Instantaneous Momentum-Based Control of Floating Base Systems

Gabriele Nava

Supervisor:	Dott. Giorgio Metta
Scientific Supervisor:	Dott. Daniele Pucci

Jury Members and Reviewers[*]

Rachid Alami	*Senior Scientist at LAAS-CNRS, Toulouse*
Antonio Franchi[*]	*Associate Professor at University of Twente*
Robert Griffin	*Research Scientist at IHMC, Pensacola*
Ludovic Righetti[*]	*Associate Professor at New York University*
Olivier Stasse	*Senior Researcher at LAAS-CNRS, Toulouse*

Fondazione Istituto Italiano di Tecnologia, Genova, Italy
Dynamic Interaction Control Lab

Dipartimento di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi, Università di Genova

A mio padre

Abstract

In the last two decades a growing number of robotic applications such as autonomous drones, wheeled robots and industrial manipulators started to be employed in several human environments. However, these machines often possess limited *locomotion* and/or *manipulation* capabilities, thus reducing the number of achievable tasks and increasing the complexity of robot-environment interaction. Augmenting robots locomotion and manipulation abilities is a fundamental research topic, with a view to enhance robots participation in complex tasks involving safe interaction and cooperation with humans. To this purpose, *humanoid robots*, *aerial manipulators* and the novel design of *flying humanoid robots* are among the most promising platforms researchers are studying in the attempt to remove the existing technological barriers. These robots are often modeled as *floating base systems*, and have lost the assumption – typical of fixed base robots – of having one link always attached to the ground.

From the robot control side, contact forces regulation revealed to be fundamental for the execution of interaction tasks. Contact forces can be influenced by directly controlling the robot's *momentum* rate of change, and this fact gives rise to several *momentum-based* control strategies. Nevertheless, effective design of force and torque controllers still remains a complex challenge. The variability of sensor load during interaction, the inaccuracy of the force/torque sensing technology and the inherent nonlinearities of robot models are only a few complexities impairing efficient robot force control.

This research project focuses on the design of balancing and flight controllers for floating base robots interacting with the surrounding environment. More specifically, the research is built upon the state-of-the-art of momentum-based controllers and applied to three robotic platforms: the humanoid robot iCub, the aerial manipulator OTHex and the jet-powered humanoid robot iRonCub. The project enforces the existing literature with both theoretical and experimental results, aimed at achieving high robot performances and improved stability and robustness, in presence of different physical robot-environment interactions.

Acknowledgements

First of all, I would like to thank all my current and former collegues of the Dynamic Interaction Control Lab for the valuable advices, the nice dinners and the serene atmosphere they contributed to create during the entire duration of my Ph.D. I cannot count how many times I asked for help and I've promptly found useful suggestions and support. I would like to thank you all, but in particular I would like to mention Francesco Nori, my master thesis supervisor, and also Silvio, Luca, Stefano, Claudia, Nuno, Francisco, Giulio, Diego, Yeshi, Prashanth, Ines, Anqing, Kourosh, Lorenzo, Milad, Carlotta. I would also like to thank my former collegues Francesco, Aiko, Johr, Marie, Yue, Marta, Maria, for all the support and the good time we had together while they were at IIT.

A special thank is for the members of the iRonCub team, which I joined in 2018: Fabio, Giuseppe, Hosam, Emilio, Giovanni and Augusto. Your professionalism and hard work allowed me to present in this thesis the first results of our team activity, and I hope we'll soon achieve further results together.

I would like to also thank my parents, and in particular my father for his invaluable support during my master thesis, that unfortunately I couldn't take advantage of during my Ph.D, and my mother for being always patient and supportive with me, especially during the most difficult moments. I also want to thank my girlfriend Laura and my friends Elena and Nicholas for their moral support in all the activities and challenges I faced during this five years.

Last but not least, I would like to thank a lot Daniele Pucci, who supervised my research activity from my master thesis to the end of my Ph.D. I really think that without his supervision and constant support I wouldn't have been able to grow from the student I was until being the coordinator of the iRonCub team. Thanks Dani, you are a good boss to me, and like an older – and wiser – brother every time difficult moments come.

Preliminaries

The following lists represent reference tables for the entire manuscript.

Nomenclature

p	Vectors and scalars (small letter)
e_i	Canonical vector
P	Matrices (capital letter)
L	Momentum vector
$1_{n,m}$	Identity matrix of dimension $n \times m$
$0_{n,m}$	Zero matrix of dimension $n \times m$
$(\,\cdot\,)^\top$	Transpose operator
$(\,\cdot\,)^{-1}$	Matrix inverse operator
$(\,\cdot\,)^\dagger$	Moore-Penrose matrix pseudoinverse operator
$S(\,\cdot\,)$	Skew-symmetric operator
$(\,\cdot\,)^\vee$	Inverse of skew operator
$\mathrm{diag}(\,\cdot\,)$	Diagonal matrix
$\mathrm{tr}(\,\cdot\,)$	Trace operator
$\mathrm{rank}(\,\cdot\,)$	Matrix rank
$\mathrm{vec}(\,\cdot\,)$	Matrix vectorization
$\det(\,\cdot\,)$	Matrix determinant
$\lvert\cdot\rvert$	Frobenius norm and L2 norm
$\dot{p}$	First time derivative
$\ddot{p}$	Second time derivative
∂_p	Partial derivative with respect to p
$\int(\,\cdot\,)$	Integral operator
$\mathcal{A}$	Reference frame (calligraphic letter)
$\mathcal{A}[\mathcal{B}]$	Frame with the origin of $\mathcal{A}$ and the orientation of $\mathcal{B}$
${}^{\mathcal{A}}p$	Vector expressed in frame $\mathcal{A}$
${}^{\mathcal{A}}R_{\mathcal{B}}$	Rotation from frame $\mathcal{B}$ to frame $\mathcal{A}$
$SO(3)$	Special Orthogonal group

$so(3)$	Algebra of $SO(3)$
$\sum$	Summation operator
Re	Real part of complex numbers
$\mathrm{sign}(p)$	Sign of $p \in \mathbb{R}$
$\exp(\,\cdot\,)$	Exponential and matrix exponential

Abbreviations and Acronyms

CoM	Center of Mass
DoF	Degrees of Freedom
arg min	Argument of the minimum
FT	Force/Torque
IMU	Inertial Measurement Unit
QP	Quadratic Programming
YARP	Yet Another Robot Platform
e.g.	exempli gratia
i.e.	id est
w.r.t.	with respect to
s.t.	subject to
CAD	Computer-Aided Design
URDF	Unified Robot Description Format
UAV	Unmanned Aerial Vehicle
MPC	Model Predictive Control
IP	Inverted Pendulum
SLP	Spring Loaded Pendulum
SLIP	Spring-Loaded Inverted Pendulum
ZMP	Zero Moment Point
CoP	Center of Pressure
LPIM	Linear Inverted Pendulum Model
TSID	Task Space Inverse Dynamics
LQR	Linear Quadratic Regulators
VTOL	Vertical Take Off and Landing
SEA	Series Elastic Actuators

Contents

Prologue

Since the dawn of robotics, the dream of creating machines capable of reproducing human motion has been a staple of robotics scientists. The way humans can easily interact with the surrounding environment to perform complex motions is fascinating but yet extremely difficult to replicate with a machine. Nevertheless, to endow robots with high locomotion and manipulations skills would be of great use to a wide variety of tasks, e.g., search and rescue operations, intervention in disaster-like scenarios, and to substitute humans in all-consuming jobs.

Researchers have accepted the challenge and envisioned new robotic platforms, that aim to achieve human-like –and maybe even more – locomotion and manipulation capabilities. The growing interest of the robotics community on this topic led to impressive results, as the performance recently achieved by Boston Dynamics on the ATLAS robot: `http://y2u.be/_sBBaNYex3E`.

Humanoid robotics is doubtless a research field where *terrestrial* locomotion and manipulation play a key role. One of the reasons accounting for this interest is the need of conceiving systems that can operate in places where humans are forbidden to access. The recent DARPA robotic challenge showed promising results as regards the use of humanoid robots in disaster-response, but also pointed out several limitations of these platforms when employed in real applications. Stability issues of both low and high level controllers for balancing and walking were among the main contributors to robot failures. A common high-level control strategy adopted during the competition was that of regulating the robot's momentum, which is usually referred to as *momentum-based* control.

Commercial drones are widespread and can perform *aerial* locomotion, but their interaction with the environment is often limited. Aerial manipulators seek to overcome this limitation by endowing aerial robots with a degree of manipulation. The operational space of such platforms considerably increases, but the control of the robot during interaction is often challenging. Multiple factors such as unmodeled phenomena, the inherent elasticity of the arm and the limited actuation may impair system's stability and efficiency. Recently, the *task-based* formalism emerged as an efficient control framework for aerial manipulators. It guarantees high flexibility in the choice of the control tasks, as well as the possibility to assign different priorities to the tasks and to include hardware and software limitations in the control design.

Attempts at *combining* aerial and bipedal terrestrial locomotion have also attracted the attention of the robotics community. The robot Leonardo at the Caltech combines two robotic legs with propellers to improve balancing and agility [Caltech, 2019]. Analogously, at Guangdong University of Technology, researchers are developing a legged robot with ducted fans installed at its feet. The goal is to allow the robot to take larger steps [Huang et al., 2017]. Yet, none of these robots is endowed with a degree of manipulation. An attempt at unifying manipulation, aerial, and bipedal terrestrial locomotion on a single robotic platform is carried out by the iRonCub project. The robot iRonCub is the first jet-powered humanoid robot: it is based on the humanoid robot iCub to which have been added four model-jet engines [Pucci et al., 2018]. The control of a flying humanoid robot opens new challenges for researches, and some of these challenges, such as how to handle take off and landing maneuvers, have been tackled for the first time in this thesis.

This research project contributes to the field of floating base robots control, with the design of momentum-based and optimization-based controllers to enhance robot interaction and locomotion capabilities. The control design is based on a dynamic model of the floating base system, and on a model of the robot interaction with the environment. Results are demonstrated on three different platforms: the humanoid robot iCub [Metta et al., 2010], the aerial manipulator OTHex [Staub et al., 2018], and a simulated iRonCub.

Most of the control algorithms developed in this thesis are public. They can be found in the GitHub organization dedicated to iCub software. The latest controllers as well as the link to other repositories concerning iCub torque control are available at the following url:

`https://github.com/robotology/whole-body-controllers`

A playlist describing the experiments presented in this thesis can be found at:

`https://bit.ly/37OTMqX`

The remainder of the thesis is organized as follows.

Part I: Background and Thesis Context

– **Chapter 1** briefly introduces the concept and the historical background of floating base robots, and recalls the main challenges in modeling and control of floating base systems.

– **Chapter 2** presents the robotic platforms and the simulation environments used in the remainder of the thesis.

– **Chapter 3** recalls the derivation of floating base systems equations of motion, and the model of kinematic and contact stability constraints that occur during robot interaction with the environment.

– **Chapter 4** reviews the state-of-the-art of control strategies for floating base robots, with a focus on optimization-based, instantaneous control techniques for humanoid robots and aerial manipulators.

– **Chapter 5** describes the thesis context and contribution.

Part II: Momentum-Based Control Strategies for Balancing

– **Chapter 6** implements a state-of-the-art momentum-based controller for balancing on rigid contacts. The control algorithm is implemented as a quadratic optimization problem with equality and inequality constraints.

– **Chapter 7** focuses on the design of momentum-based controllers with proven stability of the system zero dynamics. The controller is tested on the robot iCub while balancing on both one and two feet.

– **Chapter 8** proposes a framework for automatic gain tuning of momentum-based controllers.The framework is tested on the simulated robot iCub while balancing on both one and two feet.

– **Chapter 9** extends the momentum control framework described in Chapter 6 for balancing in highly dynamic environments. Simulation results are carried out on the robot iCub balancing on a seesaw board.

– **Chapter 10** implements a momentum-based controller for controlling robots with series elastic actuators.

– **Chapter 11** shows how to exploit the joints viscous and Coulomb friction in the momentum-based control design to improve the robot balancing performances. The control algorithm is tested on iCub performing highly dynamic movements while balancing.

– **Chapter 12** presents a *momentum jerk* control framework that aims at addressing common limitations of optimization-based momentum controllers, and also utilizes direct feedback from force/torque sensors.

Part III: Optimization-Based Control Strategies for Flying

– **Chapter 13** implements an optimization-based force control strategy for the aerial manipulator OTHex interacting with a rigid environment. Direct force feedback from a force/torque sensor mounted on the robot end effector is included in the control algorithm.

– **Chapter 14** describes a momentum based control architecture for the jet-powered humanoid robot iRonCub. Simulation results demonstrated that the robot can fly, balance and perform take off and landing maneuvers.

Publication list

Some of the results presented in this thesis have been accepted for publication in the following conferences and journals:

Ch. 7: *Stability Analysis and Design of Momentum-based Controllers for Humanoid Robots*, Gabriele Nava, Francesco Romano, Francesco Nori and Daniele Pucci, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Oct 2016;

Ch. 8: *Automatic gain tuning of a momentum based balancing controller for humanoid robots*, D. Pucci, G. Nava and F. Nori, 2016 IEEE/RAS International Conference on Humanoid Robots (Humanoids), Nov 2016.

Candidate's contribution: partecipation to the development of the theoretical framework; implementation and realization of the experiments;

Ch. 9: *Modeling and Control of Humanoid Robots in Dynamic Environments: iCub Balancing on a Seesaw*, Gabriele Nava, Daniele Pucci, Nuno Guedelha, Silvio Traversaro, Francesco Romano, Stefano Dafarra, Francesco Nori, 2017 IEEE/RAS International Conference on Humanoid Robots, Nov 2017;

Ch. 10: *Momentum Control of Humanoid Robots with Series Elastic Actuators*, G. Nava, D. Pucci and F. Nori, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sep 2017;

Ch. 11: *Exploiting Friction in Torque Controlled Humanoid Robots*, Gabriele Nava, Diego Ferigo, Daniele Pucci, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Oct 2018;

Ch. 12: *Jerk Control of Floating Base Systems with Contact-Stable Parametrised Force Feedback*, Gabriele Nava [1], Ahmad Gazar [1], Francisco Javier Andrade Chavez, Daniele Pucci, **conditionally accepted to** IEEE Transaction of Robotics (TRO), 2019;

Candidate's contribution: co-development of the theoretical framework; realization of the experiments;

Ch. 13: *Direct Force Feedback Control and Online Multi-task Optimization for Aerial Manipulators*, Gabriele Nava, Quentin Sable, Marco Tognon, Daniele Pucci, Antonio Franchi, IEEE Robotics and Automaton Letters, June 2020;

Ch. 14: *Position and Attitude Control of an Underactuated Flying Humanoid Robot*, Gabriele Nava, Luca Fiorio, Silvio Traversaro, Daniele Pucci, IEEE/RAS International Conference on Humanoid Robots (Humanoids), Nov 2018.

[1]The two authors equally contributed to the paper.

Part I

Background and Thesis Context

Chapter 1

Introduction

"Any sufficiently advanced technology is indistinguishable from magic".

Arthur C. Clarke

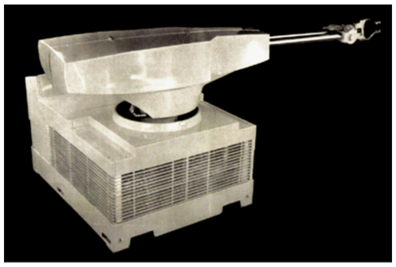
Since ancient times humanity has always been fascinated by the dream of concieving machines that can move autonomously as living creatures do. The greek mathematician Heron of Alexandria, in his volumes Pneumatica and Automata, described more than 100 machines working with air, steam and water pressure.

During the Renaissance, Leonardo da Vinci designed one of the first examples of humanoid robot, a "mechanical knight" which could stand, sit, raise its visor and independently maneuver its arms. The entire system was operated by a series of pulleys and cables. Leonardo was also the first at attempting to understand flight bird dynamics in his Codex on the Flight of Birds, in 1505.

In 18^{th} century Jacques de Vaucanson developed his famous "digestive duck", an automaton in the form of a duck appeared to have the ability of eating kernel of grain and "metabolize" them. In 19^{th} century, Hisashige Tanaka's robots were the most complex mechanized robots at the time. His creations were able to write, fire arrows, and serve tea. In the same period, artificial life started to be treated also in literature: the most famous example of this is Mary Shelley's novel Frankestein.

The word "robot" was introduced for the first time by the writer Karel Capek in his play R.U.R. (Rossum's Universal Robots), published in 1920. Capek's robots were artificial people, very similar to the modern idea of androids.

In the second half of the 20^{th} century, the technology improvements granted a remarkable development of robotics and lead to the definition of "robots" as the modern concept of mechanisms endowed with sensors, actuators and controllers. Nevertheless, these robots were initially concieved to be *attached to the ground*. Unimate, the very first industrial manipulator (Fig. 1.1a), had its base tightly bolted


(a) Unimate robot. Image source: University of Coloardo, Correll Lab.

(b) The industrial manipulator Kuka KRC2. Image source: Kuka website.

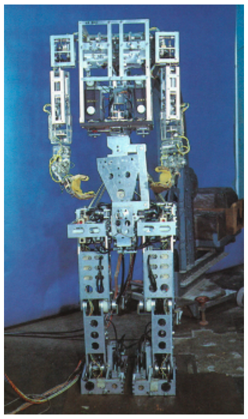
Fig. 1.1 The first manipulator Unimate (left) and a modern industrial manipulator (right).

to the soil, so that it was assumed it will never move. The robot was concieved starting from the design of a mechanical arm patented in 1954 by the american inventor George Devol, and later developed by Joseph Engelberger. It consisted of two boxes connected to an arm, and it had the possibility to memorize and execute systematic tasks.

Unimate is one of the progenitors of a large number of *fixed base* robots developed in the second half of the 20th century for industrial, medical and research applications, as the modern Kuka manipulator in Fig. 1.1b. The main characteristic of these platforms is that one of the robot link, usually referred to as the *base link*, is bolted to the ground. The fixed-base assumption renders the environment to which the robot is attached to behave as an infinite force generator: when the robot moves, reaction forces of any (reasonable) magnitude act on the robot base to ensure that it stays fixed in place.

As the technology advanced, the scientific community started dipping into robotic applications for which the fixed base assumption clearly fails. Among the others, the first humanoid robot, WABOT-1, developed in 1970 by Waseda University. It consisted of a mechanical system endowed with four limbs, a vision and conversation system (Fig 1.2a). The robot was able to walk and manipulate objects with its limbs. Later in 2000, the robot ASIMO (Fig 1.2b), developed by Honda, could walk and run like a human. After an upgrade in 2005, it became able to run at 6 $\frac{\mathrm{km}}{\mathrm{h}}$, interact with humans, and perform basic tasks like holding a platter and serving food. In 2018, the humanoid robot ATLAS developed at Boston Dynamics demonstrated impressive motion capabilities by executing a backflip (Fig. 1.3).

Robotic applications also found fertile ground in the field of aerial systems. Unmanned Aerial Vehicles (UAVs) were originally conceived as practice targets for military applications. The development of UAVs continued during the world


(a) WABOT-1.

(b) ASIMO.

Fig. 1.2 Sources: (a) http://www.humanoid.waseda.ac.jp; (b) https://favpng.com/.

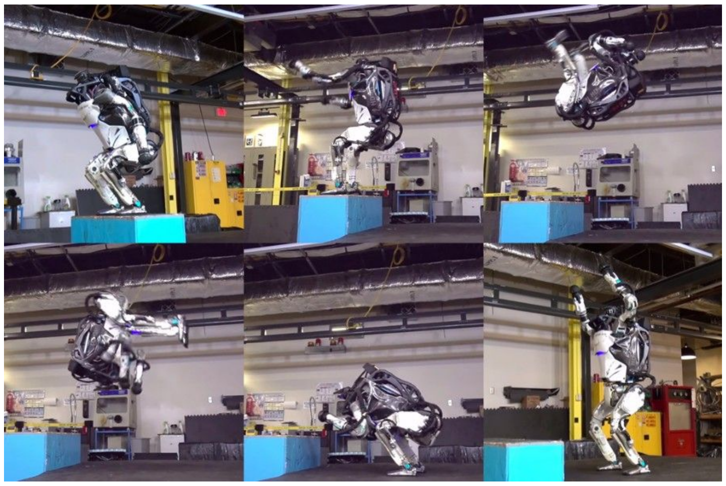


Fig. 1.3 ATLAS robot while performing a backflip. Source: http://www.pmstudio.com/.

wars, but only in the 1959 the US Air Force started planning for the use of uncrewed aircrafts. Nowadays, autonomuos and semi-autonomuos drones and quadcopters are widespread and employed for a large variety of tasks, such as sourveillance, remote monitoring and photography. In the last decade, aerial vehicles have also been equipped with rigid grippers, and, in some cases, with robotic arms with one or more degrees of freedom as in Fig. 1.4. Such *aerial manipulators* greatly enlarged the number of feasible tasks achieved by aerial robots, spanning from manipulation and grasping of objects to contact-based inspection.

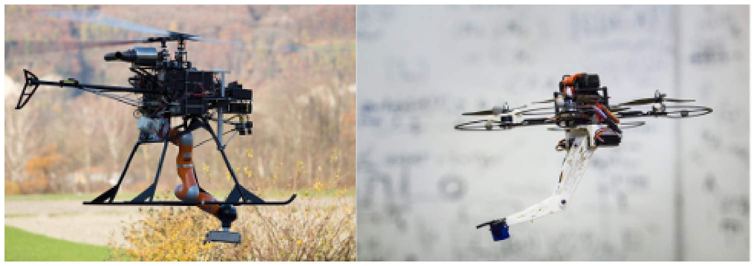


Fig. 1.4 Two aerial manipulators based on an helicopter and a quadrotor. Images sources: https://www.dlr.de/ (left) and http://naira.mechse.illinois.edu/ (right).

Contrary to fixed base systems, humanoid robots and aerial manipulators are not bolted to the ground. The base link instead *floats* above the ground and moves from place to place with the robot. Any contact the robot exerts on the ground will now produce equal and opposite reaction forces that are not canceled by the environment. If not regulated appropriately, uncontrolled contact forces may break the contact, and the robot control in this case becomes critical [Ott et al., 2011, Wensing and Orin, 2013, Kuindersma et al., 2014].

From the modeling point of view, the *floating base* formalism extends the concepts already developed for fixed base robots by adding the pose of a *virtual* link, connected passively to an inertial reference frame, to the robot model [Featherstone, 2007]. However, this additional link will introduce underactuation in the model, which in practice forbids the feedback linearization of system's dynamics [Acosta and Lopez-Martinez, 2005, Featherstone, 2007]. The underactuation is usually resolved by adding constraints to the system's dynamics: these constraints arise naturally from the contacts the robot makes with environment, and are often modelled as *rigid* constraints [Nori et al., 2015].

Next Chapters present an overview of the floating base robots which are subject of study in this research project, and deepen the state-of-the-art modeling and control strategies for floating base systems.

Chapter 2

The Robotic Platforms and Their Simulation

Three floating base robots are subject of study in this thesis: the iCub humanoid robot [Metta et al., 2010], the aerial manipulator OTHex [Staub et al., 2018], and the jet-powered humanoid robot iRonCub. This Chapter describes the hardware and software infrastructure and the control architecture of the three robots. A simulated version of the robots is also available for preliminary tests and debugging of new control strategies. In this Chapter, two simulation environments are presented: a custom Matlab-based simulator and the Gazebo-Simulink simulator.

2.1 The Humanoid Robot iCub

iCub (Fig. 2.1) is an open-source platform for research in humanoid robotics, designed by the RoboCup Consortium and built in 2004 by the Italian Institute of Technology [Metta et al., 2010]. It has the size of a human child, four limbs consisting of two legs and two arms including hands with five fingers, and it is equipped with various sensors such as force-torque sensors, a tactile skin, gyroscopes, accelerometers and cameras distributed all over its body, to be able to sense the surrounding environment. The motivations behind the strongly humanoid design is the *embodied cognition* hypothesis, that is, human-like manipulation plays a vital role in the development of human cognition. A baby learns many cognitive skills by interacting with its environment and other humans using its limbs and senses, and consequently its internal model of the world is largely determined by the form of the human body. The robot was designed to test this hypothesis, by allowing cognitive learning scenarios to be acted out by an accurate reproduction of the perceptual system and articulation of a small child, so that it could interact

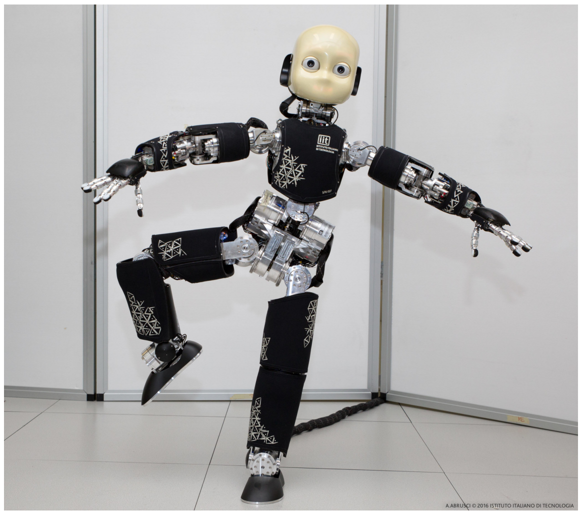


Fig. 2.1 The robot iCub.

with the world in the same way that such a child does.

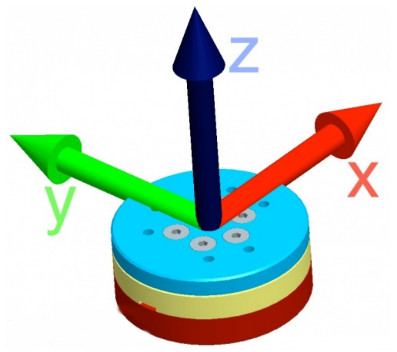
2.1.1 Mechanics

iCub is a 104 cm height robot, and weights 23-25 kg. The total mass may vary because each robot is often subject to customization by researchers. It has 53 actuated degrees of freedom, distributed as: 6 for the head, 3 for the torso, 16 for each arm (9 of which are for each hand) and 6 for each leg. Except for the hands and the head where brushed DC motors are used, all of the remaining joints are actuated by brushless DC motors with harmonic drive gears. For the purpose of this thesis, 23-25 degrees of freedom are torque controlled, while the remaining DoFs are kept at an initial configuration with position control. The material used for the body is mostly Ergal (an aluminum alloy), with some parts made of steel and plastic. Table 2.1 gives a short summary of the iCub specifications. More information about iCub robot can be found on the iCub website `http://www.icub.org/`.

2.1.2 Sensors

iCub has sensors and mechanical features that allow testing for various tasks. It is equipped with: two digital cameras in the head, microphones, force/torque sensors, tactile skin, gyroscopes and accelerometers. In addition, there are absolute position encoders in every actuated joint. The robot is also equipped with two inertial measurement units (IMU) located in the head and in the pelvis, which include a

(a) iCub mechanics.



(b) FT sensor model.

Fig. 2.2 Left: iCub mechanics; Right: FT sensors structure and sensor frame of reference.

combination of accelerometers, gyroscopes, magnetometer to measure and report robot's specific force, angular rate, and Earth magnetic field for an accurate 3D motion tracking.

Force/torque sensors

The force-torque (FT) sensors are based on a classical Wheatstone bridge design, employing 12 semiconductor strain gauges arranged in a 6 half-bridges configuration. The sensor is calibrated to measure the force/torque applied by the upper (blue) part of the sensor on the lower (red) part of sensor, and expresses it on the FT sensor reference frame.

Whole-body distributed force/torque sensors play an important role in the estimation and control of the robot joints torques. There are six force/torque sensors: one between the shoulder and the elbow on each arm, one between the hip and the knee and one between the ankle and the foot on each leg. By using these sensors, internal forces can be estimated at the joints which in turn can be used to estimate internal dynamics and external forces.

2.1.3 iCub torque control architecture

iCub is one of the few humanoid robots which is fully torque controlled. The torque control architecture is composed of two nested loops, namely the *inner* and *outer* loops. The inner loop is controlled at a frequency of 1000 Hz by an on-board PC located on the robot's head, which communicates with actuators and sensors using Ethernet protocol. The outer loop runs off-board at a frequency of 100 Hz in a cluster machine, that communicates with the on-board PC through the

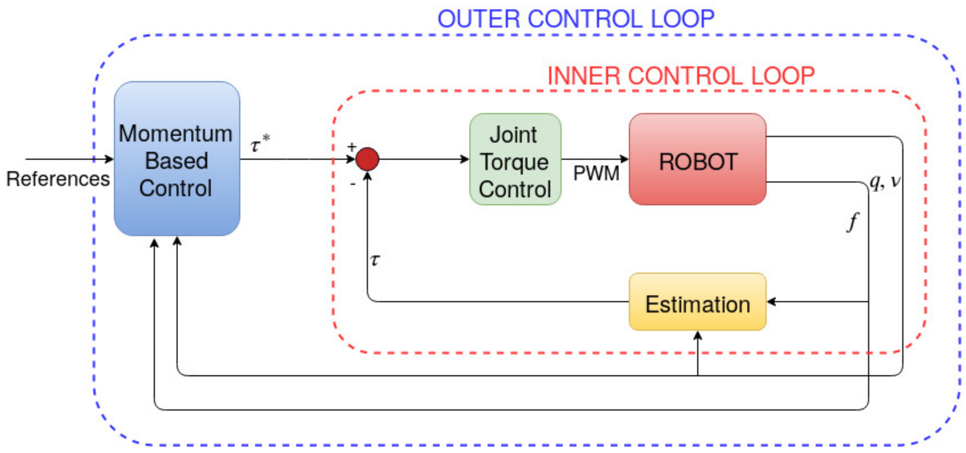


Fig. 2.3 Schematics of the iCub inner and outer loops.

Height	104 cm
Weight	23-25 kg
Sensors	Stereo Cameras, microphones, encoders, force/torque sensors, tactile sensors, gyroscopes, accelerometers
Actuators	brushless motors (150 W), DC motors. 54 motors in total
Power	220/110 V AC, tethered via 48-12 V power supply
Computing (On-board)	20 micro-controller boards for movement, 16 boards for sensors and a Pentium duo for data acquisition and synchronization
Software (On-board)	Linux
Degrees of Freedom	53 motors controlling 76 joints
Structure and Materials	Ergal, steel, plastic
Year	Started 2004, first release 2008

Table 2.1 iCub specifications.

YARP middleware [Fitzpatrick et al., 2008]. A schematic of the inner-outer control architecture is depicted in Fig. 2.3.

The outer loop

It is in charge of providing reference joint torques τ^* to the inner loop. It implements the momentum-based controllers designed in the remainder of the thesis. The control algorithm is entirely developed in Simulink. Dynamics and kinematics quantities, as well as I/O communication with the robot sensors and with the inner loop is dealt with the WBToolbox library [Romano et al., 2017]. The library wraps

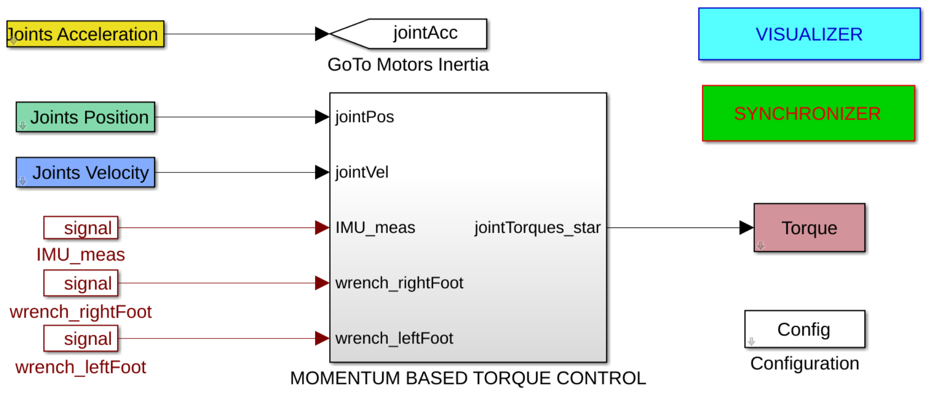


Fig. 2.4 Simulink control for iCub balancing.

the functionalities of YARP and of iDynTree, a multi-body dynamics library designed for free floating robots [Nori et al., 2015]. Fig. 2.4 shows a typical Simulink scheme used for controlling iCub.

The inner loop

The inner control loop is responsible for stabilizing the actual joint torques τ towards the reference τ^*. More specifically, the joints actuation is provided by n electric brushless motors. We assume that any reasonable motor torques τ_m can be (almost) instantly achieved by means of the fast current control loop at the motor level. This current control also compensates for the motor's back electromotive effect. Furthermore, we assume that motors and joints are rigidly connected to each other by means of the transmission element. The single joint rotation may be obtained by a linear combination of the actuators movements.

Motors dynamics

The relationship between joint and motor positions is given by:

$$s = \Gamma\theta \tag{2.1}$$

where $s \in \mathbb{R}^n$ are the joint angles, $\theta \in \mathbb{R}^n$ are the motor positions and $\Gamma \in \mathbb{R}^{n \times n}$ is a matrix that accounts for the gear box ratios and for the coupling between the input and the output rotations of the coupling mechanism. Furthermore, we also make the following assumptions [Albu-Schaffer et al., 2004, de Luca and Lucibello, 1998]:

- the friction of the mechanism is modelled as a combination of Coulomb and viscous friction only;
- the angular motor kinetic energy is due to its own spinning only, and the center of mass of each motor is along the motor axis of rotation.

Under the above assumptions the motors dynamics is then given by:

$$I_m\ddot{\theta} + K_v\dot{\theta} + K_c\text{sign}(\dot{\theta}) = \tau_m - \Gamma^\top\tau, \tag{2.2}$$

with $I_m = \text{diag}(b_i) \in \mathbb{R}^{n\times n},\ b_i > 0,\ i = 1...n$ the motors inertia matrix, while $K_v = \text{diag}(k_{v_i}) \in \mathbb{R}^{n\times n}$ and $K_c = \text{diag}(k_{c_i}) \in \mathbb{R}^{n\times n}$ are diagonal matrices collecting all the joints viscous and Coulomb friction coefficients, respectively. $\tau_m \in \mathbb{R}^n$ are the motor torques.

Low-level joint torque control

The iCub control boards allow to specify a desired Pulse Width Modulation (PWM) value to the motors. The relation between PWM and motor torques is assumed to be linear and given by: $\tau_m = K_\tau\text{PWM}$. Then, we choose $\tau_m := \tau_m^*$ as follows:

$$\tau_m^* = K_v\dot{\theta} + K_c\text{sign}(\dot{\theta}) + \Gamma^\top(\tau^* - \text{PID}(\dot{\tilde{\tau}},\ \tilde{\tau},\ \textstyle\int\tilde{\tau})). \tag{2.3}$$

where we defined the joint torques error as $\tilde{\tau} = \tau - \tau^*$ and PID stands for a classical PID controller. We substitute Eq. (2.3) into the motors dynamics Eq. (2.2) to obtain the following closed loop dynamics for the joint torques:

$$\tau = \tau^* - \Gamma^{-\top}I_m\ddot{\theta} - \text{PID}(\dot{\tilde{\tau}},\ \tilde{\tau},\ \textstyle\int\tilde{\tau}), \tag{2.4}$$

where we do not compensate for the term $I_m\ddot{\theta}$ through Eq. (2.3) because its magnitude is usually negligible compared to that of the other terms of Eq. (2.4), and its estimation is usually noisy.

2.2 iRonCub: the Flying Humanoid Robot

The iRonCub is a jet-powered version of iCub with a jet-pack mounted on the shoulders (Fig. 2.5) and two additional jet turbines mounted on the arms. The selected commercial turbines are JetCat P220 (for the jet-pack) and P100 (for the arms) and can provide a maximum thrust of 220 N and 100 N, respectively. Each turbine comes with a dedicated Electronic Control Unit (ECU) and a Ground Support Unit (GSU). The ECU can work through two different communication interfaces: throttle and serial. The throttle interface is the most basic one, and accepts

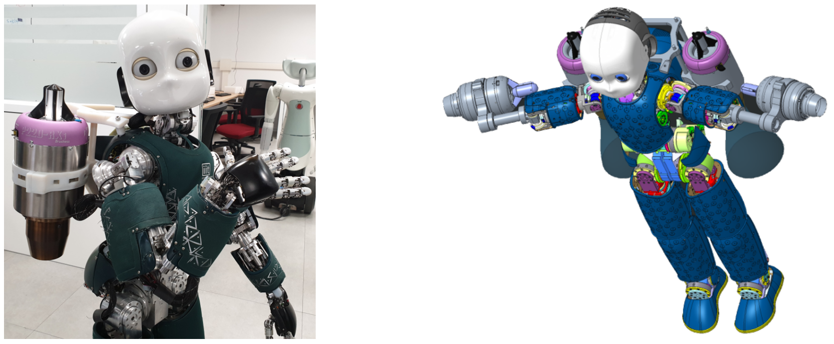


Fig. 2.5 Left: the prototype of the jet powered robot; Right: CAD model of iRonCub.

a Pulse With Modulation (PWM) input signal to regulate the turbine thrust. The serial interface receives the desired throttle as a message, but can also send a feedback message that details the running status of the engine. The default engine fuel is kerosene Jet-A1, but the performances are similar also if diesel is used. The main characteristics of the engines are in Table 2.2.

At the time of writing, flying experiments on the real platform are still in definition, and the mechanical design of the robot may change according to the requirements that come out in the experimental campaigns. A simulated version of the robot is already available in Gazebo and it is used for validating the flying control algorithms before testing on the real iRonCub.

Jet engine model	P100-RX	P220-RXi
Nominal Max. Thrust	100 N	220 N
Throttle range	$25\% - 100\%$	$25\% - 100\%$
Weight	1080 g	1850 g
Length	241 mm	307 mm
Diameter	97 mm	116.8 mm

Table 2.2 Basic specifications of the jet engines.

2.3 The Aerial Manipulator OTHex

The *Open Tilted Hexarotor* (OTHex) is a custom-made aerial vehicle developed in the LAAS laboratory in Toulouse, France. It is composed of a 3D printed base that connects 6 aluminum tubes, at the end of which are mounted coplanar-center propellers [Staub et al., 2018]. The propellers are tilted with different and optimized

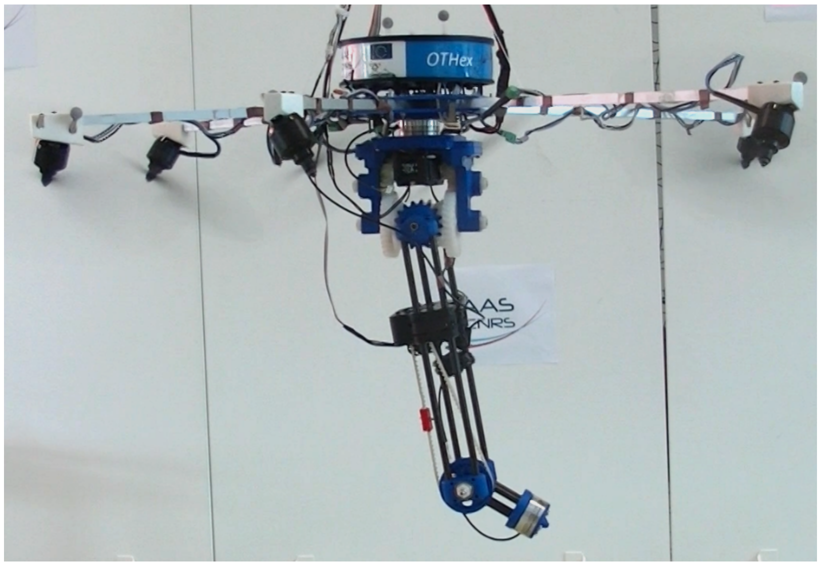


Fig. 2.6 The aerial manipulator OTHex.

angles to allow the multi-directional thrust property and therefore guarantee the local full actuation of robot dynamics. Furthermore, the frontal aperture angle between propellers has been enlarged to increase the robot's manipulation workspace. The tilting angles and other specifications are summarized in table 2.3.

2.3.1 Hardware specifications

The electronics is composed of six electronic speed controllers (ESC) BL-Ctrl-2.0 from MikroKopter, running an in-house developed firmware that performs closed-loop spinning frequency control and accepts desired spinning frequency at 1 kHz. This allows a good control of the propeller spinning velocity. Coupled with a static map of the force and moment produced at a given spinning velocity, it results in a precise force control of the platform. The robot is endowed with an on-board IMU to retrieve robot orientation and angular velocity while flying. When performing indoor experiments, IMU measurements are integrated with an external motion capture system (Optitrack MoCap) based on optical markers. Its pose measurement (120 Hz) are fused with the IMU measurements (1 kHz) using an UKF based state estimator, thus obtaining a full state estimate at 1 kHz.

2.3.2 The 3 DoF manipulator

The OTHex has been equipped with a 3D printed three degrees of freedom serial manipulator as depicted in Fig. 2.6. The manipulator has also been designed at LAAS. The motion of the first two joints is generated by means of a differential mechanism, while the last joint is belt-driven. Its end-effector can be fully customized according to the task. For the purpose of this thesis, it is composed of a pointed tool rigidly attached to a 6-axis force/torque sensor. The force sensor is the one developed at the Italian Institute of Technology for the iCub robot. The arm is controlled with a velocity control loop that commands desired motors velocities to three Dynamixel motors, whose internal encoders are also used to provide position feedback to the controller.

2.3.3 The outer loop controller

The outer loop controller is developed in Simulink and runs at 250Hz. It provides desired propellers spinning velocities and desired manipulator joint torques to the low-level motor control loops. Communication with the robot is made through two middleware: the already mentioned YARP and GenoM software [Fleury et al., 1997]. Recall that the manipulator low-level control receives reference velocities: thus, desired accelerations are computed with forward dynamics from desired torques, and numerically integrated inside the outer control loop.

OTHex total weight (without battery)	2.48 kg
Frontal aperture angle	$85°$
Propeller 1st tilt angle	$35°$
Propeller 2nd tilt angle	$-10°$
Extra admissible payload	2.9 kg
Max. lateral admissible force (hovering)	8 N

Table 2.3 OTHex specifications.

2.4 Simulation Environments

Two different simulation setups have been developed to test the outer loop controllers before implementing them on the real platforms. In both cases, we simulate the humanoid robot iCub/iRonCub with 23-25 degrees of freedom, and the OTHex with 3 degrees of freedom. The simulation environments assume that the

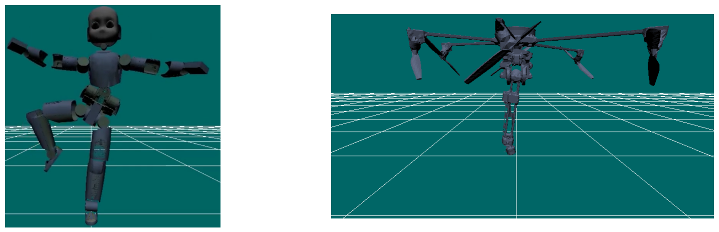


Fig. 2.7 iCub and the OTHex inside the custom simulator.

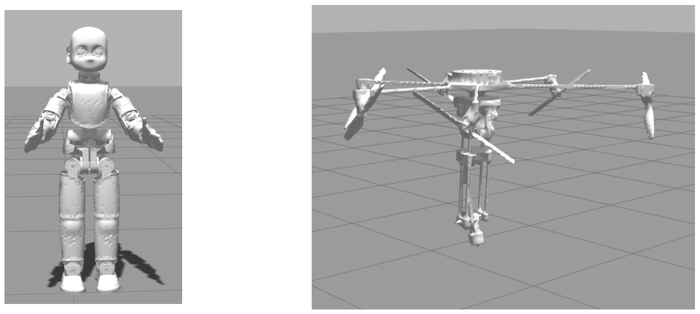


Fig. 2.8 iCub and the OTHex inside the Gazebo simulator.

reference torques τ^* and the propellers thrusts are instantly achieved, i.e. no motors dynamics (and consequently, no low-level control) is simulated. A modified simulation environment that also takes into account the motors dynamics has been exploited on the iCub robot in Chapter 10, to simulate the control of robots with elastic joints.

2.4.1 Custom setup

The simulator is entirely developed in Matlab. A robot model derived from CAD is available in URDF format. Dynamics and kinematics quantities are retrieved from the URDF file with the iDynTree software. The simulator is in charge of integrating the system's dynamics while the robot is interacting with the environment, and when $\tau = \tau^*$. The robot rotation in space is parametrized by means of a quaternion representation $\mathcal{Q} \in \mathbb{R}^4$, and the resulting state space system is then integrated through time with the numerical integrator *ode15s*. The constraints that arise during interaction, as well as $|\mathcal{Q}| = 1$, are enforced during the integration phase, and additional correction terms have been added [Gros et al., 2015]. Figure 2.7 depicts the robots iCub and OTHex inside the custom simulator.

2.4.2 Gazebo setup

The Gazebo simulator [Koenig and Howard, 2004] is the other simulation setup used for our experiments. The controller computing τ^* is developed in Simulink, and provides reference torques to Gazebo by means of Gazebo-YARP plugins. The robot dynamics is then integrated by Gazebo. Of the different physic engines that can be used with Gazebo, we chose the Open Dynamics Engine (ODE). Differently from the previous simulation environment, Gazebo allows one for more flexibility. Indeed, we only have to specify the model of the robot, and the constraints arise naturally while simulating. Furthermore, Gazebo integrates the dynamics with a fixed step semi-implicit Euler integration scheme. Another advantage of using Gazebo with respect to the custom integration scheme previously presented consists in the ability to test in simulation the same control software used on the real robot. Figure 2.8 depicts the iCub and the OTHex inside Gazebo simulator.

Chapter 3

Floating Base Systems Modeling

An appropriate model describing the system to control is the very fundation of model-based control strategies. This Chapter recalls a classical yet effective representation of floating base systems equations of motion. Furthermore, a novel derivation of the floating base system's dynamics is proposed [Traversaro et al., 2017]. The main advantage of this new representation is that the system's mass matrix is block diagonal, thus decoupling the floating base dynamics into two separate parts, the first one representing the dynamics of the robot *centroidal momentum*, the other describing the joints dynamics. Finally, we recall typical modeling strategies of the contact constraints that occur during interaction between the robot and the environment.

3.1 Floating Base Dynamics

Classical formulations of the dynamics of a multi-body system follow the rules of fundamental principles of mechanics. Systems evolving in vector spaces and Lie groups may be described using Euler-Lagrange and Euler-Poincaré formalism, respectively [Marsden and Ratiu, 2010]. Common robotics applications assume that the mechanical system is composed of a collection of $n+1$ rigid bodies, called *links*, whose relative motion is constrained by n *joints* with (usually) one degree of freedom each.

If one of the robot links is fixed with respect to an inertial reference frame $\mathcal{I}$, the system is considered *fixed base*, as it is the case for robotic manipulators attached to ground. When instead none of the robot links has an *a priori* constant pose with respect to the inertial frame, the system is referred to as *floating base* [Featherstone, 2007]. Then, any point of the system can be expressed in terms of the joint positions and the *absolute* pose of a frame attached to a link of the robot

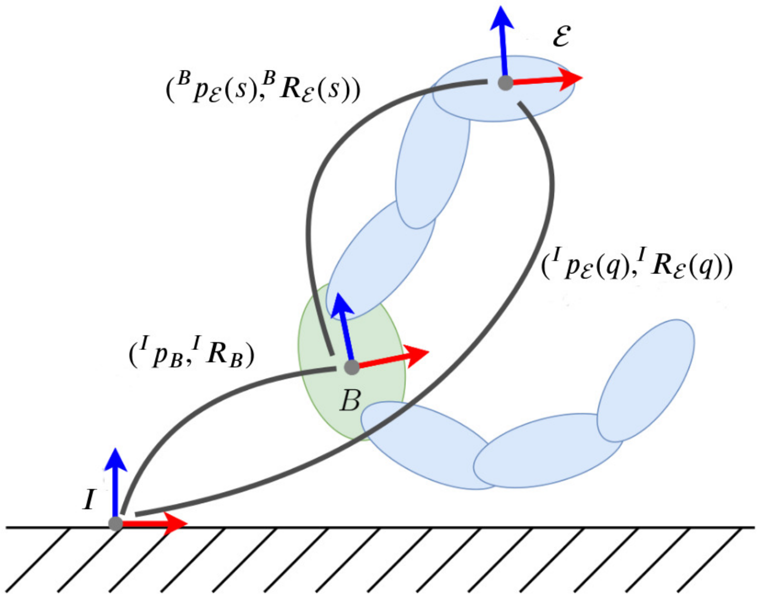

$\mathcal{E}$
$({}^{B}p_{\mathcal{E}}(s), {}^{B}R_{\mathcal{E}}(s))$
$({}^{I}p_{\mathcal{E}}(q), {}^{I}R_{\mathcal{E}}(q))$
$({}^{I}p_{B}, {}^{I}R_{B})$
B
I

Fig. 3.1 Position and orientation of any point $({}^{\mathcal{I}}p_{\mathcal{E}}, {}^{\mathcal{I}}R_{\mathcal{E}})$ of the floating base system can be expressed in terms of the joint positions and the pose $({}^{\mathcal{I}}p_{\mathcal{B}}, {}^{\mathcal{I}}R_{\mathcal{B}})$ of the base frame.

(Fig. 3.1). This frame is referred to as *base frame* $\mathcal{B}$ and it is often attached to the heaviest link of the considered system.

All robots considered in this thesis belong to this second category, and the derivation of the robot's dynamics following the floating base formalism is detailed in the next subsections.

3.1.1 Robot configuration space

The robot configuration space is characterized by the *position* and the *orientation* the base frame $\mathcal{B}$, and the joint configurations. The configuration space is then defined by

$$\mathbb{Q} = \mathbb{R}^3 \times SO(3) \times \mathbb{R}^n.$$

An element of $\mathbb{Q}$ is given by the triplet $q = ({}^{\mathcal{I}}p_{\mathcal{B}}, {}^{\mathcal{I}}R_{\mathcal{B}}, s)$, where $({}^{\mathcal{I}}p_{\mathcal{B}}, {}^{\mathcal{I}}R_{\mathcal{B}})$ denotes the origin and orientation of the *base frame* expressed in the inertial frame, and s denotes the *joint angles*. In the remainder of this thesis, the base frame rotation will be represented by means of non-minimal representations of $SO(3)$, i.e. rotation matrices and quaternions. The minimal representations, such as Euler angles, may introduce artificial singularities for the base orientation [Wieber et al., 2016]. It is now possible to define an operation associated with the set $\mathbb{Q}$ such that this set is a *Lie group*[1]. Given two elements q and ρ of the configuration space, the set $\mathbb{Q}$ is a Lie group with operation: $q \cdot \rho = (p_q + p_\rho, R_q R_\rho, s_q + s_\rho)$ [Selig, 2005].

Group algebra

The *velocity* of the multi-body system can be characterized by the *Lie algebra*[2] $\mathbb{V}$ of $\mathbb{Q}$ defined by: $\mathbb{V} = \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^n$. An element of $\mathbb{V}$ is then a triplet $\nu = ({}^\mathcal{I}v_\mathcal{B}, {}^\mathcal{I}\omega_\mathcal{B}, \dot{s}) = (\mathrm{v}_\mathcal{B}, \dot{s})$. The Lie algebra of the rotation group $so(3)$ is represented by the angular velocity vector ${}^\mathcal{I}\omega_\mathcal{B}$, which yields the following relation with the derivative of the rotation matrix [Selig, 2005]: ${}^\mathcal{I}\dot{R}_\mathcal{B} = S({}^\mathcal{I}\omega_\mathcal{B}){}^\mathcal{I}R_\mathcal{B}$.

3.1.2 System equations of motion

We assume that the robot is interacting with the environment exchanging n_c distinct forces and moments, i.e. *wrenches*[3], as in Fig. 3.2. The application of the Euler-Poincaré formalism to the multi-body system yields the equations of motion:

$$M(q)\dot{\nu} + C(q,\nu)\nu + G(q) = B\tau + \sum_{i=1}^{n_c} J_{\mathcal{C}_i}^\top(q) f_i \tag{3.1}$$

where $M(q) \in \mathbb{R}^{n+6 \times n+6}$ is the mass matrix, $C(q,\nu)\nu \in \mathbb{R}^{n+6}$ accounts for Coriolis and centrifugal effects, $G(q) \in \mathbb{R}^{n+6}$ is the gravity term, $B = (0_{n\times 6}, 1_n)^\top$ is a selector matrix, $\tau \in \mathbb{R}^n$ is a vector representing the joint torques, and $f_i \in \mathbb{R}^6$ denotes the i-th external wrench applied by the environment on the robot. We assume that the application point of the external wrench is associated with a frame $\mathcal{C}_i$, attached to the link on which the wrench acts, and has its z axis pointing in the direction of the normal of the contact plane. Then, the external wrench f_i is expressed in a frame whose orientation is that of the inertial frame $\mathcal{I}$, and whose origin is that of $\mathcal{C}_i$, i.e. the application point of the external wrench f_i. The Jacobian $J_{\mathcal{C}_i}(q)$ is the map between the robot's velocity ν and the linear and angular velocity ${}^\mathcal{I}\mathrm{v}_{\mathcal{C}_i} := ({}^\mathcal{I}v_{\mathcal{C}_i}, {}^\mathcal{I}\omega_{\mathcal{C}_i})$ of the frame $\mathcal{C}_i$, i.e. ${}^\mathcal{I}\mathrm{v}_{\mathcal{C}_i} = J_{\mathcal{C}_i}(q)\nu$.

When unnecessary, hereafter we drop the dependency of the dynamics quantities from the the robot configuration and velocity, e.g. $M(q) = M$. Similarly, we drop the superscript $\mathcal{I}$ from the kinematics quantities, e.g. ${}^\mathcal{I}p_\mathcal{B} = p_\mathcal{B}$.

[1]A group is a set equipped with a binary operation that combines any two elements to form a third element in such a way that four conditions (axioms) are satisfied: closure, associativity, identity and invertibility. A Lie group is a group whose elements are organized continuously and smoothly.

[2]Any Lie group gives rise to a Lie algebra, which is its tangent space at the identity.

[3]As an abuse of notation, we define as *wrench* a quantity that is not the dual of a *twist*.

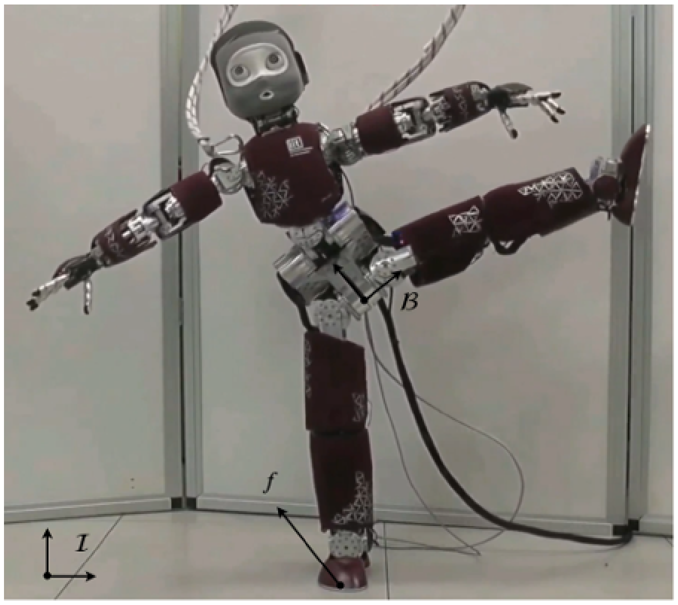


Fig. 3.2 The robot iCub while balancing. The figure depicts the inertial frame $\mathcal{I}$, the base frame $\mathcal{B}$ and the contact wrench f.

3.2 Robot Dynamics in Centroidal Coordinates

This Section recalls a new expression of the equations of motion (3.1), which resulted to be particularly useful for the design of momentum-based controllers. In particular, the following Lemma presents a change of coordinates in the state space (q, ν) that transforms the system dynamics (3.1) into a new form where the mass matrix is block diagonal, thus decoupling joint and base frame accelerations. Also, in this new set of coordinates, the first six rows of Eq. (3.1) correspond to the *centroidal dynamics*. In the specialized literature, the term centroidal dynamics is used to indicate the rate of change of the robot's momentum expressed at the center-of-mass, which then equals the summation of all external wrenches acting on the multi-body system [Orin et al., 2013].

Lemma 1 *The proof is given in [Traversaro et al., 2017]. Consider the equations of motion given by* (3.1) *and the mass matrix partitioned as following*

$$M = \begin{bmatrix} M_{\mathcal{B}} & M_{\mathcal{B}s} \\ M_{\mathcal{B}s}^{\top} & M_s \end{bmatrix}$$

with $M_{\mathcal{B}} \in \mathbb{R}^{6\times 6}$, $M_{\mathcal{B}s} \in \mathbb{R}^{6\times n}$ *and* $M_s \in \mathbb{R}^{n\times n}$. *Perform the following change* *of state variables:*

$$q := q, \;\; \bar{\nu} := T(q)\nu, \tag{3.2}$$

where:

$$T := \begin{bmatrix} {}^{\mathcal{G}[\mathcal{I}]}X_{\mathcal{B}} & {}^{\mathcal{G}[\mathcal{I}]}X_{\mathcal{B}} M_{\mathcal{B}}^{-1} M_{\mathcal{B}s} \\ 0_{n\times 6} & 1_n \end{bmatrix},$$
$${}^{\mathcal{G}[\mathcal{I}]}X_{\mathcal{B}} := \begin{bmatrix} 1_3 & -S(p_c - p_{\mathcal{B}}) \\ 0_{3\times 3} & 1_3 \end{bmatrix}$$

where the superscript $\mathcal{G}[\mathcal{I}]$ denotes the frame with the origin located at the center of mass, and with orientation of $\mathcal{I}$, while p_c is the center of mass position. Then, the equations of motion with state variables $(q, \overline{\nu})$ can be written as:

$$\overline{M}\dot{\overline{\nu}} + \overline{C}\overline{\nu} + \overline{G} = B\tau + \sum_{i=1}^{n_c} \overline{J}_{\mathcal{C}_i}^{\top} f_i, \tag{3.3}$$

with the dynamics quantities given by

$$\begin{aligned} \overline{M} &= T^{-\top} M T^{-1} = \begin{bmatrix} \overline{M}_{\mathcal{B}} & 0_{6\times n} \\ 0_{n\times 6} & \overline{M}_s \end{bmatrix}, \\ \overline{C} &= T^{-\top}(M\dot{T}^{-1} + CT^{-1}), \\ \overline{G} &= T^{-\top} G = mge_3, \\ \overline{J}_{\mathcal{C}_i} &= J_{\mathcal{C}_i} T^{-1} = \begin{bmatrix} \overline{J}_{\mathcal{C}_i}^{\mathcal{B}} & \overline{J}_{\mathcal{C}_i}^{s} \end{bmatrix}, \end{aligned}$$

and also

$$\overline{M}_{\mathcal{B}} = \begin{bmatrix} m1_3 & 0_{3\times 3} \\ 0_{3\times 3} & I \end{bmatrix}, \;\; \overline{J}_{\mathcal{C}_i}^{\mathcal{B}} = \begin{bmatrix} 1_3 & -S(p_{\mathcal{C}_i} - p_c) \\ 0_{3\times 3} & 1_3 \end{bmatrix},$$

where m is the total mass of the robot and I is the total inertia matrix computed with respect to the center of mass, with the orientation of $\mathcal{I}$.

The above Lemma points out that the mass matrix of the transformed system (3.3) is block diagonal, i.e. the transformed base acceleration is independent from the joint acceleration. More precisely, the transformed robot's velocity $\overline{\nu}$ is given by $\overline{\nu} = \left({}^{\mathcal{I}}v_c^{\top} \;\; {}^{\mathcal{I}}\omega_c^{\top} \;\; \dot{s}^{\top}\right)^{\top}$ where ${}^{\mathcal{I}}v_c$ is the velocity of the center-of-mass of the robot, and ${}^{\mathcal{I}}\omega_c$ is the so-called *average angular velocity*[4] [Jellinek and Li, 1989, Essén, 1993, Orin et al., 2013]. Hence, Eq. (3.3) unifies what the specialized robotic literature usually presents with two sets of equations: the equations

[4]The term ${}^{\mathcal{I}}\omega_c$ is also known as the *locked angular velocity* [Marsden and Scheurle, 1993].

of motion of the free floating system and the *centroidal dynamics*. For the sake of correctness, let us remark that defining the *average angular velocity* as the angular velocity of the multi-body system is not theoretically sound. In fact, the existence of a rotation matrix $R(q) \in SO(3)$ such that $\dot{R}(q)R^\top(q) = S({}^{\mathcal{I}}\omega_c)$, i.e. the integrability of ${}^{\mathcal{I}}\omega_c$, is still an open issue. Observe also that the gravity term $\overline{G}$ is constant and influences the acceleration of the center-of-mass only. This is a direct consequence of the property that $G = Mge_3$, with $e_3 \in \mathbb{R}^{n+6}$.

3.2.1 Final representation of system dynamics

In the sequel of this thesis, we assume that the equations of motion are given by (3.3), unless otherwise specified. As a consequence, the mass matrix and the centroidal momentum are of the form:

$$\overline{M} = \begin{bmatrix} \overline{M}_{\mathcal{B}} & 0_{6\times n} \\ 0_{n\times 6} & \overline{M}_s \end{bmatrix}, \quad L = \overline{M}_{\mathcal{B}} \begin{bmatrix} {}^{\mathcal{I}}v_c^\top \\ {}^{\mathcal{I}}\omega_c^\top \end{bmatrix}$$

with $L := (L_l^\top, L_\omega^\top)^\top \in \mathbb{R}^6$ the robot centroidal momentum, and $L_l, L_\omega \in \mathbb{R}^3$ the linear and angular momentum at the center of mass, respectively. As an abuse of notation but for the sake of clarity, we hereafter drop the overline notation. The equations of motion of the multi-body system are then given by:

$$M\dot{\nu} + h = B\tau + J^\top f \tag{3.4}$$

that can be splitted as

$$M_{\mathcal{B}}\dot{\mathrm{v}}_{\mathcal{B}} + h_{\mathcal{B}} = J_{\mathcal{B}}^\top f \tag{3.5a}$$

$$M_s\ddot{s} + h_s = J_s^\top f + \tau \tag{3.5b}$$

where we defined $h := C\nu + G \in \mathbb{R}^{n+6}$ and its partition $h = (h_{\mathcal{B}}, h_s), h_{\mathcal{B}} \in \mathbb{R}^6, h_s \in \mathbb{R}^n$. The term $J^\top f := \sum_{i=1}^{n_c} \overline{J}_{\mathcal{C}_i}^\top f_i$ is a compact representation of the contact wrenches acting on the system, and stacks all the contact wrenches in a single vector $f := \begin{bmatrix} f_1^\top & ... & f_{n_c}^\top \end{bmatrix}^\top$.

3.2.2 Remark on system underactuation

The control algorithms that we shall develop in this thesis consider the joint torques τ as the control input in charge of stabilizing the system's dynamics (3.4). To impose a desired dynamic behavior to the robot, we may attempt to design a controller that applies *feedback linearization* to system (3.4). However, the number of joint torques (n) is less than the number of degrees of freedom ($n+6$). This implies that

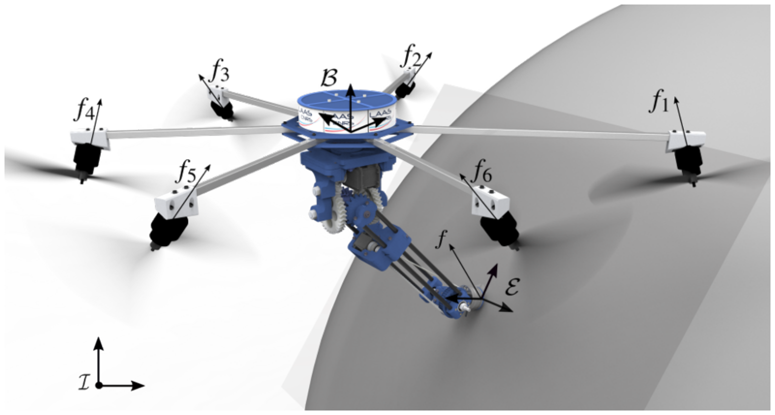


Fig. 3.3 CAD representation of the aerial manipulator OTHex interacting with the environment. Besides the contact wrench f, the model must include also the thrust forces f_{1-6}.

system (3.4) is *underactuated*, and that full feedback linearization of the underlying system is forbidden [Acosta and Lopez-Martinez, 2005]. The problem of underactuation is usually overcame by selecting a state-dependent output $y(q, \nu, \tau)$ with dimension lower than n, but whose stabilization affects the dynamics of the overall system. In the next Chapters, the output y is the chosen as the robot momentum L.

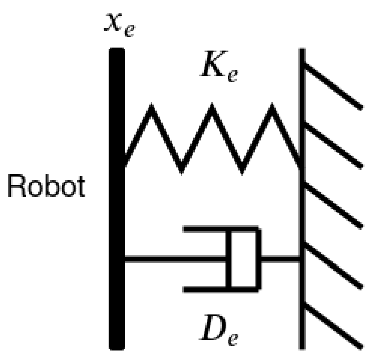
3.2.3 Modeling of aerial systems

Aerial systems as the OTHex in Fig. 3.3 are subject to additional forces (and eventually moments) to be added to Eq. (3.4). These forces represent aerodynamic effects and the effect of the thrusters mounted on the flying robot. In the remainder of the thesis, the thrust forces are considered additional control inputs to the system, while their dynamics (usually very fast) is neglected or compensated by a low-level motors control loop. The number (and orientation) of these thrusters determines whether the system remains or not underactuated. Extended dynamic models that also include thrust forces are designed in Chapters 13–14.

3.3 Contact Modeling

3.3.1 Kinematic constraints

The contact wrenches f describe how robot interaction with the environment influences the system's dynamics. The nature of these forces depends on the type of contact constraints that arise between the robot and the environment. Contact models for floating base systems are often grouped into three categories:



(a) Model of a compliant contact.

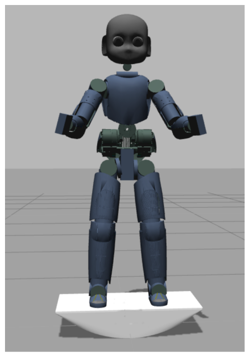

(b) iCub balancing on a seesaw.

Fig. 3.4 Contact models. a) compliant contact; b) dynamical contact.

- *Rigid* and *static* contact: the interaction occurs between two rigid objects (usually, a robot link and the environment). The environment is not moving;

- *Compliant* contact: the robot link and/or the environment are compliant. The interaction may be modeled e.g. by means of a linear spring-damper system as in Fig. 3.4a. The contact forces are calculated as [Erickson et al., 2003]:

$$f = K_e(x - x_e) + D_e(\dot{x} - \dot{x}_e),$$

  with x, x_e the end-effector and environment position, and K_e, D_e proper stiffness and damping matrices.

- *Dynamical* contact: the robot interacts with another dynamical system, whose dynamics is also affected by the robot's reactions. This type of contact occurs, for example, during human-robot interaction. To exemplify interaction with dynamical contacts, in the remainder of the thesis we consider the case study of iCub balancing on a semi-cylindrical seesaw, as depicted in Fig. 3.4b. The wrenches exchanged between the robot and the seesaw are computed in light of the seesaw dynamics:

$$f = M_e\ddot{x}_e + h_e - f_c,$$

  where M_e, h_e describe the environment dynamics, and f_c are the forces exchanged by the seesaw with the (rigid and static) ground plane. In this case effective control solutions are required to stabilize both the robot and the seesaw dynamics [Luca and Manes, 1994, Anderson and Hodgins, 2010].

In this thesis, we assume that during interaction the system dynamics evolves in a subset $\mathbb{K} \subset \mathbb{Q}$ so that the contact with the environment is always preserved, namely, a set of *holonomic constraints* acts on System (3.4). These holonomic constraints are of the form $c(q) = 0$, and may represent, for instance, a frame having a constant position and/or orientation w.r.t. the environment. Then, the environment may be *static*, or instead constituted by another dynamical system with its own dynamics.

Interaction with a static environment

In case the environment is considered static, the holonomic constraint at the i-th contact location is of the form: $J_{\mathcal{C}_i}\nu = 0$, i.e. the link in contact has zero linear and angular velocity. The holonomic constraints associated with all rigid contacts can be compactly represented as:

$$J\nu = \begin{bmatrix} J_{\mathcal{C}_1} \\ \cdots \\ J_{\mathcal{C}_{n_c}} \end{bmatrix} \nu = \begin{bmatrix} J_{\mathcal{B}} & J_s \end{bmatrix} \nu = J_{\mathcal{B}} \mathrm{v}_{\mathcal{B}} + J_s \dot{s} = 0, \tag{3.6}$$

with $J_{\mathcal{B}} \in \mathbb{R}^{6n_c \times 6}$, $J_s \in \mathbb{R}^{6n_c \times n}$ as in Eq. (3.4). By differentiating the kinematic constraint (3.6), one obtains

$$J\dot{\nu} + \dot{J}\nu = J_{\mathcal{B}}\dot{\mathrm{v}}_{\mathcal{B}} + J_s\ddot{s} + \dot{J}_{\mathcal{B}}\mathrm{v}_{\mathcal{B}} + \dot{J}_s\dot{s} = 0. \tag{3.7}$$

The combination of Eq. (3.4)–(3.7) represents the *constrained* system dynamics interacting with a rigid and static environment.

Dynamical contacts

There are situations in which the environment dynamics cannot be neglected. If the environment dynamics is not known, control solutions often make use of adaptive or robust controllers [Lee and Goswami, 2012, Kim et al., 2007]. There are also situations in which the environment dynamics may be known *a priori*, e.g. while interacting with a wheeled chair, balancing on a moving platform or even interacting with humans. In this case, the contact constraints (3.7) become:

$$J\dot{\nu} + \dot{J}\nu = \dot{\mathrm{v}}_c,$$

where $\dot{\mathrm{v}}_c \in \mathbb{R}^{6n_c}$ represents the accelerations at the contact locations. These accelerations are estimated through the dynamical model of the environment. The resulting interaction wrenches couple the robot and environment dynamics, and therefore the contact dynamics must be taken into account in the control design. A control architecture for iCub balancing on a seesaw is presented in Chapter 9.

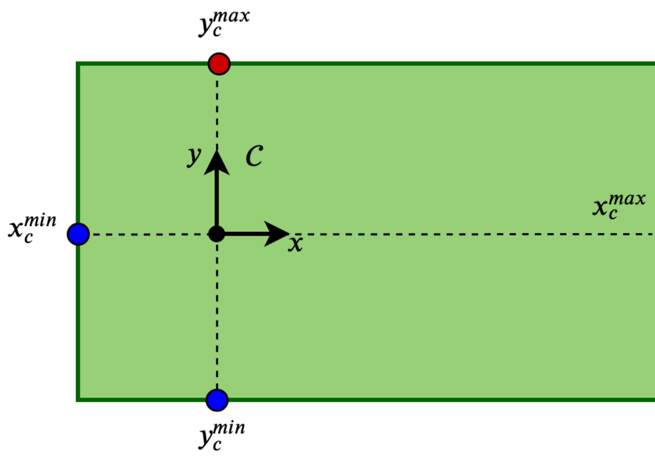


Fig. 3.5 Contact surface. The picture highlights the rectangle's dimensions w.r.t. the contact frame $\mathcal{C}$.

3.3.2 Contact stability constraints

Differently from fixed base robots, floating base systems are not bolted to the ground, but they are free to activate and deactivate contacts according to the current task. Maintaining holonomic constraints of the form $c(q) = 0$ usually require that the forces and moments at the i-th *active* contact $f^i = [f_x,\ f_y,\ f_z,\ M_x,\ M_y,\ M_z]^\top$ preserve the *contact stability* conditions. In case of *planar unilateral* contact, which is the most common type of interaction, the contact stability conditions may be represented as follows:

$$f_z > 0, \tag{3.8a}$$

$$\sqrt{f_x^2 + f_y^2} < \mu_c f_z, \tag{3.8b}$$

$$y_c^{min} < \frac{M_x}{f_z} < y_c^{max}, \tag{3.8c}$$

$$x_c^{min} < \frac{M_y}{f_z} < x_c^{max}, \tag{3.8d}$$

$$\left|\frac{M_z}{f_z}\right| < \mu_z. \tag{3.8e}$$

Being the constraints unilateral, condition (3.8a) imposes the positivity of the force normal to the contact surface in order to keep the contact constraint active. Eq. (3.8b) limits the magnitude of the forces parallel to the contact surface in order not to overcome static friction, and μ_c is the static friction coefficient. Conditions

(3.8c)-(3.8d) constrain the local Center of Pressure to remain inside the contact surface, which is assumed to be rectangular with dimensions $x_c^{min}, x_c^{max}, y_c^{min}, y_c^{max}$, calculated w.r.t. the contact reference frame $\mathcal{C}$ and defined as in Fig. 3.5. Eq. (3.8e) imposes no foot rotation along the axis normal to the contact surface, and μ_z is the torsional friction coefficient.

Chapter 4

Control Strategies for Floating Base Robots

In the first part of the Chapter we introduce the problem of controlling floating base systems, and we place emphasis on the key elements that characterize the control strategies developed in this thesis: the definition of an *instantaneous* control input and the formulation of the control problem as an optimization based on *Quadratic Programming*. The second part of the Chapter presents an overview of state-of-the-art control strategies for humanoid robots and aerial systems.

4.1 The General Control Problem

In its more general formulation, the control of floating base systems can be formulated as the problem of finding a series of control inputs u that will bring the dynamical system $\dot{x} = \mathrm{f}(x, u)$ from an initial state towards a desired one.

To achieve specific control objectives such as, e.g., robot balancing, it is useful to design a proper output function $y = y(\dot{x}, x, u)$, which in general depends on the robot state x and dynamics $\dot{x}$, and on the input u. The aim is to find the optimal input $u^*(t)$ that steers y towards the desired value $y^d(t)$, being y^d the output that realizes our control objectives. In the remainder of the thesis, the output y often coincides with the momentum dynamics $\dot{L}$.

Nevertheless, the solution to the above presented control problem may not be trivial for floating base systems: the robot underactuation and the necessity of preserving contact stability constraints during interaction can severely limit the space of fesible inputs u and impair the achievement of the desired output y^d. These limitations may be addressed as described in the next Section of this Chapter.

4.1.1 The control formulation as an optimization problem

In real robots control, the input u^* that guarantees $y = y^d$ may actually result to be sub-optimal when hardware and software limitations are also considered in the problem. For example, it is undesirable that the control input is kept constantly close to the robot actuation limits.

Sometimes, to mantain robot balancing or to achieve a desired motion for the robot end-effector, it may be sufficient to keep the error $\tilde{y} = y - y^d$ (or its norm) bounded to a small value. In these situations the control problem is often formulated by applying *optimization-based* techniques [Pratt and Tedrake, 2006]. One of the main advantages of these techniques is that contact stability constraints, actuators limits and other nonlinear effects can be included as hard constraints in the optimization problem. The (eventually local) optimal solution u^* minimizes the error $\tilde{y}$, while also respecting the software and hardware constraints on the input. The optimization problem is usually formulated as:

$$\begin{aligned} \underset{u}{\text{minimize}} = \ & ||y(\dot{x}, x, u) - y^d||^2 \qquad (4.1) \\ \text{subject to:} & \\ & \dot{x} = \mathrm{f}(x, u) \\ & x \in \mathbb{K} \\ & f \in \mathcal{K}, \\ & u_l \leq u \leq u_u \end{aligned}$$

where the constraints accounts for the system dynamics, eventual kinematic constraints such as joints limits and contact stability constraints occurring during interaction. u_l and u_u are the boundaries on the control input u and are added to the optimization problem as linear inequalities. Note that the overall optmization problem is, in general, nonlinear. Nevertheless, specific control and modeling assumptions may render the optimization (4.1) a quadratic problem [Wensing and Orin, 2013]. Quadratic Programming (QP) allows for fast resolution of the control problem (4.1), that may be solved run-time at relatively high control frequencies. Furthermore, being the QP optimization problem *convex* by construction, the solution u^* is unique and globally optimal.

The model of system dynamics (3.4) proposed in this thesis and the kinematic constraints as in (3.7) render particularly simple to formulate control algorithms as a QP problems, in which y is (usually) the momentum dynamics $\dot{L}$ and u are the robot joint torques. The only nonlinearity left in the system comes from the contact stability constraints. To this purpose, the static friction constraints in (3.8b) are approximated with linear inequalities. By consequence, the contact stability constraints appear linearly w.r.t. the contact forces f, and can be compactly written

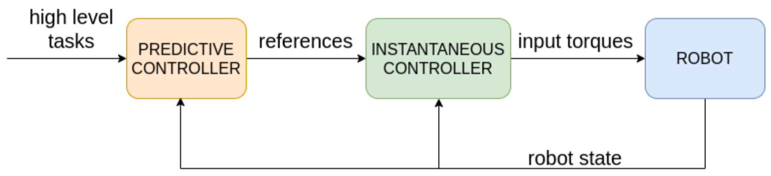


Fig. 4.1 Control framework for floating base systems with two loops: a low-frequency predictive controller (planner) provides references to a fast, instantaneous controller.

as $C_f f \leq b_f$, with C_f, b_f obtained with the linear approximation of the friction cone and by properly rearranging the other constraints in Eq. (3.8).

4.1.2 Predictive and instantaneous controllers

In order to find the optimal input u^*, two complementary strategies are usually adopted in the field of floating base robots control: *predictive* controllers, that consider the system evolution over a finite time horizon, and *instantaneous* controllers, that optimize only for the current time step. Usually, the necessity of considering the system evolution beyond the current time step arises when the control task implies to break/create new contacts with the environment, and therefore discrete events occur. For example, during push recovery the inevitably robot fall lead to dynamic equilibrium and balanced state if the foot landing –and, by consequence, the activation of the foot-ground contact– is properly planned. In this case, Model Predictive Control (MPC) techniques can be used to compute online the optimal control sequence for the next time steps [García et al., 1989].

As the time horizon increases, nonlinearities in the system dynamics and the (usually) high number of degrees of freedom render the optimal control sequence difficult to be solved online. At this stage, simplified models of the robot are often considered [Full and Koditschek, 1999]. Another possibility is to use the predicitve controller as a planner that provides reference trajectories to a fast inner control loop implementing an instantaneous control strategy, as in Fig. 4.1 [Dafarra et al., 2018]. In the extreme case, the predictive controller may be even substituted with an offline planner. Another possibility is to replace the planner with the human intervention, as it occurs during robot teleoperation [Penco et al., 2018].

If instead the task to be achieved does not require planning over time, for example during robot balancing, instantaneous controllers may be preferred over the predictive ones. Shortening the time horizon to just the current time step allows to run the controller at higher frequencies, and to implement nonlinear controllers based on the *whole-body* system dynamics [Herzog et al., 2014].

This thesis shall consider only tasks for which no planning is required, and therefore the control problem is tackled with the design of whole-body instantaneous controllers. In specific tasks where contact activation/deactivation is required, the controller passes discontinuously from two different balancing states. The reference trajectories are either selected from a set of predefined references by a finite state machine, or provided by the user by means of a joypad interface.

4.2 Humanoid Robots Control for Balancing

Balancing controllers for humanoid robots have long attracted the attention of the robotics community [Caux et al., 1998, Hirai et al., 1998]. Kinematic and dynamic controllers have been common approaches for ensuring a stable robot behavior for years [Hyon et al., 2007, Huang et al., 2000]. The common denominator of these strategies is to consider the robot attached to ground, which allows one for the application of classical algorithms developed for fixed-based manipulators.

The emergence of floating-base formalism for characterizing the dynamics of multi-body systems has loosened the assumption of having a robot link attached to ground [Featherstone, 2007]. At the control level, one of the major complexities when dealing with floating base systems comes from the robot's underactuation: in fact, the underactuation forbids the full feedback linearization of the underlying system [Acosta and Lopez-Martinez, 2005].

The system underactuation is usually addressed by means of the constraints that arise from the contacts between the robot and the environment. This requires a close attention to the forces the robot exerts on the environment, because uncontrolled forces may break the contacts, thus rendering the control of the robot critical. To this purpose, the recent research effort on humanoid robots gave impetus to development of force and torque control of floating-base systems [Mason et al., 2016, Ott et al., 2011, Hopkins et al., 2015].

Control algorithms based on contact forces regulation can be categorized according to the model criterion used for the control design [Grizzle et al., 2014, Hurmuzlu et al., 2004]. There are controllers that make use of simplified, or *template* models of the system dynamics [Full and Koditschek, 1999]. Examples of such models are the Inverted Pendulum (IP), Spring Loaded Pendulum (SLP), and Spring-Loaded Inverted Pendulum (SLIP) models [Blickhan, 1989, McMahon and Cheng, 1990]. In order to exploit these simplified models for robot control, contact stability metrics have been defined. Two of them are the *zero-moment-point* (ZMP) [Vukobratovic and Borovac, 2004] and the *center-of-pressure* of the contact forces (CoP). These two metrics can be proven to coincide [Sardain and Bessonnet, 2004]. The ZMP can be defined as "the point on the ground where the tipping moment acting on the biped, due to gravity and inertia forces, equals zero, the tipping moment being defined as the component of the moment that is tangential to the supporting surface" [Sardain and Bessonnet, 2004]. Many ZMP-based controllers thus exploit the idea that if the ZMP remains strictly within the convex hull of the stance foot, the foot does not flip [Yamaguchi et al., 1999, Yagi and Lumelsky, 2000]. Linear Inverted Pendulum Model (LPIM), which relates ZMP to the robot center-of-mass dynamics, turns out to be a very useful tool for the design of ZMP based walking controllers [Kajita et al., 2001, Hun-ok Lim et al., 2002].

A limitation of the LPIM-ZMP approach to humanoid robots control is that the model does not consider the inertial effects. To include these effects, the model is often complemented with a flying wheel [Takenaka et al., 2014]. The combination of pendulum models with a flying wheel led to the definition of the Capture Point [Pratt et al., 2006]: if an inverted pendulum with some (proper) velocity is anchored at the capture point, it will tend to the vertical position when unperturbed. Capture point (CP) based walking controllers are an effective solution to the locomotion problem of humanoid robots [Pratt and Tedrake, 2006, Pratt et al., 2012].

The control algorithms deduced by using template models usually do not exploit the complete robot dynamics, since these models represent approximations of the humanoid robot. When dealing with complete robot models, a classical control approach is the so-called Task-Space-Inverse-Dynamics (TSID). TSID based techniques have been largely employed for the control of humanoid and quadruped robots [Righetti et al., 2011, Koolen et al., 2013]. The aim of these strategies is the achievement of several control objectives, which are organized in a hierarchical structure. High priority tasks may be used to directly control the contact forces the robot exerts at the contact locations, e.g through the control of the center of mass dynamics. The possibility of defining control objectives with different priorities is an efficient way to deal with manipulation tasks while balancing [Farnioli et al., 2015]. These strategies can also avoid dangerous situations such as pushing too strongly against a rigid object, and reduce the possibility of breaking the contacts.

An efficient TSID algorithm for balancing and walking of humanoid robots is the so-called *momentum-based* control [Lee and Goswami, 2012], which often exploits *prioritized stack-of-tasks* [Mansard et al., 2009]. Several momentum-based control strategies have been successfully implemented in real applications [Stephens and Atkeson, 2010, Herzog et al., 2014, Koolen et al., 2016]. In such controllers, the primary control objective is the stabilization of the *centroidal* momentum [Orin et al., 2013]. Momentum control can be achieved by properly choosing the contact forces the robot exerts on the environment. The robot joint torques are then in charge of generating the desired forces. To get rid of the (eventual) actuation redundancy associated with momentum control, a lower priority task is usually added during the stabilization of the robot momentum, whose main role is the stabilization of the so-called robot *zero dynamics* [Isidori, 2013]. The latter objective is achieved by means of a *postural task*, which acts in the *null space* of the control of the robot's momentum [Righetti et al., 2011, Nakanish et al., 2007]. The contact forces are usually ensured to belong to a physically feasible domain by means of constrained quadratic optimization solvers [Wensing and Orin, 2013, Kuindersma et al., 2014, Hopkins et al., 2015]. Still, the resulting optimal control inputs may be discontinuous in certain cases. Another common drawback of momentum-based control strategies is that direct force feedback is often missing in the control action, thus potentially wasting important information for the stabilization of the system in presence of model uncertainties.

4.3 Control Strategies for Flying

Control architectures for aerial systems may be grouped depending on the type of aircraft. For what concerns fixed-wing aircrafts, modern control techniques include: pole-placement, linear quadratic regulators (LQR), reference tracking, model following, dynamic inversion, parabolic flight [D'Antonio and Monaco, 1993]. These techniques are based on linear approximations of the robot dynamics about an equilibrium point representing a steady-state condition [Bani Younes et al., 2012]. Despite the robustness properties of linearization based controllers [Roos et al., 2012], a drawback of these techniques is that they can only guarantee local stability of the associated nonlinear system. Also, the linearization of the robot dynamics requires a minimal parametrization of the rotation matrix representing the vehicle's orientation: minimal parametrizations of rotation matrices are undefined for some attitude configurations, and can introduce artificial singularities in the controlled system [Stuelpnagel, 1964].

The control of Vertical Take Off and Landing (VTOL) aircrafts is often addressed by neglecting the aerodynamic forces acting on the robot [Hua et al., 2013]. Linear control techniques such as Linear-Quadratic-Regulator (LQR) [Stone, 2004, Mokhtari et al., 2005] and robust control strategies have been widely applied to the control of VTOL [Civita et al., 2003, Luo et al., 2003]. As for the case of fixed-wing systems, a limitation of these techniques is that their stability results are guaranteed only in the local domain.

Nonlinear control theory has also been applied to aerial systems [Khalil, 2002]. A well-known nonlinear control method is the so-called *input-output feedback linearization* technique [Isidori, 1995]. The key point of the approach consists in making the thrust vector part of the system state, and defining an output function that can be differentiated until the system's shape allows one to linearize the robot translational dynamics [Hauser et al., 1992, Koo and Sastry, 1998]. However, the computation of higher order derivatives of the system dynamics and of the thrust rate of change may be problematic in practice.

Another nonlinear method for the control of VTOL is based on dynamic extension of the system. A *backstepping* procedure is then applied to prove the boundedness of tracking errors [Frazzoli et al., 2000, Mahony et al., 1999]. A third control strategy is based on nonlinear hierarchical controllers [Hua et al., 2013, Pflimlin et al., 2010, Marconi and Naldi, 2007]. They usually consist of a two-loops architecture composed of a *high-level* position controller, whose role is to determine the desired thrust that stabilizes the vehicle's velocity, and a *low-level* orientation controller, that regulates the thrust's direction. The high-level control problem often generates a desired rotation matrix, that is then achieved by the low-level controller [Frazzoli et al., 2000, Marconi and Naldi, 2007, 2008]. The desired rotation matrix is considered as a reference value for the stabilization of the vehicle's attitude. The stabilization of this matrix can be achieved via various techniques [Frazzoli et al., 2000, Marconi and Naldi, 2007].

To compensate for the limited interaction capablities of classical VTOL, a new branch of Robotics called Aerial Manipulation is endowing aerial systems with grippers and similar tools [Kondak et al., 2015, Pounds et al., 2011, Mellinger et al., 2011, Kamel et al., 2016], that are sometimes connected to the vehicle main body via robotic arms [Mellinger et al., 2011, Tognon et al., 2019, Kondak et al., 2014]. Physical interaction with the environment requires the aerial manipulator to precisely regulate both position of the contact point (possibly moving) and the amount of force that is exerted on it. Regarding motion control of aerial manipulators, different methods can be applied, e.g., full dynamic inversion [Yang and Lee, 2014], flatness-based [Tognon et al., 2017], and adaptive sliding mode [Kim et al., 2013]. On top of those, if the system is over actuated with respect to the desired task, a nullspace-based behavioral control [Baizid et al., 2017] or a task priority controller [Santamaria-Navarro et al., 2014] can be applied at the kinematic level exploiting the redundancy to achieve secondary tasks (e.g., obstacle avoidance, minimum energy consumption, etc.). In particular, the latter acts as a local motion planner providing the reference trajectory of each degree of freedom of the robot to the low level motion controller. The main limitation of this control architecture is the inability to consider additional hard constraints to be respected, e.g., input/state bounds, dynamics of the system which is not taken into account at the kinematic level, friction cone in case of interaction, etc. The mentioned constraints are very important because, if not respected, they might bring the whole system to instability in real scenarios.

From the interaction control side, one of the most common strategy is to use an admittance filter. In [Ryll et al., 2019] and [Suarez et al., 2018], such technique has been employed to control the interaction force in the case of a fully-actuated platform equipped with a rigid tool, and of an under-actuated platform equipped with a robotic arm, respectively. In [Rashad et al., 2019], a passivity-based controller has been proposed as well. However, in those works, the force-control is only *indirect*. In fact, the force is not directly measured but rather estimated using the robot dynamics. Since the method is strongly model based, it is thus prone to error in case of parameter uncertainties. Furthermore, if the system is affected by external disturbances, it might not be possible to discriminate them from the interaction forces. A first attempt in using a *direct* force feedback can be found in [Antonelli et al., 2016] where a force sensor has been attached to the end-effector. Nevertheless, this feedback is not used to precisely control the interaction force but is rather used in an impedance control framework to make the end-effector compliant.

Chapter 5

Motivations and Thesis Context

The Chapter describes how this research project fits in the landscape of instantaneous momentum-based and QP-based controllers for floating base systems. First, a certain number of open problems and challenges, not yet addressed by the state-of-the-art balancing and flight controllers, are presented. Secondly, we list the thesis contribution to the research field.

5.1 Open Problems and Challenges

Challenges on momentum-based control

The task-based control structure of momentum-based controllers renders the closed-loop system dynamics not trivial to analyze. While convergence to zero of the momentum error is often easy to prove, stability of the system's zero dynamics [Isidori, 2013] is challenging and usually is verified numerically. To the best of the author's knowledge, a proper stability analysis of momentum-based control strategies in the contexts of floating base systems is still missing. Furthermore, when momentum-based control algorithms are implemented in real applications, a long and tedious tuning of control gains is often required to achieve acceptable performances. Despite the large number of gain optimization procedures for classical dynamical systems (see, e.g., [Teshnehlab and Watanabe, 2002, Aphiratsakun and Parnichkun, 2009]) gain optimization techniques for floating base systems, and in particular in the field of humanoid robots, still need more investigations.

An assumption that limits the application of momentum-based controllers in a real scenario is that the robot is in contact with a rigid, static environment. From the modeling point of view, this results in neglecting the environment dynamics, i.e. the robot is subject to a set of purely *kinematic* constraints [Luca and Manes, 1994]. This assumption may be a limitation in case the robot is walking on debris or on soft ground. In this case, available solutions make use of adaptive or robust controllers [Lee and Goswami, 2012, Kim et al., 2007], but also dedicated control solutions have been concieved [Flayols et al., 2019]. There are also situations in which the environment dynamics may be known *a priori*, e.g. while interacting with a wheeled chair, balancing on a moving platform or even interacting with humans. This leads to the development of control strategies that try to stabilize both the robot and the contact dynamics [Luca and Manes, 1994, Anderson and Hodgins, 2010]. However, the applicability of momentum-based controllers to balancing on dynamic environment remains challenging.

There are also floating based systems for which the application of momentum controllers is not trivial and must be handled with care. For example, In case of robots endowed with series elastic actuators (SEA), the presence of joint elasticity complexifies the control design associated with the underlying robot, and extending momentum based controllers to the control of humanoid robots powered by series elastic actuators may not be straightforward.

Momentum-based, and more in general, QP-based control is a known technique in the field of humanoid robotics, but its applicability and effectiveness has never been demonstrated until now in the challenging field of aerial manipulators, where the physical interaction tasks are made very complex by, e.g., the absence of stabilizing contacts, limited propeller forces, inaccurate and time varying aerodynamics models, and mechanical vibrations. Thus, the applicability of QP-based controllers to the control of aerial vehicles still remains an open point of discussion.

Another unexplored area of research concerns the control of a flying humanoid robot. The control of this complex hybrid platform requires a careful analysis of the governing dynamics of the system during both flying and balancing tasks, and it is something that has never been attempted before with a humanoid robot.

An interesting research direction tries to exploit the *natural dynamics* of the system for improving performances and energy efficiency, e.g. for robot walking or running [Iida et al., 2005]. In particular, the effect of friction at all stages of the robot mechanisms and between the robot and the environment plays an important role for the stability of the controlled system [Miura et al., 2008, Panteley et al., 1998]. More generally, the *passivity-based* control strategies try to exploit the passivity properties of the overall system for regulation tasks [Li et al., 2012], and can also be extended for addressing tracking problems [Albu-Schäffer et al., 2007]. The application of these interesting concepts to momentum-based controllers is still something unexplored.

Limits of QP formulation

State-of-the-art QP-based controllers for floating base systems also suffer from specific limitations. Hereafter we enumerate three of them, that are going to be addressed in the remainder of the thesis:

1. The friction cone manifold is approximated with a set of linear inequalities in order to frame the optimisation problem as a QP;

2. The optimal solution may be discontinuous, e.g during contact switching or after sharp variations of the reference trajectory;

3. The closed-loop dynamics does not include any feedback term from the measured contact wrenches.

Limitation $1)$ does not usually have a strong impact on experimental activities, although it does remain an approximation of the static friction properties. Limitation $2)$ is often addressed by approximating the *continuity property* with a set of inequality constraints to be added to the optimization problem, but the effectiveness of this approach is often unsatisfactory from the experimental standpoint [Dafarra et al., 2018]. Limitation $3)$ is the most critical one, since FT sensor information is not used in the optimal control law that solves the QP problem, thus potentially wasting important feedback information at the control level. Let us observe that Limitation $3)$ may be attenuated when desired force tasks are added to the problem. For instance, if we assume to achieve a desired force f^d, then the force task can be addressed by adding equality constraints of the form $f = f^d$. At this point, one may attempt at using the FT measurements by replacing f^d with

$$f^* = f^d - K_I \int_0^t (f^m - f^d)dt,$$

where f^m is the measured force, and $f = f^*$ being the equality constraint. The main limitation of this approach is that this equality constraint may require f to violate the contact stability constraint. Putting the desired force as part of the cost function may be an option, but this alters the priorities that the force task has over the acceleration one.

5.2 Thesis Contribution

Part II: Momentum-Based Control Strategies for Balancing

– In **Chapter 7** we address stability issues of momentum controllers. Simulation results verify that the application of state-of-the-art momentum controllers for balancing may lead to unstable zero dynamics. A new control design with proven stability during one foot balancing is then designed and tested on the robot iCub while balancing on one and two feet.

- **Chapter 8** proposes automatic tuning of momentum control gains, thus simplifying the gain tuning procedure of momentum controllers. The gains are chosen to ensure local properties to the robot joint space dynamics.

- In **Chapter 9** we relax the assumption of robot balancing on rigid contacts, and we extend the momentum control framework described in Chapter 6 for balancing in highly dynamic environments. Simulation results are carried out on the robot iCub balancing on a seesaw board.

- **Chapter 10** implements a momentum-based controller for controlling robots with series elastic actuators. The main difference w.r.t. other control approaches is that the computation of higher level derivatives of the system dynamics is not required. Results are presented on a simulated iCub robot endowed with SEA.

- **Chapter 11** shows how to exploit the joints viscous and Coulomb friction in the momentum-based control design to improve the robot balancing performances. The control algorithm is tested on iCub performing highly dynamic movements while balancing.

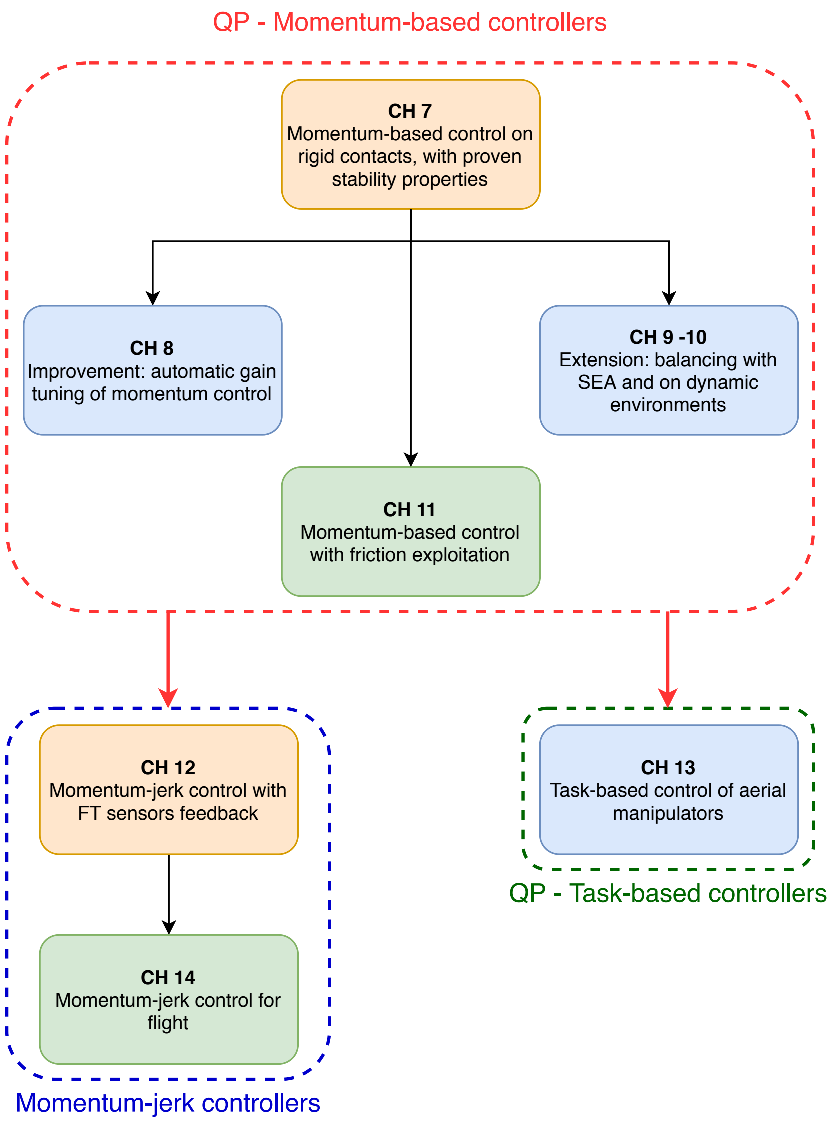
- **Chapter 12** presents a *momentum jerk* control framework that aims at addressing common limitations of QP-based momentum controllers, and also utilizes direct feedback from force/torque sensors.

Part III: Optimization-Based Control Strategies for Flying

- **Chapter 13** implements an optimization-based force control strategy for the aerial manipulator OTHex interacting with a rigid environment. Direct force feedback from a force/torque sensor mounted on the robot end effector is included in the control algorithm.

- **Chapter 14** describes a momentum based control architecture for the jet-powered humanoid robot iRonCub. Simulation results demonstrate that the robot can fly, balance and perform take off and landing maneuvers.

The block diagram depicted below briefly summarizes the interconnections between the following thesis chapters.


QP - Momentum-based controllers
CH 7
Momentum-based control on rigid contacts, with proven stability properties
CH 8
Improvement: automatic gain tuning of momentum control
CH 9 -10
Extension: balancing with SEA and on dynamic environments
CH 11
Momentum-based control with friction exploitation
CH 12
Momentum-jerk control with FT sensors feedback
CH 14
Momentum-jerk control for flight
Momentum-jerk controllers
CH 13
Task-based control of aerial manipulators
QP - Task-based controllers

Part II

Momentum-Based Control Strategies for Balancing

“Nothing happens until something moves”.

Albert Einstein

This part of the thesis implements momentum-based balancing controllers for humanoid robots. First, a state-of-the-art momentum-based control strategy for balancing is designed and implemented on the iCub humanoid robot. Then, we attempt at addressing some of the open challenges presented in Chapter 5 with the design of new momentum-based controllers for balancing, which are tested both in simulations and with experiments.

Chapter 6

Momentum-Based Control for Balancing on Rigid Contacts

The Chapter recalls a classical momentum-based control strategy for humanoid robots balancing when implemented as a two-layers stack-of-task. We assume that the objective is the control of the robot momentum and the stability of the system's *zero dynamics* [Isidori, 2013]. The controller generates reference torques that are then achieved by means of an inner joint-torque-control loop. The control framework presented in this Chapter will be used as a baseline in the next Chapters 7–12 for the design of momentum-based controllers for humanoids.

6.1 Momentum Dynamics and Control

6.1.1 Remarks on centroidal transformation

The control design proposed hereafter assumes that the robot dynamics is written in *centroidal coordinates*, hence with the mass matrix in its block diagonal form. In the remainder of the thesis, this particular expression of the equations of motion is preferred with respect to the general one (3.1), because it simplifies calculations and renders several formula more straightforward and easier to understand. We remark here that the centroidal transformation is not a necessary requirement in our control design, and the momentum-based controllers we shall propose in this Chapter and in the next ones can be derived also by assuming not block diagonal mass matrix. The mapping between the two formulations is given by Lemma 1.

6.1.2 Momentum dynamics

Thanks to the results presented in Lemma 1, the robot's momentum ${}^{\mathcal{G}[\mathcal{I}]}L \in \mathbb{R}^6$ in centroidal coordinates is given by ${}^{\mathcal{G}[\mathcal{I}]}L = M_{\mathcal{B}}\mathrm{v}_{\mathcal{B}}$. Hereafter we drop the superscript $\mathcal{G}[\mathcal{I}]$ for the momentum, i.e. ${}^{\mathcal{G}[\mathcal{I}]}L = L$, as in the remainder of the thesis we always refer to the momentum expressed in centroidal coordinates, unless otherwise specified. The rate-of-change of the robot momentum equals the net external wrench acting on the robot, which in the present case reduces to the contact wrenches f plus the gravity wrench. To control the robot momentum, we assume that f can be chosen at will. Note that given the particular form of the equations of motion, the first six rows of (3.4) correspond to the momentum dynamics:

$$\frac{d}{dt}(M_{\mathcal{B}}\mathrm{v}_{\mathcal{B}}) = J_{\mathcal{B}}^{\top} f - mge_3 = \dot{L}(f). \tag{6.1}$$

The primary control objective can then be defined as the stabilization of a desired robot momentum L^d. Let us define the momentum error as $\tilde{L} = L - L^d$. The control input f in Eq. (6.1) is chosen so as:

$$\dot{L} = \dot{L}^* := \dot{L}^d - K_P\tilde{L} - K_I I_{\tilde{L}} \tag{6.2a}$$
$$\dot{I}_{\tilde{L}} = \tilde{L}, \tag{6.2b}$$

where $K_P, \in \mathbb{R}^{6\times 6}$ is a symmetric, positive definite matrix, while a classical choice for the matrix K_I consists in [Herzog et al., 2014, Lee and Goswami, 2010]:

$$K_I = \begin{pmatrix} K_i^l & 0_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} \end{pmatrix}, \tag{6.3}$$

that is, the integral correction term at the angular momentum level is equal to zero, while the positive definite matrix $K_i^l \in \mathbb{R}^{3\times 3}$ is used for tuning the tracking of a desired center-of-mass position when the initial conditions of the integral in (6.2a) are properly set. The choice of matrix (6.3) is motivated by the expression of the angular momentum for rigid multi-body systems. More precisely, the angular momentum in centroidal coordinates is given by $L_\omega = {}^{\mathcal{G}[\mathcal{I}]}\mathbb{I}\, {}^{\mathcal{I}}\omega_o$ where ${}^{\mathcal{G}[\mathcal{I}]}\mathbb{I}(q) \in \mathbb{R}^{3\times 3}$ is the total robot's inertia expressed in the frame $\mathcal{G}[\mathcal{I}]$ and ${}^{\mathcal{I}}\omega_o \in \mathbb{R}^3$ is the so-called *locked* (or average) angular velocity [Orin et al., 2013]. When all the joint velocities are locked ($\dot{s} = 0$), ${}^{\mathcal{I}}\omega_o$ represents the angular velocity of the system, that behaves as a single rigid body. However, in general ${}^{\mathcal{I}}\omega_o$ is not associated with the derivative of a rotation matrix ${}^{\mathcal{I}}R_o$ somehow representing the overall robot orientation in space. There are precise conditions for the existence of such matrix, that are not guaranteed to be always satisfied for the case under study [Saccon et al., 2017]. Consequently, it does not necessarily exist an analytical expression for the integral of the angular momentum $I_\omega := \dot{I}_\omega = L_\omega$, and the term is often omitted from the desired momentum dynamics (6.2a).

6.1.3 Momentum control for one foot balancing

In case only one robot link has a constant position and orientation with respect to the inertial frame, the holonomic constraint is of the form $c(q) =$ constant, with $c(q) \in \mathbb{R}^3 \times SO(3)$. Without loss of generality, it is assumed that the only constrained frame is that between the environment and one of the robot feet. Consequently, one has:

$$\sum_{k=1}^{n_c} J_{\mathcal{C}_k}^\top f_k = J^\top(q) f,$$

where $J(q) \in \mathbb{R}^{6\times n+6}$ is the Jacobian of a frame attached to the foot sole in contact with the environment, and the contact wrench is $f \in \mathbb{R}^6$. Eq. (6.1) implies that the contact wrench satisfying Eq.(6.2a) can be generated as follows:

$$f^* = J_{\mathcal{B}}^{-\top} \left(\dot{L}^* + mge_3 \right). \tag{6.4}$$

6.1.4 Momentum control for two feet balancing

Let us now assume that several robot links have a constant position and orientation with respect to the inertial frame. Without loss of generality [1], it is assumed that the constraints are between the environment and the robot feet. Consequently, f is now a twelve dimensional vector, and composed of the wrenches $f_L,\ f_R \in \mathbb{R}^6$ between the floor and left and right foot, respectively. Hence, $f = \left[f_L, f_R\right] \in \mathbb{R}^{12}$. Also, let $J_L,\ J_R \in \mathbb{R}^{6\times n+6}$ denote the Jacobian of two frames associated with the contact locations of the left and right foot. Then, $J = \left[J_L^\top, J_R^\top\right]^\top \in \mathbb{R}^{12\times n+6}$. By assuming that the contact wrenches can still be used as control input in the dynamics of the robot's momentum $\dot{L}$, one is left with a six-dimensional redundancy of the contact wrenches to impose $\dot{L}(f) = \dot{L}^*$. More specifically, we parametrize the set of solutions f^* to (6.1) as:

$$f^* = f_1 + N_{\mathcal{B}} f_0 \tag{6.5}$$

with $f_1 = J_{\mathcal{B}}^{\top\dagger} \left(\dot{L}^* + mge_3 \right)$, $J_{\mathcal{B}}^{\top\dagger}$ the Moore-Penrose pseudoinverse of $J_{\mathcal{B}}^\top$ and $N_{\mathcal{B}} = 1_{6n_c} - J_{\mathcal{B}}^{\top\dagger} J_{\mathcal{B}}^\top \in \mathbb{R}^{6n_c \times 6n_c}$ the projector into the null space of $J_{\mathcal{B}}^\top$. $f_0 \in \mathbb{R}^{6n_c}$ is the wrench redundancy that does not influence $\dot{L}(f) = \dot{L}^*$.

[1]We assume that the number of constraints is always lower than the number of degrees of freedom of the system. The analysis of an overconstrained system is beyond the scope of this thesis.

6.2 Choice of the Joint Torques

The joint torques can be used to instantaneously realize the desired contact wrenches given by (6.4) and (6.5). When balancing on rigid contacts, we write explicitly the system accelerations $\dot{\nu}$ from the equations of motion (3.4) and we substitute them in the kinematic constraints (3.7), which yields the following relation between contact wrenches and joint torques:

$$JM^{-1}(J^\top f - h) + \Lambda\tau + \dot{J}\nu = 0,$$

where in view of the centroidal transformation $\Lambda = J_s M_s^{-1} \in \mathbb{R}^{6\times n}$. To achieve $f = f^*$, we choose the joint torques as follows:

$$\tau = \Lambda^\dagger(JM^{-1}(h - J^\top f^*) - \dot{J}\nu) + N_\Lambda \tau_0, \tag{6.6}$$

with $N_\Lambda = 1_n - \Lambda^\dagger\Lambda \ \in \mathbb{R}^{n\times n}$ the null-space projector of Λ and $\tau_0 \in \mathbb{R}^n$ is a free variable which accounts for the joint torques redundancy.

6.2.1 Stability of the zero dynamics

The stability of the zero dynamics is usually attempted by means of the so called *postural task*, which exploits the free variable τ_0. A classical state-of-the-art choice for the postural task consists in:

$$\tau_0 = h_s - J_s^\top f^* + M_s \ddot{s}^d - K_P^s(s - s^d) - K_D^s(\dot{s} - \dot{s}^d), \tag{6.7}$$

where $K_P^s,\ K_D^s \in \mathbb{R}^{n\times n}$ are symmetric and positive definite matrices, s^d a desired (and eventually time varying) robot posture, and the term $h_s - J_s^\top f^*$ compensates for the nonlinear effects and the external wrenches acting on the joint space of the system [2]. In case of two feet balancing, an interesting property of the closed loop system (3.4)–(6.5)–(6.6)–(6.7) is recalled in the following Lemma.

Lemma 2 *Assume that Λ is full row rank. Then, the closed loop joint space dynamics $\ddot{s}$ does not depend upon the wrench redundancy f_0.*

The proof is in Appendix A. This result is a consequence of the postural control choice (6.7) and it is of some interest: it means that the closed loop joint dynamics depends on the total rate-of-change of the momentum, i.e. $\dot{L}$, but not on the different forces generating it. Hence, any choice of the redundancy f_0 does not influence the joint dynamics $\ddot{s}$, and we can exploit it to minimize the joint torques norm.

[2]It is assumed that at every instant $f = f^*$ thanks to (6.6)

6.3 Quadratic Programming Formulation

The input torques that are in charge of stabilizing both a desired robot momentum L^d and the associated zero dynamics are given by

$$\tau^* = \Lambda^\dagger (JM^{-1}(h - J^\top f^*) - \dot{J}\nu) + N_\Lambda \tau_0 \tag{6.8a}$$

$$\tau_0 = h_s - J_s^\top f^* + M_s \ddot{s}^d - K_P^s(s - s^d) - K_D^s(\dot{s} - \dot{s}^d), \tag{6.8b}$$

with f^* as in Eq. (6.4) in case of one foot balancing, and Eq. (6.5) in the two feet balancing scenario. In the language of *Optimization Theory*, the above control objectives are usually formulated as a Quadratic Programming (QP) problem. The advantage of QP formulation is twofold: first, it allows to solve online in a numerically efficient way problem (6.8). Secondly, it allows to include hardware constraints in the optimization, such as contact stability constraints.

6.3.1 QP formulation for one foot balancing

The optimization problem in case of one foot balancing can be written as follows:

$$f^* = \underset{f}{\operatorname{argmin}} |\dot{L}(f) - \dot{L}^*|^2 \tag{6.9a}$$

$$s.t.$$

$$C_f f < b_f \tag{6.9b}$$

$$\tau^*(f) = \underset{\tau}{\operatorname{argmin}} |\tau(f) - \tau_0(f)|^2 \tag{6.9c}$$

$$s.t.$$

$$\dot{J}\nu + J\dot{\nu} = 0 \tag{6.9d}$$

$$\dot{\nu} = M^{-1}(B\tau + J^\top f - h) \tag{6.9e}$$

$$\tau_0 = h_s - J_s^\top f + M_s \ddot{s}^d - K_P^s(s - s^d) - K_D^s(\dot{s} - \dot{s}^d). \tag{6.9f}$$

The problem is designed as a cascade of QP instances. The first minimization (6.9a) aims at finding the optimal wrench f^* that stabilizes the momentum dynamics, while the second problem (6.9c) optimizes the joint torques to achieve $f = f^*$ through constraints (6.9d)–(6.9e), and addresses the postural task via (6.9f). The inequalities (6.9b) account for the contact stability constraints: they ensure that the desired contact wrench belongs to the associated friction cone, which is approximated with linear inequalities, and they apply constraints on the positivity of the vertical force and on the contact moments.

6.3.2 QP formulation for two feet balancing

In case of two feet balancing, the optimization problem has a slightly different formulation, but it can still be written as a cascade of QP instances:

$$
\begin{aligned}
f^* = \; & \underset{f}{\operatorname{argmin}} \, |\tau^*(f)|^2 && \text{(6.10a)}\\
& s.t. \\
& \quad C_f f < b_f && \text{(6.10b)}\\
& \quad \dot{L}(f) = \dot{L}^* && \text{(6.10c)}\\
& \quad \tau^*(f) = \underset{\tau}{\operatorname{argmin}} \, |\tau(f) - \tau_0(f)|^2 && \text{(6.10d)}\\
& \quad\quad s.t. \\
& \quad\quad\quad \dot{J}\nu + J\dot{\nu} = 0 && \text{(6.10e)}\\
& \quad\quad\quad \dot{\nu} = M^{-1}(B\tau + J^\top f - h) && \text{(6.10f)}\\
& \quad\quad\quad \tau_0 = h_s - J_s^\top f + M_s \ddot{s}^d - K_P^s(s - s^d) - K_D^s(\dot{s} - \dot{s}^d), && \text{(6.10g)}
\end{aligned}
$$

where the optimization problem (6.10a) now takes into account the wrench redundancy f_0, which is exploited to minimize the norm of the joint torques, while the momentum task becomes a constraint.

In the remainder of the thesis, the QP problems (6.9)–(6.10) will be solved online with qpOASES solver [Ferreau et al., 2014]. In the balancing simulations presented next, the robot also performs *contacts switching*, e.g. it changes run-time the number of feet in contacts. The control architecture then switches from one foot to two feet balancing, discontinuously by means of a state machine. The change of state is triggered with direct measurement of the contact wrench at each foot through force/torque sensors mounted on the robot.

Chapter 7

Stability Analysis of Momentum-Based Controllers

Momentum-based strategies have proven their effectiveness for controlling humanoid robots balancing, but a proper stability analysis of these controllers is still missing. In this Chapter, we first numerically show that the application of state-of-the-art momentum-based control strategies may lead to unstable zero dynamics. Secondly, we propose simple modifications to the control architecture that avoid instabilities at the zero-dynamics level. Asymptotic stability of the closed loop system is shown by means of a Lyapunov analysis on the linearized system's joint space. The theoretical results are validated with both simulations and experiments on the iCub humanoid robot.

7.1 Numerical Evidence of Unstable Zero Dynamics

To show that state-of-the-art momentum-based control strategies may lead to unstable zero dynamics, we control the linear momentum of the robot so as to follow a desired center of mass trajectory, i.e. a sinusoidal reference along the y coordinate with amplitude 0.05 m and frequency 0.3 Hz. Figure 7.1 visually describes the robot motion pattern. The joints position reference s^d is set equal to its initial value, i.e. $s^d = s(0)$.

7.1.1 Balancing on one foot in the custom simulation setup

We present simulation results obtained by applying the control law (6.4)–(6.6)–(6.7) with the gain matrix K_i as in (6.3). It is assumed that the left foot is attached to ground, and no other external wrench applies to the robot. Figure 7.2 shows

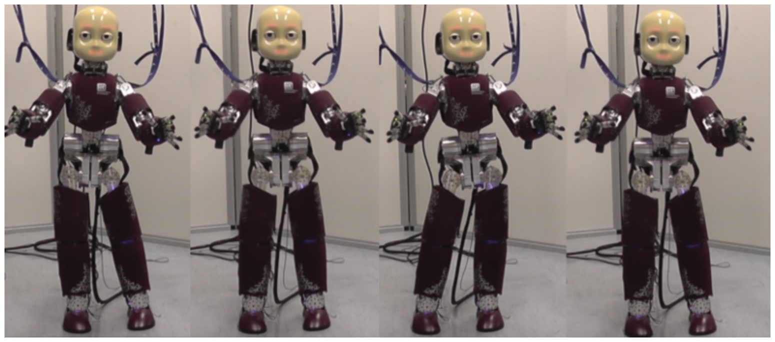


Fig. 7.1 The robot motion pattern for testing stability of the momentum-based controller.

typical simulation results of the convergence to zero of the robot's momentum error, thus meaning that the wrench (6.4) ensures the stabilization of $\tilde{L}$ towards zero. Figure 7.3, instead, depicts the joint position error norm $|s - s^d|$. This figure shows that the norm of the joint angles increases while the robot's momentum is kept equal to zero, which is a classical behavior of unstable zero dynamics.

7.1.2 Balancing on two feet in the Gazebo simulation setup

Tests on the stability of the zero dynamics have been carried out also in the case the robot stands on both feet. More precisely, the contact wrench f is now a twelve dimensional vector, and it is composed of the contact wrenches $f_L, f_R \in \mathbb{R}^6$ between the floor and left and right foot, respectively. The control algorithm running in this case is represented by the equations (6.5)–(6.6)–(6.7). Figure 7.4 depicts a typical behavior of the joint position error norm $|s - s^d|$ when the above control algorithm is applied: it is clear that the instability of the zero dynamics is observed also in the case where the robot stands on two feet.

7.2 Minimum Coordinates Representation

Recall that the configuration space of the robot evolves in a group of *dimension*[1] $n + 6$. Hence, besides pathological cases, when the system is subject to a set of holonomic constraints of dimension k, the configuration space shrinks into a space of dimension $n + 6 - k$. The stability analysis of the constrained system may then require to determine the minimum set of coordinates that characterize the evolution of the constrained system. This operation is, in general, far from obvious

[1]With group dimension we here mean the dimension of the associated algebra $\mathbb{V}$.

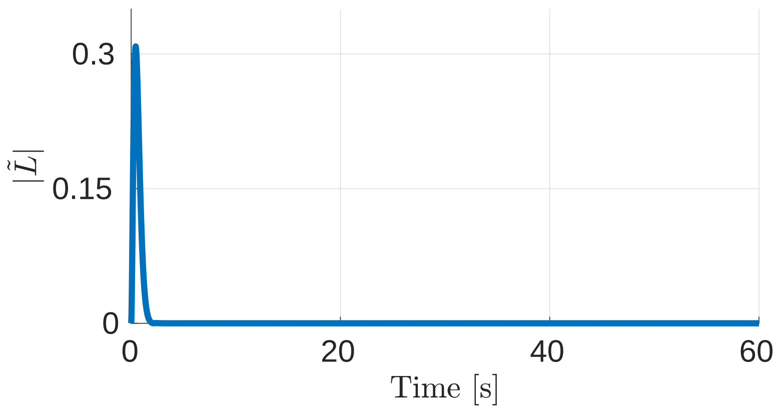


Fig. 7.2 Time evolution of the robot's momentum error when standing on one foot and the control law (6.4)–(6.6)–(6.7) is applied. Simulation run with the custom environment.

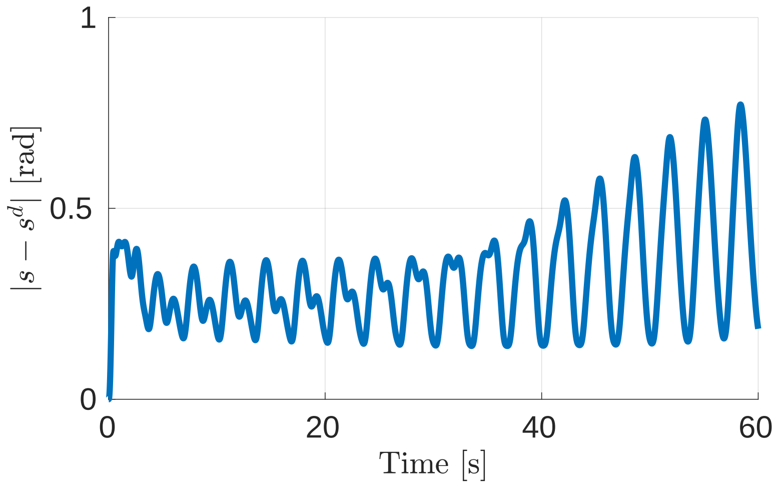


Fig. 7.3 Time evolution of the norm of the position error $|s-s^d|$ when the robot is standing on one foot and when the control law (6.4)–(6.6)–(6.7) is applied. Simulation run with the custom environment.

because of the topology of the group $\mathbb{Q}$. In light of the above, for the analytical demonstration the closed-loop system stability we make the following assumption.

Assumption 1 *Only one frame associated with a robot link has a constant position-and-orientation with respect to the inertial frame.*

Thanks to Assumption 1, one gets rid of the topology related problems of $\mathbb{Q}$ by relating the base frame $\mathcal{B}$ and the constrained frame. In this case, the minimum set

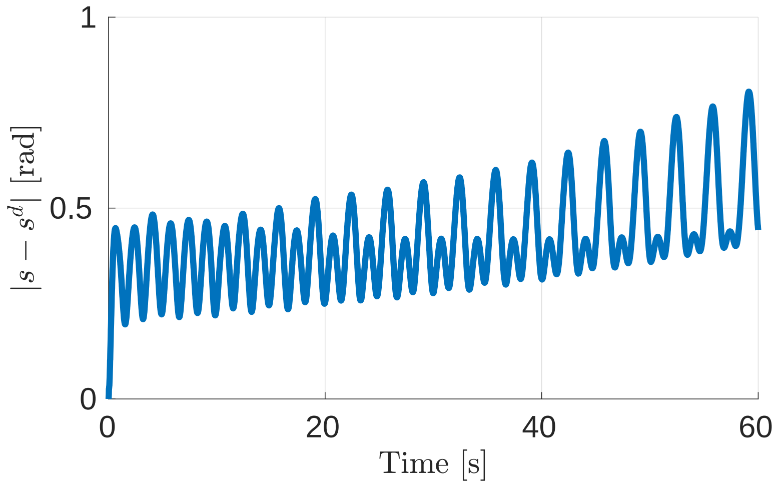


Fig. 7.4 Time evolution of the norm of the position error $|s-s^d|$ when the robot is standing on two feet and when the control law (6.5)–(6.6)–(6.7) is applied. Simulation run with the Gazebo environment.

of coordinates belongs to $\mathbb{R}^n$ and can be chosen as the joint variables s.

7.3 Control Design with Proven Stability

To circumvent the problems related to the stability of the zero dynamics, we propose a modification of the control law (6.4)–(6.6)–(6.7) that allows us to show stable zero dynamics of the constrained, closed loop system. The following results exploit the so-called *centroidal-momentum-matrix* $J_G(q) \in \mathbb{R}^{6 \times n+6}$, namely the mapping between the robot velocity ν and the robot's momentum: $L = J_G \nu$.

Remark 1 *The stability analysis is carried out assuming that only one link of the robot has a constant position and orientation w.r.t. the inertial reference frame. Extension to the case of multiple contacts is not straightforward, and it will not be considered in this thesis. On the other hand, stability of closed loop system dynamics is verified numerically when the modified control framework is applied to the two feet balancing case.*

Writing the system dynamics in *centroidal coordinates* reduces the number of calculations required for performing the stability analysis. In fact, thanks to the results of Lemma 1 one has $J_G(q) = \begin{bmatrix} M_{\mathcal{B}} & 0_{6 \times n} \end{bmatrix}$. When Assumption 1 (one foot balancing) is satisfied, the kinematic constraint Eq. (3.6) allows us to write the robot's momentum linearly w.r.t. the robot's joint velocity, i.e. $L = \bar{J}_G(s)\dot{s}$, where

$\bar{J}_G(s) \in \mathbb{R}^{6\times n}$ is:

$$\bar{J}_G(s) := \begin{bmatrix} J_G^l(s) \\ J_G^\omega(s) \end{bmatrix} = -M_\mathcal{B} J_\mathcal{B}^{-1} J_s, \tag{7.1}$$

and $J_G^l(s), J_G^\omega(s) \in \mathbb{R}^{3\times n}$. In light of the above, the following result holds.

Lemma 3 *Assume that the robot possesses more than six degrees of freedom, that is, $n \geq 6$. In addition, assume also that $L^d = 0$. Let*

$$(s, \dot{s}) = (s^d, 0) \tag{7.2}$$

denote the equilibrium point associated with the constrained, closed loop system and assume that the matrix $\Lambda = J_s(q)M_s^{-1}(q)$ from the joints torques equation (6.6) *is full row rank in a neighborhood of* (7.2)*. Apply the control laws* (6.4)–(6.6)–(6.7) *with the following modifications:*

$$\dot{I}_{\tilde{L}} = \begin{bmatrix} J_G^l(s) \\ J_G^\omega(s^d) \end{bmatrix} \dot{s} \tag{7.3}$$

$$K_I > 0, \quad K_P^s = \overline{K}_P^s N_\Lambda M_s, \quad K_D^s = \overline{K}_D^s N_\Lambda M_s, \tag{7.4}$$

where $\overline{K}_P^s, \overline{K}_D^s \in \mathbb{R}^{n\times n}$ are two positive definite matrices. Then, the equilibrium point (7.2) *of the constrained, closed loop dynamics is asymptotically stable.*

The proof is in Appendix B. Lemma 3 shows that the asymptotic stability of the equilibrium point (7.2) of the constrained, closed loop dynamics can be ensured by modifying the momentum integral correction terms, and by modifying the gains of the postural task. As a consequence, the asymptotic stability of the equilibrium point (7.2) implies that the zero dynamics are locally asymptotically stable.

The fact that the gain matrix K_I must be positive definite conveys the necessity of closing the control loop with *orientation* terms at the angular momentum level. In fact, some authors intuitively close the angular momentum loop by using the orientation of the robot's torso [Ott et al., 2011].

The proof of Lemma 3 exploits the fact that the minimum coordinates of the robot configuration space when Assumption 1 holds is given by the joint angles s. The analysis focuses on the closed loop dynamics of the form $\ddot{s} = f(s, \dot{s})$ which is then linearized around the equilibrium point (7.2). By means of a Lyapunov analysis, one shows that the equilibrium point is asymptotically stable. One of the main technical difficulties when linearizing the equation $\ddot{s} = f(s, \dot{s})$ comes from the fact that the closed loop dynamics depends on the integral of the robot momentum, i.e.

$$I_{\tilde{L}}(t) = I_{\tilde{L}}(0) + \int_0^t \begin{bmatrix} J_G^l(s(t)) \\ J_G^\omega(s^d) \end{bmatrix} \dot{s}(t)dt \tag{7.5}$$

The partial derivative of $I_{\tilde{L}}(t)$ w.r.t. the state $(s, \dot{s})$ is, in general, not obvious because the matrix $J^l_G(s)$ may not be integrable. Let us observe, however, that the first three rows of $\dot{I}_{\tilde{L}}(t)$ correspond to the velocity of the center-of-mass times the robot's mass when expressed in terms of the minimal coordinates s, i.e. $J^l_G(s)\dot{s} = mv_c$, with $v_c \in \mathbb{R}^3$ the velocity of the robot's center-of-mass. Clearly, this means that $J^l_G(s)$ is integrable, and that

$$\partial_s I_{\tilde{L}} = \begin{bmatrix} J^l_G(s) \\ J^\omega_G(s^d) \end{bmatrix} \qquad \partial_{\dot{s}} I_{\tilde{L}} = 0. \tag{7.6}$$

Remark 2 *Lemma 3 suggests that applying the control laws* (6.4)–(6.6)–(6.7) *with the control gains as* (7.4) *can still guarantee stability and convergence of the equilibrium point. First, observe that the main difference between the variable $I_{\tilde{L}}$ governed by the two expressions* (6.2b) *and* (7.3) *resides only in the last three equations. Then, more importantly, note that the momentum L when Assumption 1 holds can be expressed as follows: $L = \bar{J}_G(s^d)\dot{s} + \mathrm{o}(s - s^d, \dot{s})$ which implies that*

$$\int_0^t L dt = \bar{J}_G(s^d)(s - s^d) + \int_0^t \mathrm{o}(s - s^d, \dot{s}) dt.$$

As a consequence, the linear approximations of the integrals $I_{\tilde{L}}$ governed by (6.2b) and (7.3) coincide when

$$\lim_{(s-s^d,\dot{s})\to 0} \frac{|\int_0^t o(s - s^d, \dot{s}) dt|}{|s - s^d, \dot{s}|} = 0 \tag{7.7}$$

Under the above Assumption (7.7), the linear approximation of the control laws (6.4)–(6.6)–(6.7) when evaluated with (6.2b) and (7.3) coincide, and stability and convergence of the equilibrium point $(s^d, 0)$ can still be proven.

7.4 Simulations and Experimental Results

We show simulation and experimental results obtained by applying the control laws (6.4)–(6.6)–(6.7)–(7.4) (one foot balancing) and (6.5)–(6.6)–(6.7)–(7.4) (two feet balancing). To show the improvements of the control modification in Lemma 3, we apply the same reference signal of Section 7.1, which revealed unstable zero dynamics. Hence, the desired linear momentum is chosen so as to follow a sinusoidal reference for the center of mass. Also, control gains are kept equal to those used for the simulations presented in Section 7.1.

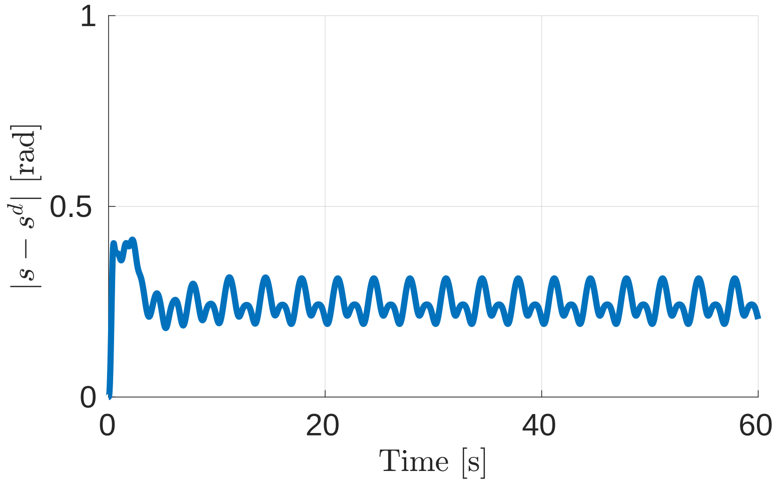


Fig. 7.5 Time evolution of the norm of the position error $|s - s^d|$ when the robot is standing on one foot and when the control law (6.4)–(6.6)–(6.7)–(7.3)–(7.4) is applied. Simulation run with the custom environment.

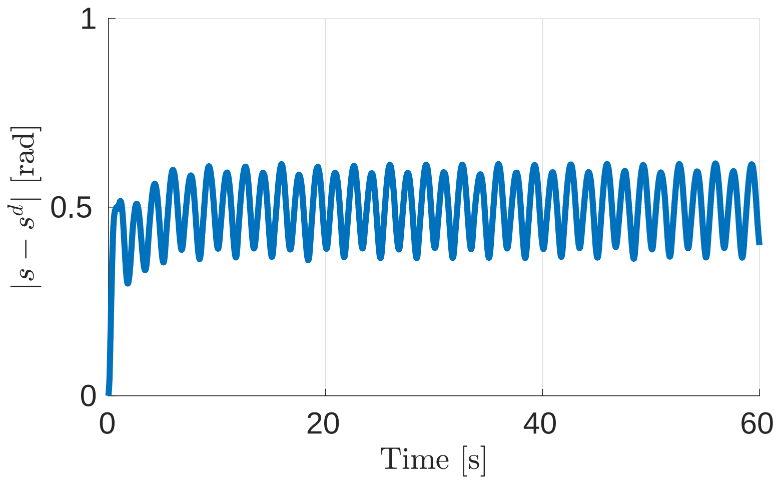


Fig. 7.6 Time evolution of the norm of the position error $|s - s^d|$ when the robot is standing on two feet and when the control law (6.5)–(6.6)–(6.7)–(7.3)–(7.4) is applied. Simulation run with the Gazebo environment.

7.4.1 Simulation results

Figures 7.5 – 7.6 show the norm of the joint errors $|s - s^d|$ when the robot stands on either one or two feet, respectively. Experiments on two feet have been performed to verify the robustness of the new control architecture. The simulations are performed both with custom and Gazebo environment. As expected, the zero dynamics is now stable, and no divergent behavior of the robot joints is observed.

7.4.2 Results on the iCub humanoid robot

We then went one step further and implemented the control algorithm (6.10) with the modification presented in Lemma 3 on the real humanoid robot. The robotic platform used for testing is the iCub. For the purpose of the proposed control law, iCub is endowed with 23 degrees of freedom. A low level torque control loop, running at 1 kHz, is responsible for stabilizing any desired torque reference signal.

Figures 7.7 – 7.8 show the joint position error $|s - s^d|$ and the center of mass error. The high center of mass error along the direction of the movement is mainly due to the presence of unmodeled friction at the joints level, that on the real robot induces a phase delay in following the commanded trajectory. This problem is tackled later in the thesis in a dedicated Chapter. Though the center of mass does not converge to the desired value, all signals are bounded, and the control modification presented in Lemma 3 does not pose any barrier for the implementation of the control algorithm (6.10) on a real platform.

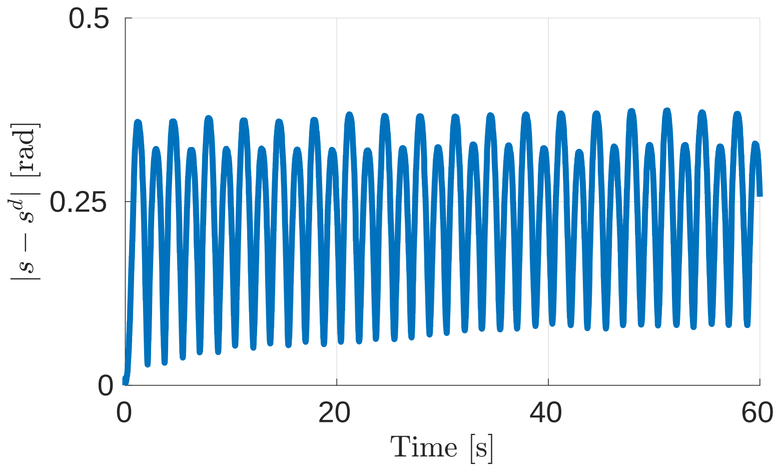


Fig. 7.7 Time evolution of the norm of the position error $|s - s^d|$ when the robot is standing on two feet and when the control law (6.5)–(6.6)–(6.7)–(7.3)–(7.4) is applied. Experiment run on the humanoid robot iCub.

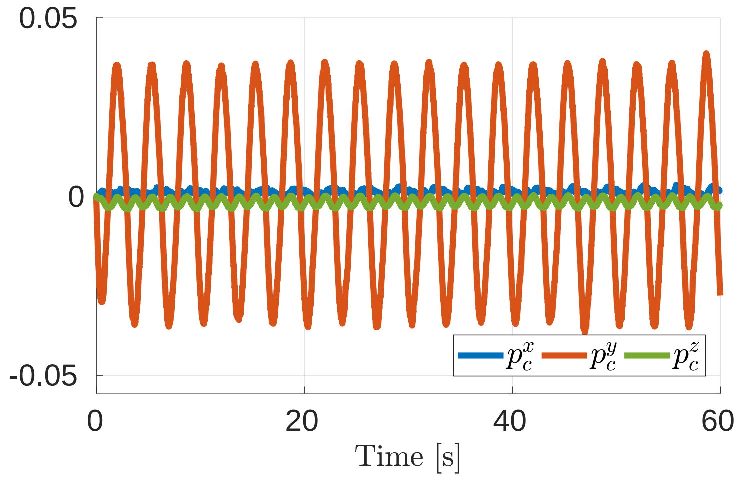


Fig. 7.8 Time evolution of robot center-of-mass errors $\tilde{p}_c = p_c - p_c^d$ when the robot is standing on two feet and when the control law (6.5)–(6.6)–(6.7)–(7.3)–(7.4) is applied. Experiment run on the humanoid robot iCub.

Chapter 8

Automatic Gain Tuning of Momentum-Based Controllers

The momentum-based controller with the modifications proposed in Chapter 7 ensures stability properties to the closed-loop system dynamics. However, a proper choice of control gains is still required in order to achieve a desired robot behavior. For complex multi-body systems gain tuning is usually a long and tedious procedure, due to the high number of tunable parameters and the nonlinearity of the system that combines the effect of different gains.

In this Chapter we propose a technique for automatic gain tuning of a momentum-based balancing controller for humanoid robots. As already done for the stability analysis of Chapter 7, the closed-loop, constrained joint space dynamics is linearized and the controller's gains are chosen so as to obtain desired properties of the linearized system. Symmetry and positive definiteness constraints of gain matrices are enforced by proposing a tracker for symmetric positive definite matrices. Simulation results are carried out on the humanoid robot iCub.

8.1 The Gain Tuning Procedure

Our goal is to impose desired local properties to the joint space dynamics $\ddot{s}$. The choice of focusing on the joint dynamics over other output functions reflects the aim of choosing stiffness and damping at the joint level via postural control, without perturbing the task hierarchy of momentum control and stability of the associated zero dynamics.

8.1.1 Joint space linearization

The control torques obtained by solving the QP optimization problems (6.9)–(6.10) depend only on the system state, i.e. $\tau = \tau(q, \nu)$. Since it is assumed that the robot stands on (at least) one foot, one can express the system state in terms of the joint position and velocity, i.e. $(q, \nu) = (q(s), \nu(s, \dot{s}))$. Then, the joint space dynamics depends only on joint position and velocity, i.e. $\ddot{s} = \ddot{s}(s, \dot{s})$ and we can linearize this dynamics about an equilibrium point $(s^d, 0)$. The process of finding the linearized joint dynamics is the same described in Appendix B, which yields:

$$\ddot{s} = -Q_1(s - s^d) - Q_2\dot{s}$$

where $Q_1, Q_2 \in \mathbb{R}^{n \times n}$ are given by

$$Q_1 = C_1(s^d)K_I C_2(s^d) + C_3(s^d)K_P^s C_4(s^d) \tag{8.1a}$$
$$Q_2 = C_1(s^d)K_P C_2(s^d) + C_3(s^d)K_D^s C_4(s^d) \tag{8.1b}$$

and $C_1 = M_s^{-1}\Lambda^\dagger J_\mathcal{B} M_\mathcal{B}^{-1}$, $C_2 = M_\mathcal{B} J_\mathcal{B}^\dagger J_s$, $C_3 = M_s^{-1}N_\Lambda$ and $C_4 = N_\Lambda M_s$. Now, let $x \in \mathbb{R}^{2n}$ be defined as follows $x := \begin{bmatrix} x_1^\top & x_2^\top \end{bmatrix}^\top = \begin{bmatrix} s^\top - s^{d\top} & \dot{s}^\top \end{bmatrix}^\top$. The linearized joint space dynamics around an equilibrium point $(s^d, 0)$ is given by

$$\dot{x} = \begin{bmatrix} \partial_s \dot{x}_1 & \partial_{\dot{s}} \dot{x}_1 \\ \partial_s \dot{x}_2 & \partial_{\dot{s}} \dot{x}_2 \end{bmatrix} x = \begin{bmatrix} 0_n & 1_n \\ -Q_1(K_I, K_P^s) & -Q_2(K_P, K_D^s) \end{bmatrix} x = Ax. \tag{8.2}$$

8.1.2 Gains optimization

The gain tuning optimization problem we attempt to solve may be stated as:

$$K_P^{s*}, K_D^{s*}, K_I^*, K_P^* = \underset{K_P^s, K_D^s, K_I, K_P}{\text{argmin}} \; |A(K_P^s, K_D^s, K_I, K_P) - A^d|^2 \tag{8.3}$$
$$s.t.$$
$$K_I = K_I^\top > 0 \tag{8.3a}$$
$$K_P = K_P^\top > 0 \tag{8.3b}$$
$$K_P^s = K_P^{s\top} > 0 \tag{8.3c}$$
$$K_D^s = K_D^{s\top} > 0 \tag{8.3d}$$

where A^d is the desired state matrix of the following form: $A^d = \begin{bmatrix} 0_n & 1_n \\ -Q_1^d & -Q_2^d \end{bmatrix}$, with $Q_1^d, Q_2^d \in \mathbb{R}^{n \times n}$ the desired stiffness and damping matrices. The optimization problem (8.3) can be solved with any nonlinear available optimizer. Yet, finding numerical solutions to the optimization problem may be time consuming, which may forbid the on-line use of the optimizer when the desired stiffness and damping are time-varying. In this case, the solutions to (8.3) may also become discontinuous at some time instants. We propose below a method for solving on-line the problem (8.3) that provide continuous solutions for the control gains.

8.1.3 Solution to the unconstrained problem

Assume that the positivity and symmetry constraints (8.3a)–(8.3d) do not hold. When the robot stands on one foot, intuition would suggest that the joint space dynamics can be imposed at will, i.e. there always exist control gains such that the matrices Q_1^d, Q_2^d can be chosen arbitrarily. This Section shows, however, that this is not possible because of the two strict stack-of-task control strategy defined in Chapter 6. To show this property, we prove that there exist some matrices Q_1^d such that no choice of the control gains renders $Q_1^d = Q_1(K_I, K_P^s)$ satisfied.

Now, if the constraints (8.3a)–(8.3d) do not hold, then the optimization problem (8.3) can be rewritten as

$$y_1 = \Psi \begin{bmatrix} k_i \\ k_p^s \end{bmatrix} \tag{8.4a}$$

$$\Psi = \begin{bmatrix} [C_2^\top \otimes C_1] & [C_4^\top \otimes C_3] \end{bmatrix} \in \mathbb{R}^{n^2 \times (36+n^2)} \tag{8.4b}$$

where $y_1, k_p^s \in \mathbb{R}^{n^2}$, $k_i \in \mathbb{R}^{36}$ are the vectorization of matrices Q_1, K_P^s and K_I obtained by reordering their columns into a single column vector, and $\otimes$ the Kronecker product. Then, the following result holds.

Lemma 4 *Recall the contact Jacobian matrix J, and assume that rank$(J) = 6n_c$, and that $n > 6n_c$. Then the matrix Ψ is not full row rank, i.e. rank$(\Psi) < n^2$.*

The proof of the above Lemma is in the Appendix C. As a consequence of Lemma 4, there exist some desired matrices Q_1^d, i.e. y_1, such that no control gain implies the exact solution to $Q_1^d = Q_1(K_I, K_P^s)$. On the other hand, the least square solution to the problem (8.4) is given by

$$\begin{bmatrix} k_i^* \\ k_p^{s*} \end{bmatrix} = \Psi^\dagger y_1 \tag{8.5}$$

Clearly, the gains K_I^*, K_P^{s*} obtained by the above solution do not in general satisfy the symmetry and positive definiteness constraints. A similar procedure can be applied to find the gains K_P^*, K_D^{s*} that do not in general satisfy (8.3a)–(8.3d), but solve in the least square sense $Q_2^d = Q_2(K_D^{s*}, K_P^*)$.

8.1.4 Enforcing symmetry and positive definiteness constraints

In the previous Section, we solved the problem (8.3) by assuming that the constraints (8.3a)–(8.3d) do not hold. Hence, we are now given with a set of gains $K_P^{s*}, K_D^{s*}, K_I^*, K_P^*$ that may not be symmetric and positive definite.

Define $K^* := \{K_P^{s*}, K_D^{s*}, K_I^*, K_P^*\}$ a matrix of proper dimension. Then, to enforce symmetry and positive definiteness constraints, we solve (on-line) a second optimization problem for each of the unconstrained optimal gain. More precisely, the problem we solve follows:

$$
\begin{aligned}
O^*, D^* = \; & \underset{(O,D)}{\operatorname{argmin}} |K^* - X(O,D)|^2 \\
\text{s.t.} & \\
& OO^\top = 1 \\
& D \text{ diagonal matrix.}
\end{aligned} \tag{8.6}
$$

with $X := O^\top \exp(D) O$, O an orthogonal matrix, and D a diagonal matrix. The solution to the problem (8.6) are the matrices O^*, D^*. Therefore, the constrained optimized gain matrix is given by:

$$
X^* = O^{*\top} \exp(D^*) O^*. \tag{8.7}
$$

Clearly, at this point we just moved the problem from solving the optimization problem (8.3) with the constraints (8.3a)–(8.3d) to solving the problems (8.6) with the constraints of the kind $OO^\top = 1$. Now, being O an orthogonal matrix, then $\dot{O} = OS(v)$, with v a vector of proper dimension depending on the dimension of O, and $S(\cdot)$ a skew symmetric matrix. Assuming v and $\dot{D}$ as exogenous control inputs, one can find Lyapunov-like solutions to the problem (8.6). More precisely, the following result holds.

Lemma 5 *Let $O, D \in \mathbb{R}^{m \times m}$ denote an orthogonal and a diagonal matrix, respectively. Consider the following system:*

$$
\dot{D} = U \tag{8.8a}
$$

$$
\dot{O} = OS(v) \tag{8.8b}
$$

where the vector $v \in \mathbb{R}^{m(m-1)/2}$ and the diagonal matrix U are considered as exogenous control inputs. Define $\tilde{K} = K^* - X(O,D)$, and the operator $(\cdot)^\vee$ such that $v = S(v)^\vee$. Apply the control inputs

$$
U = -K_U \exp(-D) diag(B_1) \tag{8.9a}
$$

$$
v = K_v \left(\frac{B_2 - B_2^\top}{2}\right)^\vee \tag{8.9b}
$$

to system (8.8)*, where K_U is a positive definite diagonal matrix, K_v is a symmetric positive definite matrix, $B_2 = O^\top \exp(D) O K^{*\top} - K^{*\top} O^\top \exp(D) O$, $B_1 = \exp(D) - O K^* O^\top$, and diag$(B_1)$ defined as follows*

$$\begin{array}{lll} \textit{diag}(B_1)_{(i,j)} = B_{1(i,j)} & & \textit{if } i = j \\ \textit{diag}(B_1)_{(i,j)} = \quad 0 & & \textit{if } i \neq j. \end{array}$$

Then, the following results hold:

- *If K^* is symmetric and positive definite, the equilibrium point of the closed-loop dynamics $\tilde{K} = 0$ is stable;*
- *The system trajectories $\tilde{K}(t)$ are globally bounded for any $K^* \in \mathbb{R}^{m \times m}$;*
- *$|\tilde{K}(t)| \leq |\tilde{K}(0)|$ for any $K^* \in \mathbb{R}^{m \times m}$.*

The proof of Lemma 5 is in the Appendix C. The above Lemma points out that the *distance* between the optimal, unconstrained solution K^* (obtained in Section 8.1.3) and the constrained (symmetric, positive definite) gain $X(O(t), D(t))$ is non increasing, i.e. $|\tilde{K}(t)| \leq |\tilde{K}(0)|$. Then, the control laws (8.9) can be viewed as a tracker for symmetric positive definite matrices even when the matrix has to track a non symmetric positive definite matrix (i.e. it does not belong to the same manifold). Let us remark that convergence of the tracking error $\tilde{K}$ to zero is not ensured *a priori*. Simulations we have performed, however, tend to show that the cases when $\tilde{K}$ does not converge to zero are limited, and the analysis on this convergence is currently being developed.

Note also that if the optimal, unconstrained solution K^* varies in time slowly, the tracker preserves its properties by continuity. Then, one may think of applying the solution (8.5)–(8.9) on-line for time-varying desired stiffness and damping $Q_1^d(t), Q_2^d(t)$. Let us finally observe that we could have avoided to find the intermediate solution (8.5), define the optimization problem (8.3) in terms of the parametrization $X(O, D)$, and then apply the procedure explained above to find time evolution for the constrained gains. Simulations we have performed tend to show that this approach performs worse than the route we propose, and further investigations in this direction are being conducted.

8.1.5 Desired matrix correction for two feet balancing

When the kinematic constraint (3.6) acting on the system represents more than one robot frame fixed with respect to the inertial frame (e.g. two feet balancing), the matrices Q_1 and Q_2 in (8.1) may not be full rank. As a matter of fact, the minimal coordinates describing the constrained mechanical system are fewer than n in this case. Then, the ranks of the desired matrices Q_1^d and Q_2^d must be equal to those of the matrices Q_1 and Q_2. In general, the desired matrices Q_1^d and Q_2^d must be corrected according to the constraints acting on the system.

To do this, observe that the *feasible* joint accelerations according to the constraints (3.6) are given by:

$$\ddot{s} = -\bar{J}_s^{\dagger}\dot{\bar{J}}_s\dot{s} + N_J\ddot{s}_0, \tag{8.10}$$

where $\bar{J}_s = (1_{6n_c} - J_{\mathcal{B}}J_{\mathcal{B}}^{\dagger})J_s$, $\dot{\bar{J}}_s$ is the time derivative of $\bar{J}_s$ and N_J is the projector onto the null space of $\bar{J}_s$. In the above equation, we have used the expression $\mathrm{v}_{\mathcal{B}} = -J_{\mathcal{B}}^{\dagger}J_s\dot{s}$ and its derivative. Then, given two desired matrices Q_1^d, Q_2^d, we project them as follows

$$\begin{aligned}\overline{Q}_1^d &= N_J Q_1^d \\ \overline{Q}_2^d &= N_J Q_2^d.\end{aligned} \tag{8.11}$$

to correct the ranks and structure of the desired stiffness and damping according to the constraints (3.6) when $n_c > 1$.

8.2 Simulation Results

8.2.1 Simulation environment

Simulations are performed on a 23 degrees of freedom robot model representing the humanoid iCub. The simulation software is developed in MATLAB in the *custom* simulation environment described in Section 2.4.1. The system state is integrated through time by means of the numerical integrator MATLAB *ode15s*. In this simulation setup both problems (8.5)–(8.9) are solved offline, before starting the state integration. However, preliminary tests with online gains optimization in Gazebo have also been performed. To integrate the variables $\dot{D}, \dot{O}$, we use a fixed step Euler integrator with time step of 0.01 s.

8.2.2 Results

Simulations are performed for both the robot balancing on one foot and two feet. We choose Q_1^d to be positive definite and diagonal, and $Q_2^d = 2\sqrt{Q_1^d}$. In this case, the desired joint space dynamics aims at the following properties:

- The joint space dynamics is locally decoupled, i.e. each joint can be tuned independently from the others;

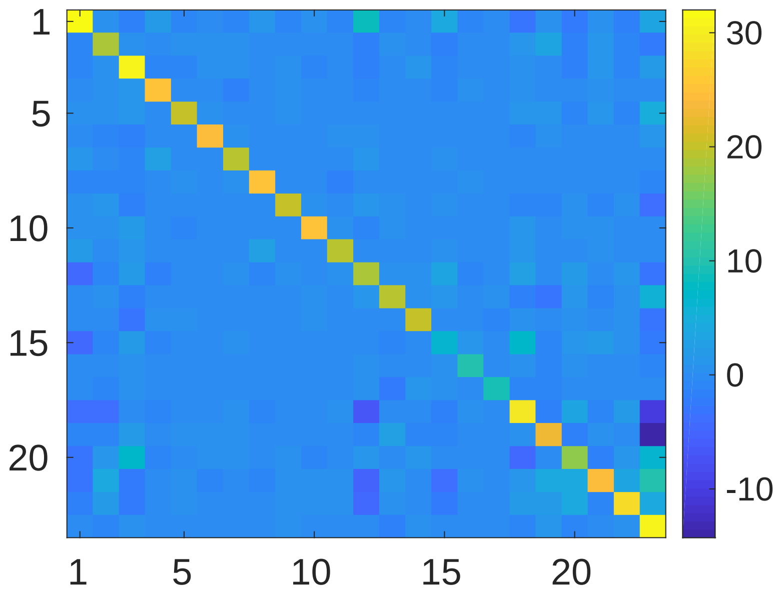


Fig. 8.1 Matrix Q_1 after the gain tuning procedure in case of one foot balancing. Simulations run in MATLAB environment.

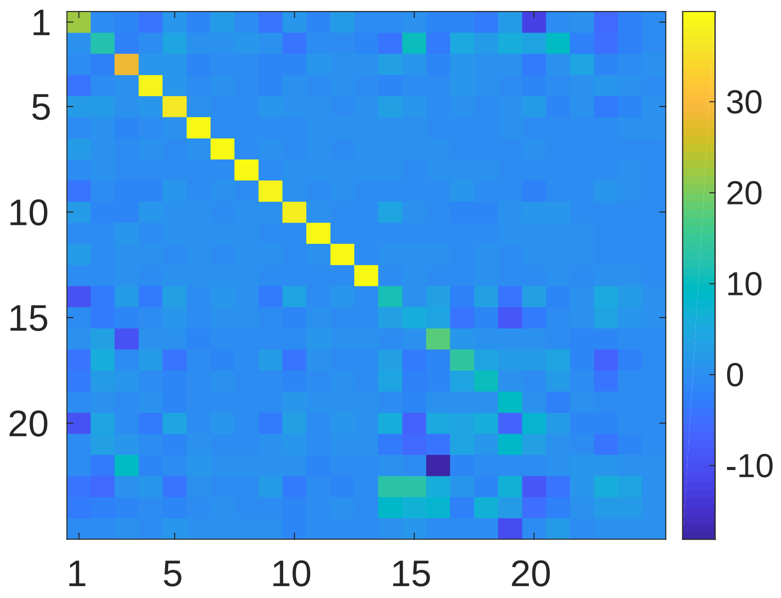


Fig. 8.2 Matrix Q_1 after the gain tuning procedure in case of two feet balancing. Simulations run in MATLAB environment.

- The linearized system is critically damped. This will avoid excessive overshoots in the joint space dynamics.

When the robot is balancing on two feet, the matrices Q_1^d, Q_2^d are corrected as in (8.11). Figures 8.1–8.2 show the shape of matrix Q_1 after gain tuning. Observe that in the case of one foot balancing, the matrix Q_1 is close to a diagonal matrix, and this implies that the joint space dynamics is almost locally decoupled. In case of two feet balancing, it is interesting to observe the effectiveness of the gain tuning procedure by looking at the difference between the first 11 rows of Q_1, and the last 12 rows, which correspond to the closed chain formed by the legs.

To verify that the obtained joint space dynamics is close to the desired dynamics, we evaluate the dynamical response of each joint to a step input. In particular, we focused on analyzing the settling time t_s. It is possible to approximate t_s as $t_s \approx \frac{-3}{\mathrm{Re}\,(\lambda)}$, where $\mathrm{Re}(\lambda)$ is the real part of system's eigenvalues. As an example, figures 8.3–8.4 show the dynamics of torso pitch for both one foot and two feet balancing. In both cases, $t_s \approx t_s^d$, and the actual dynamical behavior of the system is then closed to the desired dynamics.

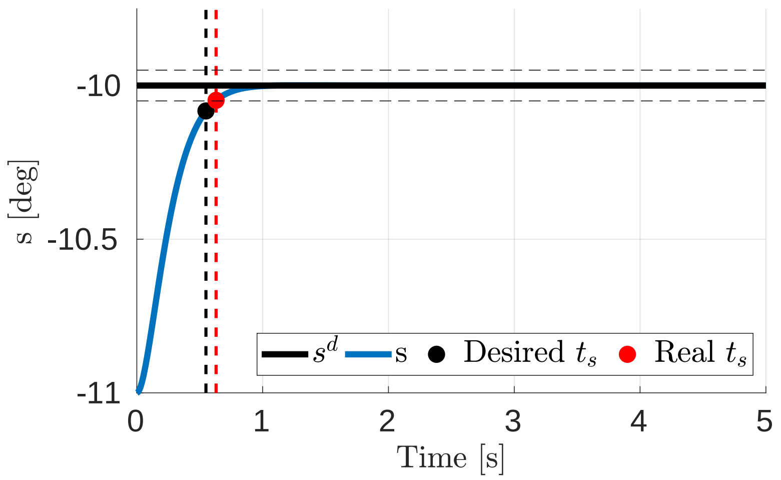


Fig. 8.3 Response of torso pitch to a step input in case of 1 foot balancing. The black dot indicates the desired settling time, while the red dot is the real settling time. The dashed horizontal lines indicate $\pm 5\%$ of the reference position. Simulations run in MATLAB.

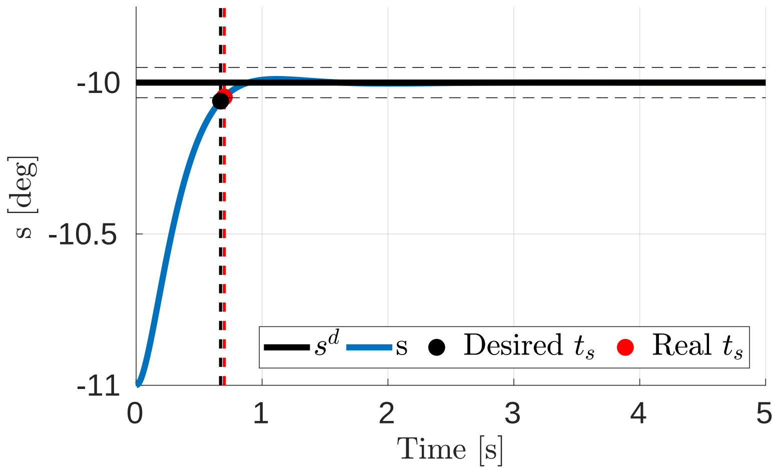


Fig. 8.4 Response of torso pitch to a step input in case of 2 feet balancing. In this case, the real and desired settling time are almost coincident. Simulations run in MATLAB.

Chapter 9

Momentum-Based Control for Balancing in Dynamic Environments

Forthcoming applications concerning humanoid robotics often involve physical interaction between the robot and a dynamic environment. In such scenario, classical balancing and walking controllers that neglect the environment dynamics may not be sufficient for achieving a stable robot behavior. This Chapter extends the momentum-based control framework presented in Chapter 6 - and modified in Chapter 7–8 - to the control of a robot in contact with a dynamic environment. We first model the robot and environment dynamics, together with the contact constraints. Then, a momentum-based control strategy for stabilizing the full system dynamics is proposed. Theoretical results are verified in simulation with the robot iCub balancing on a seesaw board.

9.1 iCub Balancing on a Seesaw

We designed a case study that exemplifies balancing on dynamical contacts: the robot is balancing with both feet leaning on a seesaw board of semi-cylindrical shape as in Fig.9.1. In what follows, we present a modeling and control framework derived from the momentum-based control (6.5)–(6.6)–(6.7) for two feet balancing on a seesaw board.

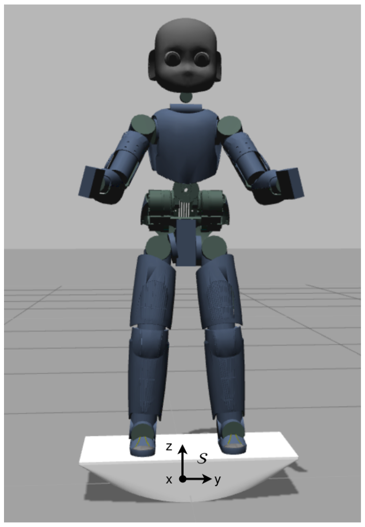


Fig. 9.1 iCub balancing on a seesaw.

9.1.1 Seesaw dynamics

The seesaw can be considered a single rigid body with no *a priori* fixed position and orientation w.r.t. the inertial frame. We denote with $\mathcal{S}$ a frame attached to the seesaw board, whose origin coincides with the seesaw center of mass. We also assume the seesaw is in contact with both the robot feet and a rigid ground, exerting on them the reaction wrenches $-f$ and the contact forces and moments $f_{\mathcal{S}}$, respectively. A complete description of the seesaw dynamics is given by the equations representing the rate of change of seesaw momentum, i.e. $\dot{L}_{\mathcal{S}}$. In particular, we project the rate of change of seesaw momentum in the seesaw frame $\mathcal{S}$ previously defined, resulting in the following equations of motion:

$$M_{\mathcal{S}} {}^{\mathcal{S}}\dot{\nu}_{\mathcal{S}} + h_{\mathcal{S}} = -J_R^\top f + J_{\mathcal{S}}^\top f_{\mathcal{S}} \tag{9.1}$$

where $M_{\mathcal{S}} \in \mathbb{R}^{6\times 6}$ is the seesaw mass matrix, ${}^{\mathcal{S}}\dot{\nu}_{\mathcal{S}} \in \mathbb{R}^6$ is the vector of seesaw linear and angular accelerations, $h_{\mathcal{S}} \in \mathbb{R}^6$ represents Coriolis, gravity and centrifugal effects. The jacobians J_R and $J_{\mathcal{S}}$ are the map between the seesaw velocity in the seesaw frame ${}^{\mathcal{S}}\nu_{\mathcal{S}}$ and the velocities at the contacts locations.

In the chosen (local) frame of reference the mass matrix $M_{\mathcal{S}}$ is given by:

$$M_{\mathcal{S}} = \begin{bmatrix} m_{\mathcal{S}} 1_3 & 0_3 \\ 0_3 & {}^{\mathcal{S}}I_{\mathcal{S}}, \end{bmatrix}$$

where $m_{\mathcal{S}}$ represents the mass of the seesaw, while the inertia matrix ${}^{\mathcal{S}}I_{\mathcal{S}} \in \mathbb{R}^{3\times 3}$ is constant, thus simplifying the formulation of seesaw dynamics. Further details on the derivation of Eq. (9.1) can be found in Appendix D. As an abuse of notation but for the sake of clarity let us omit from now on the superscript $\mathcal{S}$, e.g. ${}^{\mathcal{S}}\nu_{\mathcal{S}} = \nu_{\mathcal{S}}$.

9.1.2 Modeling contact constraints

It is assumed that the robot feet are always attached to the seesaw, resulting in the following set of constraints:

$$\mathrm{v}_{feet} = J_R \nu_{\mathcal{S}} = J\nu. \tag{9.2}$$

Note that the above equation is coupling the seesaw and the robot dynamics. In fact, by differentiating Eq. (9.2), one has:

$$J\dot{\nu} + \dot{J}\nu = J_R \dot{\nu}_{\mathcal{S}} + \dot{J}_R \nu_{\mathcal{S}}. \tag{9.3}$$

Lastly, we define the contact point P as the intersection between the contact line of the seesaw with the ground and its frontal plane of symmetry. We assume that the seesaw is only rolling, and this implies that the linear velocity of the contact point P is $v_P = 0$. Furthermore, let the seesaw frame be oriented as in figure 9.1. The shape of the (semi-cylindrical) seesaw constrains the rotation along the y and z axis. We model all these constraints as follows:

$$J_{\mathcal{S}} \nu_{\mathcal{S}} = 0, \tag{9.4}$$

and differentiating Eq. (9.4) gives:

$$J_{\mathcal{S}} \dot{\nu}_{\mathcal{S}} + \dot{J}_{\mathcal{S}} \nu_{\mathcal{S}} = 0. \tag{9.5}$$

The shape of the Jacobian matrices $J_R \in \mathbb{R}^{6n_c \times 6}$, $J_{\mathcal{S}} \in \mathbb{R}^{5 \times 6}$ and their time derivatives are detailed in Appendix D.

9.1.3 Constrained robot and environment dynamics

Recall that the robot dynamics is described by the floating base system of equations (3.4). Then, the system dynamics for the robot balancing on a seesaw is given by the following set of equations:

$$\begin{aligned}
&\text{floating base dynamics} \\
&M\dot{\nu} + h = B\tau + J^\top f && \text{(9.6a)} \\
&\text{seesaw dynamics} \\
&M_{\mathcal{S}}\dot{\nu}_{\mathcal{S}} + h_{\mathcal{S}} = -J_R^\top f + J_{\mathcal{S}}^\top f_{\mathcal{S}} && \text{(9.6b)} \\
&\text{constraint: feet attached to the seesaw} \\
&J\dot{\nu} + \dot{J}\nu = J_R\dot{\nu}_{\mathcal{S}} + \dot{J}_R\nu_{\mathcal{S}} && \text{(9.6c)} \\
&\text{constraint: seesaw is only rolling} \\
&J_{\mathcal{S}}\dot{\nu}_{\mathcal{S}} + \dot{J}_{\mathcal{S}}\nu_{\mathcal{S}} = 0. && \text{(9.6d)}
\end{aligned}$$

The above equations are valid for the specific case of a semi-cylindrical seesaw, but the approach we followed for obtaining Eq. (9.6) is more general, and can be reused in case the robot is interacting with different dynamic environments.

9.2 The Balancing Control Strategy

To achieve robot balancing on the seesaw, we may define an output variable that attempts at stabilizing both the seesaw and the robot dynamics. For example, let us consider the control output $[\dot{L}^\top,\ \dot{\nu}_{\mathcal{S}}^\top]^\top$ composed of the robot momentum rate of change and the seesaw accelerations. A first observation is that both outputs depend (linearly) on the feet wrenches f, that we may assume to be our control variable. In fact, by substituting the seesaw dynamics Eq. (9.1) into (9.5), one has:

$$J_{\mathcal{S}}M_{\mathcal{S}}^{-1}(J_{\mathcal{S}}^\top f_{\mathcal{S}} - h_{\mathcal{S}} - J_R^\top f) + \dot{J}_{\mathcal{S}}\nu_{\mathcal{S}} = 0. \tag{9.7}$$

Writing explicitly $f_{\mathcal{S}}$ from Eq. (9.7) gives:

$$f_{\mathcal{S}} = \Gamma^{-1}(J_{\mathcal{S}}M_{\mathcal{S}}^{-1}(h_{\mathcal{S}} + J_R^\top f) - \dot{J}_{\mathcal{S}}\nu_{\mathcal{S}}),$$

with $\Gamma = (J_{\mathcal{S}}M_{\mathcal{S}}^{-1}J_{\mathcal{S}}^\top)$. Now, substitute the above equation into Eq. (9.1):

$$M_{\mathcal{S}}\dot{\nu}_{\mathcal{S}} + \bar{h}_{\mathcal{S}} = A_{\mathcal{S}}f \tag{9.8}$$

where $\bar{h}_{\mathcal{S}} = (1_6 - J_{\mathcal{S}}^\top\Gamma^{-1}J_{\mathcal{S}}M_{\mathcal{S}}^{-1})h_{\mathcal{S}} + J_{\mathcal{S}}^\top\dot{J}_{\mathcal{S}}\nu_{\mathcal{S}}$, while the matrix multiplying the feet forces and moments is given by $A_{\mathcal{S}} = -(1_6 - J_{\mathcal{S}}^\top\Gamma^{-1}J_{\mathcal{S}}M_{\mathcal{S}}^{-1})J_R^\top$.

Observe that the output obtained combining the robot momentum rate of change and the seesaw accelerations yields the following matrix that multiplies the feet wrenches: $A_f = \begin{bmatrix} J_{\mathcal{B}}^\top \\ M_{\mathcal{S}}^{-1}A_{\mathcal{S}} \end{bmatrix}$. Let us now recall that the seesaw can only roll due to its costraints with the environment. As a consequence, we may focus on controlling only the robot momentum and the seesaw rolling motion.

However, a singular values decomposition (SVD) analysis on matrix A_f for different robot and seesaw configurations pointed out that matrix A_f has $\text{rank}(A_f) = 6$, and not 7 as it may be expected from the considerations stated previously. This implies that seesaw and robot momentum dynamics are not independent outputs, and therefore we can only control a subset (or a combination) of dimension 6 of the seesaw and robot equations (6.1)–(9.8).

To overcome this limitation, in what follows we propose two different control outputs to achieve balancing on the seesaw. The first algorithm controls the robot's momentum only, thus directly extending the momentum-based controller designed for rigid contacts and presented in Chapter 6. The boundedness of the seesaw rolling motion is then verified numerically. The second strategy is to control the robot linear momentum and the *total* angular momentum of the system.

9.2.1 Control of robot momentum only

Being not possible to separately control both the seesaw and the robot momentum dynamics, we first decided to select only the robot momentum as primary control objective, and then to verify numerically that the seesaw angle trajectory always remains bounded within a limited range. For the given task, the desired contact wrenches f^* are obtained as in Eq. (6.5):

$$f^* = (J_{\mathcal{B}}^\top)^\dagger \left(\dot{L}^* + mge_3\right) + N_{\mathcal{B}} f_0.$$

9.2.2 Control of mixed momentum

Another possibility is to control a quantity which depends on both the robot and the seesaw dynamics, for example the rate of change of the total momentum L_T:

$$\dot{L}_T = J_T^\top f_{\mathcal{S}} - (m_{\mathcal{S}} + m)ge_3 = J_T^\top A_T f + f_{bias} \tag{9.9}$$

where the formulation of Eq. (9.9) including the definition of J_T, A_T and f_{bias} is described in Appendix D. Being $f_{\mathcal{S}} \in \mathbb{R}^5$, matrix J_T has dimensions $\mathbb{R}^{5\times 6}$ and its maximum rank is 5, hence the total momentum dynamics cannot be (always) fully controlled. To finally achieve a full rank matrix multiplying the feet wrenches, we decide to control the linear momentum of the robot, together with the angular momentum of the whole system. Let us also observe that there is a difference of magnitude between the robot mass (31 kg) and the seesaw mass (4 kg), which implies that the center of mass of the overall system (and therefore the total linear momentum) is close to the center of mass of the robot.

Consider the following partitions of robot and total momentum:

$$L := \begin{bmatrix} L_l \\ L_\omega \end{bmatrix}, L_T := \begin{bmatrix} L_{Tl} \\ L_{T\omega} \end{bmatrix}$$

where $L_l, L_\omega \in \mathbb{R}^3$ are the linear and angular momentum of the robot, whereas $L_{Tl}, L_{T\omega} \in \mathbb{R}^3$ are the total linear and angular momentum. We define the *mixed* momentum as $L_M = \begin{bmatrix} L_l^\top & L_{T\omega}^\top \end{bmatrix}^\top$. Being $\dot{L}_M = \dot{L}_M(f)$, we can now choose f such that:

$$\begin{aligned} \dot{L}_M^* &= \dot{L}_M^d - K_P \tilde{L}_M - K_I I_{\tilde{L}_M} \\ \dot{I}_{\tilde{L}_M} &:= \tilde{L}_M, \end{aligned} \tag{9.10}$$

where L_M^d is the desired mixed momentum, $\tilde{L}_M = L_M - L_M^d$ is the momentum error and $K_P, K_I \in \mathbb{R}^{6\times 6}$ two symmetric, positive definite matrices. The feet wrenches f^* that instantaneously realize the desired mixed momentum rate of change are given by:

$$f^* = A_M^\dagger \left(\dot{L}_M^* + \begin{bmatrix} S_l m g e_3 \\ -S_\omega f_{bias} \end{bmatrix} \right) + N_M f_0,$$

where $S_l = \begin{bmatrix} 1_3 & 0_3 \end{bmatrix}$, $S_\omega = \begin{bmatrix} 0_3 & 1_3 \end{bmatrix}$ are selector matrices, matrix A_M is given by $A_M = \begin{bmatrix} S_l J_{\mathcal{B}}^\top \\ S_\omega J_T^\top A_T \end{bmatrix}$ and $N_M = 1 - A_M^\dagger A_M$ is the null space projector of A_M.

9.2.3 Joint torques and zero dynamics

Once the desired contact wrenches have been computed, the joint torques realizing f^* can be calculated as in (6.6)–(6.7) in Chapter 6. More specifically, by substituting the robot and seesaw accelerations from Eq. (9.6a)–(9.8) into the constraints (9.6c), and writing explicitly the control torques one has:

$$\tau = \Lambda^\dagger (\bar{h} - \bar{J} f - \dot{J}\nu + \dot{J}_R \nu_{\mathcal{S}}) + N_\Lambda \tau_0 \tag{9.11}$$

with $\bar{h} = J M^{-1} h - J_R M_{\mathcal{S}}^{-1} h_{\mathcal{S}}$, and $\bar{J} = J M^{-1} J^\top - J_R M_{\mathcal{S}}^{-1} A_{\mathcal{S}}$. The joint torques redundancy is used for stabilizing the robot posture, and therefore τ_0 remains the same of Eq. (6.7) with the gains as in (7.4) and s^d a constant reference. The wrench redundancy f_0, is exploited to minimize the norm of joint torques.

9.2.4 Quadratic programming formulation

The control architecture for balancing on a seesaw can be formulated as the following optimization problem:

$$
\begin{aligned}
f^* = \ & \underset{f}{\operatorname{argmin}} |\tau^*(f)|^2 && \text{(9.12a)}\\
s.t. \ & \\
& C_f f < b_f && \text{(9.12b)}\\
& \dot{L}(f) = \dot{L}^* && \text{(9.12c)}\\
& \tau^*(f) = \underset{\tau}{\operatorname{argmin}} |\tau - \tau_0(f)|^2 && \text{(9.12d)}\\
& \quad s.t. \\
& \quad \dot{J}\nu + J\dot{\nu} = \dot{J}_R \nu_{\mathcal{S}} + J_R \dot{\nu}_{\mathcal{S}} && \text{(9.12e)}\\
& \quad \dot{J}_{\mathcal{S}} \nu_{\mathcal{S}} + J_{\mathcal{S}} \dot{\nu}_{\mathcal{S}} = 0 && \text{(9.12f)}\\
& \quad \dot{\nu} = M^{-1}(B\tau + J^\top f - h) && \text{(9.12g)}\\
& \quad \dot{\nu}_{\mathcal{S}} = M_{\mathcal{S}}^{-1}(J_{\mathcal{S}}^\top f_{\mathcal{S}} - J_R^\top f - h_{\mathcal{S}}) && \text{(9.12h)}\\
& \quad \tau_0 = h_s - J_s^\top f - \overline{K}_D^s N_\Lambda M_s \dot{s} - \overline{K}_P^s N_\Lambda M_s (s - s^d), && \text{(9.12i)}
\end{aligned}
$$

which is very similar to the QP formulation of momentum-based control for rigid contacts (6.10), with the only difference of the modified contact constaints equations (9.12e)–(9.12f) and the presence of the seesaw dynamics (9.12h). In case the primary control objective is the stabilization of the *mixed* momentum trajectories, Eq. (9.12c) is of the form: $\dot{L}_M(f) = \dot{L}_M^*$.

9.3 Simulation Results

We test the control solutions proposed in Section 9.1 by using a model of the humanoid robot iCub with 23 degrees-of-freedom (DoFs). Simulations are performed by means of a Simulink controller interfacing with Gazebo simulator [Ch. 2, Sec. 2.4.2]. The controller frequency is 100 Hz.

9.3.1 Robustness to external disturbances

We first perform a robustness test on the closed loop system (9.6)–(9.12). The control objective is to stabilize the system about the equilibrium position. After 20 seconds an external force of amplitude 100 N is applied to the robot torso along the lateral direction for a period of 0.01 seconds. Figure 9.2 shows the norm of the robot linear momentum error for both control laws. Analogously, Figure 9.3 depicts the norm of robot and system angular momentum error when the primary

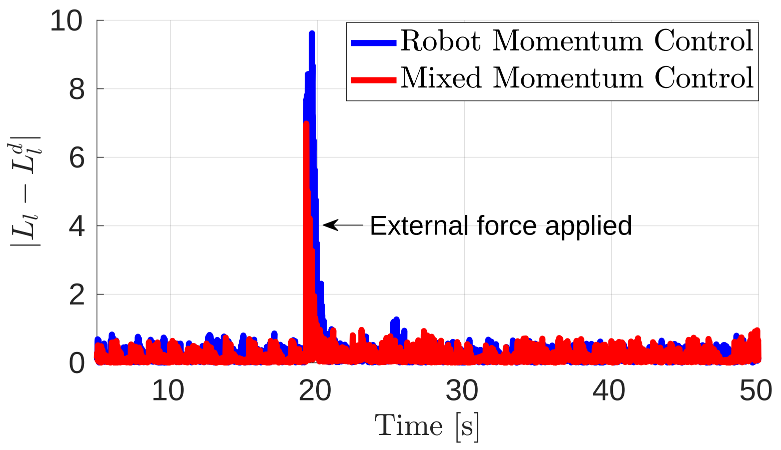


Fig. 9.2 Norm of robot linear momentum error while balancing.

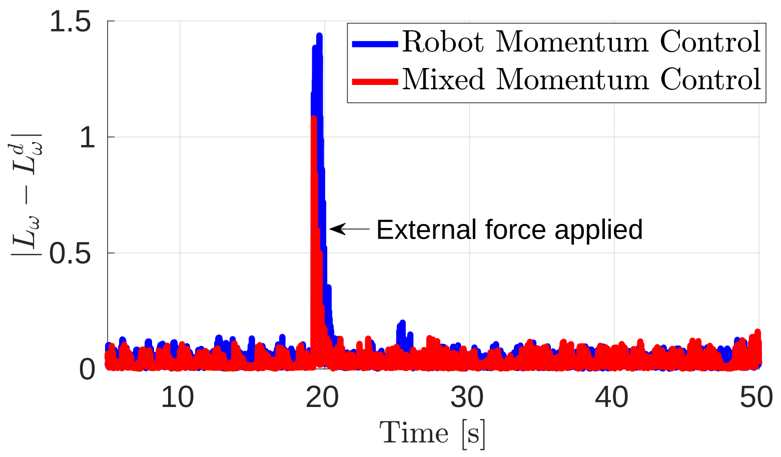


Fig. 9.3 Norm of robot and system angular momentum error while balancing.

task is to control the robot momentum and the mixed momentum, respectively. After the external force is applied, both controllers are able to bring the system back to the equilibrium position. Figure 9.4 shows instead the behaviour of the seesaw orientation θ. The blue line represents θ when the control objective is to stabilize the robot momentum, while the red line is when the primary task is to control the mixed momentum. In both cases, the trajectory of θ remains bounded even after the application of the external force, and therefore both controllers are able to stabilize also the seesaw orientation, even if it is not explicitly controlled.

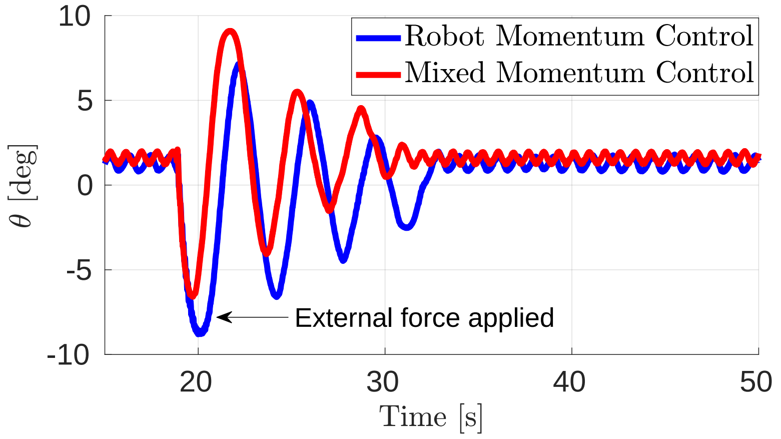


Fig. 9.4 Seesaw orientation. Even if not explicitly controlled, its trajectory remains bounded while the robot is balancing.

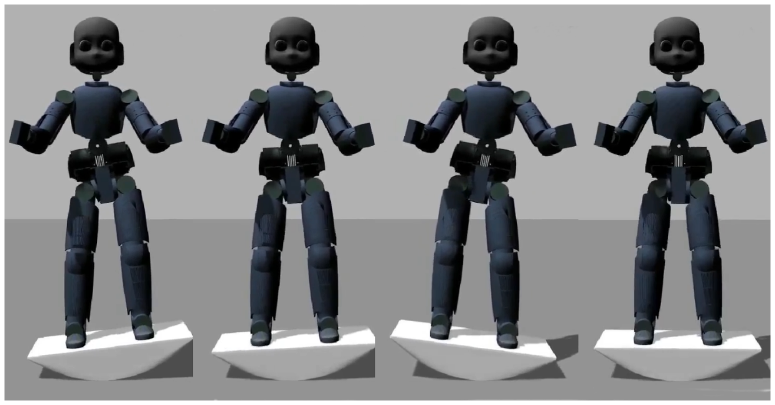


Fig. 9.5 Robot moving on the seesaw when tracking a desired CoM trajectory.

9.3.2 Tracking performances

We then evaluate the two control laws for tracking a desired trajectory of the robot center of mass. The reference trajectory for the center of mass is a sinusoidal curve with amplitude of 2.5 cm and frequency of 0.25 Hz along the robot lateral direction. The robot motion is depicted in Figure 9.5. A dedicated gain tuning has been performed on both controllers in order to achieve better tracking. It is important to point out that the main scope of this analysis is not a comparison between the controllers performances, but rather to verify the feasibility of a tracking task for both control strategies. Figure 9.6 shows the lateral component of the robot center of mass position during the tracking. The black line is the reference trajectory. The red line is the center of mass position obtained with the mixed momentum control, and the blue line is obtained with the robot momentum control. Both controllers are able to track the desired trajectory. However, with the mixed momentum control is possible to achieve qualitatively better results, while the other control strategy depicts poor tracking even after gain tuning. A possible explanation is that in order to keep balancing on the seesaw while tracking the CoM trajectory, the robot may be required to have an angular momentum reference different from zero. Therefore, trying to regulate the robot angular momentum to zero may worsen the tracking.

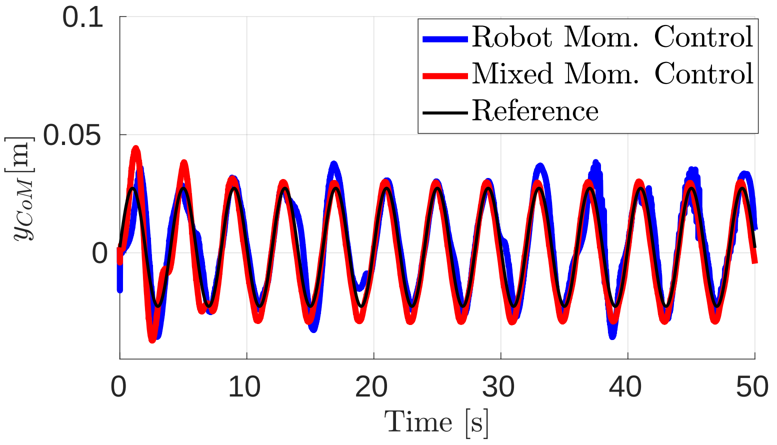


Fig. 9.6 Lateral component of robot CoM position. With mixed momentum control is possible to achieve better tracking results.

Chapter 10

Momentum-Based Control of Robots with Elastic Joints

The paradigm *the stiffer, the better* has characterised the design of robot actuation for years. This paradigm is well justified for position controlled industrial manipulators, where tasks usually require rapid and precise movements and very high repeatability. In the field of humanoid robotics, however, having a stiff actuation may be a strong limitation in terms of shock tolerance, efficiency, and safe interaction with humans [Pratt et al., 1997, Tsagarakis et al., 2009]. To add a degree of compliance, a widely held solution is the redesign of robot joints by adding a physical spring between the load and the transmission element. The presence of joint elasticity complexifies the control design associated with the underlying robot.

In this Chapter, we propose a control framework that extends momentum based controllers developed for stiff actuation to the case of series elastic actuators. The key point is to consider the motor velocities as an intermediate control input, and then apply high-gain control to stabilise the desired motor velocities achieving momentum control. Simulations carried out on a model of the robot iCub verify the soundness of the proposed approach.

10.1 Series Elastic Actuator Modeling

Chapter 6 has presented a balancing control assuming that the joints torque τ can be considered as control input. This is the case, for instance, of a torque controlled robot where the motors are rigidly connected to the joints, eventually by means of harmonic drives. In this Chapter, we present the extension of the dynamic model (3.4) when the interfaces between the motors and the joints are elastic elements, namely, the robot is powered by series elastic actuators. To this purpose, we make the following assumptions.

- The angular motor kinetic energy is due to its own spinning only, and the center of masses of each motor is along the motor axis of rotation;
- Both stiffness and damping of the series elastic actuators can be considered linear versus its relative displacement absolute value and rate-of-change;
- All motors are rigidly connected to the associated transmission element;
- Viscous and Coulomb friction, as well as the back-electromotive force, are compensated by a low-level motor control as described in Chapter 2. Hence, stiffness and damping are solely the ones of the series-elastic actuators.

Figure 10.1 depicts a simple block diagram of the series elastic actuator assumed to power the robot. In this picture, θ_i is the i-th motor position, s_i the i-th link position, η_i is the transmission ratio, b_i is the motor inertia, and k_{si} and k_{di} are the i-th link (torsional) stiffness and damping. Under the above assumptions, the

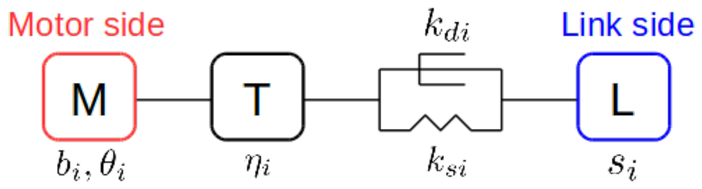


Fig. 10.1 Block diagram of the series elastic actuator.

system dynamics (3.4) can be extended by adding the dynamics of the motor angles $\theta = \begin{bmatrix}\theta_1 & \dots & \theta_n\end{bmatrix}^\top \in \mathbb{R}^n$. Then, one has (see [Albu-Schaffer et al., 2004, de Luca and Lucibello, 1998, Spong, 1990] for details):

$$\begin{aligned} M_{\mathcal{B}}\dot{\mathrm{v}}_{\mathcal{B}} &= J_{\mathcal{B}}^\top f - h_{\mathcal{B}} && (10.1a)\\ M_s\ddot{s} &= J_s^\top f - h_s + \tau && (10.1b)\\ I_m\ddot{\theta} &= \tau_m - \Gamma\tau && (10.1c)\end{aligned}$$

where the term

$$\tau := K_S(\Gamma\theta - s) + K_D(\Gamma\dot{\theta} - \dot{s}) \in \mathbb{R}^n \qquad (10.2)$$

represents the coupling between the joints dynamics and the motors dynamics. The positive definite diagonal matrices describing joints stiffness and damping are

$K_S = \text{diag}(k_{si})$ and $K_D = \text{diag}(k_{di})$, respectively. $I_m = \text{diag}(b_i) \in \mathbb{R}^{n\times n}$ is the motor inertia matrix, while $\Gamma \in \mathbb{R}^{n\times n}$ is the matrix that accounts for the gear box ratios and for the coupling between the input and the output rotations of the coupling mechanism as in (2.1). Without loss of generality, in this Chapter we consider a robot with all joints decoupled at the motor level, therefore $\Gamma = \Gamma^\top = \text{diag}(\eta_i)$. The control input is given by the motor torques $\tau_m \in \mathbb{R}^n$. The new system configuration space is the Lie Group

$$\overline{\mathbb{Q}} = \mathbb{R}^3 \times SO(3) \times \mathbb{R}^{2n},$$

and a system configuration is represented by the quadruple $\overline{q} = ({}^{\mathcal{I}}p_{\mathcal{B}}, {}^{\mathcal{I}}R_{\mathcal{B}}, s, \theta)$. The velocity is then represented by the set $\overline{\mathbb{V}} = \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^{2n}$, and an element of $\overline{\mathbb{V}}$ is then $\overline{\nu} = ({}^{\mathcal{I}}v_{\mathcal{B}}, {}^{\mathcal{I}}\omega_{\mathcal{B}}, \dot{s}, \dot{\theta})$. The contact constraint equation (3.7) remains invariant, namely, it is not affected by the addition of the motor dynamics. In fact, this equation represents the fact that the feet acceleration, expressed in terms of q, is equal to zero for all the time.

Remark 3 *We assume that the matrix M_s in Eq.* (10.1b) *does not contain terms due to the so-called* motor reflected inertia. *More precisely, the equations of motion for the rigid actuation case can be deduced by imposing $K_S \to \infty$ in Eq.* (10.1)*, which implies $s \to \Gamma\theta$, $\dot{s} \to \Gamma\dot{\theta}$, $\ddot{s} \to \Gamma\ddot{\theta}$. Then, by summing up* (10.1b)-(10.1c)*, and by multiplying* (10.1c) *times Γ^{-1}, one has:*

$$M_{\mathcal{B}}\dot{\mathrm{v}}_{\mathcal{B}} = J_{\mathcal{B}}^\top f - h_{\mathcal{B}} \tag{10.3a}$$

$$(M_s + \Gamma^{-1}I_m\Gamma^{-1})\ddot{s} = J_s^\top f - h_s + \Gamma^{-1}\tau_m. \tag{10.3b}$$

These equations of motion characterise the system evolution in the case of rigid transmissions, and the term $\Gamma^{-1}I_m\Gamma^{-1}$ is called motor reflected inertia. *Hence, we assume that M_s does not take this term into account.*

10.2 Control Design

The control of system (10.1) for robot balancing purposes may not be straightforward. In fact, assuming that the control objective is still the asymptotic stabilisation of the momentum error $\tilde{L}$, then its rate-of-change is no longer influenced by the system control input, namely, the motor torque τ_m. More precisely, the momentum rate of change equation (6.1) still holds, i.e.

$$\dot{L}(f) = J_{\mathcal{B}}^\top f - mge_3.$$

By using the (feet zero-acceleration) constraint (3.7), i.e.

$$J_{\mathcal{B}}\dot{\mathrm{v}}_{\mathcal{B}} + J_s\ddot{s} + \dot{J}_{\mathcal{B}}\mathrm{v}_{\mathcal{B}} + \dot{J}_s\dot{s} = 0,$$

with the robot acceleration deduced from (10.1a)-(10.1b):

$$\begin{aligned}\dot{\mathrm{v}}_{\mathcal{B}} &= M_{\mathcal{B}}^{-1}\left(J_{\mathcal{B}}^{\top} f - h_{\mathcal{B}}\right) \\ \ddot{s} \ &= M_s^{-1}\left(J_s^{\top} f - h_s + \tau(\theta, \dot{\theta}, s, \dot{s})\right)\end{aligned}$$

one observes that we can no longer relate the contact force f to the system input τ_m. Consequently, the momentum rate-of-change $\dot{L}$ is no longer instantaneously influenced by the system input τ_m.

For the same reason, the control algorithm recalled in Chapter 6 is no longer applicable in the case of robots powered by series elastic actuators. In fact, even if one chooses a contact force f to achieve $\dot{L}(f) = \dot{L}^*$, then this force can no longer be generated by (6.6), i.e.

$$\tau = \Lambda^{\dagger}(JM^{-1}(h - J^{\top} f) - \dot{J}\nu) + N_{\Lambda}\tau_0,$$

since τ does not depend upon the control input τ_m, but only on the system state – see Eq. (10.1c). In the language of *Control Theory*, we would say that the output L has a *relative degree* equal to two: the second order time derivative of L can be instantaneously influenced by the control input τ_m. Then, one may attempt at the control of L by performing feedback linearisation of $\ddot{L}$, and then choose the input redundancy for postural control. Analogously, one may consider $\dot{f}$ as a fictitious control input in the dynamics of $\ddot{L}$, i.e.

$$\ddot{L}(\dot{f}) = \dot{J}_{\mathcal{B}}^{\top} f + J_{\mathcal{B}}^{\top}\dot{f},$$

and then exploit the time derivative of (6.6) with (10.1c) to impose a rate-of-change of the contact force f.

The main drawbacks of these strategies – which are based on pure-feedback linearisation of an output with relative degree higher than one – are the following:

- the need of feedforward terms seldom precisely known in practice, such as $\dot{M}$, $\ddot{J}$, etc;
- when series-elastic-actuators are introduced to substitute some, or all, rigid transmission mechanisms, they usually need a new time-consuming and specific gain tuning procedure;
- they do not leverage the (usually) high frequency and reliable low-level motor velocity control.

We then follow a different route to deal with series elastic actuators that aims – when the motors are equipped with it – at exploiting the low level motor velocity control. In addition, the proposed strategy is shown to be robust against the feedforward terms usually needed by feedback linearisation, and can also exploit the gain tuning procedure developed for the rigid actuation case.

10.2.1 The balancing control for series elastic actuators

Eq. (10.1c) points out that the motor dynamics $\ddot{\theta}$ is fully actuated. Then, any desired motor velocity $\dot{\theta}^d$ can be stabilised with any desired, *small*, settling time. This in turn implies that the motor velocity can be assumed as a virtual control input in the dynamics (10.1b). In the language of Automatic Control, assuming $\dot{\theta}$ as control variable is a typical *backstepping* assumption. Then, the production of the motor torques associated with the desired motor velocities can be achieved via classical nonlinear control techniques [Khalil, 2002, p. 589] or high-gain control. We later detail an implementation of the latter for obtaining the aforementioned motor torques and perform simulation results. More precisely, we consider

$$\beta = K_D \Gamma \dot{\theta}$$

as a fictitious control input of the joints dynamics (10.1b). Then, one has:

$$M_s \ddot{s} = J_s^\top f - \bar{h}_s + \beta, \tag{10.4}$$

with

$$\begin{aligned} \bar{h}_s &= h_s - p && (10.5a) \\ p \ &= K_S(\Gamma\theta - s) - K_D \dot{s}. && (10.5b) \end{aligned}$$

Observe that (10.4) coincides with (10.1b) by substituting h_s with $\bar{h}_s$ and τ with β. In light of this, the control input β achieving balancing control – with the same objectives detailed in Chapter 6-7 – is obtained by solving the optimisation problem (6.10) with the aforementioned substitutions, i.e.

$$\begin{aligned} f^* = \ & \underset{f}{\operatorname{argmin}} \, |\beta^*(f)|^2 && (10.6a) \\ s.t. \ & \\ & C_f f < b_f && (10.6b) \\ & \dot{L}(f) = \dot{L}^* && (10.6c) \\ & \beta^*(f) = \underset{\beta}{\operatorname{argmin}} \, |\beta - \beta_0(f)|^2 && (10.6d) \\ & \quad s.t. \\ & \quad \quad \dot{J}(q,\nu)\nu + J(q)\dot{\nu} = 0 && (10.6e) \\ & \quad \quad \dot{\nu} = M^{-1}(B\beta + J^\top f - h) && (10.6f) \\ & \quad \quad \beta_0 = \bar{h}_s - J_s^\top f + u_0. && (10.6g) \end{aligned}$$

with $h := (h_{\mathcal{B}}, \bar{h}_s)$, and $u_0 := M_s \ddot{s}^d - K_P^s N_\Lambda M_s (s - s^d) - K_D^s N_\Lambda M_s (\dot{s} - \dot{s}^d)$. The optimisation problem (10.6) points out that the redundancy of the contact forces in achieving $\dot{L}(f) = \dot{L}^*$ is no longer exploited to minimise the joint torques, but rather the torques induced by the motor velocities – compare Eq. (6.10) and (10.6). This basically means that the solution to the above problem tends to minimise the desired motor velocities.

Once the optimisation problem (10.6) is solved, at each time instant the (desired) motor velocities are given by

$$\dot{\theta}^d = \Gamma^{-1} K_D^{-1} \beta^*(f^*). \tag{10.7}$$

Now, if the series-elastic actuators provide the user with a velocity control interface, one can send as desired values to this interface the velocities (10.7). On the other hand, if the series-elastic actuators provide as interface the motor torques τ_m (often related to the motor PWM), then one can apply high-gain control on the dynamics (10.1c) to stabilise the motor velocity (10.7). In particular, motor velocity control via the motor torque τ_m can be achieved by:

$$\tau_m = \, - I_m K_m(\dot{\theta} - \dot{\theta}^d) + \Gamma\tau \tag{10.8}$$

with $K_m \in \mathbb{R}^{n \times n}$ a positive diagonal matrix. The closed-loop motors dynamics is then given by:

$$\ddot{\theta} = \, - K_m(\dot{\theta} - \dot{\theta}^d), \tag{10.9}$$

which implies that the motor velocity tracking error stays relatively *small* for relatively high gains K_m. It is important to observe that the control law (10.8) misses the feed-forward component $\ddot{\theta}^d$, and it is not deduced by the application of pure backstepping techniques [Khalil, 2002, p. 589]. All these missing elements represent a robustness test for the controller presented in this Chapter.

10.2.2 The mixed actuation case: stiff and elastic actuators

The framework presented above may be useful when the humanoid robot is powered by stiff and elastic actuators. In this case, one can still solve the optimisation problem (10.6) by properly defining the vectors $\bar{h}_s$ and p in Eq. (10.5).

To provide the reader with an example of such a mixed actuation case, assume that the robot possesses m_1 stiff actuators – for which the associated joint torques $\tau_{m_1} \in \mathbb{R}^{m_1}$ can be considered as control inputs – and m_2 series-elastic actuators, with $n = m_1 + m_2$. Assume that in the serialisation of the joint angles $s = (s_{m_1}, s_{m_2})$, the first m_1 joints are powered by stiff actuators, and the remaining

m_2 by elastic ones. Then, the optimisation problem (10.6) can be solved with

$$\begin{aligned}\beta &:= \begin{pmatrix}\tau_{m_1} \\ K_D \Gamma \dot{\theta}\end{pmatrix} \\ p &= \begin{pmatrix} 0 \\ K_S(\Gamma\theta - s_{m_2}) - K_D \dot{s}_{m_2}\end{pmatrix}\end{aligned} \tag{10.10}$$

where now $\theta \in \mathbb{R}^{m_2}$.

10.3 Simulation Results

We test the proposed control solution by using a model of the humanoid robot iCub with 25 degrees-of-freedom (DoFs). The inertia values b_i are obtained from the motor data-sheets, and their order of magnitude is around 10^{-5} kgm^2. Realistic stiffness and damping values for the series elastic actuators are obtained from previous work on the design of SEA for iCub legs [Parmiggiani et al., 2012, Tisi et al., 2016]. The stiffness value used for the experiments is 350 $\frac{\mathrm{Nm}}{\mathrm{rad}}$ for all the joints while damping coefficient is 0.25 $\frac{\mathrm{Nms}}{\mathrm{rad}}$. The transmission ratio is the same for all joints and set equal to $\eta_i = \frac{1}{100}$.

The control algorithm (10.6)-(10.8) is tested by performing simulations in the MATLAB *custom* environment (see Section 2.4.1). The system evolution is obtained by integrating Eq. (10.1) with τ_m as in Eq. (10.8), using MATLAB variable step numerical integrator *ode15s*. For solving the optimization problem (10.6), we employ the *qpOASES* solver.

10.3.1 Tracking performances

We evaluate the performances of the control laws (10.6)–(10.8) for tracking a desired center of mass trajectory while the robot is balancing on one foot. The reference trajectory for the center of mass is a sinusoidal curve with amplitude of 1 cm and frequency of 0.2 Hz along the robot lateral direction. Note that stability results presented in Chapter 7 are valid for quasi-static motions while balancing on one foot. However, we also verified numerically the effectiveness of the proposed control algorithm in case of highly dynamic movements and contact switching as in Figure 10.4. Furthermore, we perform an indicative sensitivity analysis on the tracking error in case of uncertainties on damping matrix K_D. This analysis holds for this particular task, but results may vary in case other movements are required. Figure 10.2 shows the tracking error on the center of mass trajectory. The thick blue line represents the center of mass error assuming perfect knowledge of system's damping. The dashed lines represent the error when the controller overestimates the real damping by 45% and 60%, while the dotted line is obtained when

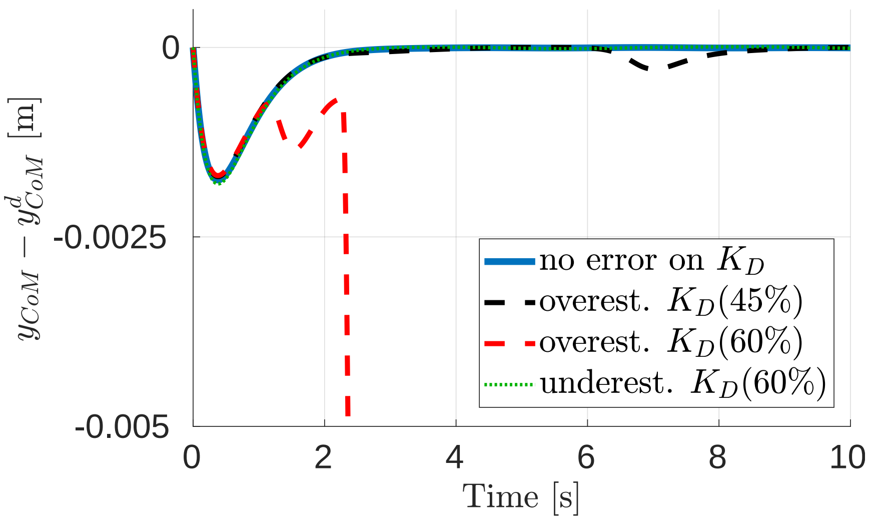


Fig. 10.2 CoM error on lateral direction. If K_D is overestimated by 60%, the closed-loop system is unstable.

K_D is underestimated by 60%. Focus on the case where K_D is overestimated by the controller, since it may lead to unstable behaviours. In particular, the center of mass error is still not increasing up to an error of 45% on K_D, while an error of 60% makes the system unstable.

10.3.2 Comparisons between different control models

Theoretically, one may apply the control law obtained from (6.10), which was developed under the assumption of stiff joints, when the system is, instead, powered by series elastic actuators, i.e. it is governed by (10.1). We here show that in this case, the closed loop system may have unstable behaviours because the control model neglects joint elasticity. In particular, the solution to the optimisation problem (6.10) is a joint torque $\tau = \Gamma^{-1}\tau_m$ (see Remark 3).

To test the controller obtained from (6.10) in the series-elastic actuator case, we impose a step response for the desired center of mass that translates into a step of 1° for all upper body joints (torso and arms). Figure 10.3 depicts the norm of the joint position errors in both stiff and elastic control model. It is clear that the closed-loop system with the rigid control (6.5)–(6.6)–(6.7) (blue line) is unstable, while the elastic control (red line) ensures the convergence to the desired position.

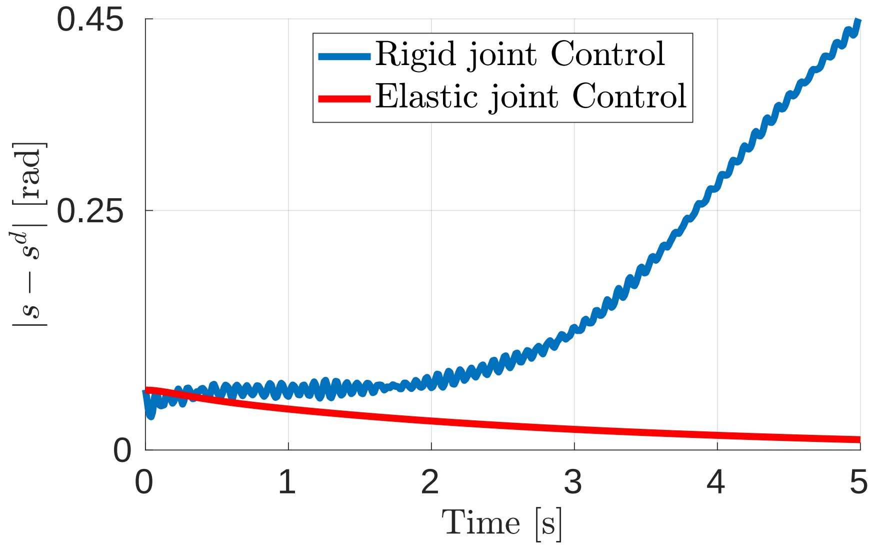


Fig. 10.3 Comparison between control law (6.5)–(6.6)–(6.7) and (10.6)–(10.8) for controlling system (10.1). The rigid joints control fails to stabilize the closed loop system about the reference position.

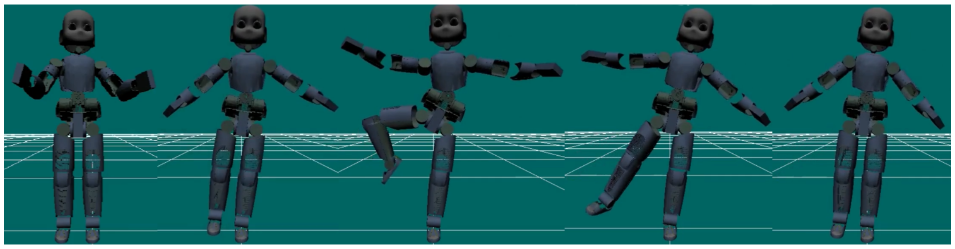


Fig. 10.4 Highly dynamic movements and contact switching with series elastic actuators.

10.3.3 Effects of torque saturation

On the real robot, it may not be possible to achieve the desired motor velocities $\dot{\theta}^d$ because of limited motor torques, and the controller might fail to stabilize the closed-loop system. To analyze the behaviour of elastic joint control in presence of limited motor torques, we add torque saturation to the simulation setup used in 10.3.1. The maximum motor torque available is obtained through motors datasheet and it is $0.34\ \mathrm{Nm}$. Figure 10.5 shows the effect of torques saturation on motor velocity of the stance foot ankle roll: the black line is the reference velocity. To better visualize the results, we cut the initial peak (around $50\ \frac{\mathrm{rad}}{\mathrm{s}}$). Dashed green line represents motor velocity without torque saturation, while the red line consid-

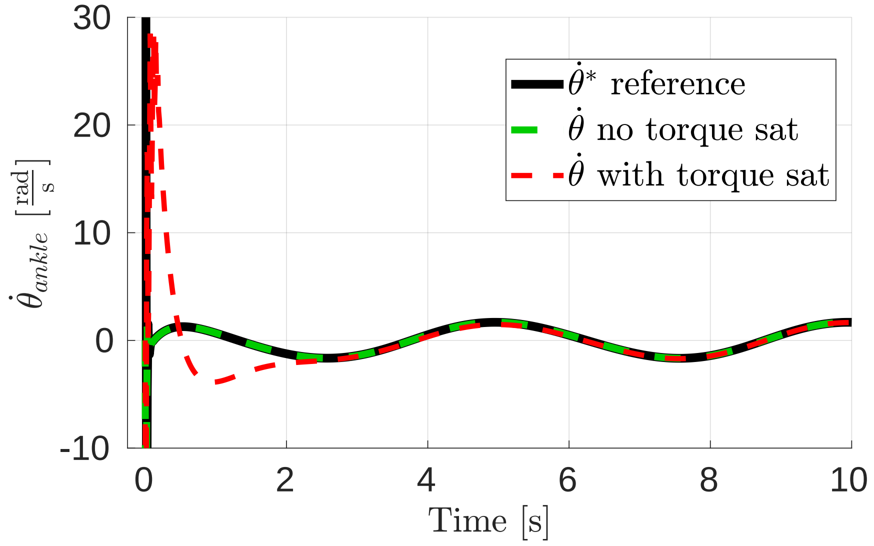


Fig. 10.5 Convergence of motor velocity to $\dot{\theta}^*$ with and without torque saturation. Even in case of limited torques, the controller is able to stabilize the closed-loop system.

ers also torque saturation. In this second case, the convergence of $\dot{\theta}$ to $\dot{\theta}^*$ is slower, but stability is still retained. For the sake of simplicity, we focused our attention on motor velocity of the stance foot ankle roll, because this is the joint that requires the biggest torque for the given task.

Chapter 11

Momentum-Based Control with Friction Exploitation

Chapter 2 describes the iCub torque control architecture as composed of two nested loops. The outer loop, which implements the momentum-based controller that was subject of the previous Chapters, generates desired joint/motor torques, while the inner loop stabilizes these desired values. In doing so, the inner loop usually compensates for joint friction phenomena, thus removing their inherent stabilizing property that may be also beneficial for high level control objectives. This Chapter shows how it is instead possible to exploit friction for joint and task space control of humanoid robots.

11.1 System Modeling

The equations representing the robot and motors dynamics in case of rigid transmissions have been already detailed in previous Chapters 2–3. The system equations of motion including motors are given by the combination of the floating base dynamics in its decoupled form as in (3.4) and the motors dynamics (2.2):

$$M_{\mathcal{B}}\dot{\mathrm{v}}_{\mathcal{B}} + h_{\mathcal{B}} = J_{\mathcal{B}}^{\top} f \tag{11.1a}$$

$$M_s\ddot{s} + h_s = J_s^{\top} f + \tau \tag{11.1b}$$

$$I_m\ddot{\theta} + K_v\dot{\theta} + K_c\mathrm{sign}(\dot{\theta}) = \tau_m - \Gamma^{\top}\tau \tag{11.1c}$$

subject to the contact constraints equations (3.7).

11.2 Control with Friction Exploitation

The two-loops control architecture presented in Chapter 2 is a common design for performing whole-body torque control of humanoid robots (see, e.g., [Del Prete et al., 2016]). However, it compensates for the effect of joints viscous and Coulomb friction, thus removing also their inherent stabilizing property that may be beneficial for achieving high level control objectives. Furthermore, the friction compensation terms in Eq. (2.3) may render the system sensitive to poor velocity measurements. In what follows, we design instead a control architecture that exploits friction to improve the tracking performances of both joints and momentum reference trajectories, and also aims at increasing the robustness of the system with respect to poor velocity measurements.

11.2.1 Reformulation of the system dynamics

Recall the kinematic relation $s = \Gamma\theta$ between the joints position s and motors position θ in case of rigid joints, and rewrite the motors dynamics Eq. (11.1c) by substituting $\ddot{\theta}$, $\dot{\theta}$ with $\ddot{s}$ and $\dot{s}$:

$$I_m\Gamma^{-1}\ddot{s} = \tau_m - K_f\Gamma^{-1}\dot{s} - \Gamma^\top\tau \tag{11.2}$$

where $K_f = K_f(\dot{s})$ is a diagonal matrix that collects the Coulomb and viscous friction coefficients. The elements along the diagonal of K_f are all positive, and of the form:

$$k_{f_{(i)}} = k_{v_{(i)}} + \frac{k_{c_{(i)}}}{|e_i^\top\Gamma^{-1}\dot{s}| + \epsilon}, \quad i = 1...n, \tag{11.3}$$

with e_i the canonical vector, consisting of all zeros but the i-th component that is equal to one, and the $\epsilon \in \mathbb{R}^+$, $\epsilon << 1$. The coefficients in Eq. (11.3) are obtained by rewriting the sign function as $\text{sign}(\dot{\theta}) = \frac{\dot{\theta}}{|\dot{\theta}|}$. The parameter ϵ is a regularization term that avoids the coefficients to reach infinite values when $\dot{s} \to 0$.

We multiply Eq. (11.2) times $\Gamma^{-\top}$ and we sum joints and motors equations of motion Eq. (11.1b)-(11.2) to get:

$$\begin{aligned}
&\overline{M}_s\ddot{s} = u - h_s + J_s^\top f - \overline{K}_f\dot{s}, \qquad (11.4)\\
&\text{with:}\\
&\overline{M}_s \quad = M_s + \Gamma^{-\top}I_m\Gamma^{-1},\\
&\overline{K}_f \quad = \Gamma^{-\top}K_f\Gamma^{-1},\ \overline{K}_f = \overline{K}_f^\top > 0,\\
&u \quad\ \ = \Gamma^{-\top}\tau_m.
\end{aligned}$$

More specifically, the term $\Gamma^{-\top} I_m \Gamma^{-1}$ is the so-called *motor reflected inertia* [Albu-Schäffer et al., 2007]. It accounts for the effect of the motors inertia on the joint space dynamics, and it has a pivotal role in improving the numerical stability of the control algorithm when the control design requires to invert the joint space mass matrix. In fact, in case of a humanoid robot the matrix M_s is often *ill-conditioned* because of the presence the of links with very different mass and inertia properties. The motor reflected inertia may be also interpreted as a *physically consistent* regularization term that decreases the condition number of the mass matrix, defined as $\text{cond}(M_s) = \frac{\sigma^{max}}{\sigma^{min}}$, with σ^i the singular values of M_s. However, it is not straightforward to analytically prove that the regularized mass matrix $\overline{M}_s = M_s + \Gamma^{-\top} I_m \Gamma^{-1}$ is no more ill-conditioned. We verified numerically this result by comparing the condition numbers of M_s and $\overline{M}_s$.

Finally, the robot and motors dynamics Eq. (11.1) can be combined in the following system:

$$M_{\mathcal{B}} \dot{\mathrm{v}}_{\mathcal{B}} + h_{\mathcal{B}} = J_{\mathcal{B}}^\top f \tag{11.5a}$$

$$\overline{M}_s \ddot{s} + h_s \quad = u + J_s^\top f - \overline{K}_f \dot{s}. \tag{11.5b}$$

Now, system (11.5) may be stabilized with a proper choice of the variable u (hence, the motor torques τ_m).

11.2.2 Control of a fixed base robot

For a better understanding of the motivations behind our control approach, at first we assume that the robot base link is fixed on a pole, i.e. $(\dot{\mathrm{v}}_{\mathcal{B}}, \mathrm{v}_{\mathcal{B}}) = (0, 0)$, and no other links are in contact with the environment. Therefore the system dynamics Eq. (11.5) reduces to:

$$\overline{M}_s \ddot{s} + h_s = u - \overline{K}_f \dot{s}, \tag{11.6}$$

that is, the joints dynamics Eq. (11.5b) with $f = 0$. The control objective is the stabilization of a desired joints trajectory $(s, \dot{s}) = (s^d, \dot{s}^d)$. We choose the input $u := u^*$ as:

$$u^* = h_s + \overline{M}_s \ddot{s}^d - K_P^s \tilde{s} - -K_D^s \dot{\tilde{s}} + \overline{K}_f \dot{s}^d \tag{11.7}$$

with K_P^s, K_D^s two symmetric and positive definite matrices and $\tilde{s} = s - s^d$. Substituting Eq. (11.7) into Eq. (11.6) yields to the following closed-loop dynamics:

$$\overline{M}_s \ddot{\tilde{s}} + (K_D^s + \overline{K}_f) \dot{\tilde{s}} + K_P^s \tilde{s} = 0. \tag{11.8}$$

In particular, being $\overline{K}_f$ symmetric and positive definite, the term $\overline{K}_f \dot{\tilde{s}}$ in Eq. (11.8) enforces the feedback term on the joints velocity error. This allows to exploit the joints friction for improving the convergence of the closed loop system dynamics to the reference trajectory. More specifically, the idea of exploiting the passive properties of the system dynamics in the control design is typical of the *passivity-based* control approach [Li et al., 2012, Albu-Schäffer et al., 2007].

The advantage of applying the control law (11.7) rather than, e.g. a classical feedback linearization with friction compensation technique, is that it may guarantee better robustness with respect to poor velocity measurements. On real robotic applications the velocity measurements are often obtained by means of numerical differentiation of the joint/motor positions measurements, and the estimated values can be noisy, or delayed in case a filtering technique is applied to the signal. In the control law Eq. (11.7) the matrix $\overline{K}_f(\dot{s})$ is multiplied by the reference velocity $\dot{s}^d$ instead of the measured one, thus rendering the control algorithm less sensitive to the noise on the velocity measurements. Furthermore, in most control algorithms, the gain matrix that multiplies the joints velocity error K^s_D is limited to relatively small values, because high values of K^s_D may lead to numerical instability. The additional term $\overline{K}_f\dot{\tilde{s}}$ in the closed loop dynamics Eq. (11.8) contributes to increase the system damping without the need of modifying K^s_D, and it may improve the tracking performances of the controlled system.

Remark 4 *The control law Eq.* (11.7) *belongs to the family of controllers:*

$$\begin{aligned} u_f &:= h_s + \overline{M}_s\ddot{s}^d - K^s_P\tilde{s} + \bar{u}(K, \dot{s}, \dot{s}^d), \\ \bar{u} &= -K\dot{\tilde{s}} + \overline{K}_f\dot{s}, \\ K &= K^\top,\ K > 0. \end{aligned} \tag{11.9}$$

Note that for any symmetric and positive definite matrix K, substituting u_f from Eq. (11.9) in the joint space dynamics Eq. (11.6) always guarantees stability and convergence of the closed loop system dynamics to the reference trajectory. Among all the possible choices of the gain K, we may be interested in finding the one that minimizes the sensitivity of $\bar{u}$ w.r.t. the joint velocities $\dot{s}$, i.e.:

$$\begin{aligned} K^* :\ &= \text{argmin}_K |\frac{\delta(\bar{u})}{\delta\dot{s}}|^2 \\ s.t.& \\ &K = K^\top,\ K > 0. \end{aligned} \tag{11.10}$$

Assume that $\overline{K}_f$ does not depend on the joint velocities $\dot{s}$ (i.e., $\overline{K}_f$ accounts for the viscous friction only, while the Coulomb friction is compensated by the inner control loop). Then, the solution to problem (11.10) is given by $K^* = \overline{K}_f$, which leads to $\bar{u}^* = \overline{K}_f\dot{s}^d$. Finally, the corresponding control input u^*_f is given by: $u^*_f = h_s + \overline{M}_s\ddot{s}^d - K^s_P\tilde{s} + \overline{K}_f\dot{s}^d$, that coincides with Eq. (11.7) when we choose $K^s_D =$

0_n. In this sense, the control law Eq. (11.7) with $K_D^s = 0_n$ can be seen as the one among the family of controllers u_f that is less sensitive to the joints velocity, and it is therefore the most robust (among u_f) against poor velocity measurements. The extension of this theoretical framework in the more general case $\overline{K}_f = \overline{K}_f(\dot{s})$ will be addressed in future work.

11.2.3 Control of a floating base robot

In what follows we propose yet another modification of the momentum-based control algorithm Eq. (6.5)–(6.6)–(6.7) that allows to exploit friction for improving the tracking of a desired momentum trajectory. First, we compactly rewrite system (11.5) as follows:

$$\overline{M}\dot{\nu} + h = J^\top f + Bu - B\overline{K}_f\dot{s}, \tag{11.11}$$

where we recall that $h = \begin{bmatrix} h_{\mathcal{B}}^\top & h_s^\top \end{bmatrix}^\top$, $J = \begin{bmatrix} J_{\mathcal{B}} & J_s \end{bmatrix}$ and the selector matrix B is of the form $B = \begin{bmatrix} 0_{n\times 6} & 1_n \end{bmatrix}^\top$. The mass matrix $\overline{M}$ is given by:

$$\overline{M} = \begin{bmatrix} M_{\mathcal{B}} & 0_{6\times n} \\ 0_{n\times 6} & \overline{M}_s \end{bmatrix}.$$

The friction component $\overline{K}_f\dot{s}$ can be related to the contact wrenches f by means of the contact constraint equations Eq. (3.7). In particular, we substitute the state acceleration $\dot{\nu} = \overline{M}^{-1}(J^\top f - h + Bu - B\overline{K}_f\dot{s})$ obtained by inverting Eq. (11.11) into the constraint (3.7), which yields:

$$J\overline{M}^{-1}(J^\top f - h + Bu - B\overline{K}_f\dot{s}) + \dot{J}\nu = 0. \tag{11.12}$$

Writing explicitly the contact wrenches from Eq. (11.12) gives:

$$f = f_m + D\overline{K}_f\dot{s}, \tag{11.13}$$

where we define f_m, D as:

$$\begin{aligned} f_m &= (J\overline{M}^{-1}J^\top)^{-1}(J\overline{M}^{-1}(h - Bu) - \dot{J}\nu) \\ D &= (J\overline{M}^{-1}J^\top)^{-1}J\overline{M}^{-1}B. \end{aligned} \tag{11.14}$$

Recall that the rate-of-change of the robot momentum Eq. (6.1) equals the net external wrench acting on the robot, and substitute the contact wrenches f obtained from Eq. (11.13) into the momentum dynamics Eq. (6.1):

$$\dot{L}(f) = J_{\mathcal{B}}^\top f_m + J_{\mathcal{B}}^\top D\overline{K}_f\dot{s} - mge_3. \tag{11.15}$$

In order to come up with a formulation similar to that of the fixed-base case, we split the joints velocity $\dot{s}$ into two components:

$$\dot{s} = -D^\top J_{\mathcal{B}} \bar{J}_G \dot{s} + (1_n + D^\top J_{\mathcal{B}} \bar{J}_G)\dot{s}, \tag{11.16}$$

where $\bar{J}_G$ is the *centroidal momentum matrix* as defined in Chapter 7 in Eq. (7.1). In particular, $\bar{J}_G$ is the mapping between the joint velocities $\dot{s}$ and the momentum, i.e. $L = \bar{J}_G \dot{s}$. Hence, Eq. (11.16) can be rewritten as:

$$\dot{s} = -D^\top J_{\mathcal{B}} L + (1_n + D^\top J_{\mathcal{B}} \bar{J}_G)\dot{s}.$$

Assuming that f_m in Eq. (11.15) can be chosen at will, we choose f_m^* as:

$$\begin{aligned} & f_m^* = f_{m1} + N_{\mathcal{B}} f_{m0}, \qquad (11.17)\\ & \text{with} \\ & f_{m1} = J_{\mathcal{B}}^{\top\dagger}\left(\dot{L}^* - J_{\mathcal{B}}^\top D \overline{K}_f (1_n + D^\top J_{\mathcal{B}} \bar{J}_G)\dot{s} + mge_3\right), \\ & \dot{L}^* = \dot{L}^d - K_P \tilde{L} - K_I I_{\tilde{L}} + T L^d, \end{aligned}$$

where we recall that $N_{\mathcal{B}} \in \mathbb{R}^{6n_c \times 6n_c}$ is a projector into the null space of $J_{\mathcal{B}}^\top$, and $f_{m0} \in \mathbb{R}^{6n_c}$ is a free variable. $T = J_{\mathcal{B}}^\top D \overline{K}_f D^\top J_{\mathcal{B}} \in \mathbb{R}^{6\times 6}$ is a symmetric and positive definite matrix. If f_m^* is chosen as in Eq. (11.17), the closed loop momentum dynamics remains:

$$\dot{\tilde{L}} + (K_P + T)\tilde{L} + K_I I_{\tilde{L}} = 0.$$

Note that the same considerations about robustness and tracking performances done for the fixed robot closed loop dynamics Eq. (11.8) can be now applied also to the closed loop momentum dynamics. To determine the control input u^* that instantaneously realize f_m^*, we make use of Eq. (11.14), which yields:

$$u^* = \overline{\Lambda}^\dagger (J\overline{M}^{-1}(h - J^\top f_m^*) - \dot{J}\nu) + \overline{N}_\Lambda u_{null} \tag{11.18}$$

with $\overline{\Lambda} = J_s \overline{M}_s^{-1} \in \mathbb{R}^{6n_c \times n}$, $\overline{N}_\Lambda \in \mathbb{R}^{n\times n}$ a projector onto the nullspace of $\overline{\Lambda}$, and $u_{null} \in \mathbb{R}^n$ a free variable. Following the control law Eq. (11.7) developed for stabilizing the joints dynamics of a fixed base robot, and recalling the choice of the joint torques redundancy τ_0 as in Eq. (6.7)–(7.4), we design u_{null} as follows:

$$u_{null} = h_s - J_s^\top f + \overline{K}_f \dot{s}^d + u_0, \tag{11.19}$$

with $u_0 := \overline{M}_s \ddot{s}^d - K_P^s \overline{N}_\Lambda \overline{M}_s \tilde{s} - K_D^s \overline{N}_\Lambda \overline{M}_s \dot{\tilde{s}}$. The additional term $\overline{K}_f \dot{s}^d$ may help to improve also the tracking of the postural task references. Asymptotic stability of the closed loop system Eq. (11.11)–(11.17)–(11.18)–(11.19) and the effectiveness of the control algorithm in improving the momentum tracking performances are then verified experimentally.

11.2.4 Quadratic programming optimization

The QP problem associated to (11.14)–(11.18)–(11.19) can be stated as:

$$f_m^* = \underset{f_m}{\operatorname{argmin}} |u^*(f_m)|^2 \tag{11.20a}$$

$$s.t.$$

$$f = f_m + D\overline{K}_f \dot{s} \tag{11.20b}$$

$$C_f f < b_f \tag{11.20c}$$

$$\dot{L}(f) = \dot{L}^* \tag{11.20d}$$

$$u^*(f_m) = \underset{u}{\operatorname{argmin}} |u - u_{null}|^2 \tag{11.20e}$$

$$s.t.$$

$$\dot{J}(q,\nu)\nu + J(q)\dot{\nu} = 0 \tag{11.20f}$$

$$\dot{\nu} = \overline{M}^{-1}(Bu + J^\top f - h - B\overline{K}_f \dot{s}) \tag{11.20g}$$

$$u_{null} = h_s - J_s^\top f + \overline{K}_f \dot{s}^d + u_0. \tag{11.20h}$$

We choose the wrench redundancy f_{m0} to minimize the norm of the control input u. The relation between f and f_m is also added as equality constraint in (11.20b).

11.2.5 Inner control loop

The inner control loop described by Eq. (2.2)–(2.3) must be modified in order stabilize the new input u towards the reference u^*. More specifically, recall that $u = \Gamma^{-\top}\tau_m$. Then, we choose τ_m^* as:

$$\tau_m^* = \Gamma^\top (u^* - K_I \int_0^t (u - u^*)\, dt), \tag{11.21}$$

that yields to the following closed loop dynamics for the control input: $u = u^* - K_I \int_0^t (u - u^*)\, dt$.

11.3 Experimental Results

We tested the control algorithms presented in this Chapter on the iCub humanoid robot, with 23 degrees of freedom. The inner control loop runs at 1 kHz, while the balancing controller runs at 100 Hz. During all the experiments we only considered the effect of the viscous friction in the harmonic drive gearboxes, that on iCub gives the major contribution to friction effects while the robot is moving. This implies that in the experiments $k_{c_{(i)}} = 0\ \forall i$.

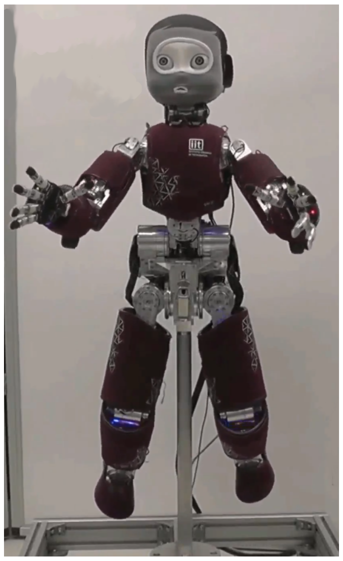


Fig. 11.1 The robot iCub fixed on a pole.

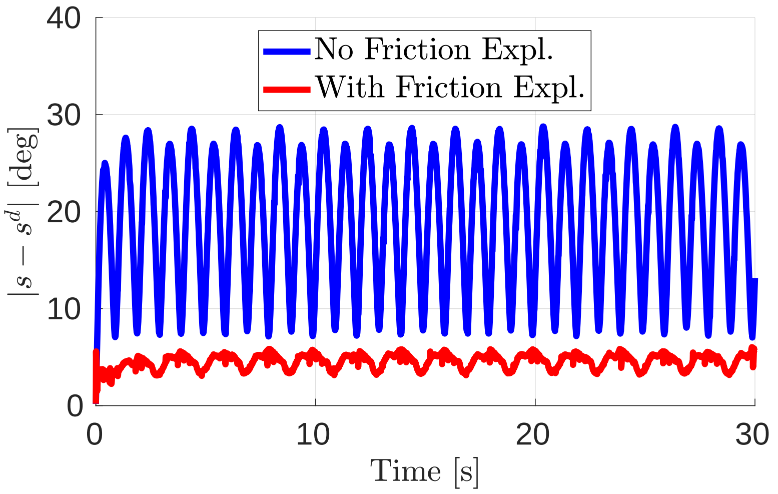


Fig. 11.2 Norm of joint position error with the fixed robot. The control law that exploits friction shows better tracking performances.

11.3.1 Joints tracking on a fixed based robot

The first experimental setup is carried out with the robot *pelvis* fixed on a pole as in Fig. 11.1. A sinusoidal reference trajectory of amplitude 15° and frequency 0.5 Hz is applied to each controlled joint. We evaluated the performances of the control

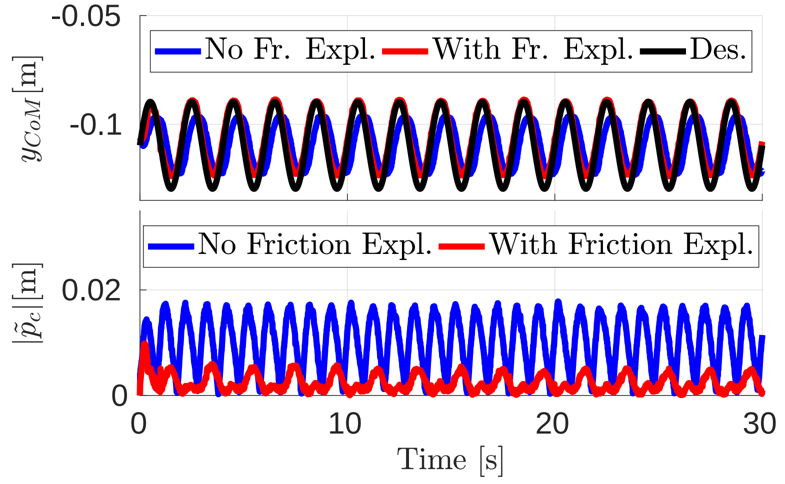


Fig. 11.3 The upper figure shows the CoM reference trajectory versus the CoM measured position, while the lower figure is the error norm of the CoM position. The control law that exploits friction shows better tracking performances.

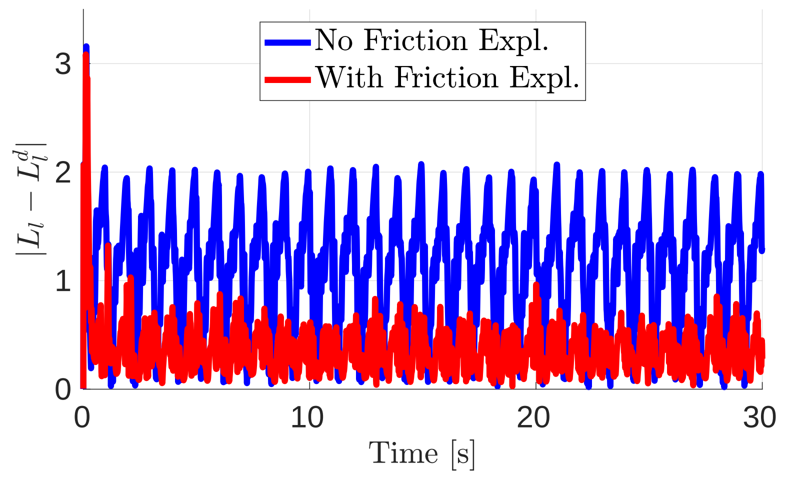


Fig. 11.4 Linear momentum tracking error while balancing. The control law that exploits friction shows better tracking performances.

algorithm Eq. (11.7) and of a standard *computed torque* control that compensates for the joints friction in the inner control loop. Fine tuning of the control gains has been performed in order to achieve the best possible tracking performances. Fig. 11.2 shows the norm of the joints position errors while executing the task. Despite this is not a proper performances comparison between the two controllers, it is possible to observe that the tracking performances of the default controller are limited by the small range of derivative gains that can be chosen without affecting the system stability. The control law that exploits friction allows to achieve good tracking performances without the need of increasing the derivative gains.

11.3.2 Momentum tracking on a floating based robot

The second experiment is carried out with the robot balancing on its feet. The robot moves its CoM from the left to the right foot, following a sinusoidal trajectory of amplitude 4 cm and frequency 0.5 Hz. The center of mass trajectory can be tracked by controlling the robot's linear momentum dynamics. We evaluated the performances of the control laws (6.5)–(6.6)–(6.7) and (11.17)–(11.18)–(11.19). Fig. 11.3 represents both the CoM error norm and the reference CoM trajectory signal versus the measured CoM position. Fig. 11.4 represents the linear momentum error norm during left and right movements. As for the fixed base robot experiment, the tracking performance of the default controller is affected by the limited choice of the gain that multiply the momentum error $\tilde{L}$. The control law that exploits friction shows instead better tracking performances.

11.3.3 The contribution of motors reflected inertia

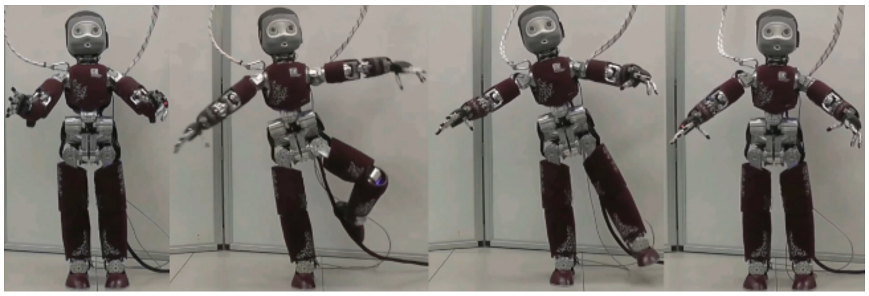


Fig. 11.5 The robot iCub performing highly dynamic movements while balancing.

For the humanoid robot iCub the condition number of the joint space mass matrix is $\text{cond}(M_s) \approx 20000$. With the addition of motors reflected inertia, the condition number of the mass matrix decreased to $\text{cond}(\overline{M}_s) \approx 800$. The increased numerical stability of the control algorithm Eq. (11.17)–(11.18)–(11.19) allowed to perform very fast dynamic movements while balancing as depicted in Fig. 11.5, that are difficult to achieve with the default controller Eq. (6.5)–(6.6)–(6.7).

Chapter 12

Momentum Jerk Control

Nonlinear controllers for floating base systems in contact with the environment are often framed as quadratic programming (QP) optimization problems. Common drawbacks of such QP based controllers are: the friction cone constraints are approximated with a set of linear inequalities; the control input often experiences discontinuities; no force feedback from Force/Torque (FT) sensors installed on the robot is taken into account. This Chapter attempts at addressing these limitations through the design of *jerk* controllers. These controllers assume the rate-of-change of the joint torques as control input, and exploit the system position, velocity, accelerations, and contact wrenches as measurable quantities. The key ingredient of the presented approach is a one-to-one correspondence between free variables and the manifold defined by the contact stability constraints. This parametrisation allows us to transform the underlying constrained optimisation problems into one that is unconstrained. Then, we propose a *jerk* control framework that exploits the proposed parametrisation and uses FT measurements in the control loop. Furthermore, we present Lyapunov stable controllers for the system momentum in the *jerk* control framework. The approach is validated with simulations and experiments using the iCub humanoid robot.

12.1 Rationale

A common approach for controlling floating base systems is the *stack-of-task* approach, which usually considers several control objectives organized in a hierarchical or weighted prioritization. This approach characterizes the momentum-based controllers designed in previous Chapters 6–11. More precisely, let a^* be a desired acceleration that the system should achieve. Then, a single priority[1] stack-of-task can be represented by the following optimisation problem:

$$\underset{y=(\dot{\nu},f,\tau)}{\text{minimize}} \ |Ay - a^*|^2 \tag{12.1}$$

subject to:

$$\begin{bmatrix} M(q) & -J^\top & -B \\ J & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\nu} \\ f \\ \tau \end{bmatrix} = \begin{bmatrix} -h(q,\nu) \\ -\dot{J}\nu \end{bmatrix} \tag{12.1a}$$

$$f \in \mathcal{K} \tag{12.1b}$$

with A a proper projection matrix, and $\mathcal{K}$ the manifold given by the contact stability constraints (3.8). The equality constraints (12.1a) take into account the relations between the optimization variables $\dot{\nu}, f, \tau$, i.e. the system dynamics (3.4) and the contact constraints (3.7).

The above optimisation problem is usually framed as a Quadratic Programming (QP) one, and its solutions may suffer from the following limitations:

1. The friction cone manifold (3.8b) is approximated with a set of linear inequalities in order to frame the optimisation problem (12.1) as a QP;

2. The optimal solution may be discontinuous, e.g during contact switching or after sharp variations of the reference trajectory;

3. The closed-loop dynamics does not include any feedback term from the measured contact wrenches f^m.

Limitation $1)$ does not usually have a strong impact on experimental activities, although it does remain an approximation of the static friction properties. Limitation $2)$ is often addressed by approximating the *continuity property* with a set of inequality constraints to be added to (12.1), but the effectiveness of this approach is often unsatisfactory from the experimental standpoint [Dafarra et al., 2018]. Limitation $3)$ is the most critical one, since FT sensor information are not used in the optimal control law τ that solves (12.1), thus potentially wasting important feedback information at the control level.

Let us observe that Limitation $3)$ may be attenuated when desired force tasks are added to the problem (12.1). For instance, if we assume to achieve a desired force f^d, then the force task can be obtained by adding equality constraints in the

[1]When several priorities are defined into the optimisation problem, higher priority tasks can be defined as constraints of (12.1).

form $f = f^d$ to the problem (12.1). At this point, one may attempt at using the FT measurements by replacing f^d with

$$f^* = f^d - K_I \int_0^t (f^m - f^d)dt,$$

where f^m is the measured force, and $f = f^*$ being the equality constraint. The main limitation of this approach is that this equality constraint may require f to violate the constraint $f \in \mathcal{K}$. Putting the desired force as part of the cost function of (12.1) may be an option, but this alters the priorities that the force task has over the acceleration one.

What follows presents an alternative, theoretically sound approach that aims at addressing the above limitations (1, (2, and (3 of classical stack-of-task approaches for the control of floating base systems in contact with the environment.

12.2 A Contact-Stable Wrench Parametrization

Parametrisations can be an effective way to transform constrained optimisation problems into unconstrained ones [Charbonneau et al., 2016]. Consider, for instance, the following optimisation problem:

$$\begin{aligned} &\underset{y}{\text{minimize}} \; \text{cost}(y) \\ &\text{subject to} \\ &\qquad y > 0, \end{aligned} \tag{12.2}$$

with $\text{cost}(\,\cdot\,) : \mathbb{R} \to \mathbb{R}$, and $y \in \mathbb{R}$. If there exists a solution to (12.2), the process of seeking this solution is equivalent to solving the following problem:

$$\underset{\xi}{\text{minimize}} \; \text{cost}(e^{\xi}) \tag{12.3}$$

with $\xi \in \mathbb{R}$. For this specific case, it is trivial to find a parametrisation ensuring $y > 0$. Note, however, that the mapping $y = e^{\xi}$ is one-to-one, and its gradient is always invertible, namely $\frac{\partial}{\partial \xi}(e^{\xi}) = e^{\xi} \neq 0 \; \forall \xi$. These two additional properties are of particular importance for the numerical stability of solvers addressing the problem (12.3).

Next Lemma proposes a wrench parametrisation that may be used to remove the constraint $f \in \mathcal{K}$ from the optimisation problem (12.1). The parametrisation also catches completely the friction cones (3.8b), thus avoiding the approximation of these cones as a set of inequality constraints.

Lemma 6 *Consider the k-th external wrench $f^k = [f_x,\ f_y,\ f_z,\ M_x,\ M_y,\ M_z]^\top$, and recall the contact stability constraints for rigid contacts Eq.* (3.8)*. Then, consider the parametrization $f^k = \phi(\xi)$, with*

$$\phi(\xi) := \begin{bmatrix} \mu_c \frac{\tanh(\xi_1)\,(e^{\xi_3}+f_z^{min})}{\sqrt{1+\tanh^2(\xi_2)}} \\ \mu_c \frac{\tanh(\xi_2)\,(e^{\xi_3}+f_z^{min})}{\sqrt{1+\tanh^2(\xi_1)}} \\ e^{\xi_3} + f_z^{min} \\ (\delta_y \tanh(\xi_4) + \delta_{y0})\,(e^{\xi_3} + f_z^{min}) \\ (\delta_x \tanh(\xi_5) + \delta_{x0})\,(e^{\xi_3} + f_z^{min}) \\ \mu_z \tanh(\xi_6)\,(e^{\xi_3} + f_z^{min}) \end{bmatrix}, \qquad (12.4)$$

$f_z^{min} > 0$ the minimum magnitude of the vertical force f_z,

$$\delta_x := \frac{x_c^{max} - x_c^{min}}{2},\ \delta_{x0} := \frac{x_c^{max} + x_c^{min}}{2}$$
$$\delta_y := \frac{y_c^{max} - y_c^{min}}{2},\ \delta_{y0} := \frac{y_c^{max} + y_c^{min}}{2},$$

with μ_c, μ_z the static friction coefficients, and x_c^{max}, y_c^{max} and x_c^{min}, y_c^{min} the contact surface dimensions as described in Fig. 3.5. Then, these properties hold:

1. *The contact constraints* (3.8) *are always satisfied, i.e., $\phi(\xi) \in \mathcal{K},\ \ \forall \xi \in \mathbb{R}^6$.*

2. *The function $\phi(\xi) : \mathbb{R}^6 \to \mathcal{K}$ is a bijection, namely, a one-to-one correspondence from $\mathbb{R}^6$ to $\mathcal{K}$.*

3. *Let the gradient of the function $\phi(\xi)$ be*

$$\Phi(\xi) := \left[\frac{\partial \phi}{\partial \xi_1}, \dots, \frac{\partial \phi}{\partial \xi_6}\right] \in \mathbb{R}^{6\times 6}. \qquad (12.5)$$

Then, $\Phi(\xi)$ is an invertible matrix $\forall\, \xi \in \mathbb{R}^6$.

The proof is in the Appendix E. Lemma 6 shows that there exists a one-to-one correspondence between the manifold $\mathcal{K}$, i.e. the set defined by (3.8), and a set of free parameters ξ. Clearly, one may find other functions for which the contact constraints (3.8) are always satisfied. But the proposed function $\phi(\,\cdot\,)$ in (12.4) has an image that corresponds to the set $\mathcal{K}$ *uniquely* and *completely*. Let us also observe that the gradient of this function is invertible for any value of the parameter ξ. This property will be of pivotal importance in Sections 12.3 and 12.4 when designing stable controllers for floating base systems.

The parametrization (12.4) can be easily extended in case of n_c distinct contact wrenches. In this case, define:

$$f = [f^1, ..., f^{n_c}]^\top := [\phi(\xi^1), ..., \phi(\xi^{n_c})]^\top, \tag{12.5a}$$
$$\dot{f} = \Phi(\xi)\dot{\xi} \tag{12.5b}$$

where $\Phi = \text{blkdiag}(\Phi_1, ..., \Phi_{n_c}) \in \mathbb{R}^{6n_c \times 6n_c}$ and $\xi = [\xi^1, ..., \xi^{n_c}]^\top \in \mathbb{R}^{6n_c}$. It is then straightforward to verify that the properties described in Lemma 6 are retained even in case of multiple contact wrenches.

12.3 Jerk Control

This Section proposes control laws that exploit the contact wrench parametrisation (12.4) and attempt to address the limitations – listed in Section 12.1 – of the classical torque-based controllers framed as stack-of-tasks optimisation problems.

12.3.1 Jerk control with parametrised contact wrenches

The wrench parametrisation (12.4) can be used to remove the constraint $f \in \mathcal{K}$ from the optimisation problem (12.1). This process would lead to the formulation:

$$\underset{y=(\dot{\nu},\xi,\tau)}{\text{minimize}} \ |Ag(y) - a^*|^2 \tag{12.6}$$
subject to:
$$\begin{bmatrix} M & -J^\top & -B \\ J & 0_{6n_c} & 0_{6n_c,n} \end{bmatrix} \begin{bmatrix} \dot{\nu} \\ \phi(\xi) \\ \tau \end{bmatrix} = \begin{bmatrix} -h(q,\nu) \\ -\dot{J}\nu \end{bmatrix}$$

with $g(y) := (\dot{\nu}, \phi(\xi), \tau)^\top$. The main drawbacks of the above approach are: $i)$ the optimisation problem (12.6) can no longer be casted in a QP being the parametrisation $\phi(\xi)$ nonlinear; we would then need nonlinear – and often slower than QPs – optimisers to solve (12.6); $ii)$ the limitations (2 and (3 listed in Section 12.1 are still not addressed.

To include feedback terms into the control laws, the contact wrenches, or accelerations, shall become part of the system state. In the language of Automatic Control, we shall then proceed with augmenting the relative degree of the output (or task) that one wants to stabilise [Isidori, 2013].

More precisely, assume that: $hp{-}i)$ the control objective is the stabilisation of a desired jerk $\dot{a}^*$; $hp{-}ii)$ the joint torque rate-of-change $\dot{\tau}$ can be considered as a control input; $hp{-}iii)$ both the joint torques τ and the contact forces f are measurable quantities, so the robot acceleration $\dot{\nu}$ – if not measurable – can be obtained from the dynamics Eq. (3.4). Now, define

$$D := \begin{bmatrix} M & -J^\top & -B \\ J & 0_{6n_c} & 0_{6n_c,n} \end{bmatrix}, \tag{12.7a}$$

$$\beta := \begin{bmatrix} -h(q,\nu) \\ -\dot{J}\nu \end{bmatrix}. \tag{12.7b}$$

As a consequence of $hp{-}iii)$, one has a measurement of the vector $y = (\dot{\nu}, f, \tau)$, while the variable $\dot{y}$ can be used as a search variable. Then, control laws for $\dot{\tau}$ that contain feedback information from FT sensor are obtained as an outcome of the following optimisation problem:

$$\underset{\dot{y}=(\ddot{\nu},\dot{f},\dot{\tau})}{\text{minimize}} \; |\dot{A}y + A\dot{y} - \dot{a}^*|^2 \tag{12.8a}$$

subject to:

$$\dot{D}y + D\dot{y} = \dot{\beta}$$

$$f \in \mathcal{K}. \tag{12.8b}$$

The solutions to the above problem are continuous in τ (even if $\dot{\tau}$ is discontinuous) and contain FT measurement feedback from the vector y. One of the main difficulties when solving (12.8) is given by the constraint (12.8b). Since the variable f no longer is a search variable, in fact, one cannot instantaneously choose values of the contact wrenches such that $f \in \mathcal{K}$. One may attempt at making (12.8b) satisfied by regulating the variable $\dot{f}$, which influences the wrench f at the next time stamp.

A possibility to make (12.8b) satisfied is to use the parametrisation in Lemma 6: the gradient of the parametrisation automatically enforces the fact that $f(t) \in \mathcal{K}\,\forall t$. More precisely, in view of (12.5), one has $\dot{y} = (\ddot{\nu}, \Phi(\xi)\dot{\xi}, \dot{\tau})$, which leads to the following optimisation problem

$$\underset{u=(\ddot{\nu},\dot{\xi},\dot{\tau})}{\text{minimize}} \; |\dot{A}y + APu - \dot{a}^*|^2 \tag{12.9a}$$

subject to:

$$\dot{D}y + DPu = \dot{\beta}, \tag{12.9b}$$

with P defined as:

$$P := \begin{pmatrix} \mathbf{I}_{n+6} & 0_{n+6,6n_c} & 0_{n+6,n} \\ 0_{6n_c,n+6} & \Phi(\xi) & 0_{6n_c,n} \\ 0_{n,n+6} & 0_{n,6n_c} & \mathbf{I}_n \end{pmatrix}.$$

In order to be solved at each time instant, the optimisation problem (12.9) requires the variable ξ. This variable may be retrieved from either time integration of $\dot{\xi}$, or by inverting the relationship $f = \phi(\xi)$: being the parametrisation a one-to-one correspondence (see Lemma 6), there exists a unique ξ for any value of the contact wrench f provided that the contact constraints (3.8) are satisfied. This latter way allows us to inject further information from the FT sensor measurements into the optimal control laws u solving (12.9). Note also that the matrix P is invertible thanks to the property (3 of Lemma 6. The invertibility of P clearly plays a pivotal role when solving the optimisation problem (12.9).

12.3.2 On the modeling and control requirements for jerk control

The optimal value $\dot{\tau}$ solving (12.9) may be sent directly to the robot low-level control system if it allows to set desired rate-of-changes of joint torques. This may be feasible, for instance, when the low-level robot control exploits the model between the joint torques τ and the motor currents ι, e.g. $\tau = k_\tau \iota$. More precisely, the motor currents are usually subject to electrical dynamics of the kind $\frac{d}{dt}\iota = k_\iota v$, where v is often the motor voltages to be applied to the motors – namely, the real control input. Then, it is straightforward to express the optimisation problem (12.9) so as the search variable u contains v. Let us observe, however, that this architecture in general requires high-frequency control loops (e.g. $5 - 20$ KHz) for generating the motor voltages v: these loops have to compute inverse dynamics within the short control cycle. If the control loops are not fast enough, sampling effects may be preponderant phenomena that render the assumption $\frac{d}{dt}\iota = k_\iota v$ not representative of the underlying physical dynamics. In this case, the associated control strategy resulting from (12.9) may prove to be ineffective.

Another necessary requirement for achieving jerk control is the calculation of the terms $\dot{A}$, $\dot{D}$ and $\dot{\beta}$ to solve the optimisation problem (12.9). These terms in general depend on the robot configuration space q, velocity ν, and accelerations $\dot{\nu}$, and need the derivatives of the system inverse dynamics. Besides numerical approximations for computing these terms, existing libraries nowadays provide users with the support of automatic differentiation and/or directly derivatives of inverse dynamics [Carpentier and Mansard, 2018, Andersson et al., In Press, 2018]. If some of the terms in (12.9) are not available, one may attempt at setting them equal to zero and tune the feedback control gains in $\dot{a}^*$ so as to achieve robustness against them. However, we present below a jerk control architecture that overcomes the above modeling and control limitations of the mere application of (12.9).

12.4 Momentum-Based Jerk Control

This Section proposes control laws that can be obtained from (12.9) when explicitly solved and extended for a two layer stack-of-task. These laws can also be shown to possess stability properties. Interestingly, the architecture presented below does not require the feedforward terms that depend on the inverse dynamics derivatives required by (12.9). This is achieved by loosing the continuity property of τ but retaining the continuity of the contact wrenches f.

More precisely, we assume that: the highest priority task is the stabilisation of a desired robot centroidal momentum [Traversaro et al., 2017, Orin and Goswami, 2008]; the lower priority task aims at stabilising the robot *posture* to *regulate* the system *zero dynamics* [Isidori, 2013].

Let us recall that the momentum rate-of-change equals the summation of all the external wrenches acting on the robot. In a multi-contact scenario, the external wrenches reduce to the contact wrenches plus the gravity force:

$$\begin{aligned} \dot{L} &= \sum_{k=1}^{n_c} A_k f^k - mge_3 = Af - mge_3, \qquad (12.10)\\ A &:= [A_1, \dots, A_{n_c}] \in \mathbb{R}^{6\times 6n_c},\\ A_k &= \begin{bmatrix} 1_3 & 0_3 \\ S({}^{\mathcal{I}}p_{\mathcal{C}_k} - {}^{\mathcal{I}}p_c) & 1_3 \end{bmatrix}, \end{aligned}$$

where $L \in \mathbb{R}^6$ is the robot's momentum, $A_k \in \mathbb{R}^{6\times 6}$ is the matrix mapping the k-th contact wrench to the momentum dynamics [2], ${}^{\mathcal{I}}p_{\mathcal{C}_k} \in \mathbb{R}^3$ is the origin of the frame associated with the k-th contact, and ${}^{\mathcal{I}}p_c \in \mathbb{R}^3$ is the CoM position.

Recall that the rate-of-change of the robot momentum (12.10) is related to the system accelerations (e.g. acceleration of the system center of mass). So, to derive jerk-based control laws, we need to differentiate (12.10) w.r.t. time, which writes:

$$\begin{aligned} \ddot{L} &= A\dot{f} + \dot{A}f = A\Phi(\xi)\dot{\xi} + \dot{A}f, \qquad (12.11)\\ \dot{A} &:= [\dot{A}_1, \dots, \dot{A}_k]\ \forall k = 1, \dots, f,\\ \dot{A}_k &= \begin{bmatrix} 0_3 & 0_3 \\ S({}^{\mathcal{I}}v_{\mathcal{C}_k} - {}^{\mathcal{I}}v_c) & 0_3 \end{bmatrix}. \end{aligned}$$

Note that Eq. (12.11) is linear w.r.t. $\dot{\xi}$. Thus, optimisation problems similar to (12.9) may be laid down. In particular, to obtain stabilisation of the robot momentum, one may: $i)$ consider $\dot{\xi}$ as control input – or search variable – of the momentum acceleration (12.11); $ii)$ apply feedback linearization to (12.11) in order to

[2]Note that $A = J_{\mathcal{B}}^{\top}$ when the equations of motion are written in centroidal coordinates.

impose a momentum acceleration $\ddot{L}^*$ of the form:

$$\ddot{L}^* = \ddot{L}^d - K_D \dot{\tilde{L}} - K_P \tilde{L} - K_I \int_0^t \tilde{L} dt \tag{12.12}$$

where $K_D, K_P, K_I \in \mathbb{R}^{6\times 6}$ are symmetric and positive definite matrices, L^d is the reference momentum and $\tilde{L} = L - L^d$ is the momentum error.

Observe that it is always possible to find $\dot{\xi}$ such that

$$\ddot{L}(\dot{\xi}) = \ddot{L}^* \tag{12.13}$$

because of the item (3 of Lemma 6. More precisely, the gradient Φ being always invertible ensures that the matrix $A\Phi$ in (12.11) is full rank $\forall\ \xi$. Consequently, $\dot{\xi}$ has full control authority on the momentum acceleration for any value of ξ. Clearly, one can impose (12.13) as long as ξ remains bounded.

The third order system (12.13), however, is in general very sensitive to gain tuning, as not all the possible combinations of the gain matrices guarantee stability of the associated closed-loop system. This limitation affects the controller's performances when applied to the real robot, where phenomena as modeling errors, measurements noise and external disturbances further limit the control gain choice.

12.4.1 Momentum-based jerk control laws

We propose a control algorithm alternative to *pure* feedback linearization with the goal of facilitating the gain tuning of the closed-loop dynamics. In particular, consider as control objective the stabilization of $[I, \tilde{L}, \zeta]^\top$ towards the reference values $[0, 0, 0]^\top$. Consequently, one has

$$\begin{aligned}
&I := \int_0^t \tilde{L} dt, \quad \tilde{L} = L - L^d \\
&\zeta := Af - mge_3 - \dot{L}^d + K_D \tilde{L} + K_P I,
\end{aligned}$$

whose dynamics writes:

$$\dot{I} := \tilde{L} \tag{12.13a}$$

$$\dot{\tilde{L}} := Af - mge_3 - \dot{L}^d = \zeta - K_D \tilde{L} - K_P I \tag{12.13b}$$

$$\dot{\zeta} := \dot{A} f + A\Phi(\xi)\dot{\xi} - \ddot{L}^d + K_D \dot{\tilde{L}} + K_P \tilde{L}. \tag{12.13c}$$

Then, the following result holds.

Lemma 7 *Assume that the robot makes at least one rigid contact with the environment, i.e. $n_c \geq 1$, and that $\dot{\xi}$ can be chosen at will. In particular, choose:*

$$\begin{aligned}\dot{\xi} = &(A\Phi)^{\dagger}\,[\ddot{L}^d - (K_D + 1_6)\dot{\tilde{L}} \\ &- (K_D + K_O^{-1} + K_P)\tilde{L} - K_P I - \dot{A} f] \\ &+ N_{A\Phi}\dot{\xi}_0,\end{aligned} \tag{12.14}$$

with $K_O, K_P, K_D \in \mathbb{R}^{6\times 6}$ symmetric and positive definite matrices,

$$N_{A\Phi} = (1_6 - (A\Phi)^{\dagger} A\Phi)$$

the projector in the null space of $A\Phi$, and $\dot{\xi}_0$ a free variable. Then:

- *the equilibrium point $(I, \tilde{L}, \zeta)^{\top} = (0, 0, 0)^{\top}$ is locally (globally) asymptotically stable if ξ is locally (globally) bounded, respectively.*

The proof is in the Appendix E. Lemma 7 shows that there exist control laws for the robot momentum that possess stability properties despite the constraints (3.8) on the generated contact wrenches $f(t) = \phi(\xi(t))$. These constraints remain satisfied while ensuring stability properties of the associated closed-loop system, and such a claim cannot usually be made in classical stack-of-tack approaches (12.1).

The control law (12.14) contains both feedforward and feedback terms that depend on the measured contact wrenches. It makes use, in fact, of (12.10) for computing $\dot{L}$, which depends on the measured contact wrenches. In the case of a single contact, there exists a unique control input $\dot{\xi}$ that satisfies (12.14), and consequently one has that the null space of the matrix $A\Phi$ is empty, i.e. $N_{A\Phi} = 0$. In the case of multiple contacts ($n_c > 1$), instead, infinite control inputs satisfy (12.14). We solve the associated redundancy using the free variable $\dot{\xi}_0$ to minimize the norm of the joint torques. The computation of $\dot{\xi}_0$ is detailed in Appendix E.

Let us remark again the importance of the the invertibility of the gradient Φ – see Lemma 6. This property guarantees that the matrix $A\Phi$ in (12.14) is full rank, so $\dot{\xi}$ has full control authority on the momentum acceleration for any value of ξ.

12.4.2 Computation of f, $\dot{L}(f)$ and $\Phi(\xi)$

The control input (12.14) requires: the contact wrenches f; the momentum derivative $\dot{L} = \dot{L}(f)$; and the associated variable ξ such that $\phi = \phi(\xi)$. The contact wrenches can be measured/estimated using the measurements from 6-axis FT sensors installed on the robot. Once the wrenches f are retrieved, we can compute the momentum rate of change via (12.10). The associated variable ξ can be computed by applying the parametrisation *inverse*, namely $\xi = \phi^{-1}(f)$. The inverse mapping exists provided that the measured contact wrenches remain inside the set $\mathcal{K}$ defined by (3.8). If the measured wrenches do not belong to $\mathcal{K}$ (because, e.g., measurements noise, external unmodeled disturbances, etc.), a saturation shall be applied in the calculation of the inverse mapping so that the control input ξ always remains finite.

12.4.3 Computation of the joint torques to realize $\dot{\xi}$

To realize a $\dot{\xi}$, e.g. the law in (12.14), we have to choose the *real* control input of the system properly. We assume in this Section that the control input is the joint torque τ, so we cannot impose a desired $\dot{\tau}$ instantaneously.

As mentioned in 12.3.2, a possibility for finding the joint torques is that of finding $\dot{\tau}$ realising $\dot{\xi}$, and then perform time-integration of $\dot{\tau}$ to obtain τ. This procedure, however, requires some derivatives of the inverse dynamics, which may not be always be available in practice.

For this reason, we follow here another route for finding the joint torques τ attempting to realise $\dot{\xi}$. First, we find the instantaneous relationship between the joint torques τ and the contact wrenches f. This relationship is the one presented in Chapter 6, Section 6.2, and it is obtained by substituting the state accelerations $\dot{\nu}$ from (3.4) into the constraints (3.7), which leads to:

$$JM^{-1}(J^\top f - h) + \Lambda\tau + \dot{J}\nu = 0 \tag{12.15}$$

with $\Lambda = JM^{-1}B$. Then, we proceed as follows:

- Integrate the control input $\dot{\xi}$ to obtain ξ. The initial conditions for the integrator can be calculated by measuring the initial contact forces f_0 and by applying the parametrization *inverse mapping*, i.e. $\xi_0 = \phi^{-1}(f_0)$;

- Apply the parametrization direct mapping to evaluate the wrenches f from ξ, i.e. $f = \phi(\xi)$. By doing so, note that f always satisfy the contact stability constraints;

- Retrieve the input torques τ from (12.15), which write

$$\tau = \Lambda^\dagger(JM^{-1}(h - J^\top f) - \dot{J}\nu) + N_\Lambda\tau_0, \tag{12.16}$$

where $N_\Lambda = (1_n - \Lambda^\dagger\Lambda)$ is the projector in the null space of Λ, and τ_0 is a free variable, that can be chosen to attempt the stabilization of the system's zero-dynamics as in Chapter 7:

$$\begin{aligned}\tau_0 &= h_s - J_s^\top f + u_0,\\ u_0 &= M_s\ddot{s}^d - K_P^s N_\Lambda M_s(s - s^d) - K_D^s N_\Lambda M_s(\dot{s} - \dot{s}^d),\end{aligned}$$

where we partitioned $h := (h_{\mathcal{B}}, h_s)^\top$, $h_{\mathcal{B}} \in \mathbb{R}^6$ and $h_s \in \mathbb{R}^n$; $J := [J_{\mathcal{B}}, J_s]$, $J_{\mathcal{B}} \in \mathbb{R}^{6n_c \times 6}$ and $J_s \in \mathbb{R}^{6n_c \times n}$, and because the floating base equations of motions are in centroidal coordinates, the mass matrix is:

$$M = \begin{bmatrix} M_{\mathcal{B}} & 0_{6,n} \\ 0_{n,6} & M_s \end{bmatrix},$$

with $M_{\mathcal{B}} \in \mathbb{R}^{6\times 6}$ and $M_s \in \mathbb{R}^{n\times n}$.

12.5 Simulations and Experimental Results

12.5.1 Simulation environment

The modeling and control framework presented in Sec. 12.2–12.3 is tested on the 23-DoFs iCub humanoid robot, both on the real robot and on simulations using Gazebo simulation environment. The controller is implemented in Simulink, and runs at a frequency of 100 Hz.

On the real iCub, the Simulink controller runs on an external PC and provides reference joint torques to the *inner* joint torque control loop described in Chapter 2. At the moment, iCub is not endowed with joint torques sensors. The measured joint torques are achieved by combining the FT sensors information, the joint encoders, IMU information and the robot model. We recall that iCub is endowed with 6 FT sensors, two of them located in the robot's upper body, two of them in the legs and two in the robot's feet.

12.5.2 Differences between simulation and real robot

A preliminary analysis of the robot balancing behavior with the control law (12.14)–(12.16) indicates that the proposed jerk control strategy may be particularly sensitive to bias errors on the estimated momentum rate of change $\dot{L}$, which is used as feedback in Eq. (12.14). The momentum rate of change is estimated as detailed in Sec. 12.4.2. The sources of this bias may be errors in the robot dynamic parameters, or low FT sensors accuracy. To enforce the closed loop system robustness w.r.t. errors in the estimation of $\dot{L}$, we modified Eq. (12.14) by adding a regularization term as follows:

$$\begin{aligned} \dot{\xi}^* = &(A\Phi)^\dagger \, [\ddot{L}^d - (K_D + 1_6)\dot{\tilde{L}} \\ &- (K_D + K_O^{-1} + K_P)\tilde{L} - K_P I - \dot{A}f] \\ &+ N_{A\Phi}\dot{\xi}_0 - k_e(\xi^* - \xi^d), \end{aligned} \tag{12.17}$$

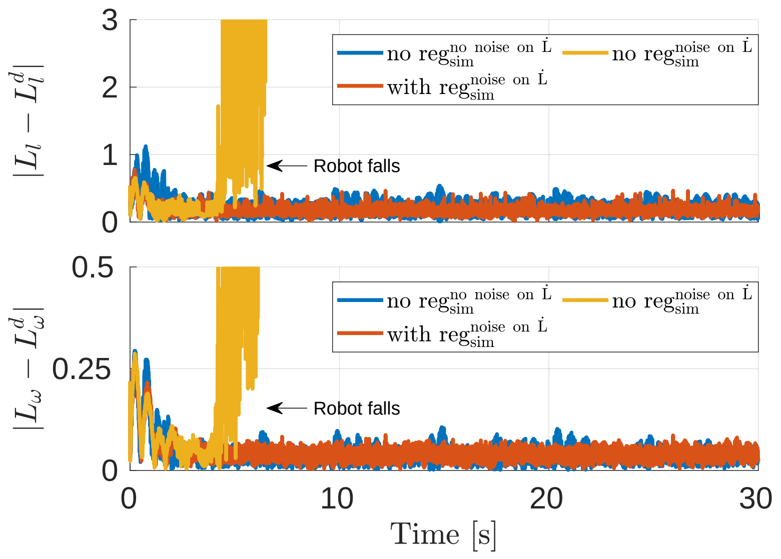


Fig. 12.1 Linear (top) and angular (bottom) momentum error norm during two feet balancing simulations when the dynamic model is overestimated by 7.5%. The robot can balances with no regularization when the momentum rate of change is not affected by error. If $\dot{L}$ is biased, the robot falls unless the regularization is used.

Robustness w.r.t. modeling errors on real robot	
Error on dynamic model $[\%]$	Success rate $[\text{trials}]$
Overest. 5%	$5/5$
Overest. 7.5%	$3/5$
Overest. 10%	$1/5$
Underest. 5%	$1/5$

Table 12.1 Robustness tests on the real iCub while performing highly dynamic movements.

where $k_e > 0$ is a positive scalar, ξ^* the integral of $\dot{\xi}^*$ and ξ^d is obtained by applying the parametrization inverse mapping on the set of wrenches satisfying the desired momentum rate of change, i.e. $\dot{L}^d(f^d)$. In case of multiple solutions, the one ensuring minimum norm of f^d is chosen.

It is important to point out that the regularization term only accounts for limitations on the FT measurements, and it is not a requirement from the theoretical point of view. To prove this statement, Figure 12.1 shows the linear and angular momentum error norm in simulation when the noise on $\dot{L}$ is due to the overestimation of model parameters M, m, C and G by 7.5%. The robot falls after few seconds if no regularization is added to Eq. (12.14) (orange line). When the regu-

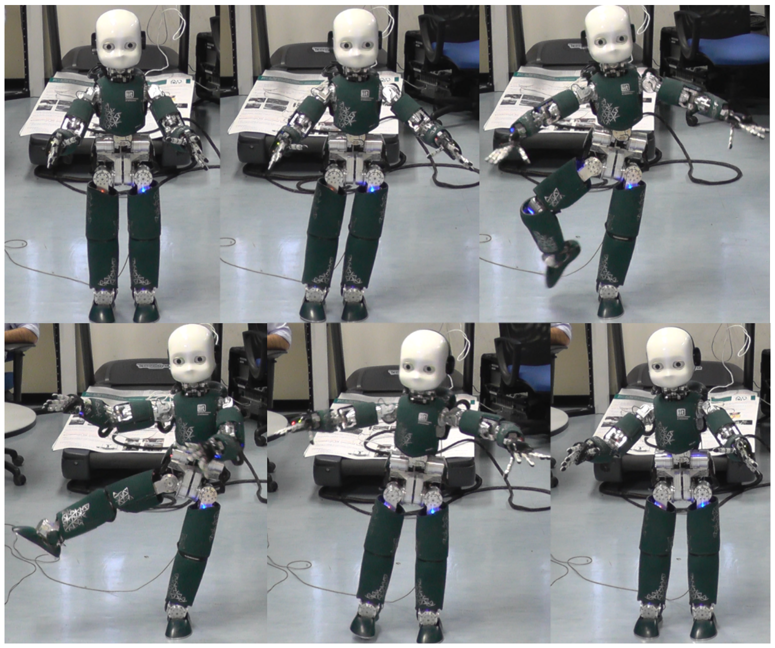


Fig. 12.2 iCub performing highly dynamics movements while balancing.

larization term is added (red line) or the noise on $\dot{L}$ is removed (blue line), stability is retained. Note that in this last case only the estimation of $\dot{L}$ is evaluated correctly, but the model errors are kept present for all the other calculations. We also carried out robustness tests on the real iCub during a contact switching scenario. Initially, the robot starts balancing on two feet, then it switches to balance on the left foot via a finite state machine, and performs highly dynamic movements on the left foot. Finally, the robot returns back to two feet balancing. The robot motion is depicted in Figure 12.2. Results are reported in Table 12.1: the robot succeeded to conclude the demo 60% of times in case of parameters overestimation by 7.5%, while dealing with underestimation seems more critical despite the presence of the regularization term.

When mounted on the robot the FT sensors accuracy is affected by several phenomena such as temperature, internal stresses and vibrations. More specifically, we observed that, even after FT sensor fine calibration, the linear forces measurements still have an offset of $\pm 2.5N$, and this offset biases the estimation of $\dot{L}$. Figure 12.3 shows the behavior of the linear and angular momentum error norm during several two feet balancing simulations and experiments. On the real iCub, the robot falls after few seconds when $k_e = 0$, as pointed out by the *green* line. When adding the regularization term as in Eq. (12.17), stability is retained and the momentum error

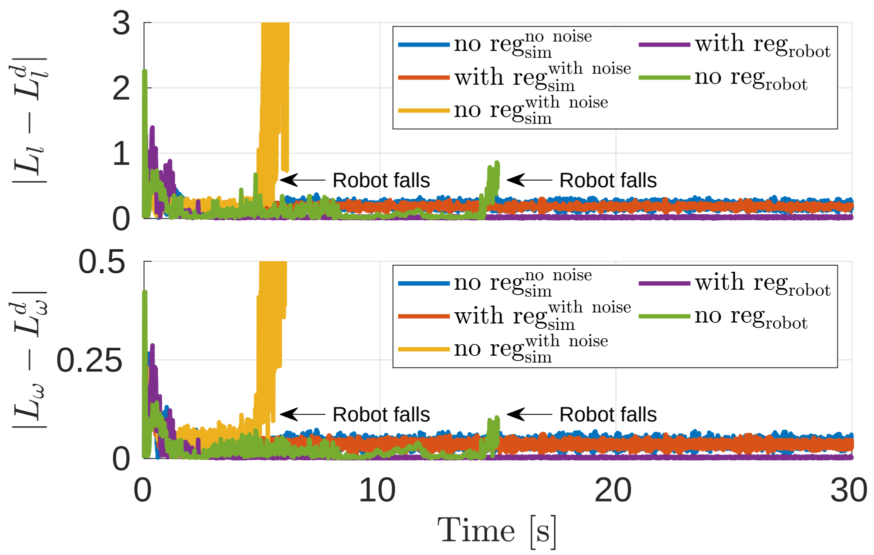


Fig. 12.3 Linear and angular momentum error norm during two feet balancing. On the real robot the additional regularization term is required, while in simulation it is not required. Adding noise on the FT sensors measurements in simulation generates a response similar to the one on the real iCub.

norm does not diverge (*purple* line). On the other hand, the *blue* line is obtained in simulation with perfect estimation of the external forces, and we set $k_e = 0$. In this case, the momentum error norm does not diverge, thus proving that the regularization term is not needed when there is no noise on the FT sensor measurements. During simulation, we injected a constant offset of amplitude $2.5\ N$ to the "measured" f_x component of one of the two contact wrenches. Results correspond to the *orange* line in Figure 12.3: as it is possible to see, stability is no more retained and the robot falls after balancing for few seconds. With the addition of the regularization term, the previous balancing performances are restored (*red* line).

12.5.3 Comparison with a momentum-based QP controller

We designed an experiment to compare the performance of the momentum-jerk controller with the momentum-based QP controller implementing the optimization problem Eq. (6.9)–(6.10) on the real iCub, during the contact switching scenario introduced in Section 12.5.2.

The two controllers have both been fine tuned for the specific demo. In Figure 12.4 we show the norm of the left foot input contact forces and moments for both the momentum-based jerk control and the momentum-based QP control. Results

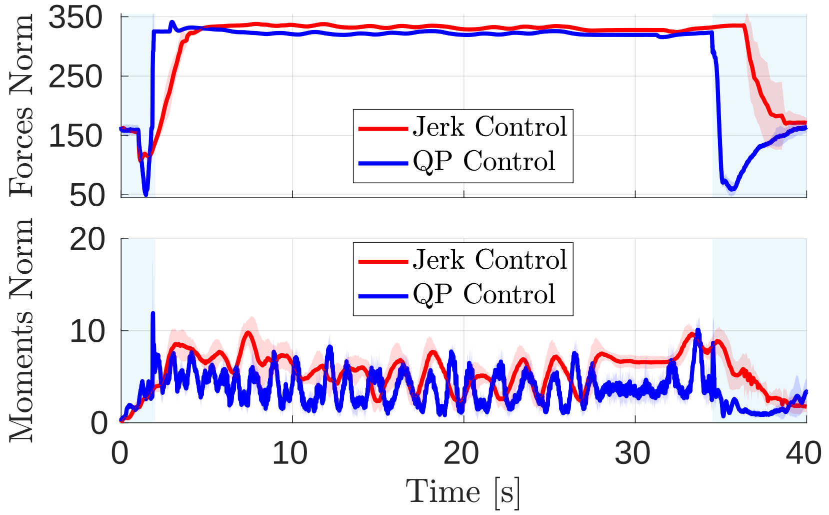


Fig. 12.4 Norm of the left foot input forces and moments (f^*). The momentum-based jerk controller (red line) helps in providing smoother references to the torque controller while switching from one foot to two feet balancing and vice-versa (blue background).

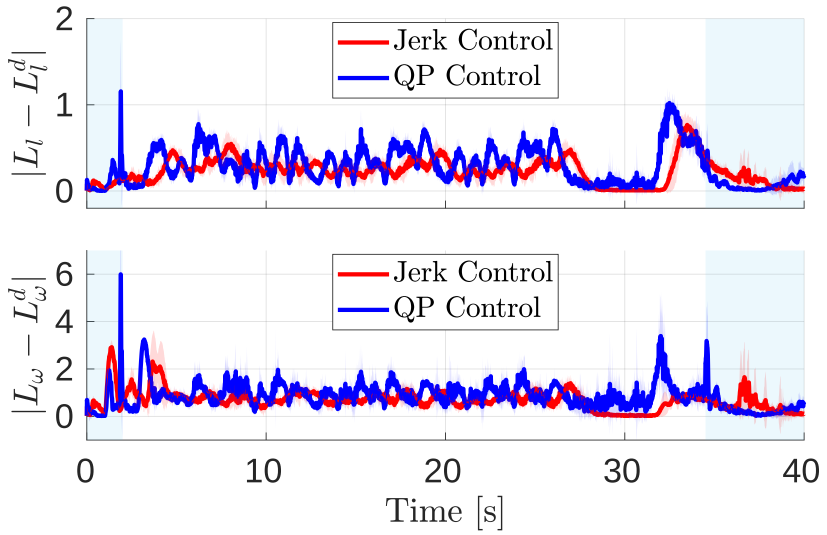


Fig. 12.5 Linear and angular momentum error norm during contact switching from two feet to one foot and vice-versa (blue background) and performing dynamic movements on the left foot (white background). The performances of the two controllers are comparable.

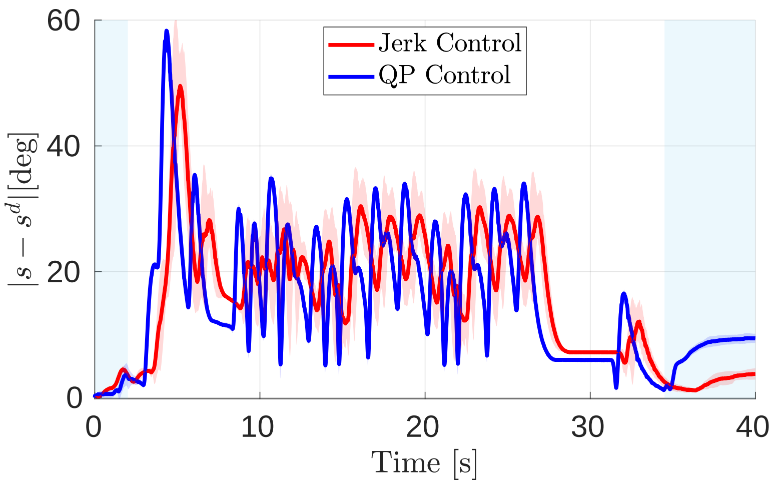


Fig. 12.6 Joint position error norm during highly dynamics movements. The plot shows that the system's zero dynamics does not diverge while achieving the primary task.

have been achieved by running 10 experiments for each control strategy. The solid lines represents the average values, while the transparent regions are the variance over the 10 experiments. The blue background represents the instants in which the robot is balancing on two feet, while the white background is when the robot is balancing on the left foot. The momentum-based jerk controller helps in providing smoother references to the torque controller during transitions. In Figure 12.5 we compared the norm of the linear and angular momentum error for the selected demo. During transition from two feet to left foot balancing, a peak of error is present when the momentum-QP control is used. The peak is caused by the sharp change in the input forces. When jerk control is used (and smoother force input is required), the peak error is reduced by 90%. During highly dynamic movements (white background), the jerk controller and momentum-based QP control show similar tracking performances.

In both controllers the input torques are calculated as in Eq. (12.16). Figure 12.6 verifies the stability of the system's *zero dynamics*: in both cases, the zero dynamics does not diverge. Convergence to zero of the joints position error is not necessarily expected as the controllers implement strict task priorities, and the *postural task* is chosen as the lowest priority task.

12.5.4 Disturbance rejection

To evaluate the robustness of the momentum-based jerk controller against unmodeled external force perturbations, we performed the following experiment: the

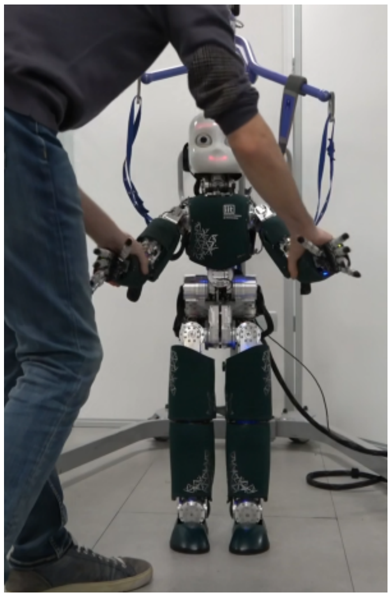


Fig. 12.7 The setup of the disturbance rejection test.

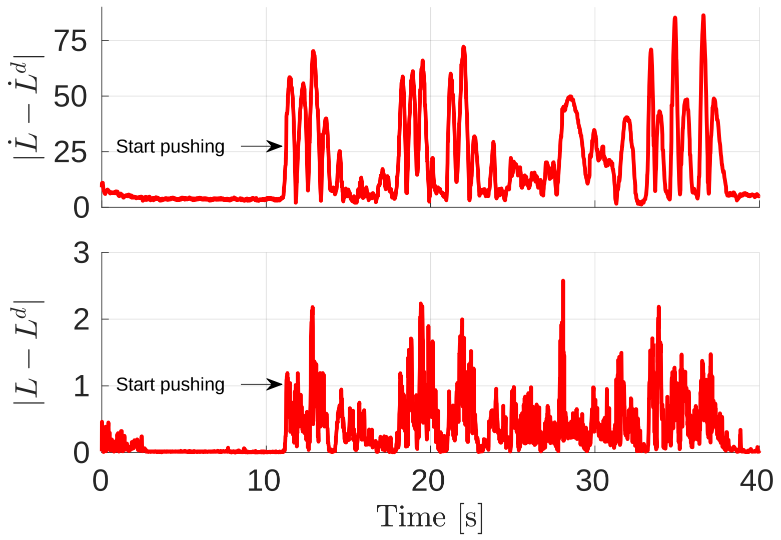


Fig. 12.8 Momentum rate of change and momentum error norm during the disturbance rejection experiment. The arrow represents the time from which the external disturbance is applied. The controller can retain stability despite the action of the external forces.

robot balances on two feet. Meanwhile, a person pushes and pulls continuously the robot's upper body as in Fig. 12.7. The applied external force is unmodeled, and it is treated as a disturbance by the momentum-based jerk controller. Figure 12.8 shows the momentum rate of change error norm and the momentum error norm during interaction. It is possible to verify that despite the high peaks of errors while the external unmodeled force is applied, the controller is still able to retain stability and when at the end the force is removed the momentum error and its rate of change still converge to a stable value. Exact convergence to zero of the momentum derivative error on the real iCub is difficult to obtain because of the low sensitivity of the FT sensors, therefore a compromise is achieved by properly tuning the corresponding feedback gains.

Part III

Optimization-Based Control Strategies for Flying

"Once you have tasted flight, you will forever walk the earth with your eyes turned skyward, for there you have been, and there you will always long to return".

Leonardo da Vinci

Despite the wide number of control techniques for achieving aerial locomotion, effective control of aerial platforms during interaction with the environment is still a challenging task. This part of the thesis designs optimization-based controllers that can address both aerial locomotion and robot interaction with the environment. More specifically, the control strategies presented in Part II of this thesis will be redesigned and adapted to the control of two aerial platforms: the aerial manipulator OTHex and the jet-powered humanoid iRonCub, both performing tasks that require robot-environment interaction.

Chapter 13

Optimization-based Control of an Aerial Manipulator

The objective of this Chapter is to design a force control strategy for aerial manipulators in physical contact with the environment. We first show that underactuation of aerial vehicles may forbid feedback linearization of the momentum rate of change, thus limiting the application of the momentum-based controller as designed in (6.4)–(6.5). Secondly, we present as alternative a task-based method for controlling aerial manipulators. The multi-task control problem, which includes hybrid force-motion tasks, energetic tasks, and position/postural tasks, is recast as a quadratic programming problem with equality and inequality constraints, which is solved online. Thanks to this method, the aerial platform can be exploited at its best to perform the multi-objective tasks, with tunable priorities, while hard constraints such as contact maintenance, friction cones, joint limits, maximum and minimum propeller speeds are all respected. An onboard force/torque sensor mounted at the end effector is used in the feedback loop in order to cope with model inaccuracies and reject external disturbances. Real experiments with a multi-rotor platform and a multi-DoF lightweight manipulator (Fig. 13.1) demonstrate the applicability and effectiveness of the proposed approach in the real world.

13.1 Modeling of Aerial Manipulators

We consider an aerial manipulator composed of an aerial platform equipped with a robotic arm as in Fig. 13.1. Given the flying nature of the system, we model it as a floating base system. We represent the robot dynamics with Euler-Poincaré

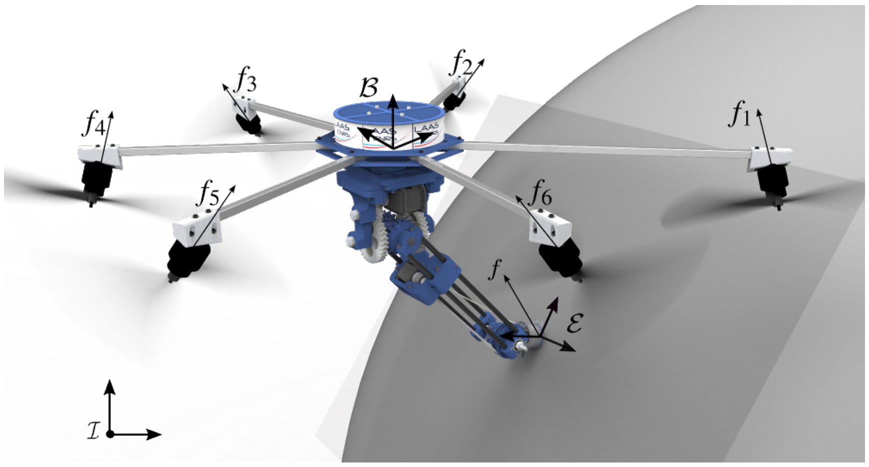


Fig. 13.1 CAD rendering of the aerial manipulator developed at LAAS-CNRS.

formalism, which yields:

$$M(q)\dot{\nu} + h(q,\nu) = \begin{bmatrix} f_P^\top & \tau^\top \end{bmatrix}^\top + J(q)^\top f, \tag{13.1}$$

where $M(q) \in \mathbb{R}^{(n+6)\times(n+6)}$ is the mass matrix, $h(q,\nu) \in \mathbb{R}^{n+6}$ accounts for the Coriolis, centrifugal and gravity effects, $f_P \in \mathbb{R}^6$ is the control wrench (forces and moments) applied to the aerial vehicle by the propellers, and $\tau \in \mathbb{R}^n$ are the joint torques of the manipulator. We assume that the interaction with the environment occurs at the manipulator end-effector only. When the robot end-effector is in contact with the environment, the external wrench $f \in \mathbb{R}^6$ has to be included in the equations. The Jacobian $J(q) \in \mathbb{R}^{6\times(n+6)}$ is the map between the system velocity ν and the linear and angular velocities of the end-effector at the contact location. Note that the formulation of the system dynamics in this Chapter is presented in its general form as in Eq. (3.1) and *not* in centroidal coordinates as in (3.4). Furthermore, it is assumed that for the applications proposed in this thesis the aerodynamics phenomena such as, e.g., wind effect, blade flapping, cross interference between propellers, and ground/wall effect do not significantly affect the robot dynamics, and therefore can be neglected. Considering the aerial vehicle actuated by a set of $n_p \in \mathbb{N}$ rigidly attached propellers, the control wrench f_P can be conveniently rewritten as a function of the propellers angular velocities. In particular we consider, as usual, a quadratic relation between propeller angular velocity and corresponding generated thrust:

$$f_P = G_w(q)\omega_P^2 \coloneqq G_w(q)(\omega_P \odot \omega_P), \tag{13.2}$$

where $\omega_P \in \mathbb{R}^{n_p}$ are the propellers angular velocities, $\odot$ is the component-wise product between two vectors, and $G_w \in \mathbb{R}^{6\times n_p}$ is the mapping between the propellers square angular velocities and the control wrench. In particular we can partition G_w in two blocks, $G_{w1} \in \mathbb{R}^{3\times n_p}$ and $G_{w2} \in \mathbb{R}^{3\times n_p}$ such that $G_w = [G_{w1}^\top \; G_{w2}^\top]^\top$. $G_{w1}^\top$ and $G_{w2}^\top$ map the propellers square angular velocities into control force and moment, respectively.

In this work, we consider both cases of 1. *under-actuated*, and 2. *fully-actuated* vehicles. The first one is the case of standard quadrotors where the propellers are all collinear and the thrust direction is fixed w.r.t. the body-fixed frame $\mathcal{B}$. On the contrary, for fully-actuated vehicles, $\mathrm{rank}(G_w) = 6$ (it has to be that $n_p \geq 6$), which means that the total thrust can change in all directions.

13.1.1 Contact modeling

We assume that the robot interacts with a *rigid* and *planar* environment.[1] In an industrial environment, such assumptions may occur while executing tasks such as polishing, inspection or welding. Possible types of interaction from the end-effector constraints point of view are:

- *Fully constrained*: the end-effector position and orientation remain always constant w.r.t. the inertial frame;
- *Only the position is constrained*: the end-effector can freely rotate, but it cannot change its position. This is the case of a single contact point;
- *Only the normal translation is constrained*: the translation in the direction normal to the contact plane is constrained, while the end-effector is free to move in the perpendicular directions. The end-effector can also rotate;
- *Rotations and the normal translation are constrained*: the end-effector position is constrained as in the previous case. However, the end-effector can rotate about the normal direction only;

We model those interactions by a set of *holonomic constraints* which describe the limitations of the end-effector motion [Ortenzi et al., 2017, de Wit et al., 1996]. Often it is easier to express those constraints w.r.t. a local reference frame $\mathcal{E}$ attached to the robot's end-effector. Differentiating the constraints w.r.t. time one has:

$$S_c \left[{}^{\mathcal{E}}v_E^\top \quad {}^{\mathcal{E}}\omega_E^\top \right]^\top = 0,$$

[1]The planarity assumption is adopted here for simplicity and could be replaced by a less stringent assumption of surface smoothness with small changes in the model and control law.

where ${}^{\mathcal{E}}v_E$ and ${}^{\mathcal{E}}\omega_E$ are the end-effector linear and angular velocities w.r.t. $\mathcal{E}$, while $S_c \in \mathbb{R}^{n_c \times 6}$ is a selector of the constrained directions of motion. $n_c \in \mathbb{N}$ represents the number of motion constraints applied to the end-effector. Note that in the local reference frame the selector matrix S_c usually remains constant during each interaction task. Using the kinematic relation, the contact constraints can be expressed as a function of the robot velocities ν:

$$S_c \bar{R} \begin{bmatrix} {}^{\mathcal{I}}v_E^\top & {}^{\mathcal{I}}\omega_E^\top \end{bmatrix}^\top = J_c \nu = 0, \tag{13.3}$$

where[2] $J_c := S_c \bar{R} J \in \mathbb{R}^{n_c \times 6+n}$, $\bar{R} = \text{blkdiag}({}^{\mathcal{E}}R_{\mathcal{I}}, {}^{\mathcal{E}}R_{\mathcal{I}})$ with ${}^{\mathcal{E}}R_{\mathcal{I}} \in SO(3)$ being the rotation matrix describing the orientation of $\mathcal{E}$ w.r.t. $\mathcal{I}$. The equations complementary to (13.3) represent instead the directions of motion of the end-effector that remain free to move, and can be written as:

$$S_f \bar{R} \begin{bmatrix} {}^{\mathcal{I}}v_E^\top & {}^{\mathcal{I}}\omega_E^\top \end{bmatrix}^\top = J_f \nu =: v_f, \tag{13.4}$$

where $J_f := S_f \bar{R} J \in \mathbb{R}^{6-n_c \times 6+n}$ with $S_f \in \mathbb{R}^{6-n_c \times 6}$ the selector matrix complementary to S_c. Rewriting the system dynamics (13.1) including (13.2) and taking into account the contact model (13.3)–(13.4) gives:

$$M\dot{\nu} + h = Bu + J_c^\top f + J_f^\top f_f \tag{13.5a}$$

$$J_c \dot{\nu} + \dot{J}_c \nu = 0, \tag{13.5b}$$

where $B = \text{blkdiag}(G_w, 1_3)$, $u = [\omega_P^{2\top} \;\; \tau^\top]^\top$, $f \in \mathbb{R}^{n_c}$ are the contact forces and/or moments, while $f_f \in \mathbb{R}^{6-n_c}$ represent forces and moments that may arise in the unconstrained directions of motion, e.g., viscous friction during motion. Equation (13.5b) is the time differentiation of (13.3) and highlights the constraints on the system accelerations $\dot{\nu}$.

13.2 Control Design

Let us denote the vector of outputs of interest (*tasks*) with $y = [y_1 \;\; \dots \;\; y_m]^\top \in \mathbb{R}^m$, where $m \in \mathbb{N}$. The control method is composed of an *outer loop* and an *inner loop*. The outer-loop assumes that a certain time-derivative for each task, defined by the symbol[3] $a = [y_1^{(r_1)} \;\; \dots \;\; y_m^{(r_m)}]^\top$, is controllable by a virtual input $a^\star$, i.e.:

$$a = a^\star. \tag{13.6}$$

[2]From now on, with the aim of compactness, we avoid to report the state dependence of some quantities as the Jacobian and mass matrix, and so on.

[3]For a given variable $x \in \mathbb{R}$, $x^{(r)}$ indicates the derivative of order r of x.

Based on this assumption any kind of stabilizing controller (PID, sliding mode, robust control, etc) can be applied by the outer loop designing $a^\star$ such that to steer y along a sufficiently smooth desired trajectory $y^d(t)$. The role of the inner-loop is instead to compute the real system inputs in order to verify as much as possible (13.6) by solving the optimization problem [Charbonneau et al., 2018]:

$$\underset{u}{\text{minimize}}\ (a - a^\star)^\top W_a(a - a^\star), \tag{13.7}$$

subjected to several constrains, e.g.: 1. input and state boundaries, 2. contact stability constraints, 3. system dynamics, etc. The (semi) positive and diagonal weight matrix $W_a \in \mathbb{R}^{m\times m}$ allows to defines *soft* priorities among all the tasks.

This technique can handle complex system dynamics and a large number of tasks. An additional advantage is the flexibility in adding and removing both tasks and constraints. As an example, if *strict* tasks priorities are required, the optimization problem can be easily modified by transforming some of the elements of the cost function into equality constraints.

The input-output asymptotic stability is obtained if it exists $u^\star$ such that $a = a^\star$, i.e., if the optimal control input is a feedback linearizing control. This implies that a careful choice of a is required. However, the definition of a may not be easy in case the robot has to perform a complex operation in a real environment. Furthermore, the presence of several constraints may prevent to find a solution that guarantees (13.6) for all the components of a. In this case, the solution will privilege the high priority tasks while keeping bounded the the error on lower priority tasks. The priority among tasks, and consequently the behavior of the robot, can be modified by properly choosing W_a.

In the following, we first show that the control of the full robot momentum as output of interest may be limited in case underactuated aerial vehicles. Then, considering a contact-based application requiring a hybrid position/force control of the end-effector, we propose an appropriate set of tasks and the implementation of the inner- and outer-loop controllers.

13.2.1 Limits in the control of robot momentum

Let us assume that the control objective is the stabilization of the robot momentum. For the sake of simplicity we consider the case in which the robot is hovering with no contact with the environment. The momentum rate of change at the center of mass is given by:

$$\dot{L} = G_{wc}(q)\omega_P^2 - mge_3,$$

where $G_{wc} = \begin{bmatrix} G_{wc1}^\top & G_{wc2}^\top \end{bmatrix}^\top$ is the matrix mapping the propellers wrench into the centroidal momentum dynamics. Then, being ω_P^2 part of the input u, we may try to select ω_P^2 in order to achieve $\dot{L} = \dot{L}^\star$, with $\dot{L}^\star$ a desired momentum dynamics. However, in case of underactuated aerial vehicles such as quadrotors, it results that $\mathrm{rank}\,(G_{wc}) < 6$, thus forbidding instantaneous feedback linearization of the momentum dynamics [Franchi et al., 2018, Nguyen et al., 2015]. This poses a limitation in the application of the momentum-based control laws (6.4)–(6.5) to the control of underactuated aerial manipulators.

The literature on aerial vehicles is rich of control solutions to overcome the underactuation of robot momentum. For example, a widely used control framework is the so-called *vectored-thrust* paradigm, that exploits the coupling between the vehicle's position and attitude dynamics. In particular, the robot's attitude together with the thrust force magnitude are considered virtual control variables to achieve a desired robot position. Then, the attitude dynamics is stabilized by applying, e.g., *backstepping* or *high gain* control approach [Hua et al., 2013, Naldi et al., 2017].

An interesting research work considers the case of a quadrotor equipped with a rigid tool [Nguyen et al., 2015]. In particular, the authors demonstrate that Cartesian control at the tool-tip can be generated if and only if the tool-tip is located strictly above or below the quadrotor's center-of-mass. Furthermore, a necessary condition for stability of the internal dynamics is that the tool-tip should be located above the quadrotor's center-of-mass.

In the following, we propose a set of tasks that can handle the control of undetactuated aerial manipulators. In view of the above literature result, we intuitively choose to substitute the control of the momentum with the control of a point located above the system's center of mass. To simplify formulation and without loss of generalities, this point has been chosen as the origin of the body-fixed frame $\mathcal{B}$.

13.2.2 Control task definition

For our case study, we define the output as $y = (p_\mathcal{B}, {}^\mathcal{I}R_\mathcal{B}, s, p_f, f, r_a)$ and the corresponding time-derivatives to be assigned by the outer loop as

$$a = \begin{bmatrix}\dot{v}_\mathcal{B}^\top & \dot{\omega}_\mathcal{B}^\top & \ddot{s}^\top & \dot{v}_f^\top & f^\top & r_a^\top\end{bmatrix}^\top .$$

The tasks $p_\mathcal{B}$, ${}^\mathcal{I}R_\mathcal{B}$ (base frame position and orientation) and s (joints configuration) represent the full aerial manipulator configuration, and can be used to cover several real world objectives, from simple changes of the overall position, to more complex coordinated maneuvers in cluttered environment. The task $p_f \in \mathbb{R}^{6-n_c}$ and its time-derivative v_f represent the end-effector positions and velocities in the unconstrained directions of motion. The role of v_f and f is to specify the hybrid force-motion behavior of the aerial manipulator perform, when in contact. Finally, the term r_a is a propeller regularization task which is defined as a function of the propeller forces and such that $r_a = 0$ when all the forces are the same. The regularization term is designed as follows:

$$r_a = D_{r_a} \omega_P^2,$$

where $D_{r_a} \in \mathbb{R}^{n_p - 1 \times n_p}$ is a matrix whose elements are 1 along the main diagonal and -1 right above the main diagonal, and all other elements are equal to zero. The goal of this task is to balance the propeller forces in order to avoid energetically unfavorable solutions where some propellers deliver almost zero thrust and others are close to saturation.[4] Notice that the virtual input associated to the tasks f and r_a are the variable themselves (zero-order time-derivative) being such tasks algebraic functions of the input.

13.2.3 Inner-loop (QP-based optimization)

Let us consider the extended input $u' = [u^\top \;\; f^\top]^\top \in \mathbb{R}^{n+n_p+n_c}$ in which the contact wrench is added to the real control inputs u. Such input extension avoids the explicit inversion of J_c and the related singularity issues that may arise in some configuration. Inverting the dynamics (13.5a) and the time derivative of (13.4) one can express a as a linear function of u':

$$a = H(q)u' + h_{bias}(q, \nu), \tag{13.8}$$

where $H(q) \in \mathbb{R}^{m \times (n+n_p+n_c)}$ and $h_{bias}(q, \nu) \in \mathbb{R}^m$ contain all the terms that do not depend on u'. More specifically, the matrix $H(q)$ and the bias vector $h_{bias}(q, \nu)$ are given by:

$$H(q) = \begin{bmatrix} M^{-1}B & M^{-1}J_c^\top \\ J_f M^{-1}B & J_f M^{-1}J_c^\top \\ 0_{(n_c \times n_p)} & 1_{(n_c \times n_c)} \\ D_{r_a} & 0_{(n_p - 1 \times n_c)} \end{bmatrix},$$

$$h_{bias}(q, \nu) = \begin{bmatrix} M^{-1}(J_f^\top f_f - h) \\ J_f M^{-1}(J_f^\top f_f - h) + \dot{J}_f \nu \\ 0 \\ 0 \end{bmatrix}.$$

To ensure the satisfaction of the contact constraint the equation (13.5b) is added to the optimization problem (13.7) as a constraint.

[4]This task is particularly important in the case that the number of propellers and their arrangement is redundant for the execution of the remaining tasks. as, e.g., in the case of an underactuated floating base with six or more propellers all pointing in the same direction.

For the considered interaction-based scenario, the inner-loop control problem (13.7) is then formulated as

$$\underset{u'}{\text{minimize}} \ (a - a^\star)^\top W_a (a - a^\star) \tag{13.9a}$$

$$\text{subject to:} \quad u_l \leq u \leq u_u \tag{13.9b}$$

$$A_f f \leq b_f \tag{13.9c}$$

$$M\dot{\nu} + h = Bu + J_c^\top f + J_f^\top f_f \tag{13.9d}$$

$$J_c \dot{\nu} + \dot{J}_c \nu = 0 \tag{13.9e}$$

The constraint (13.9b) takes into account the bounds on the control inputs, such as saturations in the joint torques and propeller velocities, while (13.9c) ensures the respect of contact stability conditions such as friction cone (approximated with linear inequalities) and positivity of the normal force. Constraints (13.9e)-(13.9d) describe the dynamics of the aerial manipulator in contact with the environment.

Observing (13.9), it is easy to verify that is an instance of a Quadratic Programming problem. In fact, the cost function and constraints are a quadratic and linear functions of the optimization variable u', respectively. The QP problem allows for a fast and efficient solution that provides online the control inputs $u^\star$, such that $u'^\star = [u^{\star\top} \ \ f^{\star\top}]^\top$ is solution of (13.9).

13.2.4 Outer-loop (desired dynamics)

Given a desired output trajectory $y^d(t)$ the task virtual inputs $a^\star$ are chosen as:

$$a^\star = \begin{bmatrix} \dot{v}_\mathcal{B}^d - K_D^b(v_\mathcal{B} - v_\mathcal{B}^d) - K_P^b(p_\mathcal{B} - p_\mathcal{B}^d) \\ \dot{\omega}_\mathcal{B}^d - K_{D\omega}^b(\omega_\mathcal{B} - \omega_\mathcal{B}^d) - K_{P\omega}^b e_{R\omega} \\ \ddot{s}^d - K_D^s(\dot{s} - \dot{s}^d) - K_P^s(s - s^d) \\ \dot{v}_f^d - K_D^f(v_f - v_f^d) - K_P^f(p_f - p_f^d) \\ f^d - K_P^c(f^m - f^d) - K_I^c \displaystyle\int_0^t (f^m - f^d)dt \\ 0 \end{bmatrix} \tag{13.10}$$

where the K_* are symmetric and positive definite matrices and the rotation error $e_{R\omega} \in \mathbb{R}^3$ is given by $e_{R\omega} = \frac{1}{2}({}^\mathcal{I}R_\mathcal{B}^\top {}^\mathcal{I}R_\mathcal{B}^d - {}^\mathcal{I}R_\mathcal{B}^{d\top} {}^\mathcal{I}R_\mathcal{B})^\vee$. Direct feedback from FT sensors measurements is used in the force control task for computing the force error $f^m - f^d$, with f^m being the measured contact force.

13.3 Results

13.3.1 Experimental setup

The control framework presented in Sec. 13.2 is tested on the *Open Tilted Hexarotor* (OTHex) [Staub et al., 2018], equipped with a three degrees of freedom serial manipulator as described in Chapter 2, Section 2.3. The manipulator's reference velocities for the velocity control loop are computed by numerically integrating the commanded joints acceleration $\ddot{s}^{\star}$, that are obtained inverting the system's dynamics (13.5) when $u' = u'^{\star}$.

All experiments are performed in an indoor arena using the Motion Capture (MoCap) system. The Control algorithm is implemented in Matlab-Simulink and runs on an external PC at a frequency of 250 Hz. The inner-loop optimization problem in (13.9) is resolved run-time by means of qpOASES solver. As it often occurs in real world applications, the hard QP equality constraints are softened by moving (13.9e) into the set of virtual inputs (13.10). This modification helps to reduce the discontinuities in the control input when the contact constraints are activated/deactivated. Contact constraints are then heavily weighted by properly designing matrix W_a, to enforce the achievement of the corresponding task in (13.10). We performed three different experiments and a simulation, that will be all detailed in the next paragraphs. The QP weights used for the different tasks during each experiment are listed in Table 13.1.

Weights for the selected tasks during experiments						
	$p_{\mathcal{B}}$	${}^{\mathcal{I}}R_{\mathcal{B}}$	s	p_f	f	r_a
Push with dist.	1	1	0.1	0	2	1e-5
Push and slide	0.1	1.5	0.025	2.5	1	1e-5
Push away	2	2	0.05	0	0	1e-5
Simulation	1	0.01	0.01	0	1	1e-5

Table 13.1 Weights of the selected tasks for the QP cost function, during the three experiments and in simulation.

13.3.2 Pushing with disturbances

During this experiment, the robot's end-effector gets in contact with a rigid surface. The robot is then required to push against the surface with a normal force of 5 N. After few seconds, a virtual force disturbance of 3.5 N is applied at the OTHex base link. The disturbance persists for 5 seconds, and it is removed after. The purpose

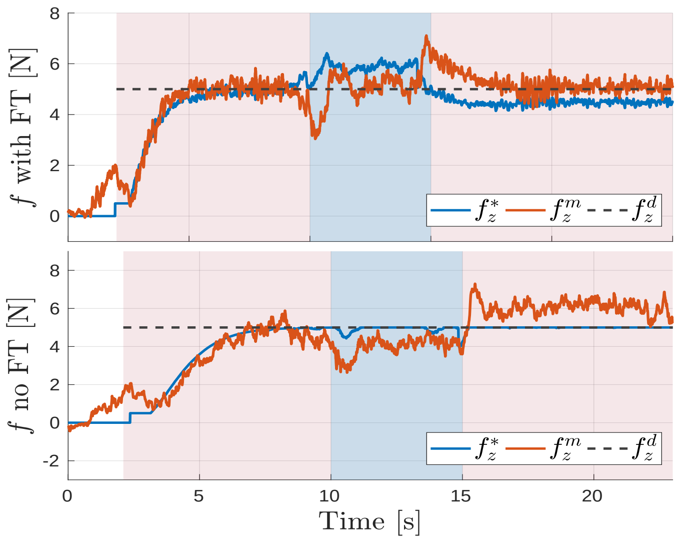


Fig. 13.2 Normal force at the end-effector when the robot is in contact. With FT sensors feedback (top plot), the measured force remains close to the desired value. With no feedback (bottom plot), the force drifts of ± 1 N.

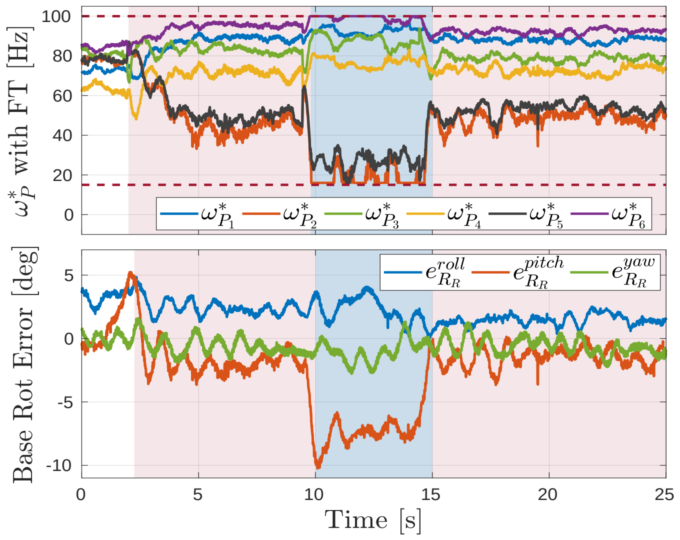


Fig. 13.3 Propellers commanded angular velocities and base rotation error during pushing with disturbance. when the disturbance is applied, the propellers saturate. The robot error along the pitch direction increases to keep small the error of higher priority tasks, such as the contact force one.

of the experiment is to understand the benefit of adding direct force feedback from the force/torque sensor when tracking a reference force in presence of external disturbances. To this purpose, we compared two different scenario: in the *'no FT'* case, the reference output force from (13.10) is computed as $f = f^d$, thus not adding any feedback from the measured contact forces. In the *'with FT'* case, the reference output force is computed including feedback terms as in (13.10).

Figure 13.2 describes the behavior of the normal force during the push with disturbance experiment for the two different cases. In particular, the plot shows the measured (f_z^m) and commanded ($f_z^\star$) normal forces during the interaction task. The dashed black line is the desired normal force. The red region denotes the period of time during which the end-effector is in contact with the surface, and the blue region is the time period when the disturbance is applied.

If feedback from force/torque sensors is used in the control law (top plot of Figure 13.2), the commanded vertical force (blue line) increases to 6N when the disturbance is applied. This due to the presence of force feedback, that attempts at compensating for the external disturbance. In fact, as a consequence of the new commanded force, the measured force f_z^m (red line) remains close to 5N after a short transient phase. When no force feedback is present instead, as in the bottom plot of Figure 13.2, the commanded force does not change magnitude and the error between measured and desired force increases up to $1\ \text{N}$. We recall that in this second case the force/torque sensor information is only used as ground-truth, but is not actively employed in the control algorithm.

The top plot of Figure 13.3 shows the frequency of the propellers commanded angular velocities $\omega_P^\star$ during the experiment with FT sensor feedback. When the external force is applied, three propellers reach saturation, which is represented in the figure by the dotted horizontal lines. Therefore, the solution $a = a^\star$ cannot be achieved anymore, and the QP penalizes tasks with lower priority, such as the OTHex orientation, in the attempt of maintaining a small error on the higher priority task. This effect is visible in the bottom plot of Figure 13.3, where the error along the pitch angle of the OTHex increases up to 7° during the saturation phase.

13.3.3 Push and slide

In this experiment, the robot pushes against the surface with a normal force of $2.5\ \text{N}$, while sliding along the surface following a reference end-effector trajectory. We made use of FT sensors feedback for improving the tracking of the normal force, and the FT sensor information is also used for compensating the viscous friction forces acting on the surface while sliding.

The top part of Figure 13.4 compares the measured and the desired end-effector trajectory along the contact surface. Despite some noise due to vibrations and

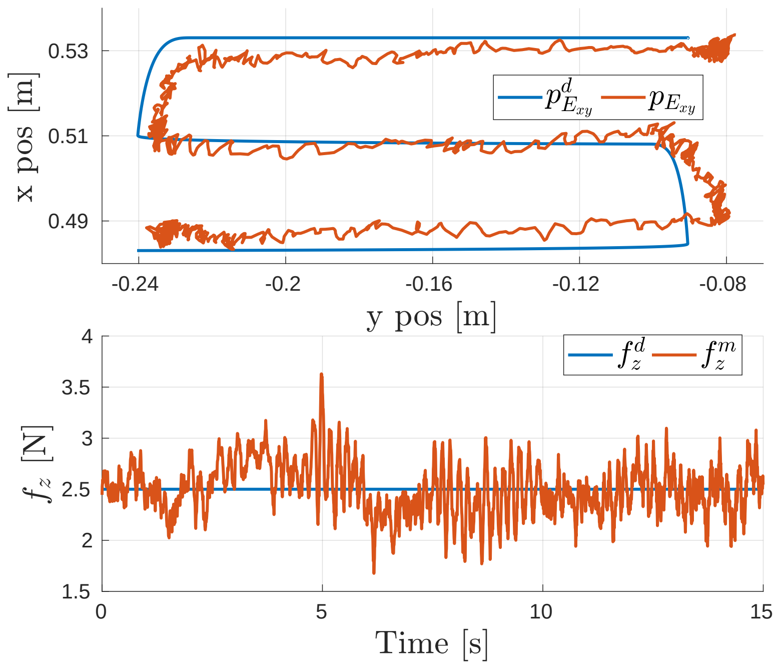


Fig. 13.4 End-effector measured and reference trajectory on the contact surface (top plot) and measured and reference normal force (bottom plot). The robot achieves good tracking performances while keep pushing with the required force.

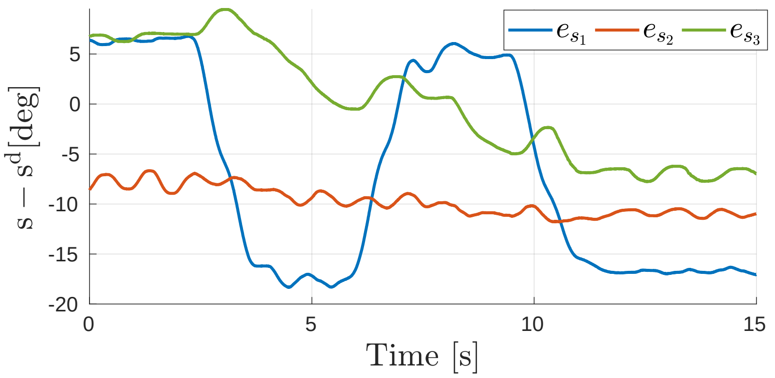


Fig. 13.5 Joints position error during push and slide task. The low-priority task is penalized to allow the achievement of higher priority tasks.

compliance of the arm, the tracking error always remains small (< 1 cm along the y direction and < 0.5 cm along the x direction). The bottom part of Figure 13.4 shows the desired and measured normal contact force during sliding. The robot is

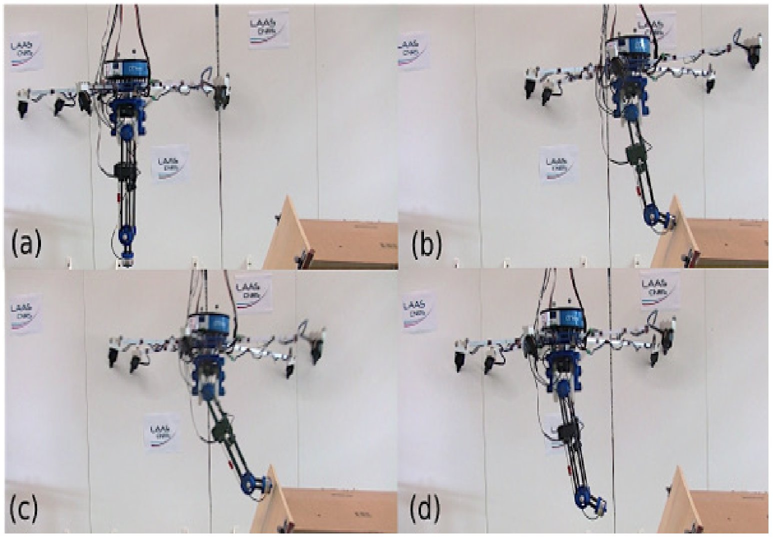


Fig. 13.6 Phases of the pushing away with contact experiment. a) approaching, b) making contact, c) pushing, d) resting.

capable of pushing with the required normal force during the whole task.

As it is possible to see in Table 13.1, during this experiment the number of active tasks is greater than the number of independent inputs. As for the previous experiment, tasks with lower priority are penalized in order to achieve tasks with higher priority. In fact, Figure 13.5 shows that the joints position error increases during the sliding task. The reference joints position is a fixed posture s^d which is not related to the end-effector references.

13.3.4 Pushing away

In this experiment the robot performs an aggressive maneuver while hovering. In particular, the OTHex is required to move horizontally of 35 cm in 0.5 seconds. The robot performs the fast movement in two scenario: while hovering with no contact with the environment, and while hovering in contact with a planar surface. In the second case, the contact is also exploited to perform the task. Figure 13.6 describes the experiment in the scenario when the robot is in contact. Figure 13.7 points out how the contact is exploited for performing the fast motion: a high normal force is requested by the controller in order to ”push away” the robot from the contact. The peak force is achieved by means of the manipulator: in particular, by commanding a high acceleration on the second arm joint (red line), which is

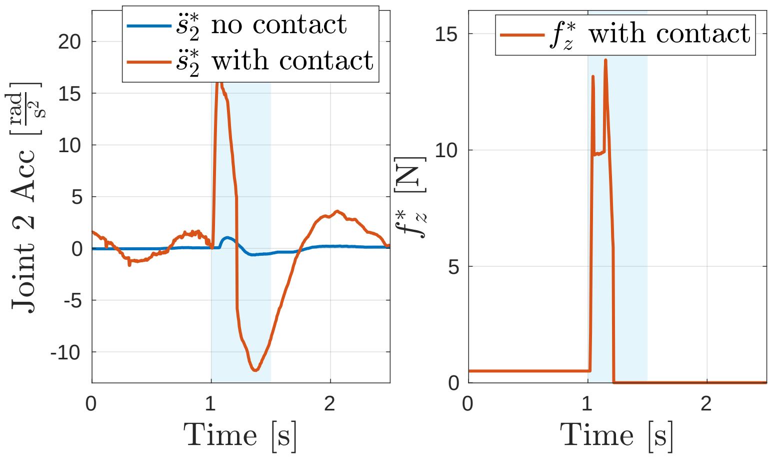


Fig. 13.7 Joint 2 acceleration (left plot) and normal force (right plot) required by the controller during push away task. When in contact, the robot exploits the contact to achieve a faster base acceleration.

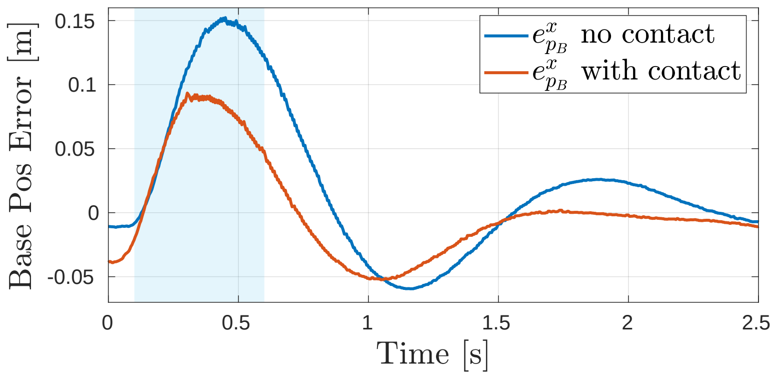


Fig. 13.8 Position error along the direction of the fast motion during push away task. Exploiting the contact reduces the peak error and allows faster convergence.

absent in the scenario in which the robot is not in contact (blue line). The blue area in the plot indicates the time at which the robot performs the fast movement. At half of the time, the contact constraint equations (13.9e)–(13.9c) are removed from the QP to allow the robot to move forward. Figure 13.8 is the position tracking error along the direction of the fast movement. Exploiting the contact allowed to reduce by 6cm the peak error and the robot stabilizes on the desired position in a shorter time. Furthermore, Figure 13.9 shows that exploiting the contact also helped in reducing the amount of saturation in the propellers angular velocities.

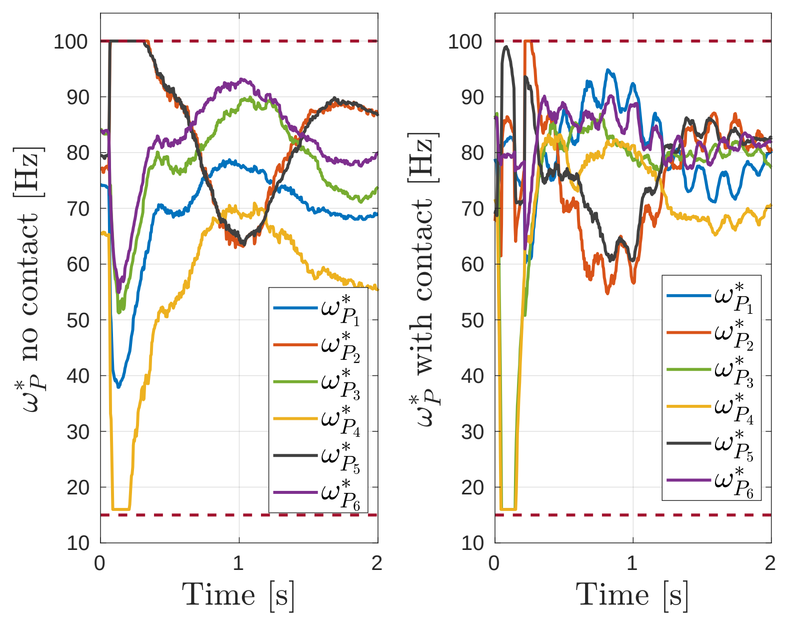


Fig. 13.9 Propellers angular velocities during push away task. Exploiting the contact helped in limiting propellers saturation.

13.3.5 Force control with underactuation

To verify if the proposed control framework can be effectively applied also for controlling underactuated aerial systems, we performed a force control task in simulation with an underactuated hexarotor equipped with a three degrees of freedom manipulator. We control the simulated robot with same Matlab-Simulink environment used for the experiments, while dynamics integration is performed by Gazebo simulator. A virtual force/torque sensor is simulated inside Gazebo and mounted on the robot's end-effector.

The task is to achieve a desired normal force of 5 N while pushing against a rigid surface. The top plot in Figure 13.10 shows that the robot is capable of achieving the required force despite the underactuation. The bottom plot depicts the propellers angular velocities. While achieving the force task, propellers still remain far from saturation. Figure 13.11 depicts instead the error on the base rotation task. Differently from the fully actuated case of Figure 13.3, where the rotation error only increased in presence of external disturbances and propellers saturation, because of underactuation the robot anyways needs to tilt of $10°$ along the pitch direction in order to be able to execute the force task.

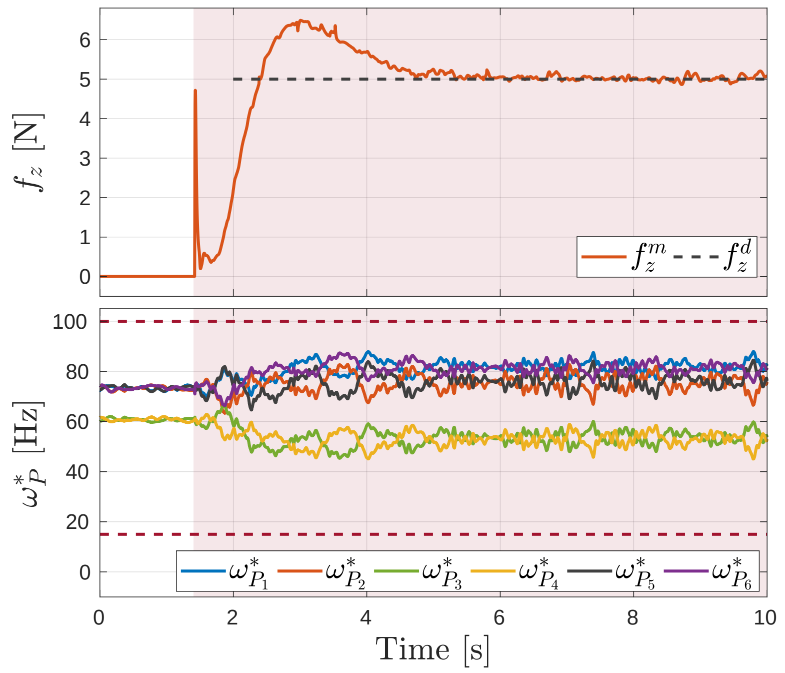


Fig. 13.10 Normal force during contact (top plot) and propellers velocities (bottom plot) in simulation. The robot can successfully achieve force control despite the underactuation.

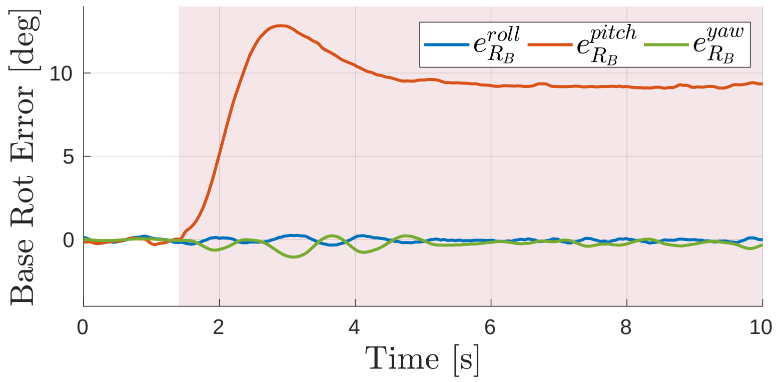


Fig. 13.11 Base rotation error while performing force control task in simulation. Because of underactuation, the robot needs to tilt in order to achieve the required contact force.

Chapter 14

Momentum Control of a Flying Humanoid Robot

Aerial manipulators combine manipulation with aerial locomotion, but they often lack of terrestrial locomotion abilities. Not being able to walk or balance may limit the efficiency of these robots in cluttered environments and in presence of adverse climatic events such as strong wind. On the other hand, humanoid robots have terrestrial locomotion and manipulation capabilites, but despite the advancements in humanoid robot control, achieving robot bipedal locomotion on rough terrains is still an open problem. A jet-powered humanoid robot is a platform conceived for addressing the above mentioned limitations. Such robot combines manipulation, terrestrial and aerial locomotion in a single robotic platform, and it can potentially operate in any condition in a large number of scenarios.

This Chapter takes on the challenge of controlling a jet-powered flying humanoid robot. Following the results achieved both with humanoid robots (Chapters 6–12) and with aerial manipulators (Chapter 13), we designed a momentum-based control framework that can handle both robot balancing and flying, and can also deal with transition maneuvers such as take off and landing. Asymptotic stability of the closed loop system during flight is demonstrated by means of Lyapunov analysis, and verified numerically during balancing. Experiments are carried out on a simulated jet-powered, joypad-controlled iCub performing balancing and flying in a disaster-like scenario.

14.1 Recalls on System Dynamics

We assume that the robot moves at relatively small velocities. This assumption renders the aerodynamic forces negligible when compared to gravity, and drag

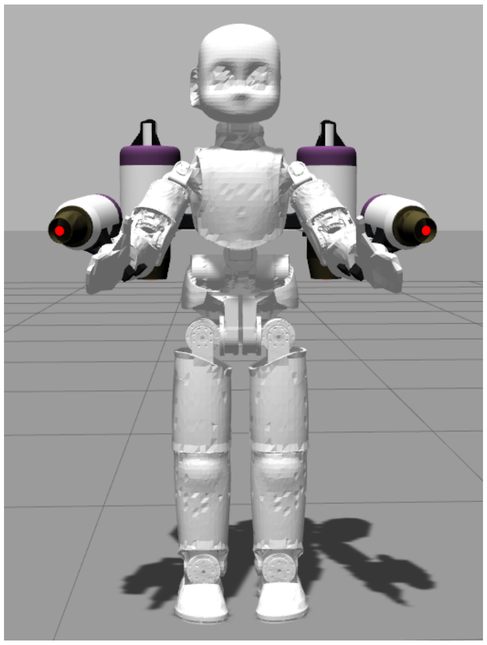


Fig. 14.1 The jet-powered iCub.

effects are but seldom taken into account. The equations of motion of the robot can be derived by applying Euler-Poincaré formalism:

$$M\dot{\nu} + h = B\tau + \alpha J^\top f + \sum_{i=1}^{n_p} J_{t_i}^\top f_{t_i}, \tag{14.1}$$

where both feet are considered in contact (i.e. $f \in \mathbb{R}^{12}$) and $\alpha \in \mathbb{R}$ is a boolean variable which is equal to 1 when the robot is balancing and 0 if the robot is flying. $f_{t_i} \in \mathbb{R}^3$ is the force applied on the robot by the $i - th$ jet. In particular, $f_{t_i} = {}^{\mathcal{I}}l_i(q)\mathrm{w}_{t_i}$, where ${}^{\mathcal{I}}l_i \in \mathbb{R}^3$ is the thrust direction, and $\mathrm{w}_{t_i} \in \mathbb{R}$ the thrust magnitude. The thrust directions move accordingly with the robot's joints, while we assume that each thrust intensity can be regulated by controlling its rate of change $\dot{\mathrm{w}}_{t_i}$. The jacobian $J_{t_i}(q) \in \mathbb{R}^{3\times n+6}$ is the map between the system's velocity ν and the linear velocity ${}^{\mathcal{I}}v_{t_i}$ of the $i - th$ thrust application point. By defining the vector of thrust magnitudes $\mathrm{w}_t := [\mathrm{w}_{t_1}, ..., \mathrm{w}_{t_{n_p}}]^\top$, we can compactly rewrite $J_t^\top \mathrm{w}_t := \sum_{i=1}^{n_p} J_{t_i}^\top {}^{\mathcal{I}}l_i \mathrm{w}_{t_i}$. In what follows, we assume that the robot is powered by $n_p = 4$ jets, two of them located on the robot arms and two of them mounted on a jet-pack attached to the iCub shoulders. The robot jets configuration is depicted in Fig. 14.1. Furthermore, we assume that a desired thrust forces magnitude can be achieved by directly controlling the rate of change $\dot{\mathrm{w}}_t$. In the control design proposed in this thesis, the jets dynamics is neglected, i.e. $u_1 := \dot{\mathrm{w}}_t$. Without loss of generality but to facilitate the control design proposed next, the formulation of the system dynamics in this Chapter is presented in its general form as in Eq. (3.1).

14.1.1 Momentum dynamics

The rate of change of the robot centroidal momentum equals the summation of all the external forces and moments acting on the system, that in the case study are the thrust forces, the contact wrenches (if present) and the gravity force:

$$\dot{L} = A_t(q)\mathrm{w}_t + \alpha A_c(q) f - mge_3 \tag{14.2}$$

where we define:

$$\begin{aligned}
A_t(q) &= \left[\bar{S}(r_1)^{\mathcal{I}} l_1,\ ...,\ \bar{S}(r_{n_p})^{\mathcal{I}} l_{n_p}\right] \in \mathbb{R}^{6\times n_p},\\
A_c(q) &= \left[\bar{S}(r_L),\ \begin{bmatrix} 0_3\\ 1_3 \end{bmatrix},\ \bar{S}(r_R),\ \begin{bmatrix} 0_3\\ 1_3 \end{bmatrix}\right] \in \mathbb{R}^{6\times 12},\\
r_i &= {}^{\mathcal{I}}p_i - {}^{\mathcal{I}}p_c,\ \forall i \in (1,\ ...,\ n_p),\\
r_{L(R)} &= {}^{\mathcal{I}}p_{L(R)} - {}^{\mathcal{I}}p_c,\\
\bar{S}(r_i) &= \begin{bmatrix} 1_3\\ S(r_i) \end{bmatrix},
\end{aligned}$$

with ${}^{\mathcal{I}}p_c \in \mathbb{R}^3$ the center of mass position, ${}^{\mathcal{I}}p_{L(R)} \in \mathbb{R}^3$ the contact location on the left (right) foot and ${}^{\mathcal{I}}p_i \in \mathbb{R}^3$ the i-th thrust application point.

14.2 Control Design

Our goal is to design a control algorithm for stabilizing the momentum dynamics along a reference trajectory $L^d(t)$. The advantage on choosing the momentum as control output w.r.t., for example, the acceleration of a generic frame $\dot{\mathrm{v}}_k = \dot{\mathrm{v}}_k(q,\nu,\tau,\mathrm{w}_t,f)$ associated to one of the robot links, resides in its independence from the joint torques τ, that clearly identifies the key role of the thrust forces regulation in the stabilization of the underactuated part of system (14.1). Now, define the momentum error $\tilde{L} = L - L^d$. At the equilibrium configuration $(\tilde{L}, \dot{\tilde{L}}) = (0, 0)$ the effect of the thrust and contact forces must oppose the gravity plus the term $\dot{L}^d(t)$, i.e.:

$$0 = A_t(q)\mathrm{w}_t + \alpha A_c(q) f - mge_3 - \dot{L}^d(t). \tag{14.3}$$

The equivalence (14.3) may be achieved by assuming that w_t and f can be chosen at will, and later obtain w_t and f by means of the real control inputs $\dot{\mathrm{w}}_t$ and τ. However, two limitations arise during the flight phase:

1. while flying, the virtual input for controlling the momentum dynamics is reduced to the thrust magnitude only, whose dimension is $n_p = 4$. Thus, the output (14.3) becomes underactuated;

2. the angular momentum of a rigid multi-body system is given by ${}^{\mathcal{G}[\mathcal{I}]}L_\omega = {}^{\mathcal{G}[\mathcal{I}]}I(q)\,{}^{\mathcal{I}}\omega_o$ where ${}^{\mathcal{G}[\mathcal{I}]}I(q) \in \mathbb{R}^{3\times 3}$ is the total robot's inertia and ${}^{\mathcal{I}}\omega_o \in \mathbb{R}^3$ is the *average* angular velocity [Orin et al., 2013]. In general ${}^{\mathcal{I}}\omega_o$ is not associated with the derivative of a rotation matrix ${}^{\mathcal{I}}R_o$ somehow representing the robot orientation in space. This is an important limitation in the control of the flight phase: to control the robot orientation while flying is fundamental, in particular when performing critical maneuvers such as landing.

To overcome the first limitation, reminiscent of the *vectored-thrust* paradigm [Hua et al., 2013], we use the robot joints to align the thrust force $f'_t := A_t(q)\mathrm{w}_t$ to the gravity and the desired momentum rate of change. To this purpose, the momentum acceleration (or center of mass jerk) $\ddot{L}$ can be proven to be linear w.r.t. the joints velocities $\dot{s}$, and these velocities can be used to influence the thrusts orientation. Therefore we make the following assumption:

Assumption 2 *the joint velocities $\dot{s} := u_2$ can be chosen at will and then considered as control inputs, as well as the thrust and contact wrench rate of change $\dot{w}_t := u_1$, $\dot{f} := u_3$.*

The second limitation is resolved by applying additional constraints on the angular momentum and by designing an orientation controller based on these constraints.

14.2.1 Control of the angular momentum

A key point for the angular momentum control is to observe that the locked angular velocity ${}^{\mathcal{I}}\omega_o$ can be expressed as a linear combination of the joint velocities and the angular velocity of one of the robot's links [Traversaro et al., 2017]. In our control design we choose as link the pelvis, which is also the heaviest link of the robot. Without loss of generality, from now on we refer to this link as the *base* link. Then, one has:

$$ {}^{\mathcal{I}}\omega_o = {}^{\mathcal{I}}\omega_{\mathcal{B}} + J^s_\omega \dot{s}, \tag{14.4}$$

where ${}^{\mathcal{I}}\omega_{\mathcal{B}}$ is the base angular velocity in the inertial frame and $J^s_\omega \in \mathbb{R}^{3\times n}$ relates joints velocity and locked angular velocity. Assumption 2 implies that the joints velocity can be chosen at will. Because of (14.4), one may choose them to instantly influence the robot angular momentum. More specifically, we perform the following change of coordinates: ${}^{\mathcal{G}[\mathcal{B}]}L_\omega := {}^{\mathcal{B}}R_{\mathcal{I}}{}^{\mathcal{G}[\mathcal{I}]}L_\omega$. The angular momentum in these *body* coordinates is related to the joint velocities as follows:

$$
{}^{\mathcal{G}[\mathcal{B}]}L_\omega = {}^{\mathcal{G}[\mathcal{B}]}\mathbb{I}\,{}^{\mathcal{B}}\omega_{\mathcal{B}} + \bar{J}^s_\omega \dot{s}, \tag{14.5}
$$

where ${}^{\mathcal{G}[\mathcal{B}]}\mathbb{I} = {}^{\mathcal{B}}R_{\mathcal{I}}{}^{\mathcal{G}[\mathcal{I}]}\mathbb{I}\,{}^{\mathcal{I}}R_{\mathcal{B}}$ is the total inertia matrix in the new coordinates, ${}^{\mathcal{B}}\omega_{\mathcal{B}}$ the angular velocity in base coordinates and $\bar{J}^s_\omega = {}^{\mathcal{B}}R_{\mathcal{I}}{}^{\mathcal{G}[\mathcal{I}]}\mathbb{I}(q)J^s_\omega(q)$. The main advantage of writing the angular momentum in this new reference frame resides in the property that the total inertia matrix only depends on the joints configuration, i.e. ${}^{\mathcal{G}[\mathcal{B}]}\mathbb{I} = {}^{\mathcal{G}[\mathcal{B}]}\mathbb{I}(s)$. The idea is now to use the joint velocities to impose a constraint on the angular momentum of the form: ${}^{\mathcal{G}[\mathcal{B}]}L_\omega = {}^{\mathcal{G}[\mathcal{B}]}L^*_\omega$.

Constraint on the angular momentum

While the robot is flying or balancing, the position of most of the joints only has *small* and *smooth* changes with respect to the initial robot configuration. This is true in particular for the joints associated to the links that mostly contributes to the robot's total inertia, such as the robot chest, pelvis and upper legs. It is possible to verify numerically that for the case study the total inertia rate of change always remains ${}^{\mathcal{G}[\mathcal{B}]}\dot{\mathbb{I}} \approx 0_3$. Therefore, we choose

$$
{}^{\mathcal{G}[\mathcal{B}]}L^*_\omega = {}^{\mathcal{G}[\mathcal{B}]}\mathbb{I}_0{}^{\mathcal{B}}\omega_{\mathcal{B}}, \tag{14.6}
$$

with ${}^{\mathcal{G}[\mathcal{B}]}\mathbb{I}_0 = {}^{\mathcal{G}[\mathcal{B}]}\mathbb{I}(t = 0)$. The constraint is in accordance with the actual robot behavior during the considered tasks, thus increasing its probabilities to remain feasible, i.e., matrix $\bar{J}^s_\omega$ always remains full rank. Note that as long as constraint (14.6) holds, the *base* angular velocity coincides with the *average* angular velocity. The angular momentum is then forced to coincide at each instant to that of a single rigid body, with inertia ${}^{\mathcal{G}[\mathcal{B}]}\mathbb{I}_0$ and the angular velocity of the base link. Consequently, the robot *overall* orientation is associated to the orientation of the base link ${}^{\mathcal{I}}R_{\mathcal{B}}$. Provided that $\bar{J}^s_\omega$ remains full rank, any joint movement that is in the *null space* of $\bar{J}^s_\omega$ does not violate the angular momentum constraint, therefore the robot can still control the thrusts orientation by moving its joints. A control design that exploits the angular momentum constraint as in (14.6) is stated next.

Control of robot orientation

After imposing the constraint (14.6), we select the angular momentum acceleration ${}^{\mathcal{G}[\mathcal{B}]}\ddot{L}_\omega = {}^{\mathcal{G}[\mathcal{B}]}\mathbb{I}_0{}^{\mathcal{B}}\ddot{\omega}_{\mathcal{B}}$ as control variable to stabilize the angular momentum and the *base link orientation* towards desired values. In particular, we are given with the following Lemma.

Lemma 8 *Consider the following system:*

$$
\begin{aligned}
{}^{\mathcal{I}}\dot{R}_{\mathcal{B}} &= {}^{\mathcal{I}}R_{\mathcal{B}}S({}^{\mathcal{B}}\omega_{\mathcal{B}}) && \textit{(14.7a)}\\
{}^{\mathcal{G}[\mathcal{B}]}\dot{L}_{\omega} &= {}^{\mathcal{G}[\mathcal{B}]}\mathbb{I}_0{}^{\mathcal{B}}\dot{\omega}_{\mathcal{B}} && \textit{(14.7b)}\\
{}^{\mathcal{G}[\mathcal{B}]}\ddot{L}_{\omega} &= {}^{\mathcal{G}[\mathcal{B}]}\ddot{L}^{*}_{\omega}, && \textit{(14.7c)}
\end{aligned}
$$

and assume that the angular momentum acceleration ${}^{\mathcal{G}[\mathcal{B}]}\ddot{L}^{*}_{\omega} \in \mathbb{R}^3$ *can be chosen at will. Apply the following control law to system* (14.7)*:*

$$
\begin{aligned}
{}^{\mathcal{G}[\mathcal{B}]}\ddot{L}^{*}_{\omega} = {}^{\mathcal{G}[\mathcal{B}]}\ddot{L}^{d}_{\omega} - (1_3 + c_0{}^{\mathcal{G}[\mathcal{B}]}\mathbb{I}_0^{-1}){}^{\mathcal{G}[\mathcal{B}]}\dot{\tilde{L}}_{\omega} + \\
- c_1{}^{\mathcal{G}[\mathcal{B}]}\mathbb{I}_0^{-1}{}^{\mathcal{G}[\mathcal{B}]}\tilde{L}_{\omega} - \dot{skv} - c_o{}^{\mathcal{G}[\mathcal{B}]}\mathbb{I}_0^{-1}skv,
\end{aligned}
\quad \textit{(14.8)}
$$

where we defined: the rotation error $skv := \frac{1}{2}(R^{d\top}{}^{\mathcal{I}}R_{\mathcal{B}} - {}^{\mathcal{I}}R_{\mathcal{B}}^{\top}R^{d})^{\vee}$, *the angular momentum error* ${}^{\mathcal{G}[\mathcal{B}]}\tilde{L}_{\omega} := {}^{\mathcal{G}[\mathcal{B}]}L_{\omega} - {}^{\mathcal{G}[\mathcal{B}]}L^{d}_{\omega}$, *and the user-defined control gains* $c_0,\ c_1 \in \mathbb{R}^{+}$. *Then, the equilibrium point*

$$
({}^{\mathcal{G}[\mathcal{B}]}\dot{L}_{\omega}, {}^{\mathcal{G}[\mathcal{B}]}L_{\omega}, {}^{\mathcal{I}}R_{\mathcal{B}}) = ({}^{\mathcal{G}[\mathcal{B}]}\dot{L}^{d}_{\omega}(t), {}^{\mathcal{G}[\mathcal{B}]}L^{d}_{\omega}(t), R^{d}(t))
$$

is locally asymptotically stable.

The proof is given in Appendix F. We are then left to relate ${}^{\mathcal{G}[\mathcal{B}]}\ddot{L}^{*}_{\omega}$ with the control variables $\dot{\mathrm{w}}_t,\ \dot{s}$ and $\dot{f}$.

Angular momentum acceleration

The momentum acceleration in centroidal coordinates is given by the time differentiation of (14.2):

$$
{}^{\mathcal{G}[\mathcal{I}]}\ddot{L} = A_t u_1 + \Lambda_s(q, f, \mathrm{w}_t)u_2 + \alpha A_c u_3 + \Lambda_{\mathcal{B}}(q, f, \mathrm{w}_t)\mathrm{v}_{\mathcal{B}}, \tag{14.9}
$$

where the control variables $u_1 = \dot{\mathrm{w}}_t,\ u_2 = \dot{s}$ and $u_3 = \dot{f}$ all appear linearly. Matrices Λ_s and $\Lambda_{\mathcal{B}}$ are obtained by rearranging the time derivatives of A_t and A_c, see Appendix F for the details. Now, let us split the momentum acceleration into linear and angular part:

$$
\begin{aligned}
{}^{\mathcal{G}[\mathcal{I}]}\ddot{L}_l &= A_{t_l}u_1 + \Lambda_{s_l}u_2 + \alpha A_{c_l}u_3 + \Lambda_{\mathcal{B}_l}\mathrm{v}_{\mathcal{B}} && (14.10a)\\
{}^{\mathcal{G}[\mathcal{I}]}\ddot{L}_{\omega} &= A_{t_\omega}u_1 + \Lambda_{s_\omega}u_2 + \alpha A_{c_\omega}u_3 + \Lambda_{\mathcal{B}_\omega}\mathrm{v}_{\mathcal{B}} && (14.10b)
\end{aligned}
$$

where we properly partitioned $A_t = [A_{t_l}^{\top},\ A_{t_\omega}^{\top}]^{\top}$, $A_c = [A_{c_l}^{\top},\ A_{c_\omega}^{\top}]^{\top}$, $\Lambda_s = [\Lambda_{s_l}^{\top},\ \Lambda_{s_\omega}^{\top}]^{\top}$ and $\Lambda_{\mathcal{B}} = [\Lambda_{\mathcal{B}_l}^{\top},\ \Lambda_{\mathcal{B}_\omega}^{\top}]^{\top}$. Recall now that the orientation control Eq.

14.8 considers the momentum in body coordinates. The angular momentum acceleration in body coordinates is related to the centroidal angular momentum by:

$$\begin{aligned}{}^{\mathcal{G}[\mathcal{B}]}\ddot{L}_\omega &= {}^{\mathcal{B}}\ddot{R}_{\mathcal{I}}{}^{\mathcal{G}[\mathcal{I}]}L_\omega + 2\,{}^{\mathcal{B}}\dot{R}_{\mathcal{I}}{}^{\mathcal{G}[\mathcal{I}]}\dot{L}_\omega + {}^{\mathcal{B}}R_{\mathcal{I}}{}^{\mathcal{G}[\mathcal{I}]}\ddot{L}_\omega = \\ &= \sigma + {}^{\mathcal{B}}R_{\mathcal{I}}{}^{\mathcal{G}[\mathcal{I}]}\ddot{L}_\omega.\end{aligned} \tag{14.11}$$

Because of the angular momentum constraint 14.6, it can be verified that the bias term σ does not depend on the control variables, i.e. $\sigma = \sigma(q, f, \mathrm{w}_t, \omega_{\mathcal{B}})$. This implies that the following equations:

$${}^{\mathcal{G}[\mathcal{B}]}\ddot{L}_\omega = \sigma + {}^{\mathcal{B}}R_{\mathcal{I}}(A_{t_\omega}u_1 + \Lambda_{s_\omega}u_2 + \alpha A_{c_\omega}u_3 + \Lambda_{\mathcal{B}_\omega}\mathrm{v}_{\mathcal{B}}) \tag{14.12}$$

can be feedback-linearized through a proper choice of the control terms $u := [u_1^\top,\ u_2^\top,\ u_3^\top]^\top$. We rewrite Eq. 14.12 as:

$$\begin{aligned}{}^{\mathcal{G}[\mathcal{B}]}\ddot{L}_\omega &= H_\omega u + h_\omega, \\ h_\omega &:= \sigma + {}^{\mathcal{B}}R_{\mathcal{I}}\Lambda_{\mathcal{B}_\omega}\mathrm{v}_{\mathcal{B}}, \\ H_\omega &:= {}^{\mathcal{B}}R_{\mathcal{I}}\left[A_{t_\omega}, \quad \Lambda_{s_\omega}, \quad \alpha A_{c_\omega}\right].\end{aligned} \tag{14.13}$$

Thus, the angular momentum control is achieved from 14.13 by choosing ${}^{\mathcal{G}[\mathcal{B}]}\ddot{L}_\omega = {}^{\mathcal{G}[\mathcal{B}]}\ddot{L}^*_\omega$, provided that matrix H_ω always remains full rank.

14.2.2 Control of the linear momentum

The control of the linear momentum ${}^{\mathcal{G}[\mathcal{I}]}L_l$ is designed similarly to the one presented in [Pucci et al., 2018] and in Chapter 12 for the momentum-jerk control framework. More specifically, let us define the state variable:

$$\begin{aligned}\tilde{\xi} &:= A_{t_l}\mathrm{w}_t + \alpha A_{c_l}f + F, \\ F &:= -mge_3 - \dot{L}^d_l + K^l_D\tilde{L}_l + K^l_P I(t)_l\end{aligned}$$

with $K^l_P, K^l_D \in \mathbb{R}^{3\times3}$ two symmetric and positive definite matrices. The variable $I(t)_l$ represents the integral of the linear momentum error $\tilde{L}_l$. Then, consider the following system of equations:

$$\dot{I}_l = \tilde{L}_l \tag{14.14a}$$

$$\dot{\tilde{L}}_l = \tilde{\xi} - K^l_D\tilde{L}_l - K^l_P I_l \tag{14.14b}$$

$$\dot{\tilde{\xi}} = A_{t_l}u_1 + \Lambda_{s_l}u_2 + \alpha A_{c_l}u_3 + \Lambda_{\mathcal{B}_l}\mathrm{v}_{\mathcal{B}} - \ddot{L}^d_l + K^l_D\dot{\tilde{L}}_l + K^l_P\tilde{L}_l. \tag{14.14c}$$

It can be proven that there are conditions for the existence of a smooth control input $u = [u_1^\top,\ u_2^\top,\ u_3^\top]^\top$ such that the closed loop equilibrium point $(I_l, \tilde{L}_l, \tilde{\xi}) =$

$(0,0,0)$ of system (14.14) is globally asymptotically stable [Pucci et al., 2018]. In particular, recall that the centroidal momentum is linear versus the system velocity ν, i.e. ${}^{\mathcal{G}[\mathcal{I}]}L = J_G(q)\nu = J_G^{\mathcal{B}}\mathrm{v}_{\mathcal{B}} + J_G^s\dot{s}$. The matrix $J_G = [J_G^{\mathcal{B}},\ J_G^s]$ is the *centroidal momentum matrix* [Orin et al., 2013]. Then, define:

$$\begin{aligned}
&h_l := (\Lambda_{\mathcal{B}_l} + \tilde{K}J_G^{\mathcal{B}l})\mathrm{v}_{\mathcal{B}} + (K_D^l + 1_3)\dot{\tilde{L}}_l + K_P^l I_l - \ddot{L}_l^d - \tilde{K}L_l^d \\
&H_l := \left[A_{t_l},\quad \Lambda_{s_l} + \tilde{K}J_G^{sl},\quad \alpha A_{c_l}\right] \\
&\tilde{K} := K_P^l + K_D^l + K_O^{-1}
\end{aligned}$$

with $K_O \in \mathbb{R}^{3\times 3}$ a symmetric and positive definite matrix and the partition $J_G^l := [J_G^{\mathcal{B}l}, J_G^{sl}] \in \mathbb{R}^{3\times n+6}$ corresponds to the first three rows of J_G. If there exist a control input u such that:

$$H_l u + h_l = 0, \tag{14.15}$$

then the closed loop equilibrium point $(I_l, \tilde{L}_l, \tilde{\xi}) = (0,0,0)$ is globally asymptotically stable. More specifically, as long as $\mathrm{rank}(H_l) = 6$, a solution to Eq. (14.15) always exists [Pucci et al., 2018].

14.2.3 Quadratic programming formulation

The linear and angular momentum tasks (14.13)–(14.15) can be framed as a QP optimization problem. The advantage of QP formulation is twofold: first, additional tasks of lower priority can be added to the cost function to resolve eventual actuation redundancy. Secondly, it is possible to account for software and hardware constraints in the control input definition. The QP problem is formulated as:

$$\begin{aligned}
u^* = \ &\underset{u}{\mathrm{argmin}}\,(\lambda_\omega |H_\omega u + h_\omega - {}^{\mathcal{G}[\mathcal{B}]}\ddot{L}_\omega^*|^2 + \lambda_l |H_l u + h_l|^2 + \\
&\qquad \lambda_s |u_2 - \dot{s}^*|^2) && (14.16a)\\
&s.t. \\
&\qquad \alpha J\nu = 0, && (14.16b)\\
&\qquad {}^{\mathcal{G}[\mathcal{B}]}L_\omega = {}^{\mathcal{G}[\mathcal{B}]}\mathbb{I}_0{}^{\mathcal{B}}\omega_{\mathcal{B}}, && (14.16c)\\
&\qquad l_b^1(\mathrm{w}_t) < u_1 < u_b^1(\mathrm{w}_t), && (14.16d)\\
&\qquad l_b^2(s) < u_2 < u_b^2(s), && (14.16e)\\
&\qquad l_b^3(t) < u_3 < u_b^3(t). && (14.16f)
\end{aligned}$$

The parameters $\lambda_\omega, \lambda_l, \lambda_s \in \mathbb{R}^+$ are positive weighting constants. The low priority task $u_2 = \dot{s}^*$ exploits the actuation redundancy to keep the robot joints close to a desired robot posture:

$$\dot{s}^* := -K_P^s(s - s^d)$$

where $K_P^s \in \mathbb{R}^{n \times n}$ is a symmetric and positive definite matrix and $s^d \in \mathbb{R}^n$ a reference position for the joints configuration. The equality constraint (14.16b) accounts for the *feet fixed* assumption while the robot is balancing. The angular momentum constraint (14.16c) allows to perform orientation control while flying.

Formulation of the input boundaries

The upper and lower bounds of the control variables are included in the QP formulation with (14.16d)–(14.16e)–(14.16f). These boundaries have been expressed following a novel formulation, whose advantage is twofold:

- it allows to take into account also the boundaries of the integrals of u_1 and u_2, namely the thrust magnitude w_t and the joints position s;
- it allows semi-automatic take off and landing by proper regulation of the feet vertical forces rate of change.

The expressions of the boundaries of u_1 and u_2 are given by:

$$\begin{aligned} l_b^1 &= \tanh(\epsilon_1(\mathrm{w}_t - \mathrm{w}_t^{min}))\dot{\mathrm{w}}_t^{min} \\ u_b^1 &= \tanh(\epsilon_1(\mathrm{w}_t^{max} - \mathrm{w}_t))\dot{\mathrm{w}}_t^{max} \\ l_b^2 &= \tanh(\epsilon_2(s - s^{min}))\dot{s}^{min} \\ u_b^2 &= \tanh(\epsilon_2(s^{max} - s))\dot{s}^{max}, \end{aligned}$$

with $\epsilon_{1,2}$ positive constants. This formulation of the input boundaries allows to also take into account in the QP the saturation of the thrust magnitude and of the joint angles. The proof is in Appendix F. This is a considerable advantage as including joints and thrusts limits is fundamental to enforce the reliability of the control algorithm on real applications.

The boundaries on the contact wrench rate of change u_3 are exploited for performing semi-automatic take off and landing. The procedure is simple: select the upper and lower bounds corresponding to the feet vertical forces, i.e. $l_b^{3z}, u_b^{3z} \in \mathbb{R}^2$. During take off, a trigger (e.g. a button pressed by the user or the entrance in a new state of a state machine) renders the upper bound $u_b^{3z} < 0$. The modified boundary constrains the QP to find balancing solutions that implies the vertical forces rate of change to be negative, thus reducing (up to zero) the feet forces and consequently increasing the thrust magnitude to compensate gravity. Similarly, during the landing phase a trigger renders the lower bound $l_b^{3z} > 0$. Vertical forces then increase up to half of the gravity force, while the thrust magnitude is reduced to zero.

Remark: algebraic loop during the balancing phase

A long but straightforward analysis on all the terms composing the cost function (14.16a) shows that they depend non-linearly on the base linear and angular velocity $\mathrm{v}_{\mathcal{B}}$ and on the contact wrenches f. During the balancing phase both these quantities are a function of the joints velocities $\dot{s}$ because of the *feet fixed* assumption $J\nu = 0$, thus there is an algebraic loop. Note that the loop is instead absent during the flying phase.

Let us observe however that both in simulation and on the real iCub the contact forces and the base velocity are directly measured by FT sensors, by an on board IMU and external cameras. Therefore, we may try to ignore the algebraic loop during the balancing phase and directly use these measurements in the control algorithm. Closed-loop system stability is then verified numerically in simulation. On the real robot, unavoidable delays in sensing technology and network communication may be sufficient to resolve the algebraic loop during the balancing phase.

14.3 Joint Torque Control

The solution to the optimization problem (14.16) is given by the triple $[u_1^*,\ u_2^*,\ u_3^*]$. The thrusts intensities rate of change $\dot{\mathrm{w}}_t^* = u_1^*$ can be directly commanded to the jet turbines, whose dynamics –for the purpose of this thesis – is neglected. To achieve the requested joint velocities $\dot{s}^* = u_2^*$ and contact forces rate of change $\dot{f}^* = u_3^*$, we resort to the robot joint torques. More specifically, let us design another minimization procedure:

$$
\begin{aligned}
(\tau^*, f^*) = \ & \underset{(\tau,f)}{\operatorname{argmin}} \, (\lambda_\tau |\ddot{s} - \ddot{s}^*|^2 + \alpha\lambda_f |f - I_{u_3^*}|^2) && \text{(14.17a)}\\
& s.t. \\
& \qquad \alpha(J\dot{\nu} + \dot{J}\nu) = 0, && \text{(14.17b)}\\
& \qquad \alpha C_f f \leq \alpha b_f, && \text{(14.17c)}\\
& \qquad l_b^\tau < \tau < u_b^\tau. && \text{(14.17d)}
\end{aligned}
$$

The cost function (14.17a) is composed of two tasks: one achieves desired joints accelerations and the other desired contact wrenches. The joints acceleration are obtained by partitioning the system dynamics Eq. (14.1):

$$\ddot{s} = B^\top M^{-1}(J^\top f + J_t^\top \mathrm{w}_t - h + B\tau).$$

Then, reference joints acceleration $\ddot{s}^*$ are chosen to perform high gain control and stabilize $\dot{s} = u_2^*$:

$$\ddot{s}^* = -K_P^s(\dot{s} - u_2^*) - K_I^s \int_0^t (\dot{s} - u_2^*)dt.$$

The force task attempts at selecting contact wrenches such that $f^* = I_{u_3^*}$, where we define $I_{u_3^*} = \int_0^t u_3^* dt$. The parameters λ_τ and λ_f are positive weighting constants.

The equality constraints (14.17b) highlight the dependency between contact wrenches and joint torques, i.e. $f = f(\tau)$. The remaining inequalities (14.17c)–(14.17d) account for contact stability constraints and joint torques saturation.

Let us observe that the dimension of the control variables during balancing $[u_2^*, u_3^*] \in \mathbb{R}^{n+12}$ is greater than the number of torques $\tau \in \mathbb{R}^n$. Thus, the optimization problem (14.17) does not have an exact solution, and therefore it finds the closest solution in the least square sense that also respects the constraints.

14.4 Simulation Results

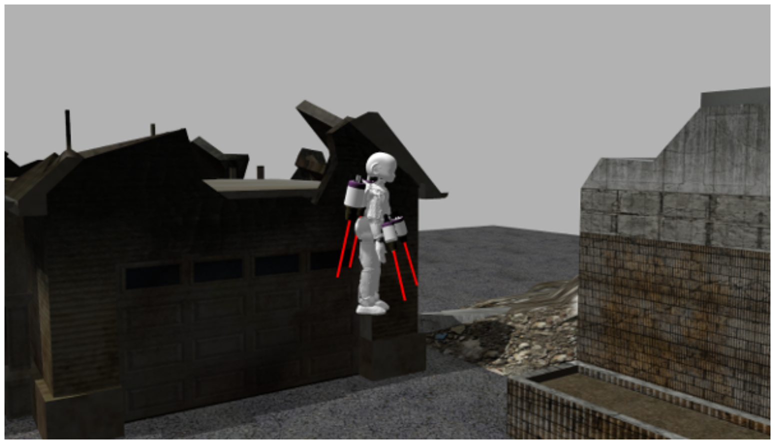


Fig. 14.2 iRonCub flying in a disaster-like scenario.

14.4.1 Simulation environment

The controller is implemented in Simulink and tested in Gazebo simulator. The controller frequency is 100 Hz. All changes in linear and angular momentum references, as well as take off and landing, are triggered by means of a joypad that communicates with Simulink through YARP network. The simulation environment is depicted in Fig. 14.2 and it is composed of several destroyed buildings endowed with physical properties so that the robot can interact with them.

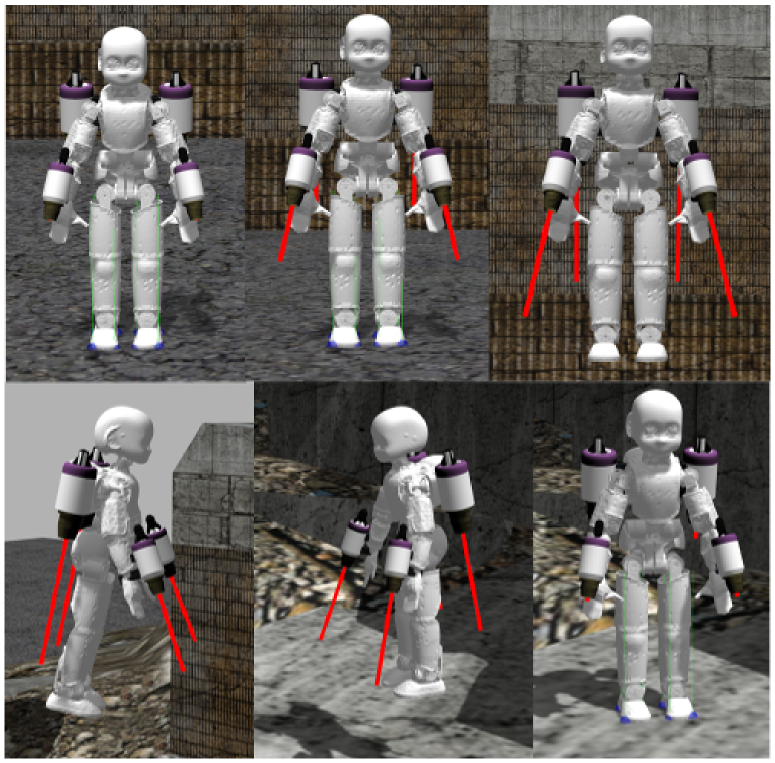


Fig. 14.3 A typical flight sequence. From left to right, top to bottom: balancing; taking off; hovering; flying; approaching; landed.

14.4.2 Flying simulation with joypad

We perform multiple take off, landing, balancing and flying tests in simulation with joypad control. The joypad control allows to perform a wide number of maneuvers during flight, without the necessity to design a proper planner and a complex state machine to manage take off and landing. Fig. 14.3 shows a typical sequence of movements of the robot while flying and landing on the roof of a building.

Range of motion of the flying robot

Figures 14.4–14.5 show the center of mass position and the base link orientation while flying, compared to their reference values. Observe that reference variations are not provided one at a time, but the robot can move along several directions at the same time. The areas depicted in red correspond to the *balancing* phases, including take off and landing.

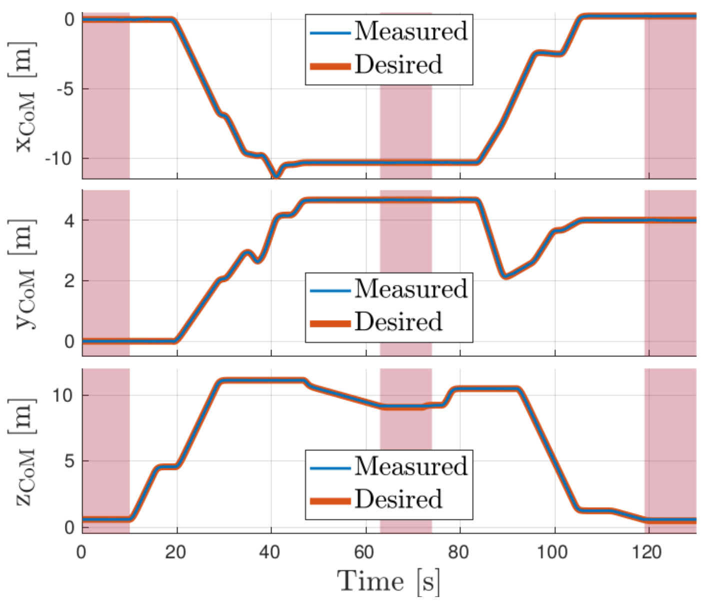


Fig. 14.4 Center of mass measured and desired position during flight simulation.

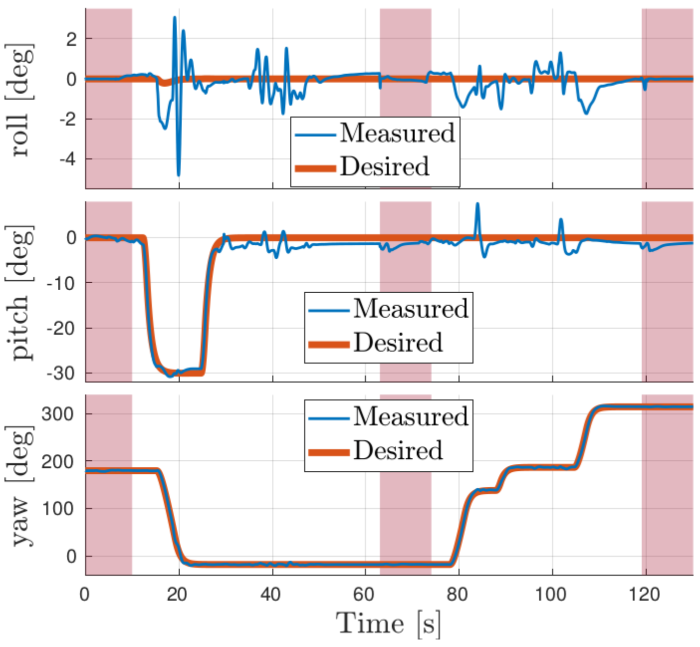


Fig. 14.5 Measured and desired base orientation during flight simulation.

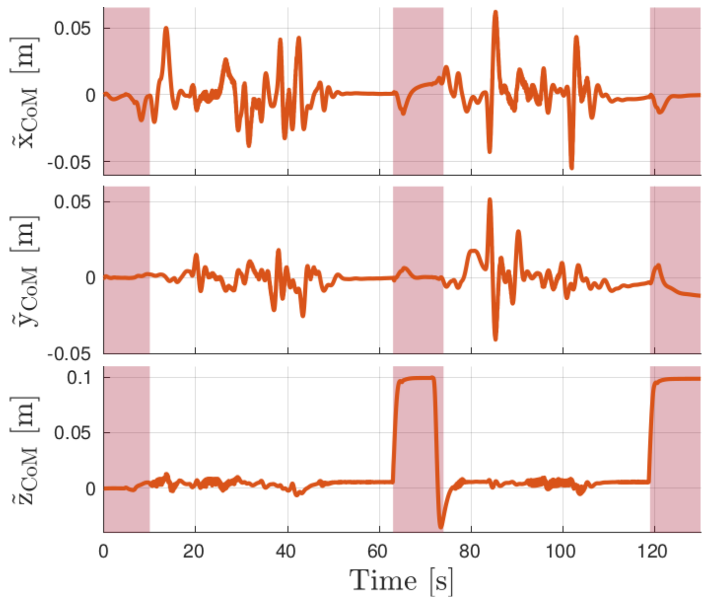


Fig. 14.6 Center of mass position error during flight simulation.

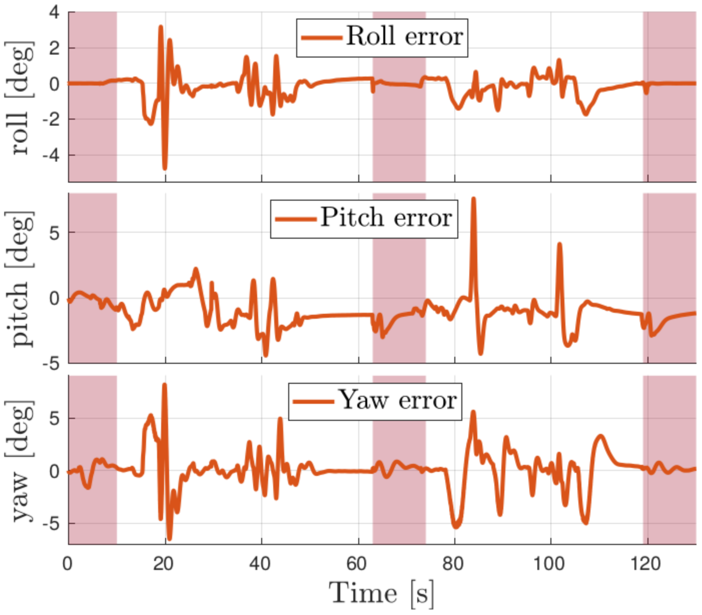


Fig. 14.7 Base orientation error during flight simulation.

From the pictures it is possible to understand the range of motion of the flying robot: during our tests, the center of mass position varies considerably in all directions, with a variation around $5\ \mathrm{m}$ along the y direction and a variation around $10\ \mathrm{m}$ along x and z directions. The base orientation reference instead has been limited by purpose in the pitch and roll angles. In absence of a proper planner, the bounds of pitch and roll are necessary to avoid that the user provides unfesible references. In fact, perfect decoupling of robot position and orientation is not always possible due to actuation limits and joints boudaries. The orientation limit references have been estimated experimentally, and are $\pm 2.5°$ for the roll and $[-30°, 0°]$ for the pitch angle. The base yaw instead can vary with no limits. Despite the reduced base orientation motion, our simulations demonstrate that the robot can still perform complex maneuvers during flight.

Trajectory tracking errors

Position and orientation tracking errors during the flight simulation are depicted in Figures 14.6–14.7. Both the base orientation and the center of mass position errors remain sufficienty small to allow the robot to perform all tasks smoothly. In particular, the center of mass error remains smaller than $5\ \mathrm{cm}$, excluding the error along the z direction ($10\ \mathrm{cm}$) during balancing after landing. The high error during this phase is due to the way the controller handles the landing maneuver: in order to activate landing, the robot needs to push against the contact surface and overcome a force threshold. A non-zero initial feet wrench when landing facilitates the transition from flying to balancing. The pushing is obtained by providing a reference for the CoM height which is lower (by some centimeters) than the minimum CoM height the robot can achieve when balancing with straight legs. Therefore the robot in landing configuration cannot properly track the reference CoM along the z direction, and this justifies the high error along z direction during balancing.

The orientation error always remains smaller than about $6°$. Note that gain tuning of the robot orientation is particularly challenging, as the control couples the errors of the three angles. Despite this limitation, the orientation tracking performance is sufficient to allow the robot to fly and perform maneuvers while flying smoothly. Figure 14.8 shows the norm of the joint position error, which correspond to the task of lower priority. The joint position error norm remains bounded during the whole flight, showing that the QP formulation (14.16) also allows to stabilize the zero dynamics of the system.

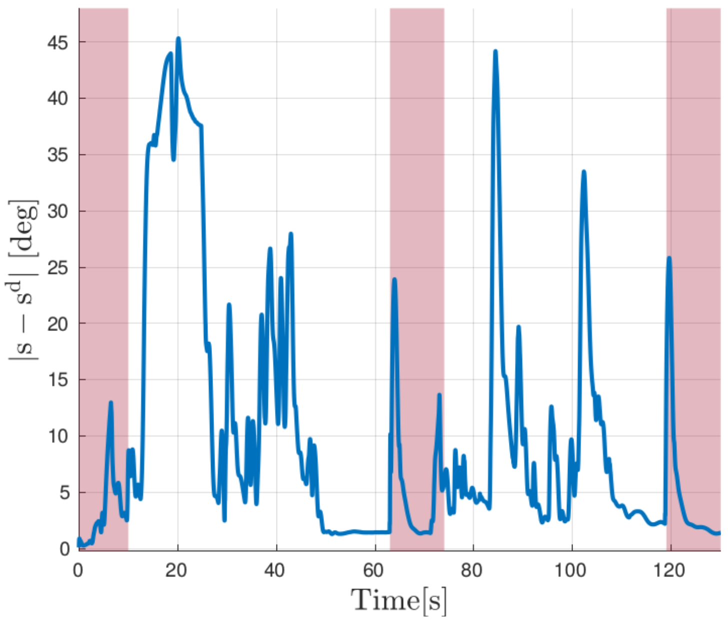


Fig. 14.8 Joint position error norm during flight simulation.

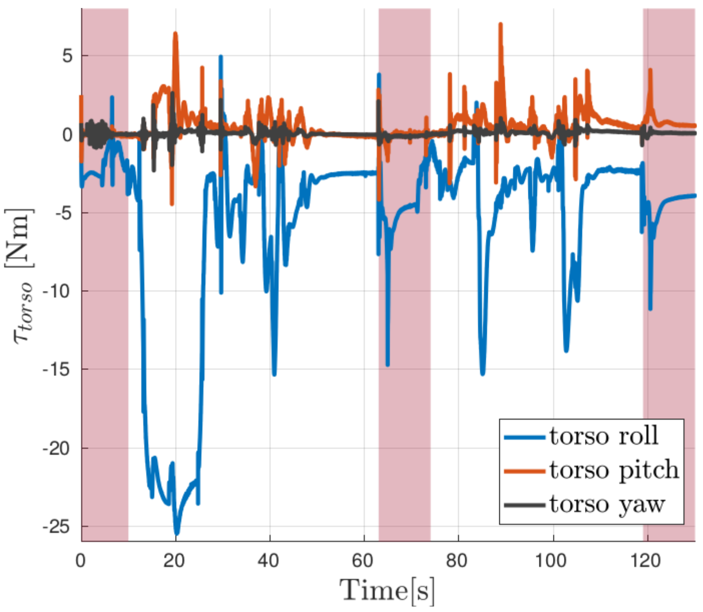


Fig. 14.9 Desired joint torques at robot torso during flight simulation.

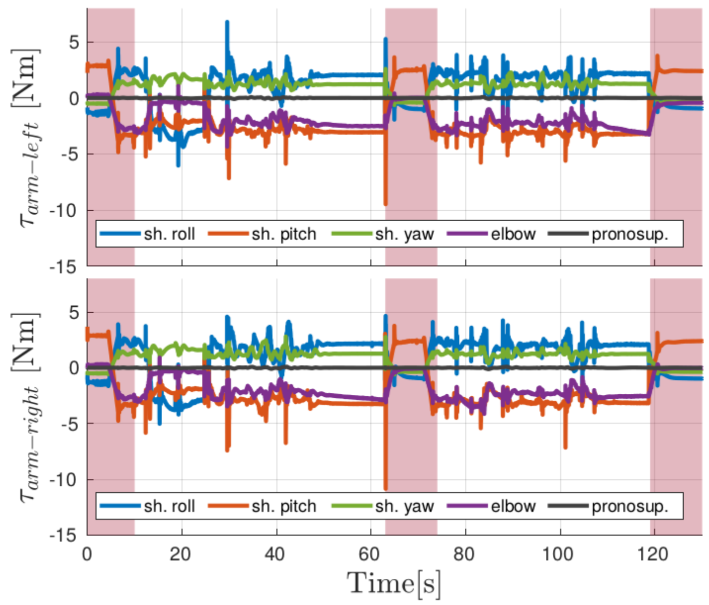


Fig. 14.10 Desired joint torques at robot arms during flight simulation.

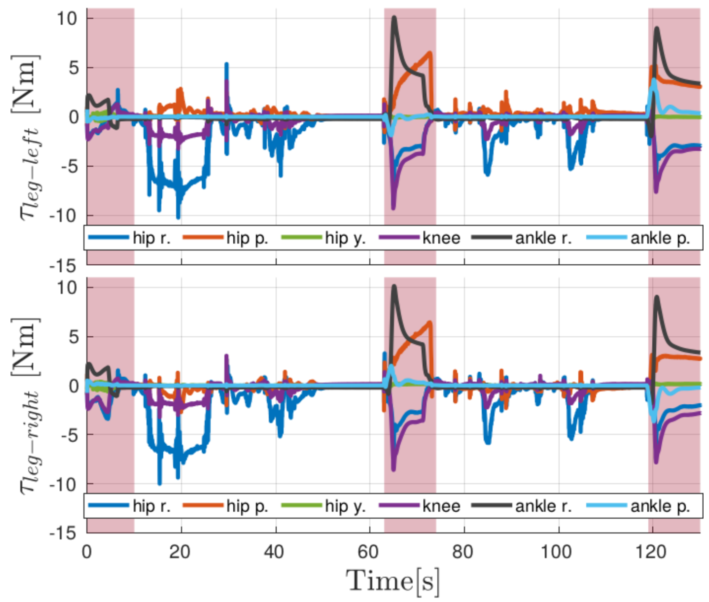


Fig. 14.11 Desired joint torques at robot legs during flight simulation.

Joint torques and thrust forces

The joint torques required by the controller during the entire simulation are depicted in Figures 14.9 (torso), 14.10 (arms) and 14.11 (legs). On the iCub robot, torque saturation is about $34\mathrm{Nm}$. It can be easily verified that the controller provides relatively smooth torques that always remain far from saturation.

Figure 14.12 represents instead the measured jets thrust forces during flight. The simulated jet turbines are the JetCat P220 on the jetpack and JetCat P100 at the hands. The maximum thrust provided by the P220 and the P100 is 220 N and 100 N, respectively. The analysis of the measured thrust pointed out that the hands turbines are often operating around 90% of their power, while the jetpack turbines seldom overcome 50%. A possible explanation may be that the jetpack thrusts, which are located behind the robot center of mass, generate a moment that tends to rotate the robot along its pitch. The high thrust magnitude provided by the hands turbines is then required to compensate for this rotation. In fact, one may observe that the hands thrust is reduced considerably when the base pitch angle is rotated (seconds $15 - 30$ in the figures).

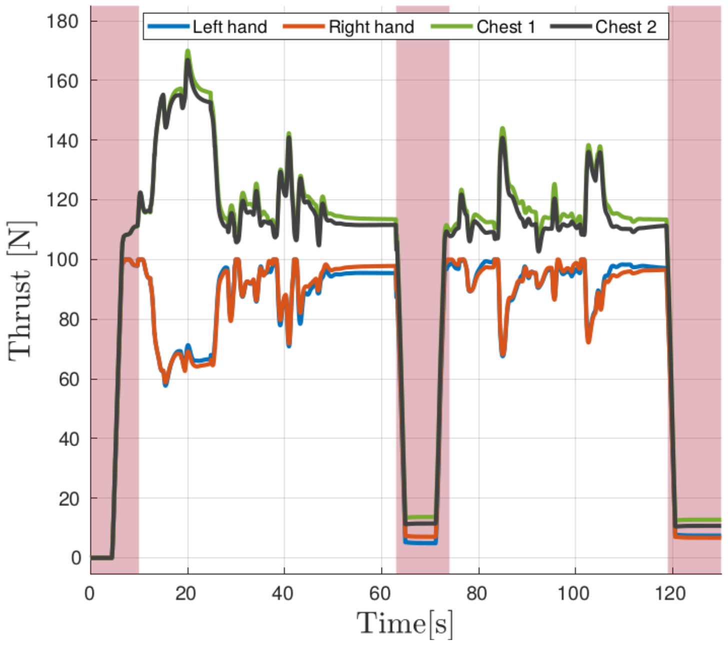


Fig. 14.12 Measured jets thrust magnitude during flight simulation.

Epilogue

This three years Ph.D project focused on improving the state-of-the art of instantaneous momentum-based and optimization-based controllers for floating base robots. The final achievement is the design of controllers with proven stability and robustness properties, that can be implemented on different robotic platforms and employed for solving a large variety of tasks. The proposed control frameworks have been tested both in simulation and on real robots, namely, the humanoid robot iCub, the aerial manipulator OTHex and the robot iRonCub.

Yet, some of the problems stated in Chapter 5 still remain open. There are also issues that have been partially addressed, but effective implementation on the real robot is still challenging. In what follows, every treated topic in the thesis is recalled together with a discussion on the achieved results and the open issues that still need to be addressed in future works.

Part II

In **Chapter 7** we presented numerical evidence that a classical stack-of-task approach to momentum controllers may lead to an instability of the zero dynamics. To ensure stability, we demonstrated that it is necessary to close the loop with *orientation* terms at the momentum level. We designed a modification of state-of-the-art momentum based control strategies that ensure asymptotic stability, which was shown by performing Lyapunov analysis on the linearization of the closed loop system around the equilibrium point. Simulation and experimental tests validate the presented analysis.

A critical point needed to prove the stability of the constrained dynamical system is to define the minimum set of coordinates identifying its evolution. This can be straightforward in case of balancing on one foot, but the extension to multiple contacts is not trivial and must be considered carefully.

Chapter 8 proposed a gain tuning procedure for momentum-based controllers. The objective is the achievement of a desired local dynamics for the robot joints. Constraints on symmetry and positive definiteness of the tuned gain matrices are enforced thanks to a tracker for symmetric positive definite matrices. Simulation results show the effectiveness of the gain tuning procedure for both the robot balancing on one foot and two feet.

Further improvements on the gain tuning procedure may be developed in future works. The gains optimization presented in this Chapter can be applied to different equilibrium points along a joint reference trajectory, and may be an efficient tuning strategy in case of humanoid walking. Online implementations of the presented algorithm on the real humanoid robot is still being investigated.

Robot balancing in highly dynamic environments is the topic addressed in **Chapter 9**. In particular, the case study is iCub balancing on a seesaw board. The seesaw equations of motion and the contact constraints equations are obtained following a general procedure, that can be reused in case the robot is in contact with a different object. The momentum-based control algorithm is redesigned taking into account the seesaw dynamics, and two different controllers are proposed.

Only simulation results are presented. However, preliminary tests on the real robot iCub have been performed, depicting some limitations of our control approach. For example, the presence of modeling errors and network delays can strongly affect the controllers' performances. Future work might be focused on reducing the modeling and estimation errors on the real platform.

In **Chapter 10** we proposed a framework for extending a momentum based controllers developed for humanoid robots with stiff actuators to the case of series elastic actuators. The key point is to consider motor velocities as a fictitious control input of the robot's dynamics. Then, fast convergence of the motor velocities to the desired values is obtained through high gain control of motors dynamics. Compared to other strategies, our control framework is robust against the feedforward terms usually needed by pure feedback linearization techniques, and allows us to easily extend the momentum based controllers developed for rigid joints to the elastic joint case.

In this Chapter, only simulation results are presented because series elastic actuators developed in [Parmiggiani et al., 2012, Tisi et al., 2016] are currently not installed on the real iCub. Experimental validation may be performed on other humanoid robots endowed with Series Elastic Actuators.

Classical algorithms for whole-body torque control of humanoid robots compensate for the effect of the joints viscous and Coulomb friction. **Chapter 11** proposed instead a torque control framework that allows to exploit the inherent stabilizing nature of the joints friction. We first developed a control algorithm for fixed base robots that exploits friction for improving the robot joints tracking, and that can also ensure better robustness of the controlled system w.r.t. noisy velocity measurements. Then, we extended our formulation to the control of a floating base robot. In particular, the joints friction was exploited to improve the tracking of a desired momentum trajectory. Experimental results on iCub show the effectiveness of the proposed approach.

Future work may focus on analyzing if other dynamical effects normally neglected or compensated can be included in the control algorithm to improve its performances. Furthermore, a stronger theoretical justification of the control framework when applied to floating base systems may be required.

Chapter 12 addressed some common limitations of force and torque controllers for floating base systems based on Quadratic Programming. More specifically, we removed inequality constraints from the optimization problem by designing an invertible, one-to-one mapping that parametrises the contact wrenches into a new set of unconstrained variables. This parametrization guarantees that the output wrenches always satisfy the contact stability constraints. Based on this mapping we designed a general jerk control optimization framework for floating base systems. We then analyzed a specific use case of the jerk controller, namely a momentum-based jerk control architecture for balancing a 23 DoFs humanoid robot. The controller has been validated both in simulation and on the real iCub, and compared with a classical momentum-based controller. Sensitivity to errors in the momentum rate of change estimation is identified as a drawback of the approach, as it may affect stability and convergence of the closed loop dynamics. A solution for increasing robustness of the controller w.r.t. biases on momentum estimation is proposed.

A limitation is that the proposed jerk control architecture does not take into account joints position and torque limits. A future work may involve the integration of these limits in the control framework. Further development will be done in order to extend the jerk controller to the humanoid walking task.

Part III

In **Chapter 13** we considered the challenging problem of precise position and force control of an aerial manipulator. The proposed method takes inspiration from whole-body control methods applied to humanoid robots. In particular, our control method is based on a multi-task optimization problem, solved via quadratic programming. This allows to consider different control objectives (e.g., interaction force, position of the end-effector, full pose of the robot, etc.) and hard constraints that should be respected to ensure the stability of the system and the feasibility of the problem solution. The precision of the method during interaction is enhanced by a direct force-feedback exploiting a FT sensor integrated at the manipulator end-effector. We performed experimental and simulation tests based on fully- and under-actuated aerial platforms respectively. The corresponding results show the great flexibility of the method and the improvement in the force-tracking thanks to the explicit force-feedback.

In the experiments, only the point contact scenario has been considered. In future work we shall also consider other kind of interactions, and consequently design control algorithms that will include full contact wrench feedback. Furthermore, other challenging tasks will be subject of study, such as manipulation and grasping, and multi-contact interaction. Further experiments will also be carried on with underactuated platforms.

Chapter 14 proposes a momentum jerk control for the stabilization of a jet-powered flying humanoid robot. The control algorithm can guarantee global tracking of a desired CoM position and local asymptotic stabilization of a reference attitude trajectory while flying. Also, the controller can handle take off and landing maneuvers. Numerical validation shows that stability is retained also when the robot is balancing on rigid contacts. Simulation results are obtained in Gazebo with a joypad-controlled iRonCub.

Nevertheless, unmodeled phenomena such as high temperature effects, high frequency vibrations, ground effect, and the jets' own dynamics may impair the effectiveness of real robot control. Future work will investigate these issues in order to lead to the implementation of the control algorithm on the real iRonCub.

Bibliography

J. A. Acosta and M. Lopez-Martinez. Constructive feedback linearization of underactuated mechanical systems with 2-DOF. *44th IEEE Conference on Decision and Control, 2005 European Control Conference CDC-ECC.*, 2005.

A. Albu-Schaffer, C. Ott, and G. Hirzinger. A passivity based cartesian impedance controller for flexible joint robots - part ii: full state feedback, impedance design and experiments. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 3, pages 2666–2672 Vol.3, April 2004. doi: 10.1109/ROBOT.2004.1307463.

Alin Albu-Schäffer, Christian Ott, and Gerd Hirzinger. A unified passivity-based control framework for position, torque and impedance control of flexible joint robots. *The International Journal of Robotics Research*, 26(1):23–39, 2007. doi: 10.1177/0278364907073776. URL `http://dx.doi.org/10.1177/0278364907073776`.

S. O. Anderson and J. K. Hodgins. Adaptive torque-based control of a humanoid robot on an unstable platform. In *2010 10th IEEE-RAS International Conference on Humanoid Robots*, pages 511–517, Dec 2010. doi: 10.1109/ICHR.2010.5686343.

Joel A E Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, In Press, 2018.

G. Antonelli, E. Cataldi, G. Muscio, M. Trujillo, Y. Rodriguez, F. Pierri, F. Caccavale, A. Viguria, S. Chiaverini, and A. Ollero. Impedance control of an aerial-manipulator: Preliminary results. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 3848–3853, Daejeon, South Korea, 2016.

Narong Aphiratsakun and Manukid Parnichkun. Fuzzy based Gains Tuning of PD controller for joint position control of AIT Leg Exoskeleton-I (ALEX-I). *Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on*, Feb 2009.

K. Baizid, G. Giglio, F. Pierri, M. A. Trujillo, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, and A. Ollero. Behavioral control of unmanned aerial vehicle manipulator systems. *Autonomous Robots*, 41(5):1203–1220, 2017.

Ahmad Bani Younes, James Turner, D. Mortari, and John Junkins. A survey of attitude error representations. In *AIAA/AAS Astrodynamics Specialist Conference 2012*, 08 2012. ISBN 978-1-62410-182-3. doi: 10.2514/6.2012-4422.

R. Blickhan. The spring-mass model for running and hopping. *Journal of Biomechanics*, 22(11):1217 – 1227, 1989. ISSN 0021-9290. doi: https://doi.org/10.1016/0021-9290(89)90224-8. URL `http://www.sciencedirect.com/science/article/pii/0021929089902248`.

Caltech. Caltech building agile humanoid robot by combining legs with thrusters, 2019. URL `https://bit.ly/34TC2f7`.

Justin Carpentier and Nicolas Mansard. Analytical derivatives of rigid body dynamics algorithms. In *Robotics: Science and Systems*, 2018.

S. Caux, E. Mateo, and R. Zapata. Balance of biped robots: special double-inverted pendulum. *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*, 1998.

M. Charbonneau, F. Nori, and D. Pucci. On-line joint limit avoidance for torque controlled robots by joint space parametrization. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 899–904, Nov 2016. doi: 10.1109/HUMANOIDS.2016.7803379.

Marie Charbonneau, Valerio Modugno, Francesco Nori, Giuseppe Oriolo, Daniele Pucci, and Serena Ivaldi. Learning robust task priorities of QP-based whole-body torque-controllers. In *2018 IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 11 2018. doi: 10.1109/HUMANOIDS.2018.8624995.

Marco La Civita, G. Papageorgiou, William Messner, and Takeo Kanade. Design and flight testing of a gain-scheduled h-infinity loop shaping controller for wide-envelope flight of a robotic helicopter. In *Proceedings of the 2003 American Control Conference*, pages 4195 – 4200, June 2003.

S. Dafarra, G. Nava, M. Charbonneau, N. Guedelha, F. Andradel, S. Traversaro, L. Fiorio, F. Romano, F. Nori, G. Metta, and D. Pucci. A control architecture with online predictive planning for position and torque controlled walking of humanoid robots. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9, Oct 2018.

L. D'Antonio and Salvatore Monaco. A nonlinear controller for parabolic flight. *Proceedings on the 22th Conference on Decision and Control*, 01 1993. doi: 10.1109/CDC.1993.325441.

A. de Luca and P. Lucibello. A general algorithm for dynamic feedback linearization of robots with elastic joints. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, volume 1, pages 504–510 vol.1, May 1998. doi: 10.1109/ROBOT.1998.677024.

Carlos Canudas de Wit, Bruno Siciliano, and Georges Bastin. *Theory of Robot Control: Ch.: Motion and force control*, pages 141–175. Springer London, London, 1996.

Andrea Del Prete, Nicolas Mansard, Oscar E. Ramos, Olivier Stasse, and Francesco Nori. Implementing torque control with high-ratio gear boxes and without joint-torque sensors. *International Journal of Humanoid Robotics*, 13(01):1550044, 2016. doi: 10.1142/S0219843615500449. URL `https://doi.org/10.1142/S0219843615500449`.

D. Erickson, M. Weber, and I. Sharf. Contact stiffness and damping estimation for robotic systems. *Int. J. Robotics Research*, 22, Jan 2003.

Hanno Essén. Average angular velocity. *European journal of physics, IOP Publishing*, 14(5):201, 1993.

E. Farnioli, M. Gabiccini, and A. Bicchi. Optimal contact force distribution for compliant humanoid robots in whole-body loco-manipulation tasks. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5675–5681, May 2015. doi: 10.1109/ICRA.2015.7139994.

Roy Featherstone. *Rigid Body Dynamics Algorithms*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007. ISBN 0387743146.

Hans Joachim Ferreau, Christian Kirches, Andreas Potschka, Hans Georg Bock, and Moritz Diehl. qpoases: a parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 6(4):327–363, Dec 2014. ISSN 1867-2957. doi: 10.1007/s12532-014-0071-1. URL `https://doi.org/10.1007/s12532-014-0071-1`.

Paul Fitzpatrick, Giorgio Metta, and Lorenzo Natale. Towards long-lived robot genes. *Robotics and Autonomous Systems*, 56(1):29 – 45, 2008. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2007.09.014. URL `http://www.sciencedirect.com/science/article/pii/S0921889007001364`. Human Technologies: “Know-how”.

Thomas Flayols, Andrea Del Prete, Majid Khadiv, Nicolas Mansard, and Ludovic Righetti. Balancing Legged Robots on Visco-Elastic Contacts. working paper or preprint, October 2019. URL `https://hal.archives-ouvertes.fr/hal-02310082`.

Sara Fleury, Matthieu Herrb, and Raja Chatila. Genom: A tool for the specification and the implementation of operating modules in a distributed robot architecture. In *International Conference on Intelligent Robots and Systems*, pages 842–848, 1997.

A. Franchi, R. Carli, D. Bicego, and M. Ryll. Full-pose tracking control for aerial robotic systems with laterally bounded input force. *IEEE Transactions on Robotics*, 34(2):534–541, April 2018. ISSN 1941-0468. doi: 10.1109/TRO.2017.2786734.

E. Frazzoli, M. A. Dahleh, and E. Feron. Trajectory tracking control design for autonomous helicopters using a backstepping algorithm. In *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No.00CH36334)*, volume 6, pages 4102–4107 vol.6, June 2000. doi: 10.1109/ACC.2000.876993.

R.J. Full and D.E. Koditschek. Templates and anchors: neuromechanical hypotheses of legged locomotion on land. *Journal of Experimental Biology*, 202(23): 3325–3332, 1999. ISSN 0022-0949. URL `https://jeb.biologists.org/content/202/23/3325`.

Carlos E. García, David M. Prett, and Manfred Morari. Model predictive control: Theory and practice—a survey. *Automatica*, 25(3):335 – 348, 1989. ISSN 0005-1098. doi: https://doi.org/10.1016/0005-1098(89)90002-2. URL `http://www.sciencedirect.com/science/article/pii/0005109889900022`.

Jessy W. Grizzle, Christine Chevallereau, Ryan W. Sinnet, and Aaron D. Ames. Models, feedback control, and open problems of 3d bipedal robotic walking. *Automatica*, 50(8):1955 – 1988, 2014. ISSN 0005-1098. doi: https://doi.org/10.1016/j.automatica.2014.04.021. URL `http://www.sciencedirect.com/science/article/pii/S0005109814001654`.

Sebastien Gros, Marion Zanon, and Moritz Diehl. Baumgarte stabilisation over the SO(3) rotation group for control. *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 620–625, December 2015.

John Hauser, Shankar Sastry, and George Meyer. Nonlinear control design for slightly non-minimum phase systems: Application to v/stol aircraft. *Automatica*, 28(4):665 – 679, 1992. ISSN 0005-1098. doi: https://doi.org/10.1016/0005-1098(92)90029-F. URL `http://www.sciencedirect.com/science/article/pii/000510989290029F`.

A. Herzog, L. Righetti, F. Grimminger, P. Pastor, and S. Schaal. Balancing experiments on a torque-controlled humanoid with hierarchical inverse dynamics. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 981–988, Sept 2014. doi: 10.1109/IROS.2014.6942678.

K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka. The development of Honda humanoid robot. *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, 1998.

M.A. Hopkins, R.J. Griffin, A. Leonessa, B.Y. Lattimer, and T. Furukawa. Design of a compliant bipedal walking controller for the darpa robotics challenge. In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pages 831–837, Nov 2015. doi: 10.1109/HUMANOIDS.2015.7363450.

M. D. Hua, T. Hamel, P. Morin, and C. Samson. Introduction to feedback control of underactuated vtol vehicles: A review of basic control design ideas and principles. *IEEE Control Systems*, 33(1):61–75, Feb 2013. ISSN 1066-033X. doi: 10.1109/MCS.2012.2225931.

Qiang Huang, K. Kaneko, K. Yokoi, S. Kajita, T. Kotoku, N. Koyachi, H. Arai, N. Imamura, K. Komoriya, and K. Tanie. Balance control of a biped robot combining off-line pattern with real-time modification. *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, 2000.

Z. Huang, B. Liu, J. Wei, Q. Lin, J. Ota, and Y. Zhang. Jet-hr1: Two-dimensional bipedal robot step over large obstacle based on a ducted-fan propulsion system. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pages 406–411, Nov 2017. doi: 10.1109/HUMANOIDS.2017.8246905.

Hun-ok Lim, Y. Kaneshima, and A. Takanishi. Online walking pattern generation for biped humanoid robot with trunk. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, volume 3, pages 3111–3116 vol.3, May 2002. doi: 10.1109/ROBOT.2002.1013705.

Yildirim Hurmuzlu, Frank Génot, and Bernard Brogliato. Modeling, stability and control of biped robots—a general framework. *Automatica*, 40(10):1647 – 1664, 2004. ISSN 0005-1098. doi: https://doi.org/10.1016/j.automatica.2004.01.031. URL `http://www.sciencedirect.com/science/article/pii/S0005109804000998`.

S. Hyon, J.G. Hale, and G. Cheng. Full-body compliant human-humanoid interaction: Balancing in the presence of unknown external forces. *Robotics, IEEE Transactions on*, Oct 2007. ISSN 1552-3098. doi: 10.1109/TRO.2007.904896.

F. Iida, G. Gomez, and R. Pfeifer. Exploiting body dynamics for controlling a running quadruped robot. In *ICAR '05. Proceedings., 12th International Conference on Advanced Robotics, 2005.*, pages 229–235, July 2005. doi: 10.1109/ICAR.2005.1507417.

A. Isidori. *Nonlinear Control Systems*. Third. Springer Verlag, 1995.

Alberto Isidori. The zero dynamics of a nonlinear system: From the origin to the latest progresses of a long successful story. *European Journal of Control*, 19(5): 369 – 378, 2013. ISSN 0947-3580. doi: https://doi.org/10.1016/j.ejcon.2013.05.014.

Julius Jellinek and DH Li. Separation of the energy of overall rotation in any n-body system. *Physical review letters*, 62(3):241, 1989.

S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa. The 3d linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 239–246 vol.1, Oct 2001. doi: 10.1109/IROS.2001.973365.

M. Kamel, S. Comari, and R. Siegwart. Full-body multi-objective controller for aerial manipulation. In *2016 24th Mediterranean Conference on Control and Automation (MED)*, pages 659–664, June 2016. doi: 10.1109/MED.2016.7536005.

Hassan K. Khalil. *NonLinear System (3rd edition)*. Pearson, 2002.

Jung-Yup Kim, Ill-Woo Park, and Jun-Ho Oh. Walking control algorithm of biped humanoid robot on uneven and inclined floor. *Journal of Intelligent and Robotic Systems*, 48(4):457–484, Apr 2007. ISSN 1573-0409. doi: 10.1007/s10846-006-9107-8. URL `https://doi.org/10.1007/s10846-006-9107-8`.

S. Kim, S. Choi, and H. J. Kim. Aerial manipulation using a quadrotor with a two dof robotic arm. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 4990–4995, Tokyo, Japan, Nov. 2013.

N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, pages 2149 – 2154, 2004.

K. Kondak, F. Huber, M. Schwarzbach, M. Laiacker, D. Sommer, M. Bejar, and A. Ollero. Aerial manipulation robot composed of an autonomous helicopter and a 7 degrees of freedom industrial manipulator. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2107–2112, May 2014. doi: 10.1109/ICRA.2014.6907148.

Konstantin Kondak, Aníbal Ollero, Ivan Maza, Kai Krieger, Alin Albu-Schaeffer, Marc Schwarzbach, and Maximilian Laiacker. *Unmanned Aerial Systems Physically Interacting with the Environment: Load Transportation, Deployment, and Aerial Manipulation*, pages 2755–2785. Handbook of Unmanned Aerial Vehicles. Springer Netherlands, Dordrecht, 2015. ISBN 978-90-481-9707-1.

doi: 10.1007/978-90-481-9707-1_77. URL https://doi.org/10.1007/978-90-481-9707-1_77.

T. J. Koo and S. Sastry. Output tracking control design of a helicopter model based on approximate linearization. In *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171)*, volume 4, pages 3635–3640 vol.4, Dec 1998. doi: 10.1109/CDC.1998.761745.

T. Koolen, J. Smith, G. Thomas, S. Bertrand, J. Carff, N. Mertins, D. Stephen, P. Abeles, J. Englsberger, S. McCrory, J. van Egmond, M. Griffioen, M. Floyd, S. Kobus, N. Manor, S. Alsheikh, D. Duran, L. Bunch, E. Morphis, L. Colasanto, K. H. Hoang, B. Layton, P. Neuhaus, M. Johnson, and J. Pratt. Summary of team ihmc's virtual robotics challenge entry. In *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 307–314, Oct 2013. doi: 10.1109/HUMANOIDS.2013.7029992.

Twan Koolen, Sylvain Bertrand, Gray Thomas, Tomas de Boer, Tingfan Wu, Jesper Smith, Johannes Englsberger, and J Pratt. Design of a momentum-based control framework and application to the humanoid robot atlas. *International Journal of Humanoid Robotics*, 13:34, March 2016.

S. Kuindersma, F. Permenter, and R. Tedrake. An efficiently solvable quadratic program for stabilizing dynamic locomotion. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2589–2594, May 2014. doi: 10.1109/ICRA.2014.6907230.

Sung-Hee Lee and A. Goswami. Ground reaction force control at each foot: A momentum-based humanoid balance controller for non-level and non-stationary ground. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3157–3162, Oct 2010. doi: 10.1109/IROS.2010.5650416.

Sung-Hee Lee and Ambarish Goswami. A momentum-based balance controller for humanoid robots on non-level and non-stationary ground. *Autonomous Robots*, 33(4):399–414, 2012. ISSN 1573-7527. doi: 10.1007/s10514-012-9294-z. URL http://dx.doi.org/10.1007/s10514-012-9294-z.

Z. Li, N. G. Tsagarakis, and D. G. Caldwell. A passivity based admittance control for stabilizing the compliant humanoid coman. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pages 43–49, Nov 2012.

A. De Luca and C. Manes. Modeling of robots in contact with a dynamic environment. *IEEE Transactions on Robotics and Automation*, 10(4):542–548, Aug 1994. ISSN 1042-296X. doi: 10.1109/70.313104.

Chi-Chung Luo, Ru-Feng Liu, Ciann-Dong Yang, and Yeong-Hwa Chang. Helicopter h infinite control design with robust flying quality. *Aerospace Science* *and Technology*, 7(2):159 – 169, 2003. ISSN 1270-9638. doi: https://doi.org/10.1016/S1270-9638(02)00012-3. URL http://www.sciencedirect.com/science/article/pii/S1270963802000123.

R. Mahony, T. Hamel, and A. Dzul. Hover control via lyapunov control for an autonomous model helicopter. In *Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No.99CH36304)*, volume 4, pages 3490–3495 vol.4, Dec 1999. doi: 10.1109/CDC.1999.827874.

N. Mansard, O. Stasse, P. Evrard, and A. Kheddar. A versatile generalized inverted kinematics implementation for collaborative working humanoid robots: The stack of tasks. In *2009 International Conference on Advanced Robotics*, pages 1–6, June 2009.

Lorenzo Marconi and Roberto Naldi. Robust full degree-of-freedom tracking control of a helicopter. *Automatica*, 43(11):1909 – 1920, 2007. ISSN 0005-1098. doi: https://doi.org/10.1016/j.automatica.2007.03.028. URL http://www.sciencedirect.com/science/article/pii/S0005109807002154.

Lorenzo Marconi and Roberto Naldi. Aggressive control of helicopters in presence of parametric and dynamical uncertainties. *Mechatronics*, 18(7):381 – 389, 2008. ISSN 0957-4158. doi: https://doi.org/10.1016/j.mechatronics.2007.10.004. URL http://www.sciencedirect.com/science/article/pii/S0957415807001201. Special Section of Revised Papers from the 8th International IFAC Symposium on Robot Control.

Jerrold E. Marsden and Tudor S. Ratiu. *Introduction to Mechanics and Symmetry: A Basic Exposition of Classical Mechanical Systems*. Springer Publishing Company, Incorporated, 2010. ISBN 1441931430, 9781441931436.

Jerrold E Marsden and Jürgen Scheurle. The reduced euler-lagrange equations. *Fields Institute Comm*, 1:139–164, 1993.

S. Mason, N. Rotella, S. Schaal, and L. Righetti. Balancing and walking using full dynamics lqr control with contact constraints. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 63–68, Nov 2016. doi: 10.1109/HUMANOIDS.2016.7803255.

Thomas A. McMahon and George C. Cheng. The mechanics of running: How does stiffness couple with speed? *Journal of Biomechanics*, 23:65 – 78, 1990. ISSN 0021-9290. doi: https://doi.org/10.1016/0021-9290(90)90042-2. URL http://www.sciencedirect.com/science/article/pii/002192909090042Z. International Society of Biomechanics.

D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar. Design, modeling, estimation and control for aerial grasping and manipulation. In *2011 IEEE/RSJ*

International Conference on Intelligent Robots and Systems, pages 2668–2673, Sep. 2011. doi: 10.1109/IROS.2011.6094871.

Giorgio Metta, Lorenzo Natale, Francesco Nori, Giulio Sandini, David Vernon, Luciano Fadiga, Claes von Hofsten, Kerstin Rosander, Manuel Lopes, José Santos-Victor, Alexandre Bernardino, and Luis Montesano. The iCub humanoid robot: An open-systems platform for research in cognitive development. *Neural Networks*, 23(8–9):1125 – 1134, 2010. ISSN 0893-6080. doi: http://dx.doi.org/10.1016/j.neunet.2010.08.010. Social Cognition: From Babies to Robots.

K. Miura, S. Nakaoka, M. Morisawa, K. Harada, and S. Kajita. A friction based twirl for biped robots. In *Humanoids 2008 - 8th IEEE-RAS International Conference on Humanoid Robots*, pages 279–284, Dec 2008. doi: 10.1109/ICHR.2008.4755965.

Abdellah Mokhtari, Abdelaziz Benallegue, and Abdelkader Belaidi. Polynomial linear quadratic gaussian and sliding mode observer for a quadrotor unmanned aerial vehicle. *Journal of Robotics and Mechatronics*, 17(4):483–495, 2005. doi: 10.20965/jrm.2005.p0483.

Jun Nakanish, Michael Mistry, and Stefan Schaal. Inverse Dynamics Control with Floating Base and Constraints. *Robotics and Automation, 2007 IEEE International Conference on*, pages 1942 – 1947, 2007.

R. Naldi, M. Furci, R. G. Sanfelice, and L. Marconi. Robust global trajectory tracking for underactuated vtol aerial vehicles using inner-outer loop control paradigms. *IEEE Transactions on Automatic Control*, 62(1):97–112, Jan 2017. ISSN 0018-9286. doi: 10.1109/TAC.2016.2557967.

Hai-Nguyen Nguyen, ChangSu Ha, and Dongjun Lee. Mechanics, control and internal dynamics of quadrotor tool operation. *Automatica*, 61:289 – 301, 2015. ISSN 0005-1098. doi: https://doi.org/10.1016/j.automatica.2015.08.015. URL `http://www.sciencedirect.com/science/article/pii/S0005109815003416`.

Francesco Nori, Silvio Traversaro, Jorhabib Eljaik, Francesco Romano, Andrea Del Prete, and Daniele Pucci. icub whole-body control through force regulation on rigid noncoplanar contacts. *Frontiers in Robotics and AI*, 2(6), 2015. ISSN 2296-9144. doi: 10.3389/frobt.2015.00006. URL `http://www.frontiersin.org/humanoid_robotics/10.3389/frobt.2015.00006/abstract`.

Reza Olfati-Saber. *Nonlinear Control of Underactuated Mechanical Systems with Application to Robotics and Aerospace Vehicles*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2001. AAI0803036.

David E. Orin and Ambarish Goswami. Centroidal momentum matrix of a humanoid robot: Structure and properties. *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 653 – 659, 2008.

DavidE. Orin, Ambarish Goswami, and Sung-Hee Lee. Centroidal dynamics of a humanoid robot. *Autonomous Robots*, 2013. ISSN 0929-5593. doi: 10.1007/s10514-013-9341-4.

V. Ortenzi, R. Stolkin, J. Kuo, and M. Mistry. Hybrid motion/force control: a review. *Advanced Robotics*, 31(19-20):1102–1113, 2017.

C. Ott, M.A. Roa, and G. Hirzinger. Posture and balance control for biped robots based on contact force optimization. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages 26–33, Oct 2011. doi: 10.1109/Humanoids.2011.6100882.

Elena Panteley, Romeo Ortega, and Magnus Gäfvert. An adaptive friction compensator for global tracking in robot manipulators. *Systems and Control Letters*, 33(5):307 – 313, 1998. ISSN 0167-6911. doi: https://doi.org/10.1016/S0167-6911(97)00113-8. URL `http://www.sciencedirect.com/science/article/pii/S0167691197001138`.

A. Parmiggiani, G. Metta, and N. Tsagarakis. The mechatronic design of the new legs of the icub robot. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pages 481–486, Nov 2012. doi: 10.1109/HUMANOIDS.2012.6651563.

L. Penco, B. Clement, V. Modugno, E. Mingo Hoffman, G. Nava, D. Pucci, N. G. Tsagarakis, J. . Mouret, and S. Ivaldi. Robust real-time whole-body motion retargeting from human to humanoid. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pages 425–432, Nov 2018. doi: 10.1109/HUMANOIDS.2018.8624943.

J.-M. Pflimlin, P. Binetti, P. Souères, T. Hamel, and D. Trouchet. Modeling and attitude control analysis of a ducted-fan micro aerial vehicle. *Control Engineering Practice*, 18(3):209 – 218, 2010. ISSN 0967-0661. doi: https://doi.org/10.1016/j.conengprac.2009.09.009. URL `http://www.sciencedirect.com/science/article/pii/S0967066109001828`.

Pauline Pounds, Daniel Bersak, and Aaron Dollar. Grasping from the air: Hovering capture and load stability. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2491 – 2498, 06 2011. doi: 10.1109/ICRA.2011.5980314.

Gill A. Pratt, Matthew M. Williamson, Peter Dillworth, Jerry Pratt, and Anne Wright. *Stiffness isn't everything*, pages 253–262. Experimental Robotics IV: The 4th International Symposium, Stanford, California, June 30 – July 2, 1995.

Springer Berlin Heidelberg, Berlin, Heidelberg, 1997. ISBN 978-3-540-40942-7. doi: 10.1007/BFb0035216. URL `http://dx.doi.org/10.1007/BFb0035216`.

J. Pratt, J. Carff, S. Drakunov, and A. Goswami. Capture point: A step toward humanoid push recovery. In *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pages 200–207, Dec 2006. doi: 10.1109/ICHR.2006.321385.

J.E. Pratt and R. Tedrake. *Fast Motions in Biomechanics and Robotics: Optimization and Feedback Control. Ch.: Velocity-Based Stability Margins for Fast Bipedal Walking*, pages 299–324. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-36119-0. doi: 10.1007/978-3-540-36119-0_14. URL `https://doi.org/10.1007/978-3-540-36119-0_14`.

Jerry Pratt, Twan Koolen, Tomas de Boer, John Rebula, Sebastien Cotton, John Carff, Matthew Johnson, and Peter Neuhaus. Capturability-based analysis and control of legged locomotion, part 2: Application to m2v2, a lower-body humanoid. *The International Journal of Robotics Research*, 31(10):1117–1133, 2012. doi: 10.1177/0278364912452762. URL `https://doi.org/10.1177/0278364912452762`.

D. Pucci, S. Traversaro, and F. Nori. Momentum control of an underactuated flying humanoid robot. *IEEE Robotics and Automation Letters*, 3(1):195–202, Jan 2018. doi: 10.1109/LRA.2017.2734245.

R. Rashad, F. Califano, and S. Stramigioli. Port-hamiltonian passivity-based control on se (3) of a fully actuated uav for aerial physical interaction near-hovering. *IEEE Robotics and Automation Letters*, 4(4):4378–4385, 2019.

Ludovic Righetti, Jonas Buchli, Michael Mistry, and Stefan Schaal. Inverse dynamics control of floating-base robots with external constraints: A unified view. *IEEE International Conference on Robotics and Automation*, May 2011.

F. Romano, S. Traversaro, D. Pucci, J. Eljaik, A. D. Prete, and F. Nori. A whole-body software abstraction layer for control design of free-floating mechanical systems. In *2017 First IEEE International Conference on Robotic Computing (IRC)*, pages 148–155, April 2017. doi: 10.1109/IRC.2017.43.

Clément Roos, Carsten Döll, and Jean-Marc BIANNIC. Flight control laws: Recent advances in the evaluation of their robustness properties. *AerospaceLab Journal*, 05 2012.

M. Ryll, G. Muscio, F. Pierri, E. Cataldi, G. Antonelli, F. Caccavale, D. Bicego, and A. Franchi. 6D interaction control with aerial robots: The flying end-effector paradigm. *The International Journal of Robotics Research*, 38(9):1045–1062, 2019. doi: 10.1177/0278364919856694.

A. Saccon, S. Traversaro, F. Nori, and H. Nijmeijer. On centroidal dynamics and integrability of average angular velocity. *IEEE Robotics and Automation Letters*, 2(2):943–950, April 2017.

A. Santamaria-Navarro, V. Lippiello, and J. Andrade-Cetto. Task priority control for aerial manipulation. In *IEEE Int. Symp. on Safety, Security and Rescue Robotics*, pages 1–6, Oct 2014. doi: 10.1109/SSRR.2014.7017672.

Philippe Sardain and Guy Bessonnet. Forces acting on a biped robot. center of pressure—zero moment point. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 34:630 – 637, 10 2004. doi: 10.1109/TSMCA.2004.832811.

Jonathan Selig. *Geometric Fundamentals of Robotics*. Springer-Verlag New York, 01 2005. ISBN 0387208747.

M.W. Spong. *Control of Flexible Joint Robots: A Survey*. UILU-ENG / 22: UILU-ENG. University of Illinois, Coordinated Science Laboratory, 1990. URL `https://books.google.it/books?id=6EyhPgAACAAJ`.

N. Staub, D. Bicego, Q. Sablé, V. Arellano-Quintana, S. Mishra, and A. Franchi. Towards a flying assistant paradigm: the OTHex. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 6997–7002, Brisbane, Australia, May 2018.

B.J. Stephens and C.G. Atkeson. Dynamic balance force control for compliant humanoid robots. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1248–1255, Oct 2010. doi: 10.1109/IROS.2010.5648837.

R. H. Stone. Control architecture for a tail-sitter unmanned air vehicle. In *2004 5th Asian Control Conference (IEEE Cat. No.04EX904)*, volume 2, pages 736–744 Vol.2, July 2004.

John Stuelpnagel. On the parametrization of the three-dimensional rotation group. *SIAM Review. Society for Industrial and Applied Mathematics*, 6(4):422–430, 1964. ISSN 00361445. URL `http://www.jstor.org/stable/2027966`.

A. Suarez, G. Heredia, and A. Ollero. Physical-virtual impedance control in ultra-lightweight and compliant dual-arm aerial manipulators. *IEEE Robotics and Automation Letters*, 3(3):2553–2560, July 2018. doi: 10.1109/LRA.2018.2809964.

Toru Takenaka, Takashi Matsumoto, and Takahide Yoshiike. Real-time dynamics compensation with considering ground reaction force and moment limit for biped robot. *Journal of the Robotics Society of Japan*, 32:295–306, 01 2014. doi: 10.7210/jrsj.32.295.

M. Teshnehlab and K. Watanabe. Self tuning of computed torque gains by using neural networks with flexible structures. *IEE Proceedings - Control Theory and Applications (Volume:141 , Issue: 4)*, Feb 2002.

Stefano Tisi, Stefano Saliceti, Daniele Pucci, Paolo Silvestri, Francesco Nori, and Giorgio Metta. Design and validation of a series rotary elastic actuator for humanoid robots. In *Proceedings of the 19th International Conference on CLAWAR 2016*, Oct 2016.

M. Tognon, B. Yüksel, G. Buondonno, and A. Franchi. Dynamic decentralized control for protocentric aerial manipulators. In *IEEE Int. Conf. on Robotics and Automation*, pages 6375–6380, Singapore, May 2017.

M. Tognon, H. A. Tello Chávez, E. Gasparin, Q. Sablé, D. Bicego, A. Mallet, M. Lany, G. Santi, B. Revaz, J. Cortés, and A. Franchi. A truly redundant aerial manipulator system with application to push-and-slide inspection in industrial plants. *IEEE Robotics and Automation Letters*, 4(2):1846–1851, 2019. doi: 10.1109/LRA.2019.2895880.

Silvio Traversaro, Daniele Pucci, and Francesco Nori. A unified view of the equations of motion used for control design of humanoid robots. *Submitted to Multibody System Dynamics*, 01 2017.

N. G. Tsagarakis, M. Laffranchi, B. Vanderborght, and D. G. Caldwell. A compact soft actuator unit for small scale human friendly robots. In *2009 IEEE International Conference on Robotics and Automation*, pages 4356–4362, May 2009. doi: 10.1109/ROBOT.2009.5152496.

Miomir Vukobratovic and Branislav Borovac. Zero-moment point - thirty five years of its life. *I. J. Humanoid Robotics*, 1:157–173, 03 2004. doi: 10.1142/S0219843604000083.

P.M. Wensing and D.E. Orin. Generation of dynamic humanoid behaviors through task-space control with conic optimization. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3103–3109, May 2013. doi: 10.1109/ICRA.2013.6631008.

Pierre-Brice Wieber, Russ Tedrake, and Scott Kuindersma. *Modeling and Control of Legged Robots*, pages 1203–1234. Springer Handbook of Robotics. Springer International Publishing, Cham, 2016. ISBN 978-3-319-32552-1. doi: 10.1007/978-3-319-32552-1_48. URL `https://doi.org/10.1007/978-3-319-32552-1_48`.

M. Yagi and V. Lumelsky. Synthesis of turning pattern trajectories for a biped robot in a scene with obstacles. In *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113)*, volume 2, pages 1161–1166 vol.2, Oct 2000. doi: 10.1109/IROS.2000.893176.

J. Yamaguchi, E. Soga, S. Inoue, and A. Takanishi. Development of a bipedal humanoid robot-control method of whole body cooperative dynamic biped walking. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, volume 1, pages 368–374 vol.1, May 1999. doi: 10.1109/ROBOT.1999.770006.

H. Yang and D. J. Lee. Dynamics and control of quadrotor with robotic manipulator. In *IEEE Int. Conf. on Robotics and Automation*, pages 5544–5549, Hong Kong, China, May 2014.

Appendices

Appendix A

Proof of Lemma 2

Recall that in view of Lemma 1, the mass matrix M is block diagonal. The joint space dynamics is given by the last n rows of Eq. (3.4):

$$M_s \ddot{s} = J_s^\top f - h_s + \tau. \tag{A.1}$$

Moreover, we can rewrite the term $JM^{-1}(h - J^\top f^*)$ in the control torques equations Eq. (6.6) as follows:

$$JM^{-1}(h - J^\top f^*) = J_\mathcal{B} M_\mathcal{B}^{-1}(h_\mathcal{B} - J_\mathcal{B}^\top f^*) + \Lambda(h_s - J_s^\top f^*).$$

In view of $N_\Lambda = 1_n - \Lambda^\dagger \Lambda$ and the choice of τ_0 as in (6.7), the control torques (6.6) can be simplified as:

$$\tau = h_s - J_s^\top f^* + \Lambda^\dagger (J_\mathcal{B} M_\mathcal{B}^{-1}(h_\mathcal{B} - J_\mathcal{B}^\top f^*) - \dot{J}\nu) + N_\Lambda u_0, \tag{A.2}$$

with $u_0 = M_s \ddot{s}^d - K_D^s(\dot{s} - \dot{s}^d) - K_P^s(s - s^d)$. Substituting Eq. (A.2) into Eq. (A.1), and assuming that at every instant $f = f^*$, gives:

$$M_s \ddot{s} = \Lambda^\dagger (J_\mathcal{B} M_\mathcal{B}^{-1}(h_\mathcal{B} - J_\mathcal{B}^\top f^*) - \dot{J}\nu) + N_\Lambda u_0. \tag{A.3}$$

The only term which contains the wrenches f^* in Eq. (A.3) is multiplied by $J_\mathcal{B}^\top$. Since $f^* = f_1 + N_\mathcal{B} f_0$, and by definition $J_\mathcal{B}^\top N_\mathcal{B} = 0_{6n_c}$, we have that $J_\mathcal{B}^\top f^* = J_\mathcal{B}^\top f_1$. Hence, vector f_0 does not influence the joint space dynamics Eq. (A.3).

Appendix B

Proof of Lemma 3

The proof is composed of two steps. First, we linearize the constrained closed loop dynamics around the equilibrium point $(s^d, 0)$. Then, by means of Lyapunov analysis, we show that the equilibrium point is asymptotically stable.

B.1 Linearization

Consider that Assumption 1 holds, and that we apply the control laws (6.4)–(6.6)–(6.7) with the gains as (7.3)–(7.4). The closed loop joint space dynamics of system (3.4) constrained by (3.7) is given by the following equation:

$$M_s \ddot{s} + h_s - J_s^\top f = \tau \tag{B.1}$$

Now, rewrite the joint torques in (6.6) as follows: $\tau = \Lambda^+(\Lambda(h_s - J_s^\top f) + J_\mathcal{B} M_\mathcal{B}^{-1}(h_\mathcal{B} - J_\mathcal{B}^\top f) - \dot{J}\nu) + N_\Lambda \tau_0$. Therefore, Eq. (B.1) can be simplified into:

$$\ddot{s} = M_s^{-1}(\Lambda^+(J_\mathcal{B} M_\mathcal{B}^{-1}(h_\mathcal{B} - J_\mathcal{B}^\top f) - \dot{J}\nu) - N_\Lambda u_0) \tag{B.2}$$

where $u_0 = K_P^s(s - s^d) + K_D^s \dot{s}$, while $h_\mathcal{B} = C_\mathcal{B}\nu_\mathcal{B} + C_{\mathcal{B}s}\dot{s} + mge_3$, and $C_\mathcal{B} \in \mathbb{R}^{6\times 6}$, $C_{\mathcal{B}s} \in \mathbb{R}^{6\times n}$, $C_{s\mathcal{B}} \in \mathbb{R}^{n\times 6}$, $C_s \in \mathbb{R}^{n\times n}$ are obtained from the following partition of the Coriolis matrix:

$$C = \begin{bmatrix} C_\mathcal{B} & C_{\mathcal{B}s} \\ C_{s\mathcal{B}} & C_s \end{bmatrix}$$

Substituting (6.4) into (B.2) and grouping together the terms that are linear with respect of joint velocity yield:

$$\ddot{s} = -M_s^{-1}\left[\Lambda^\dagger(J_\mathcal{B} M_\mathcal{B}^{-1}\dot{L}^* + \Psi\dot{s}) + N_\Lambda u_0\right] \tag{B.3}$$

where $\Psi = \dot{J}_s - J_{\mathcal{B}} M_{\mathcal{B}}^{-1} C_{\mathcal{B}s} + (J_{\mathcal{B}} M_{\mathcal{B}}^{-1} C_{\mathcal{B}} - \dot{J}_{\mathcal{B}}) J_{\mathcal{B}}^{-1} J_s$. Define the state x as $x := \begin{bmatrix} x_1^\top & x_2^\top \end{bmatrix}^\top = \begin{bmatrix} s^\top - s^{d\top} & \dot{s}^\top \end{bmatrix}^\top$. Since $L^d \equiv 0$, the linearized dynamical system about the equilibrium point $(s^d, 0)$ is given by

$$\dot{x} = \begin{bmatrix} \partial_s \dot{x}_1 & \partial_{\dot{s}} \dot{x}_1 \\ \partial_s \dot{x}_2 & \partial_{\dot{s}} \dot{x}_2 \end{bmatrix} x = \begin{bmatrix} 0_{n\times n} & 1_n \\ A_1 & A_2 \end{bmatrix} x \tag{B.4}$$

To find the matrices $A_1, A_2 \in \mathbb{R}^{n\times n}$, one has to evaluate the following partial derivatives

$$\begin{aligned} \partial_y \ddot{s} = &- \sum_{i=1}^{6} \partial_y (M_s^{-1} \Lambda^\dagger J_{\mathcal{B}} M_{\mathcal{B}}^{-1} e_i) e_i^\top \dot{L}^* - M_s^{-1} N_\Lambda \partial_y u_0 \\ &- \sum_{i=1}^{n} \partial_y (M_s^{-1} N_\Lambda e_i) e_i^\top u_0 - M_s^{-1} \Lambda^\dagger J_{\mathcal{B}} M_{\mathcal{B}}^{-1} \partial_y \dot{L}^* \end{aligned}$$

with $y = \{s, \dot{s}\}$. Note that $\dot{L}^* = 0$ and $u_0 = 0$ when evaluated at $s = s^d$ and $\dot{s} = 0$. We thus have to compute only the partial derivatives of $\dot{L}^*$ and u_0. The latter is trivially given by $\partial_s u_0 = \overline{K}_P^s N_\Lambda M_s$ and $\partial_{\dot{s}} u_0 = \overline{K}_D^s N_\Lambda M_s$. The former can be calculated via Eq. (7.6). In light of the above we obtain the expressions of the matrices in (B.4):

$$\begin{aligned} A_1(s^d) &= -M_s^{-1} \Lambda^\dagger J_{\mathcal{B}} M_{\mathcal{B}}^{-1} K_I M_{\mathcal{B}} J_{\mathcal{B}}^{-1} J_s - M_s^{-1} N_\Lambda \overline{K}_P^s N_\Lambda M_s \\ A_2(s^d) &= -M_s^{-1} \Lambda^\dagger J_{\mathcal{B}} M_{\mathcal{B}}^{-1} K_P M_{\mathcal{B}} J_{\mathcal{B}}^{-1} J_s - M_s^{-1} N_\Lambda \overline{K}_D^s N_\Lambda M_s. \end{aligned}$$

B.2 Proof of Asymptotic Stability

Consider now the following Lyapunov candidate:

$$V(x) = \frac{1}{2} \left[x_1^\top M_s^\top Q_1 M_s x_1 + x_2^\top M_s^\top Q_2 M_s x_2 \right]$$

where $M_s = M_s(s^d)$, and

$$\begin{aligned} Q_1 &:= \Lambda^\top J_{\mathcal{B}}^{-\top} M_{\mathcal{B}}^\top K_I M_{\mathcal{B}} J_{\mathcal{B}}^{-1} \Lambda + N_\Lambda \overline{K}_P^s N_\Lambda \\ Q_2 &:= \Lambda^\top J_{\mathcal{B}}^{-\top} M_{\mathcal{B}}^\top M_{\mathcal{B}} J_{\mathcal{B}}^{-1} \Lambda + N_\Lambda \end{aligned}$$

calculated at $x_1 = 0$, $x_2 = 0$. V is a properly defined Lyapunov candidate, in fact $V = 0 \iff x = 0$ and is positive definite otherwise. Indeed, Q_1 can be rewritten in the following way:

$$Q_1 = \begin{bmatrix} \Lambda^\top & N_\Lambda \end{bmatrix} \begin{bmatrix} J_{\mathcal{B}}^{-\top} M_{\mathcal{B}}^\top K_I M_{\mathcal{B}} J_{\mathcal{B}}^{-1} & 0 \\ 0 & \overline{K}_P^s \end{bmatrix} \begin{bmatrix} \Lambda \\ N_\Lambda \end{bmatrix}.$$

and, because Λ and N_Λ are orthogonal Q_1 is positive definite. The same reasoning can be applied to Q_2. We can now consider the time derivative of V:

$$\begin{aligned}\dot{V} &= x_1^\top M_s^\top Q_1 M_s x_2 + x_2^\top M_s^\top Q_2 M_s \dot{x}_2 \\ &= -x_2^\top M_s^\top (\Lambda^\top J_{\mathcal{B}}^{-\top} M_{\mathcal{B}}^\top K_P M_{\mathcal{B}} J_{\mathcal{B}}^{-1} \Lambda \\ &\quad + N_\Lambda K_D^s N_\Lambda) M_s x_2 \leq 0.\end{aligned}$$

The stability of the equilibrium point $x = 0$ associated with the linear system (B.4) thus follows. To prove the asymptotic stability of the equilibrium point $x = 0$, which implies its asymptotic stability when associated with the nonlinear system (B.3), we have to resort to LaSalle's invariance principle. Let us define the invariant set $S := \{x : \dot{V}(x) = 0\}$ that implies $S = \{(x_1, 0)\}$. It is easy to verify that the only trajectory starting in S and remains in S is given by $x_1 = 0$ thus proving LaSalle's principle. As a consequence, the equilibrium point $x = 0 \Rightarrow (s, \dot{s}) = (s^d, 0)$ is asymptotically stable.

Appendix C

Proof of Lemma 4–5

C.1 Proof of Lemma 4

Given two rectangular matrices A, B, recall the following properties: $\mathrm{rank}\,(AB) \leq \min(\mathrm{rank}\,(A), \mathrm{rank}\,(B))$ and $\mathrm{rank}\,(B \otimes A) = \mathrm{rank}\,(A)\,\mathrm{rank}\,(B)$, where $\mathrm{rank}\,(\,\cdot\,)$ denotes the rank of a matrix. We apply the above properties to evaluate the rank of the matrices C_1, C_2, C_3, C_4 in Eq. (8.1a). It is straightforward to verify that: $\mathrm{rank}(C_1) \leq 6$, $\mathrm{rank}(C_2) \leq 6$, $\mathrm{rank}(C_3) \leq n - 6n_c$, $\mathrm{rank}(C_4) \leq n - 6n_c$. It is now possible to evaluate the rank of matrix

$$\Psi = \begin{bmatrix} [C_2^\top \otimes C_1] & [C_4^\top \otimes C_3] \end{bmatrix} \in \mathbb{R}^{n^2 \times (n^2+36)},$$

i.e.

$$\mathrm{rank}(\Psi) \leq \mathrm{rank}(C_2^\top \otimes C_1) + \mathrm{rank}(C_4^\top \otimes C_3) \leq 36 + (n - 6n_c)^2.$$

The condition for Ψ to be full row rank is $\mathrm{rank}(\Psi) = n^2$. This condition may be verified if $36 + (n - 6n_c)^2 = n^2$. Recall that n, n_c must be positive integers, and that $n > 6n_c$. Assume that $n = 6n_c + k$, with $k \in \mathbb{N}$. Then, one can verify that $36 + (n - 6n_c)^2 < n^2$ yields $36(n_c^2 - 1) + 12kn_c > 0$, which is always satisfied for any $n_c, k \in \mathbb{N}$. As a consequence, $\mathrm{rank}(\Psi) < n^2$.

C.2 Proof of Lemma 5

Using the properties of Frobenius matrix norms, one has that $|\tilde{K}|^2 = |K^* - X|^2 = \mathrm{tr}\,((K^* - X)^\top (K^* - X))$, where $\mathrm{tr}(\,\cdot\,)$ denotes the trace operator. Consider now the candidate Lyapunov function

$$V = |\tilde{K}|^2 = \mathrm{tr}\,((K^* - X(O,D))^\top (K^* - X(O,D))) \tag{C.1}$$

Observe that V is always positive, and $V = 0$ iif $\tilde{K} = 0$. Then, to prove the three statements in Lemma 5, it suffices to show that $\dot{V} \leq 0$. To do this, recall that O is an orthogonal matrix, i.e. $OO^\top = 1$. Observe that Eq. (C.1) can be rewritten as:

$$V = \operatorname{tr}\left(K^* K^{*\top} - 2K^{*\top} O^\top \exp(D) O + \exp(2D)\right) \tag{C.2}$$

where we used the properties $\operatorname{tr}(K^{*\top} O^\top \exp(D) O) = \operatorname{tr}(O^\top \exp(D) O K^*)$ and $\operatorname{tr}(O^\top \exp(2D) O) = \operatorname{tr}(OO^\top \exp(2D))$. To compute the time derivative of V, recall that $\dot{O} = OS(v)$, with $S(v)$ a skew-symmetric matrix, and $\dot{D} = U$. Then, $\dot{V}$ becomes:

$$\dot{V} = 2\operatorname{tr}(B_1 \exp(D) U) + 2\operatorname{tr}(B_2 S(v)),$$

where we defined $B_2 = O^\top \exp(D) O K^{*\top} - K^{*\top} O^\top \exp(D) O$, while $B_1 = \exp(D) - OK^* O^\top$. A way to ensure a negative $\dot{V}$ is to impose:

$$\operatorname{tr}(B_1 \exp(D) U) \leq 0 \tag{C.3a}$$

$$\operatorname{tr}(B_2 S(v)) \leq 0. \tag{C.3b}$$

Now, note that the term (C.3a) can be rewritten as $\sum_{i=1}^n e_i^\top B_1 \exp(D) U e_i$. Since the product $\exp(D)U$ is diagonal, then:

$$\sum_{i=1}^n e_i^\top B_1 \exp(D) U e_i = \sum_{i=1}^n e_i^\top B_1 e_i \exp(d_i) u_i,$$

where we indicate with $\exp(d_i)u_i$ the i-th element along the diagonal of $\exp(D)U$. The trace can then be rewritten as $\sum_{i=1}^n e_i^\top B_1 e_i \exp(d_i) u_i = \sum_{i=1}^n b_{1i} \exp(d_i) u_i$ where b_{1i} is the i-th element along the diagonal of B_1. A possible choice of u_i that ensures $\operatorname{tr}(B_1 \exp(D) U) \leq 0$ is:

$$u_i = -k_{Ui} \exp(-d_i) b_{1i} \quad k_{Ui} > 0. \tag{C.4}$$

Since u_i is the i-th element along the diagonal of U, Eq. (C.4) implies that $U = -K_U \exp(-D)\operatorname{diag}(B_1)$ with K_U a diagonal matrix of positive constants.

Now, recall that the matrix B_2 can be decomposed as follows:

$$B_2 = \tfrac{(B_2 + B_2^\top)}{2} + S(\omega) \tag{C.5}$$

where $\omega = \left(\tfrac{(B_2 - B_2^\top)}{2}\right)^\vee$. Recall also that the trace of a product between a symmetric and a skew-symmetric matrix is zero. Then, $\operatorname{tr}\left(\tfrac{(B_2 + B_2^\top)}{2} S(v)\right) = 0$. We are now left to evaluate $\operatorname{tr}(S(\omega)S(v))$. The trace of a matrix product can also be written as $\operatorname{tr}(X^\top Y) = \operatorname{vec}(X)^\top \operatorname{vec}(Y)$, where $\operatorname{vec}(\,\cdot\,)$ is the vectorization operator. Then $\operatorname{tr}(S(\omega)S(v)) = -\operatorname{tr}(S(\omega)^\top S(v)) = -\operatorname{vec}((S(\omega))^\top \operatorname{vec}(S(v))$.

Note that $\mathrm{vec}(S(x)) = Tx \; \forall x$, where the matrix T satisfies $\frac{T^\top T}{2} = 1$ due to the skew-symmetry of $S(\,\cdot\,)$. Hence,

$$\mathrm{tr}(S(\omega)^\top S(v)) = -\omega^\top T^\top T v = -2\omega^\top v$$

and this suggests that a possible choice of v is $v = K_v S(\omega)^\vee = K_v \left(\frac{(B_2 - B_2^\top)}{2}\right)^\vee$.

Appendix D

Modeling of Seesaw Dynamics and Constraints

D.1 Seesaw Dynamics in Frame $\mathcal{S}$

Let us define with $\mathcal{S}[\mathcal{I}]$ a reference frame whose origin is at the seesaw center of mass, and with the orientation of the inertial frame $\mathcal{I}$. The rate of change of seesaw momentum, when projected in this frame, is given by:

$$^{\mathcal{S}[\mathcal{I}]}\dot{L}_{\mathcal{S}} = \; - m_{\mathcal{S}} g e_3 - \bar{J}_R^\top f + \bar{J}_{\mathcal{S}}^\top f_{\mathcal{S}} \tag{D.1}$$

where the matrices $\bar{J}_R$ and $\bar{J}_{\mathcal{S}}$ are defined in the next subsection D.2. Note that the mapping between frame $\mathcal{S}[\mathcal{I}]$ and the seesaw frame $\mathcal{S}$ is given by the relative rotation between the inertial frame and the seesaw frame, i.e. $^{\mathcal{I}}R_{\mathcal{S}}$. Therefore, the projection of the seesaw momentum $^{\mathcal{S}[\mathcal{I}]}L_{\mathcal{S}}$ into the seesaw frame $\mathcal{S}$ is given by:

$$^{\mathcal{S}[\mathcal{I}]}L_{\mathcal{S}} = {}^{\mathcal{I}}\bar{R}_{\mathcal{S}} {}^{\mathcal{S}}L_{\mathcal{S}} = \begin{bmatrix} ^{\mathcal{I}}R_{\mathcal{S}} & 0_3 \\ 0_3 & ^{\mathcal{I}}R_{\mathcal{S}} \end{bmatrix} {}^{\mathcal{S}}L_{\mathcal{S}}. \tag{D.2}$$

By differentiating Eq. (D.2), one has:

$$^{\mathcal{S}[\mathcal{I}]}\dot{L}_{\mathcal{S}} = {}^{\mathcal{I}}\dot{\bar{R}}_{\mathcal{S}} {}^{\mathcal{S}}L_{\mathcal{S}} + {}^{\mathcal{I}}\bar{R}_{\mathcal{S}} {}^{\mathcal{S}}\dot{L}_{\mathcal{S}}. \tag{D.3}$$

Recall that $^{\mathcal{S}}L_{\mathcal{S}} = M_{\mathcal{S}} {}^{\mathcal{S}}\nu_{\mathcal{S}}$, and that the derivative of a rotation matrix is given by: $^{\mathcal{I}}\dot{R}_{\mathcal{S}} = {}^{\mathcal{I}}R_{\mathcal{S}} S({}^{\mathcal{S}}\omega_{\mathcal{S}})$, being $^{\mathcal{S}}\omega_{\mathcal{S}}$ the angular velocity of the seesaw projected in the seesaw frame. Also, in the seesaw frame the mass matrix $M_{\mathcal{S}}$ is constant. Therefore, one has:

$$^{\mathcal{S}[\mathcal{I}]}\dot{L}_{\mathcal{S}} = {}^{\mathcal{I}}\bar{R}_{\mathcal{S}} (\bar{S}({}^{\mathcal{S}}\omega_{\mathcal{S}}) M_{\mathcal{S}} {}^{\mathcal{S}}\nu_{\mathcal{S}} + M_{\mathcal{S}} {}^{\mathcal{S}}\dot{\nu}_{\mathcal{S}}). \tag{D.4}$$

with $\bar{S}({}^{\mathcal{S}}\omega_{\mathcal{S}})$ a proper block diagonal matrix. Finally, by substituting Eq. (D.4) into (D.1), and multiplying both sides by ${}^{\mathcal{I}}\bar{R}_{\mathcal{S}}^{-1} =^{\mathcal{I}} \bar{R}_{\mathcal{S}}^{\top}$, one is left with Eq. (9.1), where we define:

$$\begin{aligned}
J_R &= \bar{J}_R{}^{\mathcal{I}}\bar{R}_{\mathcal{S}} \\
J_{\mathcal{S}} &= \bar{J}_{\mathcal{S}}{}^{\mathcal{I}}\bar{R}_{\mathcal{S}} \\
h_{\mathcal{S}} &= \bar{S}({}^{\mathcal{S}}\omega_{\mathcal{S}})M_{\mathcal{S}}{}^{\mathcal{S}}\nu_{\mathcal{S}} +^{\mathcal{I}} \bar{R}_{\mathcal{S}}^{\top} m_{\mathcal{S}} g e_3
\end{aligned}$$

D.2 Derivation of Matrices J_R and $J_{\mathcal{S}}$

Recall the vector of feet linear and angular velocities expressed in the inertial frame:

$${}^{\mathcal{I}}\mathrm{v}_{feet} = \begin{bmatrix} {}^{\mathcal{I}}v_{Lfoot} \\ {}^{\mathcal{I}}\omega_{Lfoot} \\ {}^{\mathcal{I}}v_{Rfoot} \\ {}^{\mathcal{I}}\omega_{Rfoot} \end{bmatrix}.$$

The constraint of having the feet always attached to the seesaw implies that ${}^{\mathcal{I}}\omega_{Lfoot} = {}^{\mathcal{I}}\omega_{Rfoot} = {}^{\mathcal{I}}\omega_{\mathcal{S}}$. Also, one has:

$$\begin{aligned}
{}^{\mathcal{I}}v_{Lfoot} &= {}^{\mathcal{I}}v_{\mathcal{S}} - S({}^{\mathcal{I}}p_{\mathcal{S}L}){}^{\mathcal{I}}\omega_{\mathcal{S}} \\
{}^{\mathcal{I}}v_{Rfoot} &= {}^{\mathcal{I}}v_{\mathcal{S}} - S({}^{\mathcal{I}}p_{\mathcal{S}R}){}^{\mathcal{I}}\omega_{\mathcal{S}}
\end{aligned}$$

where ${}^{\mathcal{I}}p_{\mathcal{S}L}$, ${}^{\mathcal{I}}p_{\mathcal{S}R}$ represent the distance between the seesaw CoM and the left and right foot, respectively. Then,

$${}^{\mathcal{I}}\mathrm{v}_{feet} = \begin{bmatrix} 1_3 & -S({}^{\mathcal{I}}p_{\mathcal{S}L}) \\ 0_3 & 1_3 \\ 1_3 & -S({}^{\mathcal{I}}p_{\mathcal{S}R}) \\ 0_3 & 1_3 \end{bmatrix} {}^{\mathcal{I}}\nu_{\mathcal{S}} = \bar{J}_R{}^{\mathcal{I}}\nu_{\mathcal{S}}$$

Analogously, the constraint of only rolling implies that the linear velocity at the contact point P between the seesaw and the ground is given by: ${}^{\mathcal{I}}v_P = {}^{\mathcal{I}}v_{\mathcal{S}} - S({}^{\mathcal{I}}p_{\mathcal{S}P}){}^{\mathcal{I}}\omega_{\mathcal{S}} = 0$, thus implying ${}^{\mathcal{I}}v_{\mathcal{S}} = S({}^{\mathcal{I}}p_{\mathcal{S}P}){}^{\mathcal{I}}\omega_{\mathcal{S}}$. The variable ${}^{\mathcal{I}}p_{\mathcal{S}P}$ represents the distance between the seesaw CoM and the contact point P. Also, constraining the rotation along y and z axis implies the second and third component of the seesaw angular velocity are given by: $e_2^{\top}{}^{\mathcal{I}}\omega_{\mathcal{S}} = e_3^{\top}{}^{\mathcal{I}}\omega_{\mathcal{S}} = 0$. Then, one has:

$$\begin{bmatrix} 1_3 & -S({}^{\mathcal{I}}p_{\mathcal{S}P}) \\ 0_{1,3} & e_2^{\top} \\ 0_{1,3} & e_3^{\top} \end{bmatrix} {}^{\mathcal{I}}\nu_{\mathcal{S}} = \bar{J}_{\mathcal{S}}{}^{\mathcal{I}}\nu_{\mathcal{S}} = 0$$

The matrices $J_{\mathcal{S}}$ and J_R are then obtained from $\bar{J}_{\mathcal{S}}$ and $\bar{J}_R$ as described in the previous Section.

D.3 Total Momentum Rate of Change

The total linear and angular momentum of the system is obtained as a combination of the robot and seesaw momentum:

$$^{T}L_T = {^{T}X^*_{\mathcal{G}[\mathcal{I}]}}\,{^{\mathcal{G}[\mathcal{I}]}L} + {^{T}X^*_{\mathcal{S}[\mathcal{I}]}}\,{^{\mathcal{S}[\mathcal{I}]}L_{\mathcal{S}}}. \tag{D.5}$$

For not burdening the notation, from now on we drop the superscripts denoting the frames with respect to which $L_T, L_{\mathcal{S}}$ and L are expressed. The transformation matrices in the space of wrenches ${^{T}X^*_{\mathcal{G}[\mathcal{I}]}}$ and ${^{T}X^*_{\mathcal{S}[\mathcal{I}]}}$ are of the following form:

$$^{T}X^*_x = \begin{bmatrix} 1_3 & 0_3 \\ S(^{\mathcal{I}}p_x - {^{\mathcal{I}}p_T}) & 1_3 \end{bmatrix}$$

where $^{\mathcal{I}}p_x = {^{\mathcal{I}}p_c}$ for the robot momentum and $^{\mathcal{I}}p_x = {^{\mathcal{I}}p_{\mathcal{S}}}$ for the seesaw momentum. Then, the derivative of Eq. (D.5) is given by:

$$\dot{L}_T = {^{T}\dot{X}^*_{\mathcal{G}[\mathcal{I}]}}L + {^{T}\dot{X}^*_{\mathcal{S}[\mathcal{I}]}}L_{\mathcal{S}} + {^{T}X^*_{\mathcal{G}[\mathcal{I}]}}\dot{L} + {^{T}X^*_{\mathcal{S}[\mathcal{I}]}}\dot{L}_{\mathcal{S}}.$$

Recall that the system's center of mass position is related to the robot and seesaw center of mass positions as follows: $^{\mathcal{I}}p_T = \frac{m^{\mathcal{I}}p_c + m_{\mathcal{S}}{^{\mathcal{I}}p_{\mathcal{S}}}}{m+m_{\mathcal{S}}}$. Also, recall the property $S(x)x = 0$. Then, it is straightforward to verify that ${^{T}\dot{X}^*_{\mathcal{G}[\mathcal{I}]}}L + {^{T}\dot{X}^*_{\mathcal{S}[\mathcal{I}]}}L_{\mathcal{S}} = 0$, and therefore the rate of change of system's momentum remains:

$$\dot{L}_T = {^{T}X^*_{\mathcal{G}[\mathcal{I}]}}\dot{L} + {^{T}X^*_{\mathcal{S}[\mathcal{I}]}}\dot{L}_{\mathcal{S}}. \tag{D.6}$$

By substituting now Eq. (6.1) and (D.1) into Eq. (D.6), one has:

$$\begin{aligned} \dot{L}_T = {} & {^{T}X^*_{\mathcal{G}[\mathcal{I}]}}(J_{\mathcal{B}}^\top f - mge_3) + \\ & {^{T}X^*_{\mathcal{S}[\mathcal{I}]}}(-m_{\mathcal{S}}ge_3 - \bar{J}_R^\top f + \bar{J}_{\mathcal{S}}^\top f_{\mathcal{S}}). \end{aligned} \tag{D.7}$$

The transformations matrices $J_{\mathcal{B}}^\top$ and $\bar{J}_R^\top$ map the wrenches from the contact locations to the seesaw and robot center of mass, hence they are of the form: $J_{\mathcal{B}}^\top = \begin{bmatrix} {^{\mathcal{G}[\mathcal{I}]}X^*_{Lfoot}} & {^{\mathcal{G}[\mathcal{I}]}X^*_{Rfoot}} \end{bmatrix}$ and $\bar{J}_R^\top = \begin{bmatrix} {^{\mathcal{S}[\mathcal{I}]}X^*_{Lfoot}} & {^{\mathcal{S}[\mathcal{I}]}X^*_{Rfoot}} \end{bmatrix}$. Thus (D.6) can be further simplified by removing f. Furthermore, one can verify that $S(^{\mathcal{I}}p_{\mathcal{S}} - {^{\mathcal{I}}p_T})m_{\mathcal{S}}ge_3 + S(^{\mathcal{I}}p_c - {^{\mathcal{I}}p_T})mge_3 = 0$. This result is consistent with the definition of $\dot{L}_T$ as the summation of all external wrenches, in this case the contact forces $f_{\mathcal{S}}$ and the gravity wrench. By substituting $f_{\mathcal{S}}$ into Eq. (D.6) by means of (9.7), we finally obtain Eq. (9.9), where we defined:

$$\begin{aligned} J_T^\top &= {^{T}X^*_{\mathcal{S}[\mathcal{I}]}}\bar{J}_{\mathcal{S}}^\top \\ A_T &= \Gamma^{-1}J_{\mathcal{S}}M_{\mathcal{S}}^{-1}J_R^\top \\ f_{bias} &= \Gamma^{-1}(J_{\mathcal{S}}M_{\mathcal{S}}^{-1}h_{\mathcal{S}} - \dot{J}_{\mathcal{S}}\nu_{\mathcal{S}}) - (m_{\mathcal{S}} + m)ge_3. \end{aligned}$$

Appendix E

Proof of Lemma 6–7

E.1 Proof of Lemma 6

Proof of 1): when $f^k = \phi(\xi^k)$, the constraint on the positivity of the vertical force Eq. (3.8a) is given by:

$$f_z = e^{\xi_3} + f_z^{min} > 0,$$

which is satisfied for all finite ξ_3 provided that $f_z^{min} \geq 0$. We substitute the remaining stability constraints Eq. (3.8b)–(3.8e) with the parametrizations (12.4):

$$\sqrt{{\mu_c}^2 \frac{\tanh^2(\xi_1)\, f_z^2}{1+\tanh^2(\xi_2)} + {\mu_c}^2 \frac{\tanh^2(\xi_2)\, f_z^2}{1+\tanh^2(\xi_1)}} < \mu_c f_z, \tag{E.1a}$$

$$y_c^{min} < \frac{(\delta_y \tanh(\xi_4) + \delta_{y0})\, f_z}{f_z} < y_c^{max}, \tag{E.1b}$$

$$x_c^{min} < \frac{(\delta_x \tanh(\xi_5) + \delta_{x0})\, f_z}{f_z} < x_c^{max}, \tag{E.1c}$$

$$|\frac{\mu_z \tanh(\xi_6)\, f_z}{f_z}| < \mu_z. \tag{E.1d}$$

the vertical force f_z is greater than zero and can be simplified from system (E.1), leading to the following set of inequalities:

$$\sqrt{\frac{\tanh^2(\xi_1)}{1+\tanh^2(\xi_2)} + \frac{\tanh^2(\xi_2)}{1+\tanh^2(\xi_1)}} < 1, \tag{E.2a}$$

$$y_c^{min} < \delta_y \tanh(\xi_4) + \delta_{y0} < y_c^{max}, \tag{E.2b}$$

$$x_c^{min} < \delta_x \tanh(\xi_5) + \delta_{x0} < x_c^{max}, \tag{E.2c}$$

$$|\tanh(\xi_6)| < 1, \tag{E.2d}$$

where also the coefficients μ_c and μ_z have been collected and simplified from Eq. (E.1a) and (E.1d), respectively. It is now straightforward to verify that constraint (E.2d) is verified for all finite ξ_6. Direct calculations on Eq. (E.2b)–(E.2c) lead to the following expressions:

$$
\begin{aligned}
y_c^{min} &< \frac{y_c^{max}(1+\tanh(\xi_4)) + y_c^{min}(1-\tanh(\xi_4))}{2} < y_c^{max} \\
x_c^{min} &< \frac{x_c^{max}(1+\tanh(\xi_5)) + x_c^{min}(1-\tanh(\xi_5))}{2} < x_c^{max}
\end{aligned}
$$

which is always satisfied for all finite ξ_4, ξ_5. Concerning the remaining constraint Eq. (E.2a), the argument of the square root can be rearranged as follows:

$$
\frac{\tanh^2(\xi_1)(1+\tanh^2(\xi_1)) + \tanh^2(\xi_2)(1+\tanh^2(\xi_2))}{\tanh^2(\xi_1) + \tanh^2(\xi_2) + \tanh^2(\xi_1)\tanh^2(\xi_2) + 1}
$$

which is always lower than 1 for all finite ξ_1, ξ_2.
Proof of 2): assume we are given with a *feasible* wrench f^k. It is straightforward to compute the *inverse mapping* of the vertical force and moments parametrization:

$$
\begin{aligned}
\xi_3 &= \ln\left(f_z - f_z^{min}\right) \\
\xi_4 &= \text{atanh}\left(\frac{M_x - \delta_{y0} f_z}{\delta_y f_z}\right) \\
\xi_5 &= \text{atanh}\left(\frac{M_y - \delta_{x0} f_z}{\delta_x f_z}\right) \\
\xi_6 &= \text{atanh}\left(\frac{M_z}{\mu_z f_z}\right).
\end{aligned}
$$

Since the above equations are composed of one-to-one correspondences (hyperbolic tangent, logarithm), the solution $[\xi_3, \xi_4, \xi_5, \xi_6]^\top$ is unique.

For what concerns the tangential forces f_x and f_y, let us recall the expressions for the parametrization of f_x and f_y:

$$
f_x = \mu_c \frac{\tanh(\xi_1)\, f_z}{\sqrt{1+\tanh^2(\xi_2)}} \tag{E.3a}
$$

$$
f_y = \mu_c \frac{\tanh(\xi_2)\, f_z}{\sqrt{1+\tanh^2(\xi_1)}}. \tag{E.3b}
$$

An easy way to compute the inverse mapping is to raise to the square Eq. (E.3), which gives the following expressions:

$$f_x^2 = {\mu_c}^2 \frac{\tanh^2(\xi_1)\, f_z^2}{1+\tanh^2(\xi_2)} \tag{E.4a}$$

$$f_y^2 = {\mu_c}^2 \frac{\tanh^2(\xi_2)\, f_z^2}{1+\tanh^2(\xi_1)}. \tag{E.4b}$$

In the resulting equations, the square hyperbolic tangents $\tanh^2(\xi_1)$, $\tanh^2(\xi_2)$ appear linearly. Therefore they can be computed through matrix inversion:

$$\begin{bmatrix}\tanh^2(\xi_1)\\ \tanh^2(\xi_2)\end{bmatrix} = \begin{bmatrix}{\mu_c}^2 f_z^2 & -f_x^2\\ -f_y^2 & {\mu_c}^2 f_z^2\end{bmatrix}^{-1} \begin{bmatrix}f_x^2\\ f_y^2\end{bmatrix}. \tag{E.5}$$

Resolving Eq. (E.5) w.r.t. ξ_1, ξ_2 gives *two* possible solutions, namely $\xi_{1(2)} = \pm\text{atanh}(\sqrt{\tanh^2(\xi_{1(2)})})$. However, only one of the two solutions satisfies the parametrization Eq. (E.3): in fact, the terms on the right-hand side of Eq. (E.3) f_z, μ_c and the square root $\sqrt{1+\tanh^2(\xi_{1(2)})}$ are always positive. Therefore the sign of ξ_1 (ξ_2) must correspond to the sign of f_x (f_y), leading to the solution:

$$\xi_1 = \text{sign}(f_x)\text{atanh}(\sqrt{\tanh^2(\xi_1)})$$
$$\xi_2 = \text{sign}(f_y)\text{atanh}(\sqrt{\tanh^2(\xi_2)}).$$

Remark 5 *It is possible to verify that if f_x and f_y belong to $\mathcal{K}$, then the matrix inversion in Eq.* (E.5) *can always be performed. In fact, singularities arise when* $\det(\begin{bmatrix}{\mu_c}^2 f_z^2 & -f_x^2\\ -f_y^2 & {\mu_c}^2 f_z^2\end{bmatrix}) = \mu_c^4 f_z^4 - f_x^2 f_y^2 = 0$. *Substituting Eq.* (E.4) *in the expression of the determinant allows to verify that the condition $\mu_c^4 f_z^4 = f_x^2 f_y^2$ never occurs for any ξ_1, ξ_2.*

Proof of 3): let $\Phi_k \in \mathbb{R}^{6\times 6}$ denote the gradient of $f^k = \phi(\xi^k)$. Φ_k is then a matrix of the following shape:

$$\Phi_k = \begin{bmatrix} F_{11} & F_{12} & F_{13} & 0 & 0 & 0\\ F_{21} & F_{22} & F_{23} & 0 & 0 & 0\\ 0 & 0 & F_{33} & 0 & 0 & 0\\ 0 & 0 & F_{43} & F_{44} & 0 & 0\\ 0 & 0 & F_{53} & 0 & F_{55} & 0\\ 0 & 0 & F_{63} & 0 & 0 & F_{66}\end{bmatrix}.$$

Applying the Laplace's formula for the calculation of the determinant of Φ_k gives:

$$\det(\Phi_k) = F_{66}F_{55}F_{44}F_{33}\det\begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix}$$

where one has:

$$\begin{aligned}
F_{11} &= \frac{\mu_c(1-\tanh^2(\xi_1))(e^{\xi_3}+f_z^{min})}{\sqrt{1+\tanh^2(\xi_2)}}, \\
F_{12} &= \frac{\mu_c \tanh(\xi_1)(e^{\xi_3}+f_z^{min})}{(1+\tanh^2(\xi_2))^{\frac{3}{2}}}(\tanh^3(\xi_2)-\tanh(\xi_2)), \\
F_{21} &= \frac{\mu_c \tanh(\xi_2)(e^{\xi_3}+f_z^{min})}{(1+\tanh^2(\xi_1))^{\frac{3}{2}}}(\tanh^3(\xi_1)-\tanh(\xi_1)), \\
F_{22} &= \frac{\mu_c(1-\tanh^2(\xi_2))(e^{\xi_3}+f_z^{min})}{\sqrt{1+\tanh^2(\xi_1)}}, \\
F_{33} &= e^{\xi_3}, \\
F_{44} &= \delta_y(1-\tanh^2(\xi_4))(e^{\xi_3}+f_z^{min}), \\
F_{55} &= \delta_x(1-\tanh^2(\xi_5))(e^{\xi_3}+f_z^{min}), \\
F_{66} &= \mu_z(1-\tanh^2(\xi_6))(e^{\xi_3}+f_z^{min}).
\end{aligned}$$

It can be easily verified that F_{33}, F_{44}, F_{55} and F_{66} are always different from zero for any finite $\xi_3, \xi_4, \xi_5, \xi_6$. We are then left to evaluate the determinant of $\begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix}$, which is $F_{11}F_{22} - F_{12}F_{21}$. After noting that the product $F_{11}F_{22}$ is contained in the expression of $F_{12}F_{21}$, one is left with the following expression:

$$\begin{aligned}
\det\begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} &= \mu_c^2(e^{\xi_3}+f_z^{min})^2 \cdot \\
&\quad \frac{(1-\tanh^2(\xi_1))(1-\tanh^2(\xi_2))}{\sqrt{(1+\tanh^2(\xi_1))(1+\tanh^2(\xi_2))}} \cdot \\
&\quad \frac{1+\tanh^2(\xi_1)+\tanh^2(\xi_2)}{(1+\tanh^2(\xi_1))(1+\tanh^2(\xi_2))}
\end{aligned}$$

which is non-zero for any finite ξ_1, ξ_2, ξ_3. As a conseguence, matrix Φ_k is invertible for any finite ξ^k.

E.2 Proof of Lemma 7

Stability: consider the following Lyapunov function candidate:

$$V(I, \tilde{L}, \zeta) := \frac{1}{2} I^\top K_P I + \frac{1}{2} \tilde{L}^\top \tilde{L} + \frac{1}{2} \zeta^\top K_O \zeta.$$

Note that $V = 0 \iff (I, \tilde{L}, \zeta)^\top = (0, 0, 0)^\top$. Compute the Lyapunov function derivative $\dot{V}$:

$$\begin{aligned}
\dot{V} &= I^\top K_P \tilde{L} + \tilde{L}^\top \dot{\tilde{L}} + \zeta^\top K_O \dot{\zeta} \\
&= I^\top K_P \tilde{L} + \tilde{L}^\top (\zeta - K_P I - K_D \tilde{L}) + \zeta^\top K_O \dot{\zeta} \\
&= -\tilde{L}^\top K_D \tilde{L} + \zeta^\top K_o (\dot{\zeta} + K_O^{-1} \tilde{L})
\end{aligned}$$

It is clear that $\dot{V} \leq 0$ when $\dot{\zeta} + K_O^{-1} \tilde{L} = -\zeta$. We substitute $\dot{\zeta}$ with the left-hand side of Eq. (12.13c) and ζ with its definition:

$$\begin{aligned}
&\dot{A} f + A \Phi \dot{\xi} - \ddot{L}^d + K_D \dot{\tilde{L}} + K_P \tilde{L} + K_O^{-1} \tilde{L} \\
&\quad = -A f + m g e_3 + \dot{L}^d - K_D \tilde{L} - K_P I,
\end{aligned} \tag{E.6}$$

and after a rearrangement Eq. (E.6) leads to the definition of the control input $\dot{\xi}^*$ as in Eq. (12.14), which gives $\dot{V} = -\tilde{L}^\top K_D \tilde{L} - \zeta^\top K_O \zeta \leq 0$. This implies the stability of the equilibrium point and the boundedness of system's trajectories. Furthermore, as long as Eq. (E.6) holds, the closed loop dynamics is given by $\dot{\zeta} = -\zeta - K_O^{-1} \tilde{L}$, and Eq. (12.13a)–(12.13b). The system is therefore autonomous, and the convergence of $\tilde{L}, \zeta$ and $\dot{\tilde{L}}, \dot{\zeta}$ to zero can be proved by resorting to the LaSalle's theorem. Convergence to zero of I can be proven by computing Eq. (12.13b) at steady state.

Note that the possibility of achieving $\dot{\xi} = \dot{\xi}^*$ and, by consequence, the soundness of the stability analysis, depends on the rank of matrix Φ. In Appendix E.1, it has been proved that the matrix Φ is always invertible for any finite ξ. This consideration leads to the conclusion that if ξ always remains locally (or globally) bounded, the equilibrium point $(I, \tilde{L}, \zeta)^\top = (0, 0, 0)^\top$ can be proven to be locally (globally) asymptotically stable.

Computation of $\dot{\xi}_0$: we rewrite Eq. (12.16) as:

$$\tau^* = \Theta f^* + \theta \tag{E.7a}$$

$$\Theta := -\Lambda^\dagger J M^{-1} J^\top \tag{E.7b}$$

$$\theta := \Lambda^\dagger [J M^{-1} h - \dot{J} \nu] + N_\Lambda \tau_0. \tag{E.7c}$$

Then, consider the following Lyapunov function:

$$\begin{aligned} V &= \frac{1}{2}\tau^{*\top}\tau^{*} \\ \dot{V} &= \tau^{*\top}\dot{\tau}^{*} = \tau^{*\top}(\dot{\Theta}f + \Theta\Phi\dot{\xi}^{*} + \dot{\theta}), \end{aligned}$$

and substitute the expression of $\dot{\xi}^{*}$ with the right-hand side of Eq. (12.14) into the Lyapunov function derivative $\dot{V}$, which leads to:

$$\dot{V} = \tau^{*\top}(\dot{\Theta}f + \Theta\Phi\dot{\xi}_1^{*} + \Theta\Phi N_{A\Phi}\dot{\xi}_0^{*} + \dot{\theta}), \tag{E.8}$$

where $\dot{\xi}_1 = (A\Phi)^{\dagger}\,[\ddot{L}^d - (K_D + 1_6)\dot{\tilde{L}} - (K_D + K_O^{-1} + K_P)\tilde{L} - K_P I - \dot{A}f]$. A solution for minimizing the norm of joint torques is to impose:

$$\dot{\Theta}f + \Theta\Phi\dot{\xi}_1^{*} + \Theta\Phi N_{A\Phi}\dot{\xi}_0^{*} + \dot{\theta} = -K_\tau\tau^{*}, \tag{E.9}$$

with K_τ a symmetric and positive definite matrix. When the equivalence (E.9) is satisfied, the Lyapunov derivative Eq. (E.8) becomes $\dot{V} = -\tau^{*\top}K_\tau\tau^{*} \leq 0$ and the input joint torques converge to zero. However, this is not the case as the rank of the matrix $(\Theta\Phi N_{A\Phi})$ that multiplies the free variable $\dot{\xi}_0$ is lower than the dimension of the joint torques vector $\tau^{*} \in \mathbb{R}^n$. Nevertheless, we compute the closest solution to Eq. (E.9), that leads to the following expression of $\dot{\xi}_0^{*}$:

$$\dot{\xi}_0^{*} = -(\Theta\Phi N_{A\Phi})^{\dagger}(\dot{\Theta}f + \Theta\Phi\dot{\xi}_1^{*} + \dot{\theta} + K_\tau\tau^{*}).$$

Appendix F

iRonCub Flight Control

F.1 Proof of Lemma 8

For the sake of clarity we omit the subscripts and superscripts and we define $R = {}^{\mathcal{I}}R_{\mathcal{B}}$, $\tilde{L}_\omega = {}^{\mathcal{G}[\mathcal{B}]}\tilde{L}_\omega$, $\omega = {}^{\mathcal{B}}\omega_{\mathcal{B}}$, $I_0^{-1} = {}^{\mathcal{G}[\mathcal{B}]}I_0^{-1}$. Then, consider the following Lyapunov function candidate:

$$\begin{aligned} V &= V_1 + V_2 + V_3, \\ V_1 &= \frac{c_0 + c_1}{2}\,\mathrm{tr}(1_3 - R^{d\top}R), \\ V_2 &= \frac{c_0 + c_1}{2}\,\tilde{L}_\omega^\top I_0^{-1}\tilde{L}_\omega, \\ V_3 &= \frac{1}{2}|\dot{\tilde{L}}_\omega + \tilde{L}_\omega + skv|^2. \end{aligned} \tag{F.1}$$

It is possible to verify that (F.1) is a valid Lyapunov function candidate. The term V_1 is always positive, and it is zero iif $R = R^d$ (see also [Olfati-Saber, 2001, Sec 5.11.6]). Recall I_0^{-1} is symmetric and positive definite. Then, V_2 is always positive and $V_2 = 0$ iif $\tilde{L}_\omega = 0$. The last term V_3 is always positive and it has several solutions such that $V_3 = 0$, but the only one that guarantees also V_1 and V_2 to be zero is $R = R^d,\ \tilde{L}_\omega = 0,\ \dot{\tilde{L}}_\omega = 0$ (in particular, recall that $skv(R = R^d) = 0$). Therefore one has $V \geq 0$, $V = 0$ iif $R = R^d,\ \tilde{L}_\omega = 0,\ \dot{\tilde{L}}_\omega = 0$. The time derivative of V is given by:

$$\begin{aligned} \dot{V} &= \dot{V}_1 + \dot{V}_2 + \dot{V}_3, \\ \dot{V}_1 &= (c_0 + c_1)\,\tilde{L}_\omega^\top I_0^{-1} skv, \\ \dot{V}_2 &= (c_0 + c_1)\,\tilde{L}_\omega^\top I_0^{-1}\dot{\tilde{L}}_\omega, \\ \dot{V}_3 &= (\dot{\tilde{L}}_\omega + \tilde{L}_\omega + skv)^\top(\ddot{\tilde{L}}_\omega + \dot{\tilde{L}}_\omega + s\dot{k}v). \end{aligned} \tag{F.2}$$

The derivatives $\dot{V}_2$ and $\dot{V}_3$ are straightforward to compute, while the derivative $\dot{V}_1$ can be obtained by recalling that $\frac{d(\mathrm{tr}(1_3 - R^{d\top} R))}{dt} = 2\tilde{\omega}^\top skv(R, R^d)$ with $\tilde{\omega} = \omega - \omega^d$ (see also [Olfati-Saber, 2001, Sec 5.11.6]). Also, if Eq. (14.6) holds, then $\tilde{\omega} = I_0^{-1}\tilde{L}_\omega$, which in turns leads to (F.2). Direct substitution of $\ddot{L}_\omega$ from Eq. (14.8) into $\dot{V}$ gives, after long but straightforward calculations:

$$\begin{aligned} \dot{V} = & -c_1 \tilde{L}_\omega^\top I_0^{-1} \tilde{L}_\omega + \\ & -c_0(\dot{\tilde{L}}_\omega^\top I_0^{-1} \dot{\tilde{L}}_\omega + skv^\top I_0^{-1} skv + 2\dot{\tilde{L}}_\omega^\top I_0^{-1} skv) \\ = & -c_1 \tilde{L}_\omega^\top I_0^{-1} \tilde{L}_\omega - c_0(\dot{\tilde{L}}_\omega + skv)^\top I_0^{-1}(\dot{\tilde{L}}_\omega + skv) \end{aligned} \tag{F.3}$$

Being I_0^{-1} symmetric and positive definite, one has $\dot{V} \leq 0$. In particular:

1. $\dot{V} \leq 0$ implies that $\tilde{L}$ and $\dot{\tilde{L}}$ are bounded, being V a non-increasing function (R is bounded by definition);

2. $\ddot{V}$ is bounded because of $1)$ and because of the choice of $\ddot{\tilde{L}}$ as in (14.8);

3. following the Barbalat's Lemma, $\ddot{V}$ bounded implies that $\dot{V} \to 0$;

4. $\dot{V} \to 0$ implies $\tilde{L} \to 0$ and $(\dot{\tilde{L}} + skv) \to 0$;

5. being $\ddot{\tilde{L}}$ bounded, then $\dot{\tilde{L}} \to 0$;

6. finally, because of $4)$ and $5)$ one has that also $skv \to 0$ and this implies the local convergence of $R \to R^d$ as detailed in [Olfati-Saber, 2001].

F.2 Computation of Λ_s and $\Lambda_\mathcal{B}$

The momentum acceleration is calculated as:

$$\ddot{L} = A_t \dot{\mathrm{w}}_t + A_c \dot{f} + \dot{A}_t \mathrm{w}_t + \dot{A}_c f, \tag{F.4}$$

with the time derivatives $\dot{A}_t$ and $\dot{A}_c$ given by:

$$\begin{aligned} \dot{A}_t \quad &= \left[\dot{\bar{S}}(\dot{r}_1)^\mathcal{I} l_1, \dots, \dot{\bar{S}}(\dot{r}_{n_p})^\mathcal{I} l_{n_p}\right] + \left[\bar{S}(r_1)^\mathcal{I} \dot{l}_1, \dots, \bar{S}(r_{n_p})^\mathcal{I} \dot{l}_{n_p}\right], \\ \dot{A}_c \quad &= \left[\dot{\bar{S}}(\dot{r}_L), \quad \begin{bmatrix} 0_3 \\ 0_3 \end{bmatrix}, \quad \dot{\bar{S}}(\dot{r}_R), \quad \begin{bmatrix} 0_3 \\ 0_3 \end{bmatrix}\right], \\ \dot{\bar{S}}(\dot{r}_i) :=& \begin{bmatrix} 0_3 \\ S(\dot{r}_i) \end{bmatrix}, \; i = \{1, \dots n_p; \; R; \; L\}. \end{aligned}$$

Recall that the i-th thrust reference frame $\mathcal{T}_i$ is chosen with its z axis pointing in the direction of the thrust. Then, the thrust direction is the last column of the rotation matrix ${}^{\mathcal{I}}R_{\mathcal{T}_i}$, i.e. ${}^{\mathcal{I}}l_i = {}^{\mathcal{I}}R_{\mathcal{T}_i}e_3$. Its derivative is computed as: ${}^{\mathcal{I}}\dot{l}_i = {}^{\mathcal{I}}\dot{R}_{\mathcal{T}_i}e_3 = S(\omega_{t_i}){}^{\mathcal{I}}l_i$, with ω_{t_i} the angular velocity of the i-th thrust reference frame. Then, apply the property $S(y)x = -S(x)y$ to the products $\dot{\bar{S}}(\dot{r}_i){}^{\mathcal{I}}l_i$ and $\bar{S}(r_i){}^{\mathcal{I}}\dot{l}_i$:

$$\begin{aligned}
\dot{A}_t &= \left[\dot{\bar{S}}(\dot{r}_1){}^{\mathcal{I}}l_1,\ \dots,\dot{\bar{S}}(\dot{r}_{n_p}){}^{\mathcal{I}}l_{n_p}\right] + \left[\bar{S}(r_1){}^{\mathcal{I}}\dot{l}_1,\ \dots,\bar{S}(r_{n_p}){}^{\mathcal{I}}\dot{l}_{n_p}\right] = \\
&= -\left[\tilde{S}_1\begin{bmatrix}\dot{r}_1\\ \omega_1\end{bmatrix},\ \dots,\tilde{S}_{n_p}\begin{bmatrix}\dot{r}_{n_p}\\ \omega_{n_p}\end{bmatrix}\right], \\
\tilde{S}_i &:= \begin{pmatrix} 0_3 & S({}^{\mathcal{I}}l_i) \\ S({}^{\mathcal{I}}l_i) & S(r_i)S({}^{\mathcal{I}}l_i)\end{pmatrix}.
\end{aligned}$$

Define the Jacobian J_T mapping the system velocity ν into the velocities $\Omega := \left[\dot{r}_1^\top,\omega_1^\top,\ \dots,\ \dot{r}_{n_p}^\top,\omega_{n_p}^\top\right]^\top \in \mathbb{R}^{6n_p}$. Then, rearranging the term $\dot{A}_t\mathrm{w}_t$ that appears in the momentum acceleration equations (F.4) gives:

$$\dot{A}_t\mathrm{w}_t = -\left[\mathrm{w}_{t_1}\tilde{S}_1\ \dots,\mathrm{w}_{t_{n_p}}\tilde{S}_{n_p}\right]J_T\nu = \Lambda_t\nu.$$

Analogously, we apply the property $S(y)x = -S(x)y$ to the term $\dot{A}_cf$:

$$\dot{A}_cf = -\begin{bmatrix} 0_3 & 0_3 \\ S(f_L^l) & S(f_R^l)\end{bmatrix}J_f\nu = \Lambda_f\nu.$$

where $f_L^l, f_R^l \in \mathbb{R}^3$ are the feet forces and the Jacobian J_f maps the system velocity ν into the velocities $\left[\dot{r}_L,\ \dot{r}_R\right]$. From Λ_t and Λ_f we can finally obtain the expression of Λ_s and $\Lambda_{\mathcal{B}}$ as they appear in Eq. (14.9):

$$\begin{aligned}
\Lambda &:= \Lambda_t + \alpha\Lambda_f, \\
\Lambda_{\mathcal{B}} &:= \Lambda\begin{pmatrix} 1_6 \\ 0_{n\times 6}\end{pmatrix}, \\
\Lambda_s &:= \Lambda\begin{pmatrix} 0_{6\times n} \\ 1_n\end{pmatrix}.
\end{aligned}$$

F.3 QP Input Boundaries

Assume we are trying to solve the following optimization problem:

$$\dot{x}(t)^* = \underset{\dot{x}}{\operatorname{argmin}}\, |A\dot{x} + b|^2 \tag{F.5a}$$

$$s.t.$$

$$l_b^{\dot{x}} \le \dot{x} \le u_b^{\dot{x}}, \tag{F.5b}$$

where $l_b^{\dot{x}}$ and $u_b^{\dot{x}}$ are the lower and upper bounds of the variable $\dot{x}$. Furthermore, the integral given by $x(t) = \int_0^t \dot{x}(t)dt$ is also bounded, and the boundaries are:

$$l_b^x \leq x(t) \leq u_b^x$$

We would like to include the boundaries of x inside the QP problem (F.5), so that both the limits on $\dot{x}$ and x are respected. We can state our requirements as follows:

- if $l_b^x \leq x(t) \leq u_b^x$, the current value of x is respecting its limits. Therefore its derivative $\dot{x}$ can be any value, provided that it respects the derivative bounds $l_b^{\dot{x}} \leq \dot{x}(t) \leq u_b^{\dot{x}}$;
- if $x(t) = l_b^x$, the variable x reached the lower bound. The derivative $\dot{x}$ can be either 0 or positive, but it cannot be negative (thus asking x to further decrease). The bounds of $\dot{x}$ need to be modified as $0 \leq \dot{x}(t) \leq u_b^{\dot{x}}$;
- the same approach can be used when $x(t) = u_b^x$ by setting $l_b^{\dot{x}} \leq \dot{x}(t) \leq 0$;

Furthermore, assume that x instantly overcomes the limits. This may occur on a real application because of unmodeled phenomena or disturbances that bring the variable x outside the limits. In this case, we must avoid $\dot{x}$ to keep increasing (or decreasing) thus bringing x far from the upper (lower) limit. Instead, $\dot{x}$ must be strictly negative (positive) when x overcomes the upper (lower) limit, in order to force x to decrease (increase) and bringing it back in between the boundaries. All these requirements can be formulated analytically thanks to the properties of the hyperbolic tangent:

$$\tanh(\epsilon_b(x - l_b^x))l_b^{\dot{x}} \leq \dot{x}(t) \leq \tanh(\epsilon_u(u_b^x - x))u_b^{\dot{x}},$$

where ϵ_b, ϵ_u are positive scalars defining the sharpness of the hyperbolic tangent.