorck 1996]. In our case, we know the mathematical description of the V triangles that form a mesh that approximates the cover of the robot arm. We can force the point to belong to a triangle by imposing four linear constraints: one equation for the plane and three inequalities for the three edges of the triangle. For every triangle v (see Fig. 2.5) we solve a constrained problem with this form:

$$ \begin{aligned} p_j^v &= \operatorname*{argmin}_{p_j \in \mathbb{R}^3} ||K_j([F\times]_j p_j + M_j)||^2 \\ s.t.\quad & (e_2^v \times e_3^v)(p_j - v_1^v) = 0 \\ & (e_2^v \times e_1^v)(e_1^v \times p_j) \le (e_1^v \times v_1^v)(e_2^v \times e_1^v) \\ & (e_3^v \times e_2^v)(e_2^v \times p_j) \le (e_2^v \times v_2^v)(e_3^v \times e_2^v) \\ & (e_1^v \times e_3^v)(e_3^v \times p_j) \le (e_3^v \times v_3^v)(e_1^v \times e_3^v) \end{aligned} \tag{2.9} $$

where e_1^v, e_2^v and e_3^v are the three edges of the v-th triangle, and v_1^v, v_2^v and v_3^v are the three vertices of the v-th triangle. In the end, the solution with the minimum sum of the squares of the residuals is chosen:

$$\begin{aligned} p_j^* &= \underset{p_j \in \mathbb{R}^3}{\operatorname{argmin}} ||K_j([F\times]_j p_j + M_j)||^2 \\ s.t. \quad & p_j \in \{p_j^v : v \in \mathbb{N}, 1 \leq v \leq V\} \end{aligned}$$

In practice, solving (2.9) for all the V triangles composing the surface mesh is computationally expensive. We can identify the subset of triangles that are most likely to result in the minimum squared residuals and solve (2.9) for this subset only. Of course these are the triangles that are closest to the force axes associated to the interested taxel. A quick way to determine these triangles is to solve (2.9) first without considering the constraints, and then consider only the triangles that are closest to the unconstrained solution.

2.3 EXPERIMENTAL RESULTS

The two proposed approaches have been tested on the skin of the right forearm of the iCub. The *single point contact* has been approximated by poking the robot skin using a metal tool, with a flat square tip. This tool has been preferred to the human finger because of its small tip (about 4x4 mm), which on average activates just three taxels at a time, as opposed to seven.

We collected several datasets, each corresponding to a different configuration of the arm. During each data collection the robot was controlled to keep a fixed position. The external wrench has been computed as the difference between the wrench measured by the F/T sensor and the constant wrench due to the weight of the arm. Several arm positions have been considered (see Table 2.1), so as to span a significant variety of arm configurations. It is worth not-

Table 2.1: Joint angles of the shoulder (j_0, j_1, j_2) and the elbow (j_3, j_4) set during the tests. For each joint we also report the range of motion.

N	DURATION	j_0 (°)	j_1 (°)	j_2 (°)	j_3 (°)	j_4 (°)
	(min)	-95°:10°	0°:160°	-37°:80°	15°:106°	-90°:90°
1	13	-10	20	20	94	0
2	5	-30	30	0	15	-90
3	4	-30	30	0	35	-40
4	7	-10	20	20	94	0
5	7	-30	30	0	94	0
6	7	-10	20	20	45	0
7	7	-10	20	20	94	-90

ing that only the two elbow joints, j_3 and j_4, affect the position of the forearm with respect to the F/T sensor reference frame.

To assess the two methods, a reference model of the taxel positions has been compared with the results, measuring the displacement of every taxel from its expected location.

2.3.1 *Single dataset*

The data collected in each test (see Table 2.1) have been processed individually with both the proposed methods, leading to quite diverse results, reported in Fig. 2.6. For the first method the mean error ranges from 8.1 mm to 18.5 mm, with the standard deviation going from 4.2 mm to 8.3 mm. For the second method the mean error ranges from 5.2 mm to 19.3 mm, with the standard deviation going from 3.1 mm to 13.1 mm. The significant differences between the various datasets can be attributed to the F/T measurements, whose precision varies depending on the magnitude of the forces and torques that are measured. These, in turns, depend on the arm configuration. By comparing the two methods it is clear that constraining the points on the skin surface generally improves the results, even if in one case (dataset 2) the opposite occurs.

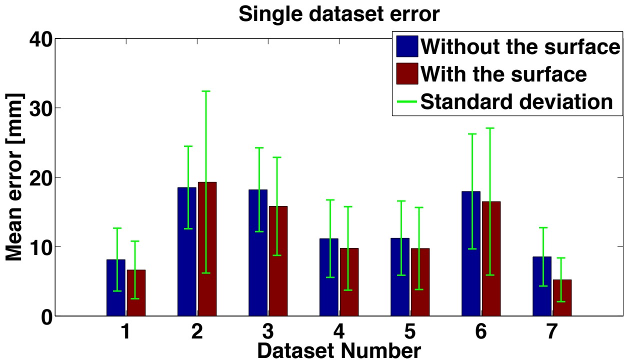


Figure 2.6: Mean and standard deviation of the error of the 84 taxel estimations, computed for every dataset. The red bars show the error of the results obtained constraining the points on the skin surface, whereas the blue bars represent the error obtained without using the surface. As expected the results improve when the surface constraint is imposed.

2.3.2 *All datasets*

If all the estimated taxel positions are represented with respect to the same reference frame, it is possible to compute their average to derive a more accurate estimation. This was computed iteratively. In other words at each step the taxel positions found using a new dataset were averaged with the current position estimations (see Fig. 2.7, 2.8). To weight each estimation we used the sum of all the weights defined in (2.6). The results of the average for both the proposed methods are depicted in Fig. 2.9. In both cases the quality of the

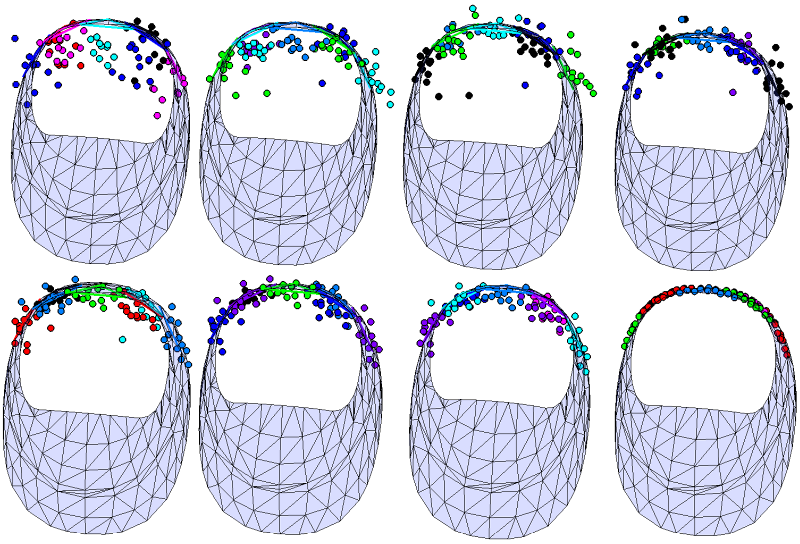


Figure 2.7: This figure shows, starting from the top-left corner, the incremental estimations of the taxel positions obtained without constraining the points on the skin surface. At each step the taxel positions found using a new dataset were averaged with the current position estimations. In the bottom-right corner there is the reference model. The forearm is seen from the front.

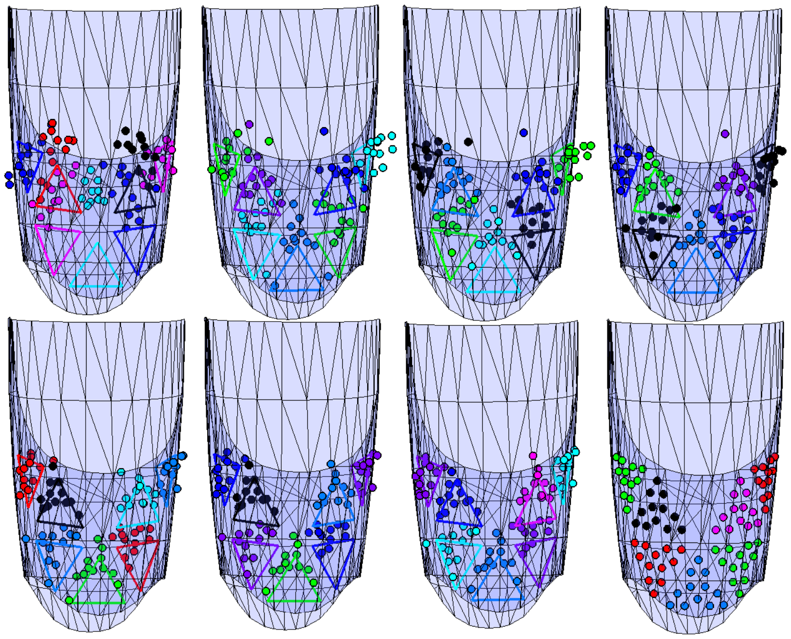


Figure 2.8: This figure shows, starting from the top-left corner, the incremental estimations of the taxel positions obtained without constraining the points on the skin surface. At each step the taxel positions found using a new dataset were averaged with the current position estimations. In the bottom-right corner there is the reference model. The forearm is seen from the top.

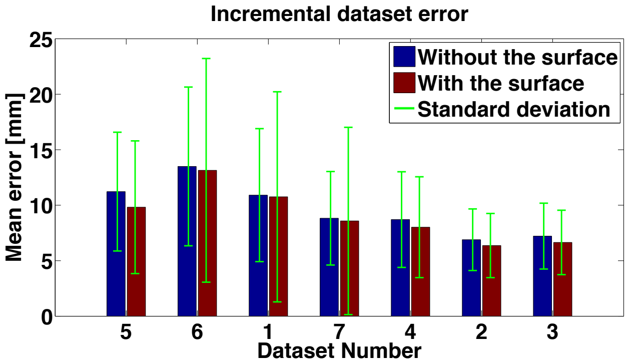


Figure 2.9: Mean and standard deviation of the error of the 84 taxel estimations computed after the estimation from each dataset is incrementally averaged. The red and blue bars show the error of the results obtained respectively with or without constraining the points on the skin surface. The estimation improves as more data are computed. The final mean error is 6.6 mm for the "surface case", and 7.2 mm for the "no surface case". The max error is 18.5 mm without the surface and 19.4 mm with the surface.

result improves as more data are added to the average, leading to a final average error of 7.2 mm for the first method and 6.6 mm for the second method. As expected the precision of the results improves when the knowledge of the surface is employed to constrain the solution of the minimization.

It is worth noticing that for the first method the final error (mean 7.2 mm, standard deviation 3 mm) is better than the best of the errors obtained from the single datasets, (mean 8.1 mm, standard deviation 4.2 mm). For the second method this did not happen, because the final error (mean 6.6 mm, standard deviation 2.9 mm) is worse than the error obtained with the single dataset 7 (mean 5 mm, standard deviation 3.1 mm). Nonetheless, considering the wide range of errors resulted from the single datasets – up to 19.3 mm – the error obtained merging all the estimations can still be regarded favorably.

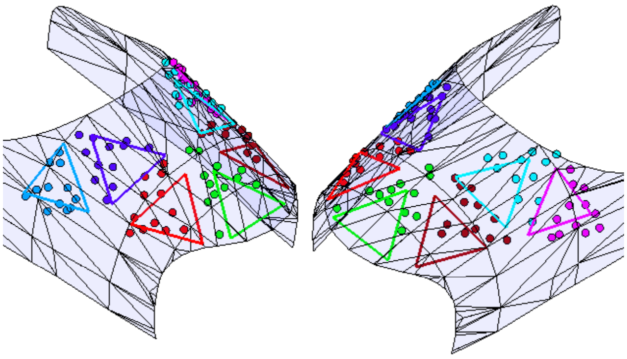


Figure 2.10: Final taxel position estimations obtained constraining the points on the skin surface. As reference, the edges of the seven skin triangles are drawn over the forearm surface. The same colors are used to draw the triangles and the corresponding (estimated) taxels. Two views of the same result are depicted, to better show the taxel positions.

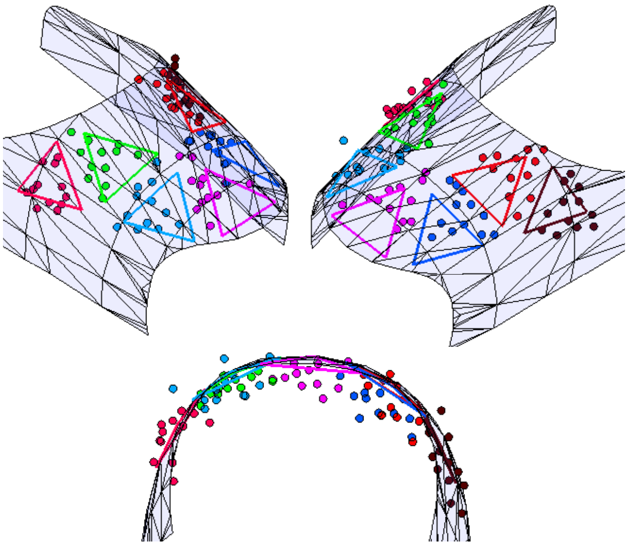


Figure 2.11: Final taxel position estimations obtained without constraining the points on the skin surface. As reference, the edges of the seven skin triangles are drawn over the forearm surface. The same colors are used to draw the triangles and the corresponding (estimated) taxels. Three views of the same result are depicted, to better show the taxel positions.

Finally, to better visualize the results, in Fig. 2.10 and 2.11 we plot the estimated taxel positions obtained using all the datasets, respectively with and without constraining the points on the arm surface.

2.3.3 *Error analysis*

To complete the assessment of the methods, some words should be spent regarding the sources of error. The quality of the taxel position estimation depends on several aspects, and most of them resides in the robotic platform rather than in the method itself. First of all, errors in the force and torque measurements, respectively ε_m and ε_f, directly affect the estimate, turning equation (2.4) into:

$$m_s + \varepsilon_m = \hat{c} \times (f_s + \varepsilon_f).$$

Hence the estimated force axis is equal to:

$$\hat{c} \in \left\{ \frac{(f_s + \varepsilon_f) \times (m_s + \varepsilon_m)}{||f_s + \varepsilon_f||^2} + \lambda(f_s + \varepsilon_f) \right\} = \hat{r}.$$

Clearly if ε_m and ε_f are too large the estimated force axis will be too distant from the contact point, resulting in a large error in the taxel position estimations. We can compute the distance between the estimated force axis $\hat{r}$ and the real contact point c as a function of the force and torque measure errors:

$$d(c, \hat{r}) = \frac{||(f_s + \varepsilon_f) \times (c - \frac{(f_s+\varepsilon_f)\times(m_s+\varepsilon_m)}{||f_s+\varepsilon_f||^2})||}{||f_s + \varepsilon_f||}.$$

Another contribution to the error comes from the elasticity of the robotic structure that damps the applied forces and torques thus altering the measurements. Although it can be compensated, at the time of this work, a model of the elasticity of the robot was unavailable and the measurements could not be adjusted.

Uncertainties in the kinematic model of the robot affect the position estimate too. As it can be seen in (2.8), to use all the datasets taken in different poses of the robot arm, the taxel positions have to be expressed w.r.t. a common fixed reference frame. An error in the transformation between the F/T sensor reference frame and the common reference leads to a wrong estimate of the taxel position.

The proposed approach makes some hypotheses that do not hold completely in a real scenario. Although the robot has been poked with a small tip tool, the contact type is surely not a single point contact, thus introducing further uncertainty in the localization. Moreover, the method estimates the taxel positions by trying to find the point that best approximates the intersection among axes of different forces. If the axes are almost parallel the problem is ill-posed, thus much more sensitive to errors in the input data.

2.4 CONCLUSIONS AND FUTURE WORK

This chapter describes two methods for the estimation of the position of tactile sensing elements on robot body parts. Differently from previous works [Kuniyoshi et al. 2004; Modayil 2010; Noda et al. 2008], these methods estimate not only the network topology, but also the location of each sensor with respect to a known reference frame. This is a crucial point if the sensor data have to be integrated with other information (e.g. kinematics, force/torque measurements) for controlling the robot or estimating its state.

The first presented approach does not require the knowledge of the robot surface. By averaging the results of the estimation from different arm configurations, this procedure allows to determine the position of each taxel with an average error of 7.2 mm. This can be further reduced to 6.6 mm in case the shape of the robot surface is known, as it can be appreciated qualitatively by looking at the reconstruction in Fig. 2.10. These results are remarkable, considered that the calibration requires few hypotheses and that it is carried out on a real robot, using noisy data (i.e. the robot kinematics and the F/T sensor measures). The second approach assumes that a mathematical description of the surface on which the skin is placed is available, so as to constrain the taxel position estimations on it. This assumption is reasonable, since the surface of the robot can be derived once from the CAD model of the robot, and it is independent of how the skin is mounted.

The average calibration error is about 7 mm.

As it can be noticed in Fig. 2.11 the contact areas are not localized precisely, but the neighboring relationships between taxels are preserved (i.e. taxels that are neighbors in the real sensor network are neighbors in the network reconstruction too). The precision achieved is sufficient for localizing the contact points and estimating the external contact forces, as it will be explained in

The calibrated tactile sensors allowed us to estimate contact forces and implement force control paradigms.

chapter 3. This is important information that has been used to implement force control, parallel control and hybrid control, as reported in chapter 4.

The causes of the errors, as discussed in Section 2.3.3, are mainly related to noisy force/torque measurements and to interaction events that involve more than a single contact point. In the reported experiments, the interaction forces were produced manually using a tool with a small tip to make sure that the contact activated a small number of taxels. As a result, the data collection was quite tedious and time consuming. We plan to make the calibration completely autonomous. To enforce the hypothesis that contacts occur only at single points it is possible to proceed along two directions. First, by checking that contacts indeed occur only at few (or even just one) taxels, and filter all the contact events that do not satisfy this hypothesis. Second, by maximizing the chance that this kind of events occurs. Along the lines of [Cannata et al. 2010], we are planning to implement a motion control law that controls the robot so that it comes into contact with a small, fixed object. In this scenario both the proposed calibration methods are applicable. The advantage of an autonomous procedure is that it would allow to collect a larger amount of data, which would help to improve the quality of the final results.

Towards an autonomous calibration. . .

We are also investigating the possibility to add other constraints to the problem formulation in order to improve the precision of the final results. Instead of specifying the exact surface on which the skin is mounted, we could simply constrain that surface to be smooth. Moreover, since we know that the distance between neighboring taxels is within certain limits (approximately 4 mm to 6 mm), we could use this information as a constraint.

Adding constraints to improve the results.

Finally, the 84 taxels located on the upper part of the iCub forearm sufficed to test the presented approaches. Nonetheless, the iCub is equipped with a much larger amount of tactile elements, covering the arms and torso, and even more are going to be mounted on it. The proposed calibration procedure has been used on all the tactile sensors located on the iCub's arm (which are about 1500 taxels). The tactile sensors on the torso of the robot has not been calibrated yet because, at the moment, it is not possible to measure contact forces applied on the iCub's torso.

3 CONTACT FORCE ESTIMATIONS USING TACTILE SENSORS AND FORCE/TORQUE SENSORS

In this chapter we present a numerical method, based on the recursive Newton-Euler algorithm (RNEA), to estimate forces and moments applied to a robotic chain. The method uses a six axis force/torque (F/T) sensor, located at the base of the kinematic chain, together with a tactile sensor network, which covers most of the surface of the robot. The tactile sensors measure the contact locations, whereas the F/T sensor measures the magnitude and the direction of the contact forces. We show that the number of contacts that is possible to estimate reliably is strictly dependent on the amount of information retrieved from the sensors. When this critical number is exceeded, infinite solutions satisfy the estimation problem, though, in some cases, we can choose one of the possible solutions and draw the probability distribution of its error.

3.1 INTRODUCTION

In most robotics applications, robots come into contact with the environment just with their end-effectors. Nevertheless, robots could benefit from making contacts with other parts of their bodies, in the same way humans do when performing tasks such as writing, carrying heavy objects, or balancing.

This lack is mainly due to the fact that nowadays most robots do not have an artificial skin that allows them to detect and localize contacts. Either joint torque sensors or 6 axis F/T sensors are usually used to provide contact feedback to robots. However, with these sensors it is not possible to retrieve the exact contact location, unless we make strong assumptions about the contact (e.g. zero moment applied, known force direction). Moreover, the wrench (i.e. force and moment) that is applied at the contact point cannot be measured, unless the contact location is somehow known. Often these applications focus on controlling the joint torques rather than the contact wrenches, but this approach is not generally applicable because it does not control the interaction wrenches.

Robots interaction capabilities are limited by the lack of tactile sensing.

In the end, force control applications suffer from at least one of these limitations:

- the contact point has to be fixed and known a priori (usually the end-effector)
- the geometry of the robot and the environment have to be known
- it is not possible to control the contact wrenches, but just the joint torques

Sentis et al. [2009] presented a theoretical framework to model and control robots that are subject to multiple contacts, but the authors do not discuss how to localize the contacts and estimate the contact forces/pressures. Park and Khatib [2005] presented a compliant motion control framework for multiple contacts and they tested it with a PUMA560 manipulator. In the experiments the geometry and the stiffness of the environment are assumed to be known a priory in order to compute the contact points and the contact forces. A probabilistic approach has been proposed in [Petrovskaya et al. 2007], in which the authors used an active sensing strategy to estimate at the same time the shape of the robot and the contact point. Nevertheless, they tested the method only with extremely simple geometries of robot and environment (the environment was a point and the robot link was a line) and the authors said that, for more complex geometries, more sophisticated active exploration strategies would likely be needed.

When both tactile sensors and F/T sensors are available, we can estimate contact locations and contact wrenches, and so implement reliable contact force control. In this paper we present a method to estimate internal and external wrenches that is based on the well-known *Recursive Newton-Euler Algorithm* (RNEA) [Siciliano and Khatib 2008]. We implemented the method as an extension of the library *iDyn* [Ivaldi et al. 2011], and we tested it on the iCub humanoid robot [Metta et al. 2008]. We assume to know the kinematics and dynamics parameters of the robotic chain and the position of each tactile sensor (we calibrated the tactile sensors using the method described in chapter 2). Also the joint positions, velocities and accelerations are assumed to be known. We computed joint velocities and accelerations through numerical differentiation of the joint positions, using the first-order adaptive windowing method described in [Janabi-Sharifi 2000]. The first step of the proposed algorithm is unmodified with respect to the standard RNEA: it computes velocities and accelerations of all the links starting with the known velocity and acceleration of the chain base (either the base is fixed or an inertial sensor is necessary). In the second step we solve a system of linear equations to estimate the contact wrenches. Finally the classic recursive wrench propagation is computed in order to compute internal wrenches and joint torques.

Section 3.2 explains how to build and solve the linear system for estimating the contact wrenches. Section 3.3 discusses the capabilities and the limitations of the presented method, looking at some future extensions.

3.2 METHOD

Let us consider a kinematic chain composed by N links, having a F/T sensor at the base (see Fig. 3.1), where w_i is the wrench (i.e. a six dimensional vector containing a force and a moment) exerted from link i to link i+1, $\ddot{p}_{c_i}$ is the acceleration of the center of mass of link i and m_i is the mass of link i. Note that $\ddot{p}_{c_i}$ contains also the gravity acceleration. We know w_0 (i.e. the F/T sensor measure), the contact locations $r_{0,ei}$, and we want to estimate the K contact wrenches $w_{e1}, \dots, w_{eK}$.

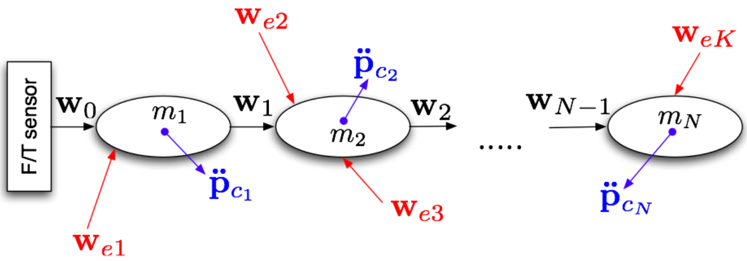


Figure 3.1: Generic kinematic chain with F/T sensor at the base.

Writing Newton's equation for each link and recursively substituting the internal forces we get:

$$f_0 + \sum_{i=1}^{K} f_{ei} = \sum_{i=1}^{N} m_i \ddot{p}_{c_i} \tag{3.1}$$

We can do the same with Euler's equation:

$$\mu_0 + \sum_{i=1}^{K} (\mu_{ei} + r_{0,ei} \times f_{ei}) = \sum_{i=1}^{N} (r_{0,c_i} \times m_i \ddot{p}_{c_i} + I_i^i \dot{\omega}_i + \omega_i \times I_i^i \omega_i), \tag{3.2}$$

where I_i^i is the inertia of link i, ω_i and $\dot{\omega}_i$ are the angular velocity and acceleration of link i, and r_{0,c_i} is the vector connecting the chain base to the center of mass of link i. Given that the joint positions, velocities and accelerations are assumed to be known, the only unknowns in (3.1) and (3.2) are the contact wrenches. The estimation problem may be solved rewriting these equations in matrix form $Ax = b$, where $x \in \mathbb{R}^u$ contains all the u contact unknowns, whereas $A \in \mathbb{R}^{6\times u}$ and $b \in \mathbb{R}^6$ are completely determined. For instance, in case of two external contact wrenches, the system is:

$$\begin{bmatrix} I & 0 & I & 0 \\ S(r_{0,e1}) & I & S(r_{0,e2}) & I \end{bmatrix} \begin{bmatrix} f_{e1} \\ \mu_{e1} \\ f_{e2} \\ \mu_{e2} \end{bmatrix} = \begin{bmatrix} -f_0 + \sum_{i=1}^{N} m_i \ddot{p}_{c_i} \\ -\mu_0 + \sum_{i=1}^{N} (r_{0,c_i} \times m_i \ddot{p}_{c_i} + I_i^i \dot{\omega}_i + \omega_i \times I_i^i \omega_i) \end{bmatrix}$$

where $S(v) \in R^{3\times 3}$ is the operator performing the cross product $v\times$. Building the system we may consider three different types of contacts, depending on our priors:

WRENCH : 6 unknowns, w_e, no priors

PURE FORCE : 3 unknowns, f_e, the moment is known (usually it is supposed to be zero)

FORCE MODULE : 1 unknown, $||f_e||$, both the force direction and the moment are known

The *pure force* contact can be used to reduce the number of unknowns when the contact area is considered so small that almost no moment can be applied. The *force module* contact can be used if the tactile sensors can measure the contact force direction.

3.2.1 *Building A*

The matrix A is built by adding columns for each contact. For every unknown wrench n we add 6 columns to A:

$$\begin{bmatrix} I & 0 \\ S(r_{0,en}) & I \end{bmatrix}$$

For every unknown pure force n we add 3 columns to A:

$$\begin{bmatrix} I \\ S(r_{0,en}) \end{bmatrix}$$

For every unknown force module n we add 1 column to A:

$$\begin{bmatrix} \hat{u}_n \\ r_{0,en} \times \hat{u}_n \end{bmatrix}$$

where $\hat{u}_n$ is the versor of the contact force f_{en}, defined as $\hat{u}_n = \frac{f_{en}}{||f_{en}||}$.

3.2.2 *Building b*

The 6 dimensional vector b is defined as:

$$b = \begin{bmatrix} f_b \\ \mu_b \end{bmatrix} = \begin{bmatrix} f_{eTot} \\ \mu_{eTot} - \mu_{eKnown} \end{bmatrix}$$

where μ_{eKnown} is the sum of the known external moments applied to the robotic chain (usually zero), whereas f_{eTot} and μ_{eTot} are defined as:

$$\begin{aligned} f_{eTot} &= -f_0 + \sum_{i=1}^{N} m_i \ddot{p}_{c_i} \\ \mu_{eTot} &= -\mu_0 + \sum_{i=1}^{N} (r_{0,c_i} \times m_i \ddot{p}_{c_i} + I_i^i \dot{\omega}_i + \omega_i \times I_i^i \omega_i) \end{aligned}$$

3.2.3 *Solving the system*

Once A and b have been computed, we can distinguish two cases. If the number of unknowns is less than or equal to the rank of A, then there is a unique x^* that minimizes the square error residual:

$$x^* = \operatorname*{argmin}_{x \in \mathbb{R}^u} ||Ax - b||^2$$

On the other hand, if the number of unknowns is bigger than the rank of A then the system admits infinite solutions. Unless we have some priors about the external wrench distribution, a reasonable choice is to select the solution that minimizes the norm of x, which is the solution that equally distributes the total external wrench among all the external contacts.

$$\begin{aligned} x^* &= \underset{x \in \mathbb{R}^u}{\operatorname{argmin}} ||x||^2 \\ s.t. \quad Ax &= b \end{aligned}$$

In both cases the solution x^* may be computed as:

$$x^* = A^+ b$$

where A^+ is the Moore-Penrose pseudoinverse of A.

The method has been implemented as part of the *iDyn* library (called *iDyn-Contact*) and it has been integrated with other software modules to create an efficient software system to estimate internal and external wrenches of the whole *iCub* robot. The estimated contact wrenches, together with a model of the robot, can be depicted in real time in a gui, as it can be seen if Fig. 3.2.

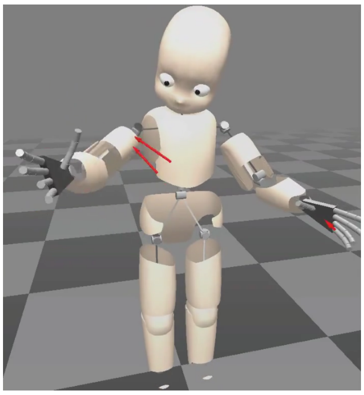


Figure 3.2: Gui depicting the iCub robot and the estimations of the contact forces as red arrows.

3.3 DISCUSSION AND CONCLUSIONS

When only one contact is detected, the presented method can easily estimate the contact wrench. When two contacts are detected the number of unknowns in the linear system is twice the number of equations, so the contact wrenches are poorly estimated. To reduce uncertainties, if the contact areas are small, it is reasonable to assume that the applied moments are zero, so that the number of unknowns drops from 12 to 6. Unfortunately, when we do this, the rank of A drops from 6 to 5 (because the cross product matrix is rank deficient), so the system still admits infinite solutions and it takes this form:

$$\begin{bmatrix} I & I \\ S(r_{0,e1}) & S(r_{0,e2}) \end{bmatrix} \begin{bmatrix} f_{e1} \\ f_{e2} \end{bmatrix} = \begin{bmatrix} -f_0 + \sum_{i=1}^{N} m_i \ddot{p}_{c_i} \\ -\mu_0 + \sum_{i=1}^{N} (r_{0,c_i} \times m_i \ddot{p}_{c_i} + I_i^i \dot{\omega}_i + \omega_i \times I_i^i \omega_i) \end{bmatrix}$$

However, we carried out a numerical and analytical analysis, simulating random forces with norm uniformly distributed in [0, K] Newton. We found out that the norm of the error of the estimated forces was distributed as a *unilateral Gaussian*, with zero mean and standard deviation equal to about $K/4$. That is equivalent to say that the error of the estimate of the two forces, in norm, is less than $K/4$ with probability 0.68, and less than $K/2$ with probability 0.95.

Whenever more than two contacts are detected, it is impossible to get a reliable estimate of the contact wrenches without imposing some constraints to the system. Noticing that contact forces are almost always directed towards the robot (i.e. pushing) and quasi-normal to the robot surface, we may constrain the force estimations to lie inside a cone built around the normal of the contact surface.

Additional information regarding the force direction and magnitude may be retrieved through the tactile sensors. For instance, with appropriate technology we may be able to estimate the directions of the contact forces with the tactile sensors. In this case the only unknown left would be the force intensity, so up to six contact forces could be estimated *reliably* (considering zero contact moments).

In the next year, we plan to extend this estimation method in two directions: types of sensors and estimated quantities. Currently, the estimation process exploits only 6-axis force/torque sensors and tactile sensors, but other sensors have recently been — or are soon going to be — mounted on the iCub, such as accelerometers, gyroscopes, and joint torque sensors. The mathematical method used for estimating contact forces is going to be generalized to consider this additional information, and to estimate other quantities, such as velocities and accelerations of joints and base of the robot. In particular, for each joint torque sensor we can add a linear equation to the system, increasing the number of unknowns that can be estimated reliably.

Part II

CONTROL

4

CONTROL OF CONTACT FORCES: THE ROLE OF TACTILE FEEDBACK FOR CONTACT LOCALIZATION

This chapter investigates the role of precise estimation of contact points in force control. This analysis is motivated by scenarios in which robots make contacts, either voluntarily or accidentally, with different parts of their body. Control paradigms that are usually implemented in robots with no tactile system, make the hypothesis that contacts occur at the end-effectors only. In this chapter we try to investigate what happens when this assumption is not verified. First we consider a simple feedforward force control law, and then we extend it by introducing a proportional feedback term. For both controllers we find the error in the resulting contact force, that is induced by a hypothetic error in the estimation of the contact point. We show that, depending on the geometry of the contact, incorrect estimation of contact points can induce undesired joint accelerations. We validate the presented analysis with tests on a simulated robot arm. Moreover we consider a complex real world scenario, where most of the assumptions that we make in our analytical derivation do not hold completely. Through tests on the iCub humanoid robot we see how errors in contact localization affect the performance of a parallel force/position controller. In order to estimate contact points and contact forces on the forearm of the iCub we do not use any model of the environment, but we exploit its 6-axis force/torque sensor and its sensorized skin.

4.1 INTRODUCTION

After decades of research in robotics, robots are still a long way off being able to operate in human environments. The inability to deal with uncertainties in the geometry of the environment is maybe the major limitation that prevents robots from safely interacting with humans.

Research has tried to tackle this problem at the control level. Traditional stiff position control tries to follow a desired position trajectory considering external forces as disturbances, hence large contact forces may be exerted, leading to instability or physical damage. Controlling the interaction forces has been shown to be a powerful technique to overcome these limitations. In the literature we can find different approaches to force control: explicit force control, impedance/admittance control [Hogan 1985], hybrid control [Khatib 1987], hybrid impedance control [Anderson and Spong 1988], parallel control [Chiaverini and Sciavicco 1993].

Nevertheless, most of these works have only focused on controlling forces that are exerted at the end-effectors, namely the hands for manipulation tasks and the feet for walking. Restricting contacts to end-effectors is a quite strong assumption, since i) uncertainties in the environment may result in unplanned contacts at other body parts, and ii) contacts at other body parts may be necessary to perform certain tasks. If you are reading this thesis, for instance, you are likely to be making contact with the environment with at least four points that are not end-effectors: bottom and back against the chair, and elbows on the desk.

Some works do not make any assumption about the contact location, trying to control the joint torques rather than the contact forces. This approach ensures bounded contact forces, but do not allow a real contact force control. Various theoretical frameworks for modeling and control of robots that are subject to multiple contacts have been proposed [Park and Khatib 2006; Righetti et al. 2011a; Sentis et al. 2009], but they have not gone beyond end-effector contacts when tested on real platforms.

What has really prevented contact force control from spreading in robotics is the lack of robust contact force measurements, that is the ability to measure multiple contact forces and their locations on the robot's body. Gordon et al.[Gordon and Townsend 1989] have verified that it is not possible to obtain high-resolution contact-position measurements from only joint torque sensing. In this scenario a tactile system becomes paramount.

Despite these difficulties, in the literature we can find some examples of contact force control. In [Park and Khatib 2005], Park et al. presented the first implementation of multi-link multi-contact force control, with demonstration on a 6-DOF PUMA560 manipulator. The robot was able to control three contact forces, distributed on its end-effector and third link, while motion was controlled in the remaining three DOFs through null space control. Due to the limited sensing capabilities of the test platform, the measure of contact forces and contact locations required i) an external force sensor mounted on one of the contact points and ii) geometric models of both robot and environment. Relying on geometric models is risky because modeling uncertainties may cause the controller to perform inconsistently in response to small errors.

In [Jain et al. 2011], the authors describe a model predictive controller (MPC) that allows a robot arm to reach with its end-effector, while regulating contact forces across its entire surface. This work exploits tactile sensing, it does not use a model of the environment and it handles contacts at unpredictable locations.

Motivated by works such as [Park and Khatib 2005] and [Jain et al. 2011], in this chapter we evaluate analytically the impact of errors in the contact point estimation on the performance of contact force control. Even though the effect that errors in the estimation of contact location have on contact force control is not surprising, to the best of our knowledge this is the first analytical analysis of their relationship. First, in section 4.2.1, we consider a simple feedforward force control law and we derive the equations relating the error in the estimation of the contact point with the errors in the contact force and the joint accelerations. Then, in section 4.2.2, we consider a proportional feedback force control law and we repeat the same analysis. Finally, we extend our analysis to a complex real world case, where most of the assumptions that we make in the analytical derivation do not hold. In section 4.2.3

we describe a parallel control law that is used to carry out an empirical analysis on the iCub humanoid robot. Section 4.3 reports the results of the tests, both in simulation and on the real robot. The importance of tactile sensing is confirmed by both our analytical analysis and our empirical results, which demonstrate a clear improvement in force and position errors. Finally section 4.4 discusses the results of the tests and future work.

4.2 CONTACT FORCE CONTROL

4.2.1 *Feedforward Force Control with Contact Point Estimation Error*

In this section we compute the error in the wrench that is controlled at the contact point, resulting from an error in the estimation of the position of the contact point. Consider the equation of motion of a rigid manipulator in contact with the environment [Siciliano and Khatib 2008]:

$$M(q)\ddot{q} + h(q,\dot{q}) - J_c(q)^T w = \tau, \tag{4.1}$$

where $q \in \mathbb{R}^n$ is the vector of joint coordinates, $\tau \in \mathbb{R}^n$ is the vector of joint torques, $M(q) \in \mathbb{R}^{n\times n}$ is the joint space inertia matrix, $h(q,\dot{q}) \in \mathbb{R}^n$ is the vector containing all the non-linear terms (e.g. Coriolis, centrifugal and gravity terms), $J_c(q) \in \mathbb{R}^{6\times n}$ is the Jacobian of the contact point and $w^T = \begin{bmatrix} f^T & \mu^T \end{bmatrix} \in \mathbb{R}^6$ is the contact wrench vector. In the following, dependency upon q and $\dot{q}$ is no longer denoted to simplify notation. We assume that the manipulator is in rigid contact with the environment and that there are $k \leq 6$ constrained directions (i.e. $rank(J_c) = k$), hence:

$$\begin{aligned} \dot{x}_c &= J_c\dot{q} = 0 \\ \ddot{x}_c &= J_c\ddot{q} + \dot{J}_c\dot{q} = 0 \end{aligned}$$

Under these assumptions, the contact wrench w can be computed as [Mistry et al. 2010]:

$$\begin{aligned} w &= (J_c M^{-1} J_c^T)^{-1}(J_c M^{-1}(h-\tau) - \dot{J}_c\dot{q}) = \\ &= \Lambda_c(J_c M^{-1}(h-\tau) - \dot{J}_c\dot{q}), \end{aligned} \tag{4.2}$$

where we defined the constraint space inertia matrix as $\Lambda_c = (J_c M^{-1} J_c^T)^{-1}$. Consider a simple feedforward force control law:

$$\tau^* = -J^T w_d + \tilde{h} - J_c^T \tilde{\Lambda}_c \dot{J}_c \dot{q}, \tag{4.3}$$

where $\tau^* \in \mathbb{R}^n$ are the control torques, $w_d^T = \begin{bmatrix} f_d^T & \mu_d^T \end{bmatrix} \in \mathbb{R}^6$ is the desired contact wrench, $\tilde{h}$ and $\tilde{\Lambda}_c$ are the estimates of h and Λ_c, and $J \in \mathbb{R}^{6\times n}$ is another Jacobian of the contact point that considers also the non-constrained directions. Assuming perfect modeling of the dynamic parameters (i.e. $\tilde{h} = h$, $\tilde{\Lambda}_c = \Lambda_c$), perfect tracking of the joint torques (i.e. $\tau = \tau^*$), and assuming that the contact geometry allows the manipulator to exert wrench in the desired direction (i.e. $J^T w_d = J_c^T w_d$), this control law results in $w = w_d$ (it is easy to prove it by substituting (4.3) into (4.2)).

In order to simplify the analysis, in the following we will assume that the manipulator moves slowly, so that $\dot{J}_c\dot{q} \approx 0$. This assumption is reasonable if we add a damping term to our control law, which then becomes:

$$\tau^* = -J^T w_d + \tilde{h} - k_d\dot{q}, \tag{4.4}$$

where $k_d > 0$ is a scalar gain. Suppose now that there is an error $e_p \in \mathbb{R}^3$ in the estimate of the contact position and an error $e_o \in \mathbb{R}^3$ in the estimate of the contact orientation:

$$\bar{p} = p + \begin{bmatrix} e_p \\ e_o \end{bmatrix},$$

where $p \in \mathbb{R}^6$ is the real contact point and $\bar{p} \in \mathbb{R}^6$ is the estimated contact point. Assuming that p and $\bar{p}$ are on the same link, their Jacobians are related by this equation:

$$\bar{J} = \begin{bmatrix} I_{3\times3} & -[e_p\times] \\ 0_{3\times3} & I_{3\times3} \end{bmatrix} J = S(e_p)J,$$

where $[e_p\times] \in \mathbb{R}^{3\times3}$ is the cross product matrix parametrized by the vector e_p. The torques computed by the control law (4.4) are:

$$\begin{aligned} \bar{\tau}^* &= -\bar{J}^T w_d + \tilde{h} - k_d\dot{q} \\ &= -J^T S(e_p)^T w_d + \tilde{h} - k_d\dot{q} \\ &= -J^T \left(w_d + \begin{bmatrix} 0_3 \\ e_p \times f_d \end{bmatrix} \right) + \tilde{h} - k_d\dot{q} \\ &= -J^T w_d - J_a^T e_p \times f_d + \tilde{h} - k_d\dot{q} \\ &= \tau^* - J_a^T e_p \times f_d \end{aligned} \tag{4.5}$$

where $J_a \in \mathbb{R}^{3\times n}$ is the angular part of the contact point Jacobian $J^T = \begin{bmatrix} J_l^T & J_a^T \end{bmatrix}$. Substituting (4.5) into (4.2) we can compute the contact wrench at steady-state (i.e. when $\dot{q} = 0$):

$$\begin{aligned} w &= \Lambda_c J_c M^{-1}(J^T w_d + J_a^T e_p \times f_d - \tilde{h} + h) \\ &= w_d + \Lambda_c J_c M^{-1} J_a^T e_p \times f_d \\ &= w_d + J_c^{T+} J_a^T e_p \times f_d \\ &= w_d + e_{wff} \end{aligned} \tag{4.6}$$

where $J_c^{T+} = \Lambda_c J_c M^{-1}$ is the dynamically-consistent pseudo-inverse of J_c^T [Khatib 1987]. Substituting (4.6) and (4.5) into (4.1) we can compute the joint accelerations:

$$\ddot{q} = M^{-1}(\tau + J_c^T w - h) = -M^{-1}(I - J_c^T J_c^{T+}) J_a^T e_p \times f_d = -M^{-1} N_c J_a^T e_p \times f_d \tag{4.7}$$

Eq. (4.6) and (4.7) clearly express the relationship between the contact point estimation error, e_p, and the errors resulting from the control law (4.3). These equations are quite complex, but we can note that:

1. both e_{wff} and $\ddot{q}$ contain the term $J_a^T e_p \times f_d$, which is the error induced by e_p in the commanded torques
2. the part of $J_a^T e_p \times f_d$ that is selected by J_c^{T+} affects the contact wrench, whereas the remaining part, selected by the null space projection matrix of J_c, generates joint accelerations
3. both e_{wff} and $\ddot{q}$ depend on the desired force f_d, but they are independent of the desired moment μ_d (this makes sense since the relationship between the contact moment and the joint torques does not depend on the contact point);
4. if the desired force f_d is parallel to the contact point estimation error e_p then both e_{wff} and $\ddot{q}$ are zero;
5. in case we cannot have a precise estimate of the contact point, but we can set an upper bound on $||e_p||$, then it is possible to compute an upper bound of $||e_{wff}||$ and $||\ddot{q}||$ as a function of the maximum norm of f_d and the norm of the dynamically-consistent pseudo-inverse of the contact point Jacobian (which depends on the robot kinematics and dynamics parameters).

4.2.2 *Feedback Force Control with Contact Point Estimation Error*

While in theory a feedforward control law such as (4.4) should achieve the desired contact force, in practice we know that this is not true. Different sources of error (e.g. dynamic parameters, actuator dynamics, contact point estimation) affect the control action, resulting in errors in the controlled quantity. Often these unknown errors are modeled as an additive disturbance $d \in \mathbb{R}^n$, acting at joint torque level (see Fig. 4.1):

$$\tau = \tau^* + d,$$

where $\tau^* \in \mathbb{R}^n$ are the desired joint torques. Applying the control law (4.4),

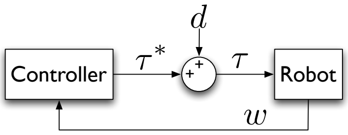


Figure 4.1: Scheme representing the relationship between the disturbance d, the control action τ^* and the joint torques τ.

the resulting contact wrench is then affected by the disturbance d:

$$w = w_d - J_c^{T+} d$$

In particular, we have seen in the previous section that in case of an error in the contact localization, the resulting disturbance at joint torque level is $d = -J_a^T e_p \times f_d$. In general, assuming that we can measure the contact wrench w, we can then reduce the effect of the disturbance d by introducing a proportional feedback term in the control law, so that it becomes:

$$\tau^* = -J^T(w_d + k_p(w_d - w)) + \tilde{h} - k_d\dot{q}, \tag{4.8}$$

where $k_p > 0$ is a scalar proportional gain. Using this new control law the contact wrench becomes:

$$\begin{aligned} w &= w_d + J_c^{T+} J^T k_p(w_d - w) - J_c^{T+} d \\ (1+k_p)w &= (1+k_p)w_d - J_c^{T+} d \\ w &= w_d - \frac{1}{1+k_p} J_c^{T+} d, \end{aligned}$$

where we used the fact that $J^T w_d = J_c^T w_d$ and $J^T w = J_c^T w$. The first is the same assumption that we made in the previous section, whereas the second is always true since, by definition, wrench can be applied in the constrained directions only. By tuning k_p we can reduce the effect of the disturbance as much as we like. This proves that a feedback term is able to reduce the effect of errors that can be modeled as additive input disturbances.

Now let us analyze whether a feedback term may be of help in case of errors in the estimation of the contact point. Suppose, as before, that there is an error e_p in the estimate of the contact point. Besides affecting the Jacobian of the contact point, e_p may induce an error in the wrench feedback too. In particular, since we measure the contact wrench using a 6-axis F/T sensor (we follow the procedure described in [Fumagalli et al. 2010]), the wrench measurement can be expressed as:

$$\bar{w} = w - \begin{bmatrix} 0_3 \\ e_p \times f \end{bmatrix} = w - e_w$$

We can compute the commanded torques:

$$\begin{aligned} \bar{\tau} &= \bar{J}^T(w_d - k_p(\bar{w} - w_d)) + \tilde{h} - k_d\dot{q} \\ &= J^T S(e_p)^T (w_d - k_p(w - e_w - w_d)) + \tilde{h} - k_d\dot{q} \\ &= J^T(w_d + e_{w_d} - k_p(w - w_d - e_{w_d})) + \tilde{h} - k_d\dot{q} \\ &= (1+k_p)J^T(w_d + e_{w_d}) - k_p J^T w + \tilde{h} - k_d\dot{q} \end{aligned}$$

where $e_{w_d}^T = \begin{bmatrix} 0_3^T & (e_p \times f_d)^T \end{bmatrix}$. As before, we can compute the contact wrench at steady-state using (4.2):

$$\begin{aligned} w &= \Lambda_c J_c M^{-1}((1+k_p)J^T(w_d + e_{w_d}) - k_p J^T w) \\ (1+k_p)w &= (1+k_p)J_c^{T+} J^T(w_d + e_{w_d}) \\ w &= w_d + J_c^{T+} J^T e_{w_d} \\ w &= w_d + J_c^{T+} J_a^T e_p \times f_d, \end{aligned} \tag{4.9}$$

which is equal to the wrench we obtained with the feedforward control law, that is (4.6). Interestingly, the wrench does not depend on k_p and the intro-

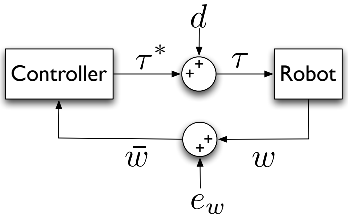


Figure 4.2: Control scheme with a disturbance d on the input joint torques, and a disturbance e_w on the wrench feedback.

duction of the feedback term does not help to decrease the error. This is due to the fact that contact point estimation errors cannot be modeled as an additive input disturbance (as depicted in Fig 4.1), because they also affect the wrench feedback w. To model these errors we must also include an additive disturbance on the wrench feedback, as depicted in Fig. 4.2. In general, considering a disturbance d on the control torques, a disturbance e_w on the wrench feedback and the proportional feedback control law (4.8) we get:

$$w = w_d - \frac{k_p}{1+k_p} e_w - \frac{1}{1+k_p} J_c^{T+} d$$

Increasing k_p we can reduce the error induced by d, but at the same time we also increase the error induced by e_w, so in general there is no way to get $w = w_d$. The joint accelerations induced by e_p can be computed as before:

$$\ddot{q} = (1+k_p) M^{-1} N_c^T J_a^T e_p \times f_d,$$

which are the same as in the feedforward control law, but multiplied by $(1 + k_p)$.

The situation is different in case only force is controlled, because the force feedback is not affected by e_p:

$$\begin{aligned}
\bar{\tau} &= \bar{J}_l^T (f_d - k_p(f - f_d)) + \tilde{h} - k_d \dot{q} \\
&= (J_l^T + J_a^T [e_p \times])(f_d - k_p(f - f_d)) + \tilde{h} - k_d \dot{q} \\
&= (1+k_p) \bar{J}_l^T f_d - k_p \bar{J}_l^T f + \tilde{h} - k_d \dot{q},
\end{aligned}$$

where $J_l \in \mathbb{R}^{3\times n}$ is the linear part of the Jacobian. Substituting this into (4.2):

$$\begin{aligned}
w &= J_c^{T+}((1+k_p)\bar{J}_l^T f_d - k_p \bar{J}_l^T f) \\
w + k_p J_c^{T+} \bar{J}_l^T f &= (1+k_p) J_c^{T+} \bar{J}_l^T f_d \\
\frac{1}{1+k_p} w + \frac{k_p}{1+k_p} J_c^{T+} \bar{J}_l^T f &= J_c^{T+} \bar{J}_l^T f_d
\end{aligned}$$

As k_p goes to infinity, this equation goes to:

$$J_c^{T+} \bar{J}_l^T f = J_c^{T+} \bar{J}_l^T f_d$$

If $J_c^{T+} \bar{J}_l^T$ is not singular, this implies $f = f_d$. Differently from the previous case, the introduction of a feedback term helps to reduce the error. This is due to the fact that the measured force is not corrupted by the error in the estimation of the contact point, while the moment is.

4.2.3 *Parallel Force/Position Control*

In real world applications most of the assumptions that we made in the previous derivation do not hold: i) contacts are not perfectly rigid, ii) the dynamic parameters of the robot are not perfectly estimated, iii) the contact geometry may not allow the manipulator to apply force in the desired direction. Moreover, robots rarely have to control contact forces only, but they are typically concerned with both position and force at the same time. To investigate the role of precise contact point estimation in a real complex scenario we take a practical approach. In this section we present a parallel control law that will be used on a real robot to see how performances degrade as we introduce errors in the contact point estimation.

The target task is to make the robot slide against an irregular unknown object applying a controlled force and moving the end-effector along a desired path. Given the uncertainties in the environment geometry, the specified path is almost surely incompatible with the desired intensity of the contact force, so the control law has to balance position errors and force errors. We start with one of the parallel control laws presented in [Chiaverini and Sciavicco 1993], that is the superposition of a PD position controller and a PI force controller:

$$\begin{aligned} \tau &= J^T(F - f_d) - k_d\dot{q} + h \\ F &= k_p(x_d - x) - k_f(f_d - f) - k_i \int_0^T (f_d - f)dt \end{aligned}$$

where we simply replaced the cartesian velocity term with a joint velocity term, which ensures joint space stability in case the robot is redundant w.r.t. the control task. The integral term in the force controller ensures dominance of the force loop over the position loop. We need to modify this control law because, differently from [Chiaverini and Sciavicco 1993], the interested position and force are relative to two distinct points: the end-effector x and the contact point c. The modified control law is:

$$\begin{aligned} \tau &= J_c^T(F_c - f_d) + J^T F_p - k_d\dot{q} + h \\ F_p &= k_p(x_d - x) \\ F_c &= -k_f(f_d - f) - k_i \int_0^T (f_d - f)dt \end{aligned} \tag{4.10}$$

where J and J_c are respectively the Jacobian of the end-effector and the Jacobian of the contact point. Since we are not concerned with the force direction, but just with its norm, we decided to use the direction of the measured force as desired direction:

$$f_d = f_N \frac{f}{||f||}, \quad \text{if} \quad ||f|| > \frac{f_N}{2}$$

where f_N is the desired force norm. In the approaching phase, when the norm of the measured force is less than half of the desired norm force, a predefined force direction is used. We assume that the robot has a rough estimate of the position of the object and so it can define an initial force direction. However, given the irregularities in the environment surface, the direction of the initial desired force may be inadequate (e.g. it may not be possible to apply a vertical force on a 45°inclined plane). Adapting the direction of f_d helps to compensate for the lack of knowledge about the geometry of the environment and it makes the controller more robust to errors in the initial direction of f_d.

4.3 EXPERIMENTAL RESULTS

As already mentioned in section 4.2.3, most of the simplifying assumptions that we made in our derivation do not hold on a real robot. Furthermore, we have to take into account that: i) measurements of contact forces are usually noisy and they may be affected by the errors in the dynamic parameters of the robot, ii) the quality of the tracking of the torque controllers may be poor. All these factors make the validation on the robot of the presented analysis extremely difficult. Hence we decided to use a simulated robot to validate the analytical analysis and to carry out additional tests on the real robot as an empirical analysis of more complex scenarios.

4.3.1 *Simulation Tests*

We use a simulated puma560 robot arm with 6 dofs, which is provided with the matlab robotics toolbox [Corke 2011]. Both the motors and the force/-torque sensor are simulated as low-pass filters with cut frequency of 40 Hz and 20 Hz, respectively. In the simulation the manipulator is in rigid contact with the environment at the end-effector. An error in the estimation of the contact location $e_p = [0.1; 0; 0]m$ is introduced and we command a desired force $f_d = [0; 10; 0]N$ and a desired moment $\mu_d = [0; 0; 1]Nm$. The four tests span two different constraint situations and two control laws. In the tests 4.3.1.1 and 4.3.1.3 we assume that the end-effector of the robot is completely constrained (i.e. $rank(J_c) = 6$), so it can apply force and moment in all directions, but it cannot move. In the tests 4.3.1.2 and 4.3.1.4 we assume that the end-effector can only exert forces on the environment (i.e. $rank(J_c) = 3$), so it cannot apply moment, but it can move. The first two tests use the feedforward control law (4.3), whereas the last two tests use the feedback control law (4.8).

4.3.1.1 *Feedforward Wrench Control*

As predicted by (4.6) and (4.7), the error in the commanded torques is totally projected in the contact wrench, in particular it affects the contact moment only, resulting in $f = [0; 10; 0]$ and $\mu = [0; 0; 2]$. The desired contact force is correctly applied and the joint accelerations are zero, because the contact Jacobian J_c has zero nullspace.

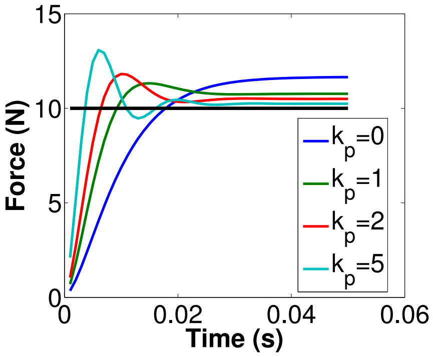


Figure 4.3: Contact force with proportional feedback force control for different values of the proportional gain k_p.

4.3.1.2 *Feedforward Force Control*

In this case the contact Jacobian J_c has a non-trivial nullspace, so a part of the error in the commanded torques is selected by J_c^{T+} and it affects the contact force, resulting in $f = [0; 11.66; 0]N$. The remaining part, which is selected by the null space projection matrix of J_c, generates joint accelerations, resulting in $||\ddot{q}|| = 3.67m/s^2$.

4.3.1.3 *Feedback Wrench Control*

As predicted by (4.9) the result is the same as in the feedforward case: the error affects the contact moment only and it is independent of the proportional gain k_p. Joint accelerations are zero.

4.3.1.4 *Feedback Force Control*

Similarly to test 4.3.1.2, the contact localization error affects both the contact force and the joint accelerations. Since the force feedback is not affected by e_p, it helps to reduce both errors. Fig. 4.3 shows how the steady-state error of the contact force decreases as k_p increases. Of course as k_p goes up the overshoot of the system rises. The joint accelerations decrease as k_p increases.

4.3.2 *Robot Tests*

In these tests we control three joints of the robot's arm: the elbow joint and two out of the three shoulder joints.

4.3.2.1 *Feedforward Force Control*

The first test uses the control law (4.3). The right forearm of the robot lies on a rope, which is fixed to a rigid structure located above the robot. In this configuration the robot is able to apply downward forces on the rope. A downward force of 5 N is commanded. When the tactile sensors are not used, the contact point is assumed to be at the end-effector, which is about 9 cm off the

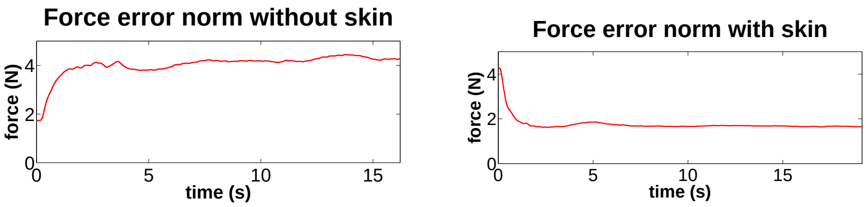


Figure 4.4: Norm of the contact force error using the feedforward force controller. The real contact point, on the forearm, is about 9 cm off the end-effector.

real contact point. Fig. 4.4 shows that, when not using the tactile sensors, the norm of the force error increases from ~ 1.7 N to ~ 4 N.

4.3.2.2 *Parallel Control*

This test uses the parallel control law (4.10) with $k_p = 150$, $k_d = 0.02$, $k_f = 0.5$ and $k_i = 0.02$. The robot starts with the lower part of its forearm in contact with an irregular rigid object, i.e. a tripod for cameras (see Fig. 4.7). The desired trajectory of the end-effector spans only the x direction, moving 5 cm back and 5 cm forth in a straight line. One whole cycle lasts about 33 seconds (the movement is quite slow). While moving the end-effector, the robot tries to maintain a contact force of 3 N, regardless of the force direction. Fig. 4.5 and 4.6 show the position and force errors that we obtained in the two cases, i.e. when the contact force is assumed to be at the end-effector, and when using the tactile sensors to locate the contact point. The performance benefits from the precise estimation of the contact point: when using the tactile sensors the mean norm of the force error decreases from

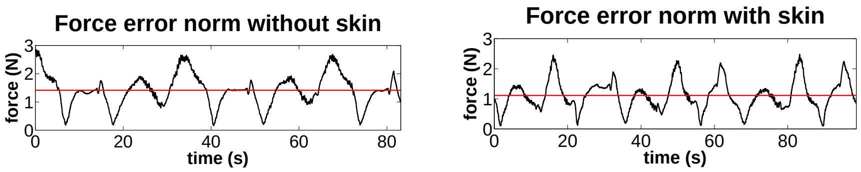


Figure 4.5: Parallel control test. The red line shows the mean value of the error, that is about 1.4 N when the contact is assumed to be at the end-effector, whereas it is about 1.1 N when using the tactile feedback.

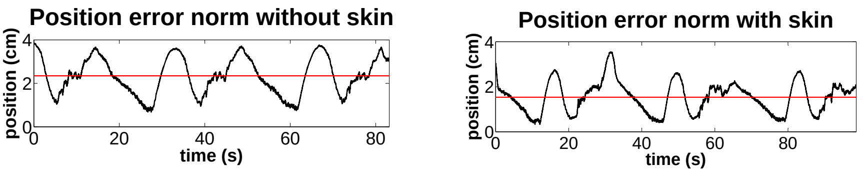


Figure 4.6: Parallel control test. The red line shows the mean value of the error, that is about 2.3 cm when the contact is assumed to be at the end-effector, and about 1.5 cm when using the tactile feedback.

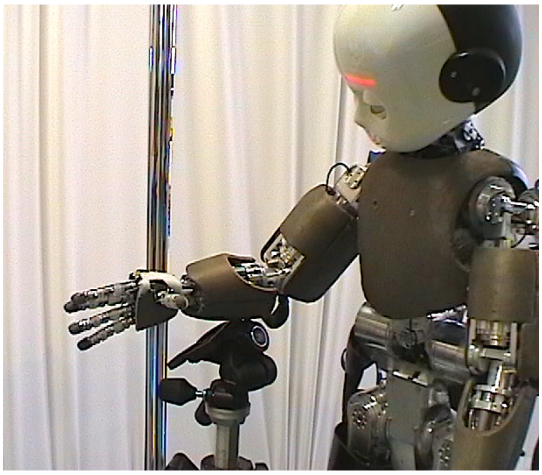


Figure 4.7: The iCub robot, with skin mounted on the torso, arms, palms and fingertips. The iCub is sliding its forearm against the top of the tripod that is standing in front of it.

$\sim$ 1.4 N to $\sim$ 1.1 N, and the mean norm of the position error decreases from $\sim$ 2.3 cm to $\sim$ 1.5 cm. The force error does not converge to a stable value despite the integral term in the control law. This is due to the fact that the integral is reset every time the reference position is modified (i.e. about every 16 seconds). As expected, the improvements due to a precise estimate of the contact point are not as large as in the previous test, because, as proved is section 4.2.2, the feedback helps to reduce the force error.

4.4 CONCLUSIONS AND FUTURE WORK

In this chapter we analyzed how knowledge of the location of the contact points affects force control tasks. We showed how errors in the estimation of contact points may affect the contact forces and, possibly, the joint accelerations. We showed that the introduction of a feedback term may not necessarily help to reduce the errors, if the feedback itself is affected by the estimation error.

Tests on a simulated 6-DOF robotic arm validated our hypotheses. Moreover, we presented the results of two experiments on the iCub robot. In these tests we could not use the equations that we found in our analytical analysis because most of the assumptions that we made in the derivation do not hold. However, the presented results are important because they empirically prove the effectiveness of precise contact point estimation, even in complex real-world scenarios. In the first test, using a feedforward force control law we measure a large drop in the force error (from 4 N to 1.7 N) when tactile feedback is used to locate the contact point. Then, in the second test, we use a parallel control law to make the iCub maintain contact with a rigid irregular object, while moving the end-effector along a straight line. Tests with and without the tactile feedback reveal a significant improvement in the performance when using the tactile system. The average force error drops from 1.4

N to 1.1 N, whereas the average position error drops from 2.3 cm to 1.5 cm. In this test the improvements are not as large as in the previous test because, as proved in section 4.2.2, the introduction of a force feedback helps to reduce the errors. Differently from most works in the literature, we do not use any geometric model of the robot or the environment. The lack of knowledge about the external world is compensated by tactile and force feedback.

The presented analysis could be extended to more complex control laws such as impedance control, hybrid control and parallel control. Also, we could investigate the case where contacts are not perfectly rigid.

5

"TASK SPACE INVERSE DYNAMICS": A NEW FRAMEWORK FOR PRIORITIZED POSITION-FORCE CONTROL

This chapter presents a new framework for multi-task position/force control of fully-actuated rigid robots. Other similar control frameworks have been presented in the last decade. Some of them are not optimal, that is, they do not find the optimal solution for the non-primary tasks. Other frameworks are optimal, but they tackle the control problem at kinematic level, hence they do not allow for force control. Still other frameworks are optimal and consider force control, but they are less efficient than ours. Our control framework, called "Task Space Inverse Dynamics", computes the optimal solution in an efficient way by decoupling kinematics and dynamics of the robot. We take into account: motion and force control, soft and rigid contacts, free and constrained robots. Tests in simulation validate our control framework, comparing it with other two state-of-the-art equivalent frameworks and showing remarkable improvements in optimality and efficiency. The presented analysis is completely theoretical, but its implications are practical because they concern the optimality and the computational cost of the control laws.

5.1 INTRODUCTION

A large number of frameworks for the multi-task control of rigid robots have been presented in the robotics literature. Most frameworks presented in the '80s and '90s [Baerlocher and Boulic 1998; Chiaverini 1997; Nakamura et al. 1987; Siciliano and Slotine 1991] work at a kinematic level, computing the desired joint velocities or accelerations. This approach is not suited for controlling robots that interact with the environment, because it does not allow for force control. This reason motivated a more recent trend of torque control strategies [De Lasa and Hertzmann 2009; Mistry and Righetti 2011; Saab et al. 2011a; Sentis and Khatib 2005], which work on the robot's dynamics, computing the desired joint torques.

Peters et al. [2007] showed that several of these well-known torque control laws can be derived under a Unifying Framework (UF), as solutions of a constrained minimization problem. However, it is still unclear how these frameworks differ from each other and what the pros and cons of each framework are.

This paper has a twofold aim: first, to provide a fair comparison of the state-of-the-art torque control frameworks; second, to present a new framework which outperforms the current state of the art. Our evaluation is based on four criteria: soundness, optimality, capabilities and efficiency. We carry out an analytical analysis of the frameworks and we test them in simulation to confirm the theoretical results.

Section 5.1.1 summarizes the related works and defines the basic tracking control problem. Section 5.2 and 5.3 describe the Unifying Framework (UF) Peters et al. [2007] and the Whole-Body Control Framework (WBCF) Sentis and Khatib [2005], which are the frameworks that we chose as representative of the state of the art. Section 5.5 presents our control framework, Task Space Inverse Dynamics (TSID). For each framework we start presenting the basic solution of a single task, then we extend it to the multi-task case, and finally we introduce the hybrid position/force control. Section 5.6 tests the three frameworks (TSID, UF, WBCF) in simulation on the same multi-task scenario, comparing their performances in terms of optimality and efficiency. The test outcomes prove that our control framework is sound, optimal and computationally more efficient than any other framework with equal capabilities.

5.1.1 *Related Works*

Table 5.1 lists the control frameworks that we considered in our analysis (included the one presented here: Task Space Inverse Dynamics), summarizing their main features. We value a prioritized control framework in terms of soundness, optimality, capabilities and efficiency. Table 5.1 also specifies the motor commands computed by each framework (column "Out."), which can be joint torques τ, joint velocities $\dot{q}$ or joint accelerations $\ddot{q}$. A framework is

Table 5.1: Control frameworks.

Framework	Optimal	Efficient	Force Ctrl	Under.	Out.
TASK SPACE INVERSE DYNAMICS (TSID)	×	×	×		τ
Peters et al. [2007] (UF)		×	×		τ
Sentis and Khatib [2005] (WBCF)	×		×	(×)	τ
Mistry and Righetti [2011]			×	×	τ
Saab et al. [2011a]	×		×	×	τ
De Lasa and Hertzmann [2009]	×		×	×	τ
Jeong [2009]	×	×			$\tau/\ddot{q}$
Chiaverini [1997]		×			$\dot{q}$
Siciliano and Slotine [1991]	×	×			$\dot{q}$
Baerlocher and Boulic [1998]	×	×			$\dot{q}$
Nakamura et al. [1987]		×			$\dot{q}/\ddot{q}$

sound if the control action of any task does not affect the performance of any higher priority tasks. A framework is *optimal* if its control action minimizes the error of each task, under the constraint of being *sound*. The *capabilities* of a framework concern the types of tasks and systems that it allows to control. Finally, a framework is *efficient* if its computational complexity is lower than or equal to the computational complexity of any other equivalent - in terms of soundness, optimality and capabilities - framework.

All the control frameworks that we analyzed are *sound*. In terms of *capabilities*, table 5.1 reports whether a framework allows for force control and whether it can control underactuated systems. Since we are interested in controlling robots that interact with the environment we focus on frameworks that allow for force control. An interesting capability - not reported in table 5.1 - of the framework by Saab et al. [2011a] is the inequality task, which can be used for joint limit avoidance, balance, visibility, collision avoidance. However, this feature comes at a price: the algorithm can no longer compute the solution using pseudoinverses, but it requires a QP solver. Even if Escande et al. [2010] proposed an optimized algorithm for this kind of problems, the computation time is still bigger than 1 ms (too much for implementing 1 kHz control loops Albu-Schäffer et al. [2007]). For this reason our control framework does not include inequality constraints.

This work is motivated by the fact that no control framework that allows for force control is both optimal and efficient. Between the five frameworks that allow for force control, we select two as representative of the state of the art and we describe them in the next sections. Our first choice is the Unifying Framework (UF) Peters et al. [2007], because it is the only one that allows for force control while being *efficient*. Our second choice is the Whole-Body Control Framework (WBCF) Sentis and Khatib [2005], because it represents the category of "optimal but not efficient" frameworks (i.e. the frameworks Saab et al. [2011a] and De Lasa and Hertzmann [2009]). Even though the WBCF was extended to floating-base systems, here we consider the formulation for fully-actuated robots presented in Sentis and Khatib [2005] and implemented in Philippsen et al. [2011].

5.1.2 *Problem definition*

We want to design position tracking control laws for a rigid manipulator with n degrees of freedom. The equation of motion of a manipulator in free space may be written as [Siciliano and Khatib 2008]:

$$M(q)\ddot{q} + h(q, \dot{q}) = \tau, \tag{5.1}$$

where $q \in \mathbb{R}^n$ is the vector of joint coordinates, $\tau \in \mathbb{R}^n$ is the vector of joint torques, $M(q) \in \mathbb{R}^{n \times n}$ is the joint space inertia matrix, and $h(q, \dot{q}) \in \mathbb{R}^n$ contains all the nonlinear terms such as Coriolis, centrifugal and gravity torques. A position tracking task for the robot is described in the form of a time-varying constraint $f(q) = x_r(t)$, where $x_r(t) \in \mathbb{R}^m$ is the reference task trajectory. Differentiating this constraint twice with respect to time we get:

$$\begin{aligned} J(q)\dot{q} &= \dot{x}_r(t) \\ J(q)\ddot{q} + \dot{J}(q)\dot{q} &= \ddot{x}_r(t) \end{aligned} \tag{5.2}$$

where $J(q) = \frac{\partial}{\partial q} f(q) \in \mathbb{R}^{m \times n}$ is the task Jacobian. In the following, dependency upon t, q and $\dot{q}$ is no longer denoted to simplify notation. Since we are going to apply (5.2), which is the second derivative of the constraint, a drift is likely to occur. To prevent deviations from the desired trajectory and to ensure disturbance rejection we design a proportional-derivative feedback control law:

$$\ddot{x}^* = \ddot{x}_r + K_d(\dot{x}_r - \dot{x}) + K_p(x_r - x),$$

where $K_d, K_p > 0$, $K_d, K_p \in \mathbb{R}^{m \times m}$ are diagonal positive definite matrices acting as derivative and proportional gains, respectively.

5.2 UNIFYING FRAMEWORK (UF)

The Unifying Framework (UF) formulates the control problem as a constrained minimization:

$$\begin{aligned} & \tau^* = \underset{\tau \in \mathbb{R}^n}{\text{argmin}}\, ||\ddot{x} - \ddot{x}^*||^2 \\ s.t. \quad & M\ddot{q} + h = \tau \\ & J\ddot{q} + \dot{J}\dot{q} = \ddot{x} \end{aligned} \tag{5.3}$$

We multiply the first constraint times JM^{-1} and then we substitute $J\ddot{q}$ from the second constraint:

$$\ddot{x} - \dot{J}\dot{q} + JM^{-1}h = JM^{-1}\tau$$

Solving this equation for $\ddot{x}$ and substituting it in the cost function we get an unconstrained minimization problem that is equivalent to (5.3):

$$\tau^* = \underset{\tau \in \mathbb{R}^n}{\text{argmin}}\, ||JM^{-1}(\tau - h) + \dot{J}\dot{q} - \ddot{x}^*||^2 \tag{5.4}$$

In case $m < n$ this problem has infinite solutions:

$$\begin{aligned} \tau^* =& (JM^{-1})^+_V(\ddot{x}^* - \dot{J}\dot{q} + JM^{-1}h) + \\ & (I - (JM^{-1})^+_V JM^{-1})\tau_0 \end{aligned} \tag{5.5}$$

where $\tau_0 \in \mathbb{R}^n$ is an arbitrary vector, $V \in \mathbb{R}^{n \times n}, V > 0$ is an arbitrary matrix and $A^+_V = VA^T(AVA^T)^{-1}$ is the pseudoinverse of the matrix A weighted by the matrix V (see appendix A for a review on pseudoinverses). Choosing a particular pair (V, τ_0) we get the solution that minimizes $||V^{-\frac{1}{2}}(\tau - \tau_0)||^2$ [Bjorck 1996]. Setting $\tau_0 = 0$ and varying V we get different well-known control laws, reported in table 5.2. The second row in table 5.2 reports the

Table 5.2: Control laws for different values of weight matrix V

V	MINIMIZE	CONTROL LAW, τ^*
I	$\|\|\tau\|\|^2$	$M^{-1}J^T(JM^{-2}J^T)^{-1}(\ddot{x}^* - \dot{J}\dot{q} + JM^{-1}h)$
M	$\tau^T M^{-1}\tau$	$J^T(JM^{-1}J^T)^{-1}(\ddot{x}^* - \dot{J}\dot{q} + JM^{-1}h)$
M^2	$\|\|M^{-1}\tau\|\|^2$	$MJ^T(JJ^T)^{-1}(\ddot{x}^* - \dot{J}\dot{q} + JM^{-1}h)$

well-known Operational Space control law of Khatib [Khatib 1987]. This solution selects the torques that would be generated by a force applied at the control point.

Without loss of generality, given that $M > 0$, we can set $V = M^2W$, where $W > 0$ is another arbitrary matrix, so that (5.5) simplifies to:

$$\tau^* = MJ_W^+(\ddot{x}^* - \dot{J}\dot{q} + JM^{-1}h) + MN_W M^{-1}\tau_0 \tag{5.6}$$

where $N_W = I - J_W^+ J$ is a weighted nullspace projection matrix associated with J.

5.2.1 *Hierarchical Extension*

The Unifying Framework can manage an arbitrary number of tasks N, each characterized by a desired acceleration $\ddot{x}_i^*$ and a Jacobian J_i. To ensure the correct management of task conflicts, the tasks need prioritization; the higher the number of the task, the higher its priority.

$$\begin{aligned} \tau^* &= \sum_{i=1}^{N} \tau_i \\ \tau_i &= MJ_{iW}^+(\ddot{x}_i^* - \dot{J}_i\dot{q} + J_i M^{-1}(h - \sum_{j=0}^{i-1}\tau_j)) \end{aligned} \tag{5.7}$$

where $i = N \ldots 1$. The torque vector of each task τ_i is projected into the nullspace of the higher priority tasks; this guarantees that the framework is *sound*. However, this approach is not *optimal*, because each task is solved independently, and then projected onto the nullspace of the higher priority tasks. This does not ensure the minimization of the error of each task (see Baerlocher and Boulic [1998]; Chiaverini [1997] for a thorough explanation).

5.2.2 *Hybrid Control*

The Unifying Framework allows for hybrid position/force control by setting the joint space control torques to:

$$\tau_0 = h - J_c^T f^*,$$

where $J_c(q) \in \mathbb{R}^{k \times n}$ is the contact Jacobian, $f^* \in \mathbb{R}^k$ are the desired contact forces and $k \in \mathbb{R}$ is the number of independent directions in which the robot applies force. Substituting τ_0 into the desired control torques (5.6) we get:

$$\tau^* = M J_W^+ (\ddot{x}^* - \dot{J}\dot{q}) + h - M N_W M^{-1} J_c^T f^*,$$

where the applied forces act in the nullspace of the tracking task.

5.3 WHOLE-BODY CONTROL FRAMEWORK (WBCF)

In this section we describe the WBCF presented by Sentis and Khatib [2005]. This framework is based on the Operational Space Formulation [Khatib 1987], which is equivalent to the UF for a particular choice of the weight matrix W. Setting $W = M^{-1}$ in the control law (5.6) we get the Operational Space control law:

$$\tau^* = J^T \underbrace{(J M^{-1} J^T)^{-1}}_{\Lambda} (\ddot{x}^* - \dot{J}\dot{q} + J M^{-1} h) + (I - J^T \bar{J}^T)\tau_0,$$

where $\bar{J} = M^{-1} J^T \Lambda$ is the dynamically consistent Jacobian pseudoinverse and Λ is the task space inertia matrix.

5.3.1 *Hierarchical Extension*

While in case of a single task the WBCF and the UF are equivalent, their hierarchical extensions differ substantially:

$$\begin{aligned} \tau^* &= \sum_{i=1}^{N} J_{p(i)}^T F_{p(i)} \\ F_{p(i)} &= \Lambda_{p(i)} (\ddot{x}_i^* - \dot{J}_i \dot{q} + J_i M^{-1} (h - \sum_{j=1}^{i-1} J_{p(j)}^T F_{p(j)})) \\ J_{p(i)} &= J_i (I - \sum_{j=1}^{i-1} \bar{J}_{p(j)} J_{p(j)}) \end{aligned} \tag{5.8}$$

The difference with the formulation (5.7) is not limited to the choice of the weight matrix W; the prioritization strategy is different as well. The WBCF minimizes the error of each task under the constraint of not conflicting with any higher priority tasks, namely it is *optimal*.

5.3.2 *Hybrid Control*

The WBCF allows for hybrid position/force control by setting:

$$F_{p(i)} = \Omega_f f_i^* + \Lambda_{p(i)} (\Omega_m \ddot{x}_i^* - \dot{J}_i \dot{q} + J_i M^{-1} (h - \sum_{j=1}^{i-1} J_{p(j)}^T F_{p(j)})),$$

where the selection matrices Ω_f and Ω_m split the control space into force and motion components, respectively.

5.4 SOME OBSERVATIONS

Before deriving the new control framework “Task Space Inverse Dynamics”, we point out some facts regarding hierarchical control frameworks. This section is not essential to understand the derivation presented in section 5.5, but it should help the reader to understand the motivations and the choices underlying this work.

5.4.1 *Observation 1: Derivation of dynamically consistent pseudoinverse*

The second row in table 5.2 reports the famous operational space control law of Khatib [Khatib 1987], which is usually written as:

$$\tau^* = J^T F = J^T(\Lambda \ddot{x}^* + \mu + p) \tag{5.9}$$

where $\Lambda = (JM^{-1}J^T)^{-1}$ is the operational space inertia matrix, $\mu = \bar{J}^T b - \Lambda \dot{J}\dot{q}$, $p = \bar{J}^T g$ and $\bar{J}$ is the renowned *dynamically consistent* Jacobian pseudoinverse:

$$\bar{J} = M^{-1}J^T(JM^{-1}J^T)^{-1} = M^{-1}J^T\Lambda \tag{5.10}$$

At first this solution may appear not to have any particular property, but looking throughout the literature one may find out that [Bruyninckx and Khatib 2000]: i) it is consistent with the principle of virtual work of D’Alambert, ii) it follows Gauss’ Principle of Least Constraint, iii) it is the natural choice to decouple the internal motion dynamics from the end-effector dynamics. In other words, this solution selects the torques that would be generated by a force applied at the control point. Indeed, we can derive (5.9) by adding the constraint $\tau = J^T f$ to the minimization problem (5.4):

$$\begin{aligned} \tau^* &= J^T f^* \\ f^* &= \underset{f \in \mathbb{R}^m}{\text{argmin}} \, ||JM^{-1}(J^T f - h) + \dot{J}\dot{q} - \ddot{x}^*||^2 \end{aligned}$$

This problem has a unique solution because $JM^{-1}J^T$ has a unique inverse.

5.4.2 *Observation 2: Non-uniqueness of task consistent Jacobian pseudoinverse*

Looking at a big portion of the literature [Khatib 1987, 1995; Mistry and Righetti 2011; Park and Khatib 2008; Sentis et al. 2009], a less than meticulous reader may be led to believe that the dynamically consistent pseudoinverse is the only one that ensures that secondary tasks (i.e. τ_0) do not affect the primary task (i.e. $\ddot{x}$ in case of position control). Statements such as “only one of these generalized inverses is consistent with the system dynamics”[Khatib 1995] actually hide the non-explicit assumption that the author was considering only joint torques that are in the range of J^T (i.e. torques that can be generated by applying a force at the control point).

In [Khatib 1995] Khatib splits the control torques into two parts, the first one resulting from a force applied at the control point, and the second one acting within the nullspace of the control point:

$$\tau = J^T F + (I - J^T J_V^{T+})\tau_0 \tag{5.11}$$

Applying this control torques to the equation of motion of the manipulator, it turns out that the only choice of V that results in τ_0 not affecting $\ddot{x}$ is $V = M$. Comparing the nullspace projectors in (5.5) and (5.11) we can see that limiting the control torques to the form (5.11) is equivalent to looking for a V such that:

$$I - (JM^{-1})_V^+ JM^{-1} = I - J^T J_X^{T+},$$

where X is an arbitrary PD matrix. This problem has only one solution, that is $V = M$ and $X = M^{-1}$.

On the other hand, if we do not limit the control torques to be in the range of J^T, then we can choose among the infinity of pseudoinverses. We can prove it applying the control torques (5.6) to the manipulator dynamics:

$$M\ddot{q} + h = MJ_W^+(\ddot{x}^* - \dot{J}\dot{q} + JM^{-1}h) + MN_W M^{-1}\tau_0$$

Multiplying both sides by JM^{-1} the term containing τ_0 becomes zero, because:

$$JM^{-1}MN_W M^{-1}\tau_0 = JN_W M^{-1}\tau_0 = 0,$$

so we get:

$$\begin{aligned} J\ddot{q} + JM^{-1}h &= JJ_W^+(\ddot{x}^* - \dot{J}\dot{q} + JM^{-1}h) \\ \ddot{x} - \dot{J}\dot{q} + JM^{-1}h &= \ddot{x}^* - \dot{J}\dot{q} + JM^{-1}h \\ \ddot{x} &= \ddot{x}^* \end{aligned}$$

As long as the control torques are computed using (5.6), τ_0 does not affect $\ddot{x}$, regardless of the choice of W.

5.4.3 *Observation 3: Weight Matrix and Joint Space Stabilization*

The weight matrix V (or equivalently W) introduced in the resolution of (5.3) can play two different roles. In case there is no secondary task (i.e. $\tau_0 = 0$), V determines the quantity that we minimize (e.g. $||\tau||^2$, $||\ddot{q}||^2$, $||M^{-\frac{1}{2}}\tau||^2$). In case there is a secondary task, V specifies the metric that is used to measure the distance between τ and τ_0.

The idea of using the nullspace of the task to minimize some measure of effort is very appealing, mainly because it is rooted in the study of human motion [Flash and Hogan 1985]. This approach may be feasible in simulation, but unfortunately in reality it leads to singular configurations and hitting of joint limits [Peters et al. 2007]. The subspace of joint accelerations that do not affect the task is not controlled, so its behavior is determined by disturbances and errors in the model of the manipulator. Even in simulation, if the initial conditions of the robot have non-zero joint velocities, not using a secondary task may result in joint space instability. The reason for this behavior is obvious: the effort of stabilizing in joint space is not task relevant and it would increase the cost [Peters et al. 2007].

Peters et al. [2007] suggest to add a joint space motor command for stabilization. A common approach is to design the postural task to attract the robot towards a desired posture q_0. We compute the desired joint accelerations as $\ddot{q}_0^* = K_p(q_0 - q) - K_d\dot{q}$, where $K_p > 0$ and $K_d > 0$ are matrices acting as proportional and damping gains, respectively. In the following we will always include the postural task as the lowest priority task in the task hierarchy. The postural task minimizes the following quantity:

$$||\ddot{q} - \ddot{q}_0^*||^2,$$

under the constraint of not affecting the higher priority tasks. This ensures stabilization of the manipulator in joint space. We want to find the pair (V, τ_0) in (5.5) such that we use the task nullspace to minimize $||\ddot{q} - \ddot{q}_0||^2$. Recalling that in general we minimize $||V^{-\frac{1}{2}}(\tau - \tau_0)||^2$, if we set $V = I$ then we minimize this quantity:

$$||\tau - \tau_0||^2 = ||M\ddot{q} + h - (M\ddot{q}_0 + h)||^2 = ||M(\ddot{q} - \ddot{q}_0)||^2$$

On the other hand, setting $V = M$ we minimize this quantity:

$$||M^{-\frac{1}{2}}(\tau - \tau_0)||^2 = ||M^{\frac{1}{2}}(\ddot{q} - \ddot{q}_0)||^2$$

Finally, if $V = M^2$ we minimize this quantity:

$$||M^{-1}(\tau - \tau_0)||^2 = ||\ddot{q} - \ddot{q}_0||^2$$

Our solution is then to set $V = M^2$ and $\tau_0 = M\ddot{q}_0 + h$. The resulting control law is:

$$\tau^* = M(J^+(\ddot{x}^* - \dot{J}\dot{q}) + N\ddot{q}_0) + h \tag{5.12}$$

In this scenario the choice $V = M^2$ is the most appropriate, because it leads to a minimization of the joint acceleration error.

5.4.4 *Observation 4: Hierarchical frameworks*

In the literature we can see two different approaches to multi-task management: the *error minimization* approach, call it A, and the *nullspace projection* approach, call it B (see [Baerlocher and Boulic 1998] [Chiaverini 1997] for a thorough comparison). In this subsection we describe both approaches, explaining pros and cons and emphasizing the role of the weight matrix.

5.4.4.1 *Approach A - Error minimization*

This approach minimizes the error of each task, under the constraint of not conflicting with higher priority tasks. Assume that we have two tasks that are described by the constraints:

$$J_2\ddot{q} + \dot{J}_2\dot{q} = \ddot{x}_2^* \qquad J_1\ddot{q} + \dot{J}_1\dot{q} = \ddot{x}_1^*,$$

where task 2 has higher priority with respect to task 1. We formulate a minimization problem:

$$\begin{aligned} &\tau^* = \underset{\tau\in\mathbb{R}^n}{\operatorname{argmin}} \, ||J_1\ddot{q} + \dot{J}_1\dot{q} - \ddot{x}_1^*||^2 \\ s.t. \qquad &M\ddot{q} + h = \tau \\ &\tau \in \left\{ \begin{aligned} \tau_2^* \mid \tau_2^* = &\underset{\tau_2\in\mathbb{R}^n}{\operatorname{argmin}} \, ||J_2\ddot{q} + \dot{J}_2\dot{q} - \ddot{x}_2^*||^2 \\ s.t. \qquad &M\ddot{q} + h = \tau_2 \end{aligned} \right\} \end{aligned}$$

The second constraint is equivalent to problem (5.3) so we replace it with (5.6).

$$\begin{aligned} &\tau^* = \underset{\tau\in\mathbb{R}^n}{\operatorname{argmin}} \, ||J_1\ddot{q} + \dot{J}_1\dot{q} - \ddot{x}_1^*||^2 \\ s.t. \qquad &M\ddot{q} + h = \tau \\ &\tau = M\ddot{q}_2^* + MN_{2W}M^{-1}\tau_0 \\ &\ddot{q}_2^* = J_{2W}^+(\ddot{x}_2^* - \dot{J}_2\dot{q} + J_2M^{-1}h) \end{aligned}$$

The last two constraints specify the form of the solution, leaving the choice of τ_0 for minimizing the cost function.

$$\begin{aligned} \tau^* =& M\ddot{q}_2^* + MN_{2W}M^{-1}\tau_0^* \\ \tau_0^* =& \underset{\tau_0\in\mathbb{R}^n}{\operatorname{argmin}} \, ||J_1\ddot{q} + \dot{J}_1\dot{q} - \ddot{x}_1^*||^2 \\ &s.t. \qquad M\ddot{q} + h = M\ddot{q}_2^* + MN_{2W}M^{-1}\tau_0 \end{aligned}$$

Computing $\ddot{q}$ from the constraint and substituting it in the cost function we get an unconstrained minimization problem:

$$\tau_0^* = \underset{\tau_0\in\mathbb{R}^n}{\operatorname{argmin}} \, ||J_1(\ddot{q}_2^* + N_{2W}M^{-1}\tau_0 - M^{-1}h) + \dot{J}_1\dot{q} - \ddot{x}_1^*||^2$$

The resolution of the minimization is then quite straightforward and it leads to:

$$\begin{aligned} \tau^* =& M(\ddot{q}_2^* + N_{2W}(\ddot{q}_{1|2}^* + N_{1|2W}\ddot{q}_0)) \\ \ddot{q}_2^* =& J_{2W}^+(\ddot{x}_2^* - \dot{J}_w\dot{q} + J_2M^{-1}h) \\ \ddot{q}_{1|2}^* =& J_{1|2W}^+(\ddot{x}_1^* - \dot{J}_1\dot{q} + J_1(M^{-1}h - \ddot{q}_2^*)) \end{aligned} \tag{5.13}$$

where $J_{1|2} = J_1N_{2W}$ and $N_{1|2W}$ is its weighted nullspace projector. This formula:

1. finds the space of solutions of task 2, i.e. it computes $\ddot{q}_2^*$ and N_{2W}

2. inside the solution space of task 2, it finds the point(s) that minimize the error on task 1

3. among these points (if more than one), it selects the one that minimizes $||W^{-\frac{1}{2}}(\ddot{q}-\ddot{q}_0)||^2$

This approach ensures zero error on any task, as long as it does not conflict with any higher priority task. Unfortunately this approach also has a drawback: it suffers from *algorithmic singularities*, namely singularities that are due to conflicts between tasks. In the subsection 5.4.4.3 we will explain in more detail what this means.

5.4.4.2 *Approach B - Nullspace Projection*

This approach is somewhat simpler because it is the most intuitive extension of the control law (5.6) to the multi-task case. Assuming that we have two tasks as in the previous subsection, we have already seen that to ensure that task 2 is satisfied the control torques have to take the form:

$$\begin{aligned}\tau_2 =& MJ_{2W}^{+}(\ddot{x}_2^* - \dot{J}_2\dot{q} + J_2M^{-1}h) + MN_{2W}M^{-1}\tau_1 = \\ =& M\ddot{q}_2^* + MN_{2W}\ddot{q}_1,\end{aligned} \tag{5.14}$$

where $\tau_1 = M\ddot{q}_1$ is an arbitrary torque vector that can be used to achieve a secondary task, in our case, task 1. Applying the same principles that we used to formulate (5.14) we can compute the torques to achieve task 1 as:

$$\begin{aligned}\tau_1 =& MJ_{1W}^{+}(\ddot{x}_1^* - \dot{J}_1\dot{q} + J_1M^{-1}h) + MN_{1W}M^{-1}\tau_0 = \\ =& M\ddot{q}_1^* + MN_{1W}\ddot{q}_0\end{aligned} \tag{5.15}$$

Putting it all together we get:

$$\tau_2 = M(\ddot{q}_2^* + N_{2W}(\ddot{q}_1^* + N_{1W}\ddot{q}_0)) \tag{5.16}$$

This formula:

1. computes $\ddot{q}_1$, that is the solution of task 1 that minimizes $||W^{-\frac{1}{2}}(\ddot{q}-\ddot{q}_0)||^2$

2. projects $\ddot{q}_1$ onto the space of solutions of task 2 or, in other words, it finds the solution of task 2 that minimizes $||W^{-\frac{1}{2}}(\ddot{q}-\ddot{q}_1)||^2$

The drawback of this approach is that, except for the highest priority task, it does not ensure that a task is achieved, even if it is not in conflict with any task with higher priority.

5.4.4.3 *Comparison: A vs B*

A geometric view can give us a better insight into the two approaches. Let us consider a 2-DOF robot performing two one-dimensional tasks, with task 2 having higher priority. The solution space is the two-dimensional space of joint accelerations, depicted in Fig. 5.1. Each task has infinite solutions

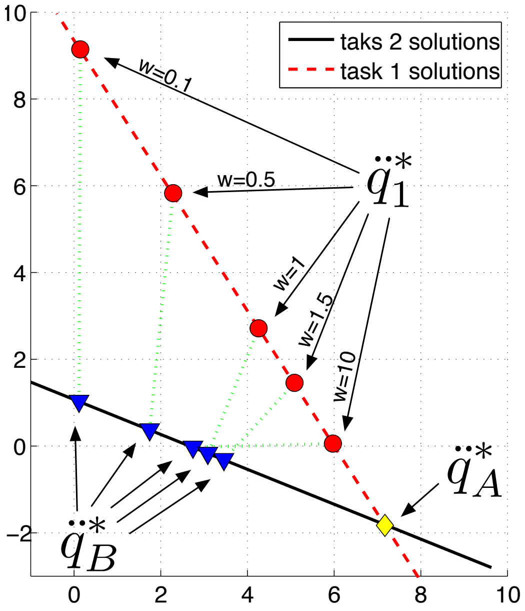


Figure 5.1: Geometric view of multi-task resolution with approach A and B. As far as approach B is concerned, the plot shows five different solutions corresponding to five different values of the weight matrix W (the values used for w are $[0.1; 0.5; 1; 1.5; 10]$). The solution computed by approach A is independent of the weight matrix.

(because $m = 1 < n = 2$), which may be represented as a line on the joint acceleration plane. In particular, in the case depicted in Fig. 5.1, the solution space of task 2 (black solid line) intersects the solution space of task 1 (red dashed line) in one point. The two tasks are not in conflict and that point represents the solution that satisfies both tasks.

Approach B computes $\ddot{q}_1^*$ and projects it onto the solution space of task 2, finding $\ddot{q}_B^*$. Fig. 5.1 depicts $\ddot{q}_1^*$ and $\ddot{q}_B^*$ for different values of the weight matrix W. In particular we set:

$$W = \begin{bmatrix} w & 0 \\ 0 & 1 \end{bmatrix}$$

and we computed the solution for five values of w ranging between 0.1 and 10. Although the two tasks are not in conflict, all the solutions perfectly achieve task 2, but they have a non-zero error on task 1. Note that when $W \neq I$ the projection that is performed by the nullspace projector is not *orthogonal*.

On the other hand, the solution computed by approach A, that is $\ddot{q}_A^*$, is exactly the intersection of the two lines. As expected, $\ddot{q}_A^*$ does not depend on the weight matrix W. In this particular example there is no value of w that results in $\ddot{q}_A^* = \ddot{q}_B^*$. However, this is not always the case.

When two tasks are in conflict their solution spaces are parallel, hence they do not intersect. Both approaches give the same solution $\ddot{q}_A^* = \ddot{q}_B^* = \ddot{q}_2^*$, which results in a non-zero error on task 1. When two tasks are *almost* in conflict their solution spaces are *almost* parallel and approach A suffers from an *algorithmic singularity*. The two solution spaces intersect in a subspace that is distant from the origin, hence $||\ddot{q}^*_A||$ is large. Filtering techniques, such as damped pseudoinverses, are commonly used to tackle this problem [Chiaverini 1997].

Despite the drawbacks of approach B, sometimes it has been preferred over approach A [Chiaverini 1997; Khatib 1995; Mistry and Righetti 2011; Peters et al. 2007] because: i) it is older and hence more popular in the robotics community, ii) it is simpler to understand and to implement, iii) it does not suffer from algorithmic singularities. However, none of these reasons seems a valid argument for choosing a sub-optimal approach over an optimal one. The only case in which approach B should be preferred is when the robot performs just one task and the kinematic redundancy is used for joint stabilization only. We showed in section 5.4.3 that in this case the choice of the weight matrix $V = M^2$ is the most appropriate because it minimizes the joint acceleration error.

On the other hand, if the robot has to perform two or more tasks with different priorities, then approach A has to be taken to get optimal results. Many frameworks for multi-task control that follow the principles of approach A have been presented [Jeong 2009; Saab et al. 2011a; Siciliano and Slotine 1991]. Even the operational space formulation [Khatib 1995], which in its original form used approach B, has been extended later to approach A [Khatib et al. 2004; Sentis and Khatib 2006] (see [Philippsen et al. 2011] for an open-source software implementation of this formulation). When Khatib et al say that the original operational space formulation "did not consider the dynamics of the posture itself", whereas its new extension "compensates for the dynamics within the restricted posture space" [Khatib et al. 2004] they refer to this transition from approach B to A.

5.5 ORIGINAL CONTRIBUTION - TASK SPACE INVERSE DYNAMICS (TSID)

In this section we design the TSID control framework, which is the main contribution of the chapter. The TSID is *sound*, *optimal*, *efficient* — as confirmed by the simulation tests — and allows for both position and force control.

5.5.1 *Motivations*

The WBCF is *sound* and *optimal*, but it is not *efficient* because it requires the computation of the *operational space inertia matrices* Λ's. The simplest way to compute them is using the formula $\Lambda = (JM^{-1}J^T)^{-1}$, which has a complexity of O(n^3): the computation of M - with Recursive Newton-Euler Algorithm (RNEA) or Composite-Rigid-Body algorithm [Siciliano and Khatib 2008] - has a complexity of O(n^2) for serial robots and O(nd) for multi-branch robots (where d is the tree depth). More efficient algorithms [Chang and Khatib 2000] can compute Λ with a complexity of O($nm^2 + m^3$), where m is the dimension of the task.

On the other hand the UF is *sound* and it can be *efficient*: choosing $V = M^2$, the solution takes the form $\tau^* = M\ddot{q}_N + h$, which we can calculate without explicitly computing M, through the O(n) RNEA. Nonetheless, the UF is not *optimal*: even if a task does not conflict with any higher priority tasks, it may not be performed correctly.

The derivation of our framework, TSID, follows the same principles underlying the UF, but with a different hierarchical extension. We minimize the error of each task under the constraint of not affecting any higher priority task. At each minimization step, we carefully select the weight matrices used in the pseudoinverses, so as to simplify the resulting control laws. This leads to an efficient formulation, while preserving the optimality property. We start considering position tracking control only, then we introduce force control tasks.

5.5.2 *Framework Derivation*

Consider a general scenario in which the robot has to perform N position tracking tasks. Consider also a postural task - defining the desired joint accelerations $\ddot{q}_0^*$ - to stabilize any redundancy left. Taking inspiration from the UF and from De Lasa and Hertzmann [2009] we formulate the multi-task control problem as a sequence of constrained minimization:

$$\begin{array}{llllll} T_N & r_N & = \min\limits_{\tau \in \mathbb{R}^n} g_N(\tau) & s.t. & M\ddot{q} + h = \tau & \\ T_i & r_i & = \min\limits_{\tau \in \mathbb{R}^n} g_i(\tau) & s.t. & M\ddot{q} + h = \tau & \\ & & & & g_j(\tau) = r_j & \forall j > i \\ T_0 & \tau^* & = \operatorname*{argmin}\limits_{\tau \in \mathbb{R}^n} ||\ddot{q} - \ddot{q}_0^*|| & s.t. & M\ddot{q} + h = \tau & \\ & & & & g_j(\tau) = r_j & \forall j > 0 \end{array} \quad (5.17)$$

where $g_i(\tau)$ is the quantity to minimize associated to task T_i:

$$g_i(\tau) = ||J_i\ddot{q} + \dot{J}_i\dot{q} - \ddot{x}_i^*||^2$$

The solution of (5.17) is given by:

$$\begin{aligned} \tau^* &= M(\ddot{q}_1 + N_{p(0)}(N_{p(0)})_W^+(\ddot{q}_0^* + M^{-1}h - \ddot{q}_1)) \\ \ddot{q}_i &= \ddot{q}_{i+1} + N_{p(i)}(J_iN_{p(i)})_W^+(\ddot{x}_i^* - \dot{J}_i\dot{q} + J_i(M^{-1}h - \ddot{q}_{i+1})) \\ N_{p(i)} &= N_{p(i+1)} - (J_{i+1}N_{p(i+1)})_W^+ J_{i+1}N_{p(i+1)}, \end{aligned} \quad (5.18)$$

for $i \in [1, N]$. The computation is initialized setting $\ddot{q}_{N+1} = 0$ and $N_{p(N)} = I$. Once again, selecting the weight matrix W we can vary the form of the control law. Interestingly enough though, the solution τ^* is independent of W. This is because the only role of W is to weight the quantity that is minimized in the nullspace of all the tasks, but here the postural task ensures that there is no nullspace left (because any control action would affect at least the postural task). The most reasonable choice is then to set W to the most convenient value from a computational point of view. If we set $W = I$ then all the nullspace projectors $N_{p(i)}$ become orthogonal projectors, so they are idempotent and equal to their pseudoinverses:

$$N_{p(i)} N_{p(i)} = N_{p(i)}, \qquad N_{p(i)}^{+} = N_{p(i)}$$

This simplifies the formulation (5.18) to:

$$\begin{aligned} \tau^* =& M\ddot{q}^* + h \\ \ddot{q}^* =& \ddot{q}_1 + N_{p(0)}\ddot{q}_0^* \\ \ddot{q}_i =& \ddot{q}_{i+1} + (J_i N_{p(i)})^{+}(\ddot{x}_i^* - \dot{J}_i\dot{q} - J_i\ddot{q}_{i+1}) \\ N_{p(i)} =& N_{p(i+1)} - (J_{i+1}N_{p(i+1)})^{+} J_{i+1} N_{p(i+1)}, \end{aligned} \tag{5.19}$$

with $i \in [1, N]$. In this new form, kinematics and dynamics are completely decoupled: first we solve the multi-task prioritization at kinematic level computing $\ddot{q}^*$, then we compute the torques to get the desired joint accelerations. This formulation does not require the computation of a pseudoinverse for the postural task, because it exploits the property of orthogonal projectors of being equal to their pseudoinverses. Moreover, it does not contain the term M^{-1}, and the mass matrix M appears only in the term $M\ddot{q}^*$, which can be efficiently computed with the Recursive Newton-Euler Algorithm.

5.5.3 *Force Control*

This subsection extends the TSID control framework to force control. If the manipulator is in contact with the environment its equation of motion becomes:

$$M(q)\ddot{q} + h(q, \dot{q}) - J_c(q)^T f = \tau, \tag{5.20}$$

where $J_c(q) = \frac{\partial}{\partial q} x_c \in \mathbb{R}^{k \times n}$ is the contact Jacobian (or constraint Jacobian) and $f \in \mathbb{R}^k$ are the contact forces (or constraint forces). To control the contact forces we need a model of the contact dynamics. We consider two cases, which are the most common in the literature: the *linear spring contact* model and the *rigid contact* model.

5.5.3.1 *Soft Force Control - Linear Spring Contact*

In the *linear spring contact* model the environment at the contact point is assumed to behave like a linear spring, so the contact force is proportional to the contact point displacement:

$$\begin{aligned} k_s(x_c - C) &= f \\ k_s\dot{x}_c &= \dot{f} \\ k_s\ddot{x}_c &= \ddot{f} \end{aligned}$$

where k_s is the stiffness of the contact, $x_c \in \mathbb{R}^k$ is the robot contact point and $C \in \mathbb{R}^k$ is the point of the environment where the contact occurred. Similarly to the position control problem, to ensure robustness to disturbances, the reference force acceleration can be computed as:

$$\ddot{f}^* = \ddot{f}_d + K_d(\dot{f}_d - \dot{f}) + K_p(f_d - f),$$

where $f_d(t) \in \mathbb{R}^k$ is the desired force trajectory, while $K_d > 0, K_p > 0$ are matrices acting as derivative and proportional gains, respectively. The problem of controlling the contact forces can then be formulated as:

$$\begin{aligned} & \tau^* = \underset{\tau \in \mathbb{R}^n}{\operatorname{argmin}} ||\ddot{f} - \ddot{f}^*||^2 \\ s.t. \qquad & M\ddot{q} + h - J_c^T f = \tau \\ & k_s(J_c\ddot{q} + \dot{J}_c\dot{q}) = \ddot{f} \end{aligned}$$

Assuming that we know k_s and we can measure the contact forces f, the solutions can be calculated as:

$$\tau^* = MJ_{cW}^+(k_s^{-1}\ddot{f}^* - \dot{J}_c\dot{q} + J_c M^{-1}(h - J_c^T f)) + MN_{cW}M^{-1}\tau_0$$

where $N_{cW} = I - J_{cW}^+ J_c$. Park and Khatib [2008] treated extensively the problem of estimating the contact stiffness k_s.

5.5.3.2 *Rigid Force Control - Rigid Contact*

If the contact between the manipulator and the environment is perfectly rigid then the manipulator motion is subject to a set of k nonlinear constraints:

$$c(q, t) = 0$$

The constraints may be time-varying, hence we can model contacts with curved surfaces, as long as c is sufficiently smooth. The rigid force control problem is defined as:

$$\begin{aligned} & \tau^* = \underset{\tau \in \mathbb{R}^n}{\operatorname{argmin}} ||f - f^*||^2 \\ s.t. \qquad & M\ddot{q} + h - J_c^T f = \tau \\ & J_c\ddot{q} + \dot{J}_c\dot{q} = 0 \end{aligned}$$

where $f^* \in \mathbb{R}^k$ are the desired contact forces. We can prove that the following control law is a solution of the rigid force control problem (see appendix B.1 for the proof):

$$\tau^* = -J_c^T f^* - MJ_c^+(\dot{J}_c\dot{q} - J_c M^{-1}h) + MN_c M^{-1}\tau_0$$

The term M^{-1} still appears in the equation but, as we have already seen for position control, it can be removed by the postural task, setting $\tau_0 = M\ddot{q}_0^* + h$:

$$\tau^* = -J_c^T f^* + M(-J_c^+\dot{J}_c\dot{q} + N_c\ddot{q}_0^*) + h$$

5.5.4 *Integration of Force Control in Hierarchical Framework*

We extend the multi-task formulation (5.19) to include force control tasks. The rigid force control task, if any, has to take the highest priority because it is a physical constraint that cannot be violated by definition. We assume that the robot has to perform $N-1$ tasks, which can be either position or soft force control tasks. Soft force control tasks are treated as position control tasks, in which we set the reference acceleration based on the contact stiffness and the reference force acceleration: $\ddot{x}_i^* = k_i^{-1}\ddot{f}_i^*$. On top of that there is a rigid force control task, with reference force f^* and Jacobian $J_N = J_c$:

$$\begin{aligned}
\tau^* =& M\ddot{q}^* + h - J_c^T f^* \\
\ddot{q}^* =& \ddot{q}_1 + N_{p(0)}\ddot{q}_0^* \\
\ddot{q}_i =& \ddot{q}_{i+1} + (J_i N_{p(i)})^+(\ddot{x}_i^* - \dot{J}_i\dot{q} - J_i\ddot{q}_{i+1}) \\
N_{p(i)} =& N_{p(i+1)} - (J_{i+1}N_{p(i+1)})^+ J_{i+1}N_{p(i+1)},
\end{aligned} \tag{5.21}$$

with $i \in [1, N]$, $\ddot{x}_N^* = 0$, $\ddot{q}_{N+1} = 0$, and $N_{p(N)} = I$. Even after the extension to force control, kinematics and dynamics are still decoupled, so τ^* can be efficiently computed with the Recursive Newton-Euler Algorithm.

5.6 TESTS

5.6.1 *Simulation Environment*

We tested the presented control framework on a customized version of the Compliant huManoid (CoMan) simulator [Dallali et al. 2013]. The robot has 23 DoFs: 4 in each arm, 3 in the torso and 6 in each leg. We adapted the simulator to make the robot rigid and fully-actuated (we fixed the robot base and we removed the joint passive compliance). Direct dynamics and inverse dynamics, both in simulation and control, were efficiently computed using C language functions, generated with the Robotran webpage [2012] symbolic engine. Contact forces were simulated using linear spring-damper models with realistic friction. To simulate rigid contacts we used the stiffness and damping values proposed by Dallali et al. [2013], that are, respectively, $2\cdot 10^5 N/m$ and $10^3 Ns/m$. To integrate the equations of motion we used the Simulink variable step integrator *ode23t*, with relative and absolute tolerance of 10^{-3} and 10^{-6}, respectively. The tests were executed on a computed with a 2.83 GHz CPU and a 4 GB RAM.

5.6.2 *Test Details*

To generate reference position/velocity/acceleration trajectories we used the approach proposed by Pattacini et al. [2010], which provides approximately minimum jerk trajectories. The trajectory generator is a 3rd order dynamical system that takes as input the desired trajectory $x_d(t)$ and outputs the three position/velocity/acceleration reference trajectories $x_r(t), \dot{x}_r(t), \ddot{x}_r(t)$.

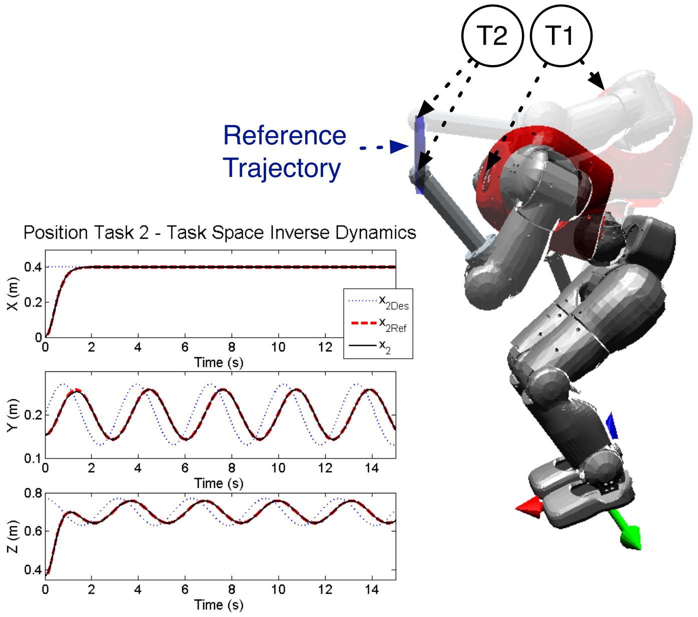


Figure 5.2: On the right, two superimposed screenshots of CoMan executing the higher priority task T2 with its left hand and the lower priority task T1 with its neck base. The blue line shows the reference trajectory for the task T2. On the left, a plot of the desired, reference and real trajectory of the left hand, using the TSID; since the tracking is almost perfect, the real trajectory (black line) and the reference trajectory (red dashed line) overlap.

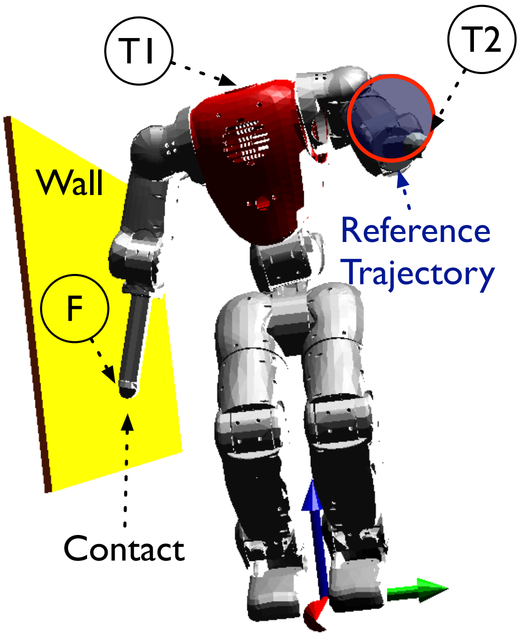


Figure 5.3: CoMan executing three tasks. The force task F controls the force exerted by the right hand against the wall. The tracking task T2 moves the left hand along the circular reference trajectory depicted as a red circumference. The tracking task T1 moves the neck base back and forth along the x axis.

The reference position trajectory follows the desired position trajectory with a velocity that depends on the parameter "trajectory time" (always set to $1.0s$ in our tests). The desired accelerations were computed as:

$$\ddot{x}^* = \ddot{x}_r + K_d(\dot{x}_r - \dot{x}) + K_p(x_r - x),$$

where K_p and K_d were diagonal matrices with all entries equal to 10 and 5, respectively.

The controllers used damped pseudoinverses [Chiaverini 1997] to ensure stability near singularities. The damping factor was always set to $5 \cdot 10^{-3}$. To avoid interferences between tasks, the nullspace projection matrices were computed without any damping.

5.6.3 *Test 1 - Multi-task Position/Force Control*

We tested our control framework - Task Space Inverse Dynamics - against the Unifying Framework (UF) [Peters et al. 2007] and the Whole-Body Control Framework (WBCF) [Sentis and Khatib 2005]. In this test the robot performs four tasks, listed here with decreasing priority:

F: 3 DoF, control the contact force exerted with the right hand on the wall

T2: 3 DoF, track a circular trajectory with the left hand

T1: 1 DoF (x axis), track a sinusoidal reference with the neck base

P: 23 DoF, maintain the initial joint posture

The first three tasks are compatible, so the robot should be able to perform them with negligible errors. Table 5.3 reports the mean error norm for each task and the mean computation time of the control loop. Referring to the criteria proposed in section 5.1.1 to evaluate a control framework, let us clarify how they relate to the data reported in table 5.3. The error on the primary task F concerns the *soundness*, the errors on the secondary tasks T2, T1, P concern the *optimality*, and the computation time concerns the *efficiency*. As expected, the UF performs poorly on the non-primary tasks, because it is not optimal. Both the WBCF and the TSID achieve good tracking on the tasks F, T2 and T1, but the computation time of the WBCF is more than twice the computation time of our framework.

5.6.4 *Test 2 - Soft Force Control*

In this test the robot controlled the force applied on an elastic environment, using our Task Space Inverse Dynamics control framework. In particular we used the "spring contact model" presented in 5.5.3.1. Since in real world scenarios it is difficult to estimate the contact stiffness k_s, we tested the robustness of the controller with respect to errors in the estimate of k_s. The real contact stiffness was set to 10^3, while the estimates of k_s used in the controller were set to five values in the range $[10^2, 10^4]$. The feedback gains of

Table 5.3: Results of test 1. We tested three controllers (Whole Body Control Framework, Unifying Framework, Task Space Inverse Dynamics) on the same four tasks (F, T2, T1, P); we measured the mean error norm for each task and the mean computation time of the control loop.

Related to	*Soundness*		*Optimality*		*Efficiency*
Controller	F Error	T2 Error	T1 Error	P Error	Computation
	(N)	(mm)	(mm)	(°)	Time (ms)
TSID	**0.46**	**0.3**	**0.7**	**10.4**	**0.298**
WBCF	0.46	0.1	0.9	10.4	0.681
UF	0.46	139.7	145.2	6.2	0.316

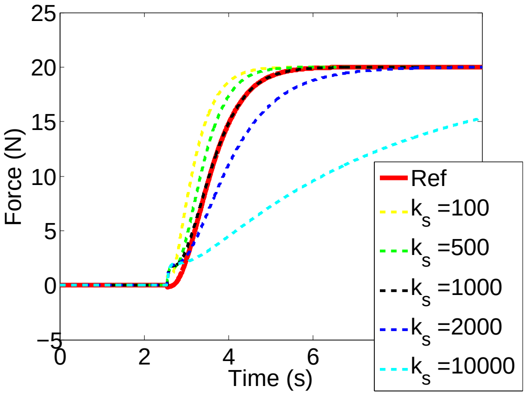


Figure 5.4: Force applied by the robot on the environment as the estimate of the contact stiffness k_s ranges from 10^2 to 10^4 (the real contact stiffness is 10^3).

the force control loop were $K_p = 50$ and $K_d = 25$. Fig. 5.4 shows the quality of the tracking of the reference force as k_s changes. As expected, when the estimate of k_s is precise the tracking of the reference force is almost perfect.

5.7 CONCLUSIONS AND FUTURE WORK

We presented and validated a new theoretical control framework, called Task Space Inverse Dynamics, for prioritized position and force control of fully-actuated rigid robots. To the best of our knowledge, this framework outperforms every other control framework with equal capabilities. Its main features are:

1. *optimality*: the error of each task is minimized under the constraint of not affecting any higher priority task
2. *capabilities*: our framework allows for tracking position control, soft contact force control, rigid contact force control

3. *efficiency*: solutions can be computed in O(n) using the Recursive Newton-Euler Algorithm because it does not need the joint space inertia matrix M or the task space inertia matrices Λ's

We compared the presented control framework with other two state-of-the-art control frameworks (UF and WBCF), both analytically and through simulation tests. The results reported in table 5.3 confirm that our framework outperforms the other two frameworks, either in terms of optimality or in terms of efficiency.

In the next chapter we extend this framework and the related analysis to the control of floating-base (i.e. underactuated) robots.

6

"TASK SPACE INVERSE DYNAMICS" — PRIORITIZED POSITION-FORCE CONTROL OF FLOATING-BASE CONSTRAINED ROBOTS

This chapter extends the framework presented in chapter 5 to the control of underactuated mechanical systems. In particular, we deal with floating-base systems, a special class of underactuated systems in which the base is a moving body. The contributions of this chapter regard the case in which these systems are constrained, namely they cannot move freely because their motion is constrained (for instance by a rigid contact). This scenario is very common in humanoid robotics, because it occurs any time the robot is in contact with the ground — typically with one or both feet.

6.1 INTRODUCTION AND MOTIVATIONS

A system is underactuated if the number of its actuators is less than the number of its DoFs. Underactuated mechanical systems arise in many applications, such as space and undersea robots, mobile robots, flexible robots, walking, brachiating and gymnastic robots. Well-known underactuated systems are the acrobot [Spong 1995] and the cart-pole [Spong 1998], both of which have one actuator and two DoFs. Humanoid robots are underactuated because they typically have n actuators and $n + 6$ DoFs, where n is the number of joints of the robot and 6 are the DoFs representing position and orientation of the base of the robot.

Controlling underactuated systems is more challenging than controlling fully-actuated systems. Fully-actuated mechanical systems are feedback linearizable: most problems for fully-actuated mechanical systems can be reduced to equivalent problems for linear systems. The equations of motion of a fully-actuated mechanical system are:

Fully-actuated systems are feedback linearizable, but underactuated systems are not.

$$M(q)\ddot{q} + h(q, \dot{q}) = \tau,$$

where $q \in \mathbb{R}^n$ is the vector of generalized coordinates, $\tau \in \mathbb{R}^n$ is the vector of generalized actuator forces, $M(q) \in \mathbb{R}^{n\times n}$ is the positive definite inertia matrix, and $h(q, \dot{q}) \in \mathbb{R}^n$ is the bias vector, which typically contains Coriolis, centrifugal and gravity forces. To move such a system from an initial configuration q_0 to a final configuration q^* in a finite time T, one can compute an acceleration trajectory $\ddot{q}(t)^*$ such that:

$$q^* = q_0 + \int\limits_0^T \int\limits_0^T \ddot{q}(t)^* \, \mathrm{d}\, t^2,$$

where we assumed zero initial velocity. Then the generalized force trajectory that moves the system from q_0 to q^* is given by:

$$\tau(t)^* = M\ddot{q}(t)^* + h$$

On the other hand, underactuated systems are not feedback linearizable. The equations of motion of an underactuated mechanical system with n DoFs and $a(< n)$ actuators are:

$$M(q)\ddot{q} + h(q,\dot{q}) = S^T\tau, \tag{6.1}$$

where $S^T \in \mathbb{R}^{n \times a}$ is a matrix that selects the a actuated degrees of freedom, and $\tau \in \mathbb{R}^a$. The problem of moving such a system from an initial configuration q_0 to a final configuration q^* is not trivial. At any instant of time the system can only accelerate in certain directions. In particular the system accelerations $\ddot{q}$ lie on an a-dimensional manifold, which is configuration dependent:

$$\ddot{q} \in \{z \mid z = M(q)^{-1}(S^T\tau - h(q,\dot{q})), \tau \in \mathbb{R}^a\}$$

This implies that a particular acceleration trajectory that drives the system from q_0 to q^* may be unfeasible for an underactuated system. However, there may exists another acceleration trajectory that also drives the system from q_0 to q^*, which is feasible. It is important to note that many underactuated systems are *controllable*, namely it is always possible to find an input trajectory $\tau(t)$ that steers the state of the system from q_0 to q^* in finite time T, for any q_0, q^*, T. The problem of finding feasible acceleration trajectories is tackled in the field of *non-holonomic* motion planning, that models the underactuation as a non-integrable constraint on the system's accelerations.

Non-holonomic motion planning deals with the problem of finding feasible trajectories for underactuated systems.

However, rather than focusing on motion planning, in this chapter we focus on the problem of controlling underactuated systems. We assume that a desired acceleration is always given, in either task space (denoted $\ddot{x}^*$) or configuration space (denoted $\ddot{q}^*$). In general we assume that desired accelerations may be unfeasible, so we seek the control inputs that generate accelerations that are "as close as possible" to the desired accelerations. The same applies in case we control contact forces rather than accelerations.

We start with the motion control of *unconstrained* floating-base robots (which are a class of underactuated systems), both in configuration space (section 6.2.1) and in task space (section 6.2.2). Floating-base systems can be modeled as fixed-base systems, by installing a 6-DoF joint between the fixed base and the body representing the floating base [Featherstone 2008]. Then, we move to the control of both motion and contact forces of *constrained* floating-base systems (section 6.3). We show how the presented control laws simplify when these systems are *sufficiently constrained* (see section 6.22), i.e. they are subject to a number of constraints that is sufficient to make their constrained motion completely feasible. Along the way, we pay particular attention to the *computational cost* of the devised control laws, trying to avoid, whenever possible, the computation of the joint space inertia matrix M.

6.1.1 *Dynamics of Floating-Base Systems*

We introduce the notation that we use in this chapter, which is typical of floating-base mechanical systems. First of all we introduce two selection matrices:

$$\begin{aligned} S &= \begin{bmatrix} 0_{n\times 6} & I_n \end{bmatrix} \\ U &= \begin{bmatrix} I_6 & 0_{6\times n} \end{bmatrix}, \end{aligned} \tag{6.2}$$

where $S \in \mathbb{R}^{n\times n+6}$ is the matrix selecting the n actuated joints, whereas $U \in \mathbb{R}^{6\times n+6}$ is the matrix selecting the 6 (floating-base) passive joints. The state of a floating-base system is not uniquely represented by just its joint positions $q_j \in \mathbb{R}^n$. A 6-DoF virtual joint is attached to the robot base to encode its position and orientation in the space:

$$q = \begin{bmatrix} x_b \\ q_j \end{bmatrix} \qquad x_b \in \mathbb{R}^6,\ q_j \in \mathbb{R}^n$$

The Jacobian $J = \frac{\partial f(q)}{\partial q} \in \mathbb{R}^{m\times n+6}$ of a generic point of the robot can then be decomposed as:

$$J = \begin{bmatrix} \frac{\partial f(q)}{\partial x_b} & \frac{\partial f(q)}{\partial q_j} \end{bmatrix} = \begin{bmatrix} J_b & J_j \end{bmatrix} \qquad J_b \in \mathbb{R}^{m\times 6},\ J_j \in \mathbb{R}^{m\times n}$$

The equations of motion of a floating-base robot are:

$$M(q)\ddot{q} + h(q,\dot{q}) = S^T\tau, \tag{6.3}$$

where $M \in \mathbb{R}^{n+6\times n+6}$ is the positive definite joint space inertia matrix, $\ddot{q} \in \mathbb{R}^{n+6}$ are the joint accelerations, $h(q,\dot{q}) \in \mathbb{R}^{n+6}$ are the bias forces, and $\tau \in \mathbb{R}^n$ are the actuated joint torques. This equation can also be written in block matrix form, separating the motion of the floating-base from the motion of the real joints:

$$\begin{bmatrix} M_b & M_{bj} \\ M_{bj}^T & M_j \end{bmatrix} \begin{bmatrix} \ddot{x}_b \\ \ddot{q}_j \end{bmatrix} + \begin{bmatrix} h_b \\ h_j \end{bmatrix} = \begin{bmatrix} 0_{6\times n} \\ I_n \end{bmatrix} \tau \tag{6.4}$$

where $M_j \in \mathbb{R}^{n\times n}$, $M_b \in \mathbb{R}^{6\times 6}$, $M_{bj} \in \mathbb{R}^{6\times n}$, $\ddot{x}_b \in \mathbb{R}^6$, and $\ddot{q}_j \in \mathbb{R}^n$. The inverse of the inertia matrix M^{-1} may also be represented as a block matrix:

$$\begin{aligned} M^{-1} &= \begin{bmatrix} N_b & N_{bj} \\ N_{bj}^T & N_j \end{bmatrix} \\ N_b &= (M_b - M_{bj}M_j^{-1}M_{bj}^T)^{-1} \\ N_j &= (M_j - M_{bj}^T M_b^{-1} M_{bj})^{-1} \\ N_{bj} &= -N_b M_{bj} M_j^{-1} = -M_b^{-1} M_{bj} N_j \end{aligned} \tag{6.5}$$

The pseudoinverses of S and U, weighted with M^{-1}, are going to play an important role in the derivation of some control laws, so we report them here:

$$\bar{S} = S^{+}_{M^{-1}} = M^{-1}S^T(SM^{-1}S^T)^{-1} = \begin{bmatrix} N_{bj} \\ N_j \end{bmatrix} N_j^{-1} = \begin{bmatrix} -M_b^{-1}M_{bj} \\ I_n \end{bmatrix}$$

$$\bar{U} = U^{+}_{M^{-1}} = M^{-1}U^T(UM^{-1}U^T)^{-1} = \begin{bmatrix} N_b \\ N_{bj}^T \end{bmatrix} N_b^{-1} = \begin{bmatrix} I_6 \\ -M_j^{-1}M_{bj}^T \end{bmatrix}$$

Note that the final expressions of $\bar{S}$ and $\bar{U}$ do not depend on the whole inertia matrix, but they depend on M_b and M_{bj} only, so they are easier to compute. This decomposition of the inverse of the inertia matrix is going to be used in the next sections (see the appendix B.2 for the proof):

$$M^{-1} = \bar{S}N_j\bar{S}^T + U^T M_b^{-1} U \tag{6.6}$$

6.2 UNCONSTRAINED FLOATING-BASE SYSTEMS

This section reviews the basic techniques for the position control of unconstrained floating-base systems, both in joint space and in task space.

6.2.1 *Joint Space Control*

Since we cannot control the entire state of an underactuated system, we aim to control a subset of its joints, which typically is either the set of active joints or the set of passive joints. It is well-known that the portion of the dynamics corresponding to the active DoFs may be linearized by nonlinear feedback [Spong 1994]. Furthermore, under a condition called "*strong inertial coupling*", also the portion of the dynamics corresponding to the passive DoFs may be linearized by nonlinear feedback.

6.2.1.1 *Collocated Partial Feedback Linearization (PFL)*

The problem of controlling the motion of the active joints q_j, may be formulated as:

$$\tau^* = \underset{\tau\in\mathbb{R}^n}{\operatorname{argmin}} ||\ddot{q}_j - \ddot{q}_j^*||^2$$
$$s.t. \quad M\ddot{q} + h = S^T\tau,$$

where $\ddot{q}_j^* \in \mathbb{R}^n$ are the desired active joint accelerations. This problem has a unique solution (see appendix B.3 for the complete derivation), commonly known as *Collocated Partial Feedback Linearization* [Spong 1998], and it is:

$$\tau^* = N_j^{-1}\ddot{q}_j^* + \bar{S}^T h \tag{6.7}$$

This control law cannot be computed using a standard inverse dynamics algorithm, such as the RNEA. For underactuated systems the *inverse dynamics* problem is actually a *hybrid dynamics* problem. Hybrid dynamics is a generalization of forward and inverse dynamics, in which the forces are known at some joints, the accelerations at the rest, and the task is to calculate the unknown forces and accelerations [Featherstone 2008]. In particular, for a floating-base robot, we know the accelerations at the active joints, the forces (which are zero) at the base joints, and we want to compute the joint torques. Even though we are not interested in the base accelerations, they need to be computed to calculate the joint torques. Without entering into the details of the hybrid dynamics algorithms, we just report that the control law (6.7) can be computed with a computational cost of O$(2n)$, that is approximately twice the cost of the RNEA. This is the minimum cost for an “inverse dynamics” control law for floating-base robots, so, in the following, we aim at getting all the control laws in the form of (6.7).

Partial feedback linearization can be computed with an hybrid dynamics algorithm.

In conclusion, applying the control law (6.7) to a floating-base system, the resulting motion is:

$$\begin{aligned} \ddot{q}_j &= \ddot{q}_j^* \\ \ddot{x}_b &= -M_b^{-1}(M_{bj}\ddot{q}_j^* + h_b) \end{aligned}$$

Only the part of the dynamics describing the motion of the active joint has been linearized; that is the reason why this technique is called *partial* feedback linearization.

6.2.1.2 *Non-Collocated Partial Feedback Linearization (PFL)*

Under the condition of *strong inertial coupling*, it is possible to control the passive DoFs of underactuated systems [Spong 1998]. In particular, in case of floating-base robots, we want to control the floating-base accelerations, that is we want to solve the following problem:

$$\begin{aligned} \tau^* = \operatorname*{argmin}_{\tau \in \mathbb{R}^n} & \, ||\ddot{x}_b - \ddot{x}_b^*||_E^2 \\ s.t. & \quad M\ddot{q} + h = S^T\tau, \end{aligned}$$

where $\ddot{x}_b^* \in \mathbb{R}^6$ are the desired base accelerations and $E > 0$ is an arbitrary weight matrix. Depending on the rank of M_{bj}, this problem may have one or infinite solutions. In general, the solutions take this form (see appendix B.4 for the complete derivation):

$$\begin{aligned} \tau^* = &-(N_b M_{bj} M_j^{-1})^+_{W,E}(\ddot{x}_b^* + N_b \bar{U}^T h) \\ &+ (I - (N_b M_{bj} M_j^{-1})^+_{W_E} N_b M_{bj} M_j^{-1})\tau_0, \end{aligned} \tag{6.8}$$

where $W > 0$ is an arbitrary weight matrix and τ_0 is an arbitrary vector. This expression gives the solution that minimizes $||\tau - \tau_0||_W$. The condition of *strong inertial coupling* requires that rank$(M_{bj}) \geq 6$. If this condition is satisfied then we can get $\ddot{x}_b = \ddot{x}_b^* \; \forall \ddot{x}_b^* \in \mathbb{R}^6$, and (6.8) simplifies to:

Under the condition of strong inertial coupling we have complete control of the passive DoFs.

$$\tau^* = -M_j M_{bj}^+(N_b^{-1}\ddot{x}_b^* + \bar{U}^T h) + M_j(I - M_{bj}^+ M_{bj})M_j^{-1}\tau_0 \tag{6.9}$$

Alternatively, this control law can be also put in the convenient hybrid dynamics form (such as (6.7)):

$$\tau^* = N_j^{-1}\underbrace{\left(-M_{bj}^+(M_b\ddot{x}_b^* + h_b) + (I - M_{bj}^+M_{bj})N_j\tilde{\tau}_0\right)}_{\ddot{q}_j^*} + \bar{S}^T h, \tag{6.10}$$

Under the condition of *strong inertial coupling*, both (6.9) and (6.10) result in $\ddot{x}_b = \ddot{x}_b^*$, but they use the task nullspace (if any) to minimize different quantities. In particular (6.9) minimizes $||M_j^{-1}(\tau - \tau_0)||$, whereas (6.10) minimizes $||N_j(\tau - \bar{S}^T h - \tilde{\tau}_0)||$.

6.2.2 *Task Space Partial Feedback Linearization (PFL)*

Task space PFL generalizes PFL to the control in task space.

We can exploit the principle of PFL to control the system in *task space*. Task space PFL [Shkolnik and Tedrake 2008] is a generalization of collocated and non-collocated PFL. As usual, we can formulate the control problem as a constrained minimization:

$$\tau^* = \underset{\tau\in\mathbb{R}^n}{\operatorname{argmin}} ||J\ddot{q} + \dot{J}\dot{q} - \ddot{x}^*||^2$$
$$s.t. \quad M\ddot{q} + h = S^T\tau,$$

where $\ddot{x}^* \in \mathbb{R}^m$ are the desired task space accelerations, and $J = \frac{\partial x}{\partial q} \in \mathbb{R}^{m\times n+6}$ is the task Jacobian. If $m < n$ this problem has infinite solutions (see appendix B.5 for the derivation), which can be written as:

$$\begin{aligned}\tau^* &= N_j^{-1}\ddot{q}_j^* + \bar{S}^T h\\ \ddot{q}_j^* &= (J\bar{S})_W^+(\ddot{x}^* - \dot{J}\dot{q} + J_bM_b^{-1}h_b) + (I - (J\bar{S})_W^+J\bar{S})\ddot{q}_{j0},\end{aligned} \tag{6.11}$$

where $\ddot{q}_{j0} \in \mathbb{R}^n$ is an arbitrary vector, and $W > 0, W \in \mathbb{R}^{n\times n}$ is an arbitrary weight matrix. The control law takes the convenient hybrid dynamics form of (6.7). Note the introduction of a new Jacobian $\hat{J}$, defined as:

$$\hat{J} = J\bar{S} = J_j - J_bM_b^{-1}M_{bj},$$

The generalized Jacobian considers the dynamic singularities of floating-base systems.

which is commonly known in space robotics as the *Generalized Jacobian* [Umetani and Yoshida 1989]. This Jacobian takes into account the dynamic coupling between the base and the joints of the robot. Through a singular value decomposition (see appendix A.1) of $\hat{J}$ we can find the so-called *dynamic singularities* [Papadopoulos and Dubowsky 1992], that are configurations in which the control point cannot accelerate in some directions.

Differently from the fully-actuated case (see chapter 5) it is not possible to completely decouple kinematics and dynamics. Part of the robot dynamics has to be taken into account in the passage from task space to joint space accelerations. However, this control law does not require the computation of the whole mass matrix, but it needs only M_b and M_{bj}, which can be computed, for instance, with 6 iterations of the RNEA. Especially for humanoid robots, in which n is typically much larger than 6, not having to compute M_j can heavily improve the performance of the controller.

The collocated and non-collocated PFL can be derived as special cases of the task space PFL (6.11) (see appendix B.5). In particular, if we set $J = S$ and $\ddot{x}^* = \ddot{q}_j^*$ we get the collocated PFL. On the other hand, if we set $J = U$ and $\ddot{x}^* = \ddot{x}_b^*$ we get the non-collocated PFL.

6.3 CONSTRAINED FLOATING-BASE SYSTEMS

This section deals with the control of constrained floating-base systems. First we define constrained floating-base systems and we derive their equations of motion. Then we review the classic approaches for the control of these systems, which are based on the "constraint nullspace projection" technique. Starting from section 6.3.5, we tackle the problem of controlling constrained floating-base systems. First we deal with position control, including joint space, task space and prioritized multi-task control. Then we move to force control, considering both the control of all the constraint forces and the control of a subset of the constraint forces.

6.3.1 *Dynamics of Constrained Floating-Base Systems*

Consider a mechanical system that is subject to a set of k nonlinear equality constraints:

$$c(q, \dot{q}, t) = 0$$

For instance the system could be in rigid contact with the environment and the constraints could represent the fact that the contact point(s) do not move in the constrained directions. This is the typical constraint that we take as reference throughout the rest of this chapter, so we use interchangeably the terms "contact force/Jacobian" and "constraint force/Jacobian". Differentiating the constraints twice with respect to time we get:

In this thesis we implicitly assume that constraints are due to rigid contacts.

$$\begin{aligned} J_c \dot{q} &= 0 \\ J_c \ddot{q} &= -\dot{J}_c \dot{q}, \end{aligned} \tag{6.12}$$

where $J_c = \frac{\partial c}{\partial q} \in \mathbb{R}^{k \times n+6}$ is the constraint Jacobian. The equations of motion of a constrained floating-base system are then:

$$M\ddot{q} + h - J_c^T f = S^T \tau, \tag{6.13}$$

where $f \in \mathbb{R}^k$ are the constraint forces, which prevent the system from violating the constraints. The introduction of these unknown contact forces f complicates the control problem because we need to know f in order to compute τ.

A possible way around this problem is to measure the contact forces using force/torque sensors. However, force/torque measurements are noisy and introduce delay in the control action. Moreover, in the field of walking robots, very few platforms are equipped with enough sensors to measure contact

Relying on force/torque sensors is not the only (or the best) solution.

forces on their whole body, so this approach may be unfeasible. The next sections present different approaches to solve this problem without force/torque sensors.

6.3.2 *Constraint Nullspace Projection*

The classical way to tackle this problem [Park 2006] is to exploit the fact that the contact forces may be computed as a function of τ:

$$f = \Lambda_c(-\dot{J}_c\dot{q} + J_c M^{-1}(h - S^T\tau)),$$

where $\Lambda_c = (J_c M^{-1} J_c^T)^{-1}$ is the constraint space inertia matrix. Substituting f in (6.13) we get:

$$M\ddot{q} + \bar{N}_c^T h + J_c^T \Lambda_c \dot{J}_c \dot{q} = \bar{N}_c^T S^T \tau, \tag{6.14}$$

where $\bar{J}_c = J_{c\,M^{-1}}^{+}$ is the *dynamically consistent* Jacobian pseudoinverse [Khatib 1987], and $\bar{N}_c = I - \bar{J}_c J_c$ is its nullspace projector. This equation describes the motion of the system without the contact forces, so it can be used to compute the control torques τ^* that generate the desired joint accelerations $\ddot{q}^*$:

$$\tau^* = (\bar{N}_c^T S^T)_W^{+}(M\ddot{q}^* + \bar{N}_c^T h + J_c^T \Lambda_c \dot{J}_c \dot{q}) \tag{6.15}$$

The main drawback of this equation is that it requires the computation of M^{-1} — which is hidden inside $\bar{N}_c$ — an operation that may be computationally expensive for a real-time controller.

A more recent approach [Aghili 2005; Righetti et al. 2011a] suggests to eliminate the constraint forces from (6.13) by projecting it in the nullspace of the constraints. This is done by multiplying the robot dynamics times $N_{cW}^T = (I - J_{cW}^{+} J_c)^T$, so that we get (see the appendix B.6 for a detailed description):

$$\begin{aligned} N_{cW}^T[M\ddot{q} + h - J_c^T f = S^T\tau] \\ N_{cW}^T(M\ddot{q} + h) = N_{cW}^T S^T \tau \end{aligned} \tag{6.16}$$

This equation is a convenient alternative to (6.14), because it does not use M^{-1} and it is overall simpler. To get the simplest formulation, in the following we are going to use $W = I$, so we can also exploits the symmetry of orthogonal projectors: $N_c^T = N_c$. The control torques to get the desired joint accelerations $\ddot{q}^*$ can then be computed as:

$$\tau^* = (N_c^T S^T)^+ N_c(M\ddot{q}^* + h)$$

This expression, differently from (6.15), does not require the computation of the inertia matrix M.

6.3.3 *Constrained Direct Dynamics*

Even though the focus of this work is on the control side, it is sometimes necessary to express the joint accelerations $\ddot{q}$ as a function of the joint torques τ (i.e. to solve the direct dynamics problem). Using (6.14) we can compute $\ddot{q}$ by multiplying both sides times M^{-1}:

$$\ddot{q} = M^{-1}(\bar{N}_c^T(S^T\tau - h) - J_c^T\Lambda_c\dot{J}_c\dot{q}) \tag{6.17}$$

As we already mentioned, this equation is complex and in the previous section we reported a convenient alternative, that is (6.16). Unfortunately, we cannot use (6.16) as it is to compute $\ddot{q}$, because the matrix that premultiplies $\ddot{q}$, that is $N_{cW}^T M$, is not invertible. To solve this issue we can multiply (6.12) times J_{cW}^+ and sum the resulting equation to (6.16) to get [Aghili 2005]:

$$\begin{aligned}
N_{cW}^T(M\ddot{q} + h) + J_{cW}^+ J_c\ddot{q} &= N_{cW}^T S^T\tau - J_{cW}^+\dot{J}_c\dot{q} \\
\underbrace{(N_{cW}^T M + J_{cW}^+ J_c)}_{M_c}\ddot{q} &= N_{cW}^T(S^T\tau - h) - J_{cW}^+\dot{J}_c\dot{q}
\end{aligned}$$

Since Aghili [2003] proved that M_c is always invertible, we can compute $\ddot{q}$ as:

$$\ddot{q} = M_c^{-1}(N_{cW}^T(S^T\tau - h) - J_{cW}^+\dot{J}_c\dot{q}) \tag{6.18}$$

Moreover, differently from (6.17), this expression does not require the set of constraints to be linearly independent.

6.3.4 *Sufficiently Constrained Floating-Base Systems*

Using the *singular value decomposition* (see appendix A.1) of the constraint Jacobian J_c we may gain some insights into the dynamics of constrained systems. Without loss of generality, we assume that J_c is full rank (if it is not, we can always find a full-rank Jacobian that represents the system's constraints), hence its SVD takes the form:

$$J_c = U\Sigma V = U\begin{bmatrix}\Sigma_1 & 0\end{bmatrix}\begin{bmatrix}V_1^T \\ V_2^T\end{bmatrix} = U\Sigma_1 V_1^T,$$

where $V_1 \in \mathbb{R}^{n+6\times k}$ is an orthonormal basis of the *active space* of J_c, whereas $V_2 \in \mathbb{R}^{n+6\times n+6-k}$ is an orthonormal basis of the *nullspace* of J_c. By *active space* of a matrix, we mean the subspace that is the orthogonal complement of the *nullspace* of the matrix. We can then write the second derivative of the constraints as:

$$\begin{aligned}
J_c\ddot{q} &= -\dot{J}_c\dot{q} \\
\ddot{q} &= -J_c^+\dot{J}_c\dot{q} + V_2\ddot{q}_c,
\end{aligned}$$

where $\ddot{q}_c \in \mathbb{R}^{n+6-k}$ represents the actual freedom of motion of the constrained system. Substituting this expression of $\ddot{q}$ into the system dynamics we get:

$$M(-J_c^+\dot{J}_c\dot{q} + V_2\ddot{q}_c) + h - V_1\Sigma_1 U^T f = \tau$$

We can now project the system dynamics onto the nullspace of J_c multiplying times V_2^T, so that we get the *motion dynamics*:

$$V_2^T M V_2 \ddot{q}_c - V_2^T M J_c^+ \dot{J}_c \dot{q} + V_2^T h = V_2^T \tau \tag{6.19}$$

Similarly, we can get the *constraint force dynamics* by multiplying times $V_1^T M^{-1}$:

$$V_1^T M^{-1} V_1 \Sigma_1 U^T f - V_1^T J_c^+ \dot{J}_c \dot{q} + V_1^T M^{-1} h = V_1^T M^{-1} \tau \tag{6.20}$$

We decoupled the motion dynamics and the constraint force dynamics.

Equation (6.19) describes the relationship between τ and $\ddot{q}_c$, independently of the constraint forces f. On the other hand, equation (6.20) describes the relationship between τ and f, independently of the joint accelerations $\ddot{q}_c$. In other words, these two sets of equations decouple the motion dynamics and the constraint force dynamics.

Looking at (6.19) it is clear that the (constrained) motion of the system is completely controllable, because V_2^T is full-row rank and, as a consequence, $V_2^T M V_2$ is invertible. This implies that $\forall \ddot{q}_c^* \in \mathbb{R}^{n+6-k}$ there exists at least one value of τ such that $\ddot{q}_c = \ddot{q}_c^*$. In particular, that value of τ would be:

$$\tau^* = V_2 V_2^T (M \underbrace{(V_2 \ddot{q}_c^* - J_c^+ \dot{J}_c \dot{q})}_{\ddot{q}^*} + h) = N_c (M \ddot{q}^* + h)$$

However, if the system is underactuated the situation changes. The motion dynamics (6.19) becomes:

$$V_2^T M V_2 \ddot{q}_c - V_2^T M J_c^+ \dot{J}_c \dot{q} + V_2^T h = V_2^T S^T \tau \tag{6.21}$$

While V_2^T is always full-row rank, the matrix $V_2^T S^T \in \mathbb{R}^{n+6-k \times n}$ may be not. In that case, there would be some values of $\ddot{q}_c$ that could not be generated by any τ. In particular, if $k < 6$ then $V_2^T S^T$ is skinny, hence it cannot be full-row rank by definition. If instead $k \geq 6$ then $V_2^T S^T$ can be (and often times it is) full-row rank, so any motion can be achieved. In the following, if an underactuated system is subject to enough constraints to verify the condition:

$$\text{rank}(V_2^T S^T) = n + 6 - k, \tag{6.22}$$

then we say that the system is "*sufficiently constrained*". Note that the condition (6.22) is not equivalent to $k \geq 6$, but it is tighter, meaning that (6.22) implies $k \geq 6$, but $k \geq 6$ does not imply (6.22). When a floating-base system is *sufficiently constrained*, it means that:

- inside the constraint-consistent motion manifold, the system behaves as if it were fully-actuated;
- the contact constraints reduce the degrees of freedom of the system, but the underactuation does not reduce them any further.

We have to remember that we are going to implement these controllers on a digital machine, on which the concept of “rank of a matrix” depends on the threshold used to discriminate when a value is small enough to be considered zero. Typical values for this threshold are around 10^{-10}, but when evaluating the rank of $V_2^T S^T$, we may want to set that threshold higher than usual. Using a threshold that is too small, we may believe that a system is *sufficiently constrained*, while to accelerate in certain directions it requires joint torques that are so large, that from a practical standpoint the system is actually *not sufficiently constrained*. In the next sections we are going to exploit this analysis to simplify the control of floating-base systems.

6.3.5 *Joint Control*

We want to control the motion of a constrained floating-base system in joint space, that is we want to solve this problem:

$$\begin{aligned}\tau^* = \underset{\tau\in\mathbb{R}^n}{\operatorname{argmin}}\, &||\ddot{q} - \ddot{q}^*||_E^2 \\ s.t. \quad & M\ddot{q} + h - J_c^T f = S^T\tau \\ & J_c\ddot{q} = -\dot{J}_c\dot{q},\end{aligned}$$

where $E > 0$ is an arbitrary weight matrix, and $\ddot{q}^* \in \mathbb{R}^{n+6}$ are the desired joint accelerations, which we assume to be constraint consistent. We can easily check whether the desired joint accelerations are constraint consistent by computing:

$$J_c\ddot{q}^* + \dot{J}_c\dot{q},$$

and verifying that it is zero. In case it is not, we can project the desired joint accelerations into the constraint-consistent motion manifold, to get the “closest” constraint-consistent joint accelerations.

In general there is no τ such that $\ddot{q} = \ddot{q}^*$, even if $\ddot{q}^*$ are constraint consistent, because the system is subject to the underactuation constraints, which may make the desired motion unfeasible. The solution of this problem is (see appendix B.7 for the derivation):

$$\tau^* = (EM_c^{-1}N_cS^T)^+ EM_c^{-1}N_c(M\ddot{q}^* + h)$$

If we want to minimize the norm of the joint acceleration error, we have to set $E = I$. Doing that, we get a control law that is computationally expensive, because it requires the computation of M (which appears inside M_c). However, if the system is *sufficiently constrained* (i.e. $V_2^T S^T$ is full-row rank) then τ^* will result in $\ddot{q} = \ddot{q}^*$. In this case the matrix E does not affect the solution, so we can set it to the most convenient value, that is $E = M_c$:

$$\tau^* = (N_cS^T)^+ N_c(M\ddot{q}^* + h) \tag{6.23}$$

This control law is very efficient, because it can be computed with one iteration of RNEA. Note that using (6.23) when the system is not *sufficiently* *constrained* means that we do not minimize the joint acceleration error, but a weighted version of it. Alternatively, rather than using N_c, we could use any matrix whose columns span the nullspace of J_c, e.g. the matrix V_2 from the SVD of J_c:

$$\tau^* = (V_2^T S^T)^+ V_2^T (M\ddot{q}^* + h)$$

This form has the advantage of computing the pseudoinverse of a smaller matrix: $V_2^T S^T \in \mathbb{R}^{n+6-k\times n}$, while $N_c S^T \in \mathbb{R}^{n+6\times n}$.

6.3.6 *Task Space Control*

We want to control the motion of a floating-base constrained system in task space, that is we want to solve this problem:

$$\begin{aligned} \tau^* = &\underset{\tau\in\mathbb{R}^n}{\operatorname{argmin}} ||J\ddot{q} + \dot{J}\dot{q} - \ddot{x}^*||_E^2 \\ &s.t. \quad M\ddot{q} + h - J_c^T f = S^T\tau \\ &\qquad\quad J_c\ddot{q} = -\dot{J}_c\dot{q}, \end{aligned} \tag{6.24}$$

where $E > 0$ is an arbitrary weight matrix, $\ddot{x}^* \in \mathbb{R}^m$ are the desired task space accelerations and $J = \frac{\partial x}{\partial q} \in \mathbb{R}^{m\times n+6}$ is the task Jacobian. Using (6.18) (with $W = I$) we can formulate an equivalent unconstrained problem:

$$\tau^* = \underset{\tau\in\mathbb{R}^n}{\operatorname{argmin}} ||JM_c^{-1}(N_c(S^T\tau - h) - J_c^+\dot{J}_c\dot{q}) + \dot{J}\dot{q} - \ddot{x}^*||_E^2$$

This problem has in general infinite solutions, which can be expressed as:

$$\begin{aligned} \tau^* = &(JM_c^{-1}N_cS^T)^+_{E,W}(\ddot{x}^* - \dot{J}\dot{q} + JM_c^{-1}(N_ch + J_c^+\dot{J}_c\dot{q})) \\ &+ (I - (JM_c^{-1}N_cS^T)^+_{E,W}JM_c^{-1}N_cS^T)\tau_0, \end{aligned}$$

where $\tau_0 \in \mathbb{R}^n$ is an arbitrary torque vector, $W > 0, W \in \mathbb{R}^{n\times n}$ weighs the distance between τ and τ_0, and $E > 0, E \in \mathbb{R}^{m\times m}$ weighs the task acceleration error. This expression is rather complex and it requires the computation of M^{-1}. If the matrix $JM_c^{-1}N_cS^T \in \mathbb{R}^{m\times n}$ is full-row rank, then the task is feasible and the matrix E does not affect the solution. Unfortunately, there is no choice of E (or W) that simplifies this expression, but we can exploit our previous analysis to find a way around this issue.

We know that if the system is *sufficiently constrained* (i.e. rank$(V_2^T S^T) = n + 6 - k$) then the constraint-consistent motion is completely attainable. If this condition is verified, and the desired task is constraint consistent (i.e. rank$(JN_c) =$ rank(J)), then we can solve the problem at kinematic level, neglecting the underactuation constraint and considering only the contact constraints. Once we have computed the desired joint accelerations, we can compute the corresponding joint torques using (6.23). In conclusion, if the system is *sufficiently constrained* then the problem (6.24) is equivalent to this problem:

$$\begin{aligned} \tau^* =&(N_cS^T)^+N_c(M\ddot{q}^* + h) \\ \ddot{q}^* =&\underset{\ddot{q}\in\mathbb{R}^n}{\operatorname{argmin}} ||J\ddot{q} + \dot{J}\dot{q} - \ddot{x}^*||_E^2 \\ &s.t. \quad J_c\ddot{q} = -\dot{J}_c\dot{q} \end{aligned}$$

The solutions of this problem are:

$$\ddot{q}^* = -J_c^+ \dot{J}_c \dot{q} + (JN_c)^+ (\ddot{x}^* - \dot{J}\dot{q} + JJ_c^+ \dot{J}_c \dot{q}) + (I - (JN_c)^+ JN_c)\ddot{q}_0, \quad (6.25)$$

where $\ddot{q}_0 \in \mathbb{R}^{n+6}$ is an arbitrary joint acceleration vector. Note that $\ddot{q}^*$ can always be generated, because it is constraint consistent.

6.3.7 *Hierarchical Extension*

We generalize the solution (6.25) to the case of an arbitrary number of tasks N, for the case of *sufficiently constrained* system:

$$\begin{aligned} \tau^* =& (N_c S^T)^+ N_c (M\ddot{q}^* + h) \\ \ddot{q}^* =& \ddot{q}_1 + N_{p(0)} \ddot{q}_0^* \\ \ddot{q}_i =& \ddot{q}_{i+1} + (J_i N_{p(i)})^+ (\ddot{x}_i^* - \dot{J}_i \dot{q} - J_i \ddot{q}_{i+1}) \qquad i \in [1, N] \quad (6.26) \\ N_{p(i)} =& N_{p(i+1)} - (J_{i+1} N_{p(i+1)})^+ J_{i+1} N_{p(i+1)} \\ \ddot{q}_{N+1} =& -J_c^+ \dot{J}_c \dot{q}, \qquad N_{p(N)} = N_c, \end{aligned}$$

In this form, kinematics and dynamics are completely decoupled: first we solve the multi-task prioritization at kinematic level computing $\ddot{q}^*$, then we compute the torques to generate the desired joint accelerations. This decoupling is only possible because we are assuming that the system is *sufficiently constrained*, hence we can neglect the system dynamics when handling the prioritization.

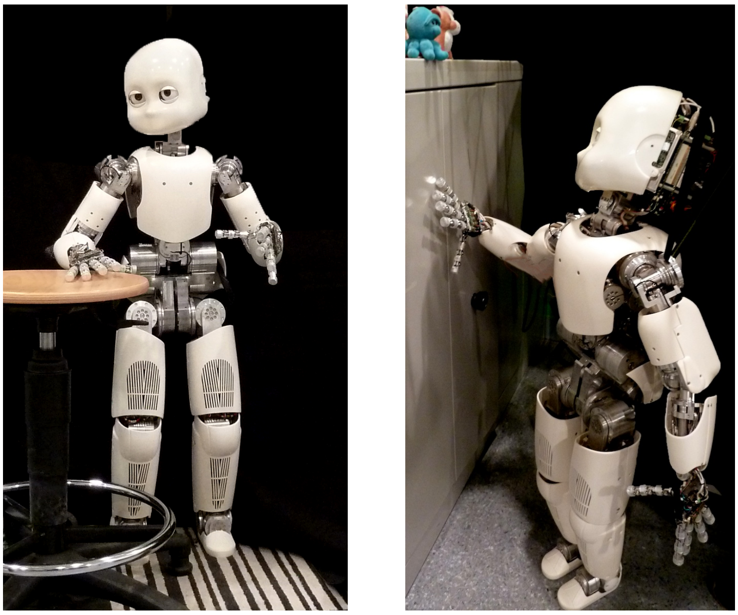
For sufficiently constrained robots, we can decouple kinematics and dynamics in the control problem.

6.3.8 *Force Control*

Even if we are considering systems that are in contact with the environment (i.e. constrained), so far we have dealt with position control only. The strategy of removing the constraint forces from the dynamics of the system proved very useful, because it allowed us to derive control laws that do not need contact force measurements. However, this approach is safe only as long as the contact geometry does not allow the robot to apply unbounded contact forces. For instance, if the robot is in contact with the ground only (typically with its feet), we know that the contact forces are limited by the weight of the robot, so we can safely decide not to control them. On the contrary, if the robot makes additional contacts with the environment (see Fig. 6.1 [1]), then it can apply much higher contact forces, which in general are only limited by the power of the motors. In these situations we need to control — besides the constraint consistent motion — the constraint forces too. In particular, we can distinguish two cases: either we want to control all the constraint forces, or we want to control just a subset of them. The next two subsections deal separately with these two cases.

In certain situations projecting the dynamics into the constraint nullspace does not work!

[1] Photos by Serena Ivaldi.



(a) (b)

Figure 6.1: The iCub robot standing while making contact with the environment.

6.3.8.1 *Complete Force Control*

In case we want to control all the constraint forces acting on the robot, we can formulate the control problem as:

$$
\begin{aligned}
\tau^* = \underset{\tau \in \mathbb{R}^n}{\operatorname{argmin}} \, & ||f - f^*||^2 \\
s.t. \quad & M\ddot{q} + h - J_c^T f = S^T \tau \\
& J_c \ddot{q} = -\dot{J}_c \dot{q}
\end{aligned}
$$

This problem has in general infinite solutions (see appendix B.8 for the complete derivation), which can be written in the convenient hybrid dynamics form of (6.7), with an extra term for the contact force:

$$
\begin{aligned}
\tau^* &= -(J_c\bar{S})^T f^* + N_j^{-1} \ddot{q}_j^* + \bar{S}^T h \\
\ddot{q}_j^* &= (J_c\bar{S})^+ (J_{cb} M_b^{-1} (h_b - J_{cb}^T f^*) - \dot{J}_c \dot{q}) + (I - (J_c\bar{S})^+ J_c\bar{S}) \ddot{q}_{j0},
\end{aligned}
$$

where $\ddot{q}_{j0} \in \mathbb{R}^n$ is an arbitrary joint acceleration vector. Similarly to the task-space PFL (6.11), the task Jacobian J_c is replaced by the *Generalized Jacobian* $J_c\bar{S}$, which takes into account the dynamic coupling between floating-base and actuated joints. The nullspace of the force control task can be used to perform secondary tasks; considering the general case of N tasks, in which the higher priority task regulates the constraint forces (i.e. $J_N = J_c$ and $\ddot{x}_N = 0$), we have:

$$
\begin{aligned}
\tau^* =& -(J_c\bar{S})^T f^* + N_j^{-1}\ddot{q}^* + \bar{S}^T h \\
\ddot{q}^* =& \ddot{q}_1 + N_{p(0)}\ddot{q}_0^* \\
\ddot{q}_i =& \ddot{q}_{i+1} + (J_i\bar{S}N_{p(i)})^+(\ddot{x}_i^* - \dot{J}_i\dot{q} + J_i(U^T M_b^{-1}(h_b - J_{cb}^T f) - \bar{S}\ddot{q}_{i+1})) \\
N_{p(i)} =& N_{p(i+1)} - (J_{i+1}\bar{S}N_{p(i+1)})^+ J_{i+1}\bar{S}N_{p(i+1)},
\end{aligned}
\tag{6.27}
$$

with $i \in [1, N]$. The computation is initialized setting $\ddot{q}_{N+1} = 0$ and $N_{p(N)} = I$. If $J_c\bar{S}$ is full rank, the constraint forces f are equal to the desired forces, hence we can set $f = f^*$. If this is not the case, we can either measure the constraint forces or compute them as:

$$
f = (J_c M^{-1} J_c^T)^{-1}(J_c M^{-1}(h - S^T\tau) - \dot{J}_c\dot{q})
$$

Unfortunately this expression requires the inverse of the inertia matrix, which is expensive to compute.

6.3.8.2 *Partial Force Control*

If an underactuated system is subject to more constraints than those necessary for guaranteeing the *sufficiently constrained* condition, then it may be desirable to control a subset of the constraint forces. For instance, consider a humanoid robot with both feet in rigid contact with the ground and one hand in rigid contact with a table (see Fig. 6.1). The 12 constraints due to the contact with the ground are definitely enough to make the system *sufficiently constrained*, so the 6 generalized forces due to the contact with the table can be controlled.

Let us introduce a new contact Jacobian $J_f \in \mathbb{R}^{k_f \times n+6}$, which is associated to the constraint forces that we wish to control (i.e. controlled constraints), whereas $J_c \in \mathbb{R}^{k \times n+6}$ is the Jacobian associated to the remaining constraints (i.e. supporting constraints). The control problem may be formulated as:

$$
\begin{aligned}
\tau^* = \underset{\tau \in \mathbb{R}^n}{\operatorname{argmin}} & \; ||f - f^*||^2 \\
s.t. \quad & M\ddot{q} + h - J_c^T f_c - J_f^T f = S^T\tau \\
& J_c\ddot{q} = -\dot{J}_c\dot{q} \\
& J_f\ddot{q} = -\dot{J}_f\dot{q}
\end{aligned}
$$

Appendix B.9 derives the solution of this problem; we report here only the final results. The problem has in general infinite solutions, given by:

$$
\begin{aligned}
\tau^* = & (\hat{J}\hat{M}^{-1}V_2^T S^T)_W^+((J_f J_c^+ \dot{J}_c - \dot{J}_f)\dot{q} + \hat{J}\hat{M}^{-1}\hat{h} - \hat{J}\hat{M}^{-1}\hat{J}^T f^*) \\
& + (I - (\hat{J}\hat{M}^{-1}V_2^T S^T)_W^+ \hat{J}\hat{M}^{-1}V_2^T S^T)\tau_0,
\end{aligned}
\tag{6.28}
$$

where V_2 is a matrix whose columns are an orthogonal base for the nullspace of J_c and, to simplify the notation, we defined the new variables $\hat{M}$, $\hat{h}$ and $\hat{J}$ as:

$$\begin{aligned}\hat{M} &= V_2^T M V_2 \\ \hat{h} &= V_2^T h - V_2^T M J_c^+ \dot{J}_c \dot{q} \\ \hat{J} &= J_f V_2\end{aligned}$$

This solution is complex and computationally inefficient but, under the assumption that the controlled system is *sufficiently constrained*, we can simplify it to:

$$\tau^* = (V_2^T S^T)^+ V_2^T (-J_f^T f^* + M(-J_{cf}^+ \dot{J}_{cf} \dot{q} + N_{cf} \ddot{q}_0) + h),$$

where $J_{cf}^T = \begin{bmatrix} J_c^T & J_f^T \end{bmatrix}$ is the Jacobian that considers both the supporting constraints and the controlled constrained. Differently from (6.28), this solution does not require the computation of M.

We can extend this control law to the general case in which the robot has to perform N tasks. The partial force control task has highest priority, because it is a physical constraint and hence it cannot be violated by definition. The control torques can be computed as:

$$\begin{aligned}\tau^* =&(V_2^T S^T)^+ V_2^T (M\ddot{q}^* + h - J_f^T f^*) \\ \ddot{q}^* =&\ddot{q}_1 + N_{p(0)} \ddot{q}_0^* \\ \ddot{q}_i =&\ddot{q}_{i+1} + (J_i N_{p(i)})^+ (\ddot{x}_i^* - \dot{J}_i \dot{q} - J_i \ddot{q}_{i+1}) \qquad \forall i \in [1, N] \\ N_{p(i)} =&N_{p(i+1)} - (J_{i+1} N_{p(i+1)})^+ J_{i+1} N_{p(i+1)},\end{aligned} \tag{6.29}$$

where $J_N = J_{cf}$, $N_{p(N)} = I$, $\ddot{x}_N^* = 0$ and $\ddot{q}_{N+1} = 0$. Kinematics and dynamics are decoupled, so τ^* can be efficiently computed with the Recursive Newton-Euler Algorithm. Soft force control tasks (see section 5.5.3.1) can be controlled as well, treating them as position control tasks, in which we set the reference acceleration as:

$$\ddot{x}_i^* = k_i^{-1} \ddot{f}_i^*,$$

where k_i is the i-th contact stiffness and $\ddot{f}_i^*$ is the i-th reference force acceleration.

6.3.9 *Summary*

To summarize the theoretical results of this chapter, we report here the basic control laws that we derived. We restrict this list to the control laws for *sufficiently constrained* floating-base robots, because our interest lies in humanoid robots, which, in most cases, are sufficiently constrained.

- Joint space position control of sufficiently constrained robot:

$$\tau^* = (V_2^T S^T)^+ V_2^T (M\ddot{q}^* + h)$$

- Task space position control of sufficiently constrained robot:

$$\tau^* = (V_2^T S^T)^+ V_2^T (M\ddot{q}^* + h)$$
$$\ddot{q}^* = -J_c^+ \dot{J}_c \dot{q} + (JV_2)^+ (\ddot{x}^* - \dot{J}\dot{q} + JJ_c^+ \dot{J}_c \dot{q}) + (I - (JV_2)^+ JV_2)\ddot{q}_0$$

- Partial force control of sufficiently constrained robot:

$$\tau^* = (V_2^T S^T)^+ V_2^T (M(-J_{cf}^+ \dot{J}_{cf} \dot{q} + N_{cf}\ddot{q}_0) + h - J_f^T f^*)$$

Note that in these control laws the matrix V_2, whose columns span the nullspace of the constraint Jacobian J_c, can be replaced by the nullspace projector N_c.

6.4 TESTS

This section presents some tests to validate the control laws for constrained floating-base robots that we derived in this chapter. These tests mainly focus on the control of a robot with both feet lying flat on the ground. The contacts between feet and ground result in 12 independent constraints, so the system is *sufficiently constrained*. This allowed us to exploit the simplified control laws for prioritized position control (6.26) and partial force control (6.29). We did not use in the tests the law for complete force control (6.27), because we think it is more suited to the control of space robots than humanoid robots.

Tests 1 and 2 focus on the control of the center of mass and its projection on the ground, so as to move the robot while balancing. In test 3 the robot made contact on a wall with its right hand, and it controlled the contact interaction employing the partial force control framework (6.29). Test 4 investigates how to solve the discontinuity due to transitions in the number of contacts, by exploiting partial force control.

6.4.1 *Test Details*

In every test, for each task we report a plot depicting three trajectories:

- x_d: desired trajectory specified by the user and processed by a trajectory generator to produce the reference trajectory;
- x_r: reference trajectory produced by a trajectory generator along with the reference velocity and acceleration trajectories (i.e. $\dot{x}_r, \ddot{x}_r$);
- x: real trajectory measured during the test.

To generate reference position/velocity/acceleration trajectories we used the approach proposed by Pattacini et al. [2010], which provides approximately minimum jerk trajectories. The trajectory generator is implemented as a 3rd order dynamical system, which takes as input the desired trajectory $x_d(t)$ and outputs the three position/velocity/acceleration reference trajectories $x_r(t)$, $\dot{x}_r(t)$, $\ddot{x}_r(t)$. The reference position trajectory follows the desired position trajectory with a velocity that depends on the parameter “trajectory time”. A

PID control law computes the commanded accelerations $\ddot{x}^*$, starting from the reference positions, velocities and accelerations:

$$\ddot{x}^* = \ddot{x}_r + K_d(\dot{x}_r - \dot{x}) + K_p(x_r - x),$$

where $K_d, K_p > 0$, $K_d, K_p \in \mathbb{R}^{m \times m}$ are diagonal matrices acting as derivative and proportional gains, respectively. We express any force and position with respect to the world reference frame, which is located on the ground, between the feet of the robot (see Fig. 6.2). From the standpoint of the robot, the x axis (red) points forward, the y axis (green) points leftward, and the z axis (blue) points upward.

The controllers used damped pseudoinverses [Chiaverini 1997] to ensure stability near singularities. The damping factor was always set to $5 \cdot 10^{-3}$. To avoid interferences between tasks, the nullspace projection matrices were computed without any damping.

We tested the presented control framework on a customized version of the Compliant huManoid (CoMan) simulator [Dallali et al. 2013] (the same used in chapter 5). The robot has 23 DoFs: 4 in each arm, 3 in the torso and 6 in each leg. To compute direct and inverse dynamics we used C functions, generated with the Robotran symbolic engine webpage [2012]. Rigid contact forces were simulated using linear spring-damper models, with the stiffness and damping values proposed by Dallali et al. [2013], that are, respectively, $2 \cdot 10^5 N/m$ and $10^3 Ns/m$. The *Simulink* variable step integrator *ode23t* integrated the equations of motion of the robot, with relative and absolute tolerance of 10^{-3} and 10^{-6}, respectively.

6.4.2 *Test 1: Static Balance*

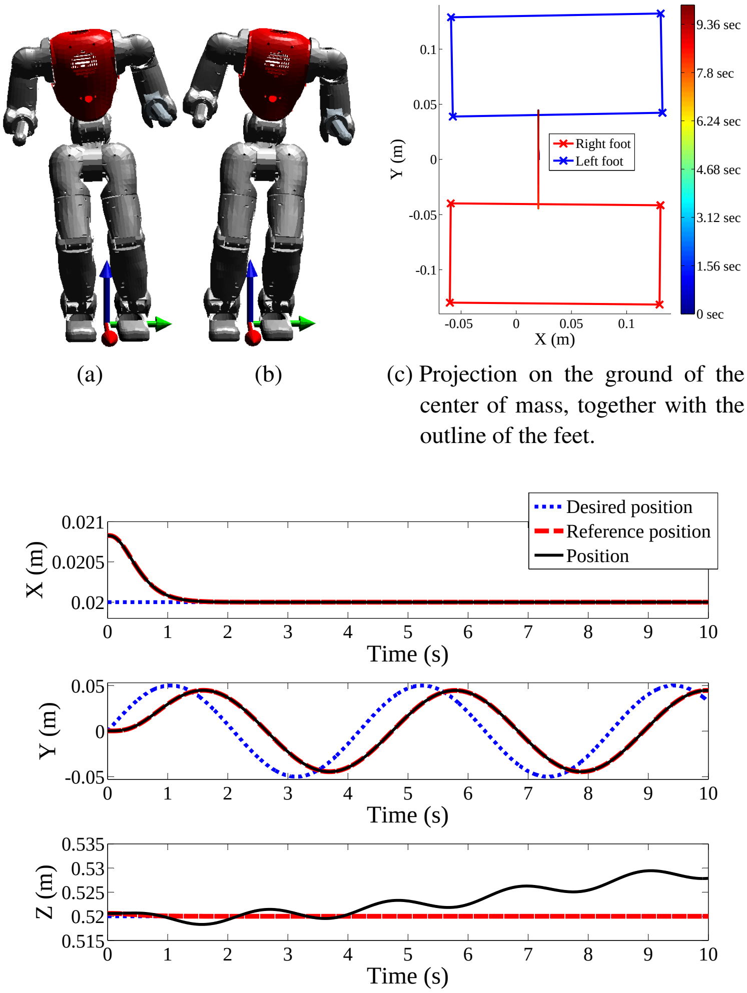
In this test we controlled a (sufficiently) constrained floating-base robot in task space, using the control laws derived in section 6.3.6. To control a floating-base robot, first of all we need to control its center of mass (COM), because it is at the basis of any balance strategy. The COM position $x_{COM} \in \mathbb{R}^3$ is defined as:

$$x_{COM} = \frac{1}{M} \sum_{i=1}^{N_B} m_i x_{C_i},$$

where M is the total mass of the robot, N_B is the number of links of the robot, m_i is the mass of link i, and $x_{C_i} \in \mathbb{R}^3$ is the position of the center of mass of link i. To control the COM we need to define its Jacobian $J_{COM} = \frac{\partial x_{COM}}{\partial q} \in \mathbb{R}^{3 \times n+6}$:

$$J_{COM} = \frac{1}{M} \sum_{i=1}^{N_B} m_i J_{C_i},$$

where $J_{C_i} = \frac{\partial x_{C_i}}{\partial q} \in \mathbb{R}^{3 \times n+6}$ is the Jacobian of the center of mass of link i. This first test controlled the projection of the COM on the ground (2 DoFs),



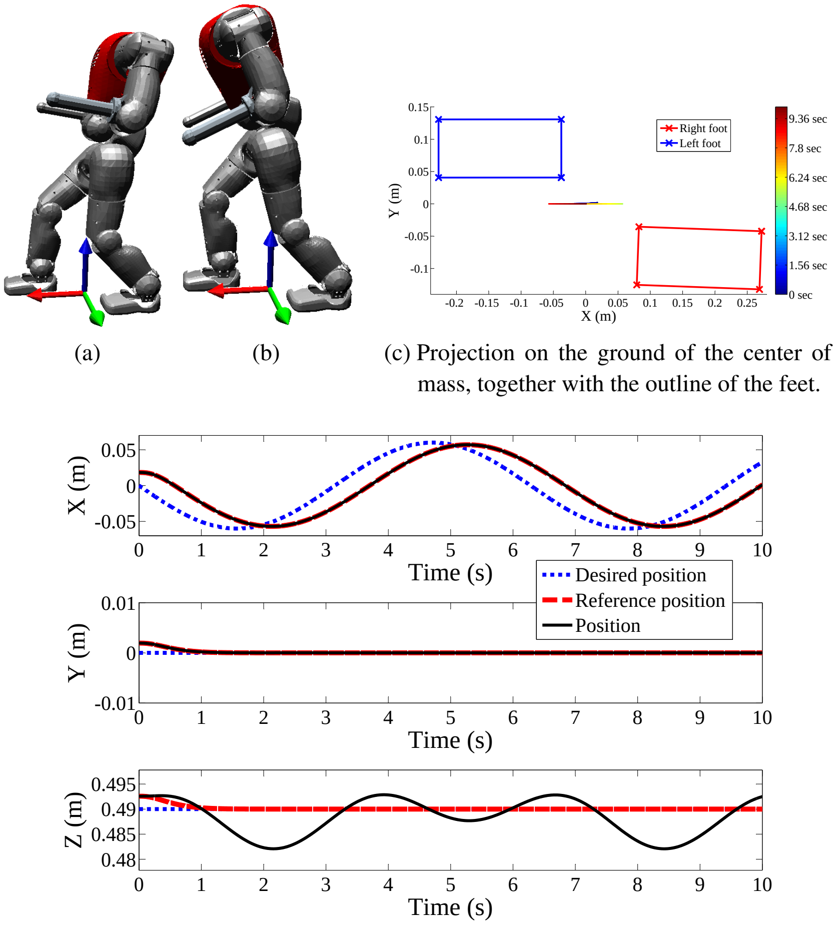
(a)

(b)

(c) Projection on the ground of the center of mass, together with the outline of the feet.

(d) Tracking of the center of mass.

Figure 6.2: Test 1: side swing. The robot controls the projection of its center of mass on the ground, moving it left and right. Note that the altitude of the COM is not controlled.



(a)

(b)

(c) Projection on the ground of the center of mass, together with the outline of the feet.

(d) Tracking of the center of mass.

Figure 6.3: Test 1: back-forth swing. The robot controlled the projection of its center of mass on the ground, moving it back and forth. The altitude of the COM was not controlled.

keeping it inside the support polygon defined by its feet. A secondary postural task stabilized the robot in joint space, attracting it towards the initial joint configuration. Overall, the robot performed two tasks, listed here with decreasing priority:

COM: 2 DoF (xy plane), control the projection of the COM on the ground;

POSTURE: 29 DoF, maintain the initial joint posture.

This test was divided into two subtests. In the first subtest the feet of the robot lied on the ground, one next to each other, and we moved the desired COM position sideways, i.e. along the y direction (see Fig. 6.2). In the second subtest the right foot was placed about 30 cm in front of the left foot, and we moved the desired COM position back and forth, i.e. along the x direction (see Fig. 6.3). In both subtests the tracking of the reference position was almost perfect (the RMSE was about 0.03 mm).

6.4.3 *Test 2: Squat*

In this test we controlled all the three components of the COM position. As in the previous test, the robot was (sufficiently) constrained and we performed a task space control, using the control laws derived in section 6.3.6. We gave higher priority to the control of the x and y COM position, because they are critical for the balance of the robot. The control of the COM position along the z direction was then a secondary task. Overall, the robot performed three tasks, listed here with decreasing priority:

COM: 2 DoF (xy plane), control the projection of the COM on the ground;

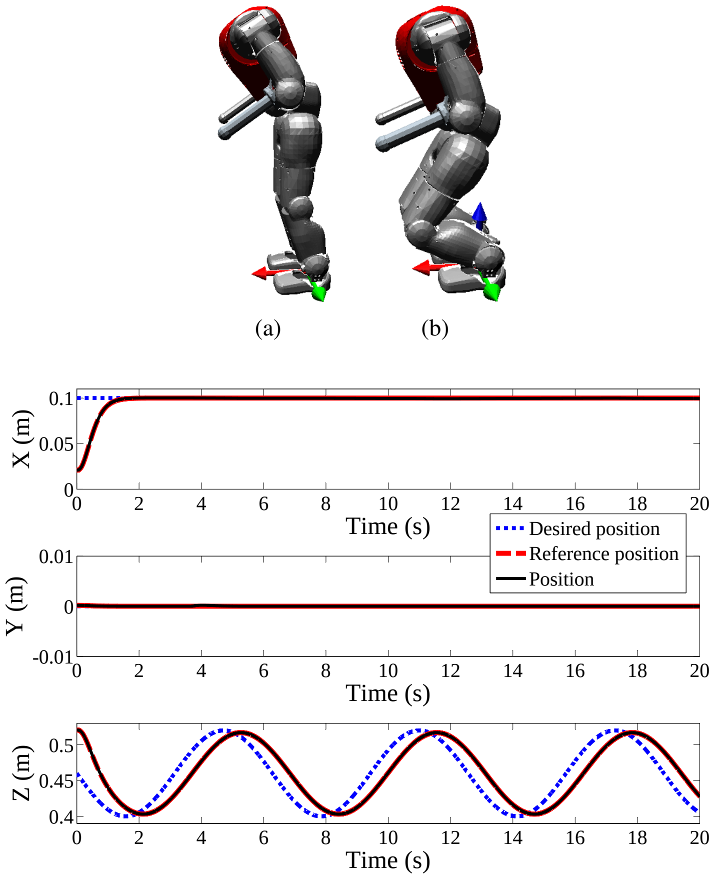
COM: 1 DoF (z direction), control the altitude of the COM;

POSTURE: 29 DoF, maintain the initial joint posture.

We maintained the desired COM position constant along the x and y directions; along the z direction the desired COM position was a sinusoid of amplitude 12 cm and period of about 6 seconds. The resulting motion was an "up and down" squat. Fig. 6.4 shows that the tracking of the reference COM position was almost perfect (the RMSE was 0.03 mm for the higher priority task, and 0.34 mm for the lower priority task).

6.4.4 *Test 3: Partial Force Control*

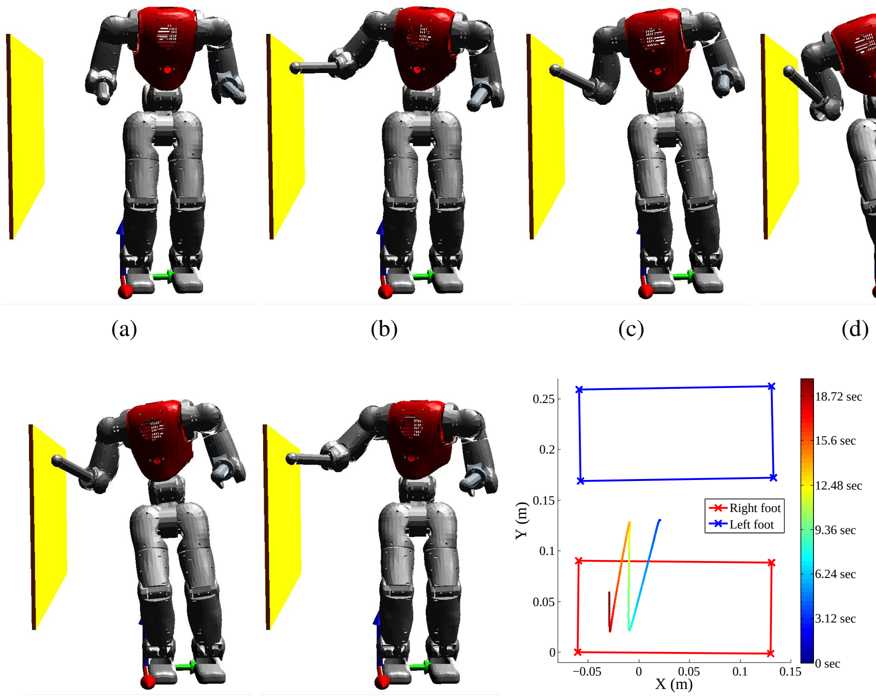
In this test we controlled the robot using the partial force control law introduced in section 6.3.8.2. The term "partial" refers to the fact that we controlled only a subset of the forces acting on the robot. In particular, in this test the robot made contact with a rigid wall using its right hand, and it regulated the contact force to the specified value of 20 N. On the contrary, the contact forces at the feet were not controlled. After making contact, we shifted the desired position of the COM along the y direction, so that the robot leaned



(a) (b)

(c) Tracking of the center of mass.

Figure 6.4: Test 2: squat. The robot controls its center of mass, moving it up and down.



(a)

(b)

(c)

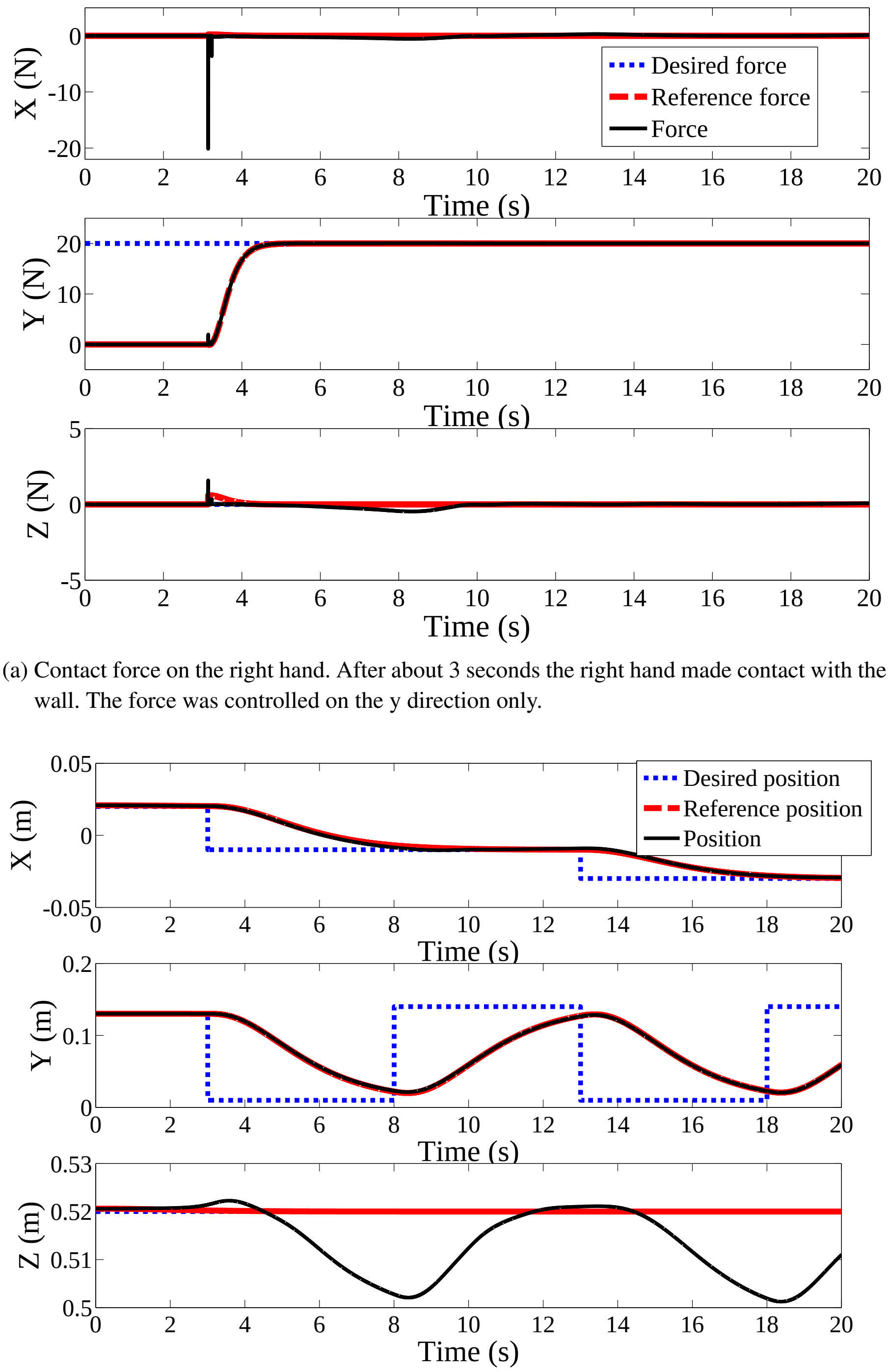
(d)

(e)

(f)

(g) Projection on the ground of the center of mass, together with the outline of the feet.

Figure 6.5: Test 3: partial force control. First the robot made contact on the yellow wall with its right hand; then it moved its COM towards the wall; finally it moved its COM back towards the middle of its feet.



(a) Contact force on the right hand. After about 3 seconds the right hand made contact with the wall. The force was controlled on the y direction only.

(b) Position of the center of mass. Only the x and y directions were controlled.

Figure 6.6: Test 3: force control. The robot controlled both the contact force on its right hand and the projection of its center of mass on the ground, moving it along the y axis.

against the wall, exploiting the additional support provided by the contact on its hand. Overall, the robot performed three tasks, listed here with decreasing priority:

FORCE: 1 DoF (y direction), control the contact force exerted with the right hand on the wall;

COM: 2 DoF (xy plane), control the projection of the COM on the ground;

POSTURE: 29 DoF, maintain the initial joint posture.

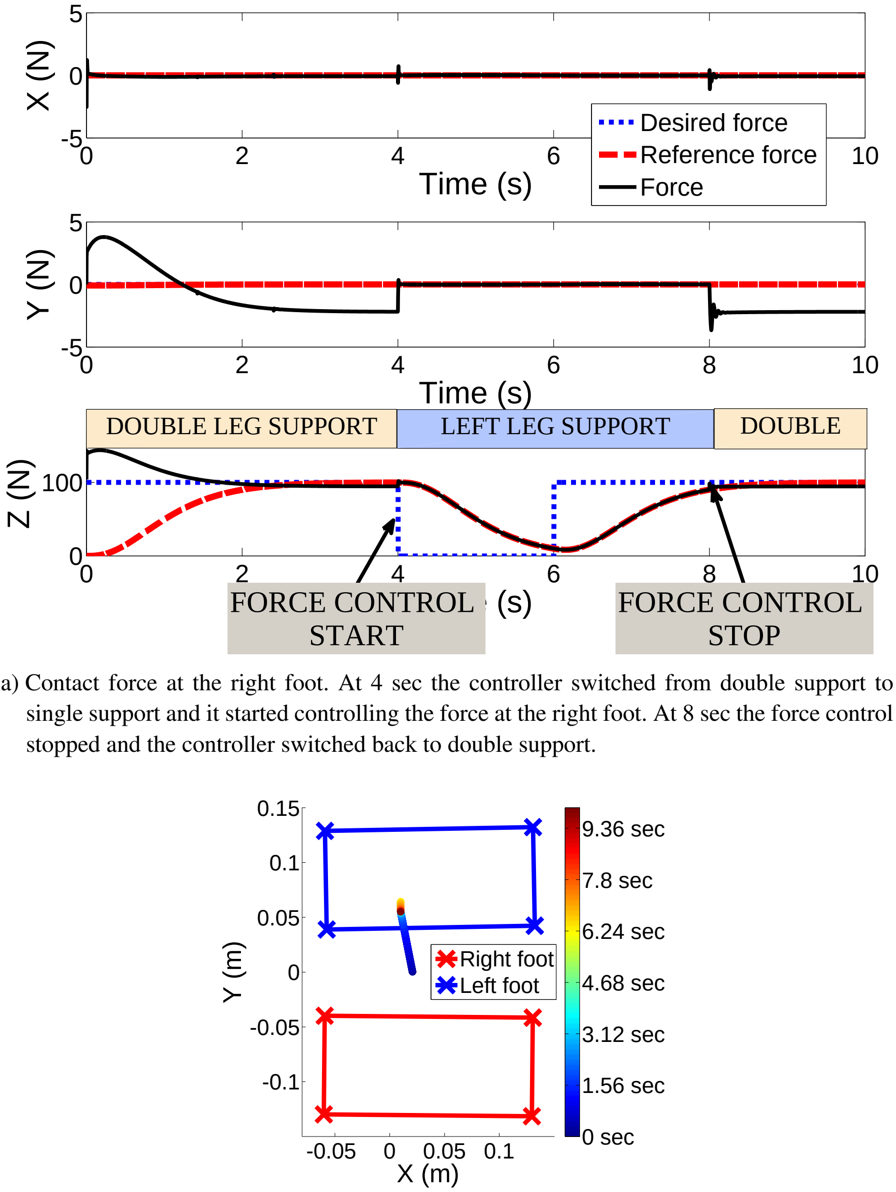
The robot managed to successfully control both the force exerted on the wall and the position of its COM: the RMSE for the force task was about 0.01 N, while for the COM task it was about 0.6 mm.

6.4.5 *Test 4: Switching Supporting Leg*

In this test we tackle the problem of switching from double support to single support. When walking, the contact constraints change discontinuously: the robot alternates between double support phases, having both feet in contact with the ground, and single support phases, having only one foot in contact with the ground. These discontinuities in the contact constraints cause discontinuities in the control action, which in turn may result in jerky motion, or even instability. In this test we show how the partial force control law introduced in section 6.3.8.2 can eliminate the discontinuities due to changes in the contact constraints. The strategy is simple: when switching from double to single support, we want to avoid abrupt changes in the contact forces, so, we actively control the constraint force associated to the constraint that we just removed, to generate a smooth transition of that force to zero. In particular, in this test we switched from double support to single support on the left leg. After moving the COM on top of the left foot, we switch to single support, removing the constraints generated by the contact between the right foot and the ground. This would normally result in a discontinuity in the contact forces at the right foot. To avoid that, we activated a (partial) force control, which regulated the contact force at the right foot, starting from its value at the time of the switch (i.e. about 100 N) towards zero. At this point the foot was ready to be moved. However, we did not move it, because we wanted to switch back to double support. This time we increased the contact force at the right foot to 100 N, and then we switched the controller back to double support, reintroducing the constraints at the right foot. This ensured a smooth transition between the different constraint situations (see the absence of discontinuities in the contact force in Fig. 6.7). Overall, the robot performed three tasks, listed here with decreasing priority:

FORCE: 3 DoF, control the contact force exerted with the right foot on the ground;

COM: 2 DoF (xy plane), control the projection of the COM on the ground;



(a) Contact force at the right foot. At 4 sec the controller switched from double support to single support and it started controlling the force at the right foot. At 8 sec the force control stopped and the controller switched back to double support.

(b) Projection on the ground of the center of mass, together with the outline of the feet.

Figure 6.7: Test 4: switching supporting leg. First the robot moved its COM on the left foot; then, right after switching to single support, it regulated the force between the right foot and the ground to zero; finally, it increased the same contact force to 100N, to prepare for the switch to double support.

POSTURE: 29 DoF, maintain the initial joint posture.

The robot managed to successfully control both the force exerted on the ground and the position of its COM: the RMSE for the force task was about 0.01 N, while for the COM task it was about 3 mm.

6.5 CONCLUSIONS AND FUTURE WORK

This chapter extended the “Task Space Inverse Dynamics” control framework to the control of floating-base systems, such as humanoid robots. First, we treated the case in which the robot is not in contact with the environment (i.e. unconstrained), typical of space robotics. For this scenario, we reviewed the “partial feedback linearization” technique, in its two well-known versions: collocated (to control the active joints) and non-collocated (to control the passive joints). We also reviewed the “task-space partial feedback linearization”, which is a generalization of “partial feedback linearization” to the control in task space.

Second, we focused on robots in contact (i.e. constrained), starting with a review of the “constraint nullspace projection” technique, which allows to express the dynamics of the systems independently of the constraint forces. Using this technique we derived position control laws that do not require measurements of contact forces. This is an important point, because most robotic platforms can not measure contact forces on their whole body. Moreover, force measurements are typically noisy and need to be filtered, hence they introduce disturbances and delays in the control action. We analyzed the dynamics of constrained floating-base systems, showing that, when these systems are *sufficiently constrained*, the constrained motion is completely attainable. This has strong implications, because it allows us to neglect the underactuation constraints in the derivation of their control laws (i.e. to treat them as fully-actuated systems), simplifying the resulting analytical expressions. We derived position control laws for constrained floating-base robots, both in joint space and in task space, showing how these expressions simplify in case of *sufficiently constrained* robots.

Finally, we tackled force control, dealing first with the control of all the constraint forces, and then with the control of a subset of the constraint forces. Also in this case we show how the control laws may benefit from the *sufficiently constrained* condition.

Simulation tests with a 23-DoF humanoid robot validated the presented control framework, proving its effectiveness in balancing the robot on its feet while performing other position/force tasks. We also showed how to ensure smooth transitions in the contact state of the robot, which naturally arise in walking and running.

6.5.1 *Limitations and Extensions*

This section briefly discusses the limitations of the presented control framework, looking at its future extensions. *Task Space Inverse Dynamics* is a control framework for position and force control of constrained floating-base robots. Under many aspects, it is similar to other control frameworks, such as De Lasa and Hertzmann [2009]; Jeong [2009]; Mistry and Righetti [2011]; Peters et al. [2007]; Saab et al. [2011a]; Sentis and Khatib [2005], and it shares with them most of its limitations.

The main assumption when dealing with constrained robots is the definition of the constraints: we always assumed no motion along the constrained directions. Considering the typical case of a rigid contact constraint, along the normal to the surface, motion is actually constrained in only one direction: for instance a foot can not penetrate the ground, but it can move upward. Along the tangent plane instead, friction provides bilateral constraints, but only as long as the applied forces stay within the cone of friction. Since the control laws that we derived in this chapter are based on the assumption of no motion along the constrained directions, they could try to generate constraint forces that are not physically feasible. Righetti et al. [2011b] tried to solve this issue by using the constraint redundancy (in the typical case in which there are more than 6 constraints) to minimize a quadratic cost in the constraint forces. This cost can be designed to minimize the tangential forces and/or the moments around the feet, but still it can not guarantee the physical consistency of the constraint forces. A safer approach is to include inequality constraints into the control problem [Righetti and Schaal 2012; Saab et al. 2011a,b], so that it allows for unilateral contact constraints and (an approximation of) friction cone constraints. The resolution of the control problem requires then a quadratic programming (QP) solver, which is computationally more expensive than a solution based on pseudoinverses; however, the authors claim that the computation time of the controller is still less than 1 ms, and so perfectly suitable for fast torque control loop.

Another limitation of the presented control framework is that it does not consider joint limits and motor torque limits. These limits can be easily included in the control problem as inequality constraints, using then a QP solver to compute the control torques.

A much more complex problem is due to the need for planning: these control frameworks guarantee instantaneous (local) optimality, but they do not reason globally. For instance, it may happen that a lower priority task leads the robot into a configuration in which a higher priority task becomes singular, hence unfeasible. To tackle this issue the cost function should consider not only the instantaneous acceleration/force error, but it should take into account the error over a certain time horizon. When the time horizon is finite, this approach takes the name of model predictive control [Manchester et al. 2011; Tassa et al. 2012]. It would be interesting to extend the presented framework to the case of finite horizon optimal control, but many issues needs to be considered, such as prioritization and computational cost of the solution.

7 CONCLUSIONS AND FUTURE WORK

This thesis dealt with the control of contact forces on a humanoid robot, using tactile sensors and force/torque sensors. This work belongs in the field of active force control, that is it did not resort to passive compliant elements (i.e. springs). The first part of the thesis treated calibration and estimation issues, which were preparatory for the force control framework presented in the second part.

7.1 SUMMARY

Chapter 2 described a new method for the spatial calibration of a network of tactile sensors, namely the estimation of the 3D position of each tactile sensor. The method is based on the measurement of external forces applied on the sensorized part of the robot's body. A similar technique had been previously used to estimate contact positions from force/torque measurements [Bicchi et al. 1993b], but it required a mathematical description of the robot surface. Our calibration method, instead, can work either with or without the description of the surface where the tactile sensors are located. We validated the presented technique by calibrating all the tactile sensors on the arms of the iCub robot (about 1500 sensors), with a precision of about 7 mm.

Chapter 3 leveraged the calibrated tactile sensors, together with the distributed force/torque sensors, to estimate external contact forces. We have implemented the presented method as part of the open source C++ library *iDyn* [Fumagalli et al. 2012], and we have tested it on the iCub robot. This technique relies on the Recursive Newton-Euler Algorithm [Siciliano and Khatib 2008] to estimate locations and magnitudes of an arbitrary number of contacts occurring on any part of the robot.

Chapter 2 and 3 laid the foundations for the implementation of whole-body force control on the iCub robot. Chapter 4 investigated the role of tactile sensors in explicit force control: we have carried out an analytical analysis on the effect that errors in contact localization have on contact forces. This gave us the necessary tools to quantify the expected uncertainties in contact forces that are due to uncertainties in the spatial calibration of the tactile sensors. Besides the theoretical analysis — which has been validated in simulation — we have implemented explicit force control and parallel control on the iCub robot, and we have empirically evaluated the benefits of precise contact localizations. With explicit force control, introducing of an error of about 9 cm in the contact localization, we measured an error in the contact force of about 50% of the commanded value. Using parallel control, with the same error in contact localization, the RMSE has increased by 20% for the force tracking, and by 35% for the position tracking.

Chapters 5 and 6 introduced a new control framework, called “Task Space Inverse Dynamics”, which allows for different control paradigms, such as explicit force control and parallel control — already implemented on the iCub robot in chapter 4. Even though similar control frameworks already existed, our work was motivated by the lack of a framework that addressed all these aspects: optimality of the solution, capability to control interaction forces, and computational cost of the algorithm. In chapter 5 we have derived “Task Space Inverse Dynamics” as solution of a point-wise optimal control problem, taking inspiration from the Unifying Framework [Peters et al. 2007]. Tests in simulation have proved the advantages of our control framework with respect to other two state-of-the-arm equivalent frameworks (i.e. Unifying Framework [Peters et al. 2007] and Whole-Body Control Framework [Sentis and Khatib 2005]). Remarkably, with “Task Space Inverse Dynamics” the control loop has taken less than half the time than with the Whole-Body Control Framework [Sentis and Khatib 2005], while it has still guaranteed the optimality of the solution. Finally, in chapter 6, we have extended “Task Space Inverse Dynamics” to the control of floating-base systems, such as humanoid robots. Tests in simulation validated the extended framework, proving its effectiveness in balancing the robot, while performing position/force control tasks. We also exploited the presented control framework to ensure smooth transitions in the contact state, such as the switch from double support to single support, typical of walking and running behaviors.

7.2 DISCUSSION AND FUTURE WORK

This section points out the potential impact of this thesis for the scientific community, stressing the possible applications of the presented methods and control strategies. The work presented in this thesis advanced the current state of the art in the field of active force control of rigid robots. Hopefully, our efforts represent another step towards a new generation of force-controlled robots that can safely and effectively interact with humans and the environment.

The methods for the calibration of tactile sensors (chapter 2) and the estimation of contact forces (chapter 3) apply to any robot that is equipped with distributed force and tactile sensors. Both types of sensors are becoming increasingly common in robotics, and this boosts the significance of the related works, such as this thesis. We have implemented our method for the estimation of contact forces in an open-source C++ library, *iDyn*, which is freely available online for anyone. Moreover, we plan to generalize the method, and the related library implementation, providing a useful software tool to the robotic community. While the current version of the library exploits only tactile sensors and 6-axis force/torque sensors, the new version will include also accelerometers, inertial measurement units and joint torque sensors. Also, the estimated quantities will extend beyond contact forces, including joint velocities, joint accelerations and the pose of the floating-base of the robot.

The control framework “Task Space Inverse Dynamics” (chapters 5 and 6) is the principal contribution of this thesis. It can be exploited for the position and force control of any rigid robot, including fixed-base manipulators and floating-base systems, such as humanoid robots. At this time, we have not provided an open-source C++ implementation of the framework, but we plan to do it during the next year. Anyway, the algorithm relies only on pseudoinverses and inverse dynamics of the robot, which can be computed using the open-source C++ libraries Eigen [Gael Guennebaud et al. 2010] and KDL [Smits 2007], respectively. Moreover, one of the assets of the framework is the simplicity of the algorithm, which would make its implementation straightforward, as well as efficient.

Recalling what we said in section 1.3.1, this work is based on a few assumptions, which can help us to understand its current limitations and future directions. When treating the control problem, we have always assumed to have a reference trajectory to track, being either a position or a force trajectory, but we have not investigated how we can generate this trajectory. In our tests we have used an approximately minimum-jerk trajectory generator, which provided position-velocity-acceleration reference trajectories. However, this simple approach fails when the desired final position is outside the robot’s workspace; in this case the robot typically reaches singular configurations, with the unpleasant consequence that even higher priority tasks may become unfeasible. Since the task prioritization works only locally, i.e. instantaneously, lower priority tasks may still globally conflict with higher priority tasks, leading the robot in configurations in which those tasks become unfeasible. To overcome these issues we should move from local optimality to global optimality, introducing then *planning* in the process.

Another important aspect that we overlooked is uncertainty in the kinematic/dynamic model of the robot: adaptive or robust control techniques could surely help to improve the performance. Along this line of thoughts, we plan to integrate our control framework with an online estimator of the inertial parameters of the robot, generating an adaptive control architecture.

Part III

APPENDIX

A

LINEAR ALGEBRA REVIEW

This appendix presents a brief review on linear algebra, with particular focus on pseudoinverses.

A.1 SINGULAR VALUE DECOMPOSITION (SVD)

We start by reviewing the singular value decomposition, which is fundamental for the work presented in this thesis. Given a matrix $A \in \mathbb{R}^{m \times n}$ with rank r, its singular value decomposition is:

$$A = U\Sigma V^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 S V_1^T,$$

where U and V are orthogonal matrices, whereas Σ and S are diagonal matrices. In particular we have that:

$$\begin{aligned} &UU^T = U^T U = I_m, U \in \mathbb{R}^{m \times m} \\ &VV^T = V^T V = I_n, V \in \mathbb{R}^{n \times n} \\ &U_1^T U_1 = I_r, U_1 \in \mathbb{R}^{m \times r} \\ &V_1^T V_1 = I_r, V_1 \in \mathbb{R}^{n \times r} \\ &U_2^T U_2 = I_{m-r}, U_2 \in \mathbb{R}^{m \times m-r} \\ &V_2^T V_2 = I_{n-r}, V_2 \in \mathbb{R}^{n \times n-r} \\ &\Sigma \in \mathbb{R}^{m \times n} \\ &S \in \mathbb{R}^{r \times r} \end{aligned}$$

The diagonal elements of S are called the *singular values* of A and they are ordered in decreasing order. Since the columns of U_1 (V_1) are orthogonal to all the columns of U_2 (V_2) we have that:

$$\begin{aligned} &U_1^T U_2 = 0_{m \times m-r} \\ &V_1^T V_2 = 0_{n \times n-r} \end{aligned}$$

In case A is symmetric (i.e. $A = A^T$), its singular value decomposition takes the form:

$$A = U\Sigma U^T$$

A.1.1 *Nullspace and Range*

The *nullspace* of a matrix $A \in \mathbb{R}^{m \times n}$ is the subspace of vectors that are mapped to zero when premultiplied by A, and it is defined as:

$$N(A) = \{x \in \mathbb{R}^n \,|\, Ax = 0\} \subseteq \mathbb{R}^n$$

We can also view it as the set of vectors that are orthogonal to all the rows of A. The *range* of a matrix $A \in \mathbb{R}^{m \times n}$ is the subspace of vectors that can be generated by multiplying A times an arbitrary vector, and it is defined as:

$$R(A) = \{Ax \,|\, x \in \mathbb{R}^n\} \subseteq \mathbb{R}^m$$

It can also be viewed as the span of the columns of A. As far as we know, there is no special name for the subspace of vectors that are not mapped to zero when premultiplied by A, so we decided to call it the *active space*, which is defined as:

$$\{x \in \mathbb{R}^n \,|\, Ax \neq 0\} \subseteq \mathbb{R}^n$$

The singular value decomposition of a matrix gives an orthonormal basis of its nullspace, range and active space. In particular, the columns of V_2 are an orthonormal basis of the nullspace of A. The columns of V_1 are an orthonormal basis of the active space of A. The columns of U_1 are an orthonormal basis of the range of A.

A.2 PSEUDOINVERSES

The pseudoinverse of a matrix $A \in \mathbb{R}^{m \times n}$ is a matrix $X \in \mathbb{R}^{n \times m}$ that satisfies the following equalities:

$$AXA = A \qquad XAX = X \qquad (AX)^T = AX \qquad (XA)^T = XA$$

The most widely known pseudoinverse is the Moore-Penrose pseudoinverse, which is indicated as A^+. The term pseudoinverse, without further specification, often refers to the Moore-Penrose pseudoinverse. The term *generalized inverse* is sometimes used as a synonym for pseudoinverse. A common use of the Moore-Penrose pseudoinverse (hereafter, just pseudoinverse) is to compute a 'best fit' (least squares) solution to a system of linear equations that lacks a unique solution. Another use is to find the minimum (Euclidean) norm solution to a system of linear equations with multiple solutions.

A.2.1 *Computation of Pseudoinverses*

If A is skinny and full rank, then A^+ can be computed as:

$$A^+ = (A^T A)^{-1} A^T$$

If A is fat and full rank, then A^+ can be computed as:

$$A^+ = A^T (AA^T)^{-1}$$

In general, pseudoinverses can be computed using the SVD. Given a matrix $A = U_1 S V_1^T$, its pseudoinverse can be computed as:

$$A^+ = V_1 S^{-1} U_1^T$$

Given that S is diagonal, this expression is extremely simple to compute.

A.2.2 *Linear systems*

Pseudoinverses are commonly used to find (approximate) solutions for overdetermined or underdetermined systems of linear equations. Consider the linear system:

$$Ax = y$$

where $A \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$. The system may be determined, overdetermined, underdetermined. If $m = n$ and A is full rank, then the system is determined, meaning that it has a unique solution:

$$x = A^{-1}y$$

If A is skinny (i.e. $m > n$) then the system is overdetermined (more equations than unknowns), meaning that for most y, there is no exact solution. In this case, a common approach is to approximately solve the system, finding the value that minimizes the squared error. If A is full rank there is a unique value of x that minimizes the squared error:

$$x^* = \operatorname*{argmin}_{x \in \mathbb{R}^n} ||Ax - y||^2 = A^+ y$$

If A is fat (i.e. $m < n$) and full rank, then the system is underdetermined (less equations than unknowns), meaning that there exists infinite solutions. Among the infinite solutions we can select the one with minimum distance from an arbitrary value $x_0 \in \mathbb{R}^n$:

$$x^* = \{\operatorname*{argmin}_{x \in \mathbb{R}^n} ||x - x_0||^2 \quad s.t. \quad Ax = y\} = A^+ y + (I - A^+ A)x_0$$

A common approach is to set $x_0 = 0$ to find the solution with minimum norm.

In the general case (A not full rank) the pseudoinverse gives the minimum-norm, least-squares approximate solution, namely the solution of this problem:

$$\begin{aligned} x^* = \operatorname*{argmin}_{x \in \mathbb{R}^n} &||x - x_0||^2 \\ &s.t. \quad x \in \{z : ||Az - y||^2 = \min_{x \in \mathbb{R}^n} ||Ax - y||^2\} \end{aligned}$$

A.3 WEIGHTED PSEUDOINVERSES

The weighted pseudoinverse of a matrix $A \in \mathbb{R}^{m \times n}$ is a matrix $X \in \mathbb{R}^{n \times m}$ that satisfies the following equalities:

$$AXA = A \qquad\qquad XAX = X$$

but it may not satisfy these equalities:

$$(AX)^T = AX \qquad\qquad (XA)^T = XA$$

The pseudoinverse of a skinny matrix A weighted by a positive-definite matrix $E \in \mathbb{R}^{m \times m}$ is defined as:

$$A_E^+ = (E^{\frac{1}{2}}A)^+ E^{\frac{1}{2}} = (A^T E A)^{-1} A^T E \tag{A.1}$$

where the last expression is valid only if A is full rank. The pseudoinverse of a fat matrix A weighted by a positive-definite matrix $W \in \mathbb{R}^{n \times n}$ is defined as:

$$A_W^+ = W^{\frac{1}{2}}(AW^{\frac{1}{2}})^+ = WA^T(AWA^T)^{-1} \tag{A.2}$$

where the last expression is valid only if A is full rank.

In case of overdetermined linear systems, weighted pseudoinverses find the value that minimizes the weighted squared error, that is:

$$x^* = \underset{x \in \mathbb{R}^n}{\text{argmin}} \, ||E^{\frac{1}{2}}(Ax - y)||^2 = A_E^+ y$$

In case of underdetermined linear systems, weighted pseudoinverses find the solution that minimizes the weighted distance from an arbitrary value x_0, that is:

$$x^* = \{\underset{x \in \mathbb{R}^n}{\text{argmin}} \, ||W^{-\frac{1}{2}}(x - x_0)||^2 \quad s.t. \quad Ax = y\} = A_W^+ y + (I - A_W^+ A)x_0$$

In case A is rank deficient, we can weigh both the squared error (with the matrix E) and the distance from an arbitrary x_0 (with the matrix W):

$$\begin{aligned} x^* = \underset{x \in \mathbb{R}^n}{\text{argmin}} & ||W^{-\frac{1}{2}}(x - x_0)||^2 \\ & s.t. \quad x \in \{z : ||E^{\frac{1}{2}}(Az - y)||^2 = \min_{x \in \mathbb{R}^n} ||E^{\frac{1}{2}}(Ax - y)||^2\} \end{aligned}$$

The solution is given by

$$x^* = A_{W,E}^+ y + (I - A_{W,E}^+ E^{\frac{1}{2}} A W^{\frac{1}{2}})x_0,$$

where:

$$A_{W,E}^+ = W^{\frac{1}{2}}(E^{\frac{1}{2}} A W^{\frac{1}{2}})^+ E^{\frac{1}{2}} \tag{A.3}$$

The notation for the two versions of weighted pseudoinverses is the same, but the reader can infer which one applies from the context: for skinny full-rank matrices we use (A.1), whereas for fat full-rank matrices we use (A.2). In case of rank deficient matrices both weighted pseudoinverses could apply, but the dimension of the weight matrix clarifies if we are using (A.1) or (A.2). The only ambiguous case is the weighted pseudoinverse of a square rank-deficient matrix, in which case we have to clarify which equation we are using. However, to ease the understanding of the equations, we try always to indicate the weight matrix with E when we use (A.1), and with W (or V) when we use (A.2).

A.3.1 *Weighted Nullspace Projections*

The nullspace projection matrix of a (fat or rank deficient) matrix A is given by:

$$N_A = I - A^+A = I - A^T(AA^T)^{-1}A, \tag{A.4}$$

where the last expression is valid only if A is full rank. On the other hand the weighted nullspace projection matrix of a matrix A is given by:

$$N_{A_W} = (I - A^+_W A) = (I - W^{\frac{1}{2}}(AW^{\frac{1}{2}})^+ A) = (I - WA^T(AWA^T)^{-1}A), \tag{A.5}$$

where the last expression is valid only if A is full rank. Note that the non-weighted nullspace projection matrix is symmetric, i.e. $N_A = N_A^T$, whereas its weighted version is not symmetric. The weighted nullspace projection may be also expressed as:

$$N_{A_W} = W^{\frac{1}{2}}(I - (AW^{\frac{1}{2}})^+ AW^{\frac{1}{2}})W^{-\frac{1}{2}},$$

which can be seen as a sequence of three operations:

1. a projection into the weighted space through $W^{-\frac{1}{2}}$;
2. an orthogonal projection onto the nullspace of $AW^{\frac{1}{2}}$;
3. a projection back into the non-weighted space through $W^{\frac{1}{2}}$.

The second step is an orthogonal projection performed by the operator:

$$I - (AW)^+ AW = I - WV_1(V_1^T W^2 V_1)^{-1} V_1^T W$$

Being an orthogonal projection, we know that it can be expressed as $I - \bar{V}_1 \bar{V}_1^T$, where $\bar{V}_1^T \bar{V}_1 = I$ and the columns of $\bar{V}_1$ span the active space (i.e. the orthogonal complement of the nullspace) of the matrix $AW^{\frac{1}{2}}$. We can easily infer the relationship between the orthogonal nullspace projection of A (spanned by the columns of V_1) and the orthogonal nullspace projection of $AW^{\frac{1}{2}}$ (spanned by the columns of $\bar{V}_1$):

$$\bar{V}_1 = WV_1(V_1^T W^2 V_1)^{-\frac{1}{2}}$$

Using the SVD, we can also write the weighted nullspace projection matrix as:

$$N_{A_W} = (I - A^+_W A) = I - (V_1^T)^+_W V_1^T$$

The matrix N_{A_W} is not symmetric, but the relationship with its transpose is quite simple:

$$\begin{aligned} N_{A_W}^T &= (I - A^T A_W^{+T}) = (I - A^T(W^{\frac{1}{2}}A^T)^+ W^{\frac{1}{2}}) = (I - A^T(AWA^T)^{-1}AW) = \\ &= W^{-1}(I - WA^T(AWA^T)^{-1}A)W = W^{-1}N_{A_W}W \end{aligned} \tag{A.6}$$

Note that $N_{A_W}^T$ is no longer a nullspace projector of A.

B

TASK SPACE INVERSE DYNAMICS

This appendix collects the analytical derivation of some control laws that compose the control framework presented in this thesis, i.e. "Task Space Inverse Dynamics".

B.1 RIGID FORCE CONTROL OF FIXED-BASE ROBOTS

This section deals with the problem of controlling rigid contact forces in fixed-base robots. Assume that the motion of the manipulator is subject to a set of k nonlinear constraints:

$$c(q,t) = 0$$

Differentiating twice the constraints we get:

$$\begin{aligned} \dot{c} &= J_c\dot{q} = 0 \\ \ddot{c} &= J_c\ddot{q} + \dot{J}_c\dot{q} = 0 \end{aligned}$$

where $J_c(q) = \frac{\partial}{\partial q}c \in \mathbb{R}^{k\times n}$ is the constraint Jacobian. The desired contact forces $f^* \in \mathbb{R}^k$ can be computed using a proportional-derivative control law:

$$f^* = f_r + K_p(f_r - f) + K_d(\dot{f}_r - \dot{f}),$$

where $f_r(t) \in \mathbb{R}^k$ is the reference force trajectory, while $K_d > 0, K_p > 0$ are matrices acting as derivative and proportional gains, respectively. The control problem is defined as:

$$\begin{aligned} \tau^* = &\operatorname*{argmin}_{\tau\in\mathbb{R}^n} ||f - f^*||^2 \\ s.t. \qquad &M\ddot{q} + h - J_c^T f = \tau \\ &J_c\ddot{q} + \dot{J}_c\dot{q} = 0 \end{aligned}$$

This case is very different from the "spring contact" case because the motion of the manipulator is constrained by the rigid contact. Multiplying the first constraint times J_cM^{-1} and substituting $J_c\ddot{q}$ from the second constraint, we can compute the contact forces as a function of τ:

$$f = (J_cM^{-1}J_c^T)^{-1}(J_cM^{-1}(h-\tau) - \dot{J}_c\dot{q}), \tag{B.1}$$

where we can recognize the operational space inertia matrix $\Lambda_c = (J_cM^{-1}J_c^T)^{-1}$. Substituting (B.1) into the cost function we can calculate the solutions of the problem:

$$\begin{aligned} \tau^* = &(J_cM^{-1})_V^+(-\Lambda_c^{-1}f^* - \dot{J}_c\dot{q} + J_cM^{-1}h) \\ &+ (I - (J_cM^{-1})_V^+ J_cM^{-1})\tau_0 \end{aligned}$$

Setting $V = M^2 W$ we get a simpler solution:

$$\tau^* = MJ_{cW}^{+}(-\Lambda_c^{-1} f^* - \dot{J}_c \dot{q} + J_c M^{-1} h) + MN_{cW} M^{-1} \tau_0 \tag{B.2}$$

Note that this control law does not contain the contact forces f. We see the similarity with the position control law (5.6), in which $\ddot{x}^*$ is replaced with $-\Lambda_c^{-1} f^*$. If we choose $W = M^{-1}$ then (B.2) simplifies to:

$$\tau^* = -J_c^T f^* + M\bar{J}_c(-\dot{J}_c \dot{q} + J_c M^{-1} h) + M\bar{N}_c M^{-1} \tau_0, \tag{B.3}$$

where $\bar{N}_c = I - \bar{J}_c J_c$ and $\bar{J}_c = M^{-1} J_c^T \Lambda_c$ is the dynamically consistent Jacobian pseudoinverse. Another convenient option is to choose $W = I$:

$$\tau^* = MJ_c^{+}(-\Lambda_c^{-1} f^* - \dot{J}_c \dot{q} + J_c M^{-1} h) + MN_c M^{-1} \tau_0 \tag{B.4}$$

Unfortunately both (B.3) and (B.4) are computationally expensive because they contain the terms $\bar{J}_c$ and Λ_c^{-1}, respectively. However, a simple trick may help us to get a more convenient solution. We rewrite (B.2) as:

$$\tau^* = -M(J_{cW_1}^{+} \Lambda_c^{-1} f^* + J_{cW_2}^{+}(\dot{J}_c \dot{q} - J_c M^{-1} h) - N_{cW_2} M^{-1} \tau_0)$$

Then we set $W_1 = M$ and $W_2 = I$ and we get:

$$\tau^* = -J_c^T f^* - MJ_c^{+}(\dot{J}_c \dot{q} - J_c M^{-1} h) + MN_c M^{-1} \tau_0$$

We can check that this is still a solution of the force control problem by substituting it into (B.1). Finally, the term M^{-1} can be removed by adding the postural task, i.e. setting $\tau_0 = M\ddot{q}_0^* + h$:

$$\tau^* = -J_c^T f^* + M(-J_c^{+} \dot{J}_c \dot{q} + N_c \ddot{q}_0^*) + h$$

B.2 INERTIA MATRIX INVERSE

This decomposition of the inverse of the inertia matrix of a floating-base robot is going to play an important role in the derivation of many control laws:

$$M^{-1} = \bar{S} N_j \bar{S}^T + U^T M_b^{-1} U \tag{B.5}$$

To better understand this equality, we report here the expression of the inverse of the inertia matrix M^{-1} as a block matrix:

$$\begin{aligned}
M^{-1} &= \begin{bmatrix} N_b & N_{bj} \\ N_{bj}^T & N_j \end{bmatrix} \\
N_b &= (M_b - M_{bj} M_j^{-1} M_{bj}^T)^{-1} \\
N_j &= (M_j - M_{bj}^T M_b^{-1} M_{bj})^{-1} \\
N_{bj} &= -N_b M_{bj} M_j^{-1} = -M_b^{-1} M_{bj} N_j
\end{aligned} \tag{B.6}$$

We also report the pseudoinverses of S, weighted with M^{-1}:

$$\bar{S} = S_{M^{-1}}^{+} = M^{-1} S^T (S M^{-1} S^T)^{-1} = \begin{bmatrix} N_{bj} \\ N_j \end{bmatrix} N_j^{-1} = \begin{bmatrix} -M_b^{-1} M_{bj} \\ I_n \end{bmatrix}$$

Now we can prove (B.5):

$$
\begin{aligned}
&\bar{S} N_j \bar{S}^T + U^T M_b^{-1} U = \\
&= \begin{bmatrix} -M_b^{-1} M_{bj} \\ I \end{bmatrix} N_j \begin{bmatrix} -M_{bj}^T M_b^{-1} & I \end{bmatrix} + \begin{bmatrix} M_b^{-1} & 0 \\ 0 & 0 \end{bmatrix} = \\
&= \begin{bmatrix} M_b^{-1} M_{bj} N_j M_{bj}^T M_b^{-1} & -M_b^{-1} M_{bj} N_j \\ -N_j M_{bj}^T M_b^{-1} & N_j \end{bmatrix} + \begin{bmatrix} M_b^{-1} & 0 \\ 0 & 0 \end{bmatrix} = \\
&= \begin{bmatrix} M_b^{-1} M_{bj} N_j M_{bj}^T M_b^{-1} + M_b^{-1} & N_{bj} \\ N_{bj}^T & N_j \end{bmatrix} = \\
&= \begin{bmatrix} N_b & N_{bj} \\ N_{bj}^T & N_j \end{bmatrix} = M^{-1},
\end{aligned}
$$

where we used the fact that:

$$
\begin{aligned}
&M_b^{-1} M_{bj} N_j M_{bj}^T M_b^{-1} + M_b^{-1} = \\
&= N_b M_{bj} M_j^{-1} M_{bj}^T M_b^{-1} + M_b^{-1} = \\
&= N_b (M_{bj} M_j^{-1} M_{bj}^T M_b^{-1} + N_b^{-1} M_b^{-1}) = \\
&= N_b (M_{bj} M_j^{-1} M_{bj}^T M_b^{-1} + (M_b - M_{bj} M_j^{-1} M_{bj}^T) M_b^{-1}) = N_b
\end{aligned}
$$

B.3 COLLOCATED PARTIAL FEEDBACK LINEARIZATION

This section deals with the control of the actuated joints of a floating-base robot. The resulting control law is commonly known as *Collocated Partial Feedback Linearization* [Spong 1998]. We want to solve the control problem:

$$
\begin{aligned}
\tau^* = &\operatorname*{argmin}_{\tau \in \mathbb{R}^n} ||\ddot{q}_j - \ddot{q}_j^*||^2 \\
s.t. \quad & M\ddot{q} + h = S^T \tau
\end{aligned}
$$

where $\ddot{q}_j^* \in \mathbb{R}^n$ are the desired active joint accelerations. If we multiply the constraint times $\bar{S}^T$ we get:

$$
\begin{aligned}
\bar{S}^T (M\ddot{q} + h) &= \bar{S}^T S^T \tau \\
N_j^{-1} \ddot{q}_j + \bar{S}^T h &= \tau \\
\ddot{q}_j &= N_j (\tau - \bar{S}^T h) \\
\ddot{q}_j &= (M_j - M_{bj}^T M_b^{-1} M_{bj})^{-1} (\tau - h_j + M_{bj}^T M_b^{-1} h_b)
\end{aligned}
\tag{B.7}
$$

This equation describes the dynamics of the active joints independently of the passive joints. We can now substitute $\ddot{q}_j$ in the cost function to get an unconstrained problem:

$$
\tau^* = \operatorname*{argmin}_{\tau \in \mathbb{R}^n} ||N_j(\tau - \bar{S}^T h) - \ddot{q}_j^*||^2
$$

This problem has a unique solution:

$$\tau^* = N_j^{-1}\ddot{q}_j^* + \bar{S}^T h \tag{B.8}$$

Applying the control law (B.8) to a floating-base system, the resulting motion is:

$$\begin{aligned}\ddot{q}_j &= \ddot{q}_j^* \\ \ddot{x}_b &= -M_b^{-1}(M_{bj}\ddot{q}_j^* + h_b)\end{aligned}$$

B.4 NON-COLLOCATED PARTIAL FEEDBACK LINEARIZATION

This section deals with the control of the passive joints (i.e. the floating-base virtual joints) of a floating-base robot. The resulting control law is commonly known as *Non-Collocated Partial Feedback Linearization* [Spong 1998]. We want to solve the following problem:

$$\begin{aligned}\tau^* &= \underset{\tau \in \mathbb{R}^n}{\operatorname{argmin}} \, ||\ddot{x}_b - \ddot{x}_b^*||_E^2 \\ s.t. \quad & M\ddot{q} + h = S^T\tau\end{aligned}$$

where $\ddot{x}_b^* \in \mathbb{R}^6$ is the desired base acceleration. Multiplying the constraint times $\bar{U}^T$ we get:

$$N_b^{-1}\ddot{x}_b + \bar{U}^T h = -M_{bj}M_j^{-1}\tau$$

From this expression we compute $\ddot{x}_b$ and we substitute it in the cost function, to get an unconstrained problem:

$$\tau^* = \underset{\tau \in \mathbb{R}^n}{\operatorname{argmin}} \, ||N_b(-M_{bj}M_j^{-1}\tau - \bar{U}^T h) - \ddot{x}_b^*||_E^2$$

Depending on the rank of M_{bj} this problem may have one or infinite solutions. In general, the solution takes this form:

$$\tau^* = -(N_b M_{bj} M_j^{-1})^+_{W,E}(\ddot{x}_b^* + N_b\bar{U}^T h) + (I - (N_b M_{bj} M_j^{-1})^+_{W_E} N_b M_{bj} M_j^{-1})\tau_0 \tag{B.9}$$

In case of infinite solutions, this is the one that minimizes $||\tau - \tau_0||_W$. The condition of *strong inertial coupling* requires that $\text{rank}(M_{bj}) \geq 6$, where 6 is the number of passive DoFs for floating-base robots. If this condition is satisfied then we have complete control over the passive DoFs, that is we can get $\ddot{x}_b = \ddot{x}_b^*, \forall \ddot{x}_b^* \in \mathbb{R}^6$. Furthermore, $N_b M_{bj} M_j^{-1}$ is full-row rank and so the matrix E does not affect the solution. Setting $E = N_b^{-2}$ the general solution (B.9) simplifies to:

$$\tau^* = -(M_{bj}M_j^{-1})^+_W(N_b^{-1}\ddot{x}_b^* + \bar{U}^T h) + (I - (M_{bj}M_j^{-1})^+_W M_{bj}M_j^{-1})\tau_0 \tag{B.10}$$

We can get a further simplification by setting $W = M_j^2$:

$$\tau^* = -M_j M_{bj}^+(N_b^{-1}\ddot{x}_b^* + \bar{U}^T h) + M_j(I - M_{bj}^+ M_{bj})M_j^{-1}\tau_0 \tag{B.11}$$

Alternatively, this control law can be also put in the convenient hybrid dynamics form (B.8) by setting $W = N_j^{-2}$ and $\tau_0 = \bar{S}^T h + \tilde{\tau}_0$:

$$\begin{aligned}
\tau^* &= -N_j^{-1}(M_{bj} M_j^{-1} N_j^{-1})^+ (N_b^{-1} \ddot{x}_b^* + \bar{U}^T h) \\
&\qquad + (I - N_j^{-1}(M_{bj} M_j^{-1} N_j^{-1})^+ M_{bj} M_j^{-1})(\bar{S}^T h + \tilde{\tau}_0) \\
&= -N_j^{-1} M_{bj}^+ M_b N_b (N_b^{-1} \ddot{x}_b^* + \bar{U}^T h) \\
&\qquad + (I - N_j^{-1} M_{bj}^+ M_b N_b N_b^{-1} M_b^{-1} M_{bj} N_j)(\bar{S}^T h + \tilde{\tau}_0) \\
&= -N_j^{-1} M_{bj}^+ (M_b \ddot{x}_b^* + M_b U M^{-1} h) + N_j^{-1}(I - M_{bj}^+ M_{bj}) N_j (\bar{S}^T h + \tilde{\tau}_0) \\
&= -N_j^{-1} M_{bj}^+ (M_b \ddot{x}_b^* - M_{bj} N_j \bar{S}^T h + h_b) + N_j^{-1}(I - M_{bj}^+ M_{bj}) N_j (\bar{S}^T h + \tilde{\tau}_0) \\
&= N_j^{-1} \underbrace{(-M_{bj}^+ (M_b \ddot{x}_b^* + h_b) + (I - M_{bj}^+ M_{bj}) N_j \tilde{\tau}_0)}_{\ddot{q}_j^*} + \bar{S}^T h,
\end{aligned} \tag{B.12}$$

where we used the fact that $M_{bj} M_j^{-1} N_j^{-1} = N_b^{-1} M_b^{-1} M_{bj}$, which can be derived directly from (B.5). Under the condition of *strong inertial coupling* both (B.11) and (B.12) result in $\ddot{x}_b = \ddot{x}_b^*$, but they use the task nullspace (if any) to minimize different quantities. In particular (B.11) minimizes $||M_j^{-1}(\tau - \tau_0)||$, whereas (B.12) minimizes $||N_j(\tau - \bar{S}^T h - \tilde{\tau}_0)||$.

B.5 TASK SPACE PARTIAL FEEDBACK LINEARIZATION

This section deals with the control in task space of a floating-base robot. The resulting control laws are an extension of partial feedback linearization to task space control [Shin and Lee 1997; Shkolnik and Tedrake 2008]. The control problem can be formulated as:

$$\begin{aligned}
\tau^* &= \operatorname*{argmin}_{\tau \in \mathbb{R}^n} ||J\ddot{q} + \dot{J}\dot{q} - \ddot{x}^*||^2 \\
s.t. \quad & M\ddot{q} + h = S^T \tau
\end{aligned}$$

where $\ddot{x}^* \in \mathbb{R}^m$ is the desired task space acceleration and $J = \frac{\partial x}{\partial q} \in \mathbb{R}^{m \times n}$ is the task Jacobian. We compute $\ddot{q}$ from the constraints and we substitute it in the cost function to get:

$$\tau^* = \operatorname*{argmin}_{\tau \in \mathbb{R}^n} ||JM^{-1}(S^T \tau - h) + \dot{J}\dot{q} - \ddot{x}^*||^2$$

This problem has infinite solutions (assuming $m < n$), which can be expressed as:

$$\tau^* = (JM^{-1}S^T)_V^+ (\ddot{x}^* - \dot{J}\dot{q} + JM^{-1}h) + (I - (JM^{-1}S^T)_V^+ JM^{-1}S^T)\tau_0 \tag{B.13}$$

where $V > 0$ is an arbitrary weight matrix and $\tau_0 \in \mathbb{R}^n$ is an arbitrary vector. This control law is computationally expensive because it requires the inverse of the mass matrix M^{-1}, but we can simplify it. Using the decomposition of the mass matrix (B.5) we can show that:

$$M^{-1}S^T = (\bar{S}N_j\bar{S}^T + U^T M_b^{-1}U)S^T = \bar{S}N_j$$

Substituting this expression in (B.13) and setting $V = N_j^{-2}W$, where $W > 0, W \in \mathbb{R}^{n\times n}$ is an arbitrary matrix, we get:

$$\begin{aligned}\tau^* &= (J\bar{S}N_j)^+_{N_j^{-2}W}(\ddot{x}^* - \dot{J}\dot{q} + JM^{-1}h) + (I - (J\bar{S}N_j)^+_{N_j^{-2}W}J\bar{S}N_j)\tau_0\\ &= N_j^{-1}(J\bar{S})^+_W(\ddot{x}^* - \dot{J}\dot{q} + JM^{-1}h) + (I - N_j^{-1}(J\bar{S})^+_W J\bar{S}N_j)\tau_0\\ &= N_j^{-1}(J\bar{S})^+_W(\ddot{x}^* - \dot{J}\dot{q} + JM^{-1}h) + N_j^{-1}(I - (J\bar{S})^+_W J\bar{S})N_j\tau_0\end{aligned} \tag{B.14}$$

Now we have to get rid of the M^{-1} premultiplying h. Again we use the equality (B.5) and we set $\tau_0 = \bar{S}^T h$:

$$\begin{aligned}\tau^* &= N_j^{-1}(J\bar{S})^+_W(\ddot{x}^* - \dot{J}\dot{q} + J(\bar{S}N_j\bar{S}^T + U^T M_b^{-1}U)h)\\ &\quad + N_j^{-1}(I - (J\bar{S})^+_W J\bar{S})N_j\bar{S}^T h\\ &= N_j^{-1}(J\bar{S})^+_W(\ddot{x}^* - \dot{J}\dot{q} + JU^T M_b^{-1}Uh) + \bar{S}^T h\\ &= N_j^{-1}\underbrace{(J\bar{S})^+_W(\ddot{x}^* - \dot{J}\dot{q} + J_b M_b^{-1}h_b)}_{\ddot{q}_j^*} + \bar{S}^T h\end{aligned} \tag{B.15}$$

In the end we managed to put the control law in the convenient hybrid dynamics form of (B.8). Differently from (B.13), this control law does not require the computation of the whole inertia matrix, but just the part associated to the floating-base, i.e. M_b and M_{bj}.

The collocated and non-collocated PFL can be derived as special cases of the task space PFL (B.15). In particular, if we set $J = S$ and $\ddot{x}^* = \ddot{q}_j^*$ we get the collocated PFL:

$$\tau^* = N_j^{-1}(S\bar{S})^+_W\ddot{q}_j^* + \bar{S}^T h = N_j^{-1}\ddot{q}_j^* + \bar{S}^T h$$

On the other hand, if we set $J = U$ and $\ddot{x}^* = \ddot{x}_b^*$ we get the non-collocated PFL:

$$\tau^* = N_j^{-1}(U\bar{S})^+_W(\ddot{x}_b^* + M_b^{-1}h_b) + \bar{S}^T h = N_j^{-1}(-M_b^{-1}M_{bj})^+_W(\ddot{x}_b^* + M_b^{-1}h_b) + \bar{S}^T h$$

If the condition of *strong inertial coupling* is met (i.e. $\text{rank}(M_{bj}) = 6$) then

$$(-M_b^{-1}M_{bj})^+_W = -M_{bj}{}^+_W M_b,$$

and so:

$$\tau^* = -N_j^{-1}M_{bj}{}^+_W(M_b\ddot{x}_b^* + h_b) + \bar{S}^T h$$

B.6 CONSTRAINT NULLSPACE PROJECTION

This section deals with the dynamics of constrained floating-base robots, explaining the ”constraint nullspace projection”. This technique is at the basis of the design of many control laws, because it allows to express the dynamics of this type of systems independently of the constraint forces. Consider the equation of motion of a constrained floating-base system:

$$M\ddot{q} + h - J_c^T f = S^T \tau, \tag{B.16}$$

where $f \in \mathbb{R}^k$ are the constraint forces. The system is subject to the constraints:

$$J_c \ddot{q} = -\dot{J}_c \dot{q} \tag{B.17}$$

To remove the contact forces from the system dynamics we have to: i) multiply (B.16) times $J_c M^{-1}$, ii) substitute $J_c\ddot{q} = -\dot{J}_c\dot{q}$, iii) compute f as a function of τ, iv) substitute f back in (B.16). Doing that we get an expression describing the dynamics of the constrained system, but without the constraint forces f.

$$\begin{aligned}
&J_c M^{-1}[M\ddot{q} + h - J_c^T f = S^T \tau] \\
&J_c M^{-1} J_c^T f = J_c\ddot{q} + J_c M^{-1}(h - S^T\tau) \\
&f = \Lambda_c(-\dot{J}_c\dot{q} + J_c M^{-1}(h - S^T\tau))
\end{aligned}$$

where $\Lambda_c = (J_c M^{-1} J_c^T)^{-1}$ is the constraint space inertia matrix. Now that we have computed f, we substitute it in (B.16):

$$\begin{aligned}
M\ddot{q} + h - J_c^T\Lambda_c(-\dot{J}_c\dot{q} + J_c M^{-1}(h - S^T\tau)) &= S^T\tau \\
M\ddot{q} + (I - J_c^T\Lambda_c J_c M^{-1})h + J_c^T\Lambda_c\dot{J}_c\dot{q} &= (I - J_c^T\Lambda_c J_c M^{-1})S^T\tau \\
M\ddot{q} + (I - \bar{J}_c J_c)^T h + J_c^T\Lambda_c\dot{J}_c\dot{q} &= (I - \bar{J}_c J_c)^T S^T\tau \\
M\ddot{q} + \bar{N}_c^T h + J_c^T\Lambda_c\dot{J}_c\dot{q} &= \bar{N}_c^T S^T\tau
\end{aligned} \tag{B.18}$$

where $\bar{J}_c = J_{cM^{-1}}^{+}$ is the *dynamically consistent* Jacobian pseudoinverse [Khatib 1987], and $\bar{N}_c = I - \bar{J}_c J_c$ is its nullspace projector.

A more recent approach [Aghili 2005; Righetti et al. 2011a] suggests to eliminate the constraint forces from (B.16) by projecting it in the nullspace of the constraints. This is done by multiplying the robot dynamics times $N_{cW}^T = (I - J_{cW}^{+} J_c)^T$, so that we get:

$$\begin{aligned}
&N_{cW}^T[M\ddot{q} + h - J_c^T f = S^T\tau] \\
&\quad N_{cW}^T(M\ddot{q} + h) = N_{cW}^T S^T\tau
\end{aligned} \tag{B.19}$$

If we set $W = M^{-1}$ and we substitute $\ddot{q}$ from (B.17), then we get (B.18). This equation is a convenient alternative to (B.18), because it does not use M^{-1} and it is overall simpler. Equation (B.19) can be derived using the same procedure that we used to derive (B.18), but multiplying (B.16) times $J_c W$ rather than $J_c M^{-1}$:

$$\begin{aligned}
&J_c W[M\ddot{q} + h - J_c^T f = S^T \tau] \\
&J_c W J_c^T f = J_c W(M\ddot{q} + h - S^T \tau) \\
&f = (J_c W J_c^T)^{-1} J_c W(M\ddot{q} + h - S^T \tau) \\
&f = J_{cW}^{T+}(M\ddot{q} + h - S^T \tau)
\end{aligned}$$

Then, as before, we substitute f back in (B.16):

$$\begin{aligned}
M\ddot{q} + h - J_c^T J_{cW}^{T+}(M\ddot{q} + h - S^T \tau) &= S^T \tau \\
(I - J_c^T J_{cW}^{T+})(M\ddot{q} + h) &= (I - J_c^T J_{cW}^{T+}) S^T \tau \qquad \text{(B.20)} \\
N_{cW}^T (M\ddot{q} + h) &= N_{cW}^T S^T \tau
\end{aligned}$$

To get the simplest formulation in the following we are going to use $W = I$, so we can also exploits the symmetry of the orthogonal projector: $N_c^T = N_c$.

B.7 JOINT CONTROL

This section deals with the control of constrained floating-base robots in configuration space. We want to solve this problem:

$$\begin{aligned}
\tau^* = \underset{\tau \in \mathbb{R}^n}{\text{argmin}} \, &||\ddot{q} - \ddot{q}^*||_E^2 \\
s.t. \quad &M\ddot{q} + h - J_c^T f = S^T \tau \\
&J_c \ddot{q} = -\dot{J}_c \dot{q}
\end{aligned}$$

where $\ddot{q}^* \in \mathbb{R}^{n+6}$ are the desired joint accelerations, which we assume to be constraint consistent, i.e. $J_c \ddot{q}^* + \dot{J}_c \dot{q} = 0$. Using (6.18) (with $W = I$) we can formulate an equivalent unconstrained problem:

$$\tau^* = \underset{\tau \in \mathbb{R}^n}{\text{argmin}} \, ||M_c^{-1}(N_c(S^T \tau - h) - J_c^+ \dot{J}_c \dot{q}) - \ddot{q}^*||_E^2$$

The general solution of this problem is:

$$\begin{aligned}
\tau^* &= (M_c^{-1} N_c S^T)_E^+ M_c^{-1}(M_c \ddot{q}^* + N_c h + J_c^+ \dot{J}_c \dot{q}) \\
&= (E M_c^{-1} N_c S^T)^+ E M_c^{-1}(N_c M \ddot{q}^* + J_c^+ J_c \ddot{q}^* + N_c h + J_c^+ \dot{J}_c \dot{q}) \\
&= (E M_c^{-1} N_c S^T)^+ E M_c^{-1}(N_c(M \ddot{q}^* + h) + J_c^+(J_c \ddot{q}^* + \dot{J}_c \dot{q})) \\
&= (E M_c^{-1} N_c S^T)^+ E M_c^{-1} N_c(M \ddot{q}^* + h)
\end{aligned}$$

If we want to minimize the norm of the joint acceleration error we have to set $E = I$. Doing that we get a control law that is computationally expensive, because it requires the computation of M (which appears inside M_c). However, if the system is *sufficiently constrained* (i.e. $V_2^T S^T$ is full-row rank) then τ^* will result in $\ddot{q} = \ddot{q}^*$. In this case the matrix E does not affect the solution, so we can set it to the most convenient value, that is $E = M_c$:

$$\tau^* = (N_c S^T)^+ N_c (M \ddot{q}^* + h) \qquad \text{(B.21)}$$

This control law is very efficient, because it can be computed with one iteration of RNEA. However, using (B.21) when the system is not *sufficiently constrained* means that we do not minimize the joint acceleration error, but a weighted version of it. Alternatively, rather than using N_c, we could use any matrix whose columns span the nullspace of J_c (e.g. the matrix V_2 from the SVD of J_c):

$$\tau^* = (V_2^T S^T)^+ V_2^T (M\ddot{q}^* + h)$$

This form has the advantage of computing the pseudoinverse of a smaller matrix: $V_2^T S^T \in \mathbb{R}^{n+6-k \times n}$, while $N_c S^T \in \mathbb{R}^{n+6 \times n}$.

B.8 COMPLETE RIGID FORCE CONTROL OF FLOATING-BASE ROBOTS

This section deals with the control of all the rigid contact forces acting on a constrained floating-base robot. In contrast, the next section deals with the case in which we want to control only a subset of these constraint forces. We can formulate the control problem as:

$$\begin{aligned} \tau^* = \underset{\tau \in \mathbb{R}^n}{\text{argmin}}\, & ||f - f^*||^2 \\ s.t. \quad & M\ddot{q} + h - J_c^T f = S^T \tau \\ & J_c \ddot{q} = -\dot{J}_c \dot{q} \end{aligned}$$

Solving the constraints we can get an equivalent unconstrained problem:

$$\tau^* = \underset{\tau \in \mathbb{R}^n}{\text{argmin}}\, ||(J_c M^{-1} J_c^T)^{-1}(J_c M^{-1}(h - S^T \tau) - \dot{J}_c \dot{q}) - f^*||^2$$

The solutions of this problem are:

$$\begin{aligned} \tau^* = & (J_c M^{-1} S^T)_W^+ (-J_c M^{-1} J_c^T f^* + J_c M^{-1} h - \dot{J}_c \dot{q}) \\ & + (I - (J_c M^{-1} S^T)_W^+ J_c M^{-1} S^T)\tau_0 \end{aligned}$$

Exploiting (B.5) we can simplify this expression so that it does not require the computation of M_j. First we elaborate the first part of the expression, leaving out the nullspace term:

$$\begin{aligned} & (J_c M^{-1} S^T)_W^+ (-J_c M^{-1} J_c^T f^* + J_c M^{-1} h - \dot{J}_c \dot{q}) = \\ & = (J_c \bar{S} N_j)_W^+ (-J_c \bar{S} N_j \bar{S}^T J_c^T f^* - J_c U^T M_b^{-1} U J_c^T f^* + J_c M^{-1} h - \dot{J}_c \dot{q}) = \\ & = N_j^{-1} (J_c \bar{S})_V^+ (-J_c \bar{S} N_j \bar{S}^T J_c^T f^* - J_c U^T M_b^{-1} U J_c^T f^* + J_c M^{-1} h - \dot{J}_c \dot{q}) = \\ & = -(J_c \bar{S})^T f^* + N_j^{-1} (J_c \bar{S})^+ (J_{cb} M_b^{-1} (h_b - J_{cb}^T f^*) - \dot{J}_c \dot{q} + J_c \bar{S} N_j \bar{S}^T h), \end{aligned}$$

where $J_{cb} = J_c U^T$ is the part of J_c associated to the base joints. In the first passage we set $W = N_j^{-2} V$, while in the second passage we set $V = N_j$ for the term multiplying f^*, and $V = I$ for the other terms. Following the same passages, the nullspace term becomes:

$$\begin{aligned} (I - (J_c M^{-1} S^T)_W^+ J_c M^{-1} S^T)\tau_0 & = (I - (J_c \bar{S} N_j)_W^+ J_c \bar{S} N_j)\tau_0 \\ & = (I - N_j^{-1} (J_c \bar{S})^+ J_c \bar{S} N_j)\tau_0 \\ & = N_j^{-1} (I - (J_c \bar{S})^+ J_c \bar{S}) N_j \tau_0 \end{aligned}$$

Putting it all together and setting $\tau_0 = \bar{S}^T h$:

$$\begin{aligned}\tau^* &= -(J_c\bar{S})^T f^* + N_j^{-1}(J_c\bar{S})^+(J_{cb}M_b^{-1}(h_b - J_{cb}^T f^*) - \dot{J}_c\dot{q} + J_c\bar{S}N_j\bar{S}^T h) \\ &\quad + N_j^{-1}(I - (J_c\bar{S})^+ J_c\bar{S})N_j\bar{S}^T h = \\ &= -(J_c\bar{S})^T f^* + N_j^{-1}\underbrace{(J_c\bar{S})^+(J_{cb}M_b^{-1}(h_b - J_{cb}^T f^*) - \dot{J}_c\dot{q})}_{\ddot{q}_j^*} + \bar{S}^T h\end{aligned}$$

Introducing the variable $\ddot{q}_j^*$ we clearly see the hybrid dynamics structure (B.8), with an extra term for the contact force:

$$\tau^* = -(J_c\bar{S})^T f^* + N_j^{-1}\ddot{q}_j^* + \bar{S}^T h$$

B.9 PARTIAL RIGID FORCE CONTROL OF FLOATING-BASE ROBOTS

This section deals with the control of a subset of the rigid contact forces acting on a constrained floating-base robot. Let us introduce a new Jacobian $J_f \in \mathbb{R}^{k_f \times n+6}$, which is associated to the constraint forces that we wish to control, whereas $J_c \in \mathbb{R}^{k \times n+6}$ is the Jacobian associated to the remaining constraints. The control problem may be formulated as:

$$\begin{aligned}\tau^* = \underset{\tau\in\mathbb{R}^n}{\operatorname{argmin}}\, & ||f - f^*||^2 \\ s.t. \quad & M\ddot{q} + h - J_c^T f_c - J_f^T f = S^T\tau \\ & J_c\ddot{q} = -\dot{J}_c\dot{q} \\ & J_f\ddot{q} = -\dot{J}_f\dot{q}\end{aligned}$$

From the second constraint we compute:

$$\ddot{q} = -J_c^+\dot{J}_c\dot{q} + V_2\ddot{q}_c,$$

where $V_2 \in \mathbb{R}^{n+6-k \times n+6}$ is a matrix whose columns are an orthogonal base for the nullspace of J_c. We substitute $\ddot{q}$ in the third constraint:

$$J_f V_2\ddot{q}_c = (J_f J_c^+ \dot{J}_c - \dot{J}_f)\dot{q} \tag{B.22}$$

We can remove the constraint forces from the system's dynamics multiplying both sides of the equation times V_2^T:

$$\underbrace{V_2^T M V_2}_{\hat{M}}\ddot{q}_c + \underbrace{V_2^T h - V_2^T M J_c^+ \dot{J}_c\dot{q}}_{\hat{h}} - \underbrace{V_2^T J_f^T}_{\hat{J}^T} f = V_2^T S^T\tau$$

where we defined the new variables $\hat{M}$, $\hat{h}$ and $\hat{J}$ to simplify the notation. Then we multiply both sides times $\hat{J}\hat{M}^{-1}$ to get:

$$\hat{J}\ddot{q}_c + \hat{J}\hat{M}^{-1}(\hat{h} - \hat{J}^T f) = \hat{J}\hat{M}^{-1}V_2^T S^T\tau$$

We substitute (B.22):

$$(J_f J_c^+\dot{J}_c - \dot{J}_f)\dot{q} + \hat{J}\hat{M}^{-1}(\hat{h} - \hat{J}^T f) = \hat{J}\hat{M}^{-1}V_2^T S^T\tau$$

Now, assuming that $\hat{J} = J_f V_2$ is full-row rank, we can compute f as:

$$f = (\hat{J}\hat{M}^{-1}\hat{J}^T)^{-1}((J_f J_c^+ \dot{J}_c - \dot{J}_f)\dot{q} + \hat{J}\hat{M}^{-1}(\hat{h} - V_2^T S^T \tau))$$

Substituting this expression of f in the cost function of the original control problem we get an equivalent unconstrained problem:

$$\tau^* = \underset{\tau \in \mathbb{R}^n}{\operatorname{argmin}} ||(\hat{J}\hat{M}^{-1}\hat{J}^T)^{-1}((J_f J_c^+ \dot{J}_c - \dot{J}_f)\dot{q} + \hat{J}\hat{M}^{-1}(\hat{h} - V_2^T S^T \tau)) - f^*||^2$$

This problem has in general infinite solutions:

$$\begin{aligned}\tau^* = {} & (\hat{J}\hat{M}^{-1}V_2^T S^T)^+_W((J_f J_c^+ \dot{J}_c - \dot{J}_f)\dot{q} + \hat{J}\hat{M}^{-1}\hat{h} - \hat{J}\hat{M}^{-1}\hat{J}^T f^*) \\ & + (I - (\hat{J}\hat{M}^{-1}V_2^T S^T)^+_W \hat{J}\hat{M}^{-1}V_2^T S^T)\tau_0\end{aligned} \tag{B.23}$$

To simplify this expression we exploit the fact that $V_2^T S^T$ is full-row rank (because we assume that the system is *sufficiently constrained*), so we can define a new control variable $\hat{\tau} \in \mathbb{R}^{n+6-k}$ as:

$$\hat{\tau} = V_2^T S^T \tau$$

For any value of $\hat{\tau}$ there exists at least a value of τ, which is given by:

$$\tau = (V_2^T S^T)^+ \hat{\tau}$$

We can then solve the control problem for $\hat{\tau}$ and subsequently find a corresponding value of τ.

$$\begin{aligned}\hat{\tau}^* = {} & (\hat{J}\hat{M}^{-1})^+_W((J_f J_c^+ \dot{J}_c - \dot{J}_f)\dot{q} + \hat{J}\hat{M}^{-1}\hat{h} - \hat{J}\hat{M}^{-1}\hat{J}^T f^*) \\ & + (I - (\hat{J}\hat{M}^{-1})^+_W \hat{J}\hat{M}^{-1})\hat{\tau}_0\end{aligned}$$

Setting $W = \hat{M}$ we get:

$$\hat{\tau}^* = -\hat{J}^T f^* + \hat{M}\hat{J}^+((J_f J_c^+ \dot{J}_c - \dot{J}_f)\dot{q} + \hat{J}\hat{M}^{-1}\hat{h}) + \hat{M}(I - \hat{J}^+\hat{J})\hat{M}^{-1}\hat{\tau}_0$$

Setting $\hat{\tau}_0 = \hat{h} + \bar{\tau}_0$:

$$\hat{\tau}^* = -\hat{J}^T f^* + \hat{M}\hat{J}^+(J_f J_c^+ \dot{J}_c - \dot{J}_f)\dot{q} + \hat{h} + \hat{M}(I - \hat{J}^+\hat{J})\hat{M}^{-1}\bar{\tau}_0$$

Now we can compute the joint torques as:

$$\tau^* = (V_2^T S^T)^+(-\hat{J}^T f^* + \hat{M}(\hat{J}^+(J_f J_c^+ \dot{J}_c - \dot{J}_f)\dot{q} + (I - \hat{J}^+\hat{J})\ddot{q}_0) + \hat{h})$$

where we introduced the new variable $\ddot{q}_0 = \hat{M}^{-1}\bar{\tau}_0$. Substituting the expressions for $\hat{M}$, $\hat{h}$ and $\hat{J}$ we get:

$$\begin{aligned}\tau^* = {} & (V_2^T S^T)^+ V_2^T(-J_f^T f^* + M((\underbrace{V_2(J_f V_2)^+(J_f J_c^+ \dot{J}_c - \dot{J}_f) - J_c^+ \dot{J}_c)\dot{q}}_{\ddot{q}_x} \\ & + V_2(I - (J_f V_2)^+ J_f V_2)\ddot{q}_0) + h)\end{aligned} \tag{B.24}$$

We can simplify this expression even further by noting that $\ddot{q}_x$ is just a solution of the two contact constraints, because:

$$\begin{aligned} J_f \ddot{q}_x &= J_f V_2 (J_f V_2)^+ (J_f J_c^+ \dot{J}_c - \dot{J}_f)\dot{q} - J_f J_c^+ \dot{J}_c \dot{q} \\ &= J_f J_c^+ \dot{J}_c - \dot{J}_f \dot{q} - J_f J_c^+ \dot{J}_c \dot{q} \\ &= -\dot{J}_f \dot{q} \end{aligned}$$

and also:

$$J_c \ddot{q}_x = J_c V_2 (J_f V_2)^+ (J_f J_c^+ \dot{J}_c - \dot{J}_f)\dot{q} - J_c J_c^+ \dot{J}_c \dot{q} = -\dot{J}_c \dot{q}$$

Introducing a new Jacobian $J_{cf}^T = \begin{bmatrix} J_c^T & J_f^T \end{bmatrix}$, we can then compute $\ddot{q}_x$ using a simpler expression:

$$\ddot{q}_x = -J_{cf}^+ \dot{J}_{cf} \dot{q}$$

Similarly, we can simplify the nullspace projector as:

$$V_2(I - (J_f V_2)^+ J_f V_2) = (I - J_{cf}^+ J_{cf}) = N_{cf}$$

Substituting these simplified expressions in (B.24) we get:

$$\tau^* = (V_2^T S^T)^+ V_2^T (-J_f^T f^* + M(-J_{cf}^+ \dot{J}_{cf} \dot{q} + N_{cf} \ddot{q}_0) + h) \qquad \text{(B.25)}$$

BIBLIOGRAPHY

Aghili, F. Inverse and direct dynamics of constrained multibody systems based on orthogonal decomposition of generalized force. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, pages 4035–4041, 2003. ISBN 0780377362. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1242217.

Aghili, F. A unified approach for inverse and direct dynamics of constrained multibody systems based on linear projection operator: applications to control and simulation. *Robotics, IEEE Transactions on*, 21(5): 834–849, 2005. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1512343.

Albu-Schäffer, A., Haddadin, S., Ott, C., Stemmer, A., Wimböck, T., and Hirzinger, G. The DLR lightweight robot: design and control concepts for robots in human environments. *Industrial Robot: An International Journal*, 34(5):376–385, 2007. ISSN 0143-991X. doi: 10.1108/01439910710774386. URL http://www.emeraldinsight.com/10.1108/01439910710774386.

Anderson, R. and Spong, M. W. Hybrid impedance control of robotic manipulators. *IEEE Journal on Robotics and Automation*, 4(5):549–556, 1988. ISSN 08824967. doi: 10.1109/56.20440. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=20440.

Argall, B. D., Chernova, S., Veloso, M., and Browning, B. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, May 2009. ISSN 09218890. doi: 10.1016/j.robot.2008.10.024. URL http://linkinghub.elsevier.com/retrieve/pii/S0921889008001772.

Baerlocher, P. and Boulic, R. Task-priority formulations for the kinematic control of highly redundant articulated structures. *Intelligent Robots and Systems*, (2), 1998. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=724639.

Baglini, E., Cannata, G., and Mastrogiovanni, F. Design of an embedded networking infrastructure for whole-body tactile sensing in humanoid robots. In *Proc. of 2010 IEEE-RAS Int.l Conf. on Humanoid Robots (Humanoids2010)*, pages 671–676, Nashville, TN, USA, 2010.

Bicchi, A., Salisbury, J. K., and Brock, D. L. Contact Sensing from Force Measurements. *The International Journal of Robotics Research*, 12(3):249–262, June 1993a. ISSN 0278-3649. doi: 10.1177/027836499301200304. URL http://ijr.sagepub.com/cgi/doi/10.1177/027836499301200304.

Bicchi, A., Salisbury, J. K., and Brock, D. L. Contact sensing from force measurements. *The International Journal of Robotics Research*, pages 39–50, 1993b. URL http://ijr.sagepub.com/content/12/3/249.short.

Bjorck, A. *Numerical methods for least squares problems*. Society for Industrial Mathematics, 1996. ISBN 0898713609.

Borrel, P. *Modele de comportement de manipulateurs: application a l'analyse de leurs performances et a leur commande automatique*. PhD thesis, Universit{\'e} de Montpellier, 1979.

Brooks, R. Rethink robotics - Baxter, 2008. URL http://www.rethinkrobotics.com.

Bruyninckx, H. and Khatib, O. Gauss' principle and the dynamics of redundant and constrained manipulators. *Robotics and Automation*, (April): 2563–2568, 2000. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=846414.

Calinon, S. and Billard, A. G. Incremental learning of gestures by imitation in a humanoid robot. In *Proceeding of the ACM/IEEE international conference on Human-robot interaction - HRI '07*, page 255, New York, New York, USA, 2007. ACM Press. ISBN 9781595936172. doi: 10.1145/1228716.1228751. URL http://portal.acm.org/citation.cfm?doid=1228716.1228751.

Cannata, G., Maggiali, M., Metta, G., and Sandini, G. An embedded artificial skin for humanoid robots. *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 434–438, Aug. 2008. doi: 10.1109/MFI.2008.4648033. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4648033.

Cannata, G., Denei, S., and Mastrogiovanni, F. Towards automated self-calibration of robot skin. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 4849–4854. IEEE, 2010. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5509370.

Chang, K. and Khatib, O. Operational space dynamics: efficient algorithms for modeling and control of branching mechanisms. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 1, pages 850–856. Ieee, 2000. ISBN 0-7803-5886-4. doi: 10.1109/ROBOT.2000.844156. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=844156.

Chiaverini, S. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Transactions on Robotics and Automation*, 13(3):398–410, June 1997. ISSN 1042296X. doi: 10.1109/70.585902. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=585902.

Chiaverini, S. and Sciavicco, L. The parallel approach to force/position control of robotic manipulators. *IEEE Transactions on Robotics and Automation*, 9(4):361–373, 1993. ISSN 1042296X. doi: 10.1109/70.246048. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=246048.

Chiaverini, S., Siciliano, B., and Villani, L. A survey of robot interaction control schemes with experimental comparison. *Mechatronics, IEEE/ASME Transactions on*, 4(3):273–285, 1999. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=789685.

Corke, P. Robotics, Vision and Control. In *Robotics Vision and Control*, pages 452–495. Springer Berlin Heidelberg, 2011.

Craig, J. and Raibert, M. A systematic method of hybrid position/force control of a manipulator. *COMPSAC 79. Proceedings. Computer Software and The IEEE Computer Society's Third International Applications Conference, 1979.*, pages 446–451, 1979. doi: 10.1109/CMPSAC.1979.762539. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=762539.

Dallali, H., Mosadeghzad, M., Medrano-Cerda, G. A., Docquier, N., Kormushev, P., Tsagarakis, N., Li, Z., and Caldwell, D. Development of a Dynamic Simulator for a Compliant Humanoid Robot Based on a Symbolic Multibody Approach. In *International Conference on Mechatronics, Vicenza, Italy.*, 2013.

De Lasa, M. and Hertzmann, A. Prioritized optimization for task-space control. *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3(x):5755–5762, Oct. 2009. doi: 10.1109/IROS.2009.5354341. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5354341.

De Schutter, J. and Bruyninckx, H. Force control: a bird's eye view. *Control Problems in Robotics and Automation*, 1998. URL http://www.springerlink.com/index/W27TW96671470873.pdf.

Del Prete, A., Denei, S., Natale, L., Mastrogiovanni, F., Nori, F., Cannata, G., and Metta, G. Skin Spatial Calibration Using Force/Torque Measurements. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, 2011. URL http://people.liralab.it/iron/Papers/conference/947_DelPrete_etal2011.pdf.

Del Prete, A., Natale, L., Nori, F., and Metta, G. Contact Force Estimations Using Tactile Sensors and Force / Torque Sensors. In *Human-Robot Interaction (HRI), workshop on Advances in Tactile Sensing and Touch based Human-Robot Interaction, Boston, USA, ACM/IEEE*, pages 0–2, 2012a.

Del Prete, A., Nori, F., Metta, G., and Natale, L. Control of Contact Forces: the Role of Tactile Feedback for Contact Localization. In *Intelligent*

Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, 2012b. URL http://pasa.lira.dist.unige.it/pasapdf/1108_DelPrete_etal2012.pdf.

Del Prete, A., Nori, F., Metta, G., and Natale, L. A New Framework for Prioritized Multi-Task Position-Force Control : ‚Äú Task Space Inverse Dynamics ‚Äù. In *Robotics, Science and Systems (RSS), Submitted to*, 2013.

Eberman, B. and Salisbury, J. K. Determination of manipulator contact information from joint torque measurements. In *Experimental Robotics I*, pages 463–473. Springer, 1990. URL http://www.springerlink.com/index/F102315U847773T5.pdf.

Escande, A., Mansard, N., and Wieber, P. Fast resolution of hierarchized inverse kinematics with inequality constraints. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, number 4, pages 3733–3738. IEEE, 2010. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5509953.

Featherstone, R. *Rigid body dynamics algorithms*, volume 49. Springer Berlin:, 2008.

Flash, T. and Hogan, N. The coordination of arm movements: an experimentally confirmed mathematical model. *The journal of Neuroscience*, 5 (7):1688–1703, 1985. URL http://www.jneurosci.org/content/5/7/1688.short.

Fumagalli, M., Randazzo, M., Nori, F., Natale, L., Metta, G., and Sandini, G. Exploiting proximal F/T measurements for the iCub active compliance. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1870–1876. IEEE, 2010. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5651421.

Fumagalli, M., Ivaldi, S., Randazzo, M., Natale, L., Metta, G., Sandini, G., and Nori, F. Force feedback exploiting tactile and proximal force/torque sensing. *Autonomous Robots*, Apr. 2012. ISSN 0929-5593. doi: 10.1007/s10514-012-9291-2. URL http://www.springerlink.com/index/10.1007/s10514-012-9291-2.

Gael Guennebaud, Jacob Benoit, and Others. Eigen v3, 2010. URL http://eigen.tuxfamily.org.

Gates, B. A robot in every home. *Scientific American*, 296(1):58–65, Jan. 2007. ISSN 0036-8733. URL http://www.ncbi.nlm.nih.gov/pubmed/17186834.

Gordon, S. J. and Townsend, W. T. Integration of Tactile-Force and Joint - Torque Information in a Whole-Arm Manipulator. In *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pages 464–469, 1989.

Groome, R. C. *Force feedback steering of a teleoperator system.* PhD thesis, Massachusetts Institute of Technology, 1972.

Haddadin, S., Albu-Schäffer, A., Frommberger, M., Rossmann, J., and Hirzinger, G. The DLR Crash Report: Towards a standard crash-testing protocol for robot safety-Part I: Results. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 272–279, 2009. ISBN 9781424427895. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5152602.

Heinzmann, J. and Zelinsky, A. Quantitative safety guarantees for physical human-robot interaction. *The International Journal of Robotics Research*, 22(7-8):479–504, 2003.

Hersch, M., Guenter, F., Calinon, S., and Billard, A. G. Dynamical system modulation for robot learning via kinesthetic demonstrations. *Robotics, IEEE Transactions on*, 2008. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4674620.

Hirzinger, G. and Albu-Schäffer, A. On a new generation of torque controlled light-weight robots. In *ICRA IEEE International Conference on Robotics and Automation*, pages 3356–3363, 2001. ISBN 0780364759. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=933136.

Hoffman, G. and Weinberg, G. Gesture-based human-robot jazz improvisation. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 582–587. IEEE, 2010.

Hogan, N. Impedance control: An approach to manipulation. In *American Control Conference, 1984*, pages 304–313. IEEE, 1985. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4788393.

Ivaldi, S., Fumagalli, M., Randazzo, M., Nori, F., Metta, G., and Sandini, G. Computing robot internal/external wrenches by means of inertial, tactile and f/t sensors: theory and implementation on the icub. In *Proc. of the 11th IEEE-RAS International Conference on Humanoid Robots, Bled, Slovenia*, 2011. URL http://pasa.cognitivehumanoids.eu/pasapdf/780_Ivaldi_etal2011.pdf.

Jain, A., Killpack, M., Edsinger, A., and Kemp, C. Manipulation in Clutter with Whole-Arm Tactile Sensing. *Under Review*, pages 1–20, 2011. URL http://www.cc.gatech.edu/~advait/papers/haptic_manipulation_clutter.pdf.

Janabi-Sharifi, F. Discrete-time adaptive windowing for velocity estimation. *Control System Technolog , IEEE Transactions on*, 8(6):1003–1009, 2000. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=880606.

Jeong, J. A Task-priority Based Framework for Multiple Tasks in Highly Redundant Robots. *Intelligent Robots and Systems, 2009.*, pages 5886–5891, 2009. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5354115.

Khatib, O. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, Feb. 1987. ISSN 0882-4967. doi: 10.1109/JRA.1987.1087068. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1087068.

Khatib, O. Inertial Properties in Robotic Manipulation: An Object-Level Framework. *The International Journal of Robotics Research*, 14(1):19–36, Feb. 1995. ISSN 0278-3649. doi: 10.1177/027836499501400103. URL http://ijr.sagepub.com/cgi/doi/10.1177/027836499501400103.

Khatib, O. and Maitre, L. E. Dynamic control of manipulators operating in a complex environment. *Symposium on Theory and Practice of Robots and Manipulators, 3 rd, Udine, Italy*, 1980. URL http://www.csa.com/partners/viewrecord.php?requester=gs&collection=TRD&recid=A8045685AH.

Khatib, O., Sentis, L., Park, J., and Warren, J. Whole-body dynamic behavior and control of human-like robots. *International Journal of Humanoid Robotics*, 1(1):29–43, 2004.

Kim, H.-r., Lee, K., Kwon, D.-s., and Me, G.-d. K. Emotional Interaction Model for a Service Robot. *ROMAN 2005. IEEE International Workshop on Robot and Human Interactive Communication, 2005.*, pages 672–678, 2005. doi: 10.1109/ROMAN.2005.1513857. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1513857.

Kormushev, P., Calinon, S., and Caldwell, D. G. Imitation Learning of Positional and Force Skills Demonstrated via Kinesthetic Teaching and Haptic Input. *Advanced Robotics*, 25(5):581–603, Jan. 2011. ISSN 0169-1864. doi: 10.1163/016918611X558261. URL http://www.tandfonline.com/doi/abs/10.1163/016918611X558261.

Kuniyoshi, Y., Yorozu, Y., Ohmura, Y., Terada, K., Otani, T., Nagakubo, A., and Yamamoto, T. From humanoid embodiment to theory of mind. *Embodied artificial intelligence*, pages 629–629, 2004. URL http://www.springerlink.com/index/2NK90YDCURLU4RBM.pdf.

Laurin-Kovitz, K. Design of components for programmable passive impedance. *...and Automation, 1991.* ..., 1991. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=131824.

Liegeois, A. Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Transactions on Systems, Man, and Cybernetics*, (12):868–871, 1977. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4309644.

Luh, J., Walker, M., and Paul, R. Resolved-acceleration control of mechanical manipulators. *Automatic Control, IEEE Transactions on*, 1980. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1102367.

Lumelsky, V. *Sensing, intelligence, motion: how robots and humans move in an unstructured world*. Wiley-Blackwell, 2006. ISBN 0471707406.

Manchester, I. R., Mettin, U., Iida, F., and Tedrake, R. Stable dynamic walking over uneven terrain. *The International Journal of Robotics Research*, 30(3):265–279, Jan. 2011. ISSN 0278-3649. doi: 10.1177/0278364910395339. URL http://ijr.sagepub.com/cgi/doi/10.1177/0278364910395339.

Mansard, N. A dedicated solver for fast operational-space inverse dynamics. *2012 IEEE International Conference on Robotics and Automation*, pages 4943–4949, May 2012. doi: 10.1109/ICRA.2012.6224851. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6224851.

Mansard, N., Khatib, O., and Kheddar, A. A unified approach to integrate unilateral constraints in the stack of tasks. *Robotics, IEEE Transactions on*, 25(3):670–685, 2009. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4926146.

Marques, H., Jantsch, M., Wittmeier, S., Holland, O., Alessandro, C., Diamond, A., Lungarella, M., and Knight, R. ECCE1: the first of a series of anthropomimetic musculoskeletal upper torsos. In *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, pages 391 – 396, Nashville, TN, USA, 2010.

Metta, G., Fitzpatrick, P., and Natale, L. Yarp: Yet another robot platform. *International Journal on Advanced ...*, 2006. URL http://www.intechopen.com/source/pdfs/4161/InTech-Yarp_yet_another_robot_platform.pdf.

Metta, G., Sandini, G., Vernon, D., Natale, L., and Nori, F. The iCub humanoid robot: an open platform for research in embodied cognition. In *Workshop on Performance Metrics for Intelligent Systems, National Institute of Standards and Technology*, Washington DC, USA, 2008. ACM. URL http://portal.acm.org/citation.cfm?id=1774683.

Minato, T., Yoshikawa, Y., Noda, T., and Ikemoto, S. CB 2 : A Child Robot with Biomimetic Body for Cognitive Developmental Robotics. In *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, pages 557–562, 2007. ISBN 9781424418626.

Mistry, M. and Righetti, L. Operational Space Control of Constrained and Underactuated Systems. In *Proceedings of robotics: science and systems*, 2011. URL http://www.disneyresearch.com/research/projects/mistry_rss11_final.pdf.

Mistry, M., Buchli, J., and Schaal, S. Inverse dynamics control of floating base systems using orthogonal decomposition. *2010 IEEE International Conference on Robotics and Automation*, (3):3406–3412, May 2010. doi: 10.1109/ROBOT.2010.5509646. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5509646.

Modayil, J. Discovering sensor space: Constructing spatial embeddings that explain sensor correlations. In *Development and Learning (ICDL), 2010 IEEE 9th International Conference on*, pages 120–125. IEEE, 2010. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5578854.

Nakamura, Y., Hanafusa, H., and Yoshikawa, T. Task-Priority Based Redundancy Control of Robot Manipulators. *The International Journal of Robotics Research*, 6(2):3–15, June 1987. ISSN 0278-3649. doi: 10.1177/027836498700600201. URL http://ijr.sagepub.com/cgi/doi/10.1177/027836498700600201.

Nevins, J. L. and Whitney, D. E. *The force vector assembler concept*. MIT Charles Stark Draper Laboratory, 1973.

Noda, T., Miyashita, T., Ishiguro, H., and Hagita, N. Super-Flexible Skin Sensors Embedded on the Whole Body, Self-Organizing Based on Haptic Interactions. *Proc. of Robotics: Science and Systems IV*, 2008.

Ohmura, Y., Kuniyoshi, Y., and Nagakubo, A. Conformable and scalable tactile sensor skin for curved surfaces. *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, (May):1348–1353, 2006. doi: 10.1109/ROBOT.2006.1641896. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1641896.

Papadopoulos, E. and Dubowsky, S. Dynamic singularities in free-floating space manipulators. *Space Robotics: Dynamics and Control*, 1992. URL http://www.springerlink.com/index/T1267UX52NJ58322.pdf.

Park, J. *Control strategies for robots in contact*. PhD thesis, Stanford, 2006. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.126.6622&amp;rep=rep1&amp;type=pdf.

Park, J. and Khatib, O. Multi-link multi-contact force control for manipulators. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, number Figure 2, pages 3613–3618. IEEE, 2005. ISBN 0-7803-8914-X. doi: 10.1109/ROBOT.2005.1570670. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1570670http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1570670.

Park, J. and Khatib, O. Contact consistent control framework for humanoid robots. In *Robotics and Automation, 2006. ICRA 2006. Pro-*

ceedings 2006 IEEE International Conference on, number 1, pages 1963–1969. IEEE, 2006. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1641993.

Park, J. and Khatib, O. Robot multiple contact control. *Robotica*, 26(05):667–677, Mar. 2008. ISSN 0263-5747. doi: 10.1017/S0263574708004281. URL http://www.journals.cambridge.org/abstract_S0263574708004281.

Pattacini, U., Nori, F., Natale, L., Metta, G., and Sandini, G. An experimental evaluation of a novel minimum-jerk cartesian controller for humanoid robots. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1668–1674. IEEE, 2010. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5650851.

Peters, J., Mistry, M., Udwadia, F. E., Nakanishi, J., and Schaal, S. A unifying framework for robot control with redundant DOFs. *Autonomous Robots*, 24(1):1–12, Oct. 2007. ISSN 0929-5593. doi: 10.1007/s10514-007-9051-x. URL http://www.springerlink.com/index/10.1007/s10514-007-9051-x.

Petrovskaya, A., Park, J., and Khatib, O. Probabilistic estimation of whole body contacts for multi-contact robot control. In *Robotics and Automation, 2007 IEEE International Conference on*, number c, pages 568–573. IEEE, Apr. 2007. ISBN 1-4244-0602-1. doi: 10.1109/ROBOT.2007.363047. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4209151http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4209151.

Philippsen, R., Sentis, L., and Khatib, O. An open source extensible software package to create whole-body compliant skills in personal mobile manipulators. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1036–1041. IEEE, 2011. ISBN 9781612844558. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6048762.

Pierce, D. and Kuipers, B. Map learning with uninterpreted sensors and effectors. *Artificial Intelligence*, 92(003658):169–229, 1997. URL http://linkinghub.elsevier.com/retrieve/pii/S0004370296000513.

Pratt, G. and Williamson, M. Series elastic actuators. In *Intelligent Robots and Systems, International Conference on*, 1995. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=525827.

Righetti, L. and Schaal, S. Quadratic programming for inverse dynamics with optimal distribution of contact forces. In *IEEE-RAS International Conference on Humanoid Robots*, pages 538–543, 2012.

Righetti, L., Buchli, J., Mistry, M., and Schaal, S. Inverse dynamics control of floating-base robots with external constraints: A unified view. *2011 IEEE International Conference on Robotics and Automation*, pages 1085–1090, May 2011a. doi: 10.1109/ICRA.2011.5980156. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5980156.

Righetti, L., Buchli, J., Mistry, M., and Schaal, S. Control of legged robots with optimal distribution of contact forces. *2011 11th IEEE-RAS International Conference on Humanoid Robots*, pages 318–324, Oct. 2011b. doi: 10.1109/Humanoids.2011.6100832. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6100832.

Rocco, P., Ferretti, G., and Magnani, G. Implicit force control for industrial robots in contact with stiff surfaces. *Automatica*, 33(11):2041–2047, 1997.

Roy, N., Baltus, G., Fox, D., Gemperle, F., Goetz, J., Hirsch, T., Margaritis, D., Montemerlo, M., Pineau, J., Schulte, J., and Thrun, S. Towards Personal Service Robots for the Elderly. In *Workshop on Interactive Robots and Entertainment (WIRE 2000)*, page 184, 2000.

Saab, L., Mansard, N., Keith, F., Fourquet, J.-Y., and Soueres, P. Generation of dynamic motion for anthropomorphic systems under prioritized equality and inequality constraints. *2011 IEEE International Conference on Robotics and Automation*, pages 1091–1096, May 2011a. doi: 10.1109/ICRA.2011.5980384. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5980384.

Saab, L., Ramos, O., Mansard, N., Soueres, P., and Fourquet, J.-y. Generic dynamic motion generation with multiple unilateral constraints. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 4127–4133. IEEE, 2011b. ISBN 9781612844558. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6048402.

Salisbury, K. and Townsend, W. Preliminary design of a whole-arm manipulation system (WAMS). In *IEEE International Conference on Robotics and Automation*, pages 254–260, 1988. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=12057.

Schmitz, A., Maiolino, P., Maggiali, M., Natale, L., Cannata, G., and Metta, G. Methods and Technologies for the Implementation of Large-Scale Robot Tactile Sensors. *IEEE Transactions on Robotics*, 27(3):389–400, June 2011. ISSN 1552-3098. doi: 10.1109/TRO.2011.2132930. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5771603.

Sentis, L. and Khatib, O. Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *International Journal of Humanoid Robotics*, 2(4):505–518, 2005. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.59.8780&amp;rep=rep1&amp;type=pdf.

Sentis, L. and Khatib, O. A whole-body control framework for humanoids operating in human environments. *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2641–2648, 2006. doi: 10.1109/ROBOT.2006.1642100. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1642100.

Sentis, L., Park, J., and Khatib, O. Modeling and control of multi-contact centers of pressure and internal forces in humanoid robots. *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 453–460, Oct. 2009. doi: 10.1109/IROS.2009.5356084. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5356084.

Shin, J. and Lee, J. Dynamic control of underactuated manipulators with free-swinging passive joints in cartesian space. *Robotics and Automation, 1997. Proceedings. ...*, 1997. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=606791.

Shkolnik, A. and Tedrake, R. High-dimensional underactuated motion planning via task space control. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3762–3768. Ieee, Sept. 2008. ISBN 978-1-4244-2057-5. doi: 10.1109/IROS.2008.4651150. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4651150.

Siciliano, B. and Khatib, O. *Springer Handbook of Robotics*, volume 15. Springer, 2008. ISBN 9783540239574. doi: 10.1007/978-3-540-30301-5. URL http://books.google.com/books?id=Xpgi5gSuBxsC&pgis=1.

Siciliano, B. and Slotine, J. J. E. A general framework for managing multiple tasks in highly redundant robotic systems. In *Advanced Robotics, 1991.'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, pages 1211–1216. IEEE, 1991. ISBN 0-7803-0078-5. doi: 10.1109/ICAR.1991.240390. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=240390http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=240390.

Siegwart, R., Arras, K. O., Bouabdallah, S., Burnier, D., Froidevaux, G., Greppin, X., Jensen, B., Lorotte, A., Mayor, L., Meisser, M., Philippsen, R., Piguet, R., Ramel, G., Terrien, G., and Tomatis, N. Robox at Expo.02: A large-scale installation of personal robots. *Robotics and Autonomous Systems*, 42(3-4):203–222, Mar. 2003. ISSN 09218890. doi: 10.1016/S0921-8890(02)00376-7. URL http://linkinghub.elsevier.com/retrieve/pii/S0921889002003767.

Smits, R. KDL: Kinematics and Dynamics Library, 2007. URL http://www.orocos.org/kdl.

Spong, M. The swing up control problem for the acrobot. *Control Systems, IEEE*, (May 1994), 1995. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=341864.

Spong, M. W. The control of underactuated mechanical systems. *First international conference on mechatronics*, 1994. URL http://www.clemson.edu/ces/crb/ece496/spring2004/GroupA/index_files/C59Spong.pdf.

Spong, M. W. Underactuated mechanical systems. *Control Problems in Robotics and Automation*, 1998. URL http://www.springerlink.com/index/m3npk47uk3341510.pdf.

Taichi, T., Takahiro, M., Hiroshi, I., and Norihiro, H. Automatic Categorization of Haptic Interactions-What are the Typical Haptic Interactions Between a Human and a Robot? In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 490–496. IEEE, 2006. ISBN 142440200X.

Tassa, Y., Erez, T., and Todorov, E. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4906–4913, 2012. ISBN 9781467317351. URL https://dada.cs.washington.edu/homes/todorov/papers/MPCGetUp.pdf.

Tegin, J. and Wikander, J. Tactile sensing in intelligent robotic manipulation - a review. *Industrial Robot: An International Journal*, 32(1):64–70, 2005. ISSN 0143-991X. doi: 10.1108/01439910510573318. URL http://www.emeraldinsight.com/10.1108/01439910510573318.

Tonietti, G., Schiavi, R., and Bicchi, A. Design and control of a variable stiffness actuator for safe and fast physical human/robot interaction. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, number April, pages 526–531, 2005. ISBN 078038914X. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1570172.

Umetani, Y. and Yoshida, K. Resolved motion rate control of space manipulators with generalized Jacobian matrix. *Robotics and Automation, IEEE ...*, 1989. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=34766.

Volpe, R. and Khosla, P. A theoretical and experimental investigation of explicit force control strategies for manipulators. *IEEE Transactions on Automatic Control*, 38(11):1634–1650, 1993. ISSN 00189286. doi: 10.1109/9.262033. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=262033.

webpage, R. http://www.robotran.be, 2012.

Whitney, D. E. Resolved motion rate control of manipulators and human prostheses. *Man-Machine Systems, IEEE Transactions on*, 1969. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4081862.

Whitney, D. E. Force feedback control of manipulator fine motions. *Productivity*, 1976. URL http://www.csa.com/partners/viewrecord.php?requester=gs&collection=TRD&recid=A7728650AH.

Whitney, D. E. Historical perspective and state of the art in robot force control. *The International Journal of Robotics Research*, 6(1):3, 1987. URL http://ijr.sagepub.com/content/6/1/3.short.

Wyrobek, K. a., Berger, E. H., Van der Loos, H. M., and Salisbury, J. K. Towards a personal robotics development platform: Rationale and design of an intrinsically safe personal robot. *2008 IEEE International Conference on Robotics and Automation*, pages 2165–2170, May 2008. doi: 10.1109/ROBOT.2008.4543527. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4543527.

Yoshikawa, T. Analysis and control of robot manipulators with redundancy. *Robotics research: the first international symposium*, 1984. URL http://www.engr.colostate.edu/~dga/mech564/handouts/yoshikawa.pdf.