Generation of the whole-body motion for humanoid robots with the complete dynamics

Oscar Efrain Ramos Ponce

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *09 Octobre 2014* **par :**

Óscar Efraín RAMOS PONCE

Generation of Whole-body Motion for Humanoid Robots with the Complete Dynamics

---

JURY

Christine CHEVALLEREAU	Directeur de Recherche	Rapporteur
Francesco NORI	Researcher	Rapporteur
Patrick DANÈS	Professeur des Universités	Examinateur
Ludovic RIGHETTI	Researcher	Examinateur
Nicolas MANSARD	Chargé de Recherche	Directeur de Thèse
Philippe SOUÈRES	Directeur de Recherche	Directeur de Thèse
Yuval TASSA	Researcher	Invité

École doctorale et spécialité :
EDSYS : Robotique 4200046
Unité de Recherche :
LAAS - CNRS (UPR 8001)
Directeur(s) de Thèse :
Philippe SOUÈRES et *Nicolas MANSARD*
Rapporteurs :
Francesco NORI et *Christine CHEVALLEREAU*

Christine CHEVALLEREAU: Directeur de Recherche, École Centrale de Nantes, France

Francesco NORI: Researcher, Italian Institute of Technology, Italy

Patrick DANÈS: Professeur des Universités, Université de Toulouse III, France

Ludovic RIGHETTI: Researcher, Max-Plank-Institute for Intelligent Systems, Germany

Nicolas MANSARD: Chargé de Recherche, LAAS-CNRS, France

Philippe SOUÈRES: Directeur de recherche, LAAS-CNRS, France

Yuval TASSA: Researcher, University of Washington, USA

Abstract

This thesis aims at providing a solution to the problem of motion generation for humanoid robots. The proposed framework generates whole-body motion using the complete robot dynamics in the task space satisfying contact constraints. This approach is known as operational-space inverse-dynamics control. The specification of the movements is done through objectives in the task space, and the high redundancy of the system is handled with a prioritized stack of tasks where lower priority tasks are only achieved if they do not interfere with higher priority ones. To this end, a hierarchical quadratic program is used, with the advantage of being able to specify tasks as equalities or inequalities at any level of the hierarchy. Motions where the robot sits down in an armchair and climbs a ladder show the capability to handle multiple non-coplanar contacts.

The generic motion generation framework is then applied to some case studies using HRP-2 and Romeo. Complex and human-like movements are achieved using human motion imitation where the acquired motion passes through a kinematic and then dynamic retargeting processes. To deal with the instantaneous nature of inverse dynamics, a walking pattern generator is used as an input for the stack of tasks which makes a local correction of the feet position based on the contact points allowing to walk on non-planar surfaces. Visual feedback is also introduced to aid in the walking process. Alternatively, for a fast balance recovery, the capture point is introduced in the framework as a task and it is controlled within a desired region of space. Also, motion generation is presented for CHIMP which is a robot that needs a particular treatment.

Résumé

Cette thèse propose une solution au problème de la génération de mouvements pour les robots humanoïdes. Le cadre qui est proposé dans cette thèse génère des mouvements corps-complet en utilisant la dynamique inverse avec l'espace des tâches et en satisfaisant toutes les contraintes de contact. La spécification des mouvements se fait à travers objectifs dans l'espace des tâches et la grande redondance du système est gérée avec une pile de tâches où les tâches moins prioritaires sont atteintes seulement si elles n'interfèrent pas avec celles de plus haute priorité. À cette fin, un QP hiérarchique est utilisé, avec l'avantage d'être en mesure de préciser tâches d'égalité ou d'inégalité à tous les niveaux de la hiérarchie. La capacité de traiter plusieurs contacts non-coplanaires est montrée par des mouvements où le robot s'assoit sur une chaise et monte une échelle.

Le cadre générique de génération de mouvements est ensuite appliqué à des études de cas à l'aide de HRP-2 et Romeo. Les mouvements complexes et similaires à l'humain sont obtenus en utilisant l'imitation du mouvement humain où le mouvement acquis passe par un processus cinématique et dynamique. Pour faire face à la nature instantanée de la dynamique inverse, un générateur de cycle de marche est utilisé comme entrée pour la pile de tâches qui effectue une correction locale de la position des pieds sur la base des points de contact permettant de marcher sur un terrain accidenté. La vision stéréo est également introduite pour aider dans le processus de marche. Pour une récupération rapide d'équilibre, le capture point est utilisé comme une tâche contrôlée dans une région désirée de l'espace. En outre, la génération de mouvements est présentée pour CHIMP, qui a besoin d'un traitement particulier.

Acknowledgements

In the first place, I would like to thank my Ph.D. advisors Nicolas Mansard and Philippe Souères for their guidance during my thesis work. Their experience, knowledge and continuous advices have allowed me to become a better researcher and have ultimately made this thesis possible. I am specially grateful to Nicolas, who has always supported me on many scientific and technical topics since I arrived to LAAS-CNRS, and to whom I owe most of the achievements of my work.

Thanks to Francesco Nori and Christine Chevallereau for accepting to examine this thesis and to be part of the jury. Their constructive feedback and their close insights have been valuable for the improvement of the manuscript. Thanks to Ludovic Righetti and Patrick Danès for accepting to be members of the jury, for their interest in my work and for their feedback on the thesis.

Many thanks to Siddhartha Srinivasa for receiving me during my stay at Carnegie Mellon University, in Pittsburgh. Special thanks to Clark Haynes, who gave me many insights, advice and support on the work with CHIMP. Thanks to Christopher Dellin, Kyle Strabala, José Gonzalez Mora, Mike Vande-Weghe, David Stager, and all the other members of the Tartan Rescue Team with whom I had the pleasure to work. Thanks for the great experience at the DARPA Robotics Challenge (DRC) Trials in Miami. I would also like to thank Abderrahmane Kheddar for receiving me at the French-Japanese Joint Robotics Lab (JRL) during my short stay at AIST in Tsukuba, Japan.

I sincerely thank the great researchers in the Gepetto team at LAAS-CNRS who, in one or other way, have contributed to my achievements. Thanks to Jean-Paul Laumond, the former head of the team who gave me the opportunity to be part of it. Thanks to Florent Lamiraux, who gave me support with the software framework. Special thanks to Olivier Stasse, with whom I had the pleasure to work for the Novela project, and whose vast experience with the robot has made possible the experiments with HRP-2.

I would also like to thank the former and current non-permanent members of the Gepetto Team. Many thanks to Layale Saab, who started the work on inverse dynamics control, to Sovannara Hak, whose expertise on motion capture was fundamental for motion imitation, to Mauricio García, whose work on stereo vision was useful for one part of this thesis, and to Andrea del Prete, with whom I have been working by the end of the thesis towards the real-time implementation of the stack of tasks. Thanks to the people in my last office,

namely Naoko Abe, Mukunda Bharatheesha, and Joseph Mirabel, who provided a suitable and friendly environment as well as interesting discussions. Thanks to Valentin Aurousseau, Renliw Fleurmond, Antonio El-Khoury, Kai Henning Koch, Olivier Roussel, Andreas Orthey, Justin Carpentier, Ganesh Kumar, Maximilien Naveau, Nirmal Giftsun, Mehdi Benallegue, Ixchel Ramirez, Galo Maldonado, and the other members of the team for providing a very friendly environment in the laboratory.

Finally, my deepest thanks go out to my parents for their constant support, advice, love and help not only during the development of this thesis, but throughout all of my life. Thanks for always being there when I most needed it, in spite of the distance; thanks for all the encouragements at all times; thanks for all the beliefs I was always taught. And, overall, thanks God for the life and for always giving me the strength to keep forward despite all the difficulties.

Contents

List of Figures

Notation Table

This is a review of the abbreviations used throughout this thesis.

Term	Meaning
CoM	Center of Mass
CoP	Center of Pressure
CP	Capture Point
DoF	Degree(s) of Freedom
HQP	Hierarchical Quadratic Program
IK	Inverse Kinematics
LIP	Linear Inverted Pendulum
MPC	Model-Based Predictive Control
OSID	Operational-Space Inverse-Dynamics
PD	Proportional-Derivative
QP	Quadratic Program
$SE(3)$	Special Euclidean Group
$se(3)$	Lie Algebra of SE(3)[1]
$se^*(3)$	Dual Space of $se(3)$
$SO(3)$	Special Orthogonal Group
SoT	Stack of Tasks
WPG	Walking Pattern Generator
ZMP	Zero-Moment Point

[1] assimilated to the set of screws.

Chapter 1

Introduction

The human body is the most astonishing natural machine known by humanity, and for many years human beings have been trying to artificially re-create the complex mechanisms that constitute it. The task is extremely complicated and it has, in a certain way, led to the creation of human-like machines that should be able to behave like humans and work in environments adapted to humans. However, the current performance of these robots, usually referred to as *humanoids*, is not yet satisfactory and they are far away to be as autonomous and independent as presented in science fiction. Although humanoid robotics has lately emerged as a research area with potentially huge applications (a humanoid robot can in theory do everything that a human being can do), these robots present challenging problems in control that need to be solved using methods that differ from classical control methods since coordination and balance are always required. This thesis presents a framework based on inverse dynamics to control and generate whole-body complex motions for humanoid robots.

1.1 Problem Statement

The design and development of human-like robots has been one of the main topics in advanced robotics research during the last years. There is a tendency to change from industrial automation systems to human friendly robotic systems. These anthropomorphic robots called humanoids are expected to be able to assist in human daily environments like houses or offices. Humanoid robots are expected to behave like humans because of their friendly design, legged locomotion and anthropomorphism that helps for proper interaction within human environments. It is unquestionable that one of the main advantages of legged robots is the ability of accessing places where wheeled robots are unsuitable, like, for instance, going up stairs. Besides that, humanoid robots would not be limited to specific operations, but they would be able to satisfy a wide variety of tasks in a sophisticated way moving in the environment designed by humans for humans.

Nevertheless, there are many challenges to overcome and right now there is no humanoid robot and control system that can operate as effective as a human in its diversity of poten-

tialities. Humanoid Robotics is full of unresolved challenges and the research addresses the study of stability and mobility of the robot under different environmental conditions, complex control systems to coordinate the whole body motion, as well as the development of fast intelligent sensors and light energy-saving actuators. Artificial intelligence and computer vision are also necessary for autonomy, locomotion and human-robot interaction. Even though the benefits of nowadays fundamental research in humanoids might not seem to be profitable, it constitutes the basis to solve these enormous challenges and, in the future, it will let Humanoid Robots exist in some of the ways that currently science fiction presents them.

The objective of this thesis is to address the problem of humanoid robots control considering not only the arms or the legs separately but as a whole that constitutes the robot. Although different methods for whole-body generation exist, they are mainly based on inverse kinematics, and need additional steps to ensure the motion feasibility and are not able to reproduce fast motions that can be more natural. Whole-body motion generation is tackled in this thesis using an inverse-dynamics approach so that the motion generated can be directly reproduced in a humanoid platform. The task-function approach is used to specify the motion objectives since it allows a more compact and reasonable representation of the targets to be achieved. The methodology is independent of the robot, but it has been mainly applied to the HRP-2 robot.

1.2 Chapter Organization

The chapters of this thesis are organized as follows. Chapter 2 introduces the generalities of humanoid robotics, as well as their broad applications, and presents the state of the art in the control of this type of robots. The main classes of approaches (motion planning, inverse kinematics, inverse dynamics and optimal control) are discussed. Chapter 3 presents the foundations of the operational-space inverse-dynamics control methodology that is proposed for whole-body motion generation of anthropomorphic robots. First the task-function approach needed for motion generation is discussed, followed by the introduction of the rigid contact constraints as well as the dynamic model of the humanoid robot. Then, the control methodology for the robot is presented followed by the different tasks that are used. Several case studies are presented in Chapter 4 both in the real HRP-2 robot as well as the simulated robot. These case studies include the robot sitting in an armchair, walking on a rough terrain, and dancing. Chapter 5 points out the conclusions of this work and some possible future work.

1.3 Publications

The different work realized during the development of the present thesis led to the following publications:

Journals

- L. Saab, O. Ramos, F. Keith, N. Mansard, P. Souères, J-Y. Fourquet: *Dynamic Whole-Body Motion Generation under Rigid Contacts and other Unilateral Constraints*, IEEE Transactions on Robotics (T-RO), Vol.29 N.2, pag. 346 - 362, April 2013.

- O. Ramos, N. Mansard, O. Stasse, P. Souères: *An Advanced Robotics Motion Generation Framework for Inferring the Organization of Human Movements*, Computer Methods in Biomechanics and Biomedical Engineering (CMBEE), Vol.16 N.1, September 2013.

- O. Ramos, M. Garcia, N. Mansard, O. Stasse, J-B Hayet, P. Souères: *Towards Reactive Vision-guided Walking on Rough Terrain: An Inverse-Dynamics Based Approach*, International Journal on Humanoid Robotics (IJHR), Vol.11 N.2, July 2014.

- T. Stentz, H. Herman, A. Kelly, E. Meyhofer, G.C. Haynes, D. Stager, B. Zajac, J.A. Bagnell, J. Brindza, C. Dellin, M. George, J. Gonzalez-Mora, S. Hyde, M. Jones, M. Laverne, M. Likhachev, L. Lister, M. Powers, O. Ramos, J. Ray, D. Rice, J. Scheifflee, R. Sidki, S. Srinivasa, K. Strabala, J-P Tardif, J-S Valois, J.M. Vande-Weghe, M. Wagner, C. Wellington: *CHIMP, the CMU Highly Intelligent Mobile Platform*, Journal of Field Robotics, 2015 (in press).

- O. Ramos, N. Mansard, O. Stasse, S. Hak, L. Saab, and C. Benazeth: *Dynamic Whole Body Motion Generation for the Dance of a Humanoid Robot*, IEEE Robotics and Automation Magazine (RAM), 2015 (in press).

Conferences

- L. Saab, O. Ramos, N. Mansard, P. Souères, J-Y. Fourquet: *Generic Dynamic Motion Generation with Multiple Unilateral Constraints*, IEEE International Conference on Intelligent Robots and Systems (IROS), San Francisco, CA, USA, September 2011.

- O. Ramos, L. Saab, S. Hak, N. Mansard: *Dynamic Motion Capture and Edition using a Stack of Tasks*, IEEE-RAS International Conference on Humanoid Robots (Humanoids), Bled, Slovenia, October 2011.

- S. Hak, N. Mansard, O. Ramos, L. Saab, O. Stasse: *Capture, Recognition and Imitation of Anthropomorphic Motion*, International Conference on Robotics and Automation (ICRA), Video Session, St. Paul, MN, USA, May 2012.

- O. Ramos, N. Mansard, O. Stasse, P. Souères: *Walking on Non-planar Surfaces using an Inverse Dynamic Stack of Tasks*, IEEE-RAS International Conference on Humanoid Robots (Humanoids), Osaka, Japan, November 2012.

- O. Ramos, N. Mansard, P. Souères: *Whole-body Motion Integrating the Capture Point in the Operational Space Inverse Dynamics Control*, IEEE-RAS International Conference on Humanoid Robots (Humanoids), Madrid, Spain, November 2014.

Chapter 2

State of the Art

Human interest in building and controlling complex human-like machines is not recent; it started very early in humanity. Ancient Chinese legends mention a craftsman called *Yanshi* who presented a robot-like creation that looked and moved like a human [Tzu 11], and the Jewish folklore talks about *Golem*, an artificial being endowed with life [Idel 90]. Also in ancient Greece, *Hephaestus*, a part of Greek mythology, was believed to be a highly talented artisan who built not only remarkable tools, but also wheelchairs that moved autonomously and even gold servants that helped people [Laumond 12]. His creations are commonly associated with nowadays robots [Paipetis 10]. However, the term *robot* was introduced much later, in 1921, by Karel Capek in his play "*Rossum's Universal Robots*" [Capek 20] as a reference to human-like machines that served human beings. Since then, science fiction has associated the word with machines resembling humans and able to effortlessly develop in human environments. But despite the desire of having human-like machines, some time had to pass until the progress in computation power and manufacturing technology allowed for the introduction of the first industrial robot, the *Unimate*, to the assembly line at a General Motors plant in 1961. Five years later, the first mobile robot called *Shakey* [Nilsson 84] was introduced as a research platform for artificial intelligence algorithms. In fact, the first developments in robotics considered only industrial manipulators and wheeled platforms in very well structured environments. But nowadays, advances in robotics research and technology are making possible that some one-time science fiction human-like machines, usually referred to as *humanoid robots* or simply *humanoids*, become reality, although not yet as powerful as desired. This chapter presents a brief introduction to this type of robots, and most importantly, the different strategies used for their control.

2.1 Introduction

Since the development of Unimate and Shakey, robotics has experienced a fast and sustained increase and diversification, both in industry and research. Not only have robotic arms and mobile robots improved, but also more complex bio-inspired machines have emerged. While

50 years ago robots could only be found in industries and specialized research laboratories, in recent years they have managed to enter the life of many people, specially through service robotics. And humanoid robots are envisioned as the ideal, but probably most complex, service robot due to their characteristics that can make them suitable for human environments.

2.1.1 Service Robotics

According to the *International Federation of Robotics* (IFR), a service robot is defined as a "robot that performs useful tasks for humans or equipment excluding industrial automation applications"[1]. These robots are mainly dedicated to serve as companions for people with reduced capabilities and to assist humans in jobs that are dirty, dull, distant, dangerous or repetitive. Moreover, the IFR divides service robots in two classes:

1. *Personal service robots.* They are used for a non-commercial task and are usually operated by a non-trained person. Examples are servant robots, automated wheelchairs [Coleman 00], personal mobility assist robots (e.g. PAMM [Spenko 06]), home cleaning robots (e.g. iRobot's Roomba [Jones 06]), domestic entertainment robots (e.g. the robotic dog AIBO [Hornby 00]), social interaction robots (e.g. Paro [Shibata 01], Pepper), and pet caring robots [Kim 09].

2. *Professional service robots.* They are used for a commercial task and are usually operated by a well-trained operator. Examples are delivery robots in offices or hospitals [Niechwiadowicz 08], fire-fighting robots [AlAzemi 13], guidance robots (e.g. REEM [Marchionni 13]), exoskeletons (e.g. BLEEX [Zoss 06]), assisted therapy and rehabilitation robots [Dellon 07], and high precision surgery robots [Hockstein 07].

Most of the previously mentioned service robots are application dependent: they have been designed to satisfy specific needs. Some of them are shown in Fig. 2.1.

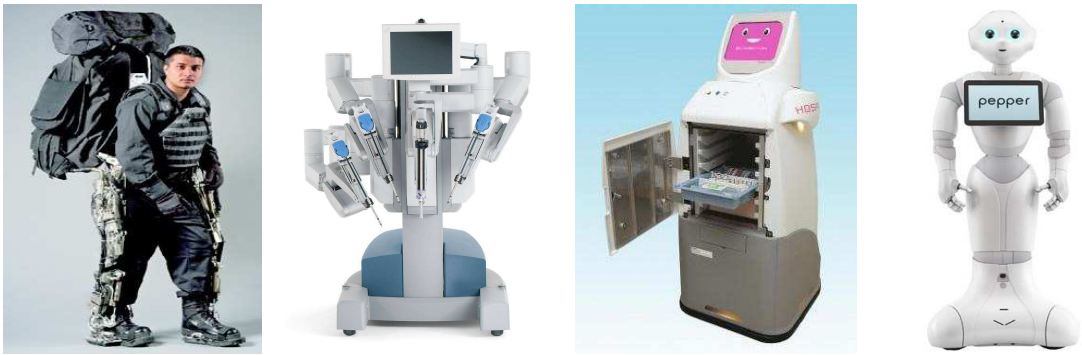
One of the main goals of service robotics is to make robots able to work and coexist side by side with humans in order to assist them. However, typical human environments are not robot-friendly; that is, they are structured for people and not for robots. For instance, human environments present doors with a great heterogeneity, various forms of windows or glasses, obstacles and furniture almost everywhere, floors made of multiple materials and not necessarily homogeneous, stairs and elevators to move from one floor to another, etc. [Soroka 12]. Hence, for robots to naturally interact with people, they must overcome all these difficulties with some degree of autonomy. Therefore, one of the biggest challenges of service robotics is to build general purpose robots that can succeed in all the areas where human beings can. Equivalently, the challenge is to build robots that act and behave as humans.

2.1.2 Why Humanoid Robots?

Perhaps the most natural response to the generic needs of service robotics is the development of robots with a human-like structure. This structure would allow them to perform well in human environments: they would be able to navigate anywhere inside a house, to go up and down by the stairs, to walk on the street, to use human tools and machines, or to assist people in their daily life. But not only does anthropomorphism provide advantages for locomotion,

[1] ISO 8373:2012 *Robots and robotic devices - Vocabulary*

(a) Roomba 880 **(b)** Sony's AIBO **(c)** Paro **(d)** PAAM



(e) Bleex exoskeleton **(f)** Surgery robot **(g)** Panasonic's Hospi **(h)** Pepper

Figure 2.1 – Examples of service robots

but it also helps in human robot interaction. An anthropomorphic design with human social characteristics is believed to increase the acceptance of robots for people [Fink 12], and its role is to take advantage of the mechanism to facilitate social interaction since people tend to attribute human characteristics to objects, perceiving them even as creatures [Oberman 07], although the *uncanny valley*[2] effect needs to be taken into account for some designs. This interaction could benefit the aging population as well as people with some type of disabilities, and it could also trigger the interest of students in science. In a professional use, operators can more intuitively learn the humanoid robot way of working since they possess an structure resembling the human structure.

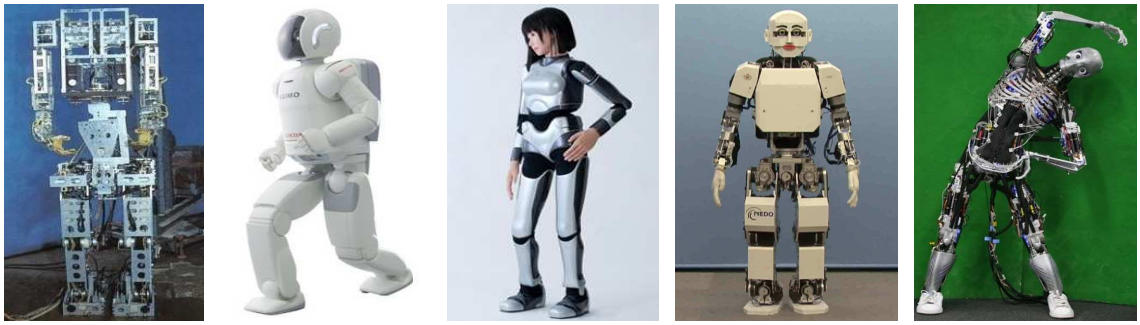
In some aspects, humanoid robots can even overperform human capabilities since they do not feel fatigue, sleepiness, boredom, or emotional distractions, and as machines, they can be doted with astonishing precision for different tasks. This could be an advantage in safety, where humanoids could work with policemen specially in risky situations. Moreover, disastrous scenarios need robots with the capability of acting in dangerous places without risking human life. For instance, the Fukushima nuclear power plant crisis in 2011 showed the need for robots in hazardous environments (namely, radioactive places) that are robust enough to overcome the terrain difficulties and to manipulate machinery designed for human manipulation. A humanoid is one of the most suitable robots for these situations. Even in the search for survivors after accidents like building collapses or fires where the conditions are dangerous or unsuitable, humanoids can be of vital help.

[2]The "uncanny valley" describes the revulsion that some people experience towards things that look and move almost, but not exactly, like natural creatures.

In general, humanoid robots emerge like a panacea for service robotics. But although the potential advantages greatly surpass the current capabilities, they provide a rationale to increase the research in humanoid robotics and their applications.

2.1.3 History of Humanoid Robots

The first humanoid robot was built in 1973 by Waseda University in Japan and was called *WABOT-1*, which stands for WAseda RoBOT 1 [Kato 73]. Although not embedded, it could walk, recognize and manipulate some visualized objects as well as understand spoken language and use an artificial voice. Its next version, *WABOT-2*, was introduced twelve years later with the remarkable capability of playing the piano [Sugano 87]. In 1986 Honda secretly started a project on this field and 10 years later, they presented *P2* which is the first embedded humanoid robot able to stably walk [Hirai 98], followed by *P3*. The world famous *ASIMO* (Advanced Step in Innovative MObility), which is a white child-size astronaut-like humanoid robot, was presented by Honda in 2000 with an astonishingly friendly design [Hirose 01]. It is sometimes believed that the impressive design and capabilities of ASIMO triggered the research on humanoid robots worldwide [Kaneko 09]. Some of these robots are shown in Fig. 2.2.



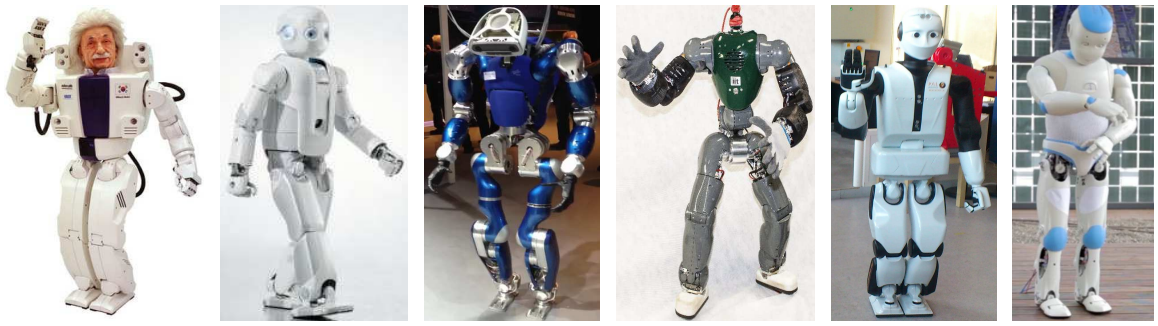
(a) WABOT-1 **(b)** ASIMO **(c)** HRP-4C **(d)** KOBIAN-R **(e)** Kenshiro

Figure 2.2 – Examples of humanoid robots (part 1)

After the outstanding work done by Honda, Japan was the country that led the development of humanoid robots. A remarkable step was the *Humanoid Robotics Project* (HRP), which was launched in 1998 by the Ministry of Economy, Trade and Industry of Japan (METI) and was developed by the Japanese National Institute of Advanced Industrial Science and Technology (AIST) together with Kawada Industries [Kaneko 02]. Within this project, different adult-size humanoids with very powerful capabilities were developed: *HRP-2P* in 2002, *HRP-2* in 2004 [Kaneko 04], *HRP-3* in 2007 [Kaneko 08], *HRP-4C* (Cybernetic Human), which has the head and figure of a Japanese girl, in 2009 [Kaneko 09], and *HRP-4* in 2010 [Kaneko 11]. Fujitsu Automation introduced the small humanoid robots *HOAP-1*, *HOAP-2*, and *HOAP-3* in 2001, 2004 [Shan 02] and 2005, respectively. Sony also developed a small humanoid robot initially called *SDR-3X* (Sony Dream Robot) but then renamed *QRIO* (Quest for cuRIOsity), which was presented in 2003 [Geppert 04]. In 2005 Toyota released *Partner*, a robot that could play the trumpet, and in 2007 they presented a version that was able to play the violin [Kusuda 08]. Waseda University continued the development of humanoids and they introduced *WABIAN* (WAseda BIpedal humANoid) in 1996, *WABIAN-2R* in 2006

[Ogura 06], and the emotion expression robot *KOBIAN-R* in 2008 [Endo 08], which is able to show different facial expressions. The University of Tokyo also developed a series of humanoid robots: *H5*, *H6* in 2000, and *H7* in 2006 [Nishiwaki 07], as well as musculo-skeletal robots as *Kenta*, a tendon-driven robot, in 2003 [Inaba 03], *Kotaro* in 2006 [Mizuuchi 06], its new version *Kojiro* in 2007 [Mizuuchi 07], and more recently, in 2012, the impressive *Kenshiro* [Nakanishi 12]. In 2013 the former Japanese Schaft Inc presented their last version of the *Schaft* robot, based on the HRP-2 series, which is currently one of the most powerful humanoid robots.

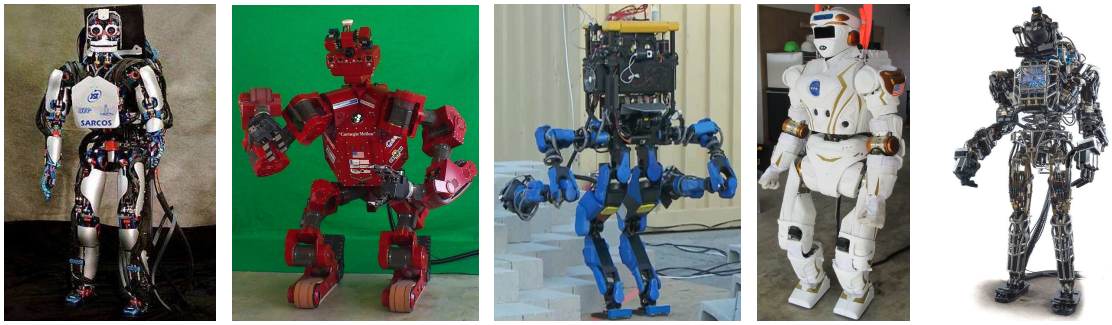
Another country that early entered the development of humanoid robots was South Korea. The Korea Advanced Institute of Science and Technology (KAIST) developed several series of powerful humanoid robots: *KHR-1* (KAIST Humanoid Robot 1) in 2002 [Kim 02], *KHR-2* in 2004 [Park 05a], *KHR-3* (better known as *HUBO*) in 2005 [Park 05b], *Albert HUBO*, a robot with the android "face" of Albert Einstein, in 2006 [Oh 06], and *HUBO 2 plus* in 2011 [Grey 13]. Also, the Korean Institute of Science and Technology (KIST) together with Samsung Electronics participated in a Network-Based Humanoid Project of the Korean Ministry of Information and Communication (MIC) and developed the first network-based humanoids in the world called *MAHRU-II* and *AHRA* in 2005 [You 05]. Then, they introduced *MAHRU-III* in 2007 [Kwon 07], *MAHRU-R* in 2008 [Chang 08], *MAHRU-Z* in 2010 [Kim 11]. More recently, Samsung introduced the torque controlled robot *Roboray* in 2012 [Kim 12]. Some of these robots can be seen in Fig. 2.3.



(a) Albert Hubo **(b)** Roboray **(c)** TORO **(d)** COMAN **(e)** REEM-C **(f)** Romeo

Figure 2.3 – Examples of humanoid robots (part 2)

Some European countries have also developed humanoid robots. In Germany, the Technical University of Munich (TUM) has developed *LOLA* in 2009 as a result of the JOHNNIE Project [Lohmeier 09], and the German Aerospace Center (DLR: Deutsches Zentrum für Luft- und Raumfahrt) has developed *TORO* (TOrque Controlled Humanoid RObot) in 2013, which has been build on the basis of their previous two legs *DLR biped* [Ott 10]. The Italian Institute of Technology (IIT), in cooperation with other universities and research laboratories along Europe, has developed the torque controlled *iCub* (Cognitive Universal Body) as part of the RobotCub project [Metta 10], and *COMAN* (COMpliant HuMANoid Platform) in 2012 [Tsagarakis 13]. In Spain, the University Carlos III has build a humanoid called *Rh-1* in 2007 [Arbulu 07], and based on it they introduced *TEO* (Task Environment Operator) [Monje 11] in 2011. Also, the Barcelona based company Pal Robotics built the humanoid *REEM-B* in 2008 as a research platform in the field of service robotics [Tellez 08], and more recently, in 2014, they presented *REEM-C*, with a more human-friendly design. The French company



(a) Sarcos **(b)** Chimp **(c)** Schaft **(d)** Valkyrie **(e)** Atlas

Figure 2.4 – Examples of humanoid robots (part 3)

Aldebaran Robotics has developed *Nao* in 2007 [Gouaillier 09], which is probably the most widely spread small humanoid robot in the world and is used for research in different domains. In 2014 they presented *Romeo*, a child-size torque-controlled humanoid developed as a project between the company and several French research laboratories.

In the United States, the first humanoid robots were built by the SARCOS Research Corporation in cooperation with ATR (Advanced Telecommunications Research Institute International) in Japan. They developed the hydraulic robots *Erato DB* (Dynamic Brain) in 2000 [Atkeson 00], and *CBi* (Computational Brain interface) in 2006 [Cheng 07], but both of them are sometimes simply called the *SARCOS* robots. In recent years the development of American humanoid robots has considerably increased, in part due to the DARPA (Defense Advanced Research Projects Agency) Robotics Challenge (DRC). Virginia Tech, in collaboration with other universities, has developed *CHARLI* (Cognitive Humanoid Robot with Learning Intelligence) in 2010 [Knabe 13], and *THOR* (Tactical Hazardous Operations Robot), which is expected to be finished by 2014. The National Robotics Engineering Center (NREC), a part of Carnegie Mellon University's Robotics Institute, has built the *CHIMP* (CMU Highly Intelligent Mobile Platform) robot in 2013 for the DRC [Stentz 15]. This robot presents the peculiarity that although it has legs, it uses tracks for its locomotion, which provide it a robust static balance. NASA John Space Center (JSC) has developed the humanoid *Valkyrie*, also for the DRC, and has introduced it in 2013. Boston Dynamics built *PETMAN* in 2012 [Nelson 12], and in 2013 they released the hydraulic *Atlas*, one of the currently most powerful humanoid robots, which was specially built for the DRC. Some of these robots are shown in Fig. 2.4.

2.1.4 Challenges

The research and interest in humanoid robotics has greatly increased over the last few years, but in spite of the current plethora of humanoid robots, their usefulness is still very limited. For example, the recent DRC trials in December 2013 showed the lack of capabilities and robustness when performing different tasks. Each task of this challenge had to be concluded in 30 minutes, while a man can easily finish the same task in less than a minute; and indeed, most robots could not successfully accomplish the tasks[3] even though remote human operators

[3]Videos of the trials are available via DARPA's youtube channel. Due to robot slowness, on the 7th April 2014 DARPA tweeted: "Can't sleep? Try watching 9 hrs of #DARPADRC robots attempting to open doors".

could control them. Moreover, all the walking humanoids fell down at least once while trying to perform the task. This clearly shows that humanoid robotics is not a mature technology yet, and there are still many control problems to overcome.

One of the biggest challenges involves the coordination and development of whole body motion, since humanoids are highly redundant typically with 30 to 40 degrees of freedom. During their motion, they usually have to interact with some objects; thus, manipulation becomes an important part of the challenge. Also, humanoid robots are underactuated systems whose position and orientation in the environment cannot be directly controlled but are the result of the interaction with the environment and the proper joints motion [Wieber 06a]. Moreover, this non-fixed interaction leads to a major concern: the robot must not fall down while performing a task. Thus, balance is of vital importance for humanoid robots, and the control of the Zero Moment Point (ZMP) [Vukobratovic 69], or the Capture Point (CP) [Pratt 06b] have been proposed, although their integration within complex control frameworks is not straight forward.

The reduction of the computational cost to process the sensors information and to generate useful movements for the robot is also currently a challenge. New, and more powerful artificial intelligence, computer vision, data fusion and navigation algorithms typically take time to compute on very powerful machines. Furthermore, their usage within control frameworks that use some model of the robot subject to physical and operational constraints to generate motion trajectories presents the need for highly performant machines. In fact the achievement of real-time usually comprises a trade-off between accuracy and efficiency. And not only is the software level a challenge but also the hardware level: perception sensors such as powerful vision systems, tactile mechanisms, as well as powerful actuators constantly need to be improved [Durán 12]. All these problems are still to be overcome for humanoid robots to behave as useful service robots able to help people in their daily life.

2.1.5 Applications of Humanoid Robots

Humanoid robots are virtually suitable for any environment where humans can work. They can become the universal worker for any manufacturing process since they can do repetitive and even "intelligent" tasks without needing a special design. The generic design can even make the mass production of humanoids possible drastically reducing their cost. Works involving maintenance tasks of industrial plants, security services of homes and offices, human care, and cooperative works can be possible. Teleoperation is also an important application provided that the human structure makes their operation more natural for operators. Also, telepresence [Lee 12] can be carried out by humanoids.

Not only can humanoid robots work where humans can, but they can also work where humans cannot. In fact it is promising to use humanoids in tasks where human lives can be at risk due to the environment or the task itself. Particularly a humanoid fire-fighter or rescuer is appealing for cases of natural disasters, fires, buildings collapse, general security or hazardous environments (like radioactive places). Space missions are another important area due to the inherent dangerousness that space poses to astronauts or the complete current inability to send a human to remote places. There is a need for a human complement or replace and projects like NASA Robonaut [Ambrose 04] or DLR Justin [Wimbock 07] are already addressing this problem. Military suits can also be tested without risking the life of people [Nelson 12].

Another potentially big domain is household. Humanoids can one day become the most used servant at home. Some projects explicitly address this problem like the Armar robot [Asfour 06], Twendy-One, or HERB [Asfour 06]. Humanoids can also be used for entertainment in robot competitions like FIRA (Federation of International Robot-soccer Association) and the RoboCup, whose initiative is that by 2050 a team of humanoid soccer players shall win the soccer game, complying with the official FIFA rules, against the winner of the most recent World Cup [Kitano 98]. Also they can be used in biomechanics to have a deeper understanding of how humans work and even think. In fact, humanoids can be used as platforms to test and develop cognition and artificial intelligence methods and some projects are also devoted to this field [Metta 10].

2.2 Approaches in the Control of Humanoid Robots

The control of a robot consists in defining a law that allows the generation of feasible motion within the limits of the mechanism such that a desired objective is achieved. As stated in Section 2.1.4, the control of humanoid robots poses particular challenges due to the redundancy, underactuation, the necessity to keep balance at every instant of time, and additional constraints to make motion human-like. Different methods can be used to make a robot move depending on the application, the complexity of the task, and even the specific nature of the robot. The main control methods currently used for humanoid robots are the following: motion planning, inverse kinematics control, inverse dynamics control, and optimal control. This section discusses the state of the art in these approaches.

2.2.1 Motion Planning

Motion planning consists in automatically finding a feasible path from an initial configuration to a given goal satisfying certain constraints. In robotics, for instance, it can compute a path that moves the hand from its current pose to a desired one avoiding collision with some obstacles in the environment. For many purposes, it can be of great interest to determine the path minimizing a convenient criterion. Currently, there are many applications in vehicles, robots, digital characters, molecules, design verification, amongst others.

Basic Concepts

Although the research in motion planning can be traced back to the late 60's, most of the problems formalization started in the 80's thanks to the combined and growing research in artificial intelligence, computer science and mathematics. Currently, the main approaches can be found in books such as [Latombe 91, Choset 05, LaValle 06]. An important concept in motion planning is the *Configuration Space* ($\mathcal{CS}$), which is the set of all the possible configurations that a mechanism can attain. This fundamental concept, borrowed from mechanics and introduced by [LozanoPerez 83], can be used with several tools from geometry, topology and algebra, and has provided the theoretical basis for motion planning.

For a robot with n independent degrees of freedom (DoF), the $\mathcal{CS}$ is an n-dimensional manifold $\mathcal{M}$ that contains all the configurations $q \in \mathcal{M}$ of the robot. The importance of the $\mathcal{CS}$ is that it changes the problem of moving a body in $SE(3)$ to moving a point in $\mathcal{CS}$. Two particular subspaces of $\mathcal{CS}$ are of interest:

- *Obstacle Configuration Space ($\mathcal{CS}_{obst}$).* It is formed by the robot configurations that generate self-collisions or collisions with some object in the environment. The space is represented as $\mathcal{CS}_{obst} \subset \mathcal{CS}$.
- *Free Configuration Space ($\mathcal{CS}_{free}$).* It is the space where the robot can freely move and is the complement of the obstacle space: $\mathcal{CS}_{free} = \mathcal{CS} - \mathcal{CS}_{obst}$.

Using these spaces, the problem of motion planning can be stated as finding the continuous path $p(t)$ from a start configuration $p(0) = q_0$ to a goal configuration $p(1) = q_f$ avoiding collisions; that is, $p : [0, 1] \rightarrow \mathcal{CS}_{free}$, where t defines some parameterization.

Generic methods

The solution to the motion planning problem can be obtained using some classical methods which can be classified as: deterministic algorithms, sampling-based algorithms, and path optimization.

a) Deterministic algorithms. These algorithms, developed since the past 30 years, always compute the same valid path $p(t)$. Methods such as cellular decomposition, Voronoi diagrams, visibility graphs and Canny's algorithm rely on the construction of an explicit and exact representation of $\mathcal{CS}_{obst}$ to build a graph, or roadmap, that represents the connectivity of $\mathcal{CS}_{free}$ [Goodman 04, Canny 88, Indyk 04]. Although these methods are complete, they become very computationally expensive in high-dimensional configuration spaces, even when the dimension of $\mathcal{CS}$ rises above 4. Other deterministic approaches use artificial attractive potentials on the goals and repulsive potentials on the obstacles and the start configurations to guide the search towards the goal avoiding the obstacles [Khatib 86]. However, the locality of this type of planners can lead to a solution that does not achieve the goal, for instance, in a maze environment.

b) Sampling-based algorithms. These methods, developed over the last 20 years, approximate the connectivity of $\mathcal{CS}_{free}$ by randomly sampling configurations from $\mathcal{CS}$ and rejecting the configurations that produce collisions through Boolean collision detection techniques [Hudson 97, Gottschalk 96, Hsu 97]. The main examples are *Probabilistic Roadmaps* (*PRM*) [Kavraki 96], and *Rapidly-exploring Random Trees* (*RRT*) [Kuffner 00]. In particular, the RRT algorithm uses the Voronoi bias to explore $\mathcal{CS}_{free}$ and grow a random tree. Each iteration attempts to extend the tree by adding new vertices in the direction of a randomly selected configuration. Trees can be grown from the initial and goal configurations, which is sometimes referred to as *Bi-directional RRT* (BiRRT), making the algorithm more efficient [Kuffner 00]. Variations that asymptotically converge towards the global minimum solution path have been proposed as *PRM** and *RTT** [Karaman 11], for generic cases, and *Kinodynamic RRT** for systems with controllable linear dynamics [Webb 13]. RRT's have also been extended in the state space to *LQR-Trees* [Tedrake 10], which use a dynamic cost-to-go distance metric, and *LQR-RRT** [Perez 12], which uses a linearization of the system to derive a coherent extension method. Stochastic-optimization for determining a new potential state based on transition tests has been called the *T-RRT* (Transition-based RRT) [Jaillet 08]. In general, the advantages of these methods are their probabilistic completeness and their ability to deal with very high-dimensional configuration spaces.

c) Path optimization. The goal of these methods is to optimize the trajectory starting with a valid collision free path and trying to shorten it while ensuring that the path is still

valid. For example, *greedy optimization* tries to directly connect the start configuration to the closest goal configuration node that generates a collision free path; then, it continues with the next nodes in the path until reaching the goal. Another example is *random optimization* which tries to ignore some random nodes keeping the rest.

Motion Planning in Humanoid Robotics

Classical motion planning methods determine collision-free trajectories involving only geometrical considerations. However, the control of polyarticulated systems needs the synthesis of models that describe the effects of joint variations on the whole robot configuration. For instance, moving an arm with only RRT's considering environmental and self-collisions can effectively achieve the goal configuration, but the arm risks to make very unnecessary, unnatural and inefficient movements. Underactuated robots present the additional problem that the planning algorithm might generate paths that do not preserve the balance of the robot or the motion might not be physically plausible. To overcome these problems within these frameworks, trajectories are generated using methods of inverse kinematics, inverse dynamics or optimal control and additional constraints are added to satisfy conditions like multiple contacts or the dynamic balance of the system. For instance, motion primitives that have been predefined by a human expert based on prior knowledge can be used to guide the planner [Zhang 14].

For humanoid walking, footsteps planning can be achieved using deterministic approaches with dynamic adjustment of the footstep transition model [Chestnutt 05] considering the smoothness of the trajectories for posture transitions [Ayaz 07]. But these approaches do not guarantee the completeness of the solutions and probabilistically complete methods have been proposed to solve this problem. For instance, RRT's can be adapted to explore the discrete footstep space [Xia 09, Perrin 12], and anytime search-based planners can be used to generate paths that are goal-directed guaranteeing to stay within bounded limits from the optimal solution [Hornung 13]. Also, features of the environment can be used to guide towards the solution; for instance, multi-contact planning can be achieved using a multiple-contact-point stance planner that looks for authorized contact surfaces to help the robot reach its goal [Bouyarmane 12a, Escande 13]. Other approaches divide the high-dimensional problem into smaller problems and solve them successively [Zhang 09]. An example is presented in [Yoshida 08], where a 36 degree of freedom robot is reduced to a 3 degree of freedom bounding box and a PRM is applied for the path planning problem of the box.

During the last few years, planning on constraint submanifolds of $\mathcal{CS}$ has been the focus of many researches. For example, manifolds can be defined by contact limb positions and static balance constraints, and the plan can be done on a union of submanifolds. In this way, statically balanced locomotion for hexapods and humanoid robots on uneven terrain can be obtained [Bretl 06, Hauser 10]. Another method proposed the planning algorithm *CBiRRT* (Constrained Bidirectional RRT) together with some Jacobian-based methods to consider end-effector pose constraints in the plan and generate humanoid motion [Berenson 11]. Also, RRT can be adapted within a random diffusion framework to work according to some constraint and generate statically stable configurations [Dalibard 13]. More recent methods combine path planning with optimal control to generate motion for humanoid robots in a cluttered environment [ElKhoury 13].

2.2.2 Inverse Kinematics (IK)

Kinematics, in general, is the study of the position, velocity and acceleration of a mechanical system without considering either the forces (or torques) that generate the motion or the dynamic properties, such as mass or inertia, of the bodies composing the system.

Basic Concepts

The *joint space*, also called the *configuration space*, of a robot with n DoF is the n-dimensional manifold $\mathcal{Q}$ containing all the possible values that the joint variables q can take. For robotic manipulators, the pose of a single element called the *end-effector* is typically of interest. However, for more generic robots such as humanoids, this is generalized to the *operational points* [Khatib 87], which can represent any part of the robot whose pose is of interest. The velocity of an operational point x can be represented by the *twist* [Murray 94] or *spatial velocity* [Featherstone 08] ξ that comprises both the linear and angular velocities; alternatively, it can be given by the variation of its representation $\dot{x}$. In robotics there exist four sub-domains related to kinematics.

- *Forward kinematics* (also called *forward geometry*[4]). It consists in finding the pose $x \in SE(3)$ of an operational point given a certain joint angle configuration $q \in \mathcal{Q}$, and is described by the map $f : \mathcal{Q} \to SE(3)$ such that $x = f(q)$.
- *Inverse kinematics* (also called *inverse geometry*[4]). It aims at finding the joint configuration $q \in \mathcal{Q}$ that achieves a given pose $x \in SE(3)$ for a certain operational point. In some specific cases, this can be represented by the inverse map $f^{-1} : SE(3) \to \mathcal{Q}$ such that $q = f^{-1}(x)$, but in general $f^{-1}(x)$ is not unique and can even be undefined.
- *Forward differential kinematics* (also called *forward instantaneous kinematics* or simply *forward kinematics*[4]). It consists in finding the operational-point twist $\xi \in se(3)$ that is produced due to joint variations $\dot{q} \in T_q(\mathcal{Q})$, which is expressed as $\xi = J_o(q)\dot{q}$, where $J_o : T_q(\mathcal{Q}) \to se(3)$ is the tangent application called the *basic* [Khatib 87] or *geometric Jacobian* [Spong 06], and $T_q(\mathcal{Q})$ is the tangent space to $\mathcal{Q}$ at q. Alternatively, it can be formulated as finding the first order variations $\dot{x}$ that are produced by $\dot{q}$, expressed as $\dot{x} = \frac{\partial x}{\partial q}\dot{q}$. In this case, $J(q) = \frac{\partial x}{\partial q}$ is a differential map called the *analytic* or *task Jacobian* [Spong 06] or simply the *Jacobian* [Khatib 87].
- *Inverse differential kinematics* (also called *inverse instantaneous kinematics* or simply *inverse kinematics*[4]). It is the problem of finding the joint variations that produce a given variation of the end effector (Section 3.2.2). It can be used to iteratively solve the inverse kinematics problem.

In general, the control objective is defined by a reference point or a reference trajectory in $SE(3)$, and the control problem consists in determining the appropriate evolution of the joint configuration $q(t)$ that achieves the objective. Consider the task i with m_i DoF. An n-DoF robot is said to be *kinematically redundant* with respect to task i if $n > m_i$, and $n - m_i$

[4]The Greek root "*kinesis*" refers to movement or motion; therefore the term *geometry* is more accurate than *kinematics* when the map is between static spaces. Similarly, when velocities are implied, *kinematics* is more accurate than *differential kinematics*.

is known as the *degree of redundancy* with respect to task i [Nakamura 90]. If there are k simultaneous tasks, each of them with m_i DoF, the system is still kinematically redundant if $n > \sum_{i=1}^{k} m_i$.

Classical Kinematic Methods

Forward kinematics is traditionally represented using the classical *Denavit-Hartenberg* (DH) parameters [Denavit 55], or the *modified* DH parameters introduced in [Craig 89], but although very popular for serial manipulators, they might lead to ambiguities for closed or tree chains [Khalil 04] and can even be not suitable for robot calibration [Everett 87]. Alternative methods have been proposed: the *Khalil-Kleinfinger* notations that are suitable for closed-loop robots [Khalil 86], the *Hayati-Roberts* coordinates that avoid coordinate singularities for parallel systems [Hayati 85, Roberts 88], the *products of exponentials* (PoE) [Park 94], *screw* theory-based kinematic modeling [Tsai 99], or even ad-hoc methods for specific cases such as for humanoid robots [Kajita 05].

For simple kinematic chains, the IK problem can be analytically solved and closed equations can be obtained through two classical types of approaches: (i) *algebraic approaches*, such as inverse transforms [Paul 81] or dialytic eliminations to derive polynomials [Raghavan 93]; and (ii) *geometric approaches* like joint decoupling when the last three joint axes intersect [Peiper 68] or the so-called *Paden-Kahan* sub-problems which are based on the PoE [Paden 86, Kahan 83]. However, as the complexity of the chain increases, the complexity of the analytical solution also increases due to the strong non-linear terms. For more complicated chains, analytical solutions become tedious or even ill-posed due to the redundancy of the system.

Kinematic Control of Redundant Robots

Even though kinematic redundancy offers more dexterity and versatility to the robot, like joint range availability, singularity avoidance, obstacle avoidance, compliant motion or energy consumption minimization, it increases the complexity of inverse kinematics [Siciliano 91]. For a given task, there is more than one solution in the joint space, and closed form solutions are in general not possible. Moreover, solutions that satisfy both the main task and other complementary tasks as closely as possible are usually desired. For instance, a redundant manipulator has typically to achieve the goal pose avoiding collisions with objects in the environment. In these cases, numerical methods are preferred and have shown to offer good results.

Numerical methods treat the IK problem as an optimization problem with, in some cases, additional constraints. There are two main approaches for solving IK through optimization: (i) *Global methods*, which find an optimal path for the entire trajectory but have a high computational cost [Baillieul 90]; and (ii) *Local methods*, which are used in real time and only compute an optimal change Δq for a small change Δx, integrating Δq to generate the joint space trajectory $q(t)$. A well-known local method is the *Resolved Motion Rate Control* [Whitney 69] which finds the change of q as a solution to the problem $\dot{x} = J(q)\dot{q}$ using the inverse J^{-1}, if J is invertible, or a suitable pseudo-inverse J^{+}. In order to avoid problems with singular configurations, the damped pseudo-inverse can be used instead [Nakamura 86, Wampler 86]. A more generic solution arbitrarily adds a projection onto the nullspace of

J since doing so does not modify the desired $\dot{x}$ but supplies some freedom for other tasks [Liegeois 77]. When there are two or more tasks to be satisfied at the same time, some compromises have to be taken into account. To this end, there are two main methods:

- *Task space augmentation.* It is a weighting approach that extends the dimension of the original task space by imposing constraints to the joint variables [Sciavicco 87]. Problems occur when tasks conflict with each other because they implicitly have the same priority. In these cases, weights are assigned to each task and a compromise among them is found, although no task is completely satisfied. To choose a constraint task that avoids conflicts with other tasks, an approach called the *Extended Jacobian method* (EJM) zeroes the projection of the cost function gradient in the null space of the Jacobian [Baillieul 85].

- *Task prioritization.* It consists in generating a set of prioritized tasks where the lower priority task produces a self-motion that does not interfere with the higher priority task leading to a multiple priority-order resolution method [Nakamura 87]. The approach finds a joint velocity $\dot{q}$ such that every task x_i is satisfied unless it conflicts with a higher-priority task x_j, $(j < i)$, in which case its error is minimized. An algorithmic solution was proposed in [Siciliano 91] and improvements in the algorithmic computation of the null-space projector were introduced in [Baerlocher 98].

Kinematic Control of Humanoid Robots

From a kinematic point of view, humanoid robots present a tree-like structure that includes multiple connected chains, and a high number of DoF. As a consequence, the kinematic structure is often redundant with respect to most tasks, in which case the robot is said to be under-constrained. Therefore, methods developed for generic redundant robots are usually applied in humanoid robotics with some adaptations or additions. Due to the complexity of the kinematic configuration, closed-form solutions for the IK problem are usually very complex, but recently some classical methods have been used and modified, to obtain closed equations for specific humanoid robots which are treated as a composition of several kinematic chains [Ali 10, Nunez 12]. However, methods based on instantaneous IK, which compute an increment in q, are usually preferred. This linearization of the problem offers an infinite number of feasible solutions for humanoid robots: there exist different joint updates that achieve the same task. This leads to the possibility of performing different tasks at the same time, and the IK control must be capable of properly handling them.

Some works have proposed the usage of a weighting approach, such as [Tevatia 00] that uses a simplified EJM augmenting the task space, or [Salini 09] where a continuous variation of the weight values is associated to each task relatively to its importance. Due to the nature of these schemes, problems arise when there are conflicting tasks since in those cases none of the tasks can be fully satisfied. Methods based on task-prioritization solve these problems and have become the preferred techniques in IK control. They can even be considered as the current "state of the art" in humanoid robotics control due to their relatively low computational cost, their straightforward implementation, and the maturity of the approach. Several works with different humanoid platforms use this methodology to solve the redundant IK problem at the velocity level using only equality constraints [Gienger 05, Yoshida 06, Mansard 07, Neo 07, Galdeano 14]. For imposing inequality constraints to the control framework at any

hierarchical level, a sequence of optimal resolutions for each priority level has been proposed in [Kanoun 09], a more efficient computation based on orthogonal decompositions can be found in [Escande 14] and a smooth interchange between priority of consecutive prioritized tasks is introduced in [Jarquin 13].

2.2.3 Inverse Dynamics

Dynamics is the study of the relation between the robot motion and the generalized forces that act on the robot generating that motion. This relation considers parameters such as lengths, masses and inertia of the elements composing the robot.

Basic Concepts and Computation

In the dynamic model, the motion is represented through joint variables acceleration $\ddot{q}$, or operational points acceleration $\ddot{x}$. For rotational joints, the generalized forces are equivalent to the joint torques, and for prismatic joints they are the joint forces. There are two main problems in dynamics:

- *Forward Dynamics.* It expresses the motion of the robot as a function of the generalized forces applied to it.
- *Inverse Dynamics.* It expresses the generalized forces acting on a robot as a function of the robot motion.

There exist two main formulations to compute the robot dynamic model: the Lagrange approach, and the Newton-Euler approach. The *Lagrange* approach was the first one used in the robotics community [Uicker 67, Kahn 71, Bejczy 74]. Its main advantage is the clear separation of each component of the model; but in general, it is computationally expensive, although an efficient formulation was presented in [Hollerbach 80]. The *Newton-Euler* approach, proposed in [Orin 79], does not provide a clear separation of the terms but due to its recursivity a lower computation time can be obtained. Thus, it is the preferred implementation for computer calculations. The most used algorithms for this approach can be found in [Featherstone 00, Featherstone 08], for instance, the *composite-rigid-body algorithm* (CRBA), the *articulated-body algorithm* (ABA), the *recursive Newton-Euler Algorithm* (RNEA), and some global analysis techniques. Although currently less used in robotics, the *Hamiltonian* approach has also been applied to the analysis of robot dynamics [Spong 92] and numerical methods for integrating Hamilton's equations are becoming more efficient [Selig 07].

The term *centroidal momentum* has been recently proposed as the sum of the individual link momenta expressed at the robot Center of Mass (CoM) [Orin 08], and the dynamics expressed at the CoM has been introduced as *centroidal dynamics* [Orin 13]. This is a particularly important concept since the motion of humanoid robots typically involves large angular momenta. The dynamic model of a robot can be expressed in two ways depending on the spaces that are used to describe the motion and the control input. The two approaches are: the joint space formulation, and the operational space (or task space) formulation. The *joint space formulation* is the classical approach and uses the joint space acceleration to specify the motion, and the generalized forces acting on the actuated joints to describe the control. The *operational space formulation* [Khatib 87] represents the motion directly using the task space acceleration, which needs a reformulation of the forces as task space generalized forces.

Dynamic Control of Robotic Manipulators

The dynamic control of a robot consists in finding the proper generalized forces that have to be applied to the joints in order to generate the desired trajectory for an operational point. Classical methods for robotic manipulators include two main types of control: (i) Joint Space Control, which is based on the joint space formulation and includes methods such as *PD* and *PID* controllers [Arimoto 83], inverse dynamics control (in this context usually called *computed torque* [Markiewicz 73]), robust control in presence of parameter uncertainty [Lozano 92], Lyapunov-based control [Slotine 87], adaptive control [Dubowsky 79], adaptive compensation of gravity [Tomei 91], amongst others; and (ii) Operational Space Control, based on the operational space formulation with methods such as inverse dynamics control [Caccavale 98] or PD control, amongst others [Nakanishi 08].

For redundant manipulators, one of the first approaches was the *Extended Operational Space* formulation that proposes the dynamic control of joint-space postures and adds nullspace vectors to the rows of the Jacobian [Park 01]. A similar approach of augmenting the Jacobian is used in [Harmeyer 04], and a projection of the dynamics onto a constrained consistent manifold using the decomposition of the constraint Jacobian is proposed in [Aghili 05]. Based on the resolved motion rate control in kinematics, the *resolved acceleration control*, which relates the task acceleration to the joint acceleration [Luh 80], was proposed for the fixed upper body of a humanoid robot but with the limitation that only kinematic constraints like collision avoidance and joint limits can be used in the framework [Dariush 10].

Dynamic Control of Humanoid Robots

For a humanoid robot, classical dynamic control techniques are not sufficient since coordination of the motion is required, environmental forces need to be considered, and balance has to be kept at all moments. For the robot balance, a stability criterion such as keeping the CoM or the ZMP (Section 3.1.2) inside the support polygon must be enforced [Wieber 02]. For planar surfaces the constraint on the ZMP implies a control of the contact forces. Thus, the interaction with the environment is always present since the feet are in contact with the ground and additional contacts of other parts of the robot body might also be necessary. These contacts generate an effect on the joints generalized forces, which cannot be controlled using only kinematic methods but with dynamic approaches. The dynamic control ensures the physical feasibility of the motion and allows for faster movements without losing balance.

The *operational-space inverse-dynamics* (OSID) is a more specific framework for controlling the whole-body of humanoid robots considering contacts and a set of different constraints. The first approach, proposed in [Khatib 04b, Sentis 05], is based on a two stage mapping to obtain consistent contact forces and is equivalent to successive projections onto the nullspaces of the previous tasks. Therefore, new tasks can be added without dynamically interfering with higher priority tasks. An approach based on orthogonal decompositions and projections to resolve over-actuation in double support, avoiding the inversion of the inertia matrix and gaining robustness against model uncertainty, was presented in [Mistry 10]. Although both approaches differ in their formulation, it has been shown that they are equivalent with respect to the minimization of different cost functions [Righetti 11b]. An improvement to this approach consists in creating an optimal distribution of contact constraints by pushing ground reaction forces inside the frictional boundaries [Righetti 11a]. However, these methods only deal with equality dynamic constraints. Generic tasks or constraints specified as inequalities,

which are important to specify joint limits, collision avoidance, visual field tasks, amongst others, cannot be taken into account in a straightforward way.

Other methods use the OSID within frameworks that involve some type of optimization to find the local solution. The most popular approaches use *Quadratic Programming* (QP), which allows for the specification of both equality and inequality constraints. The latter type of constraints is fundamental in humanoid robotics to directly model unilateral contacts, and it is also important to properly specify some particular tasks. A weighting scheme is used within a QP in [Collette 08] to control different objectives, but the approach is mainly oriented towards robust balance. Another similar weighting approach is oriented towards the sequencing of dynamic tasks, computing the torque controls to satisfy the constraints [Salini 11]. The decoupling of the humanoid dynamics into holonomic and non-holonomic part based on the Gauss principle [Wieber 06a] has been also exploited within a QP to produce feasible whole body motion in [Bouyarmane 12b]. An implementation for the lower part of a torque-controlled humanoid robot has been shown in [Herzog 13] using active set QP solvers. Also, OSID has been used within a prioritized scheme that allows the usage of for both equalities and inequalities at any level of the hierarchy [Saab 13].

Although the previous approaches exploit the full robot dynamics, the angular momentum is not explicitly controlled within these frameworks. Nevertheless, it has been shown that the angular momentum is a natural and important part of human motion, specially when performing complex and fast movements [Popovic 04]. One of the first attempts to control this physical quantity for whole-body motion used a pseudo-inverse of the inertia matrix [Kajita 03b]. Other control schemes propose to control the angular momentum through the control of the system's centroidal momentum, such as [Hofmann 09] where the focus is on gate movement, or [Moro 13] where the centroidal momentum is defined as an attractor within the framework. Also, a prioritization scheme based on a conic optimization to control the centroidal angular momentum generating movements like kick and jump has been introduced in [Wensing 13].

Dynamic Control in Computer Graphics

In computer graphics, there has also been the need of using the dynamics of human-like characters to generate motion that satisfies dynamic constraints [Popovic 00]. For instance, retargeting motion to characters with different physics adds more realism to the animation [Gleicher 98]. B-splines are also used to retarget characters [Lee 99] but some physical limits are not explicitly checked. Another approach adapts a given motion to a character keeping the contact between the limbs and external objects [Shin 01]. More recent works have introduced QPs to generate physically consistent motion guiding humanoid avatars to achieve various tasks through weights that establish the relative importance of each task [Abe 07]. Other examples of weighting schemes in this field include [Macchietto 09], where the focus is on balance, and [DaSilva 08] where a Model Predictive Control (MPC) is used to track motions. An approach based on a novel prioritization algorithm, which uses a nested sequence of objectives re-parameterized at each level of the hierarchy, has been proposed in [deLasa 10] with the name of *feature-based control*. Motions like balancing, jumping or walking for a humanoid robot-like character have been achieved using different tasks with this control method. This approach has been later extended within an optimal control framework [Brown 13]. However, there exists a gap between physically plausible animations and real motion on the robot, and

the motion generated in computer animation does not guarantee a feasible direct reproduction on a real humanoid. In reality, physical interactions between the feet and the ground significantly affect the whole body motion and errors in considering them can cause the robot to fall down making the generated motion completely unfeasible.

2.2.4 Optimal Control

Optimal control, also known in robotics as *trajectory optimization* or *trajectory filtering*, consists in finding a trajectory and its associated control law (policy) that satisfies some predefined optimality criterion. In general mathematical terms, it concerns the properties of control functions which, when inserted into a differential equation, give solutions that minimize a cost or measure of performance. But it also concerns optimization problems with dynamic constraints which might be functional differential equations, difference equations, partial differential equations or equations with another form.

Introduction

It is often said that optimal control started more than 300 years ago [Sussmann 97], with the solution to the brachystochrone problem by J. Bernoulli. But most formalization on this field began about 60 years ago with the introduction of the *linear quadratic* control problem [Wiener 49] as a minimization of a mean-square-error criterion with the form $J = E\{e^2(t)\}$, where $e(t)$ is the error and $E\{x\}$ represents the expected value of the random variable x. Another major step was the introduction of *Dynamic Programming* (DP) [Bellman 56], which reduces the search to a partial differential equation called the *Hamilton-Jacobi* equation, introduced to solve discrete-time optimal control problems. However, it is sometimes considered that optimal control was not completely formalized until the formulation of the *maximum principle* in [Pontryagin 62], which is strongly based on the *Calculus of Variations*. In fact, calculus of variations is considered as the roots for optimal control and presents the notable feature of taking account of pathwise constraints on values of the control functions. Another remarkable formulation was the *linear quadratic regulator* (LQR) and the *linear quadratic Gaussian* (LQG) in [Kalman 60] to design optimal feedback control laws.

In industry, a control methodology based on the solution of an optimal control problem at each controller update time over a time interval, known as the prediction horizon, evolved with the name of *Model-based Predictive Control* or simply *Model Predictive Control* (MPC) [Richalet 78]. It is sometimes also called *receding horizon control*, or *preview control*. Currently, MPC constitutes one of the most powerful optimality methods for automatic control. It offers the possibility of specifying high-level task goals through simple cost functions, and it automatically synthesizes all the details of the behavior and control law. Another advantage is that MPC avoids the need for extensive exploration by postponing the design of the policy until the last minute, finding controls only for the states that are currently visited. That is, it foresees the future states of the system to choose the best current state according to some pre-established criteria. Other important method is *Differential Dynamic Programming* (DDP) [Jacobson 68] which is a local trajectory-based numerical scheme that provides an efficient solver for direct implicit optimal control problems. A local-global hybrid method with constant local controllers was introduced in [Atkeson 94], and it was complemented by the usage of second order local DDP models to make locally-linear controllers in [Atkeson 03].

Optimal Control of Humanoid Robots

In robotics, optimal control can be used to find the trajectories from an initial posture to a final desired posture, specified as a whole or as a set of sub-objectives, satisfying certain constraints. Very fast and powerful movements can be generated with optimal control, which can comprehend the problems of inverse kinematics or inverse dynamics and can therefore produce better movements. The problem with IK and OSID alone is their inability to properly handle the CoM accelerations, thus overrestricting the motion. A solution typically relies on a dedicated submodel, like a linearized inverted pendulum [Kajita 03a] to capture the future of the system, but this ad-hoc resolution increases the control architecture complexity. Thus, using these schemes the future states of the system can be somehow predicted, but dedicated submodels would need to be developed for each case. However, optimal control is the most suitable approach that allows to take into account all the constraints at the same time. In fact, optimal control can automatically generate the proper trajectory for the CoM in order to achieve fast movements: it acts as a classical pattern generator used for walking schemes, but additionally incorporating whole-body motion. A serious challenge to optimization-based approaches in robotics is that the timescales of the dynamics are faster than in other applications and need a faster response. Currently, the main problem is the computational time, due to the high number of DoF, that forbids its use in real-time applications.

An optimal control method of particular interest in humanoid robotics is MPC. One of the first uses of MPC in robotics was the so-called walking pattern generator (WPG) [Kajita 03a, Herdt 10] which previews the future states of the robot to generate a trajectory for the CoM which enables dynamic stability. In case of multi-task scenarios, a weighted sum of the task objectives can be applied choosing the weights proportionally to the task priority level [Dimitrov 11], but too low weights may lead to interferences between tasks and large weights can introduce numerical conditioning problems. An alternative to weighting task errors is the *prioritized optimal control* proposed in [DelPrete 14b], which ensures that the specified priorities among the tasks are respected avoiding numerical conditioning issues. To speed up the computation of dynamic derivatives, a physics simulator called *MuJoCo* was introduced [Todorov 12], and using MPC, it has allowed to synthesize humanoid behaviors such as getting up from the ground and recovering from large disturbances [Tassa 12]. Other complex dynamic motions examples obtained using optimal control are ball-kicking [Miossec 06], weight-lifting [Arisumi 08], and parkour movements [Dellin 12].

Optimal control methods have also proved to be very powerful tools to generate multiple non-coplanar contact motion, which increases the accessible space of humanoid robots using the environment to help them achieve the goal [Lengagne 13]. For robots in cluttered environment, with many obstacles, some methods combine motion planning algorithms with optimization. For instance, *CHOMP* (Covariant Hamiltonian Optimization for Motion Planning) [Zucker 13], *STOMP* (Stochastic Trajectory Optimization for Motion Planning) [Kalakrishnan 11], and *ITOMP* (Incremental Trajectory Optimization for Real-Time Replanning in Dynamic Environments) have been recently introduced for robotic manipulators but can be extended to humanoid robots considering additional constraints [Park 14]. CHOMP uses penalty terms to obtain an efficient optimal control solver which relies on a covariant gradient descent technique, and STOMP is a similar solver but relies on trajectory stochastic perturbations so that collision-free optimal trajectories can be found without computing function Jacobians. Based on CHOMP, ITOMP provides a suboptimal solution if the optimization

task has not been solved within time constraints and updates the trajectory online.

To include inequality constraints without sacrificing convergence or computational effort, a generalization of DDP, called *control-limited DDP*, has been proposed and applied in simulation to a humanoid robot [Tassa 14]. Another variant of the DDP has been proposed with the name of *square-root DDP* [Geoffroy 14] showing that IK can be seen as a particular case of optimal control when there is no preview horizon. Similarly, OSID is a particular case when no preview horizon is used. A guided policy search algorithm that uses trajectory optimization to direct policy learning and avoid poor local optima has been proposed in [Levine 13], where DDP generates guiding samples. This method has learned neural network controllers for planar swimming, hopping, and walking, as well as simulated 3D humanoid running. Using DP, walking behaviors robust to external disturbances and modeling errors have been generated in [Whitman 13], where multiple-model variants and learning-based variants of DP are proposed. Another method uses the so called Body Retention Load Vector index to represent the physical constraints and achieve steep climbing [Noda 14]. For the animation of physics-based human-like characters, some impressive results have been shown in [AlBorno 13] whose approach optimizes a small set of simple goals applied to short space-time windows composed to express goals over an entire animation. The method synthesized motion with considerable rotation components such as rolling motion [Brown 13], or more recently, walking, hand walking, breakdancing, flips, and crawling [AlBorno 14], which have never been previously synthesized by physics-based methods.

Optimal control has also been used to model human-like running as an offline control problem solved by a direct multiple shooting method optimizing energy-related criteria [Schultz 10]. There is also research on optimization-based motion generation for walking motions without the use of a WPG but with different objective functions such as the minimization of torques, joint velocities or more complex combinations, and constraints such as the ZMP [Koch 12, ElKhoury 13]. Another area is inverse optimal control, which has been proposed as a means to determine, for a given dynamic process and an observed solution, the optimization criterion that has produced the solution. In particular, motion capture data of people has been used to determine the principles of human locomotion and the path generation to given target positions and orientations [Mombaur 10]. It has also been used to learn the function for human running on flat ground, rough terrain and under lateral perturbation [Park 13].

The main drawback of optimal control is the curse of dimensionality, which is particularly important for humanoid robots whose state space is so large that no control scheme can explore all of it in advance and prepare suitable responses for every situation. It would be desirable to obtain optimal control in real time; however, currently there is no approach that can achieve this: the solutions are very time consuming and generic solvers tend to get stuck into local minima or they even return trivial solutions. The problem of finding the proper formulation and resolution of optimal control is still an open issue in robotics.

2.3 Conclusion

Throughout recent years there has been a vast development of human-like robots, which are potentially the best candidates for service robotics since they can move in human environments and perform tasks that humans do in their daily life. This development has led to mature technologies at the hardware level: very robust and powerful platforms can be found nowadays

in research centers. However, there currently exists a gap between what a humanoid platform is able to do and what it can really do. The main reason is the lack of suitable control systems that exploit all the potential and the advantages that such platforms offer. These systems have to deal with aspects such as balance, redundancy, underactuation, motion coordination, environmental interaction amongst others, which make humanoid robot control a difficult problem.

This chapter has presented the state of the art in control approaches for humanoid robots. Motion planning was among the first frameworks used to determine paths in the configuration space and several improvements, some of which have been specially designed for humanoid robots, have recently appeared. Methods based on inverse kinematics have been recalled first for redundant robots and then specifically for humanoid robots. In many ways, inverse kinematics can be considered as the 'de facto' state of the art in robot control since it is a well understood and used approach due to its simplicity and real-time capabilities. Then, inverse dynamics control, which is a step forward in control with respect to inverse kinematics, was recalled for robotic manipulators, humanoid robots and computer animations. Finally, optimal control approaches, which constitute the most promising automatic motion generation methodologies, were briefly discussed.

Chapter 3

Inverse-Dynamics Whole Body Motion

Whole-body control of a humanoid robot is a challenging task due to the kinematic and dynamic complexity of the system. Humanoids are highly kinematically redundant robots with a non-trivial tree-like structure as well as an unstable nature due to the vertical position. At every moment, the robot must not only reproduce some task, but it should not fall while performing the task, keeping its dynamic balance. As described in the previous section, several methods aiming at solving these problems have been recently proposed and the research area is very active. The generic method proposed in this thesis is based on the inverse-dynamics model of the robot considering contact constraints and a set of tasks specified with a given priority. The resolution for each task is posed as a minimization problem, subject to several constraints that ensure dynamic feasibility, and uses a hierarchical scheme based on hierarchized QPs. Since the motion is generated using several prioritized tasks, and the dynamic model of the robot is taken into account, the term *Dynamic Stack of Tasks (SoT)* is sometimes used to refer to the framework. With respect to other operational-space inverse-dynamics (OSID) approaches, the advantages of the proposed method are the capability to handle both equality and inequality constraints at any level of the hierarchy, the fast computation time that allows it execution in real-time, and the direct feasibility of the generated motion, which does not need any further processing or validation before being executed on the robot.

This chapter presents the foundations of the proposed methodology and is divided as follows. First, the task-function formalism, which is the basic form in which the motion objectives are specified, is recalled. Then, the dynamic model of the robot is presented considering the underactuation of the system as well as the rigid contact constraints. The control scheme based on hierarchical quadratic programming is then treated in its initial formulation and also in the reduced form to speed up the computational time and achieve real-time performance. Finally, different types of tasks used throughout the thesis to generate different kinds of motion are presented.

3.1 Dynamic Considerations for a Humanoid Robot

The dynamic model of the robot states the relation between the generalized control torques τ and the generalized joint accelerations $\ddot{q}$. Unlike robotic manipulators, a humanoid robot is not attached to some part of the environment but presents a continuous interaction: the feet of the robot are continuously touching the ground, and other parts of the robot such as the hands, can also be in contact with the environment. Hence, the constraints imposed by the considered contact model must be satisfied for the motion to be feasible. Moreover, a humanoid robot is also an underactuated system, whose motion is possible due to the interaction with the environment, and balance must be kept throughout all the motion. Thus, the representation of the robot must explicit this underactuation and some balance constraints must also be satisfied. The full dynamic model includes both the contact constraints and the underactuation and is presented at the end of this section.

3.1.1 Rigid Contact Constraints

The rigid contact constraints are used to model contacts between rigid bodies. Let a rigid body B be in contact with the rigid environment E at n_c contact points, and let the reference frame attached to each contact point have its z axis pointing in the normal direction from the environment to the body, and its x, y axes forming a plane tangent to the contact. Let the position of the i^{th} contact point be represented by $x_{c_i} = (x_{c_i x}, y_{c_i y}, z_{c_i z}) \in \mathbb{R}^3$, and the contact force by $f_{c_i} = (f_{c_i x}, f_{c_i y}, f_{c_i z}) \in \mathbb{R}^3$. The analysis of the contact constraints can be divided in two parts: the normal components, and the tangential components.

Normal Constraints

These constraints establish the conditions that the normal acceleration and the normal force must satisfy to maintain the rigid bodies in contact. For notational convenience, let the perpendicular elements to the body B and the environment E be represented as $x_{c_i}^{\perp} = x_{c_i z}$ and $f_{c_i}^{\perp} = f_{c_i z}$. There exist two types of normal constraints: kinematic constrains and dynamic constraints; and they are summarized in the so-called complementarity conditions

a) Normal Kinematic Constraints Without considering impacts, and assuming finite forces that generate finite accelerations, a constraint for a contact to hold is that the normal velocity of the contact point must be equal to zero: $\dot{x}_{c_i}^{\perp} = 0$ at all times [Trinkle 97]. This holds true by the definition of a contact. The reason is that a negative perpendicular velocity, $\dot{x}_{c_i}^{\perp} < 0$, would indicate that the body will penetrate the environment in the near future, and a positive velocity, $\dot{x}_{c_i}^{\perp} > 0$, would indicate that the body comes from having penetrated the environment, both of which are not possible. At the acceleration level, to avoid penetration of the rigid body into the environment, $\ddot{x}_{c_i}^{\perp}$ must be non-negative which leads to two possibilities:

(i) The contact is maintained if the normal acceleration is null: $\ddot{x}_{c_i}^{\perp} = 0$.

(ii) The contact breaks if the normal acceleration is positive: $\ddot{x}_{c_i}^{\perp} > 0$.

b) Normal Dynamic Constraints With the same assumptions as for the kinematic constraints, the dynamic constraint imposes that the normal contact force to avoid interpenetration be non-negative $f_{c_i}^{\perp} \geq 0$. The reason for this unilateral force is that the body B can only push on the environment, but it cannot attract it, hence the force is possible only in one direction. The two possibilities for the compressive contact force are [Trinkle 97]:

(i) The contact is maintained if the normal contact force is non-negative: $f_{c_i}^{\perp} > 0$.

(ii) The contact breaks if the normal contact force is zero: $f_{c_i}^{\perp} = 0$.

c) Signorini Conditions The dynamic and the kinematic constraints that avoid interpenetration can be combined to incorporate at the same time the normal force and acceleration. The contact persists if the normal acceleration is null and the normal force is non-negative, that is: $\ddot{x}_{c_i}^{\perp} = 0$ and $f_{c_i}^{\perp} \geq 0$. On the other hand, both rigid bodies separate from each other if the normal force is null, in which case the normal acceleration is non-negative; that is: $\ddot{x}_{c_i}^{\perp} \geq 0$ and $f_{c_i}^{\perp} = 0$. Both constraints to avoid interpenetration are usually written as complementarity conditions usually known in mechanics as the *Signorini conditions* or the *non-penetration conditions* which state the following [Klarbring 97]:

$$\ddot{x}_{c_i}^{\perp} \geq 0, \qquad f_{c_i}^{\perp} \geq 0 \qquad \text{and} \qquad \ddot{x}_{c_i}^{\perp} f_{c_i}^{\perp} = 0. \tag{3.1}$$

As stated before, the only case that ensures persistence of a contact in (3.1) is $\ddot{x}_{c_i}^{\perp} = 0$ and $f_{c_i}^{\perp} \geq 0$. For a robot, the velocity of the contact point can be related to the robot kinematics through the Jacobian of the contact point, called the *contact Jacobian*, as

$$\dot{x}_{c_i} = J_{c_i}\dot{q} \qquad \text{where} \qquad J_{c_i} = \frac{\partial x_{c_i}}{\partial q} \tag{3.2}$$

Consequently, the acceleration of the contact point is $\ddot{x}_{c_i} = J_{c_i}\ddot{q} + \dot{J}_{c_i}\dot{q}$. Using the Jacobian, the non-holonomic condition to keep a rigid contact [Yamane 03] can be written as

$$J_{c_i}\ddot{q} + \dot{J}_{c_i}\dot{q} = 0 \tag{3.3}$$

and the condition on the normal force is

$$f_{c_i}^{\perp} \geq 0. \tag{3.4}$$

Friction Constraints

These constraints establish the conditions for the tangential forces according to the friction model to be used. Two main friction models are used in robotics: the point contact model, and the friction cone model. However, instead of using the whole friction cone, the linearized version of it is typically utilized due to its computational advantages.

a) Point Contact Model In this model, the contact force is assumed to be completely directed in the normal direction from the environment to the body so that the tangential components are neglected; that is, $f_{c_i} = (0, 0, f_{c_i}^{\perp})$. This model is typically used when the friction between the bodies (for instance, the foot and the ground) is considered high enough as to neglect its effects. Since only the normal components are used [Featherstone 08], the

model constraints are given by (3.3) and (3.4). In robotics, the contact force f_{c_i} acting at a rigid contact point generates an effect on the joint torques τ_{c_i}, and the virtual work principle states that $\tau_{c_i}^T \delta q = f_{c_i}^T \delta x_{c_i}$, where δq is the virtual motion of the joint, and δx_{c_i} is the virtual displacement at the contact point. Using the contact Jacobian (3.2) for infinitesimal motion, the equality $\delta x_{c_i} = J_{c_i} \delta q$ holds leading to

$$\tau_{c_i} = J_{c_i}^T f_{c_i} \tag{3.5}$$

which is the linear relation between the contact force and its effect on the generalized joint torques.

b) Friction Cone Model When friction effects cannot be neglected, a *friction cone model* based on Coulomb's friction law has to be considered for the tangential components of the contact force f_{c_i}. This model constrains the magnitude and the direction of the contact force to lie inside a cone, referred to as the *friction cone* K_i, and defined as

$$K_i = \{(f_x, f_y, f_z) \in \mathbb{R}^3 \ / \ \|(f_x, f_y)\|_2 \leq \mu f_z\} \quad \text{so that} \quad f_{c_i} \in K_i \tag{3.6}$$

where $\mu > 0$ is the friction coefficient at the contact point. From a geometric point of view, the forces f_c lie in the α-plane of the Klein quadric corresponding to the point of contact; equivalently, the interior of the friction cone corresponds to the interior of a circle in the α-plane [Selig 07]. Since forces must be non tensile, the non-negativeness of $f_{c_i}^{\perp}$ must also be hold. In fact, both normal conditions (3.3) and (3.4) must be satisfied in addition to (3.6), and the effect of the force on the robot joints is (3.5).
It should be noted from (3.6) that when there is a high friction coefficient μ, the tangential components of the contact force become very small with respect to the normal components. In these cases, the rigid contact point can be used as an approximation to the friction cone model.

c) Linearized friction cone For computational facility, the friction cone can be linearized using a polygonal approximation with n_f facets . For the i^{th} contact point, the linearization of K_i in (3.6) uses some linear basis $V_i = [v_1, \cdots, v_{n_f}] \in \mathbb{R}^{3 \times n_f}$, where $v_j \in \mathbb{R}^3$ are the edges of the approximated polygon, and V_i is sometimes referred to as the *generator* of the linearized cone. The contact force can thus be represented as the conic combination of the generators v_j as

$$f_{c_i} = \sum_{j=1}^{n_f} l_j v_j = V_i \lambda_i \tag{3.7}$$

where $l_j \in \mathbb{R}$, $l_j \geq 0, \forall j = 1, \cdots, n_f$, and $\lambda_i = (l_1, \cdots, l_{n_f}) \in \mathbb{R}^{n_f}$. Thus, the constraint for the contact force f_{c_i} to belong to the linearized cone is

$$\lambda_i \geq 0 \tag{3.8}$$

which can be seen as a generalization of (3.4) when friction is taken into account. The kinematic condition on the contact acceleration (3.3) remains valid. In this model, the effect of the contact force on the joint torques can be obtained from (3.5) replacing the force by its expression as a function of the linearized cone (3.7), and is given by

$$\tau_{c_i} = J_{c_i}^T V_i \lambda_i. \tag{3.9}$$

3.1.2 The Zero-Moment Point (ZMP)

The *ZMP*, first introduced in [Vukobratovic 69], is the point $r_z = (x_z, y_z, z_z) \in \mathbb{R}^3$ on the support polygon where the influence of all forces acting on the mechanism can be replaced by one single force [Vukobratović 04]. The wrist $\phi_{zmp} = (\tau_{zmp}, f_{zmp}) \in se^*(3)$ at r_{zmp} generated by the reaction force and the reaction torque satisfies

$$\tau_{zmp_x} = 0 \qquad \text{and} \qquad \tau_{zmp_y} = 0 \tag{3.10}$$

A biped robot is dynamically balanced as long as the ZMP exists, and for a dynamically balanced gait on a horizontal surface, the ZMP always coincides with the *Center of Pressure* (CoP) [Sardain 04]. It is sometimes said that the ZMP must lie inside the support polygon for the robot to be dynamically stable; however, the ZMP is only defined inside the support polygon and when the conditions (3.10) are satisfied outside, the point should be called the *FZMP* (fictitious ZMP) [Vukobratović 04] or the *FRI* (Foot Rotation Indicator) [Goswami 99].

Computation based on the dynamics Let the robot be modeled as a multi-body system composed of n_b rigid bodies, each with a mass m_i. The ZMP can be obtained using the basic laws of mechanics for this robot model as [Wieber 02]:

$$r_z = \frac{\hat{n} \times f_r + r_c g \sum_{i=1}^{n_b} m_i}{n \cdot \tau_r + g \sum_{i=1}^{n_b} m_i} \tag{3.11}$$

where f_r, τ_r are the linear and rotational components of the robot dynamic wrench, n is a vector orthogonal to the plane where the contact lies ($\hat{n}$ is its skew symmetric matrix), r_c is the robot CoM, and g is the gravitational constant. The robot dynamic wrench $\phi_r \in se^*(3)$ in (3.11) is

$$\phi_r = \begin{bmatrix} \tau_r \\ f_r \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n_b} m_i \hat{r}_i \ddot{r}_i + R_i I_i \dot{\omega}_i + R_i \hat{\omega}_i I_i \omega_i \\ \sum_{i=1}^{n_b} m_i \ddot{r}_i \end{bmatrix}$$

where r_i is the CoM position of the i^{th} body, R_i its rotation matrix, I_i its inertia matrix, and ω_i its angular speed. The hat denotes the corresponding skew symmetric matrix related to the cross product. However, the usage of the full model in practice is not frequent due the high computational requirements. Simplified models are typically used [Kajita 05], and a particularly common one for dynamic walking is based on the Linear Inverted Pendulum [Kajita 01] (see Appendix B).

Computation based on the contact forces The ZMP can also be computed using the punctual contact forces $f_{c_i} \in \mathbb{R}^3$ at the interface between the supporting foot and the ground, since it is equivalent to the Center of Pressure [Goswami 99]. Moreover, only the forces at the vertices of the contact convex hull are needed, and only the component perpendicular to the ground $f_{c_i}^{\perp}$ (which is typically the z component) is used. Assuming n_c contact points acting at $r_i = (x_i, y_i, z_i)$, the ZMP is computed as

$$r_z = \left(\frac{\sum_i^{n_c} x_i f_{c_i}^{\perp}}{\sum_i^{n_c} f_{c_i}^{\perp}}, \frac{\sum_i^{n_c} y_i f_{c_i}^{\perp}}{\sum_i^{n_c} f_{c_i}^{\perp}}, 0 \right) \tag{3.12}$$

which is the average of the contact points p_i weighted by the normal component of the contact forces at each point.

ZMP Measurement Although the ZMP can be obtained using the previous methods, humanoid robots typically do not possess the sensors to directly measure the variables in (3.11) or all the forces in (3.12), but wrenches can be measured at certain predefined positions. Let the n_s wrenches at some point $r_i = (x_i, y_i, z_i) \in \mathbb{R}^3$ in the foot be $\phi_{c_i} = (\tau_{c_i}, f_{c_i}) \in se^*(3)$. The relation between these wrenches at the ZMP wrench is given by

$$\phi_{zmp} = \sum_{i=1}^{n_s} \mathrm{Ad}_{s_i}^T \phi_i \tag{3.13}$$

where the adjoint matrix is

$$\mathrm{Ad}_{s_i}^T = \begin{bmatrix} R_{s_i} & \hat{r}_{iz} R_{s_i} \\ 0 & R_{s_i} \end{bmatrix},$$

the matrix R_{s_i} represents the rotation from each wrench ϕ_{c_i} frame to the reference frame, and $\hat{r}_{iz}$ is the skew-symmetric matrix corresponding to $r_i - r_z$. Without loss of generality, R_{s_i} will be assumed to be the identity, since the transformation between the wrench frame and the reference frame can be known a-priori, and thus the wrench can be already expressed in the reference frame. Using $R_{s_i} = I$, the torque component of (3.13) can be explicitly written as:

$$\begin{bmatrix} \tau_{zmp_x} \\ \tau_{zmp_y} \\ \tau_{zmp_z} \end{bmatrix} = \sum_{i=1}^{n_s} \left\{ \begin{bmatrix} 0 & z_z - z_i & y_i - y_z \\ z_i - z_z & 0 & x_z - x_i \\ y_z - y_i & x_i - x_z & 0 \end{bmatrix} \begin{bmatrix} f_{c_{i_x}} \\ f_{c_{i_y}} \\ f_{c_{i_z}} \end{bmatrix} + \begin{bmatrix} \tau_{c_{i_x}} \\ \tau_{c_{i_y}} \\ \tau_{c_{i_z}} \end{bmatrix} \right\} \tag{3.14}$$

and assuming the conditions (3.10) for the torques, the ZMP becomes

$$\begin{aligned} x_z &= \frac{\sum_{i=1}^{n_s} x_i f_{c_{i_z}} - (z_i - z_z) f_{c_{i_x}} - \tau_{c_{i_y}}}{\sum_{i=1}^{n_s} f_{c_{i_z}}} \\ y_z &= \frac{\sum_{i=1}^{n_s} y_i f_{c_{i_z}} - (z_i - z_z) f_{c_{i_y}} - \tau_{c_{i_x}}}{\sum_{i=1}^{n_s} f_{c_{i_z}}} \end{aligned} \tag{3.15}$$

which can be used to measure the ZMP. For example, the HRP-2 robot has one 6-D axis force/torque sensor close to its ankle, and the ZMP can be directly measured using it with $n_s = 1$ in (3.15).

3.1.3 Dynamic Model of a Robot

The dynamic model of a generic fully actuated robot can be found using the Lagrange formalism, the Newton-Euler equations or the Hamiltonian approach. For a compact analytical representation, the Lagrange formalism is usually preferred and is presented here.

Moment of Inertia Operator Let $\xi_i = (\omega_i, v_i) \in se(3)$ be the twist of the i^{th} body composing the robot, where $se(3)$ is the Lie algebra of $SE(3)$, and let the $\mu_i = (h_i, l_i) \in se^*(3)$ be its momentum, where h_i and l_i are the angular and linear momenta, respectively, and $se^*(3)$ is the dual of $se(3)$. The dynamics of this i^{th} rigid body is [Selig 07]:

$$\begin{aligned} h_i &= I_i \omega_i + m_i (r_{c_i} \times v_i) \\ l_i &= m_i v_i + m_i (\omega_i \times r_{c_i}) \end{aligned}$$

where $r_{c_i} \in \mathbb{R}^3$ is the CoM, m_i its mass, and $I_i \in \mathbb{R}^{3\times3}$ its inertia tensor. Letting $\hat{r}_{c_i}$ represent the skew-symmetric matrix of r_{c_i}, these motion equations can be compactly written as

$$\mu_i = N_i \xi_i$$

where

$$N_i = \begin{bmatrix} I_i & m_i \hat{r}_{c_i} \\ m_i \hat{r}_{c_i}^T & m_i I_3 \end{bmatrix} \in \mathbb{R}^{6\times6}$$

is a symmetric matrix called the *moment of inertia operator* (or tensor) and provides a linear isomorphism $N_i : se(3) \to se^*(3)$. Equivalently, N_i is an operator that converts velocities to momenta [Arnol'd 89]. To simplify this matrix, it is common practice to choose the origin of the body to coincide with its center of mass, which makes N_i a block diagonal matrix.

Kinetic Energy For the i^{th} rigid body, the moment of inertia operator N_i determines a bilinear symmetric form on $se(3)$, called the *kinetic energy* and given by [Arnol'd 89]:

$$K_i = \frac{1}{2} \xi_i^T N_i \xi_i$$

which is always positive definite. The kinetic energy for the robot is the sum of the kinetic energies for the n_b rigid bodies constituting it, and is given by

$$K(q, \dot{q}) = \frac{1}{2} \sum_{i=1}^{n_b} \xi_i^T N \xi_i = \frac{1}{2} \sum_{i=1}^{n_b} \dot{q}^T J_{o_i}^T N J_{o_i} \dot{q} \tag{3.16}$$

since $\xi_i = J_{o_i}(q)\dot{q}$. The robot *generalized mass matrix*, also called the generalized inertia matrix or the kinetic energy matrix, is defined as

$$M(q) = \sum_{i=1}^{n_b} J_{o_i}^T N J_{o_i}$$

which can be shown to be symmetric ($M(q) = M(q)^T$) and positive definite ($M(q) \succ 0$[1]) [Murray 94]. Replacing this matrix in (3.16), the kinetic energy of the robot is

$$K(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} \tag{3.17}$$

Potential Energy The gravitational *potential energy* of a rigid body is the energy accumulated when it gains height, which depends on a frame of reference. For a single body of the robot it is given by $U_i = m_i g_i h_i(q)$, where g is the acceleration of gravity and $h_i(q)$ is the height of the body with respect to a given reference frame. The potential energy of the robot is the sum of the potential energies of the constituting bodies and is

$$U(q) = g \sum_{i=1}^{n_b} m_i h_i(q) = g m^T h(q) \tag{3.18}$$

where $m = (m_1, \cdots, m_n)$ and $h(q) = (h_1(q), \cdots, h_n(q))$.

[1] For a square matrix $A \in \mathbb{R}^{s\times s}$, the notation $A \succ 0$ states that $\forall z \in \mathbb{R}^s$, $z^T A z > 0$.

Lagrange formulation The natural *Lagrangian function* is the difference between kinetic and potential energies: $L(q, \dot{q}) = K(q, \dot{q}) - U(q)$. The dynamic model of a robot can be obtained using the *Euler-Lagrange* (sometimes only called Lagrange) equation:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right)^T - \left(\frac{\partial L}{\partial q}\right)^T = \Gamma \tag{3.19}$$

where Γ is the vector of the generalized forces acting on the joints. Replacing (3.17) and (3.18) in the Lagrangian $L(q, \dot{q})$, the derivatives in (3.19) become

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right)^T = \frac{d}{dt}\left(\dot{q}^T M(q)\right)^T = M(q)\ddot{q} + \dot{M}(q)\dot{q}$$
$$\left(\frac{\partial L}{\partial q}\right)^T = \frac{1}{2}\left(\dot{q}^T \frac{\partial M(q)}{\partial q}\dot{q}\right)^T - \left(\frac{\partial U(q)}{\partial q}\right)^T$$

and substituting these elements in (3.19), the dynamic model of the robot is

$$M(q)\ddot{q} + b(q, \dot{q}) = \Gamma \tag{3.20}$$

where

$$b(q, \dot{q}) = \left(\dot{M}(q) - \frac{1}{2}\dot{q}^T\left(\frac{\partial M(q)}{\partial q}\right)^T\right)\dot{q} + \left(\frac{\partial U(q)}{\partial q}\right)^T$$

is a vector comprising the Coreolis and centrifugal forces as well as the gravity effects. This expression for $b(q, \dot{q})$ can be further decomposed to explicitly show each of the components [Siciliano 09].

3.1.4 Representation of a Humanoid Robot

One of the main differences between a humanoid robot and a robotic manipulator is that the former can freely move in its environment while the latter presents a fixed attachment to the environment. But the ability to freely move also generates a big control challenge known as *underactuation*: the motion within the environment cannot be directly controlled but is indirectly commanded through the proper motion of the actuated joints [Wieber 06a]. Thus the representation of the robot should consider this underactuation besides the joint coordinates.

Actuated Joints The actuated joints of a robot are typically composed of two types: revolute and prismatic joints. A *revolute joint* takes values in $\mathbb{S}^1$, where $\mathbb{S}^1$ is the 1-sphere manifold corresponding to the unit circle. A set of r revolute joints takes values in an r-torus $\mathbb{T}^r$ defined as $\mathbb{T}^r = \mathbb{S}^1 \times \cdots \times \mathbb{S}^1$. *Prismatic joints* take values in $\mathbb{R}$ along a certain motion axis, and a set of p prismatic joints generate the $\mathbb{R}^p$ space. Thus, for a robot with $n = r + p$ degrees of freedom, with r revolute joints and p prismatic joints, the configuration space is the $r + p$ dimensional manifold $\mathcal{Q}$ that results from the Cartesian product of all these joint spaces; that is, $\mathcal{Q} = \mathbb{T}^r \times \mathbb{R}^p$. A particular configuration of the actuated joints is represented as $q_a \in \mathcal{Q}$.

Free-Floating Base The pose of the robot in the environment is associated with the pose of some body composing the robot, which is called the *free-floating* base, the *root* base, or the *free-flyer*. For a humanoid robot, this body is typically the torso or the waist. The free-floating pose is represented by

$$x_f = \begin{bmatrix} x_{f_p} \\ x_{f_r} \end{bmatrix} \in SE(3) \tag{3.21}$$

where x_{f_p} is the position of the base, usually given by Cartesian coordinates ($x_{f_p} \in \mathbb{R}^3$), and x_{f_r} is its orientation given by some parametrization of the $SO(3)$ group. Typical orientation parameterizations are quaternions (also called Euler parameters), direction cosines, Euler angles, and the axis-angle representation. In general, x_f is a parametrization of the $SE(3)$ group and presents 6 DoF in the space, but it can contain more than 6 coordinates depending on the representation that is used. The twist of the base will be denoted by $\xi_f = (\omega_f, v_f) \in se(3)$, where ω_f, v_f are the angular and linear velocities respectively.

Generalized Coordinates The *generalized coordinates* q represent the full robot state and contain the underactuated base as well as the actuated joints. These coordinates q as well as the vector used to represent their velocity $\dot{q}$ are

$$q = \begin{bmatrix} x_f \\ q_a \end{bmatrix} \qquad \text{and} \qquad \dot{q} = \begin{bmatrix} \xi_f \\ \dot{q}_a \end{bmatrix}.$$

It is common practice to use the twist instead of the derivatives of the pose representation since the twist is independent of the parameterization of $SE(3)$. The relation between the derivative of the base pose $\dot{x}_f$ and its twist is a linear relation $\dot{x}_f = E_o(x_f)\ \xi_f$, where $E_o(x_f)$ changes according to the parameterization used for x_f [Khatib 80, Khatib 04a].

Selection Matrix The actuated joint configuration q_a can be recovered from the generalized coordinates q using the so-called *selection matrix* S that selects only the actuated part. For a robot with a b-dimensional representation of its free-floating base and with n actuated DoF, the matrix $S \in \mathbb{R}^{n \times (b+n)}$ is

$$S = \begin{bmatrix} \bar{0} & I \end{bmatrix} \qquad \text{such that} \qquad q_a = Sq$$

where $\bar{0} \in \mathbb{R}^{n \times b}$ is a zero matrix and $I \in \mathbb{R}^{n \times n}$ is an identity matrix. The selection matrix is also important for mapping the generalized torques τ acting on the actuated joints to the generalized coordinates as $S^T\tau = [\bar{0}\ \tau]^T$, where 0 corresponds to the underactuation of the system, and arises from the fact that the base x_f cannot be controlled.

3.1.5 Dynamic Model of a Humanoid Robot

The dynamic model of a robotic manipulator is given by (3.20), where a full actuation by the torques τ acting on the system is assumed. However, a humanoid robot presents two main differences with respect to a manipulator: it is an underactuated system, and its motion in the environment is done through contacts with the environment (usually both feet are in contact with the ground). The underactuation can be modeled using the generalized coordinates and the selection matrix presented in Section 3.1.4; and the environment interaction can be modeled with the virtual principle for the contact forces, as described in Section 3.1.1.

Multiple contact forces When there is more than a single contact point, all the contact forces corresponding to the contact points must be included in the model. Let the robot present n_c contact points x_{c_i} $(i = 1, \cdots, n_c)$ with the environment, let $J_{c_i}(q) = \frac{\partial x_{c_i}}{\partial q}$ be the contact Jacobian associated with the i^{th} contact point, and let f_{c_i} be the force exerted at x_{c_i}. Then, the full contact force and the corresponding full Jacobian can be represented by stacking each component as

$$f_c = \begin{bmatrix} f_{c_1} \\ \vdots \\ f_{c_{n_c}} \end{bmatrix} \qquad \text{and} \qquad J_c(q) = \begin{bmatrix} J_{c_1}(q) \\ \vdots \\ J_{c_{n_c}}(q) \end{bmatrix}.$$

Using (3.5), or alternatively (3.7), for each contact force, the effect of all the punctual contact forces on the joint torques can be compactly described as

$$\tau_c = \sum_{i=1}^{n_c} J_{c_i}^T(q) f_{c_i} = J_c^T(q) f_c \tag{3.22}$$

which has to be included in the dynamic model as an external generalized force. A similar expression can be obtained if the contact wrenches are used instead of the punctual contact forces.

Dynamic Model A humanoid robot can be modeled as an underactuated kinematic-tree chain composed of rigid bodies with a free floating base and subject to contact forces. The Euler-Lagrange formalism allows the dynamic representation of the robot as (3.20), which is generic in the sense that Γ contains all the generalized forces acting on the joints. For an underactuated robot in rigid contact with the environment, the effect of the contact forces τ_c (3.22) as well as the actuated joints $S^T\tau$ has to be included so that $\Gamma = S^T\tau - \tau_c$. The choice of the contact forces effect sign is arbitrary and depends on the chosen frame. With these considerations, the dynamic model of a humanoid robot is given by:

$$M(q)\ddot{q} + b(q, \dot{q}) + J_c(q)^T f_c = S^T\tau \tag{3.23}$$

As in (3.20), $M(q)$ is the generalized inertia matrix, and b is the term comprising the Coriolis and centrifugal forces as well as the gravity force vector. It is also possible to decouple the underactuated and actuated parts of the dynamics obtaining

$$M_u(q)\ddot{q} + b_u(q, \dot{q}) + J_{c_u}(q)^T f_c = 0 \tag{3.24}$$

$$M_a(q)\ddot{q} + b_a(q, \dot{q}) + J_{c_a}(q)^T f_c = \tau \tag{3.25}$$

where the matrices $M(q)$, $b(q, \dot{q})$ and $J_c(q)$ have been properly decomposed to show the decoupling. It can be shown that (3.25), the underactuated part, corresponds to the Newton-Euler equations of the whole system and expresses the change of momentum of the robot as a function of only the external forces [Wieber 06a].

The dynamic model (3.23) defines a surjective map from the space of joint torques and contact forces to the joint acceleration space. It is nonlinear but it is generally linearized around the current state $(q, \dot{q})$, assuming small variations, thus $M(q)$, $b(q, \dot{q})$, and $J_c(q)$ are assumed to be known. One fact is worth remarking: in the model, τ and f_c are sufficient to

describe the system dynamics; however, the robot motion is represented as a function of $\ddot{q}$, which has thus to be included in the computations. This fact allows for the decoupling of motion and actuation, described in Section 3.3.4. To handle the complementarity conditions at every contact point, (3.3) and (3.4) have to be additionally considered as constraints for a rigid contact point model, or (3.3) and (3.8) for a friction cone model.

3.2 Task Function Approach

The *task function* approach [Samson 91] is a framework that describes tasks in terms of specific output functions chosen to ease the observation and control of the task to be performed. In the field of robotics it is also referred to as the *operational space* approach [Khatib 87] and consists in generating motion based on the definition of different tasks whose control laws are expressed in a subspace of smaller dimension than the full robot state space. Then, these laws are back-projected onto the original space. For instance, if currently the robot arm is down and the task consists in raising the arm, the control law can be defined in terms of the spatial initial and final pose (position and orientation) of some part of the arm, like the hand, instead of the whole joint space. The effect of this approach is to reduce the control complexity, to make the control more intuitive, and to decouple the different parts of the motion making a clear separation of the objectives to be achieved.

3.2.1 Generic Formulation

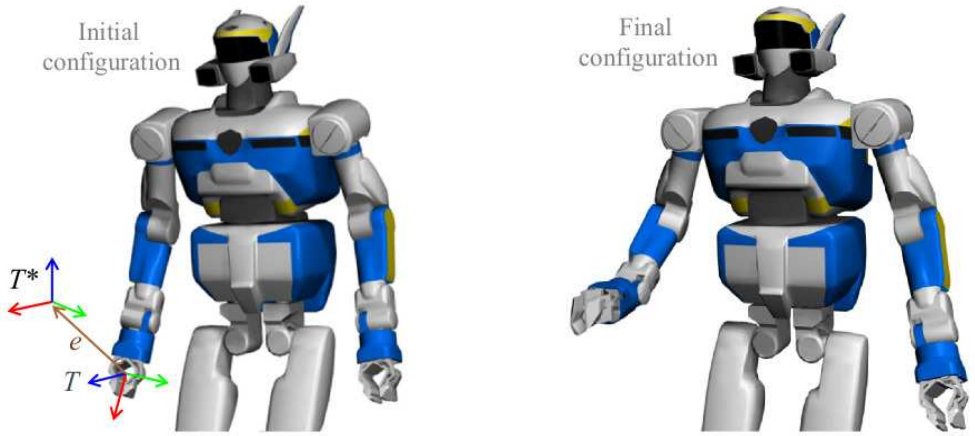
Let n be the total number of DoF of the system, $\mathcal{C}$ be an n-dimensional manifold representing the configuration space, $\mathcal{M}$ be an m-dimensional manifold representing the task space, and $\mathcal{U}$ be a p-dimensional manifold representing the control space. To fully specify a task, the following three components must be defined[2]:

1. The *task function* $e : \mathcal{C} \to \mathcal{M}$, which describes the task itself.
2. The *reference behavior* $r^* \in \mathcal{R}$, which is defined as an arbitrary desired behavior to properly achieve the task. In general, $\mathcal{R}$ is an m-dimensional manifold related to $\mathcal{M}$.
3. The *map* $G : \mathcal{U} \to \mathcal{R}$, which relates the task, in a given vector field $r \in \mathcal{R}$, to the control input $u \in \mathcal{U}$ as
$$r + \delta = Gu \tag{3.26}$$
where δ represents a generic, and sometimes unavoidable, drift of the task and belongs to the same space as r.

An example of a task function is given in Fig. 3.1. Using (3.26), the generic control law u^* that achieves a desired reference behavior r^* can be expressed as the solution to the following unconstrained minimization problem:

$$\min_u \|r^* + \delta - Gu\|_2^2 \tag{3.27}$$

[2]The generic formulation and cases for kinematics and dynamics appear in: L. Saab, O. Ramos, F. Keith, N. Mansard, P. Souères, J-Y. Fourquet, *Dynamic Whole-Body Motion Generation under Rigid Contacts and other Unilateral Constraints*, IEEE Transactions on Robotics (T-RO), Vol.29 N.2, pag. 346 - 362, April 2013 [Saab 13].



(a) Initial configuration **(b)** Final configuration

Figure 3.1 – Example of a task function. T represents the initial pose of the right hand, and T^* its desired pose. The task function in this case is $e = T \ominus T^*$, where $\ominus$ is a difference operator in SE(3).

where $\|.\|_2$ represents the L^2 norm. The solution to the minimization, and thus the desired control law, is obtained as [Saab 13]:

$$u^* = G^{\#}(r^* + \delta) + P_G \; u_2 \tag{3.28}$$

where $\{.\}^{\#}$ represents any reflexive generalized inverse, $P_G = I - G^{\#}G$ is the projector onto the nullspace of G corresponding to $G^{\#}$, and $u_2 \in \mathcal{U}$ is a secondary control input that can be used to exploit the redundancy of the system with respect to the task. The minimum norm solution is achieved if besides the minimization of (3.27), the norm of u is also minimized, and is given by $u^* = G^{\#}(r^* + \delta)$ which, as (3.28) shows, represents the special case when u_2 is null. Although the minimum norm solution might be efficient with respect to the control law, it can only be used to perform a single task, wasting the redundancy of the system for achieving more tasks. Amongst the possible reflexive generalized inverses, the Moore-Penrose pseudo-inverse is generally chosen, but other inverses, like a weighted inverse whose weights depend on some other criteria, can also be used (Appendix A presents more details on generalized inverses).

Without loss of generality, it is usual and fairly natural to define the control objective as a regulation to zero of the task function. This regulation approach is widely used in control theory. More concretely, given any observable $s \in \mathcal{M}$ willing to reach a desired $s^* \in \mathcal{M}$, the task function can be specified as

$$e = s - s^* \tag{3.29}$$

which must be regulated to zero. The desired s^* can be a fixed reference or a desired trajectory, in which case the task becomes a tracking problem. The regulation problem in (3.29) is controlled through the specification of the proper reference behavior r^*.

3.2.2 Inverse Kinematics Case

The previously described generic formulation for the task function approach (Section 3.2.1) can be directly used for inverse kinematics control. Let $\mathcal{Q}$ be the n-dimensional manifold that

represents the generalized joint space and let $q \in \mathcal{Q}$ be a particular joint configuration. As recalled in Section 2.2.2, differential kinematics linearly relates the rate of change of the task space to the rate of change of the joint space as

$$\dot{e}(q, \dot{q}) = J(q)\ \dot{q} \tag{3.30}$$

where $\dot{e}(q, \dot{q})$ is the time derivative of some task $e \in \mathcal{M}$, and the differential map $J(q) = \frac{\partial e}{\partial q} \in \mathbb{R}^{m \times n}$ is the task Jacobian. It is straightforward to see that (3.30) directly matches the pattern given by (3.26) when $\delta = 0$. Thus, the elements of the task function approach can be identified in the following straightforward way: (i) the task function is directly the task e; (ii) the reference behavior is the desired velocity for the task: $r^* = \dot{e}^*$; (iii) the map is given by the task Jacobian: $G = J(q)$.

For inverse kinematics, the control is directly given by the joint velocity so that $\mathcal{U} = T_p(\mathcal{Q})$, or equivalently, $u = \dot{q}$. Then, for a given task, the control law based on joint velocities can be expressed according to (3.28) as:

$$\dot{q}^* = J(q)^{\#}\dot{e}^* + P_J\ \overline{\dot{q}}$$

where P_J is a projector onto the nullspace of $J(q)$, and $\overline{\dot{q}} \in \mathbb{R}^n$ is an arbitrary vector that can be used to realize other tasks without interfering with the highest priority task e. When a regulation task is used, the reference behavior is usually chosen as $\dot{e}^* = -\lambda e$, where $\lambda > 0$ is a constant gain, to achieve asymptotic stability, in particular, an exponential convergence to zero. If the gain λ is increased, the task will be achieved faster; but the mechanical constraints of the robot forbid the usage of very high gains, which can generate undesired saturations.

3.2.3 Inverse Dynamics Case

The task function approach can also be extended to inverse dynamics control [Khatib 87], but the formulation is not as straightforward as for inverse kinematics. Consider the generic dynamic model of a robot without contact constraints given by (3.20). In inverse dynamics the control is done through the generalized torques τ, and the joint configuration is described by the second order joint acceleration $\ddot{q}$. Thus, to use the task function approach, the relationship between the task acceleration $\ddot{e}$ and the joint acceleration $\ddot{q}$ has to be considered. This relation can be found by taking the derivative of (3.30) and is equal to

$$\ddot{e}(q, \dot{q}, \ddot{q}) = \dot{J}(q, \dot{q})\dot{q} + J(q)\ddot{q}. \tag{3.31}$$

For notation simplicity, the dependences on q and its derivatives will be dropped whenever it is clear. Replacing $J(q)\ddot{q}$ from (3.31) in (3.20), the expression for inverse dynamics directly relating the task space to the control can be found as:

$$\begin{aligned} JM^{-1}M\ddot{q} + JM^{-1}b &= JM^{-1}\tau \\ \ddot{e} + (JM^{-1}b - \dot{J}\dot{q}) &= JM^{-1}\tau \end{aligned} \tag{3.32}$$

It can be observed that (3.32) matches the generic task function pattern (3.26) if the dynamic drift is $\delta = JM^{-1}b - \dot{J}\dot{q}$, and the control input is the generalized actuation vector: $u = \tau$. Thus, for the task function approach formalism, the following terms can be identified: (i) the task function is the task itself: e; (ii) the reference behavior is the desired task acceleration

$r^* = \ddot{e}^*$; (iii) the map is $G = JM^{-1}$, which is a function of the kinematic differential map (the Jacobian).

The inverse dynamics control is then reduced to finding the desired torque control input τ^* that will generate the reference behavior $\ddot{e}^*$ using any necessary joint accelerations $\ddot{q}$. Thus, for inverse dynamics, the control law in (3.28) becomes

$$\tau^* = (JM^{-1})^{\#}(\ddot{e}^* + JM^{-1}b - \dot{J}\dot{q}) + P_{JM^{-1}}\overline{\tau}$$

where $P_{JM^{-1}}$ is a projector onto the nullspace of JM^{-1} and $\overline{\tau}$ is an arbitrary vector that can be used to control more tasks with lower priority. The model (3.20) can also be extended to the case of dynamics with rigid contact constraints and the application of the task-function approach leads to similar results [Saab 13].

3.3 Inverse Dynamics Control

The control of reactive humanoid robots in the real world has to deal with constraints imposed by the robot mechanical properties and by the environment. Traditional control based on inverse kinematics only considers kinematical constraints but it does not take into account any of the other constraints; therefore it is not sufficient and inverse dynamics has to be considered. Inverse dynamics control is able to consider the full dynamics of the robot as well as the environmental constraints and at the same time it can generate different types of motion[3].

3.3.1 Inverse Dynamics Problem

The inverse dynamics problem consists in finding a control law that will lead to the fulfillment of a desired task satisfying the inverse dynamics model of the robot and some additional constraints to ensure the feasibility of the motion. Given a certain task represented by $h(q, \dot{q}, \ddot{q}) = \ddot{e}^* - \ddot{e}(q, \dot{q}, \ddot{q})$, the inverse dynamics problem can be written in generic form as

$$\begin{aligned} \min_{\nu} \; & \|h(q, \dot{q}, \ddot{q})\|_N^2 \\ \text{s.t.} \quad & A\nu + \beta = 0 \\ & D\nu + \gamma \geq 0 \end{aligned} \tag{3.33}$$

where ν is the control variable, A, β are matrices defining suitable equality constraints typically given by the dynamic model (3.23) and the kinematic non-holonomic contact constraints (3.3), and D, γ are matrices defining suitable inequality constraints that might represent, for instance, the unidirectionality of the contact forces (3.4) or (3.8). The minimization is done using an N-norm, which can be chosen to control the efforts over the joints [Peters 08]. In inverse dynamics, the control variable ν must contain the generalized torques τ, but since motion is specified by terms depending on joints acceleration, $\ddot{q}$ can also be included, although both terms are naturally coupled. When the unidirectionality of the forces is considered, f_c

[3]The control framework in this section first appeared in: L. Saab, O. Ramos, N. Mansard, P. Souères, J-Y. Fourquet: *Generic Dynamic Motion Generation with Multiple Unilateral Constraints*, IEEE International Conference on Intelligent Robots and Systems (IROS), San Francisco, CA, USA, September 2011 [Saab 11].

can be included in the control variable ν, but also friction models can be used. Thus, this generic problem for a single task $\ddot{e}^*$ can be written as

$$\begin{aligned} \min_{\ddot{q},\tau,f_c} \quad & \|\ddot{e}^* - \ddot{e}(q,\dot{q},\ddot{q})\|_N^2 \\ \text{s.t.} \quad & M(q)\ddot{q} + b(q,\dot{q}) + J_c(q)^T f_c = S^T\tau \\ & J_c\ddot{q} + \dot{J}_c\dot{q} = 0 \\ & f_c^{\perp} \geq 0 \end{aligned} \tag{3.34}$$

There exist two main approaches to solve this problem. The first one aims at determining an analytical solution through successive projections onto the constraints nullspaces to achieve dynamic feasibility. The other one consists in finding the solution treating the problem as a purely optimization problem which allows for the consideration of more generic constraints.

Solutions based on nullspace projections One way to solve the inverse dynamics problem in (3.34) is to consider the constraints analytically and to use the so called projected inverse-dynamics to eliminate the contact forces. These approaches only consider the equality constraints, and inequality constraints cannot be arbitrarily handled. The usage of operational space controllers for whole-body control was proposed in [Sentis 05] where the solution is obtained using the so-called dynamically consistent Jacobian, which is a particular norm N for the analytical solution of (3.34). For multiple tasks, this approach uses a prioritization scheme based on projections onto the nullspace of higher priority tasks and constraints. Another approach is based on the orthogonal projections for inverse dynamics introduced in [Aghili 05] and was proposed for legged robots with switching contact constraints in [Mistry 10]. It finds an analytical solution for the torques projecting the robot dynamics onto a reduced dimensional space independent of contact forces using the aforementioned orthogonal decomposition and only kinematic projections. Moreover, both approaches have been shown to be equivalent with respect of different minimization costs [Righetti 11b].

The limitations of these methods are the lack of possibility to include inequality constraints such as joint limits, friction cones, amongst others. This leads to obtaining unbounded contact forces which in reality can produce balance problems for a robot, or unfeasible contacts. Also, tasks expressed as inequalities cannot be formulated.

Solutions based on optimization The other approach to solve the inverse dynamics problem is to use optimization techniques to find the solution to (3.34). A typical method is through the usage of Quadratic Programs (QP). The advantage over solutions based on nullspace projections is the ability to handle inequalities such as friction cones, force unidirectionality, joint limits or other constraints/tasks. There exist two main approaches in this category when there are more than two tasks to be satisfied at the same time: weighting based schemes and prioritized schemes.

Weighting-based schemes combine the tasks by assigning a weight α_i to each of them. For n_t tasks, the minimization objective of (3.34) becomes $\sum_{i=0}^{n_t} \alpha_i \|\ddot{e}_i^* - \ddot{e}_i\|^2$. An approach that is oriented towards robust balance and uses this scheme was proposed in [Collette 08]. For whole-body control, the QP formulation based on weighted sums subject to constraints was proposed in [Bouyarmane 11] and [Salini 11]. The problem with this formulation is the effort needed to properly tune the weights, and when tasks or constraints conflict with each other,

there can be problems in the solutions: none task is fully satisfied but a compromise amongst them is obtained.

To solve this problem, *prioritized schemes* treat the problem as a hierarchy which is an ordered list of tasks where lower priority tasks should not interfere with higher priority ones. In these cases, the inverse dynamics problem is recursively solved starting with the highest priority task and setting the solution of this as a constraint for the next levels. The approach in [deLasa 10] uses a nested sequence of objectives re-parameterized at each level of the hierarchy. An application on a real humanoid robot was implemented in [Herzog 13] using slacks in the QP to account for priority feasibility and reducing the computational cost by decomposing the equations of motion in the actuated and underactuated parts. Based on this reduction, a framework for this usage in an optimal control scheme outer loop was recently proposed in [Kuindersma 14]. An analytical sparse solution that converts the problem into two smaller independent unconstrained problems reducing the computational time has been proposed in [DelPrete 14a]. The approach used in this thesis is based on prioritization and for this sake, a cascade of QPs, referred to as hierarchical quadratic program, is used at each level of the hierarchy to handle inequalities at any priority level, which cannot be obtained with other approaches.

3.3.2 Hierarchical Quadratic Program (HQP)

The *Hierarchical Quadratic Program* solver [Escande 14] is an extension of the traditional *Quadratic Program* (QP) approach that consists in using a cascade of QPs so that equalities or inequalities, representing constraints, can be handled at any level of the cascade. The cascade enforces a hierarchy since the constraints corresponding to the first QPs in the cascade will have a higher priority than the constraints in the next QPs. A constraint with higher priority is interpreted as a constraint that is satisfied as closely as possible without considering any other existing (lower-priority) constraints.

Consider a generic prioritized linear system containing N linear equalities or inequalities of the following form

$$\underline{d}_i \leq A_i x \leq \overline{d}_i \tag{3.35}$$

where x is the optimization variable, A_i is a coefficients matrix, $\underline{d}_i$ is the lower bound, $\overline{d}_i$ is the upper bound, and $i = 1, \cdots, N$ represents the priority in the hierarchy (the lower the value of i, the higher the priority). Equalities are a special case of (3.35) when $\underline{d}_i = \overline{d}_i$, and single-sided inequalities are special cases when $\underline{d}_i = -\infty$ (leading to $A_i x \leq \overline{d}_i$) or $\overline{b}_i = +\infty$ (leading to $A_i x \geq \underline{d}_i$). An HQP works in the following way. The QP for the first priority component ($i = 1$) is written as the optimization problem

$$\begin{aligned} &\min_{w_1, x} \|w_1\|_2^2 \\ \text{s.t.} \quad &\underline{d}_1 \;\leq\; A_1 x + w_1 \;\leq\; \overline{d}_1 \end{aligned}$$

where w_1 is a *slack variable* that allows the system to have some freedom in the resolution in case the exact constraint is not satisfied. The solution to the first priority QP is noted as w_1^*, which indirectly represents the best fulfillment of the first constraint. Due to the nature of the optimization, no other solution lower than w_1^* can be found. However, if the system is redundant, there is a set of x that achieves the same optimal solution, and it is used to

allow for the resolution of the other lower priority constraints. The second constraint defines a second QP as the problem

$$\begin{aligned} & \min_{w_2,x} \|w_2\|_2^2 \\ \text{s.t.} \quad & \underline{d}_2 \leq A_2 x + w_2 \leq \overline{d}_2 \\ & \underline{d}_1 \leq A_1 x + w_1^* \leq \overline{d}_1 \end{aligned}$$

where w_2 is a slack variable. However, in this case, the minimization is only effectuated for the second constraint using w_2; the variable w_1^* corresponding to the first constraint is fixed and is the optimized value computed in the previous step. This ensures that the solution for the second constraint will not modify the optimal value computed for the first constraint, ensuring a hierarchy. Generalizing the same idea, the QP corresponding to the k-th level of the hierarchy can be written as

$$\begin{aligned} & \min_{w_k,x} \|w_k\|_2^2 \\ \text{s.t.} \quad & \underline{d}_k \leq A_k x + w_k \leq \overline{d}_k \\ & \underline{d}_{k-1}^+ \leq A_{k-1}^+ x + w_{k-1}^{+*} \leq \overline{d}_{k-1}^+ \end{aligned}$$

where $\underline{d}_{k-1}^+$, $\overline{d}_{k-1}^+$, A_{k-1}^+, and w_{k-1}^{+*} are augmented matrices given by

$$\underline{d}_{k-1}^+ = \begin{bmatrix} \underline{d}_{k-1} \\ \vdots \\ \underline{d}_1 \end{bmatrix} \qquad \overline{d}_{k-1}^+ = \begin{bmatrix} \overline{d}_{k-1} \\ \vdots \\ \overline{d}_1 \end{bmatrix} \qquad A_{k-1}^+ = \begin{bmatrix} A_{k-1} \\ \vdots \\ A_1 \end{bmatrix} \qquad w_{k-1}^{+*} = \begin{bmatrix} w_{k-1}^* \\ \vdots \\ w_1^* \end{bmatrix}$$

which represent the higher priority constraints, and optimal slack variables (solved in the previous steps), ranging from $i = 1$ to $i = k - 1$.

The previous generic scheme shows that the solution to a system with equalities and inequalities is found using an HQP in such a way that the fact that level $k - 1$ has higher priority over level k implies that level k should be fulfilled as close as possible but without interfering with the other higher priority levels: $k-1$, $k-2$, $\cdots$, 2, 1. To make the computation faster, the HQP is solved with a dedicated solver in two steps. First, a classical primal active set algorithm finds the active inequality constraints and turns them into equalities since the solution does not change after this reduction. Then, a hierarchized complete orthogonal decomposition is applied to the set of equalities [Escande 14].

For a system with N linear equalities or inequalities, the HQP hierarchy is usually represented using a lexicographic order as:

$$(i) \prec (ii) \prec (iii) \prec \cdots \prec (N-1) \prec (N) \tag{3.36}$$

where (i) has the highest priority, and (N) has the lowest priority.

3.3.3 Inverse Dynamics Stack of Tasks (SoT)

For a humanoid robot, different types of motion can be generated using the task function approach and satisfying some additional constraints. These constraints arise from the continuous contact of the robot with the environment and the fact that balance should be kept at

all instants of time. A robot that achieves a desired task but falls down is useless; thus, the dynamics of the robot must play a highly important role when generating feasible motion.

As stated in Section 3.2, a task can be expressed without loss of generality as the regulation task (3.29): $e = s(q) - s^*$. As stated in Section 3.2.3, the dynamic model of the robot uses joint accelerations, and thus, to use the task function approach, a relation at the acceleration level is needed for the control. For a generic task i, the second order differential relation between the task space and the joint space is given in a similar way to (3.31) by:

$$\ddot{e}_i = \dot{J}_i(q, \dot{q})\dot{q} + J_i(q)\ddot{q} \tag{3.37}$$

where $J_i(q) = \frac{\partial e_i}{\partial q}$ is the Jacobian of the i^{th} task. Motion for the robot is generated using several instances of (3.37), each specifying an objective, with different priorities depending on the importance of the task. In fact, the several tasks make the system act like a *Stack of Tasks* (SoT), and therefore the name. Moreover, tasks can be added or removed at will at any instant of time. To enforce feasibility of the motion, the constraints regarding the dynamic model (3.23) and the contacts, which using the rigid point contact model are given by (3.3) and (3.4), have to be also considered.

Considering the tasks and the dynamic feasibility constraints, the operational-space inverse-dynamics (OSID) problem can be reduced to finding the optimization vector, sometimes called the dynamic variable, as

$$\nu = (\ddot{q}, \tau, f_c)$$

which needs to be consistent with the dynamic equations and should minimize the distance to the task reference. The approach computes at the same time the joint accelerations $\ddot{q}$, the control torques τ, and the contact forces f_c that the robot should apply to the ground. The explicit computation of these variables in the optimization solver has the advantage that forces are obtained in a straightforward way and no consistency verification or projection is necessary to guarantee their feasibility. Another advantage is that explicit constraints on each of the variables can be formulated, as "tasks", and either a torque-controlled or a position-controlled robot can be used (by integrating $\ddot{q}$ twice). With the lexicographic order introduced in Section 3.3.2, and considering n_t tasks of the form (3.37), the dynamic SoT based on the HQP[4] is given by

$$(3.23) \prec (3.3) \prec (3.4) \prec (3.37)\text{-}1 \prec \cdots \prec (3.37)\text{-}n_t.$$

where (3.37)-i represents the i-th task of the form (3.37), which is fully specified by the reference behavior $\ddot{e}_i^*$ and the Jacobian J_i. To ensure system balance it is generally better to add sufficient tasks to fill up the SoT, in which case there is no more redundancy to minimize the torques. But if the SoT is not full, a last task related to 'joints friction' or posture (also referred to as damping task) is generally added to complete it [Khatib 04c]. Fig. 3.2 represents graphically the inverse dynamics Stack of Tasks.

In cases where all the contact points are on a same horizontal plane, the inequality condition (3.4), due to its non-negativeness, implies the existence of the ZMP inside the support polygon ensuring the dynamic balance of the robot [Saab 13]. Thus, the inverse dynamics control ensures motion generation with dynamic feasibility and respecting the ZMP constraint.

[4]The approach appears in: L. Saab, O. Ramos, F. Keith, N. Mansard, P. Souères, J-Y. Fourquet, *Dynamic Whole-Body Motion Generation under Rigid Contacts and other Unilateral Constraints*, IEEE Transactions on Robotics (T-RO), Vol.29 N.2, pag. 346 - 362, April 2013 [Saab 13].

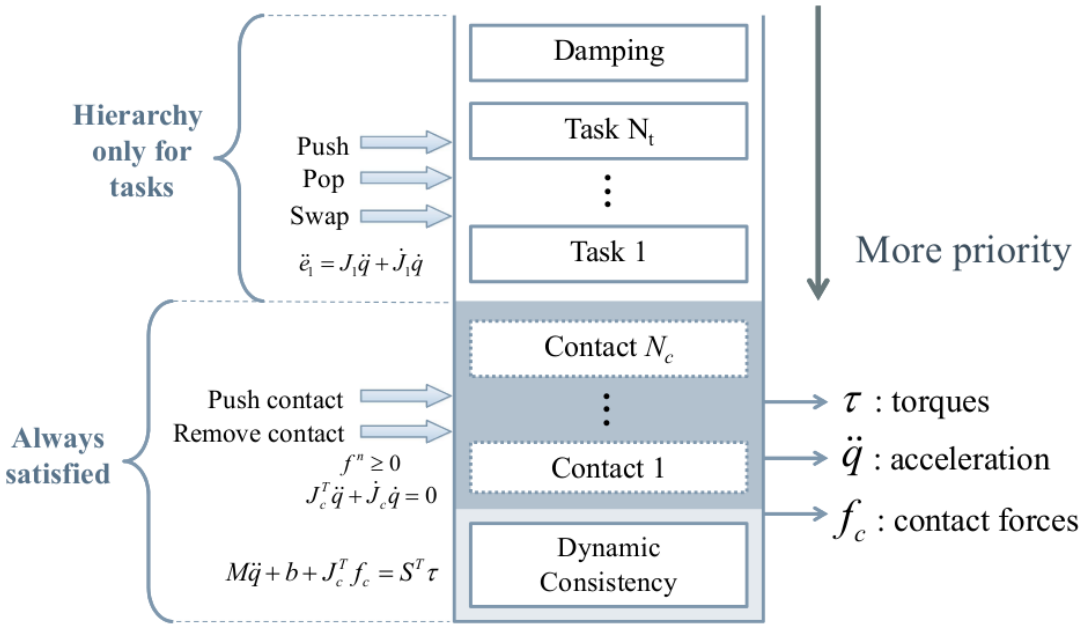


Figure 3.2 – Scheme of the Inverse Dynamics Stack of Tasks. The highest priority is at the bottom of the stack. The dynamic consistency and the contact constraints are always satisfied. It should be noted that contacts can be pushed or removed from the stack at will. The hierarchy exists at the level of the tasks, which can be added or removed as needed. The outputs of the stack are the joint torques, accelerations and the contact forces, which constitute the optimization variables.

The model used assumes that large friction coefficients are present and, thus, only normal forces are considered instead of more general friction cones. The generalization to friction cones is straightforward and (3.8) would have to be added to the solver. However, a good typical cone approximation uses 12 generators which introduce 12 new variables per contact point. This increases the complexity of the HQP solver and thus this approach is not currently used due to its computational cost.

3.3.4 Decoupling Motion and Actuation

The variables $(\ddot{q},\ \tau,\ f)$ can be decoupled to achieve a faster computation. In fact, the space given by $(\ddot{q}, f, \tau)$ in the previous HQP can be divided in three subspaces. The first one is the motion space, where joint accelerations $(\ddot{q})$ can be freely chosen, and the corresponding forces (f) and torques (τ) are accordingly set. The second is the actuation space, where the acceleration is fixed and only forces can be freely chosen forasmuch as forces and torques are related. The third space is useless since motion variables can be theoretically chosen, but resulting forces are impractical. To explicitly distinguish the actuation space from the motion space, two decoupled spaces can be introduced. To this end, an automatic formulation has been proposed in [Mansard 12]. This formulation states that instead of using the original optimization variables, the bases of the two decoupled spaces can be used and the dynamic model as well as the tasks can be reformulated in terms of these decoupled variables. This allows a faster computation of the SoT since the decoupled spaces are of lower dimension than the original coupled variables.

Decoupled Dynamics

The description presented in this section is a summary of [Mansard 12], which aims at decoupling the dynamics contained in the dynamic model given by (3.23) as well as the contact constraints. Using the Cholesky decomposition, the inverse of the inertia matrix M can be expressed as $M^{-1} = BB^T$, and thus $M = B^{-T}B^{-1}$, with B an invertible triangular matrix. Multiplying the dynamic model (3.23) by B^T the following expression is obtained

$$B^T S^T \tau = B^{-1}\ddot{q} + B^T b + G^T f_c \tag{3.38}$$

where G is defined as $G = J_c B$. The result of multiplying (3.38) by G on the left, using (3.3) to guarantee no kinematic motion at the rigid contact point, is

$$GB^T S^T \tau = GB^T b + GG^T f_c - \dot{J}_c \dot{q}$$

where the fact that $GB^{-1}\ddot{q} = J_c B B^{-1}\ddot{q} = -\dot{J}_c\dot{q}$, so that $\ddot{x}_c = 0$, has been used. Solving this expression for $B^T S^T \tau$, it gives:

$$B^T S^T \tau = B^T b + G^T f_c - G^{\#}\dot{J}_c\dot{q} + Vu$$

where V is a basis of the kernel of G, so that $VV^T = I - G^{\#}G$, and u is a vector in this kernel representing the motion of the system, as (3.42) shows. The term $\delta_c = -G^{\#} + \dot{J}_c\dot{q}$ can be associated to the contact drift, and thus, a more compact expression describing the constrained dynamics can be written as:

$$B^T S^T \tau = B^T b + G^T f_c + \delta_c + Vu. \tag{3.39}$$

An actuation free relation can be obtained if the torque τ is eliminated in (3.39). To this end, the matrix $\bar{S}$ is defined as $\bar{S} = [I\ 0]$ in such a way that it cancels out S^T, that is $\bar{S}S^T = 0$, and thus it also eliminates the effect of the torque. Multiplying both terms of (3.39) by $\bar{S}B^{-T}$, a torque-free relation between the contact forces f_c and the motion term u is obtained:

$$\bar{S}B^{-T}Vu + \bar{S}J_c^T f_c = -\bar{S}b - \bar{S}B^{-T}\delta_c. \tag{3.40}$$

This expression is composed of both (3.23) and (3.3) but explicitly using the motion (u) and contact force (f_c) variables. The other complementarity contact condition related to the unidirectionality of the forces can be directly included in the decoupled model as (3.4) shows, since the contact force is a variable in the model.

Reduction of the Force

It is possible to reduce the force variable as [Mansard 12] proposes. In this case, the contact force f_c can be solved for the system (3.40) giving

$$f_c = -(\bar{S}J_c^T)^{\#}\bar{S}(b + B^{-T}\delta_c + B^{-T}Vu) + K\psi$$

where K is a basis of the kernel of $\bar{S}J_c^T$ such that $KK^T = I - (\bar{S}J_c^T)^{\#}(\bar{S}J_c^T)$, and ψ is a vector in this kernel. Since B is triangular, the reduction can further be written as

$$f_c = -(\bar{S}G^T)^{\#}\bar{S}(B^T b + \delta_c + Vu) + K\psi. \tag{3.41}$$

This reduction contains at the same time the inverse dynamics and the kinematic contact constraint and gives a compact expression that at the same time contains the motion constraints and the force constraints.

Tasks Representation

A generic task can be typically represented as (3.37), which relates the task acceleration to the generalized joint space acceleration. The equivalence between (3.38) and (3.39) allows to eliminate the torque component and the contact force component and leads to the relation between the acceleration $\ddot{q}$ and u as:

$$\ddot{q} = B\delta_c + BVu \tag{3.42}$$

Therefore, the reduced variable u can be directly associated with the motion $\ddot{q}$ and is usually referred to as the motion variable in the reduced dynamics formulation. Using (3.42), the generic task representation (3.37) can be equivalently expressed in terms of the decoupled variable u as

$$\ddot{e}_i^* = \dot{J}_i\dot{q} + J_iB\delta_c + J_iBVu \tag{3.43}$$

which is the task formulation in the reduced dynamics framework.

Decoupled Stack of Tasks

The fully decoupled dynamics can be formulated as the following minimization problem. Considering a task i, and using (3.43) and (3.41), the optimization problem is:

$$\begin{aligned} \min_{u,\psi} \; & \| \; J_iBVu + \dot{J}_i\dot{q} + J_iB\delta_c - \ddot{e}_i^* \; \| \\ \text{s.t.} \; & -(\bar{S}G^T)^{\#}\bar{S}Vu + K\psi \geq f_0 \end{aligned} \tag{3.44}$$

where the constraint is directly obtained from (3.41), and thus, f_0 is typically computed as $f_0 = -(\bar{S}G^T)^{\#}(B^Tb + \delta_c)$. This is the most general decoupled equation and its simplicity can be observed from the fact that only a single constraint is needed, which already comprises the inverse dynamics model as well as the rigid contact conditions. For this system, and considering n_t tasks, the inverse dynamics SoT can be formulated as

$$(3.44) \prec (3.43)\text{-}1 \prec \cdots \prec (3.43)\text{-}n_t.$$

An alternative way to solve the system, making explicit the computation of the contact forces is

$$\begin{aligned} \min_{u,f_c} \; & \| \; J_iBVu + \dot{J}_i\dot{q} + J_iB\delta_c - \ddot{e}_i^* \; \| \\ \text{s.t.} \; (\bar{S}B^{-T}V)u + (\bar{S}J_c^T)f_c &= -\bar{S}(b + B^{-T}\delta_c) \\ f_c^{\perp} &\leq 0. \end{aligned}$$

using the lexicographic notation, this system can be written as a hierarchy of n_t tasks as

$$(3.40) \prec (3.4) \prec (3.43)\text{-}1 \prec \cdots \prec (3.43)\text{-}n_t.$$

The torques can be recovered from (3.39). To this end, the fact that $SS^T = I$ can be profitted. Then, multiplying (3.39) by SB^{-T} on the left, the torques are given by

$$\tau = Sb + SJ_c^Tf_c + SB^{-T}(\delta_c + Vu) \tag{3.45}$$

and thus, the decoupled dynamics can be used in the computation and the necessary torques can be then easily computed.

Spatial-force Reduction

Consider a single rigid body i in contact with the environment. The twist associated with a certain reference frame located on the body (for instance, its center of mass) will be denoted by $\xi = (v, \omega) \in se(3)$, where v represents the linear velocity, and ω the angular velocity. The relation between this twist and the twist $(v_k, \omega_k) \in se(3)$ of a generic point p_k on the body (which can be a contact point) is given by:

$$\begin{bmatrix} v_k \\ \omega_k \end{bmatrix} = \begin{bmatrix} I & -\hat{p}_k \\ 0 & I \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{3.46}$$

where $\hat{\{.\}}$ represents the skew-symmetric matrix of pre-cross product. Let the set of n_c contact points, where the forces are directly applied, be denoted by $x_c = (p_1, p_2, \cdots, p_{n_c}) \in \mathbb{R}^{3n_c}$, their velocity by $\dot{x}_c$, and their acceleration by $\ddot{x}_c$. Since these points are attached to the rigid body, they do not present angular velocity with respect to it, and further analysis will only consider linear velocity. Using a direct extension of (3.46), the linear velocity relation is:

$$\dot{x}_c = X\ \xi \tag{3.47}$$

where

$$\dot{x}_c = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n_c} \end{bmatrix} \qquad \text{and} \qquad X = \begin{bmatrix} I_3 & -\hat{p}_1 \\ I_3 & -\hat{p}_2 \\ \vdots & \\ I_3 & -\hat{p}_{n_c} \end{bmatrix}. \tag{3.48}$$

Let the geometric Jacobian of the generic frame attached to the body be denoted by J_o so that $\xi = J_o\ \dot{q}$. Using this Jacobian, the contact Jacobian ($\dot{x}_c = J_c \dot{q}$), and (3.47), it can be easily verified that the link between the contact Jacobian and the geometric Jacobian is given by X, so that

$$J_c = X J_o. \tag{3.49}$$

When there are 3 or more contact points on the same body, X is full-column rank and the null spaces of J_c and J_o are equal. In this case, J_o can be used to compute the V basis as $V = ker(J_o B)$ instead of $V = ker(J_c B)$, and the contact drift as $\delta_c = -(J_o B)^{\#} \dot{J}_o \dot{q}$. However, when there are only 1 or 2 contact points the null spaces are no longer equal. In this situation, X in (3.48) is not full-column rank, and the previous reductions where J_o could be used instead of J_c do not necessarily hold since both matrices do not have the same null space. In this case, the equivalence $J_c = X J_o$ is used and the computation of V is explicitly $ker(X J_o B)$. The relation between the derivative of both Jacobians is $\dot{J}_c = X \dot{J}_o + \dot{X} J_o$, but the second term is null ($\dot{X}$ is null provided that the contact points do not move) leading to $\dot{J}_c = X \dot{J}_o$. Then, for the contact drift in the generic case, $\delta_c = -(X J_o B)^{\#} X \dot{J}_o \dot{q}$.

3.4 Tasks for Motion Generation

Motion generation within the dynamic SoT is based upon the definition and usage of different tasks at the acceleration level. These tasks[5] are completely specified through the task reference

[5]Some of these tasks appear in: L. Saab, O. Ramos, F. Keith, N. Mansard, P. Souères, J-Y. Fourquet, *Dynamic Whole-Body Motion Generation under Rigid Contacts and other Unilateral Constraints*, IEEE Transactions on Robotics (T-RO), Vol.29 N.2, pag. 346 - 362, April 2013 [Saab 13].

behavior $\ddot{e}^*$, which can be defined using either equalities or inequalities, and the corresponding task Jacobian J. The tasks presented in this section are tasks that have been used within the framework to generate the motion of Chapter 4.

3.4.1 Proportional Derivative (PD) Tasks

A proportional derivative task is defined by setting a proportional derivative (PD) control law for the reference behavior $\ddot{e}^*$ as:

$$\ddot{e}^* = -k_p e - k_v \dot{e} \tag{3.50}$$

where $k_p > 0$ and $k_v > 0$ are gains used to set the convergence velocity and typically $k_v = 2\sqrt{k_p}$. This control law imposes an exponential decay of the task, assumed to be a regulation task, tending to zero. A fixed gain can be used for k_p, for instance when tracking a moving target, but an adaptive gain to reach a fixed target can also be defined as

$$k_p = (k_{min} - k_{max})e^{-\beta\|e\|} + k_{max}$$

where $\|e\|$ is the norm of the task error, k_{max} is the maximum gain (when the target is far), k_{min} is the minimum gain when the error approaches zero, and β is a variable that regulates the velocity of the transition.
Typical tasks that use the PD reference behavior, which only differ among them in the specification of the task error e, are the following.

Placement (6D) Task A generic placement task is a regulation task for the position and orientation of one body of the robot; for instance, the head or a hand. The frame attached to the rigid body to be controlled is usually called an *operational point*. This task is usually defined as

$$\begin{bmatrix} e_p \\ e_o \end{bmatrix} = \begin{bmatrix} x - x^* \\ r\theta \ominus r\theta^* \end{bmatrix} . \tag{3.51}$$

where x and x^* are the current and desired position of the operational point, respectively, $r\theta$ represents the axis-angle representation of the orientation, and $\ominus$ is a suitable difference group operator of $SO(3)$. The first part of the task (e_p) defines the position error as the difference between the current and the desired position, whereas the second part (e_o) defines the attitude error.

The attitude error needs a special consideration, and an example is provided here. Let $R \in SO(3)$ be the current orientation and $R^* \in SO(3)$ be the desired orientation. The correction matrix describing the attitude error can be expressed with respect to the current frame as $R_e = R^{-1}R^*$, or with respect to the desired frame as $R_e = (R^*)^{-1}R$. In either case, the tangent space of R_e is given by the skew symmetric matrix $\hat{\omega}_e \in so(3)$ so that $R_e = e^{\hat{\omega}_e}$ or, equivalently, $\hat{\omega}_e = \log(R_e)$ [Murray 94]. The vector containing the components of the skew-symmetric matrix $\omega_e = (\hat{\omega}_e)^\vee$ is the axis-angle representation of R_e, where $\|\omega_e\|$ is the angle and $\frac{\omega_e}{\|\omega_e\|}$ is the axis. Thus, the vector $\omega_e = \{\log(R_e)\}^\vee$ can be used to represent the attitude error: $e_o = \omega_e$.

Center of Mass (CoM) Task Let a robot be composed of n_b rigid bodies, and let the mass of the i^{th} body be $m_i \in \mathbb{R}$ and its CoM position $r_{c_i} \in \mathbb{R}^3$. The CoM of the robot $r_c \in \mathbb{R}^3$

is computed using the CoM of the n_b bodies as

$$r_c = \frac{\sum_{i=1}^{n_b} m_i r_{c_i}}{\sum_{i=1}^{n_b} m_i}.$$

A task for the CoM (e_{com}) is defined as the error between the current position of the CoM and its desired position $r_c^* \in \mathbb{R}^3$:

$$e_{com} = r_c - r_c^*. \tag{3.52}$$

According to the situation, the specification of the task for the CoM can involve the three position components or only some subset of them.

Posture Task This task is used to directly control the joint space of the robot, either as a final desired posture or as a trajectory to follow as closely as possible. For a robot with n DoF, let $q_{jk} = (q_j, \ldots, q_k) \in \mathbb{R}^{k-j+1}$ with $k, j = \{1, \cdots, n\}$, $j \geq 1$, $n \geq k$ and $j \leq k$ represent the vector containing the joint angular values to be controlled (the active joints), and q_{jk}^* the desired joint configuration. The posture task (e_{pos}) is expressed as

$$e_{pos} = q_{jk} - q_{jk}^* \tag{3.53}$$

which is the error between the current active joint positions and the desired ones. The Jacobian of this task only selects the active joints and is given by

$$J_{jk} = \begin{bmatrix} \bar{0}_f & \bar{0}_j & I_{jk} & \bar{0}_k \end{bmatrix} \tag{3.54}$$

where $\bar{0}_f \in \mathbb{R}^{(k-j+1)\times n_b}$ is the zero matrix corresponding to the free-flyer (whose pose representation has n_b elements), $\bar{0}_j \in \mathbb{R}^{(k-j+1)\times(j-1)}$ and $\bar{0}_k \in \mathbb{R}^{(k-j+1)\times(n-k)}$ are zero matrices corresponding to the non-active joints, and $I_{jk} \in \mathbb{R}^{(k-j+1)\times(k-j+1)}$ is the identity matrix corresponding to the active joints. These tasks can be used to optimize some performance criteria such as the actuation torques and to reproduce human-like movements.

Visual Task This task's objective is to move the head so that a certain point in the 3D space is projected to a certain position in the image seen by the robot camera. Thus, the task aims at focusing the gaze (the camera) at a certain image point so that

$$e_{gaze} = p_i - p_i^* \tag{3.55}$$

where $p_i \in \mathbb{R}^2$ is the 2D projection on the camera of some 3D point $P_i \in \mathbb{R}^3$, and $p_i^* \in \mathbb{R}^2$ is the desired position on the image plane. The visual task Jacobian in this case is given by $J_{vis} = L\ {}^cJ$, where L is the interaction matrix proposed in [Chaumette 06], which describes the relation between the velocity of the camera and the velocity of a 2D image point, and ${}^cJ$ is the Jacobian in the camera frame.

3.4.2 Joint Limits Tasks

Robot joints typically present some physical constraints with respect to their angular position and velocity: there exists a set of admissible angular positions, as well as admissible angular velocities, which define the joint limits for each joint of the robot. A motion intended to be

executed on a real robot must always consider these limits for its feasibility. Joint limits can be taken into account in control frameworks using methods like potential fields [Khatib 86], or clamping [Baerlocher 04, Raunhardt 07], but they can also be integrated as inequality constraints in the SoT using the task function approach.

Within the framework of the dynamic SoT, it is necessary to find the relation between joint positions and joint accelerations. Suppose that the system is currently at time t. Joint values at time $t + \Delta t$, which is the next instant of time, can be expressed using a Taylor expansion as:

$$q(t + \Delta t) = q(t) + \Delta t\ \dot{q}(t) + \frac{\Delta t^2}{2}\ddot{q}(t) + O(|\Delta t|^3)$$

where $O(|\Delta t|^3)$ is the remainder term of the second order approximation, which is very small compared to $|\Delta|^3$. To simplify the expression, let $\hat{q} = q(t + \Delta t)$, $q = q(t)$, $\dot{q} = \dot{q}(t)$, and $\ddot{q} = \ddot{q}(t)$. Then, since the effects of higher order terms are negligible, an accurate approximation of the joint configuration after Δt is

$$\hat{q} \approx q + \Delta t\ \dot{q} + \frac{\Delta t^2}{2}\ddot{q} \tag{3.56}$$

which gives a relation between the estimation of the next joint configuration and the current acceleration of the robot joints.

Let the maximum limits for the joint angular positions be represented by $\overline{q}$ and the minimum by $\underline{q}$, so that the constraint can be expressed using a double-side bounded inequality as $\underline{q} < \hat{q} < \overline{q}$. This inequality specifies the limits for the next joint configuration and prevents the joints from going beyond their angular limits. Considering the second order approximation of $\hat{q}$ given by (3.56), the joint angular limits inequality can be written as

$$\underline{q} \leq q + \dot{q}\Delta t + \ddot{q}\frac{\Delta t^2}{2} \leq \overline{q} \tag{3.57}$$

which is a function of the acceleration. The value of Δt is the length of the preview window. In theory, the control sampling time $T_s = 1$ ms should be used for Δt, but in practice a smoother behavior can be obtained by adjusting the value as $\Delta t = \frac{T_s}{\lambda_s}$, where λ_s can be set as the gain of the task. Using (3.57), and assuming that the task reference behavior is the joint acceleration itself ($\ddot{e}^*_{jp} = \ddot{q}$), the task can be specified in terms of the acceleration as:

$$\underline{e}_p \leq \ddot{e}^*_{jp} \leq \overline{e}_p \tag{3.58}$$

where

$$\underline{e}_p = \frac{2}{\Delta t^2}\left(\underline{q} - q - \dot{q}\Delta t\right) \qquad \text{and} \qquad \overline{e}_p = \frac{2}{\Delta t^2}\left(\overline{q} - q - \dot{q}\Delta t\right)$$

In this case, the task Jacobian is given by $J = [0 \quad I]$ since the elements corresponding to the free-floating base of the robot have to be excluded.

Joint angular velocity limits can also be taken into account as tasks. Let $\overline{\dot{q}}$ represent the upper bound velocities and $\underline{\dot{q}}$ the lower bound velocities, which define the joint velocity inequality: $\underline{\dot{q}} \leq \dot{\hat{q}} \leq \overline{\dot{q}}$. As for the angular position q, the velocity can also be approximated in terms of the acceleration and, with the first order approximation, the limits become:

$$\underline{\dot{q}} \leq \dot{q} + \ddot{q}\Delta t \leq \overline{\dot{q}}. \tag{3.59}$$

Using (3.59), the joint angular velocity limits task can be defined as

$$\underline{e}_v \leq \ddot{e}^*_{jv} \leq \overline{e}_v \tag{3.60}$$

where $\ddot{e}^*_{jv} = \ddot{q}$ and

$$\underline{e}_v = \frac{1}{\Delta t}\left(\underline{\dot{q}} - \dot{q}\right) \qquad \text{and} \qquad \overline{e}_v = \frac{1}{\Delta t}\left(\overline{\dot{q}} - \dot{q}\right)$$

The task Jacobian is defined in a similar way as for the joint angular limits, since only the actuated joints are considered.

The torque magnitude can also be bounded in the framework since the torque variable is included in the optimization process. In this case, this cannot be defined as a single task but can be defined as constraint in the hierarchized optimization process. The constraint can simply be defined as

$$\underline{\tau} \leq \tau \leq \overline{\tau} \tag{3.61}$$

where $\underline{\tau}$, $\overline{\tau}$ are the torque limits.

3.4.3 Interpolation Task

Let $x(t)$ describe the evolution of a generic feature, such as the pose of an operational point, and let $x_0 = x(t_0)$ and $\dot{x}_0 = \dot{x}(t_0)$ denote its position and velocity at the initial time t_0. After a certain time T, the final time becomes $t_f = t_0 + T$, and the desired position and velocity become $x_f = x(t_f)$ and $\dot{x}_f = \dot{x}(t_f)$. An *interpolation task* is a task that is defined to reach a desired final position and velocity $(x_f, \dot{x}_f)$ in a specific period of time T, assuming that the current position and velocity $(x_0, \dot{x}_0)$ are known. The basic difference from tasks that follow a PD control law is the fixed and hard time constraint.

Since the acceleration is needed for the task specification, a cubic polynomial for the position interpolation, which is the polynomial with minimum degree that allows the control of the acceleration, is chosen. To this end, the desired acceleration $\ddot{x}(t)$ is set to a linear function as in [deLasa 10]:

$$\ddot{x}(t) = \ddot{x}_0 + \frac{\ddot{x}_f - \ddot{x}_0}{T}(t - t_0) \tag{3.62}$$

where $\ddot{x}_0$ and $\ddot{x}_f$ are, respectively, the unknown initial and final accelerations that define the acceleration trajectory. The solution of the differential equation (3.62) leads to a quadratic velocity and a cubic position trajectories given by:

$$\dot{x}(t) = \frac{\ddot{x}_f - \ddot{x}_0}{2T}(t - t_0)^2 + \ddot{x}_0(t - t_0) + \dot{x}_0 \tag{3.63}$$

$$x(t) = \frac{\ddot{x}_f - \ddot{x}_0}{6T}(t - t_0)^3 + \frac{\ddot{x}_0}{2}(t - t_0)^2 + \dot{x}_0(t - t_0) + x_0. \tag{3.64}$$

Provided that the system is time dependent, as time passes by, the initial (current) position and velocity change but the final (desired) position and velocity remain constant. Thus, a task computed considering a "preview future horizon", from the current time to the final time t_f, can be used at every intermediate step. However, only the current reference acceleration is used, so that the task is permanently updated based on the current conditions. That is, the control law given by (3.62) is continuously updated but only $\ddot{x}_0$ is used at every instant of time. This behavior resembles the Model Predictive Control since only the present state

is used although the whole time horizon has been computed. The dynamic task can thus be simply defined as $\ddot{e}^* = \ddot{x}_0$, and can be obtained by solving the system (3.63) and (3.64) at time $t = t_f$ leading to the following desired reference behavior:

$$\ddot{e}^* = \frac{6}{T^2}(x_f - x_0) - \frac{2}{T}(\dot{x}_f + 2\dot{x}_0). \tag{3.65}$$

It should be noted that T becomes smaller after each iteration and is close to zero when the initial time reaches the final time. This situation can be handled by keeping the previous value of $\ddot{x}_0$ when $T < t_h$, where t_h is a threshold. In practice, keeping the previous value of the acceleration gives good results.

3.4.4 Capture Point (CP) Task

The *Capture Point* (CP), introduced in [Pratt 06b] and also referred to as the *Extrapolated Center of Mass* [Hof 05], is the point $\xi = [\xi_x \ \xi_y \ 0]^T$ on the ground where the robot should step on to be able to come to a complete rest. It is based on modeling the robot as a linear inverted pendulum (LIP) [Pratt 06a]. Let the position of the robot CoM be represented by $r_c = (x_c, y_c, z_c)$, so that the horizontal component of the CoM is $\tilde{r}_c = \hat{n} \times r_c \times \hat{n}$, with $\hat{n} = (0, 0, 1)$, or equivalently, $\tilde{r}_c = (x_c, y_c, 0)$. It can be shown (see Appendix C for more details) that the instantaneous Capture Point is given by

$$\xi = \tilde{r}_c + \frac{\dot{\tilde{r}}_c}{\omega} \tag{3.66}$$

where $\omega = \sqrt{\frac{g}{z_c}}$ is the eigenfrequency of the pendulum that models the robot dynamics.

The CP task[6] is proposed as an inequality task that aims at keeping the CP inside certain limits at all times. These limits can be the robot support polygon, in which case the robot motion can come to a stop without needing a step, or an exterior polygon. In the latter case the robot needs to perform a step, but the motion can be constrained so that the CP stays within some reachable region for the leg. In both cases, the task is bounded by a generic polygon. Let the limits of the generic polygon be given by $\underline{r}_p$ and $\overline{r}_p$. The CP must lie within the limits of this polygon as:

$$\begin{aligned} \underline{r}_p \leq \ \xi \ &\leq \overline{r}_p \\ \underline{r}_p \leq \tilde{r}_c + \frac{\dot{\tilde{r}}_c}{\omega} &\leq \overline{r}_p. \end{aligned} \tag{3.67}$$

From (3.67), it is evident that the CP can be bounded by indirectly controlling the CoM position and velocity. Then, the task to effectively control the CP can be defined in terms of the CoM. Since tasks are integrated in a dynamic control scheme, the CP task needs to be formulated using the acceleration; hence, $\tilde{r}_c$ and $\dot{\tilde{r}}_c$ in (3.67) need to be related to the CoM acceleration. As for the case of the joint limits, presented in Section 3.4.2, the position of the CoM can be expressed, using a Taylor series development around point $\tilde{r}_{c_i}$, as:

$$\tilde{r}_c = \tilde{r}_{c_i} + \dot{\tilde{r}}_{c_i}\Delta t + \ddot{\tilde{r}}_{c_i}\frac{\Delta t^2}{2}$$

[6]The CP task was introduced in: O. Ramos, N. Mansard, P. Souères, *Whole-body Motion Integrating the Capture Point in the Operational Space Inverse Dynamics Control*, IEEE-RAS International Conference on Humanoid Robots (Humanoids), Madrid, Spain, November 2014 [Ramos 14b].

and the velocity as

$$\dot{\tilde{r}}_c = \dot{\tilde{r}}_{c_i} + \ddot{\tilde{r}}_{c_i} \Delta t.$$

Replacing the Taylor expansion of the position and the velocity of the CoM in (3.67), and letting the task reference behavior be $\ddot{e}^*_{cp} = \ddot{\tilde{r}}_{c_i}$, the Capture Point task respecting the limits within the polygon is expressed as a function of the CoM as:

$$k_a \, \underline{e}_c \leq \ddot{e}^*_{cp} \leq k_a \, \overline{e}_c \tag{3.68}$$

where k_a is a constant given by

$$k_a = \frac{2\omega}{\Delta t(\omega \Delta t + 2)}$$

and the acceleration limits are

$$\underline{e}_c = \underline{r}_{sp} - \tilde{r}_{c_i} - \dot{\tilde{r}}_{c_i} \left(\Delta t + \frac{1}{\omega} \right)$$
$$\overline{e}_c = \overline{r}_{sp} - \tilde{r}_{c_i} - \dot{\tilde{r}}_{c_i} \left(\Delta t + \frac{1}{\omega} \right).$$

3.5 Conclusion

Based on the task function approach, this chapter has presented the operational-space inverse-dynamics control framework that is used throughout this thesis. A hierarchical QP was proposed to solve the inverse dynamics problem and give a solution to the tasks generating whole-body motion that satisfies the dynamic constraints. Finally, several tasks defined as equalities or inequalities, which can be used at any level of the hierarchy to achieve the desired effect, were proposed as the basis for motion generation.

Chapter 4

Case Studies

The whole-body motion generation described in Chapter 3 can be applied to different experiments with the robot. This chapter presents different case studies where the motion generation using the operational-space inverse-dynamics (OSID) stack of tasks (SoT) has been successfully applied to show and explore some of the possibilities that it can offer. The case studies show results obtained both in simulation, using the dynamic model of HRP-2, and in reality using the robot.

4.1 Motion with Multiple Non-coplanar Contacts

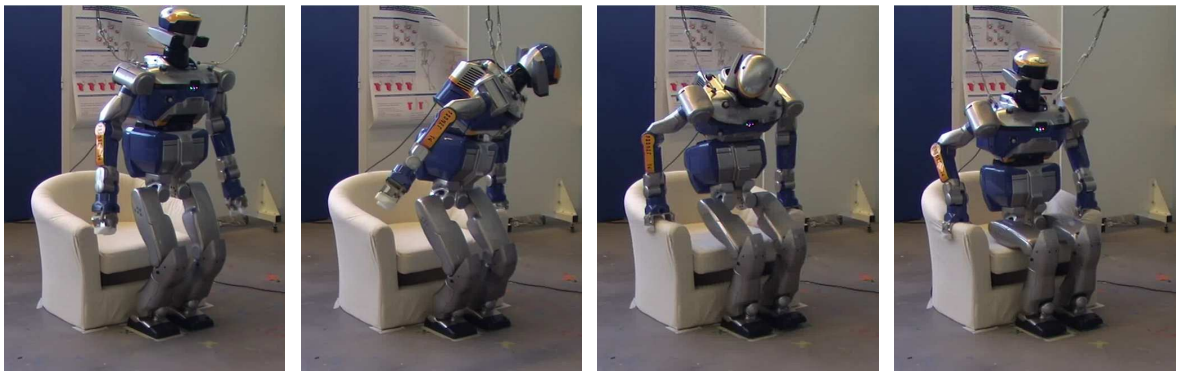
The first case study illustrates the possibility of multiple non-coplanar contacts during a complex sequence of motions. It consists on two specific motions: the first one presents the real HRP-2 sitting down in an armchair, and the second one presents the simulated Romeo robot climbing a ladder. The ability to handle multiple non-coplanar contacts is an advantage of the proposed methodology over other existing inverse dynamics approaches and is exploited in this section.

4.1.1 Sitting in an Armchair

In this sequence of movements, HRP-2 moves its whole-body to sit in an armchair as Fig. 4.1 shows[1]. The robot does not only have both feet in contact with the ground but gradually the contacts of the hands and the armrests are added and taken into account in the solver. The high-level description of the motion is the following:

- First, the robot looks left and moves its left arm towards the left armrest so that contacts between the left gripper and the left chair armrest are found (Fig. 4.1b).

[1]This experiment appears in: L. Saab, O. Ramos, F. Keith, N. Mansard, P. Souères, J-Y. Fourquet, *Dynamic Whole-Body Motion Generation under Rigid Contacts and other Unilateral Constraints*, IEEE Transactions on Robotics (T-RO), Vol.29 N.2, pag. 346 - 362, April 2013 [Saab 13].



(a) $t = 0s$ **(b)** $t = 7s$ **(c)** $t = 15s$ **(d)** $t = 19s$

Figure 4.1 – Snapshots of HRP-2 sitting in an armchair with the corresponding timing. The sequence is: (a) the robot stands on both feet, (b) it looks left and grasps the left armrest, (c) it looks right and grasps the right armrest, (d) it finally sits down.

- Then, the robot looks right moving its right arm towards the right armrest and contacts between the right gripper and the right chair armrest are found (Fig. 4.1c).
- Finally, the waist is brought down to touch the seat, adding an additional contact with the seat (Fig. 4.1d).

The whole motion is divided in several sub-parts, and each sub-part is generated using a specific sequence of tasks within the OSID SoT.

Inverse Dynamics SoT As described in Section 3.3.3, this framework uses the robot dynamic model and allows for the generation of feasible motion satisfying the imposed contact constraints and achieving the objectives through the usage of operational-space tasks. The SoT is solved using an HQP where the highest priority is given to the dynamic model of the robot followed by the contact constraints. Then, the tasks defining additional constraints or specific motions are added. In the sequel, the term "priority" strictly refers to the priority inside these tasks without considering the dynamic model or the contact constraints, which are always present at the highest level of the hierarchy for the global feasibility of the motion.

Contact Constraints They are added to the SoT according to which parts of the robot are in contact with the environment (in this case, the floor and the armrests). In this motion, the contact for both feet is always considered provided that the robot starts with a standing posture and both feet remain on the ground. For the hands, the contacts were added according to whether they touched the respective armrest or not.

Description of the Tasks At the highest priority level of the SoT, joint limits in angular position (3.57) and angular velocity (3.59), defined as inequalities, are added to ensure that the joints stay within the pre-defined safe limits satisfying mechanical constraints. These limits are always specified at the highest priority level. Then, the motion tasks are added. A task for the right hand e_{rh} and the left hand e_{lh} are defined as placement tasks given by (3.51). They are set on each robot gripper to control the position and orientation towards

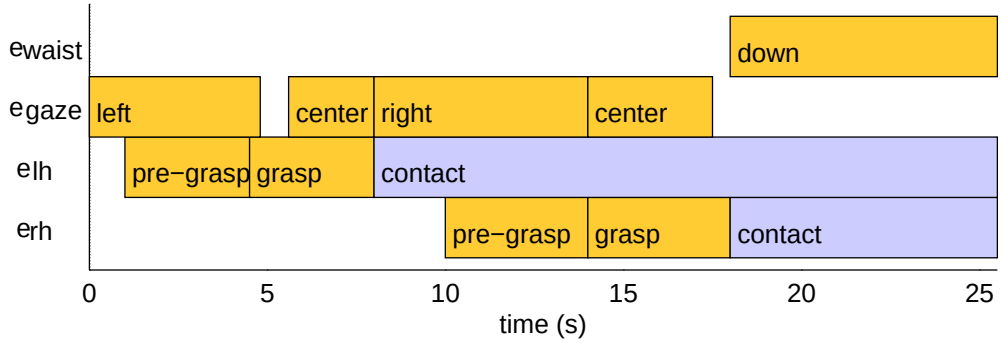


Figure 4.2 – Sequence of tasks and contacts. The gaze task focuses sequentially on the left and right armrests and on a virtual point in front of the robot. The pre-grasp tasks are set at the vertical $10cm$ above the grasp position.

the corresponding armrest. To prevent collisions when grasping, an intermediate point is first reached above the final grasping position and the gripper is opened before reaching the intermediate point. The contact of each gripper with the armrest is realized by the rear part of the open gripper using a support polygon given by a $5cm$-wide square. To improve the naturalness of the motion, a task called e_{gaze}, defined by (3.55), is set to constrain the gaze towards the armrest to be grasped. This gives the effect of the robot looking at the armrest as it is grasping it, since a simple motion of the head joints is not enough to achieve this effect. After each grasp, the gaze is brought back in front of the robot. Finally, the waist is controlled by a task called e_{waist} also defined by (3.51) where only the vertical position and sagital rotation are active: the waist is constrained to remain vertical and to move down to the seat.

Temporal Sequence The temporal sequence of tasks used for the motion generation is given in Fig. 4.2. The robot first looks left and bends to first pre-grasp and then grasp the left handle of the armchair. Once the handle is grasped, it is added as a contact constraint in the SoT. Then, the robot looks to the right (but with an intermediate center look), and bends to pre-grasp and then grasp the right handle. As it grasps the handle, the head gaze moves from the right to the center, and once the grasp finishes, the right handle is also added to the SoT as a contact constraint. Finally, using both handle supports, the robot moves the waist down to sit.
Using the lexicographic notation (3.36) to represent the hierarchy between the constraints and the tasks, the complete inverse dynamics stack of tasks, for the motion described above, is formalized as:

$$(3.23) \prec (3.3) \prec (3.4) \prec (3.57) \prec (3.59) \prec e_h \prec e_{gaze} \prec e_{waist}$$

where e_h can be the right or left hand task, when active. A final posture task was added at the last stage of the stack to constrain all the DoF.

Results

The previously described procedure was first applied in simulation and then with the real HRP-2. The inverse-dynamics SoT obtains the torques and accelerations needed for the

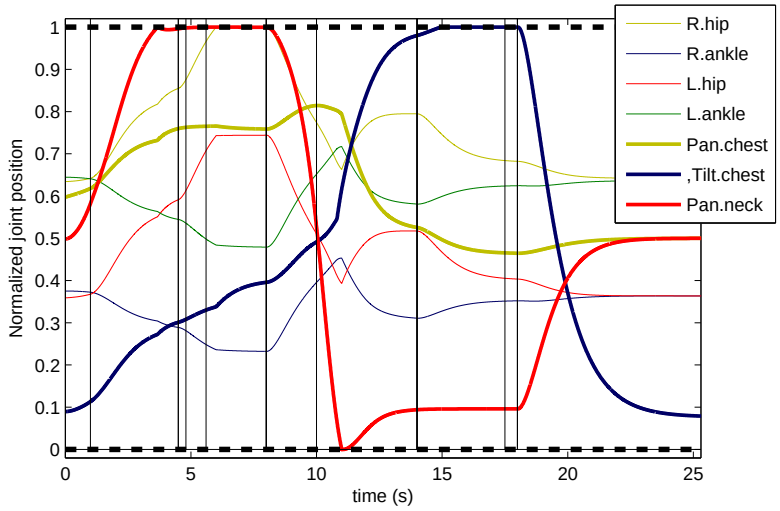


Figure 4.3 – Normalized joint angular positions of the right and left hip and ankle, chest and neck joints. Joint limits are properly avoided. When a limit is reached, one or several joints move in reaction to overcome the saturation.

control. However, since HRP-2 is a position-controlled robot, the accelerations were integrated twice to obtain the desired input positions to the robot. The snapshots of the experiment are shown in Fig. 4.1, which depicts the key frames of the motion executed by the robot described above: standing posture, grasping the left armrest, grasping the right armrest, and sitting down. It should be noted that the armrest is a common armrest used by people; it was not specifically designed for the robot, but it is suitable due to the robot anthropomorphic characteristics.

Joint Trajectories During the motion, the joint range is extensively used and joint limits are always considered within the SoT as inequalities. The most representative joint trajectories are plotted in Fig. 4.3 where the chronological sequence corresponding to the addition and removal of tasks and contacts shown in Fig. 4.2 are recalled by vertical stems at the transition instants. For homogeneity between the different joints, normalized joint positions are shown: 0 represents the lower limits and 1 the upper limits. When a limit is reached, one or several joints move in reaction to overcome the joint saturation. For instance, the neck joint reaches its limit while looking left; in reaction, all the other aligned joints move to overcome the neck limitation (chest joint, but also hip and ankle joints). The right hip then reaches its limit; as a consequence, the motion of both legs is stopped due to the lack of DoF to compensate this limit. The chest joint absorbs all the subsequent motion to fulfill the task. Later, when the robot looks right, the neck again reaches its limit. This time, the velocity of the joint when it reaches its limit is higher, which leads to a strong acceleration of the chest, and consequently brings the neck out of its limit. This behavior could be damped if necessary by tuning λ_s in (3.57). The chest joint finally reaches its limit at the end of the right-grasp task, which produces a limited overcome on the other joints. All the joints are properly stopped at the limit, and can leave the neighborhood of the limit without being stuck as it may appear with some avoidance techniques.

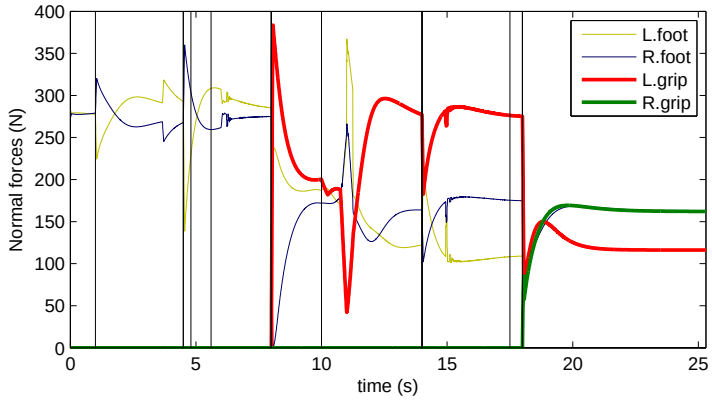


Figure 4.4 – Vertical forces distribution on the grippers and feet

Contact Considerations The definition of the contact with the two armrests is very useful to control the descent of the waist. In fact, if only the feet were considered as contacts, the motion would not be possible since the CoM would theoretically be outside the support polygon. The vertical forces on each support (grippers and feet) are plotted in Fig. 4.4, and the spatial evolution of the CoM is shown in Fig. 4.5. In the beginning of the motion, the robot is in a standing position and the weight is fully supported by the two feet, as shown in Fig. 4.5 and in the first temporal sequence of Fig. 4.4. After $t = 8s$, the left arm is used to also sustain the robot. However, the robot is still very much on the front, and this contact is not fully used yet. When going for the second armrest, the robot has to move its weight backwards, as Fig. 4.5 shows, and it uses the left-arm contact to ensure its balance: nearly half of the weight is then supported by the arm. Finally, the right armrest is grasped and the weight of the robot is properly distributed on the four contacts.

Computational Time The computation times are plotted in Fig. 4.6. It should be noted that throughout the whole motion the SoT is nearly full (the stack is said to be full when all the DoF are constrained). In that case, the computation cost is around 20ms per iteration, which is equivalent to five times the real time if controlling the robot at $200Hz$ (which is the typical control loop frequency for HRP-2). The number of tasks and, even more, the number of contacts modify the computational cost, as shown by the notorious computation increase at $t = 8s$ and $t = 18s$. The computational cost can be enhanced using the reduced approach, described in Section 3.3.4, which decouples motion and actuation. With this approach, the SoT is able to generate the control law without surpassing around 4.5ms per control cycle using a 2.9 GHz desktop computer (iCore 5 mono thread), achieving real time at 200 Hz. A similar computer is available on-board the real HRP-2 N.14.

4.1.2 Climbing up a Ladder

This approach of motion programming based on tasks is very versatile and can be applied to generate many types of movements. For example, this second motion presents the robot Romeo climbing up a ladder. This is also a type of motion where multiple non-coplanar contacts are used, and thus, it is similar in methodology to the previous motion showing the

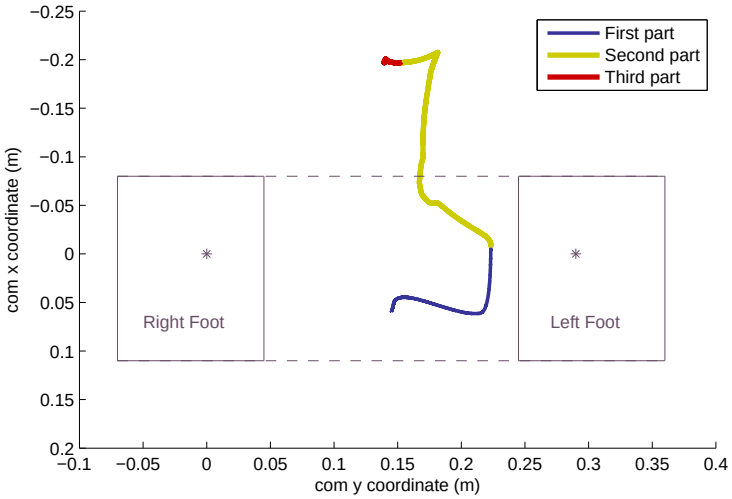


Figure 4.5 – Position of the CoM. The three phases correspond to changes in the number of contacts: first the two feet, then the left gripper and finally both feet and grippers. In a first time, the CoM stays forwards, but it is finally moved backwards to reach the second armrest and move the waist down to the seat.

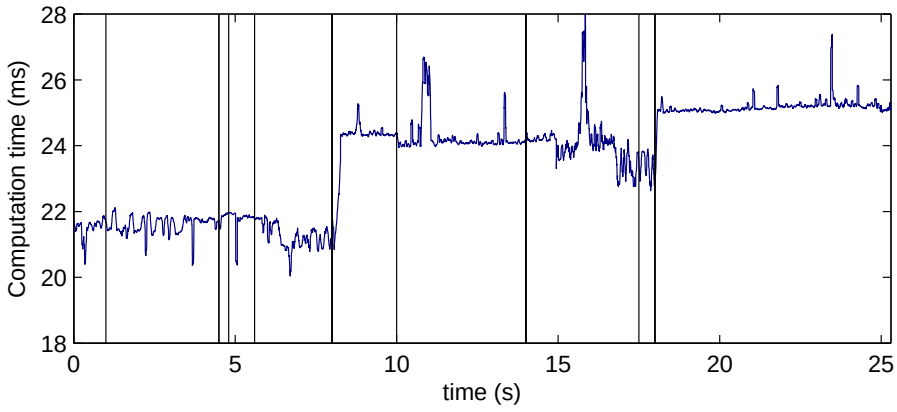


Figure 4.6 – Computation time for the robot sitting in an armchair

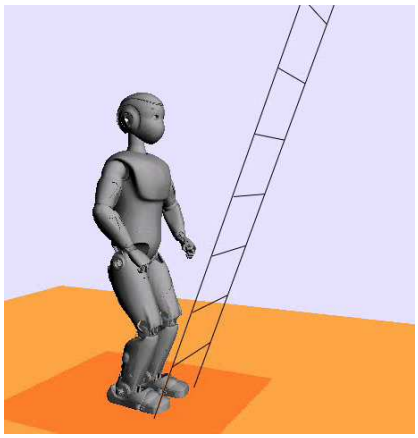


Figure 4.7 – Starting posture for Romeo

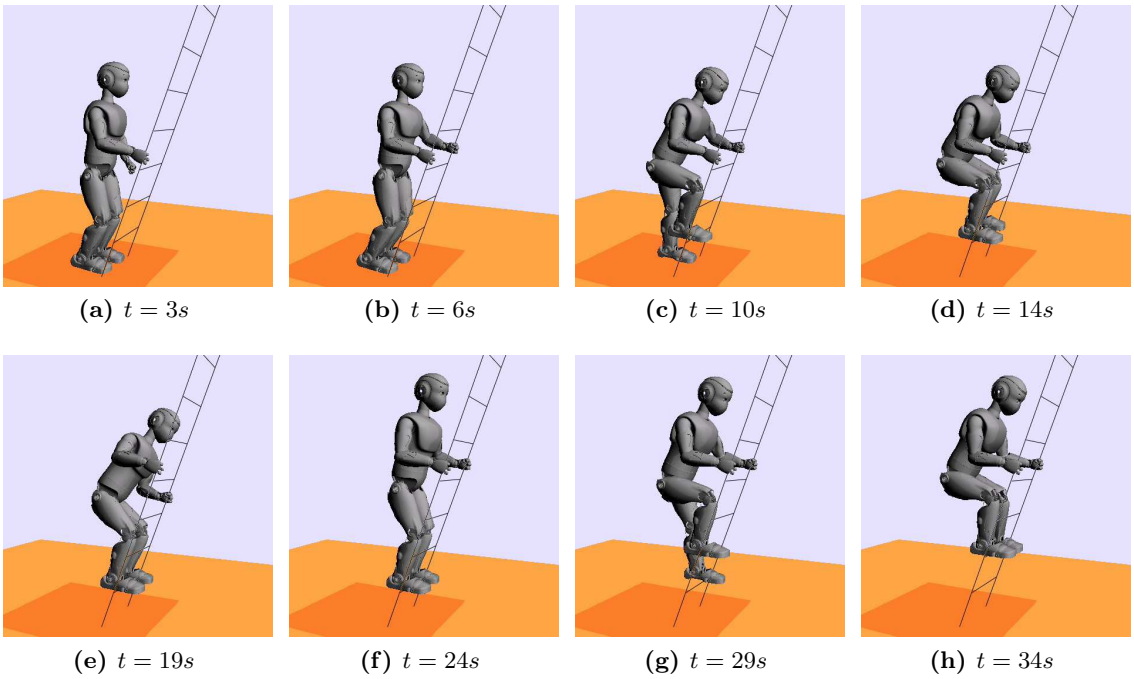
capability of the approach to handle multiple contacts. In this case, there are several additions and removals of contacts both for the hands and for the feet of the robot. The motion has been created based on a sequence of tasks assuming that the environment is completely known. If this is not the case, different sensors (typically cameras and LIDARS) can be added to get a map of the environment and to help the definition of the positions and orientations specified by the tasks. The robot starts with a standing posture where both feet are in contact with the ground as Fig. 4.7 shows. Then, the key postures of the rest of the motion are shown in Fig. 4.8.

The whole motion can be summarized as follows:

- First the robot approaches the right hand towards the ladder and grasps it, and then it does a similar motion with the left hand (Fig. 4.8a and Fig. 4.8b).
- Then, the robot raises its right foot and puts it on the first rung. The same motion is done for the left foot (Fig. 4.8c and Fig. 4.8d).
- After that, the hands are again moved towards an upper part of the ladder (Fig. 4.8e and Fig. 4.8f).
- Finally, the feet are taken to step on the second rung (Fig. 4.8g and Fig. 4.8h).

The motion was only generated for the first two rungs but it can be repeated for as many rungs as needed as long as the characteristics of the ladder remain homogeneous.

The inverse dynamics SoT described in Section 3.3.3 was also used to generate this motion. First the dynamic model of the robot and the contacts are specified as the highest level of the stack, and then the tasks are added. In this case, the contacts are added and removed more often than in the previously presented case. Once a contact is removed, a task is added to move that limb towards the desired target. The tasks are all specified as placement tasks given by (3.51) and will be called e_{lh}, e_{rh}, e_{lf}, e_{rf} where l refers to left, r to right, h to hand, and f to foot. The targets are specified in the operational space and an intermediate pose (a pre-grasping pose) is defined to avoid collisions and to have a better initial disposition for the final grasp (in case of the hand) or step (in case of the foot).



(a) $t = 3s$ **(b)** $t = 6s$ **(c)** $t = 10s$ **(d)** $t = 14s$

(e) $t = 19s$ **(f)** $t = 24s$ **(g)** $t = 29s$ **(h)** $t = 34s$

Figure 4.8 – Snapshots of Romeo in simulation climbing up a ladder. The timing is also provided.

4.2 Motion Imitation

This case study describes the integration of OSID SoT with a motion capture system to quickly produce some complex and dynamic movements that imitate the motion shown by a human performer. After the human motion has been acquired, a purely geometrical retargeting is applied and the obtained motion can be directly replayed on an anthropomorphic character, for example in the field of computer graphics if only visual resemblance is needed. This is due to the kinematic structure similarities between the humanoid robot and the human. However, this motion cannot be directly used for a robotic system since the human and the robot possess very different dynamics: the joint constraints and the mass distributions are not the same. Due to these differences, a motion that keeps the human balance can easily make the robot fall down. Thus, the inverse dynamics SoT is applied to introduce the dynamic constraints to the system while tracking the desired joint positions. Moreover, the task function approach is used to correct, enhance or edit some parts of the motion, or even to introduce some non-demonstrated features[2]. This section covers this proposed methodology in more detail.

4.2.1 Geometric Retargeting

Geometric retargeting consists in obtaining the desired trajectory $q^*(t)$ for the joints of the robot applying only geometric methods to the captured motion. Human motion can be acquired with a motion capture system and, although the methodology varies according to the

[2]This framework was introduced in: O. Ramos, L. Saab, S. Hak, N. Mansard, *Dynamic Motion Capture and Edition using a Stack of Tasks*, IEEE-RAS International Conference on Humanoid Robots (Humanoids), Bled, Slovenia, October 2011 [Ramos 11].

particular system, a typical and reliable way is using infrared cameras to track a set of markers located on the body as Section 4.2.3 discusses. Using a calibration process to solve the problem of shape and size differences between the robot and the human, the spatial trajectories defining the poses for the robot bodies can be obtained from the demonstrated motion. However, joint trajectories cannot be intuitively obtained due to the kinematic constraints imposed by the robot structure, such as joint limits, and thus geometric retargeting is used.

Geometric problem After a calibration process, the motion capture system provides the temporal evolution of the pose associated with each rigid body constituting the robot. The frame for each body is located at its associated joint, and the pose of each robot joint q_i is represented as the transformation matrix ${}^{W}T^{*}_{q_i}(t) \in SE(3)$, where W refers to the robot world reference frame (an example of this matrix is given in Section 4.2.3). To find the trajectories for the robot joints in the configuration space, the forward kinematic model, also known as the forward geometric model, will be used. For a specific joint configuration q, this model provides the pose of the frame associated with each joint q_i as the transformation ${}^{W}T_{q_i}(q) \in SE(3)$. Then, the geometric retargeting problem consists in finding the configuration q that minimizes the difference between ${}^{W}T_{q_i}(q)$ and the desired transformation matrix ${}^{W}T^{*}_{q_i}(t)$ at time t. For an n DoF robot, this problem can be written as:

$$q^{*}(t) = \arg\min_{q} \sum_{i=1}^{n} {}^{W}T^{*}_{q_i}(t) \ominus {}^{W}T_{q_i}(q) \qquad (4.1)$$
$$\text{s.t.} \qquad \underline{q} \leq q \leq \overline{q}$$

where $\underline{q}$, $\overline{q}$ are the lower and upper angular joint limits respectively, and $\ominus$ is a distance operator in the $SE(3)$ group which should penalize the differences in position and orientation between the two transformation matrices.

Distance Operator The operator used in this work weights the norms of the errors as follows. Let the position and orientation components of the transformation matrix ${}^{W}T^{*}_{q_i}(t)$ be $p^{*}_{q_i}(t)$, $R^{*}_{q_i}(t)$, and of ${}^{W}T_{q_i}(q)$ be $p_{q_i}(q)$, $R_{q_i}(q)$, respectively. The position error is given by the L^2 norm of the difference between the position components: $\parallel p^{*}_{q_i}(t) - p_{q_i}(q) \parallel_2$. For the orientation error, the difference is measured by θ_i, which is the angle corresponding to the axis-angle representation of the product $R_{d_i} = R_{q_i}(q)R^{*}_{q_i}(t)^{-1} \in SO(3)$. Let R_{d_i} be represented by

$$R_{d_i} = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix}$$

then, the angle $\theta_i \in [0, \pi]$ will be given by

$$\theta_i = \text{atan2}\left(\frac{\sqrt{(n_y - o_x)^2 + (n_z - a_x)^2 + (o_z - a_y)^2}}{2}, \frac{n_x + o_y + a_z - 1}{2} \right).$$

When the orientations are similar, the rotation matrices are close enough, R_{d_i} is close to the identity, and the angle θ_i of R_{d_i} about the arbitrary axis is small. Using the errors for the position and orientation, the operator $\ominus$ in (4.1) is given by:

$${}^{W}T^{*}_{q_i}(t) \ominus {}^{W}T_{q_i}(q) = w_{p_i} \parallel p^{*}_{q_i}(t) - p_{q_i}(q) \parallel_2 + w_{o_i}\theta_i \qquad (4.2)$$

where w_{p_i} is the weight corresponding to the position error for joint i and $w_{o_i} = 1 - w_{p_i}$ is the weight for the orientation error. The weights provide more flexibility and their value can be experimentally determined for each part of the robot.

Limitations Experimentally, the optimization in (4.1) has been robust enough to handle the approximations done from the human motion acquisition to the robot modeling. In the field of computer graphics, this motion can be used for an anthropomorphic character when the only objective is to obtain a motion similar to the demonstrated one. But, since only geometric methods have been used to generate $q^*(t)$, the reconstructed joint trajectories do not consider any dynamic properties of the robot and their direct usage might present problems such as balance loss or auto-collisions. These problems are solved using the inverse dynamics SoT as the following step to track the desired joint configurations while ensuring a sufficient recovery of the noise and inaccuracy as well as a generation of feasible dynamic motion for the humanoid robot.

4.2.2 Dynamic Retargeting

The joint trajectories obtained by pure geometric retargeting are not dynamically consistent. In particular, there is no guarantee that the robot is stably balanced, or that auto-collisions do not appear. Moreover, some important aspects of the original motion can be damaged by the previous retargeting, which was obtained as a trade-off between the positions of all the bodies. If a given body is more relevant than the others, this importance is not reflected in the obtained motion because of the differences between the two kinematic chains. For example, if both hands are clamping in the demonstration, their resulting positions after a pure geometric processing are not likely to satisfy the clamp. *Dynamic retargeting* is a way to solve these problems through the usage of the inverse dynamics SoT which enforces dynamic consistency while following some joint trajectories, and enables an easy edition of the resulting motion.

General Framework

The general scheme used for the dynamical retargeting is depicted in Fig. 4.9 which shows the inverse dynamics SoT as the main component for motion generation. As described in Section 3.3.3 and Section 3.3.4 (for the decoupled system) the highest priority components of the SoT are the enforcement of the dynamic consistency and the corresponding contact model. This also guarantees the local balance of the robot: all the contact points remain stable since their associated force remains unidirectional. From the demonstration, the contact points are extracted by detecting clusters of static points in the feet trajectories. After the dynamic constraints, the joint limits of the robot (Section 3.4.2) are enforced in the stack, and then, other robot constraints are added to generate the desired motion. The parts that are used as input to the stack in Fig. 4.9 are the following.

- *The Posture Task.* The basic element for motion imitation is shown at the top of Fig. 4.9 as a posture task (Section 3.4.1), whose function is to track at best the reference joint configuration $q^*(t)$, coming from the motion capture after the geometric retargeting step. The posture task can be applied to some parts of the robot or to the whole body. For more freedom in the control it is usually applied to each component of the chain

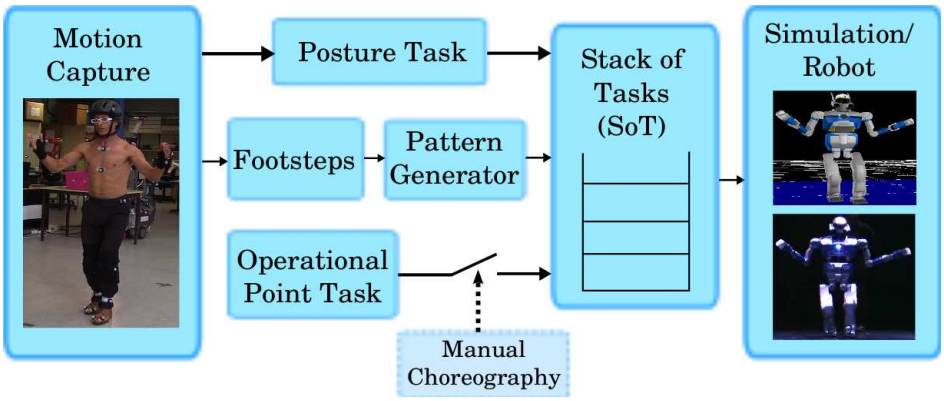


Figure 4.9 – Dynamical retargeting scheme. Motion capture information passes to the SoT through some tasks. Tasks for some operational points can be manually activated according to the choreography. The motion is finally executed in a dynamic simulator or the real robot.

(arms, legs, etc.) separately. The gain of the task (3.50) can also be different for each component.

- *The Operational Point Task.* Some characteristics of the demonstrated motion can be lost in the geometrical retargeting and in the posture task. To recover these characteristics, placement tasks (Section 3.4.1) can be added on arbitrary operational points to enforce the robot to more precisely track some part of the demonstration by following the exact operational-space trajectory of the human performer. These tasks are shown as "Operational Point" tasks in Fig. 4.9 and are the basis for the motion edition capabilities of the framework. They can be manually specified to add some key aspects to the choreography, a process referred to as "manual choreography". The important aspects of the choreography can also be automatically extracted using a frequency analysis [Calinon 07], studying the motion model of the human [Hak 12], or simply using the operational space of the acquired motion.

- *The Walking Pattern Generator (WPG).* When the motion is composed of some steps, these can be detected by the analysis of feet trajectory motion. Then, these footsteps can be used as input to a WPG which outputs the trajectory for the CoM, the legs and the waist. These trajectories can then be added to the SoT for walking while performing some other motion with the rest of the body. The walking part of the motion will typically need tasks to track each of the outputs generated by the WPG.

The inverse-dynamics SoT can then handle the different priorities of the previous tasks, if specified, and it generates an output that can be directly used as a control signal for the humanoid robot.

Edition of the Imitated Motion

The posture task in Fig. 4.9 reproduces the desired motion at the joint level satisfying the dynamic constraints. However, the PD used for this task (Section 3.4.1) acts as a low-pass filter generating undesired movements at some points or erasing some delicate or very dynamic movements, typically due to fast oscillatory motions. Nevertheless, the structure of the SoT

can be used to overcome these problems by adding operational point tasks, as explained above, which can enhance or even modify the original motion. For a given operational point x_i, a selection matrix S_x can be defined to enable and disable the control of some degrees of freedom of $SE(3)$ to give more freedom to the task. For example, if the position is represented by Cartesian coordinates and the orientation by roll, pitch and yaw angles, then $x_i \in \mathbb{R}^6$ and the selection matrix is given by

$$S_x = \begin{bmatrix} s_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & s_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & s_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & s_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & s_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & s_6 \end{bmatrix} \tag{4.3}$$

where s_j is a binary element whose value can be either 1 or 0. If it is 1, that particular element of the pose will be controlled, and if it is 0, the corresponding element will be unconstrained and its motion will be a result of the whole body dynamics. In general, the motion obtained with the posture task can be edited with the operational point task in two ways.

- *Specification of target points.* A new desired target for a chosen operational point can be specified without defining the desired trajectory to reach it. This point can be determined using forward kinematics on an operational point to compute its position from the geometric retargeting, or it can be arbitrarily set.
- *Specification of a trajectory.* Let the trajectory for the operational point x be called $x_o(t)$. The new trajectory that will be set as the desired trajectory for the operational point will be $x_n(t) = x_o(t) + x_m(t)$, where $x_m(t)$ is the trajectory modification that can be done on any of the six degrees of freedom of x. This trajectory modification $x_m(t)$ can be time varying or constant, according to the requirements.

It is important to point out that the operational task must have higher priority than the posture task it would interfere with. For instance, if an operational task is added to the hand, the priority of the arm posture task must be reduced. Alternatively, the task can be removed, but it is preferred to be kept, as it will serve as a "guide" for the new trajectory. If these priority considerations are not taken into account, the desired motion would be blinded by the solutions satisfying the posture task with higher priority, and the desired effect would not be achieved. Additional tasks, such as a task for the gaze, can also be added to modify the original motion.

4.2.3 Experimental Setup

A motion capture system, which provides the spatial trajectory for each of the markers that are distributed on the human body, is used to acquire the motion performed by a human demonstrator. The markers are located in characteristic parts of the body, as Fig. 4.10 shows [H. 11], in order to minimize the motion of the skin with respect to the bones, and to facilitate their temporal tracking. For example, they must be placed as close as possible to the center of the articulations to avoid undesired translations when joints rotate, and the configuration should not be symmetric so that the temporal tracking of the markers is not perturbed. If

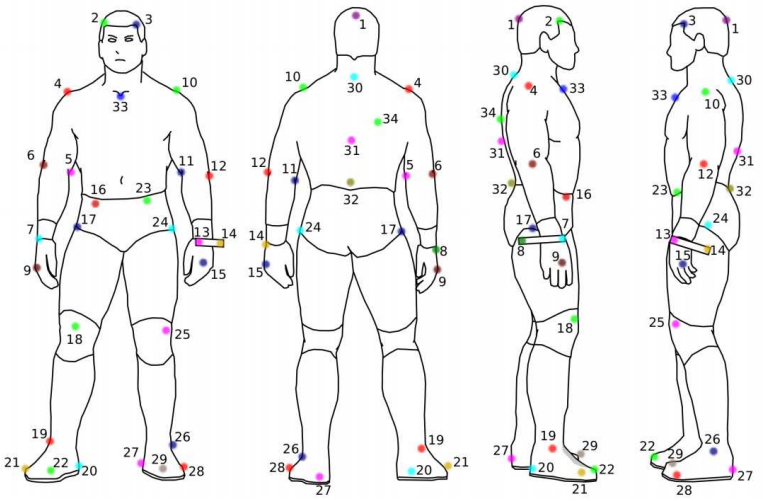


Figure 4.10 – Position of the markers in the human body and calibration position

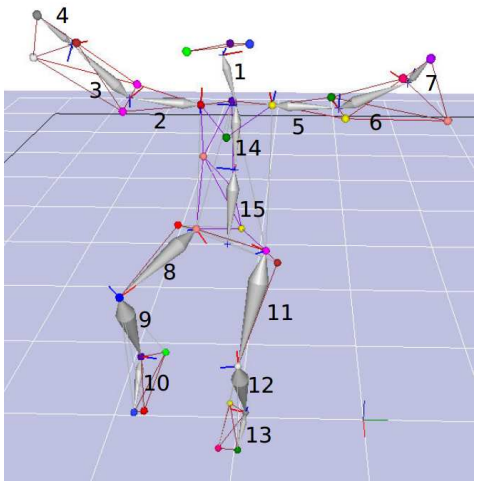


Figure 4.11 – Skeleton showing the rigid bodies and the markers associated to them

there is symmetry, some ambiguity might appear leading to problems in markers identification. Due to this reason, in Fig. 4.10 a small plank is used at each wrist so that the three markers located at the hand do not form an equilateral triangle and are easier to track. In this work, 35 markers were used.

Skeleton The markers are then manually associated to form a skeleton which is composed of a hierarchy of rigid bodies in the form of a tree. These rigid bodies are sometimes referred to as links or bones of the skeleton and each of them has a frame that is defined by three markers. The skeleton used in this work, shown in Figure 4.11, consists of 15 rigid bodies and is the one proposed in [H. 11]. It should be noted that only after the skeleton is generated, the position and orientation of the rigid bodies composing it can be obtained.

Figure 4.12 – Calibration position and markers in the body of the human dancer

Nodes The starting point of each unconstrained rigid body composing the skeleton is called a *node* and, if properly located, is equivalent to a joint. The node associated with the i^{th} bone will be represented as m_i. The acquisition system provides the pose of each node m_i in the frame of the motion capture system $\{M\}$, which will be represented by the homogeneous transformation matrix ${}^{M}T_{m_i}(t) \in SE(3)$. It should be noted that ${}^{M}T_{m_i}(t)$ purely corresponds to the human and contains no information about the robot geometry.

Calibration The link between the human and the robot configurations can be obtained through calibration since the kinematic structures of both the humanoid and the human are similar (wrists, elbows, shoulders, etc.). To this end, the person starts with a position that is well known for the robot, as Fig. 4.12 shows. Using this configuration, the transformations between each node m_i in the human skeleton and the corresponding joint q_i in the humanoid robot are obtained from a classical "calibration" step. They are represented by ${}^{m_i}T_{q_i} \in SE(3)$, which includes the differences in orientation between the frames of the nodes and the frames of the joints (defined in the robot kinematic model), as well as the differences in the segment lengths of the robot and the human. These transformations remains constant as long as the markers do not have relative motion with respect to the body they are attached to. Another constant matrix is the one relating the origin of the motion capture system $\{M\}$ to the robot reference frame $\{W\}$ represented by ${}^{W}T_{M} \in SE(3)$.

Demonstrated Motion Using these transformations and the data obtained from the motion capture system, the time-varying matrix ${}^{W}T^{*}_{q_i}(t)$ that defines the position and orientation of the i^{th} robot joint in the space is

$$ {}^{W}T^{*}_{q_i}(t) = {}^{W}T_{M}\ {}^{M}T_{m_i}(t)\ {}^{m_i}T_{q_i} \tag{4.4} $$

which constitutes the demonstrated motion in terms of the poses of the robot links. The matrix in (4.4) is the desired robot posture and is used as the objective of the geometric retargeting step.

Motion Capture System For this particular experience, the human motion was acquired using the *Motion Analysis* (http://www.motionanalysis.com/) motion capture system consisting of ten infrared cameras distributed around the experimentation zone and calibrated by the *Motion Analysis* software. The acquisition frequency was 200 Hz with a precision of 2 mm of nearly pure white noise and negligible bias. More than thirty minutes of motions were captured. The captured data for the most important movements used in this work is freely available[3] along with the corresponding geometric retargeting and the results of the dynamic retargeting that uses the inverse-dynamics approach, explained in Sections 4.2.1 and 4.2.2.

4.2.4 Robot Dancing Simulation

The motion of a person performing a pop dance was acquired using the motion capture system, as Fig. 4.13a shows. During this motion the dancer keeps the feet almost at the same position and most of the movement is done by the upper body and the legs. The acquired motion was retargeted to a modified HRP-2 model, and it was edited to correct the retargeting error and to introduce new undemonstrated features. The model of HRP-2 was modified by adding one extra degree of freedom both at the chest and the neck joints with the purpose of obtaining a model closer to the human body and more suitable for the dance imitation. The kinematical structure thus obtained contains 32 DoF and resembles more the structure of HRP-4 than that of HRP-2. For this reason, these results have only been implemented in simulation.

The initial trajectory for the joints is obtained from the captured motion using the geometric retargeting, but it is neither stable nor dynamically consistent as Fig. 4.13b shows. Then, the dynamic retargeting is applied keeping both feet in contact with the ground at all moments but only the posture task is added to the SoT to track the joint trajectories; the results are shown in Fig. 4.13c. The obtained motion is stable but some problems appear like an auto-collision between the right hand and the head shown in the fourth thumbnail. Also, the geometric and dynamic retargetting have lost some data and produced some errors compared to the initial demonstrated trajectory. For this particular motion, three editions were applied:

- The right hand motion was corrected to avoid auto-collisions.
- The knee oscillations (smoothed by the PD in (3.50)) were enhanced.
- An additional motion of the left foot (sliding), not present in the initial demonstration, was introduced.

The sequence of tasks showing the instants of time when these modifications were added is presented in Fig. 4.14 and the final motion is shown in Fig. 4.13d. More details about these three editions are provided below.

Right hand motion

The fast up and down motion of the dancer's right arm was smoothed as a consequence of the PD used in the task that tracks the joints. This was especially noted when the arm could not reach the upper positions that the dancer performed. To correct this effect, an operational task to raise more the arm was introduced. The result is shown in Fig. 4.15.

[3] `http://projects.laas.fr/gepetto/novela/noveladb`

(a) Motion performed by the dancer

(b) Motion obtained with the geometric model

(c) Motion after the posture task

(d) Final motion

Figure 4.13 – Results for the robot imitating the dance performed by a human.

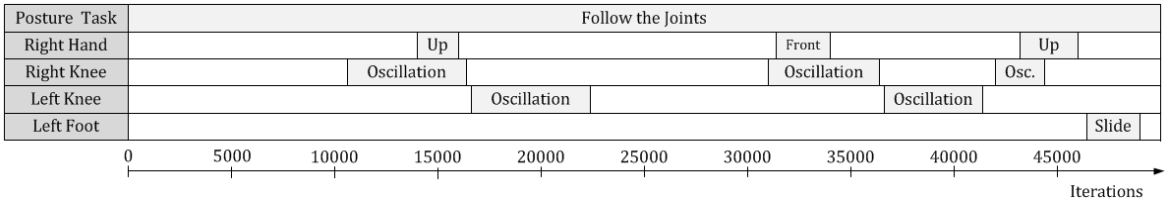


Figure 4.14 – Task sequence for the dance simulation

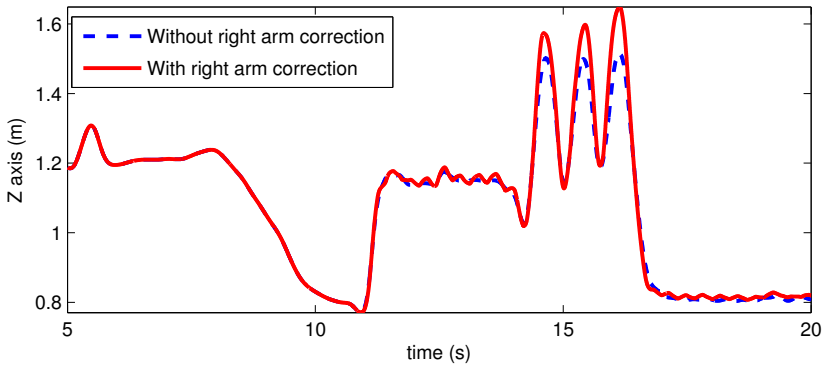
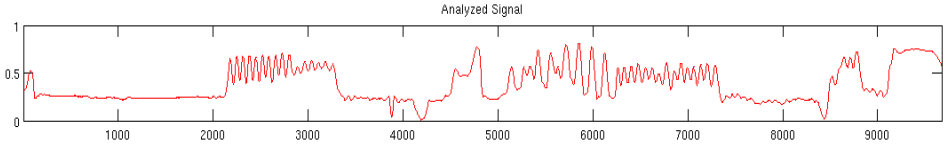


Figure 4.15 – Right hand evolution in the operational space (Z axis).

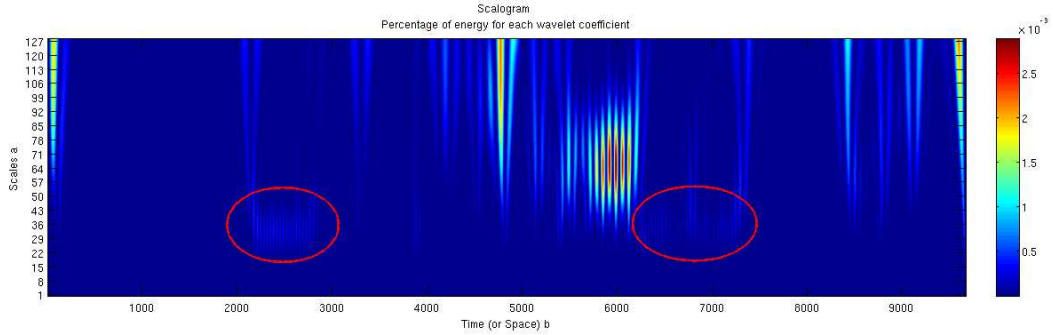
The trajectory of the right arm in the Z axis using only the posture task to track the joints is shown in blue, whereas the trajectory with the operational space correction is shown in red. The corrected trajectory improved the upper positions of the right hand. This can be achieved thanks to the hierarchical approach of the framework: the operational task was set with higher priority than the posture task for the right hand, but the posture task is still naturally able to guide the motion. Another task in the right hand was also used to avoid the auto collision of the hand with the head, shown in the fourth thumbnail of Fig. 4.13.

Knee Oscillation

The knees constitute a particular case as the dancer permanently moved them but at the dynamic level this motion was strongly weakened. The reason is the same as for the hand motion, but in this case the effect is even more pronounced: there was an oscillatory and relatively fast motion of the knees that the PD posture task smooths and is not able to follow. To correct this problem, the motion of the knee joint was analyzed and Fig. 4.16a shows the joint evolution obtained after the geometric retargeting for the right knee. It can be seen that between iterations 2000 and 2800, and between 6200 and 7300, the motion of the joint is oscillatory, which corresponds to the observed motion at the dancer's right knee. A scalogram using the Gaussian wavelet was constructed to determine the frequency of the oscillation and it is shown in Fig. 4.16b, where the red circle points out the salient frequencies corresponding to the oscillatory motion identified in Fig. 4.16a. From the scalogram, the scale a corresponding to the maximum values at the desired positions was determined to be 36. The relation between the frequency and the scales is $f = \frac{f_s f_w}{a}$, where f_s is the sampling frequency and f_w is the center frequency of the wavelet [Abry 97]. For the Gaussian derivative of order 4, $f_w = 0.5$, and considering that the sampling frequency used during the acquisition is 200



(a) Temporal evolution of the right knee joint in radians.



(b) Scalogram of the right knee joint

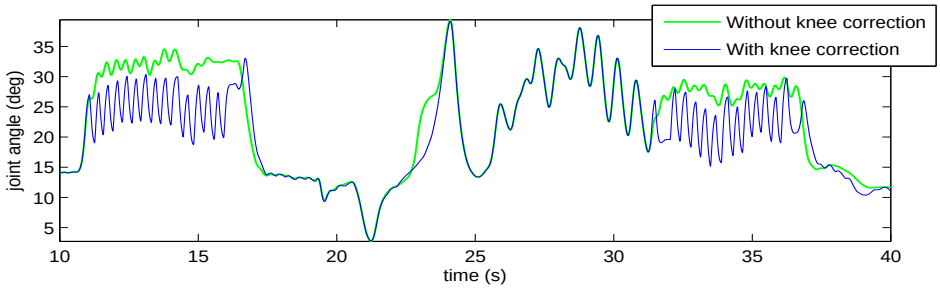
Figure 4.16 – Analysis of the right knee joint motion

Hz, the resulting frequency of the oscillation is 2.7 Hz. Then, a task on the knee was added at that frequency to correct the low-pass filter effect obtained before.

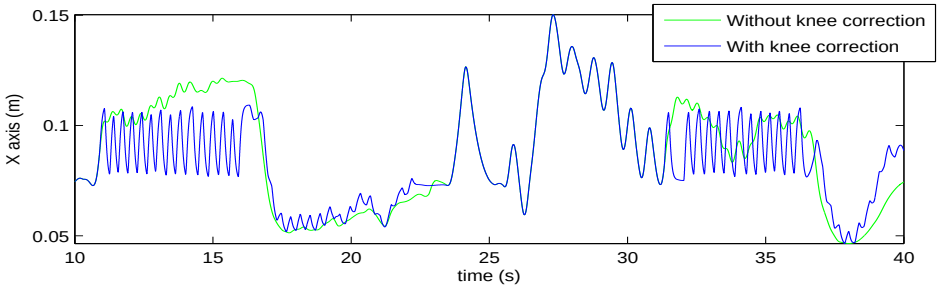
The evolution of the right knee joint in the joint space is shown in Fig. 4.17a, and the evolution in the operational space corresponding to the x axis of the knee (pointing forwards) is shown in Fig. 4.17b. The green curve shows the evolution of the joint when only the posture task for the leg is used, and the blue curve shows the evolution when the operational task is added to the knee. Even though the joint angular value is not directly controlled, Fig. 4.17a shows that the joint presents an oscillation with higher amplitude as a consequence of the addition of the operational task. The task space in the forward direction shows a consistent oscillation, which was the desired effect.

Foot Sliding introduction

This constitutes an undemonstrated feature that was arbitrarily introduced to improve the appearance of the choreography by the end of the motion. To this end, the kinematic constraint for the contact was relaxed so that the motion of the foot is restricted to lay anywhere in the horizontal plane but avoiding to leave the ground or to penetrate it. Letting the velocity components of the contact point be $\dot{x}_c = (v_{c_x}, v_{c_y}, v_{c_z}, \omega_{c_x}, \omega_{c_y}, \omega_{c_z})$, where v_c is the linear velocity, and ω_c the angular velocity, the constraint can be expressed as $v_{c_z} = 0$, $\omega_{c_x} = 0$ and $\omega_{c_y} = 0$ to avoid leaving the ground and rotating the foot in undesired directions (assuming that the z axis is oriented upwards and the x, y axes are horizontal). This constraint can be expressed as a placement task (3.51) where only some position and orientation components are controlled using a selection matrix (4.3). Thus, the desired pose of the contact point x^*



(a) Joint space of the knee



(b) Task space (x axis) corresponding to the forward direction

Figure 4.17 – Temporal evolution of the right knee

and the selection matrix S_x can be specified as

$$x^* = \begin{bmatrix} k_x \\ k_y \\ H_z \\ 0 \\ 0 \\ k_{rz} \end{bmatrix} \qquad S_x = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.5}$$

where k_x, k_y, k_{rz} are arbitrary values (which are not controlled), and H_z is the known height of the foot with respect to the ground. However, the task for foot sliding does not exactly correspond to a sliding motion in a broad sense, since sliding can accept forces orthogonal to the motion. The proposed solution is thus more restrictive than necessary. However, although forces are not considered, the visual effect is similar, and the motion is feasible and dynamically consistent.

In the experiment, an undemonstrated sliding movement of the left foot was arbitrarily introduced to prove that extra features can be added as desired. To introduce this feature, the ZMP of the motion when the dynamic retargeting was applied without this particular feature was analyzed. Fig. 4.18 shows in blue the trajectory of the ZMP in the Y axis. The red lines show the boundaries of the feet (the maximum and minimum limits constitute the support polygon in the Y axis). Between times 46.8 and 48.3, the ZMP completely lies in the area corresponding to the right foot. Then, the contact was removed and the sliding task was introduced for the left foot at that time, guaranteeing the robot stability, as the sliding foot cannot be considered part of the support polygon anymore.

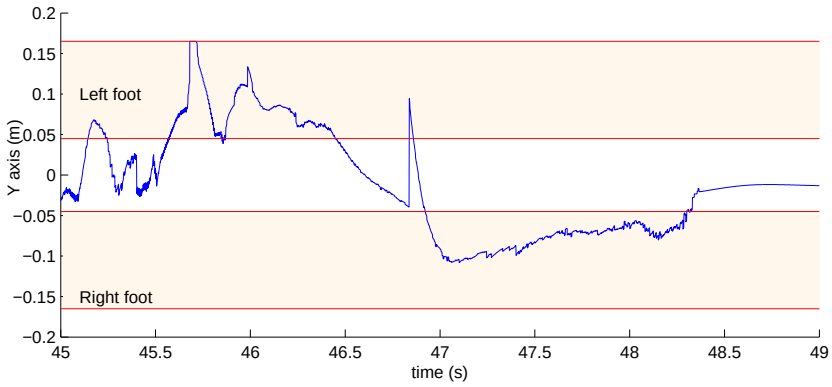


Figure 4.18 – Evolution of the ZMP with only the posture task

4.2.5 The Yoga Figure

The *standing lotus* yoga motion was the first movement achieved in the real HRP-2 using the presented framework for motion imitation[4]. The motion starts in a double-support rest position. A sweeping motion of both arms is then executed while reaching a stable single-support posture with the free foot close to the support knee and both hands joined in front of the chest. This demonstrated movement is summarized in Fig. 4.19a. The execution of the geometrically-retargeted motion in a kinematic simulator (ignoring the dynamics of the system) is shown in Fig. 4.19b. The motion globally resembles the demonstrated pattern but some problems appear such as the collision of the hands with the chest of the robot. The reason is the different kinematic structure between the human and the robot. Moreover, the application of this motion with a realistic (dynamic) simulator, shows that the robot does not reach a stable single-support posture but falls down (Fig. 4.19c) due to the different dynamic structures of the robot and the human: the mass distribution is very different, for instance, the legs of the robot are much heavier. Thus, posture alone does not ensure robot balance. Also, the hands are colliding with each other and with the chest, and the flying foot is badly positioned.

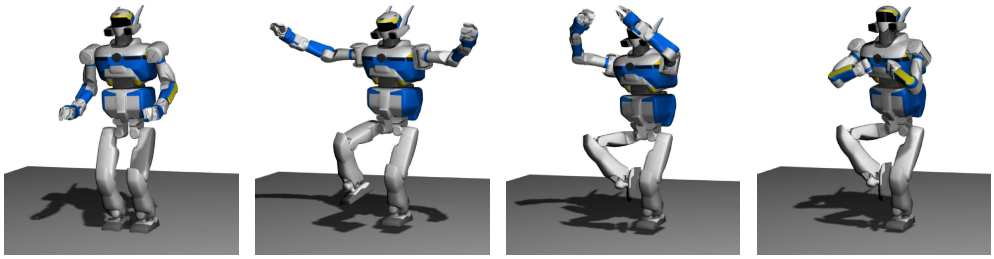
To guarantee the balance of the robot, the dynamics has to be considered and therefore, the dynamic retargeting is applied to track the demonstrated posture and to ensure balance while keeping key features. The important features of the demonstrated movement are: the balance *i.e.* the position of the CoM at the balanced position; the relative position of the hands and the chest; and the position of the free foot. To keep these features, the following tasks were used in addition to the posture task that follows the joint trajectories.

- Three operational space tasks were added to control both hands and the free foot, respectively. They follow the demonstrated trajectories of the corresponding human-body points. The tasks for the hands are also added to avoid auto-collision with the chest.

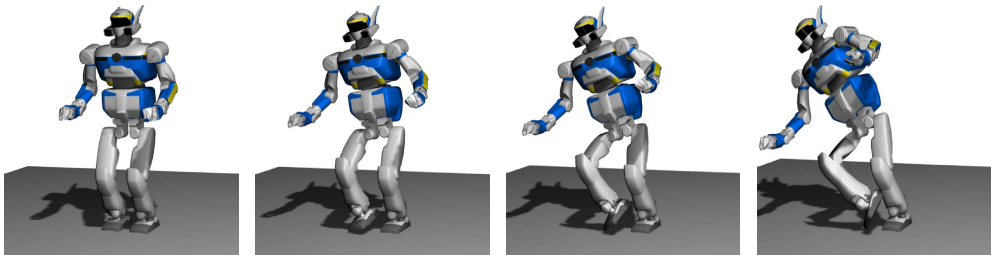
- Another task was added for the CoM. The CoM is more difficult to observe on the human, but it can be estimated by the waist position or using an inertial model with

[4]The experiments with the real robot appear in: O. Ramos, N. Mansard, O. Stasse, S. Hak, L. Saab, and C. Benazeth: *Dynamic Whole Body Motion Generation for the Dance of a Humanoid Robot*, IEEE Robotics and Automation Magazine (RAM), 2015 (in press) [Ramos 15].

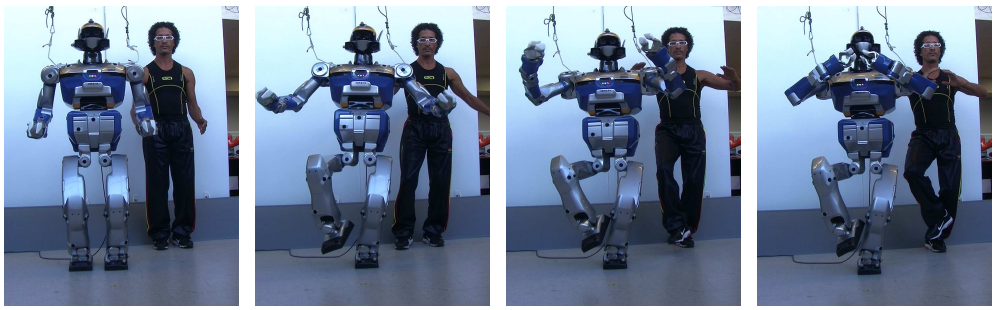
(a) Original demonstrated motion



(b) Geometric retargeting of the motion (using a kinematic integrator)



(c) Geometric retargeting of the motion (using a dynamic simulator): balance is not kept



(d) Dynamic retargeting motion executed by the robot: the robot can keep its balance.

Figure 4.19 – Yoga motion: from the demonstration to the real HRP-2

(a) (b) (c) (d)

Figure 4.20 – Motion acquisition of a human dancer

the correct mass distribution. However, even if the human CoM is recovered, it cannot be guaranteed that it generates stable motions in the robot due to the mass difference. Thus, an artificial pattern for the CoM was imposed to enforce balance since CoM accelerations might destabilize the robot and the approach cannot predict those states. The pattern specifies that the CoM lies inside the support polygon and is obtained by experimentally determining the timings for the change in the supports of the human performer.

The SoT is finally composed of the dynamics and contact constraints, a task tracking the CoM artificial pattern, a task tracking the demonstrated right hand, left hand and right foot, and a task tracking the geometrically-retargeted configuration. The task tracking the free foot is added only when the foot leaves the ground. The motion executed by the robot is shown in Fig. 4.19d.

4.2.6 Long-sequence Motion Generation

The same generic task sequence described above and used for the yoga movement can be applied to treat several motion sequences automatically. In particular, the motion corresponding to a long choreography performed by a hip-hop dancer was acquired. Fig. 4.20 shows some typical patterns of the movements demonstrated by the human dancer (with the markers on his body) in the motion capture acquisition room: motion in double support (Fig 4.20a and Fig. 4.20b), motion in single support (Fig. 4.20c), and motion of the upper body while walking (Fig. 4.20d). In this acquisition, the small plank for the hand (Section 4.2.3) was not used since the demonstrator's hands were big enough to properly locate the markers avoiding the undesired acquisition symmetries.
From the choreography performed by the human, a choreography was designed for HRP-2. This choreography was presented in October 2012 in a live demonstration in front of more than 1000 people in the city of Toulouse. The demonstration consisted in both the robot and the human performing the movements side by side as Fig 4.21 shows.
Three of the typical execution patterns performed in the demonstration, consisting in double support motion, single support motion, and motion while walking, are detailed below.

Figure 4.21 – HRP-2 and the human dancer in the live demonstration.

Double Support Motion

The first case consists in moving the whole body fast while keeping both feet in contact with the ground. Even though some parts of the body move fast and might generate some undesired angular moments, the proposed method keeps the robot balance by automatically compensating with the appropriate control. When specifically asked by the choreographer, a task was added for the hand, but no task is added for the CoM. Two examples of this type of motion are shown in Fig. 4.22. The upper case (Fig. 4.22a) shows the robot moving the arms fast while keeping its balance, and the lower case (Fig. 4.22b) shows the robot lowering its body down while also moving the arms fast. The latter motion resembles some movement performed in martial arts.

Single Support Motion

A second type of motion is a single support one, where the robot moves the whole body while keeping a single leg in contact with the ground. Since the uncontrolled motion of the CoM can generate high accelerations of the CoM near the edge of the support polygon, and the methodology cannot foresee future states to avoid these cases, a task is used to guide the CoM within appropriate limits. Like for the case of the yoga, an artificial pattern for both reaching a stable single-support position and going back to a double-support stance is used as guide for the CoM. Two examples of this motion are shown in Fig. 4.23. The first example shows the robot raising its left leg while moving the arms (Fig. 4.23a), and the second example shows the robot raising the right leg and doing a 'circumference' over the ground with it while keeping the arms still (Fig 4.23b).

Dynamic Walk

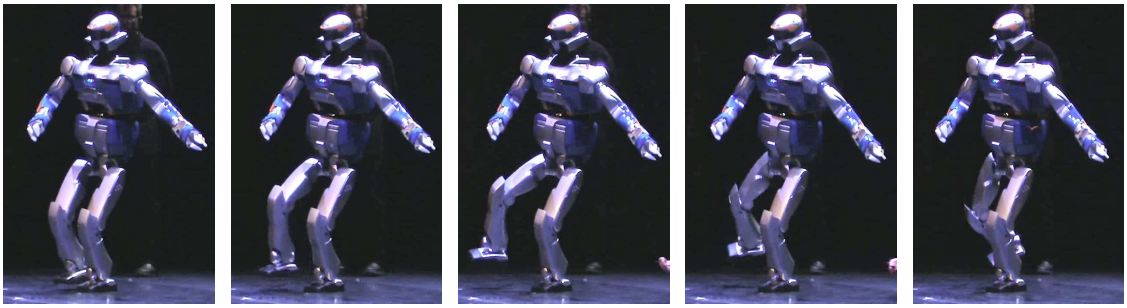
In the third case, the movement consists in walking while moving the upper body at the same time. This presents a particular problem when retargeting the motion to the robot since the acquired motion kept dynamic balance for the human, but it does not keep either dynamic or static balance for the robot due to the differences in mass distribution. A single posture task to track the acquired joint trajectories is not sufficient and would make the robot fall down.

(a) Example 1: fast arms motion

(b) Example 2: lowering down the upper body

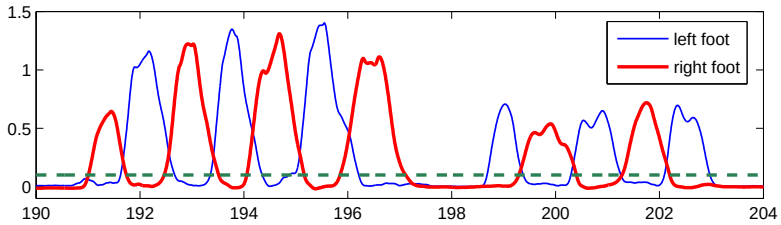
Figure 4.22 – Motion of the robot in double support

(a) Example 1: raising the left leg upwards

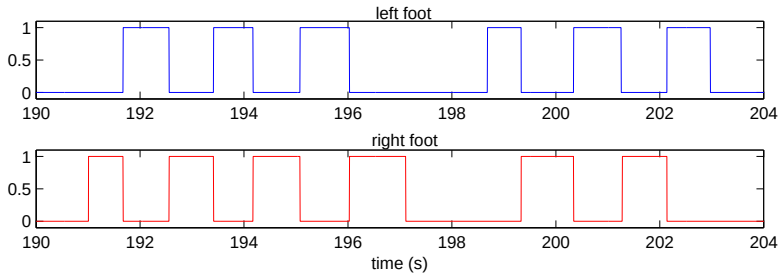


(b) Example 2: raising the right leg and moving it

Figure 4.23 – Motion of the robot in single support



(a) Function $f_{sp}(t)$.



(b) Timing for the footsteps

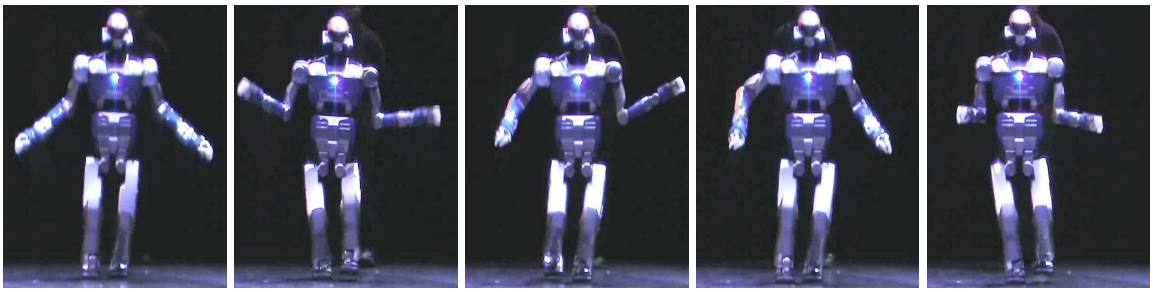
Figure 4.24 – Determination of the footsteps that will be used for the WPG

In order to generate dynamic walking for the robot a WPG, whose steps were specified based on the footprints coming from the demonstrated motion, was used. Then, the lower body followed the trajectories given by the WPG while the upper body tracked the demonstrated motion.

The positions of the footprints and their timings are extracted from the demonstration by grouping clusters of points on the foot trajectories. Let $r_l = (x_l, y_l, z_l)$ and $r_r = (x_r, y_r, z_r)$ represent the trajectories of the left and right foot, respectively, coming from the acquired motion. To automatically obtain the footprints, a 'heuristic' function $f_p(t)$ was defined to specify the amount of motion in each foot at every instant of time as:

$$f_p(t) = \alpha_x \frac{\|\dot{x}_i\|}{\max\{\|\dot{x}_i\|\}} + \alpha_y \frac{\|\dot{y}_i\|}{\max\{\|\dot{y}_i\|\}} + \alpha_z \frac{\|\dot{z}_i\|}{\max\{\|\dot{z}_i\|\}} + \alpha_h \frac{zs_i}{\max\{zs_i\}} \tag{4.6}$$

where α_x, α_y, α_z, α_h are weights, zs_i is the height z_i smoothed by a mean filter, and i can be l or r depending on whether the measure is for the left or right foot. The derivatives of the position determine the horizontal and vertical motion of the feet, whereas the value of z determines the height over the ground. The motion components were added for more robustness, since noise in the measurements did not allow to reliably discriminate between steps when only using the height z. The values of the weights are experimentally determined, and for these tests the values used were $\alpha_x = \alpha_y = \alpha_z = 1$ and $\alpha_h = 0.5$. The function $f_p(t)$ was then smoothed with a mean filter to remove the remaining noise, leading to the function $f_{sp}(t)$ which is shown in Fig. 4.24a for the instants of time corresponding to a dynamic walking. After $f_{sp}(t)$ is found, an empirically determined threshold is applied to remove the part that does not correspond to a step. In this case, the threshold was at 0.1, as Fig. 4.24a shows with a green line. The parts above the threshold correspond to demonstrated steps, and their detection is shown in Fig. 4.24b where 0 implies that the foot is on the ground and 1 implies that a step is taking part.



(a) Frontal view

(b) Lateral view

Figure 4.25 – Walking while arbitrarily moving the upper part of the body

The footprints are then used as input to a WPG [Kajita 03a] to compute a walking trajectory for both the feet and the CoM. The CoM task is added as the top-priority task, while a task is added to drive the position of the flying foot during each step. The posture task is added for the upper body and additional tasks are eventually added to improve the hand or the head placements depending on the choreography. The hands produce an important momentum, especially when the choreography imposes some movements that are contrary to the natural walking motion. The momentum is however corrected by the whole-body inverse-dynamics scheme and does not disturb the balance. Fig. 4.25 shows an example of this type of motion. It can be observed that the robot walks dynamically while moving its arms with arbitrary motion that follows the demonstrated movements.

The captured and experimental joint trajectories for two typical joints (at the shoulders) is shown in Fig. 4.26. It can be noticed that the trajectories obtained with the dynamic control follow closely the desired ones, but the control also acts as a 'low pass filter'. This attenuation can be controlled by changing the gain of the task. If the gain is larger, the joints move faster and the trajectory is closer to the desired one. However, the gain cannot be arbitrarily big since it is eventually limited by the mechanical constraints of the robot such as the maximum velocity of the joints.

4.2.7 Conclusion

The methodology presented for imitation allows to quickly and efficiently generate long sequence of dynamic movements for a humanoid robot. The motion generation is based on a combination of two very efficient tools: the motion capture, and the hierarchized operational-space inverse-dynamics, which enables the retargeted dynamics to fit with the constraints of

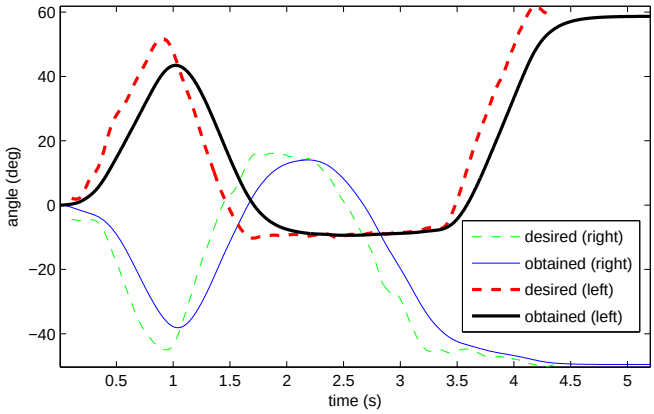


Figure 4.26 – Desired and obtained joint trajectories for the right and left shoulders

the real robot. The method produces reliable movements on the robot. In particular, these movements were executed without any security crane in public performances with a human dancer in the city of Toulouse.
From this experience, it can be considered that the operational-space inverse-dynamics is a mature tool that is able to replace the inverse kinematics approach for all robots where dynamics matters. Particularly, in humanoid robotics, inverse dynamics can be directly applied by sending the reference acceleration output, obtained with the inverse-dynamics solver, to the robot. A second conclusion is that motion capture is a very efficient tool also for robotics: it has been used for a long time in computer graphics, but it can also give the same expressive capabilities to robots, even if the retargeting is more tricky to apply. In particular, the SoT, while enforcing the dynamics of the retargeted motion, provides the robot programmer with some easy edition capabilities to correct the defects of the retargeted motion or to augment the original movement with some artificial features.

4.3 Analysis of the Organization of Human Motion

A large part of research works aiming at identifying functional principles of human movements, such as cost function minimization, co-activations or synergies, are based on the analysis of stereotyped movements involving one limb only, for instance reaching hand movements towards a target [Guigon 07]. Understanding the principles that underlie the organization of whole-body movements is a more complex task, which is more rarely considered. The dynamic motion-generation framework described in Section 3.3 can also be used to describe the organization of whole-body human motions. In fact, the same mathematical tools and inverse-dynamics resolution software, based on the stack of tasks formalism, provide a unified framework for both synthesizing complex movements in anthropomorphic systems and modeling human movements, through the definition of a hierarchy of objectives and constraints. This section presents a brief application to the analysis of the organization of human motion[5].

[5]This analysis was introduced in: O. Ramos, N. Mansard, O. Stasse, P. Souères, *An Advanced Robotics Motion Generation Framework for Inferring the Organization of Human Movements*, Computer Methods in Biomechanics and Biomedical Engineering (CMBEE), Vol.16 N.1, September 2013 [Ramos 13].

4.3.1 Methodology

This case study aims at emphasizing that the described motion generation framework, which was used to make humanoid robots execute complex tasks, provides a powerful tool for inferring the organization of human movements. Indeed, complex whole-body movements in humans rely on a set of motor objectives and constraints, which are not easy to identify. In particular, a difficult question is to determine how the motor control is distributed to guarantee, at the same time, the postural balance and the limb displacements required for the task. The proposed motion generation framework (Section 3.3) provides a standard way to synthesize such complex movements as an ordered set of objectives and constraints. Using a dynamical model of the human body, it is then possible to test some structural hypotheses about the organization of the motor control associated to a given task. By comparing the variation of key parameters given by the motion generation software with the ones measured on human subjects executing the same task, it is possible to evaluate the proposed hypotheses, and going back and forth from observation to simulation, one can then infer key elements about the dynamic structure of human movements.

4.3.2 Results

A simple application is described here: the motion generation framework is used to make the robot execute the same whole-body movement as a human. The human makes some movements with the hands and raises one foot to make a circle over the ground surface. After that, he moves his foot back to the ground. The motion to be imitated was acquired using a motion capture system, as described in Section 4.2. The captured poses were retargeted to the model of HRP-2 obtaining the joint trajectories, and the inverse-dynamics solver was applied using two tasks to make the robot move. One task was the posture trajectory for the whole body. Since there are differences in size and mass distribution between the human and the robot, following only the posture has the risk that the robot falls down when raising the foot. To ensure stability, a task to keep the center of mass inside the support polygon was added. Fig. 4.27 shows the spatial evolution of the robot and human waists as well as the robot center of mass. It can be observed that even though the robot waist was not explicitly constrained to follow the human waist when generating the motion, it implicitly inherited the human trajectory. Both curves are similar, but with different scales since both models have different dynamics. These results show that the waist trajectory is invariant after retargeting the motion to the robot. We can then deduce that the tasks used to describe the motion reflect the human motor control organization for the task.

4.3.3 Conclusions

This experiment shows how the motion generation framework can be used as a simulation tool to infer the dynamic organization of human motion. Although the robot dynamics is far simpler than the human one, including many less degrees of freedom, the presented example illustrates that synthesizing the movement as a stack of tasks is a way to preserve key features of the human motor control organization.

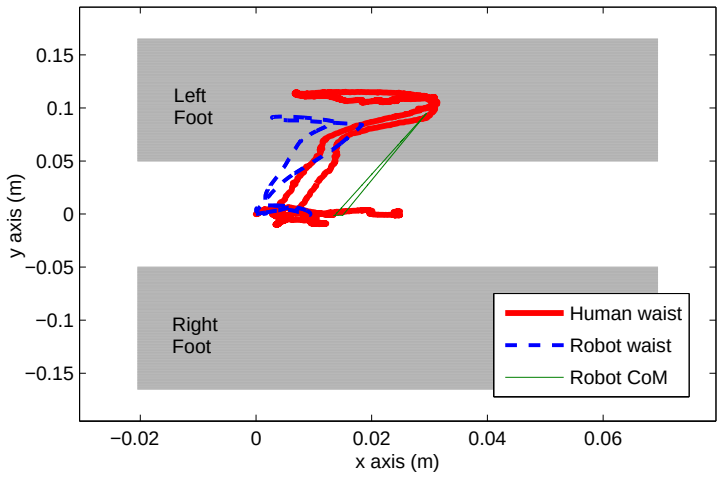


Figure 4.27 – Comparison of the waist trajectories of the human and the robot. The robot waist is not directly constrained but the obtained trajectory is similar to the one of the human.

4.4 Dynamic Walking on Non-Planar Surfaces

There exist two types of walking schemes for bipedal systems: static walking, and dynamic walking. Static walking always keeps the CoM inside the support polygon [Wieber 02] generating a slow walking pattern, whereas dynamic walking keeps the Zero Moment Point (ZMP), which is equivalent to the Center-of-Pressure (CoP) for planar surfaces [Sardain 04], inside the support polygon [Vukobratović 04] allowing the CoM to possibly leave the polygon and generating faster and more natural walking patterns. Since static walking leads to slow and unnatural motion of the biped, dynamic walking is mostly the preferred method of choice for humanoid robots.

Classical methods for dynamic walking use a Model Predictive Control (MPC) based walking pattern generator (WPG) that previews the future states of the robot to obtain the trajectory of the CoM and the positions of the footprints so that the robot is able to walk. The first schemes allowed mainly to walk on planar homogeneous surfaces [Kajita 03a], but using such schemes, several works have recently allowed real humanoid robots to walk on uneven terrain [Kaneko 04, Morisawa 11, Nishiwaki 12]. These are, typically, adaptations to a horizontally composed plane, or systems where preview control considers information from the current inclination of the upper body. Methods such as the one proposed in [Nishiwaki 12] use predictive attitude compensation control adjusting the ZMP reference to repetitive walking and updating the desired landing position based on gains of impedance. Other methods like the ones proposed in [Morisawa 11], [Kaneko 04], [Ott 11] compensate for the roughness of the terrain applying some efficient formulations of the ZMP preview window, and rejecting the error modeling using the robot attitude estimated by an accelerometer and a gyrometer. Coupling both aspects with the information provided by a portable laser, namely Hokuyo, the current best system implemented on a real humanoid robot was proposed in [Nishiwaki 12], where a 38 DoF HRP-2 is able to cope with gravels, unknown slopes of 10 degrees, as well as other irregularities. This approach is still based upon a kinematic control scheme driven to regulate global dynamical parameters such as the total angular momentum derivative about

the robot CoM. However, this method does not directly handle the dynamics of the robot, and it uses an inverse kinematics scheme where inequalities cannot be formulated. The work presented in [Morisawa 11] suffers from the same limitation.

The work of [Nishiwaki 12] also pointed out the necessity of handling noise which occurs even with a laser sensor. One problem with current laser sensor technology is time to fly and the weight. A humanoid robot needs a relatively dense reconstruction to elect potential contact points while walking, and fast enough lasers are currently too heavy to be embedded inside a humanoid robot. To reconstruct a surface, 3D information provided either by a Kinect or a stereoscopic system would need to be implemented, but it represents a difficult task both in computer vision and robotics. The problem is that depth or range sensors provide a huge amount of data that has to be processed to get high level information, which makes real time reconstruction difficult. Yet, as mentioned above, a precise dense reconstruction is critical for pattern generation in rough terrains. Precision below a few centimeters may cause strong impacts on the robot feet with the floor or, on the contrary, may cause the robot feet to never reach the floor. In either case, the robot may lose balance and fall down. One solution, which is used here, is parallel processing using GPUs that allows to get very efficient 3-D dense reconstruction systems [Newcombe 11, Rusu 11].

The approach presented in this section addresses the problem of walking on non-planar surfaces using an MPC based WPG together with the inverse-dynamics SoT framework. As the foot goes down, possible collisions are detected thanks to the reconstructed map of the terrain which was obtained by a stereo vision system. In this way, collision points are found and handled by the solver, and if necessary, some extremes of the swinging foot are properly taken towards the ground to maximize the support area without losing dynamic balance. An important assumption is that the ground is supposed to remain static. In this way, the system can autonomously adapt the swinging foot according to the terrain providing a "software-based" compliance to the walking scheme[6].

4.4.1 Task-based Foot-landing Compliance

Dynamic walking typically uses a (WPG) that assumes co-planar foot-steps and a CoM trajectory that is restricted to a plane [Kajita 03a]. Appendix B gives a review of the MPC based WPG that has been used here. Given a certain CoM reference velocity, the WPG produces feet trajectories, ZMP trajectories and a CoM trajectory which, all together, generate a balanced motion for an inverted pendulum. To generate whole-body trajectories, one approach [Kajita 03b] is to use tasks for the CoM, and for the feet to follow the given trajectories in a perfect way and in open loop (possibly using a stabilizing method [Ott 11]). If some irregularity on the horizontal surface is found, the control system is likely to fail, the robot will not be able to keep its balance and it will fall down. One of the main objectives of the control scheme in this section is to allow the foot to be compliant with respect to irregularities or roughness in the terrain. That is, if an irregularity is found, the foot should not try to push on it due to the WPG specification, but it should be able to comply, and adapt itself to this irregularity by properly rotating about some axis. This compliance will allow the robot to use the WPG to walk on rough surfaces without losing balance. More specifically the compliance

[6]This approach appears in: O. Ramos, N. Mansard, O. Stasse, P. Souères, *Walking on Non-planar Surfaces using an Inverse Dynamic Stack of Tasks*, IEEE-RAS International Conference on Humanoid Robots (Humanoids), Osaka, Japan, November 2012 [Ramos 12].

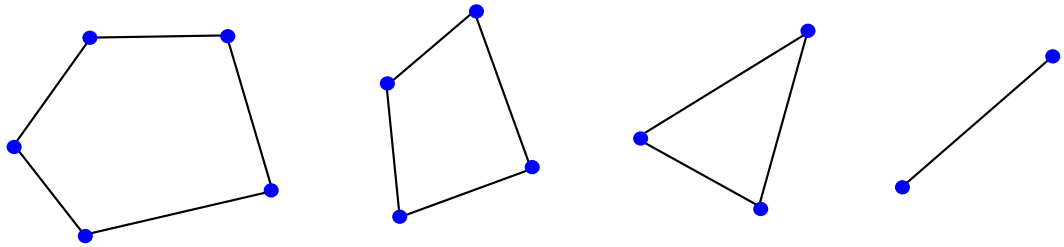


Figure 4.28 – Contact surfaces with different contact point numbers (from 5 to 2)

is at the task level and not at the mechanical level to avoid additional control for the passive material.

The approach assumes that collision detection is available and that the robot can know the exact position of the points in contact with the ground, as well as the forces at those points at every instant of time. With the information of the encoders and the Inertial Measurement Unit (IMU) this allows to reconstruct the component x_f (3.21) of the generalized coordinates. The number of points is specified in terms of the vertices of the convex hull formed by all the points belonging to the contact area, as Fig. 4.28 shows. The foot follows the steps defined by the WPG, which assumed a horizontal surface. However, when it is moving down to the ground and finds some contact point (generated by the irregular ground or by a small obstacle whose size is assumed to be smaller than the step height) it will instantaneously stop its downwards motion to comply with the irregularity that was found. The following cases might happen:

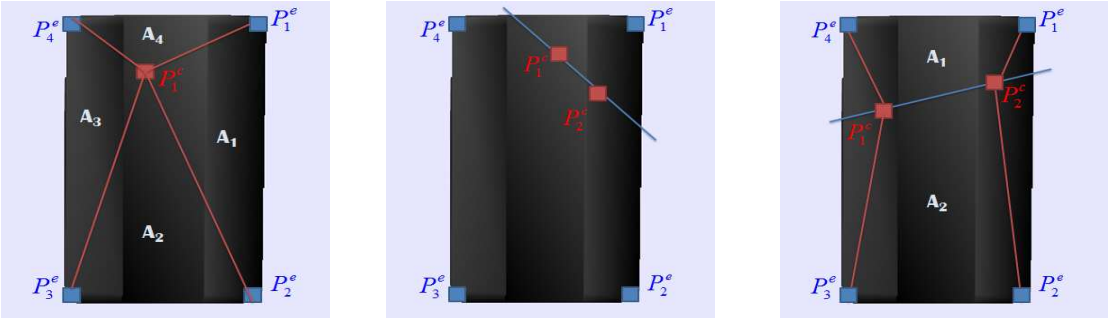
- There are three or more contact points (6D planar contact)
- There are two contact points (5D edge contact)
- There is only one contact point (3D point contact)

If there are more than three contact points, the foot is assumed to be able to safely step on those points, which will generate the support polygon, and it simply stops its motion. When there are less than 3 contact points, the foot does not continue moving down to the ground but it still needs to move to find at least one more contact to have a consistent support polygon. In this case, if the foot is left uncontrolled after the first contact(s), the dynamics of the whole-body might take the foot to an unstable position. To avoid these instabilities, the foot extremes have to be controlled so that the maximum support polygon is obtained.

For a single contact point, the foot is in the situation shown by Fig. 4.29a where the single contact point is p_1^c and the foot extremes are denoted by p_1^e, p_2^e, p_3^e and p_4^e. In this case, four triangles with areas A_i, $i = 1, 2, 3, 4$, are formed by joining the contact point to two consecutive foot extremes. The area of the triangle formed by the consecutive foot extremes p_i^e and p_{i+1}^e is given by

$$A_i = 0.5 \parallel (p_1^c - p_i^e) \times (p_1^c - p_{i+1}^e) \parallel \tag{4.7}$$

The triangle with the greatest area will contain the extremes of the foot that are farther from the contact point, and thus, it is desirable to take those extremes to the ground so that the largest support polygon is obtained. To that end, a task is assigned to each of these extremes controlling only the vertical z position so that they go to the ground. While these points are



(a) 1 contact point **(b)** 2 contact points (case 1) **(c)** 2 contact points (case 2)

Figure 4.29 – One or two contact points on the sole of the robot

moving, if a contact point is detected, it is added as a contact to the solver and its position is checked against the points that were going to the ground. The task for the closest extreme is removed. The foot continues its rotation until another contact point is detected, in which case, the remaining extreme task is removed.

If there are only two contact points, there are two possibilities shown in Fig. 4.29b and Fig. 4.29c. To obtain the case, a line L is passed through both contact points and the extremes of the foot are determined to lie in one side or on the other side of the line. Let the contact points be p_1^c and p_2^c, the vector joining these points be $v = (v_x, v_y) = p_1^c - p_2^c$, and the vector joining one point with one extreme of the foot be $v_i = (v_{ix}, v_{iy}) = p_i^e - p_2^c$. The idea is to rotate the points so that the line L is aligned with the vertical line. Then, the sign of the arc-tangent can be used to determine the side of the line in which a point lies. The angle that line L must rotate to be aligned with the vertical is $\theta = \text{atan2}(v_y, v_x)$. Then, each v_i is rotated by the angle θ as:

$$v_{ix}^f = v_{ix}\cos(-\theta) - v_{iy}\sin(-\theta) \tag{4.8}$$

$$v_{iy}^f = v_{ix}\sin(-\theta) + v_{iy}\cos(-\theta) \tag{4.9}$$

After this rotation, the angle to the line is determined as $\phi = \text{atan2}(v_{iy}^f, v_{ix}^f)$, the sign of ϕ indicating whether the point is on one side or the other of line L. If only one point is on one side and three are on the other side (Fig. 4.29b), the three points are taken to the ground. If two points are on each side (Fig. 4.29c), then, the area of the quadrilateral formed by the contact points and the extremes on each side is computed and the largest area indicates that those extremes are farther and must be taken to the ground. As in the case of a single contact, as soon as a new contact appears, it is added as a contact to the solver, and the task for the closest extreme is removed.

4.4.2 Compliant Walking Scheme

The heuristics presented in the previous section are used to determine if the foot completely stops its motion or which part of it should be moved towards the ground if there are less than 3 contact points. However, the latter case poses an additional problem for the computation of the Jacobians. The contact forces f_c can be expressed in terms of a $6D$ spatial force Φ at some frame of the foot, and it is common to have a direct measurement of the Jacobian

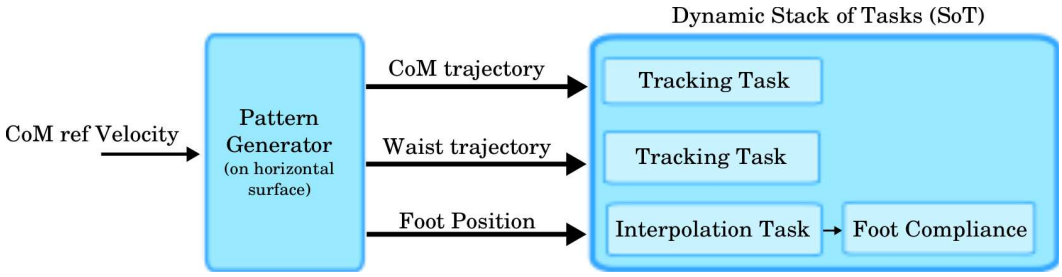


Figure 4.30 – Compliant walking scheme. The upper level specifies a direction of motion. Then the WPG finds foot-steps, foot trajectories, a CoM trajectory and a waist trajectory. The Dynamic Stack of Tasks ensures that the resulting whole-body motion is dynamically consistent with the robot model and its constraints.

relative to this frame, J_o, which is a geometric Jacobian. Then, the contact Jacobian for the contact point is computed as in (3.49); that is, $J_c = XJ_o$, where X is a matrix involving the cross product with the contact points (3.48). The problem arises from the fact that if there are only 2 or 1 contact point, X is not full-column rank and the expression for δ_c in (3.40) has to be modified to $\delta_c = -(XJ_oB)^{\#}X\dot{J}_o\dot{q}$. After this modification, the SoT described in section 3.3.4, which explicitly controls the contact force f_c, can be directly used.

The compliant walking scheme is obtained by using the output of the WPG as input to the inverse dynamics solver as shown in Fig. 4.30. The WPG outputs three elements: the trajectory for the CoM, the trajectory for the waist of the robot, and the footprints for each foot. The usage of these elements within the SoT is done with tracking and interpolation tasks. Tracking tasks are 6D operational tasks described with a PD law and aim to control the position and/or orientation of a certain characteristic of the robot (the CoM, or an operational point). The interpolation task is a task that takes an operational point from an initial to a final pose satisfying some fixed (and hard) time constraints, and it has been described in 3.4.3. The dynamic SoT that is used considers the following tasks:

- Tracking task for the horizontal (x, y) components of the CoM trajectory given by the WPG, which assumes a constant height.
- Partial tracking task of the waist trajectory. This task controls the height (position in z) of the waist at a certain constant value, as well as the horizontal orientation in x, y. As observed, not all the six degrees of freedom are controlled, and therefore the name of partial tracking task.
- Interpolation task on the swinging foot. The WPG gives the footprints on the ground. This task takes the swinging foot from its initial position to its desired final position. There are two interpolation steps: the first task takes the foot from the initial position on the ground to an intermediate position that lies halfway between the initial and final positions with a predetermined height. The second interpolation takes the foot from the intermediate to the final position, if possible. The foot reacts in a compliant way if a contact is detected before arriving to the final position on the ground, as explained in section 4.4.1.

Unlike schemes that only follow the output of the WPG using kinematic tasks, the dynamic SoT makes it possible to handle rough surfaces by adding compliance at the task level to the foot at the moment of detection of a contact with the environment.

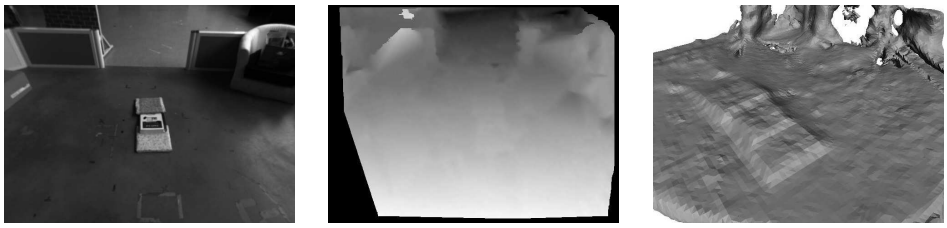


Figure 4.31 – From a pair of images of the scene in front of the robot (left) a dense disparity map is estimated (middle) and from this disparity map a dense surface integrating the previous frames into the volumetric grid is estimated (right). These images were taken from the HRP-2 stereo vision system with the cameras tilted towards the ground to allow ground reconstruction tasks. It is assumed that a mechanism is implemented to make the robot search for traversable areas [Maier 13].

4.4.3 Stereo-Reconstruction of Dense Surfaces

To perform the dense reconstruction of the floor surface in front of the robot, we rely on a real-time approach similar to the KinectFusion algorithm [Newcombe 11]. This approach, originally developed for a RGB-D sensor, models 3-D surfaces as zero-valued level sets of functions defined over the workspace volume. These functions are referred to as *Truncated Signed Distance Functions* (TSDFs) and they are incrementally built by integrating the depth measurements the sensor provides, frame after frame. TSDFs are defined in the 3D space and their value is the signed distance to the closest obstacle. Here, we extend this approach, initially proposed for RGB-D depth data, to disparity data generated from a stereo head. Although the stereo data is noisier than the one from RGB-D sensors, it is a passive sensor and can be used outdoors in sunlight conditions.

Consider, as in the previous sections, that k is a discretized time index. The idea is to update a mathematical representation of the surface through a volumetric TSDF model (defined over a 3D grid), referred to as F_k. The basic steps for integrating one new set of disparity measurements at time k, to update F_k and the corresponding surface, are the following:

(I) Filter the raw depth measurements generated from the stereo head (D_k). For that purpose, here the bilateral filtering was used.

(II) From these filtered measurements and the prediction of the estimated surface at the previous step, estimate the transformation between the measured surface and the predicted one using the iterative closest point algorithm (ICP) and update the camera pose.

(III) Compute a volumetric grid formed from "local" TSDF values F_{D_k}, to which confidence weights W_{D_k} are associated, and integrate them into the global volumetric grid $\{F_k, W_k\}$.

(IV) Predict a new surface for the next iteration by using ray-casting over the zero-crossings of the fused global volumetric grid $\{F_k, W_k\}$.

The core of this algorithm is the computation and fusion of volumetric grids (i.e., the third step mentioned above). For a 3D point p, expressed in the global frame g, its value in the

current local volumetric grid $\{F_{D_k}, W_{D_k}\}$ is computed as

$$F_{D_k}(p) = \Psi(\lambda^{-1} \|t_{g,k} - p\| - D_k(x)),$$
$$W_{D_k}(p) \propto \cos(\theta)/D_k(x),$$

with $\lambda = \left\|K^{-1}[x^\top \ \mathbf{1}]^\top\right\|$ and

$$\Psi(\eta) = \begin{cases} \min(1, \frac{\eta}{\mu}) \ \mathrm{sgn}(\eta) & \text{iff } \eta \geq -\mu \\ \text{null} & \text{otherwise} \end{cases}$$

where μ is a truncation distance (a parameter of the algorithm), and $x = \pi([K, \mathbf{1}]T_{g,k}^{-1}p) \in \mathbb{R}^2$ is the image projection of p. K is the 3×3 matrix of intrinsic parameters of the camera, π is the projection operator, $T_{g,k} = \begin{bmatrix} R_{g,k} & t_{g,k} \\ 0 & 1 \end{bmatrix}$ is the pose of the camera at time k in the global frame g, and θ is the angle between the associated pixel ray direction and the surface normal.

The global volumetric grid at time k is formed by the weighted average of all individual volumetric grids up to $k-1$. It can be shown that the optimal grid can be incrementally obtained using a simple point-wise on-line weighted average

$$\begin{aligned} F_k(p) &= \frac{W_{k-1}(p)F_{k-1}(p) + W_{D_k}(p)F_{D_k}(p)}{W_{k-1}(p) + W_{D_k}(p)}, \\ W_k(p) &= W_{k-1}(p) + W_{D_k}(p). \end{aligned}$$

To use this algorithm with stereo data and generate local data D_k, a disparity map from a pair of rectified images is estimated, from which the depth map D_k is derived assuming that the stereo rig is completely calibrated. The literature of algorithms that estimate disparity maps is huge, but since a real time one is needed for this application, the one proposed in [Geiger 10] has been used. This algorithm estimates a piece-wise disparity map using an initial sparse disparity map of high textured points as vertices that define a triangulation of the image. Then, the dense disparity map of each sub-region is estimated using the initial sparse disparity map as a prior in a probabilistic scheme. The steps of the reconstruction process are illustrated and further described in Fig. 4.31.

4.4.4 Planning on Dense Surface with Visual Reconstruction

The visual information obtained from the 3-D vision reconstruction can help the robot pose its feet on the ground as an approximation to the real ground[7]. Due to problems in light conditions (the rough terrain and the obstacles do not always behave as Lambertian surfaces in reality), noise, lack of good features for reconstruction, calibration errors among other problems related to the computer vision system, the reconstruction is not perfect. However, its usage enables the robot to foresee the ground on which it will step and to take the step in a smoother way when close to the irregularities.

Consider the scheme in Fig. 4.32 that shows the real ground in gray, the 3D visual reconstruction as an approximation to the ground-truth in green, and the foot represented by

[7]The extension that uses visual reconstruction was introduced in: O. Ramos, M. Garcia, N. Mansard, O. Stasse, J-B Hayet, P. Souères, *Towards Reactive Vision-guided Walking on Rough Terrain: An Inverse-Dynamics Based Approach*, International Journal on Humanoid Robotics (IJHR), Vol.11 N.2, July 2014 [Ramos 14a].

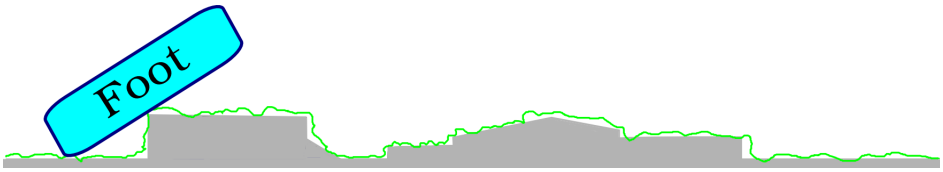


Figure 4.32 – Example of a situation handled by the approach. The gray part represents the real environment, and the green line the stereoscopic reconstruction. The foot is in its final position after the adaptation by the proposed method.

a rectangle. Without foot compliance, the foot would always try to arrive to the position of the horizontal ground, and in the case shown in the picture, it will evidently fail and make the robot fall down since it would try to continue going downwards even though there is an obstacle that stops it. Adding the foot compliance described in section 4.4.1, the foot will react to the irregularities and its position will be accordingly modified. However, since the foot initially tries to move until reaching the flat ground, in case of an irregularity, it might find the first contact with a considerable velocity which might make the impact very harsh. To avoid these strong impacts, the visual reconstruction is used.

The final position of the foot, set to the ground by the output of the WPG, is modified according to the information given by the 3D reconstruction, which is an approximation to the reality. Then, in the interpolation task of the compliant walking scheme (section 4.4.2) the final position in the ground is modified to be the one 'predicted' by the 3D reconstruction. The advantage of this modification is that the foot will arrive to the real irregularity with a lower velocity and the contact impact will be smoother. According to this prediction, the foot is moved but the real contacts from the real environment are taken into account to rotate the foot according to the heuristics for the foot compliance. The visual information only serves as a means to foresee the environment but it is the environment itself which will determine the real contact with the foot.

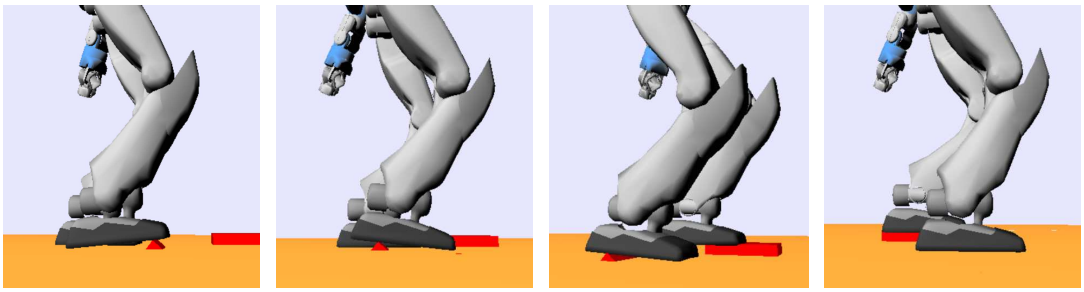
4.4.5 Results

The results will first present the application without using the visual reconstruction, relying on artificial obstacles. Then, the results using the visual reconstruction in the framework will be shown.

Walking Without Visual Reconstruction

These results are validated in a simulation of HRP-2 considering its full dynamics. The environment consists of a ground where random objects were present. The height of the obstacles was always smaller than the step height. As previously described, only the CoM and the waist are tracked by the solver. The foot positions are specified using two interpolation tasks. The robot first walks in a straight line and then slightly turns left. Some snapshots of the robot are shown in Fig. 4.33. The snapshots show the right and left foot stepping over different obstacles in different conditions. Since there is no control on the arms, they move freely to help keep the CoM position.

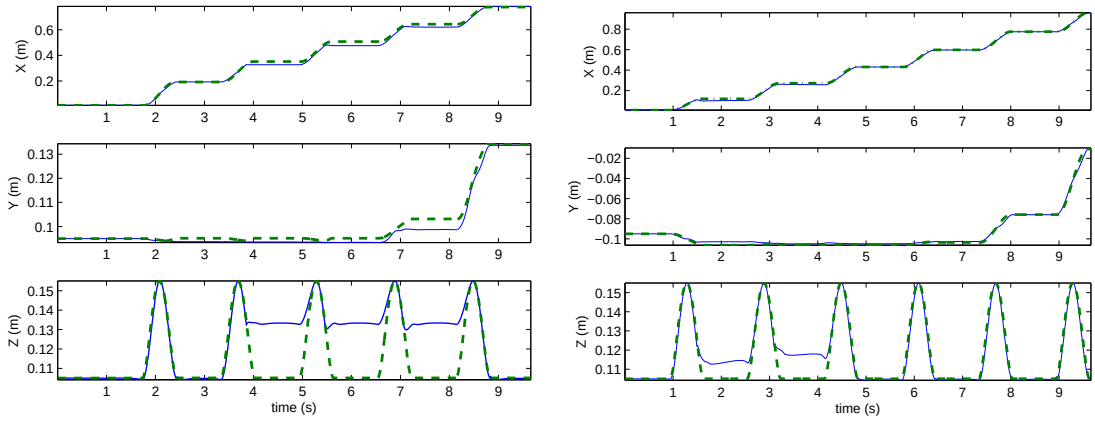
Fig. 4.34a shows the trajectory of the right foot. The dashed line shows the trajectory that is obtained from the WPG, which assumes a constant ground height. As stated before,



(a) (b) (c) (d)

Figure 4.33 – HRP-2 walking on a non-planar surface.

the right foot does not follow this trajectory but only uses the final position of the foot. In this experience, the first two steps of the right foot find an obstacle on the ground. It can be observed in the curves with solid line that even though the foot is not tracked, the interpolation task generates a trajectory similar to the one given by the WPG. The difference in the height z for the two first steps is due to the obstacle encountered by the foot, which is properly handled by the system without losing the dynamic balance. Fig. 4.34b shows the



(a) Right foot **(b)** Left foot

Figure 4.34 – Trajectory of the right and left foot. The dashed line shows the trajectory of the WPG and the solid line shows the trajectory generated by the inverse dynamics controller.

same information for the left foot. In this case, the second, third and fourth steps encounter an obstacle on the ground and that is the reason why there is an observable difference in the z axis between the output of the WPG and the trajectory generated by the dynamic solver.

Walking using the Visual Reconstruction

This part of the results shows the proposed scheme together with the visual reconstruction. A simulation using the full dynamic model of HRP-2 has been performed on a ground with some irregularities and obstacles. This ground was obtained using the cameras of the robot. Since structured light RGB-D sensors are not present on the robot, the stereo vision system located in the head has been used to reconstruct the ground. As mentioned above, the robot

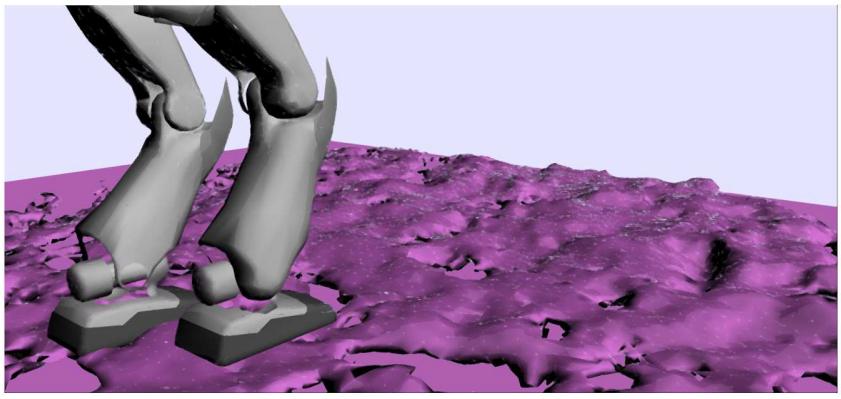


Figure 4.35 – Robot and rough ground model from visual reconstruction

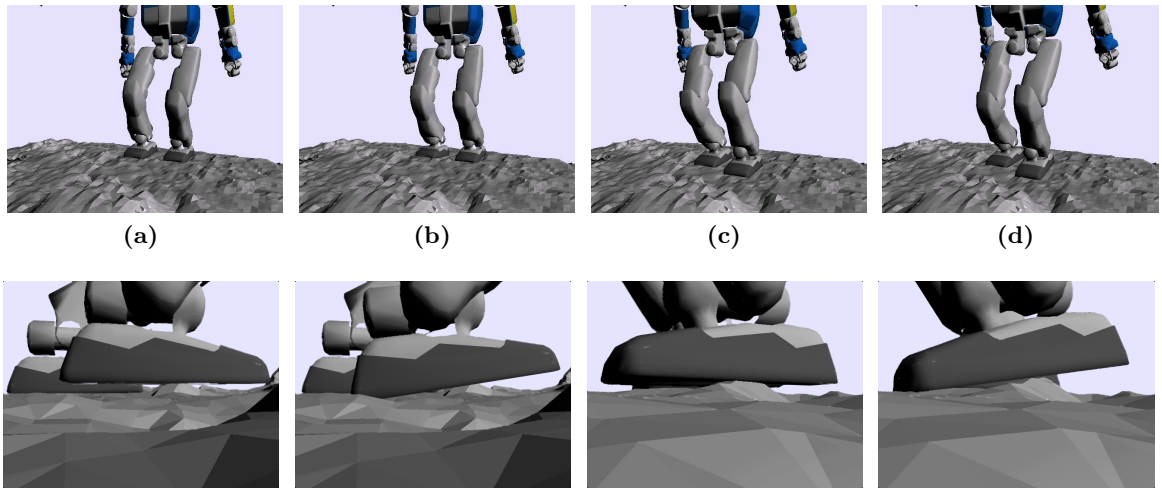


Figure 4.36 – HRP-2 walking on a rough surface. The second raw is a zoom of the foot and directly corresponds to the first raw. The view is focused on the right foot for (e), (f), and on the left foot for (g), (h).

design is such that these cameras are tilted towards the ground. Fig. 4.35 shows the robot and a rough ground model obtained by visual reconstruction.

The case where the robot walks on a rough surface is shown in Fig. 4.36, where the second row depicts a closer view of the corresponding robot foot of the first row. The robot starts from an initial position where both feet are on a flat horizontal surface. Then, the robot performs a step with the right foot. However, before arriving to the theoretical flat ground, a contact is detected and the foot stops its motion down (a),(e). Since only two contact points were detected, the right foot rotates 'backwards' as shown in part (b),(f). Then, the left foot performs a step. As with the right foot, the left foot is stopped before arriving to the flat ground since a contact is detected (c),(g). In this case, it rotates to the 'front' until it reaches the real ground and more than 2 contact points (d),(h). A similar behavior is obtained for more steps on this type of surface and the feet rotation is achieved according to the heuristics determined in section 4.4.1.

In Fig. 4.37, obstacles are added to the surface. The initial position is shown in (a). Then, the robot raises its right foot and moves it to the next theoretical footprint assuming horizontal flat ground (from the WPG). However, the foot finds an obstacle before arriving to the ground. The foot compliance is shown in (b) since the obstacle was detected, and (c) shows the foot rotating to reach a proper support polygon that will make it stable. After this, the robot continues walking. It should be noted that in this case there is only an obstacle for the right foot and the left foot steps on a theoretically flat and horizontal ground. After the step of the left foot, (d) shows again the right foot stepping on the ground. In this case, there is no need to rotate the foot since the supporting surface is enough (the foot steps are almost completely on the obstacle and, thus, a proper support polygon is obtained). Part (e) shows the left foot in the swinging phase and (f) in double support phase. It can be observed that the right foot was properly kept on the obstacle while the right foot was swinging, and the robot balance was not lost.

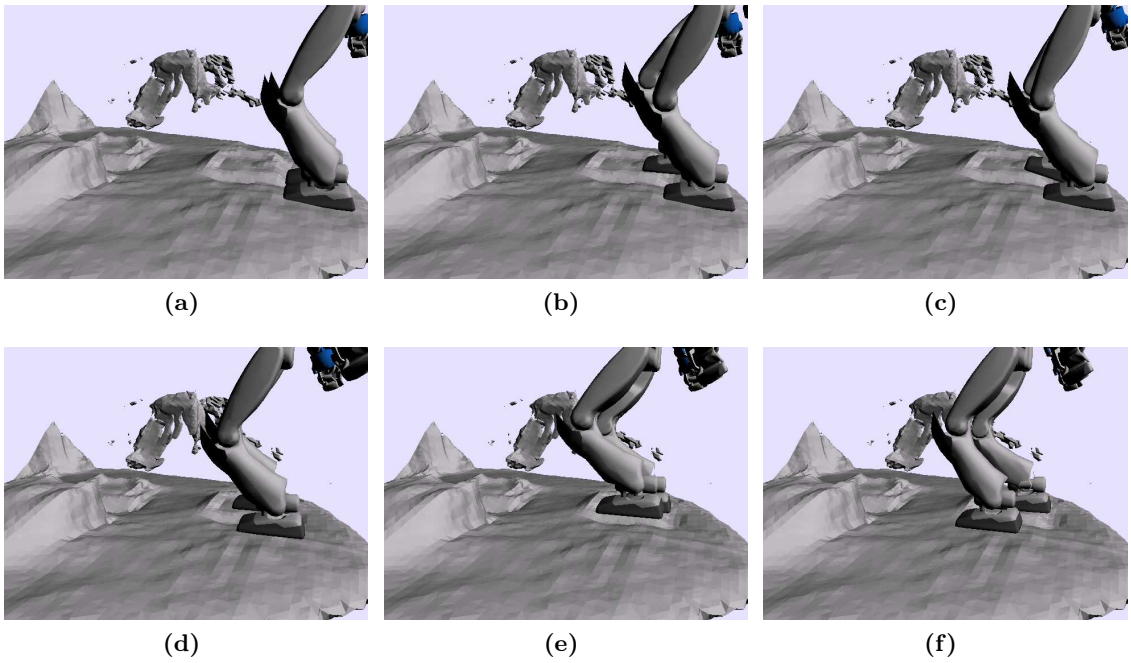


Figure 4.37 – HRP-2 walking on an obstacle.

4.4.6 Conclusions

The control scheme presented in the case study of this section makes the humanoid robot able to walk on rough terrain by detecting collision points with its stereo vision cameras and moving the foot properly to reach a larger support polygon. This behavior is equivalent to having compliance at the foot. The approach has been tested in a simulation environment. Even though surfaces might be rough, the method is limited to horizontal cases and would fail if the ground has a large slope, in which case, a modification in the WPG would be needed. However, the presented method provides a very powerful reactive navigation system able to cope with a large set of uneven grounds thanks to a very efficient inverse dynamics setup and an automatic way of finding footsteps. The user, or a high decision planner, only needs to

provide a direction of motion. Future work will apply this methodology in a real robot, and the identification of the position of contact points would be necessary through, for instance, an artificial skin at the sole of the robot.

4.5 Integration and Control of the Capture Point

This section presents the integration of the Capture Point within the inverse-dynamics SoT to allow for a better control of the CoM acceleration, acting in a certain way as a preview that can prevent dangerous CoM motion, and can automatically launch a step in case the robot is about to fall[8].

4.5.1 Introduction

The *Capture Point* (CP), also referred to as the *Extrapolated Center of Mass* [Hof 05], is the point on the ground where the robot should step on to be able to come to a complete rest [Pratt 06b], and is described in more detail in Appendix C. There exist two main applications for the CP. One is bipedal push recovery by directly stepping onto the region defined by the CP [Pratt 06a], or onto a region obtained by learning offsets to the predicted CP, which can improve the robustness of the system [Rebula 07]. The other application is the design of bipedal walking control, first proposed in [Hof 05] from a theoretical point of view, and applied in a biped robot using a pattern generator for the CP in [Englsberger 11]. The work presented in this section aims at moving the whole-body while controlling the CP at the same time. To this end, the CP task (Section 3.4.4) given by (3.68) is included in the dynamic SoT to achieve the control of the CP.

For a standing phase where both feet are on the ground and the rest of the body is moving with arbitrary motion, there are typically two ways to control the CoM. The first one is to fix it to a certain horizontal position, for instance the center of the support polygon, imposing an equality constraint. Although this approach ensures balance, it greatly affects the whole motion since it overconstrains the robot body and can make feasible tasks not achievable. The other way is to let the CoM lie anywhere inside the support polygon imposing an inequality constraint. This control is less restrictive and allows for a larger variety of motion, but it presents the potential inconvenience that if the CoM reaches the border of the polygon with any positive velocity, balance will be lost as the CoM will inevitably leave the polygon. Thus, the CoM velocity is important and if not properly considered, the border of the support polygon can become a dangerous zone that can easily lead to a loss of balance. A naive way to overcome this difficulty is by restricting the CoM inside a polygon that is strictly inside the real support polygon, giving some "security margins". However, the problem is again the imposition of unnecessary constraints to the motion, and the proper choice of those margins is not evident.

The problem with the CoM arises from the fact that inverse dynamics control considers only a linearization of the current system dynamics, and thus there is very little it can do by itself to avoid overshoots due to large CoM velocities and accelerations. The control law cannot predict future states. To overcome these limitations, optimal control can be used, but

[8]This framework appears in: O. Ramos, N. Mansard, P. Souères, *Whole-body Motion Integrating the Capture Point in the Operational Space Inverse Dynamics Control*, IEEE-RAS International Conference on Humanoid Robots (Humanoids), Madrid, Spain, November 2014 [Ramos 14b].

it is currently computationally very expensive. The approach in this section rather proposes to preview the CoM future through the CP, and therefore to constrain the CoM velocity at the limit using the CP as a measurement of its future, since both quantities are related, as (3.66) shows. While the CP remains inside the support polygon, the robot is able to freely move its whole-body with the consideration that there will always exist the possibility to come to rest. That is, the robot keeps and will keep its balance. If the CP leaves the support polygon while performing some fast motion, it will not be possible for the robot to come to rest in double support and, unless a step is taken, balance will be lost. Then, a task to impose a constraint on the CP can be a good solution.

The approach presented in this section aims at directly bounding the CoM velocity by controlling the CP, since both quantities are related, as (3.66) shows. If the CP remains inside the support polygon, the robot will be able to freely move with the consideration that there will always exist the possibility to come to rest. That is, the robot will be stable. If the CP is out of the support polygon, it will not be possible for the robot to come to rest in double support and, unless a step is taken, it will lose its balance. Thus, the CP task is proposed inside the OSID control as an inequality task to keep the CP inside the support polygon. Moreover, this task can further be used with a "larger" support polygon, in which case, a step will be required, but the CP will not move without bounds, making the step feasible. The fact of controlling the CP as a task implies that it will not escape the pre-defined polygon by implicitly constraining the whole motion of the robot. The implicit constraints might appear as motion of the parts of the robot for which no task is specified (for instance, an arm or the chest) in an attempt to compensate the otherwise fast falling motion.

4.5.2 Scheme for the Capture Point Control

The first use of the CP task introduced in Section 3.4.4 is to directly keep the CP inside the real support polygon. This is done by defining the limits of the task ($\underline{r}_p$ and $\overline{r}_p$) in terms of the real support polygon as in (3.68). The CP task is then added to the dynamic SoT, described in section 3.3.3, as an inequality task with a priority higher than the rest of the tasks, so that it is satisfied in all the cases. This leads to a more restrictive control than simply constraining the CoM within the support polygon, but it has the advantage that the CoM velocity is also controlled. This implies that the CoM at the borders of the polygon will not present high velocities that make it irreversibly exit the polygon. The reason is that by definition, if the CP is always kept inside the polygon, the CoM will be able to come to a rest within it in finite time. In other terms, the robot will be able to move without falling.

However, there are situations where the fulfillment of a task might require to move the CoM away from the support polygon (for example, if the robot has to reach an object that is farther than roughly the length of its arm), which would cause a loss of balance. For these cases, the strict balance condition can be relaxed by defining a polygon outside the limits of the real support polygon, but within an area that is reachable by a step, as Fig. 4.38 shows. Then, this 'extended' polygon is used as limit for the position of the CP. Provided that double support balance is lost, a step towards the CP will have to be taken to recover balance; otherwise the robot will fall. This scheme is summarized as follows:

- *Definition of the Capture Point task.* An 'extended' polygon covering the area that is reachable by the robot foot is defined as limit for the CP task. This will constrain

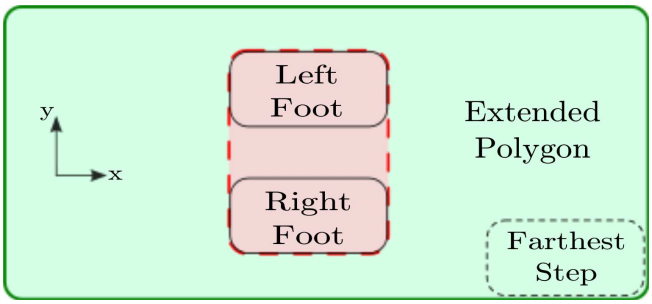


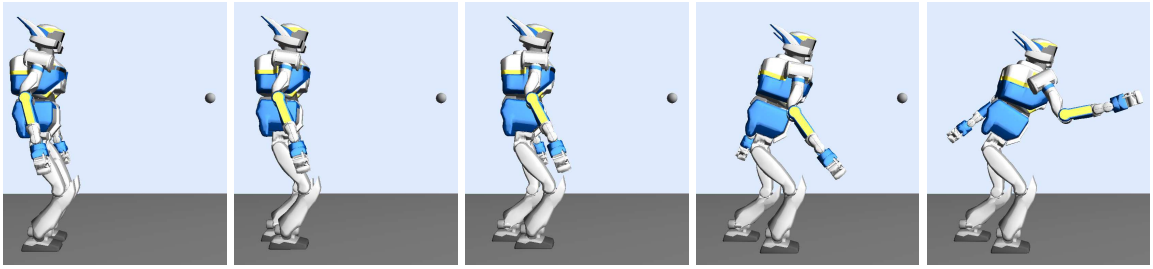
Figure 4.38 – The red polygon defines the real support polygon in the double support phase. The green polygon defines an 'extended' polygon used to limit the position of the capture point. This polygon is defined so that the farthest feasible step lies within it

the CP to always lie within the polygon and will avoid the problem of its exponential increase, which would prevent the foot from stepping over it. Then, other tasks are added to the inverse dynamics SoT to generate whole-body motion.

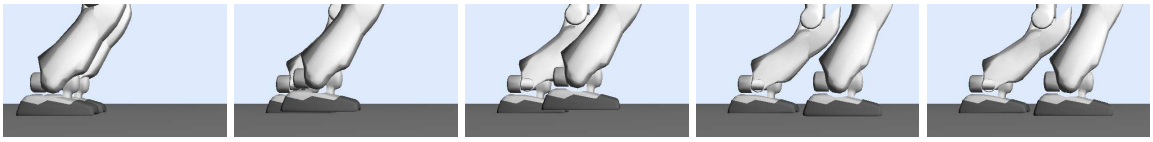
- *Loss of Balance.* If at some point of the robot motion the CP exits the real support polygon, the robot will not be able to recover its balance if it keeps its current support polygon. It will inevitably fall unless the real support polygon is extended to cover the CP position. To this end, a step must be executed.

- *Beginning the step.* To perform a step, the foot that is closer to the current CP position is selected. As soon as the CP leaves the real support polygon, the foot leaves the ground towards an intermediate position. This position is horizontally located at the current position of the CP, and the step height is pre-defined (assuming a flat horizontal ground). The horizontal position is chosen to be the current position of the CP since the final position of the CP is unknown and the motion needs to be fast (defining a position in the midway between the current foot position and the CP position reduces the capability of reaching the CP at the final stage). The desired intermediate position will change continuously as the CP moves farther. Because of this change, only the step height is considered as criteria to finish this stage. As soon as the desired step height is achieved, the foot will move towards its final position.

- *Ending the step.* After the foot completed the intermediate position, its final position is defined as the current CP position. Since the CP is still moving, the task objective will also be time-varying. However, the CP task defined at the beginning of the scheme is controlling the CP to remain inside a greater polygon preventing it from moving to an unreachable position. If no control was performed on the CP, it would exponentially move away and the foot would never be able to meet it, causing the robot to fall down in the attempt to step over the CP; thereof the importance of the CP task. When the foot reaches the CP on the ground, the robot again enters a double support phase which will now contain the current CP. Then, the CoM is ensured to be able to come to a full stop.



(a) View of the whole robot



(b) Lateral view of the feet

Figure 4.39 – Snapshots of the robot trying to reach an object (ball) which is far from the arm reachable space unless a step is performed. The robot starts in double support, then as the right arm moves towards the ball the right foot automatically starts a step to follow the CP. After the step finishes, the robot continues its motion to reach the target with the right hand.

4.5.3 Results

The described framework was implemented and tested using the dynamic model of HRP-2. The desired objective was for the right hand to reach the position specified by the ball in front of the robot (Fig. 4.39). It can be foreseen that if the robot tries to reach that position with the right hand, there are two possibilities for the CoM: (i) if the CoM is controlled to lie inside the support polygon, the robot will not achieve the goal keeping double support since the target is far away from the reachable limits; (ii) if the CoM is not controlled, the robot will fall down while trying to reach that point, unless it performs a step. The latter case without an additional step is shown in Fig. 4.40, where the right hand task drives the CoM far away from the limits of the support polygon making the robot fall. This scenario describes a typical application of the proposed approach. To overcome the problem, an extended support polygon was defined as in Fig. 4.38 covering the space that is reachable by the foot. Then, the CP task was added to the SoT having this polygon as limit.

Following the control scheme of section 4.5.2 it is observed in Fig. 4.39 that the robot starts moving the arm towards the ball. However, since the target is far, this implies an initial relatively fast motion of the arm, which generates a 'dangerous' velocity for the CoM. This is detected by the CP leaving the real support polygon as Fig. 4.41 shows (at $t = 0.27$s), which acts as a preview control for the CoM. It is at this moment that the right foot leaves the ground and moves towards the CP, which is constrained not to exponentially increase but to lie within some bounded region, as verified by Fig. 4.41 (at $t = 1$s the CP is bounded). When the foot finally reaches the CP, the robot enters a new double support phase, which now contains the CoM. After the foot reaches the ground (at $t = 1.10$s), the CoM still presents a forward motion, and therefore continues moving forwards, but eventually it is able to come to rest since the CP is now contained inside the new support polygon. At the end of the motion, the right hand reaches the far target (the ball, in this case), and both the CoM and the CP converge to the same position provided that the CoM velocity is null. Fig. 4.41 also shows

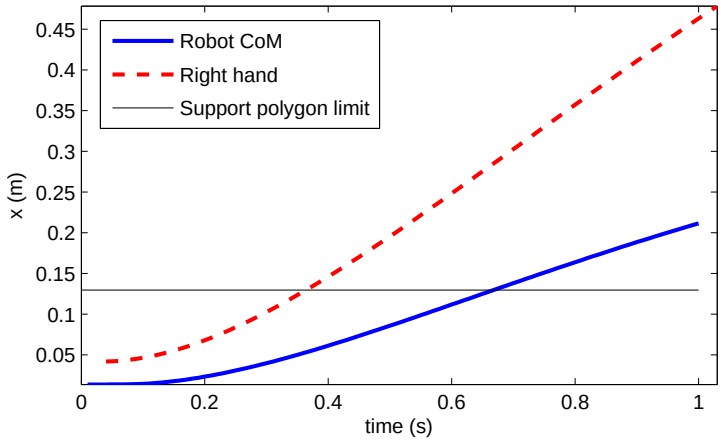


Figure 4.40 – Temporal evolution in the forward direction ('x') when there is no control of the CoM or the CP

that it is the usage of the CP that acts as a predictor for the CoM motion and allows the foot to start the step as a reaction to the velocity of the CoM, before it has escaped the support polygon and before a recovery is too late.

It should be noted that the right hand continues its motion towards the ball at all times. By the end of the motion, Fig. 4.39 shows a natural movement of the left hand backwards to compensate for the motion of the right hand, even though there is no specific task controlling the left arm. This is a consequence of the inverse dynamics SoT control and resembles the way humans move (the effect of the SoT resembling human motion was previously noted in Section 4.3). It is also important to point out that the step parameters (when to start the step and where to step to) have not been previously precomputed but have resulted as a natural consequence of the control framework in response to the loss of balance.

4.5.4 Conclusions

The proposed method proposed has shown the feasibility of incorporating the capture point inside the OSID SoT in order to perform tasks that would otherwise make the robot fall down. Stepping on the CP is a good solution but presents the inconvenience that the CP can exponentially escape the reachable zone for the foot. The control of the CP guarantees that it remains inside the specified extended polygon by implicitly constraining the whole body motion within limits that will allow the robot to attain balance after a step. One of the advantages of integrating the CP in the control scheme is the ability to determine the moment when the robot will lose its balance, and thus, when a step has to be started and where to step on. Moreover, besides balance control, the whole-body is moved with an arbitrary task generating some useful motion. The limit of the work is that currently only one step can be performed. The extension to more steps follows the same approach; however, in this case making the robot walk towards the goal might be more efficient.

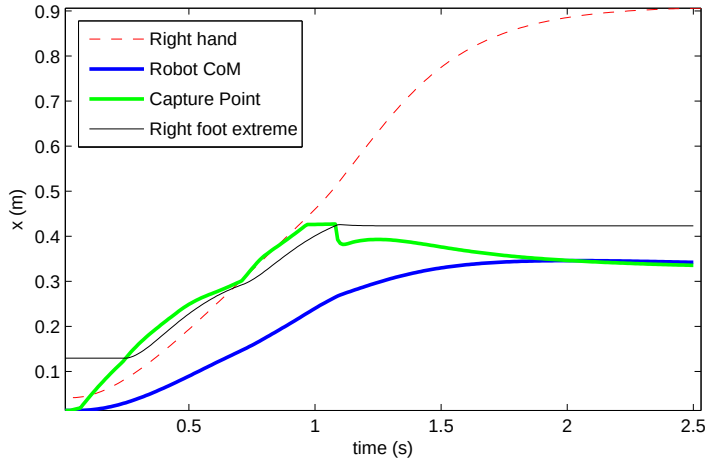


Figure 4.41 – Temporal evolution in the forward direction ('x') when the proposed control approach is used. The black line shows the frontal extreme of the right foot which determines the frontal limit of the support polygon when the foot is on the ground (as at the beginning and end of the motion)

4.6 Motion for the CHIMP Robot

This section describes some work that was done with the *Tartan Rescue* team for the DARPA Robotics Challenge (DRC) Trials, which took place on the 20$^{\text{th}}$ and 21$^{\text{st}}$ December 2013 in Miami, Florida, USA. The main objective of the DRC is the development of robot technology that can assist humans in case of disasters. Following this objective, the trials were oriented towards the achievement of several tasks that can be found in natural and man-made disasters, through the cooperation between humans and robots rather than robot autonomy. Thus, the participating robots were semi-autonomous: human operators remotely controlled the most critical decisions based on the environmental information acquired with the robot sensors. The Tartan Rescue team developed the robot called CHIMP at the National Robotics Engineering Center (NREC), which is part of the Robotics Institute at Carnegie Mellon University (CMU) in Pittsburgh, USA. The work described here was achieved during a short stay at NREC from August to December 2013. Due to the pragmatic nature of the trials, the tight schedule, and the fact that the robot was built in a very short period of time, the work of the team was in some sense more experimentally oriented with the main purpose of scoring as many points as possible profiting the robot design. At the end, the team placed 3$^{\text{rd}}$ out of 16 and continued to the DRC Finals, which will take place in June 2015.

4.6.1 Overview of CHIMP

CHIMP, which stands for *CMU Highly Intelligent Mobile Platform*[9], is a robot developed at Carnegie Mellon University in 2013 as a platform for executing complex tasks in dangerous

[9] T. Stentz, H. Herman, A. Kelly, E. Meyhofer, G.C. Haynes, D. Stager, B. Zajac, J.A. Bagnell, J. Brindza, C. Dellin, M. George, J. Gonzalez-Mora, S. Hyde, M. Jones, M. Laverne, M. Likhachev, L. Lister, M. Powers, O. Ramos, J. Ray, D. Rice, J. Scheifflee, R. Sidki, S. Srinivasa, K. Strabala, J-P Tardif, J-S Valois, J.M. Vande-Weghe, M. Wagner, C. Wellington: *CHIMP, the CMU Highly Intelligent Mobile Platform*, Journal of Field Robotics, 2015 (in press) [Stentz 15].

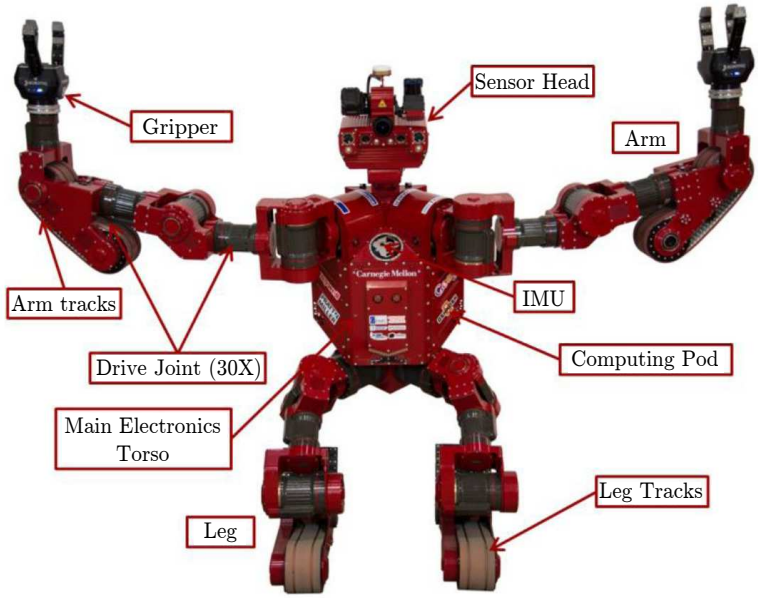


Figure 4.42 – Overview of the system that composes CHIMP [Stentz 15]

and degraded human engineered environments [Stentz 15]. The robot was designed with the peculiarity of having a track at the end of each limb in order to avoid the need for complex control by almost naturally maintaining static balance. An overview of the system composing the robot is shown in Fig. 4.42. The robot has in total 39 DoF: 7 DoF for each arm, 6 DoF for each leg, 1 DoF to control each of the 4 tracks, 4 DoF for each hand, and 1 DoF for the spinning LIDAR units at its head, which are located with an offset of 90 degrees with respect to each other. The latter DoF is always in motion to scan the environment and is typically not directly controlled. If any hazard is identified, the robot can be disabled using the system called the Mobility-Stop (M-Stop) controller, which prevents the joint motors from actuating by de-energizing them. Due to the motor brakes, the robot keeps its posture after an M-Stop.

CHIMP has various sensors embedded in its head to generate 3D representations of its environment. These sensors are: 2 LIDAR scanners that can capture 360 degrees of geometric data surrounding the robot, 2 cameras with panomorphic fisheye lenses that provide video texture data for the geometric data, a pair of stereo HDR cameras with a wide field of view used for visual odometry and obstacle detection, and a second stereo pair with a narrow field of view to obtain detailed range information for manipulation tasks. The robot also has an inertial measurement unit (IMU) system to estimate its pose. The 3D reconstruction obtained with the sensors is currently used to help human operators understand the robot environment and take decisions rather than to autonomously process the information. While the latter choice would arguably be more efficient, it implies more development and systems integration.

CHIMP possesses 3 CPUs and several FPGAs that allow it to be semi-autonomous: the operator sends high level commands and the robot executes and monitors them locally while still processing different sensor data. At the software level, the robot has an Operator Control Unit (OCU) designed to visualize the robot within its environment, to simulate its behavior and to send control commands in task and joint space. In fact, the OCU encapsulates all the

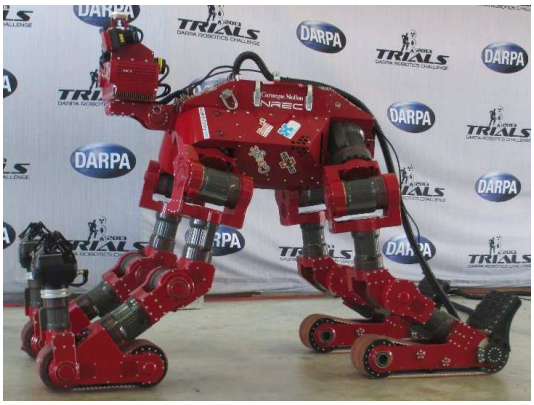


Figure 4.43 – Four-limb posture for CHIMP

methods used for motion generation, manipulation and data fusion, and it only shows the necessary information to control the robot in a "friendly" way. More details about the whole hardware and software system can be found in [Stentz 15]. It is interesting to point out that the robot was designed, built, and tested in 15 months, and it was completely assembled only 6 weeks before the trials, leaving little time for real experimentations.

4.6.2 Posture Change from Four to Two Limbs

CHIMP was designed to be a statically balanced robot and two types of postures were previewed by the design: a configuration with all four limbs on the ground, called a four-limb posture (Fig. 4.43), and a configuration with only the two legs on the ground, called a two-limb posture (Fig. 4.52). The four-limb configuration is used to drive over rough terrain using an Ackermann steering geometry, and it provides a very stable posture allowing the robot to run over obstacles controlling the tracks traction but without major modifications to the overall configuration. The two-limb configuration, although less stable, imposes no constraints on the arms, and is used for manipulation operations as well as for two-limb locomotion. The change between both postures, starting with a four-limb posture, is done in two steps: (i) the motion of the frontal tracks backwards without leaving the ground, and (ii) the motion of the robot body upwards to a two-limb posture. Before explaining the details of these two parts, the degrees of mobility of the robot in a four-limb posture will be analyzed.

Analysis of CHIMP in a Four-limb Posture

Consider a mechanism composed of N_l links and N_j joints, where joint i has f_i DoF, in a space with λ DoF. The degree of mobility M (also called degree of freedom) of this mechanism is given by:

$$M = \lambda(N_l - 1 - N_j) + \sum_{i=1}^{N_j} f_i \tag{4.10}$$

which is usually known as the *Chebychev-Grübler-Kutzbach's equation* [Gogu 05], or sometimes simply as the *Grübler's equation* [Hunt 67]. This equation will be applied to determine the degree of mobility of the robot torso when it is in a four-limb posture as shown in Fig 4.43.

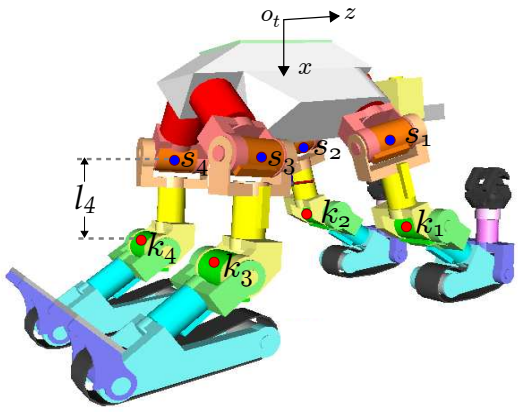


Figure 4.44 – Scheme showing the shoulders, knees and the frame used in the algorithm to move the frontal tracks backwards

Although each arm of CHIMP possesses 7 DoF (without considering the hand) and each leg 6 DoF, the kinematic chains from the torso to the tracks contain only 5 DoF for both cases. The 2 lost DoF in each arm correspond to the joints that change the orientation of the hand (hand roll and pitch in Fig 4.43), and the lost DoF in each leg corresponds to the paddles. In this configuration, the number of joints that can move the robot is 5 per limb and then $N_j = 20$. Since all joints are rotational, $f_i = 1$ for every joint i, where $i = 1, ..., N_j$. Kinematically, each limb contains 4 links from the torso to the track; additionally, the torso is a link that attaches the shoulders and the hips, and the ground is a link attaching the 4 tracks. Thus, the total number of links becomes $N_l = 18$. With these values, and using $\lambda = 6$ for the 3D space, Grübler's equation for CHIMP gives

$$M = 6(18 - 1 - 20) + \sum_{i=1}^{20} 1 = 2$$

which indicates that the robot body has $M = 2$ degrees of mobility when it is in a four-limb posture. The interpretation is the following: when the four tracks are fixed on the ground, the torso of the robot can only move in a constrained 2D manifold.

Part 1: Moving the Frontal Tracks Backwards

The motion generation method described here is a heuristic approach based on the geometry of CHIMP and consists in fixing the tracks and determining the pose for the torso that allows for the configuration to be achievable. Let the knees and the elbows be represented by k_i, and the shoulders and hips by s_i, where $i = \{1, 2, 3, 4\}$, as Fig. 4.44 shows. The length of each thigh and arm will similarly be called l_i. For a given position of the tracks, the positions of k_i are fixed since from each track to each knee/elbow k_i there are only 2 DoF and both intersect at the same point. Similarly, the positions of the shoulders and elbows s_i are fixed with respect to the robot torso.

Let the initial reference frame for the torso be located somewhere in the middle of it, oriented with the x axis pointing from the back to the front of the torso, the z axis from the hips to the shoulders, and the y axis set to complete a right-handed frame, as Fig. 4.44 shows. Let the pose of the torso with respect to this frame be $X_t = (x_t, y_t, z_t, \theta_t^x, \theta_t^y, \theta_t^z)$, where $(\theta_t^x, \theta_t^y, \theta_t^z)$

(a) CHIMP moving the frontal tracks backwards

(b) CHIMP rotating the frontal tracks

Figure 4.45 – CHIMP moving its frontal tracks (beginning of posture change from four limbs to two limbs)

is a parameterization of $SO(3)$. Given some desired k_i (obtained from the desired tracks poses), and a desired position for the torso in the horizontal plane (y_t and z_t), the problem consists in finding the good pose for the torso $X_t^{red} = (x_t, \theta_t^x, \theta_t^y, \theta_t^z)$ that does not violate physical constraints imposed by the links (thighs and arms). In other terms, the torso has to be located in such a way that it is possible for s_i to be jointed to the respective k_i through the links l_i. For a given torso pose X_t, the distance between $s_i(X_t)$ and the respective k_i is $d_i(X_t) = \|s_i(X_t) - k_i\|_2$. Letting $S(X_t) = (s_1(X_t), s_2(X_t), s_3(X_t), s_4(X_t))$, $K = (k_1, k_2, k_3, k_4)$, and $L = (l_1, l_2, l_3, l_4)$, the measure of the feasibility $C(X_t) = (c_1(X_t), c_2(X_t), c_3(X_t), c_4(X_t))$ will be

$$C(X_t) = (\|S(X_t) - K\|_2 - L)^2. \tag{4.11}$$

When $C(X_t)$ is null, a feasible torso configuration has been achieved. Although different optimization methods can be applied, a straightforward gradient descent was used to minimize $C(X_t)$, which in practice gave good results. Let $\alpha > 0 \in \mathbb{R}$ be the step size, and let $X_{t_k}^{red}$ be X_t^{red} at iteration k. The update rule using gradient descent is given by

$$X_{t_{k+1}}^{red} = X_{t_k}^{red} - \alpha J_t(X_{t_k}) C(X_{t_k}) \tag{4.12}$$

where the Jacobian is

$$J_t(X_{t_k}) = \begin{bmatrix} \frac{\partial x_t}{\partial c_1} & \frac{\partial x_t}{\partial c_2} & \frac{\partial x_t}{\partial c_3} & \frac{\partial x_t}{\partial c_4} \\ \frac{\partial \theta_t^x}{\partial c_1} & \frac{\partial \theta_t^x}{\partial c_2} & \frac{\partial \theta_t^x}{\partial c_3} & \frac{\partial \theta_t^x}{\partial c_4} \\ \frac{\partial \theta_t^y}{\partial c_1} & \frac{\partial \theta_t^y}{\partial c_2} & \frac{\partial \theta_t^y}{\partial c_3} & \frac{\partial \theta_t^y}{\partial c_4} \\ \frac{\partial \theta_t^z}{\partial c_1} & \frac{\partial \theta_t^z}{\partial c_2} & \frac{\partial \theta_t^z}{\partial c_3} & \frac{\partial \theta_t^z}{\partial c_4} \end{bmatrix}. \tag{4.13}$$

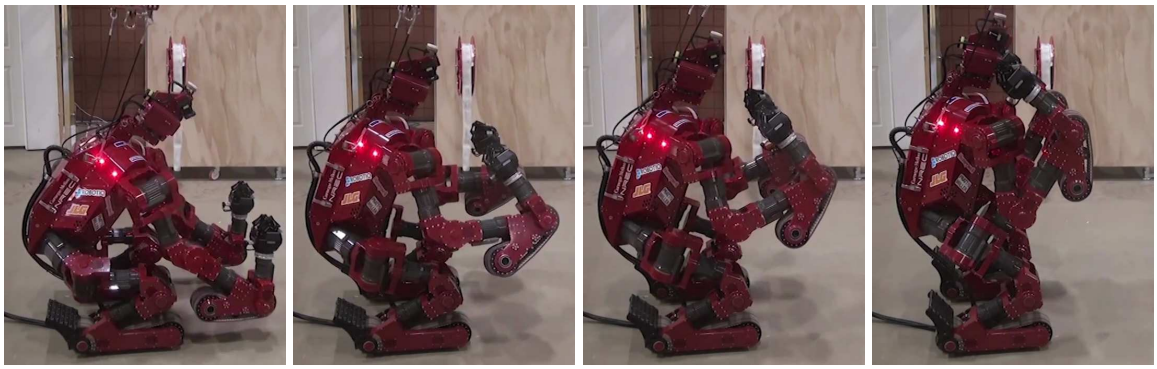


Figure 4.46 – CHIMP moving to its final 2 limb posture (second part of posture change from four limbs to two limbs)

After determining a suitable torso pose X_t, inverse kinematics is used to get the joint configurations for each limb. Using this method, the results shown in Fig. 4.45 were obtained. In Fig. 4.45a, the pose for the frontal tracks is continuously changed backwards and the pose of the torso is accordingly found. Once the projection of the CoM on the ground lies inside the polygon formed by the leg tracks, the frontal tracks can leave the ground to a two-limb posture. However, to facilitate the transition and to move the CoM to a safer position, the frontal tracks are further rotated following the same method, as Fig. 4.45b shows.

Part 2: Moving the Frontal Tracks Upwards

After the CoM projection lies inside the polygon formed by the leg tracks, the frontal tracks can leave the ground without affecting the robot balance. Once the frontal tracks leave the ground, the torso gains DoF and its pose is no longer constrained to a 2D manifold. To achieve a final pre-defined posture, the motion was decoupled in two parts: the first concerns the legs and the second the arms. The three joints at hips of CHIMP intersect at a single point and present a particular configuration similar to a Z-Y-Z mechanism. Consequently, to achieve a rotation of the hips about an arbitrary axis, the transformation between the axis-angle representation and the Z-Y-Z Euler angles can be directly used, and the rotation angles in Z, Y and Z correspond directly to each joint at the hip. Profiting this fact, the increment $\Delta\theta_h$ about the world y axis (oriented laterally) was successively given to the hips and the same increment but in an opposite direction $\Delta\theta_k = -\Delta\theta_h$ was given to the knee joints. For the arms, a cubic interpolation that accepts intermediate waypoints was used. The results are shown in Fig. 4.46 where the robot starts where Fig. 4.45 stopped and ends up in a two-limb posture which, in this case, is used for two-limb locomotion.

4.6.3 Speed and Acceleration Limits for Locomotion

Since one of the design criteria for the robot was to be statically balanced, special care had to be taken when making the robot move using the tracks. A very fast initial acceleration or a sudden stop when the robot has a high velocity can make it lose balance unless some control is applied to compensate for the destabilization. However, when the M-stop (Section 4.6.1) is applied, all the motors are de-energized and there is no possibility to apply a compensating

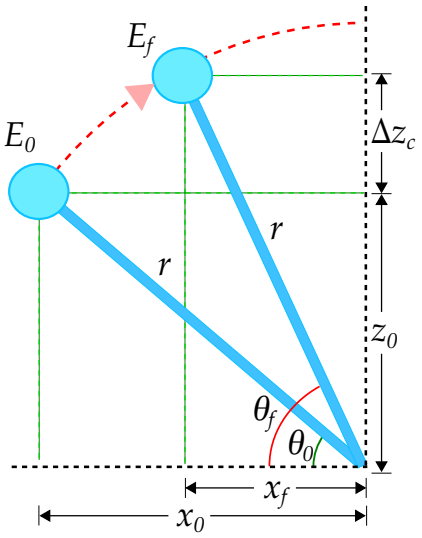


Figure 4.47 – Lateral representation of the punctual model of the robot. The robot is assumed to have stopped after having a motion to the right. The circle represents the punctual mass and the lower horizontal line represents the ground.

control. This section presents the criteria used to limit the speed and the acceleration when the M-stop is applied so that balance is not lost.

Linear Speed

The track actuators of CHIMP can provide high torques and, although the robot linear speed using the tracks can be very fast, it must be bounded in case it suddenly stops; that is, an M-stop at the robot maximum linear speed v_{max} must not make the robot tip over. Since the robot always keeps its posture while it moves using its tracks, the whole-body can be modeled as a punctual mass m located at the CoM. The analysis will be based on the transformation from kinetic to potential energy when the robot stops.

Treating CHIMP as a punctual mass, let the robot energy be $E_0 = \frac{1}{2}mv^2$ while it is moving with linear speed v. When the robot suddenly stops (M-stop), its velocity decreases but the body still tends to move due to its inertia. The energy the robot had while in motion changes to potential energy up to a moment where all the energy is $E_f = mg\Delta z_c$, where Δz_c is the variation in height from the initial to the final CoM position, as Fig 4.47 shows. Assuming the idealization that all kinetic energy has been converted to potential energy, $E_f = E_0$ and the variation in height for the CoM can be expressed in terms of the initial linear speed as

$$\Delta z_c = \frac{v^2}{2g}. \tag{4.14}$$

Let the initial position of the robot (when it suddenly stops) be (x_0, z_0) and its final position (when it lost all its kinetic energy) be (x_f, z_f), where $z_f = z_0 + \Delta z_c$. Since after the sudden stop the robot keeps its posture, if it is moving forwards, it will tend to tip over forwards about the frontal part of the tracks. Then, the CoM will follow the geometry of a circle shown in Fig 4.47, and its distance r from the frontal part of the tracks will be constant and given by $r^2 = x_0^2 + z_0^2 = x_f^2 + z_f^2$. Let the initial angle formed by the horizontal and the line from

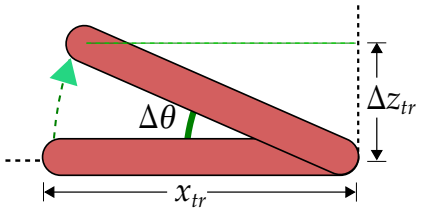


Figure 4.48 – Lateral representation of the tracks motion when robot suddenly stops.

the frontal part of the track to the CoM be θ_0, and the angle at the final CoM position be θ_f. The tangent of the latter angle is given by

$$\tan(\theta_f) = \frac{\Delta z_c + z_0}{\sqrt{r^2 - z_f^2}} \tag{4.15}$$

where the equivalence $x_f = \sqrt{r^2 - z_f^2}$ has been used. Replacing (4.14) and the equivalence of r in terms of the CoM initial position in (4.15), and after some straightforward algebra, the final angle is

$$\theta_f = \arctan\left(\frac{v^2 + 2gz_0}{\sqrt{4g^2x_0^2 - 4gz_0v^2 - v^4}}\right) \tag{4.16}$$

which, assuming a known and predefined initial CoM position (x_0, z_0), is a function of only the initial linear speed of the robot v.

The robot will not tip over unless $\theta_f > 90°$; then, the critical angle is $\theta_f = 90°$. As $\theta_f \to 90°$, $\tan(\theta_f) \to \infty$ and the denominator of the right-hand side of (4.16) must tend to zero. Thus, the critical linear speed v_{crit} is found when $4g^2x_0^2 - 4gz_0v_{crit}^2 - v_{crit}^4 = 0$ which leads to

$$v_{crit} = \sqrt{2g\left(\sqrt{x_0^2 + z_0^2} - z_0\right)}. \tag{4.17}$$

The critical linear speed v_{crit} in (4.17) is the maximum linear speed the robot can have to avoid tipping over when it suddenly comes to a stop. A safe velocity will be $v < v_{crit}$.

When the whole-body rotates about the frontal part of the tracks, the tracks themselves also rotate about that line. Let the tracks have a length x_{tr} as Fig 4.48 shows, and let the variation in height of the rear part of the tracks be Δz_{tr}. Since the tracks rotate with the rest of the body, the variation of the angle is $\Delta\theta = \theta_f - \theta_0$, or equivalently

$$\Delta\theta = \theta_f - \arctan\left(\frac{z_0}{x_0}\right) \tag{4.18}$$

where θ_f is given by (4.16). Then the geometry of Fig. 4.48 easily shows that the variation in height of the rear part of the tracks will be

$$\Delta z_{tr} = x_{tr}\sin(\Delta\theta). \tag{4.19}$$

This analysis was applied to evaluate different two-limb postures for CHIMP. For a typical two-limb posture used for two-limb locomotion, the critical linear velocity is $v_{crit} \approx 0.75$ m/s, but a safe velocity was heuristically determined to be $v_{max} = \frac{1}{3}v_{crit}$ and, with these values, the elevation of the rear part of the tracks was less than 1 cm, which is almost unnoticeable for practical purposes.

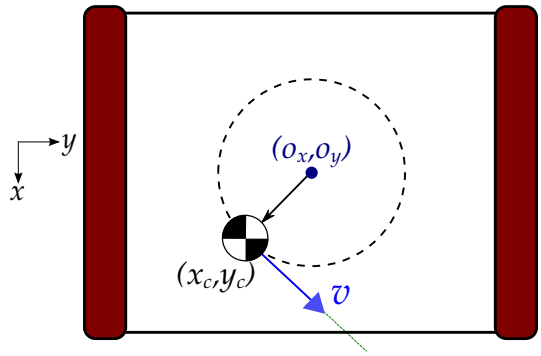


Figure 4.49 – Top representation of the robot as a punctual mass when it rotates. The rectangles at the sides represent the leg tracks

Angular Speed

When the robot is rotating using the motion of its tracks and comes to a sudden stop (M-Stop), it can eventually tip over due to its angular velocity. In other terms, the robot CoM will tend to move in the direction tangent to the (circular) trajectory it was describing before suddenly stopping. Then, it will tend to tip over as in the case of the linear speed. The robot motion due to its tracks is, for simplicity, modeled as a differential-drive, which in practice has given good results. Let the horizontal center of the robot, based on the tracks positions, be $r_o = (o_x, o_y)$ and let the CoM projection on the ground be $r_c = (x_c, y_c)$, as Fig. 4.49 depicts from a top view representation. The tangential speed at any given time will be v and it is related to the angular speed ω as $v = r\omega$, where $r = \|r_c - r_o\|_2$.

When the robot suddenly stops its angular motion, it will tend to tip over following the tangential velocity it had when it stopped, due to its inertia. First, the edge about which it would tend to tip over is found as the edge that intersects the velocity v. After the edge is identified, the previous analysis for the linear speed can be applied to bound the module of v and the relation between linear and angular speeds allows to also bound the angular speed. For typical two-limb drive postures for CHIMP, the CoM lies almost at the center of the support polygon, and the results showed that very big angular accelerations can be applied without making the robot tip over. Then, the angular acceleration was not bounded.

Initial Acceleration

When the robot starts to move forwards from rest, if its initial acceleration is too high, it can eventually tip backwards about the rear part of the tracks. Then, the initial acceleration in the forward direction $a = \ddot{x}$ must be carefully bounded since the track actuators can give the robot very high initial accelerations that can risk its balance. Let the robot CoM position be (x_0, z_0) as Fig. 4.50 shows. With these values, the angle between the horizontal plane (the ground) and the robot initial CoM is $\theta_0 = \arctan\left(\frac{z_0}{x_0}\right)$. When the robot applies a forward acceleration, there are two forces acting on the CoM: its weight mg, and the inertial reaction to the acceleration $m\ddot{x}$. These forces can be decomposed into the radial and tangential components. The radial components do not exert an important effect in this analysis since they are directed towards the origin. The tangential components can create a force able to

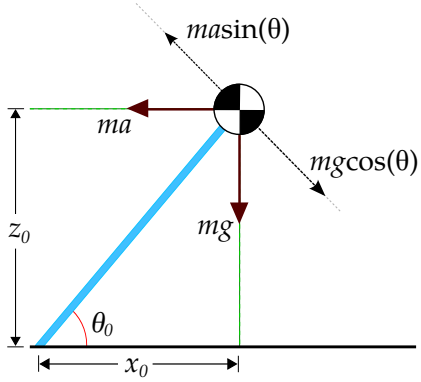


Figure 4.50 – Lateral representation of the robot when it starts accelerating.

push the robot backwards. The resultant force in the tangential direction is:

$$F_{tan} = m\ddot{x}\sin(\theta_0) - mg\cos(\theta_0). \tag{4.20}$$

For the robot to tip backwards, $F_{tan} > 0$, and if $F_{tan} < 0$ the robot can be safe. Then, using (4.20) the critical acceleration where the robot would start tipping over is given by the condition $\ddot{x}\sin(\theta_0) = g\cos(\theta_0)$. This implies that the maximum initial acceleration $\ddot{x}_{max}$ is

$$\ddot{x}_{max} = \frac{g}{\tan(\theta_0)} \tag{4.21}$$

As for the previous cases, several two-limb configurations were evaluated following this criterion. For a typical two-limb drive configuration, $\ddot{x}_{max} = 2.6$ m/s^2.

4.6.4 Static Balance Criterion

The criterion for static balance consists in having the projection of the CoM on the ground inside the support polygon. However, as the CoM projection reaches the border of the support polygon, it is relatively easier for it to compromise the balance than when it is somewhere far from all the edges. Then, possibly the most desirable position for the CoM is the barycenter of the vertices of the polygon, for rectangular polygons. When CHIMP has only the two lower tracks on the ground, the support polygon is a rectangle, but when the four tracks are on the ground, it can have different polygonal shapes. In these cases, if there are more points in one side of the polygon, they can unnecessarily bias the center, and therefore a different criterion for the most desirable CoM ground projection was proposed.

The point that achieves the "highest" static stability was proposed to be the point that maximizes the minimum distance to any edge of the polygon. Let the support polygon be composed of n_v vertices, each of which is represented as $v_i \in \mathbb{R}^2$ $(i = 1, \cdots, n_v)$, and let the edge between vertex v_i and v_{i+1} be e_i (equivalently, there are $n_e = n_v$ edges). The distance from an arbitrary point $p \in \mathbb{R}^2$ inside the support polygon to edge e_i will be represented by $d_i(p)$, and it is the component in the normal direction $\hat{n}_i$ to the edge. In case the projection of point p onto the edge normal $\hat{n}_i$ lies outside the polygon, the distance to the closest vertex v_i or v_{i+1} will be considered instead. Then, the cost function for point p is determined as

$$C(p) = \min_i\{d_i(p)\} \tag{4.22}$$

and the point that maximizes this distance, and constitutes the best desirable point, is

$$p^* = \max_p \{C(p)\} \tag{4.23}$$

which can be obtained by any optimization method. For any value of the CoM, a criterion to measure its static balance can be obtained by the value given by (4.22); that is, the minimum distance to any edge.

A simple method to obtain the solution to (4.23) is using gradient descent. To this end, an initial guess p_0 consisting of the mean of the vertices is used. Then, letting $\alpha > 0 \in \mathbb{R}$ represent the step size, and p_k be position p at discrete time k, the update rule is

$$p_{k+1} = p_k + \alpha \left(\nabla C(p_k) + \eta \sum_{i=1}^{n_e} \frac{1}{d_i(p_k)} \hat{n}_i \right) \tag{4.24}$$

where $\nabla C(p_k)$ is the gradient of $C(p_k)$ and $\eta > 0 \in \mathbb{R}$ is a weighting factor typically very small. The term in the sum has been added only to avoid getting stuck in a local minimum by pushing in a direction composed of the normals to the edges, and the weights for the normals is inversely proportional to the distance to push the new point towards edges with smaller distance and then maximize this distance. Although this is a heuristic choice, it has given good results for CHIMP.

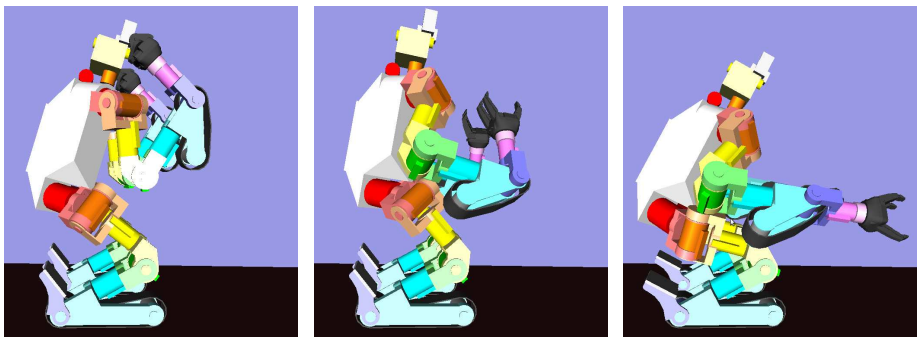
Two-limb Postures for CHIMP

For manipulation, due to time constraints and project management, only the arms were used and the rest of the robot body was kept fixed. Thus, good two-limb postures needed to be found. One characteristic of CHIMP is that its arms are particularly heavy due to the harmonic drive motors. This generates the problem that when manipulating objects, if the CoM is close to the edges of the support polygon, it can tip over. Based on the analysis of the velocity constraints, the stability criterion, as well as some empirical workarounds with different arm configurations, the postures shown in Fig. 4.51 were generated. The two-limb drive posture in Fig. 4.51a satisfies the speed and acceleration constraints specified in the Section 4.6.3. The manipulation postures ensure that for the worst cases, the CoM will be far away from the edge of the support polygon. Two manipulation postures were used, one to grasp objects that are high and the other to grasp objects that are close to the floor.

The two-limb drive posture is shown in the real robot in Fig. 4.52a, and the robot in some tasks of the trials such as the hose task and the wall task are shown in Fig. 4.52. These tasks use the previously determined manipulation tasks and only the arms move to grasp the tools keeping the rest of the body fixed.

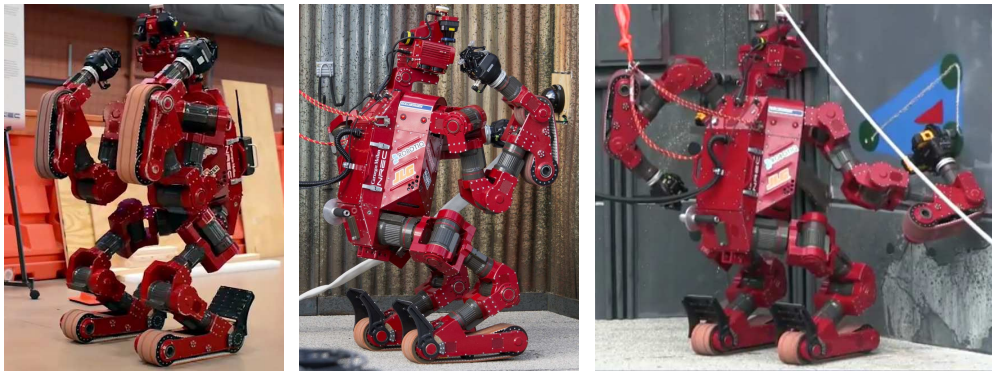
4.7 Conclusion

This chapter has shown the effectiveness of the operational-space inverse dynamics framework for motion generation. Multi-contact handling was experimentally verified with a motion that included the real HRP-2 sitting in an armchair, and the simulated Romeo and HRP-2 climbing a ladder. Human motion imitation was also presented using an acquisition system, an initial motion retargeting based on kinematics and a final retargeting based on the inverse



(a) Two-limb drive **(b)** Manipulation high **(c)** Manipulation low

Figure 4.51 – Two-limb postures for CHIMP in simulation: (a) is used for two-limb drive, (b) is used to manipulate objects that are relatively high, and (c) is used to manipulate objects that are relatively low



(a) Two-limb drive **(b)** Manipulating hose **(c)** Cutting the wall

Figure 4.52 – Examples of two-limb postures in the real robot

dynamics framework, which even allowed for easy edition of the motion. These results were first presented in simulation and then in the real HRP-2. The inverse dynamics stack of tasks was also used for analyzing the organization of human motion, which is a promising whole-body approach that contrasts with simple analysis currently performed in biomechanics.

Another application presented in this chapter includes the incorporation of a walking pattern generator as an input to the inverse dynamics control. After using some heuristics for the feet, the simulated robot was able to walk on rough terrains. Moreover, 3D visual reconstruction was used to guide the foot towards the ground and avoid harsh impacts. To keep balance, the usage of the capture point was introduced as a safe compromise when moving the whole body without over constraining the motion. Finally, several heuristics were used to verify some locomotion constraints associated with CHIMP. Moreover, the inverse dynamics approach can also be applied to it.

Chapter 5

Conclusion

The complete operational-space inverse-dynamics control scheme based on the task-function approach has been used in this thesis to generate dynamically feasible whole-body motion for humanoid robots achieving complex behaviors. Several applications showing the effectiveness of the approach have been presented and discussed.

5.1 Contributions

This thesis extended the usage of the operational-space inverse-dynamics control by adding several tasks to control the whole robot body. These tasks were formulated in terms of equalities and inequalities taking advantage of the capability of the control to use them at any level of the hierarchy. Equality tasks were used when a specific point in a certain space needed to be achieved, such as the pose of an operational point (or some components of it) within strict time constraints, or the position of a point in the visual space of the camera. Inequality tasks were used when some quantity needed to be bounded for the motion to be feasible, for instance, joint position and velocity within certain mechanically imposed limits or the center-of-mass velocity, through the control of the capture point.

The effectiveness of the task-function approach has been demonstrated by generating different types of motions. The multi-contact capabilities of the control scheme were profited to make HRP-2 sit in an armchair using a set of tasks knowing the model of the armchair and the robot, and extensions to other robot or chair models is straightforward. The model of Romeo was also used to make the robot climb up a ladder by adding and removing tasks and contacts at the hands and feet sequentially. This thesis also achieved human motion imitation based on two very efficient tools: motion capture, and the inverse dynamics control. Motion capture has shown to be a very efficient tool for motion generation when combined with the stack of tasks giving more expressiveness to the motion. The SoT control scheme enforced the dynamics of the retargeted motion enabling it to fit with the constraints of the real robot, which are different to those of the human demonstrator. This retargeting approach allowed for the quick and efficient generation of a long sequence of movements for HRP-2 including

motion in single support such as the yoga figure, fast motion in double support, and arbitrary motion of the arms while walking. Moreover, the SoT provides some easy edition capabilities to correct the defects of the targeted motion or to augment the original movement with some artificial features. Using this framework, the work done in this thesis also found that the SoT can be used as a tool to infer the dynamic organization of human motion preserving key features of the human motor control organization.

This thesis also achieved walking steps on rough terrain by detecting collision points with stereo vision reconstruction and moving the foot properly to reach a large support polygon, which is equivalent to foot-land compliance. The approach provides a powerful reactive navigation system able to deal with a large set of uneven grounds thanks to the efficient inverse dynamics setup and the automatic way of finding footsteps. In fact, the only input to the system is the direction of motion and the rest is automatically handled by the framework. The method was limited to horizontal rough surfaces due to the assumptions of the underlying walking pattern generator, but if a different pattern generator is used, the same approach would be valid for surfaces with different slopes. This thesis also presented the incorporation of the capture point inside the inverse dynamics control through the generation of a dedicated task which can bound the velocity of the center of mass. In case a task, or a set of tasks, leads the robot to positions that risk its balance, the capture point task can in principle help the balance to be kept and then a step can be automatically launched to avoid falling down. However, this step needs to be very fast, but it is the way a human would react when suddenly losing balance.

Some movements to generate posture changes for CHIMP were also introduced in this thesis. The heuristic posture change from having four limbs on the ground to only having two limbs on the ground was experimentally verified in the robot. Since the robot can move using its tracks, but keeping its whole body in a pre-determined posture, bounds to the linear and angular velocity as well as linear acceleration were determined and applied to avoid falling down when suddenly de-energizing the motors (which was a design requirement). A stability criterion for more generic support polygons, such as the ones found when CHIMP has all tracks on the ground was established as the minimum distance to the edges. It was stated that the best possible position for the center of mass would be the point that maximizes the minimum distance to the edge of the support polygon. Finally, the inverse dynamics SoT was applied to CHIMP allowing the robot to climb the first rung of a ladder.

5.2 Perspectives

On a short term basis, some practical adjustments can be considered. The inverse dynamics control is currently used to generate the desired acceleration and torque to be applied to the system, but the control applied to HRP-2 is the position obtained by double integration of the acceleration without sensor feedback. The first short term perspective is to apply the inverse dynamics control in closed loop to the robot, modeling the motors of HRP-2 and applying a torque-like control, or acceleration-like control with sensor feedback. Previously this was not achievable due to the computation times but currently the computation time is real-time at 200Hz and the implementation is only constrained by the technological limits currently imposed by the robot. A second perspective in short term is to use a different walking pattern generator scheme to allow for applying the software compliant-foot motion

in not only horizontal terrains but with a different slope. A third perspective is to further investigate the capture point integration in the control scheme, mainly by determining if it would be efficient to apply it for more than one step or a walking scheme would be more adequate in that case.

On a middle term basis, an artificial skin can be added to the sole of HRP-2 to effectively measure the position of the contact points and the corresponding contact force. Then, the approach developed in this thesis in simulation can be directly applied to the real robot. A second perspective is to integrate the kinect to the environment sensing. Alternatively, a lidar can be to scan the 3D environment since it does not depend on lighting conditions as the robot stereo-pair does, and can thus be used for a wide range of environments. The measurements can be fused with the information obtained from the stereo-pair to have a more robust and meaningful environmental reconstruction.

Finally, a long term objective would be to find a way to automatically generate the stack of tasks from high level objectives or from the direct observation of human movements. This very challenging objective, that would allow to remove the roboticist form the loop, requires to be able to automatically structure the observed motion into tasks and constraints, introducing a semantic of the anthropomorphic movement. In this quest, the understanding of the human movement will certainly provide engineers with key elements that are still to be discovered.

Appendices

Appendix A

Generalized Inverses

A.1 Generalized Inverse

Let A be a matrix in $\mathbb{R}^{m\times n}$. A *generalized inverse* of A is a matrix $A_g^{\#} \in \mathbb{R}^{n\times m}$ that satisfies the following condition:

$$AA_g^{\#}A = A$$

which is also called the first Moore-Penrose condition [BenIsrael 03]. A *reflexive generalized inverse* of A is a matrix $A_r^{\#} \in \mathbb{R}^{n\times m}$ that satisfies

$$AA_r^{\#}A = A \qquad \text{and} \qquad A_r^{\#}AA_r^{\#} = A_r^{\#},$$

which are also known as the two first conditions of Moore-Penrose. In this document, as stated in Section 3.2.1, $A^{\#}$ denotes any reflexive generalized inverse of A.

Properties

1. $\forall A \in \mathbb{R}^{m\times n}\ \exists A_g^{\#} \in \mathbb{R}^{n\times m}$, and $\rho(A_g^{\#}) \geq \rho(A)$, where $\rho(A)$ denotes the rank of A.
2. $A_g^{\#} = A_r^{\#} \Leftrightarrow \rho(A_g^{\#}) = \rho(A)$.
3. $AA_g^{\#} = (AA_g^{\#})^2$ and $A_g^{\#}A = (A_g^{\#}A)^2$.
4. If A is square and non-singular, then, $A_g^{\#}$ and $A_r^{\#}$ are unique and $A_g^{\#} = A_r^{\#} = A^{-1}$.
5. Let the sets of $A_g^{\#}$, and $A_r^{\#}$ be $S_g^{\#}$ and $S_r^{\#}$, respectively, then, $S_r^{\#} \subset S_g^{\#}$. Equivalently, any reflexive generalized inverse is a generalized inverse.

A.2 Pseudo-Inverse

The *pseudo-inverse*, also called the *Moore-Penrose inverse* of a matrix $A \in \mathbb{R}^{m \times n}$ is a matrix $A^+ \in \mathbb{R}^{n \times m}$ satisfying the four Moore-Penrose conditions [Penrose 55]:

$$AA^+A = A \qquad (AA^+)^T = AA^+$$
$$A^+AA^+ = A^+ \qquad (A^+A)^T = A^+A$$

For particular cases, the pseudo-inverse A^+ of $A \in \mathbb{R}^{m \times n}$ is given by:

- if $m < n$ and $\rho(A) = m$, then AA^T is nonsingular and $A^+ = A^T(AA^T)^{-1}$,
- if $n < m$ and $\rho(A) = n$, then A^TA is nonsingular and $A^+ = (A^TA)^{-1}A^T$,
- if $m = n = \rho(A)$, then $A^+ = A^{-1}$,
- if $A^TA = I$ (the columns are orthonormal), or $AA^T = I$ (the rows are orthonormal), then $A^+ = A^T$.

Properties

1. $\forall A \in \mathbb{R}^{m \times n}$, A^+ is unique.
2. $(A^T)^+ = (A^+)^T$.
3. $A = (A^+)^+ = (AA^+)^TA = A(A^+A)^T$.
4. $A^+ = (A^TA)^+A^T = A^T(AA^T)^+ = A^+(AA^+)^T = (A^+A)^TA^+$.
5. The matrices A^+A, AA^+, $I - A^+A$, and $I - AA^+$ are symmetric and idempotent.
6. If $A = A^T \in \mathbb{R}^{n \times n}$ and $A^2 = A$, then $\forall B \in \mathbb{R}^{m \times n}$, $A(BA)^+ = (BA)^+$.
7. Given $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{n \times p}$, if (i) $A^TA = I$, or (ii) $BB^T = I$, or (iii) A is full column rank and B is full row rank, then $(AB)^+ = B^+A^+$.

Relation with Linear Systems

Let $A \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$. The problem of finding the general solution to the linear system $Ax = b$, which can be expressed as the minimization problem

$$\min_x \|Ax - b\|_2^2$$

is given by $x^* = A^+b + (I - A^+A)z$, where $z \in \mathbb{R}^n$ is an arbitrary vector and $\|.\|_2$ denotes the L^2 norm of a vector. If additionally, the norm of x has to be minimized; that is,

$$\min_x \{\|x\|_2^2 \quad \text{such that} \quad x = \arg\min \|Ax - b\|_2^2\},$$

the result is given by $x^* = A^+b$, which is also known as the minimum norm solution.

Computation

An efficient way to compute the pseudo-inverse is using the *singular value decomposition* (SVD). The SVD decomposition of $A \in \mathbb{R}^{m \times n}$, with rank $\rho(A) = r$, is:

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

where $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices ($U^T = U^{-1}$, $V^T = V^{-1}$), and $\Sigma \in \mathbb{R}^{m \times n}$ is a diagonal matrix containing the singular values $\sigma_i \in \mathbb{R}$ of A and zeroes in the diagonal. The form of this matrix is

$$\Sigma = \begin{bmatrix} \bar{\Sigma} & \bar{0}_a \\ \bar{0}_b & \bar{0}_c \end{bmatrix} \qquad \text{where} \qquad \bar{\Sigma} = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_r \end{bmatrix} \in \mathbb{R}^{r \times r},$$

$\bar{0}_i$, with $i = \{a, b, c\}$, are zero-matrices such that $\bar{0}_a \in \mathbb{R}^{r \times (n-r)}$, $\bar{0}_b \in \mathbb{R}^{(m-r) \times r}$ and $\bar{0}_c \in \mathbb{R}^{(m-r) \times (n-r)}$, and $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r \geq 0$. Let $u_i \in \mathbb{R}^m$ and $v_i \in \mathbb{R}^n$ represent the i^{th} columns of U and V, respectively, so that

$$U = \begin{bmatrix} u_1 & \cdots & u_r & u_{r+1} & \cdots & u_m \end{bmatrix}$$
$$V = \begin{bmatrix} v_1 & \cdots & v_r & v_{r+1} & \cdots & v_n \end{bmatrix}.$$

The vectors u_i, v_i are the i^{th} left and right singular vectors of A, respectively, satisfying $Av_i = \sigma_i u_i$ and $A^T u_i = \sigma_i v_i$, for $i = 1, \cdots, r$. An orthonormal basis for the image of A is the set $\{u_1, \cdots, u_r\}$, and an orthonormal basis for the kernel of A is the set $\{v_{r+1}, \cdots, v_n\}$. Considering only the first r vectors, A can be reduced to:

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

Since the matrix $\bar{\Sigma}$ is diagonal, its inverse $\bar{\Sigma}^{-1}$ can be obtained replacing σ_i by σ_i^{-1}. Then, the computation of A^+ is directly:

$$A^+ = V \Sigma^{-1} U^T = V \begin{bmatrix} \bar{\Sigma}^{-1} & \bar{0}_a \\ \bar{0}_b & \bar{0}_c \end{bmatrix} U^T = \sum_{i=1}^{r} \frac{1}{\sigma_i} v_i u_i^T$$

In cases where $\sigma_i \to 0$, the denominator might be ill-defined. In these cases, the *damped pseudo-inverse* $A^\dagger$ can be used, and is computed as

$$A^\dagger = \sum_{i=1}^{r} \frac{\sigma_i}{\sigma_i^2 + \alpha^2} v_i u_i^T$$

where α is the damping factor. The damped pseudo-inverse is the solution $x^* = A^\dagger b$ to the problem

$$\min_x \|Ax - b\|_2^2 + \alpha^2 \|x\|_2^2$$

A.3 Weighted Generalized-Inverse

Given a matrix $A \in \mathbb{R}^{m\times n}$ and the positive definite matrices R and L, the *weighted generalized inverse* of A for the weights L and R (also called the L,R-constrained Moore-Penrose inverse of A [Yanai 11]) is a matrix $A^{L\#R} \in \mathbb{R}^{n\times m}$ satisfying the following conditions [Doty 93]:

$$AA^{L\#R}A = A \qquad (LAA^{L\#R})^T = LAA^{L\#R}$$
$$A^{L\#R}AA^{L\#R} = A^{L\#R} \qquad (RA^{L\#R}A)^T = RA^{L\#R}A$$

The weighted generalized inverse $A^{L\#R}$ is obtained as

$$A^{L\#R} = RA^T(A^TLARA^T)^+A^TL = \sqrt{R}(\sqrt{L}A\sqrt{R})^+\sqrt{L}$$

where $\sqrt{R}$ and $\sqrt{L}$ are decompositions such that $\sqrt{R}^T\sqrt{R} = R$ and $\sqrt{L}^T\sqrt{L} = L$, respectively. A usual choice is the Cholesky decomposition.

Particular Cases

The generalized inverse of matrix A weighted on the right by R is:

$$A^{\#R} = \sqrt{R}(A\sqrt{R})^+ = RA^T(ARA^T)^+,$$

and the generalized inverse of A weighted on the left by L is:

$$A^{L\#} = (\sqrt{L}A)^+\sqrt{L} = (A^TLA)^+A^TL.$$

Additionally, depending on the rank of A, the following two cases are found:

- if A is full row-rank, then $A^{\#R} = RA^T(ARA^T)^{-1}$,
- if A is full column-rank, then $A^{L\#} = (A^TLA)^{-1}A^TL$.

Relation with linear systems

Let $A \in \mathbb{R}^{m\times n}$, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, and let the M-weighted norm of vector x be $\|x\|_M = (x^TMx)^{\frac{1}{2}}$, where M is a positive definite matrix. The solution to the following optimization problem

$$\min_x\{\ \|x\|_R^2 \quad \text{such that} \quad x = \arg\min \|Ax-b\|_L^2\} \tag{A.1}$$

is given by $x^* = A^{L\#R}b$. From this generic problem, two particular cases can be identified. The first case is the minimization

$$\min_x \|Ax-b\|_L^2$$

whose solution is given by $x^* = A^{L\#}b$. The second case is the minimization

$$\min_x\{\ \|x\|_R^2 \quad \text{such that} \quad x = \arg\min \|Ax-b\|_2^2\} \tag{A.2}$$

whose solution is given by $x^* = A^{\#R}b$.

Appendix B

Model Predictive Control for Dynamic Walking

B.1 Discrete Dynamic System

Let the robot CoM position be represented by $r_c = (x^c, y^c, z^c)$. The linear inverted pendulum (LIP) model [Kajita 01] assumes that the robot CoM is maintained at a constant height z^c, so that its position can be fully represented by the two horizontal components (x^c, y^c). Since the motion in the horizontal directions (x, y) can be decoupled, only the x components will be explicitly described, but the y components are obtained in a similar way. Using a sampling period T, the discrete variable for the position is noted as $x^c_k = x^c(kT) = x^c(t_k)$, where k is the k-*th* sample, and the state variable comprising the position, velocity and acceleration of the center of mass is noted as $\hat{x}_k = (x^c_k, \dot{x}^c_k, \ddot{x}^c_k)$. The Zero Moment Point (ZMP) on the ground is represented by (x^z, y^z). Using this notation, the discrete dynamic system relating the ZMP and the CoM is [Kajita 03a]:

$$\begin{bmatrix} x^c_{k+1} \\ \dot{x}^c_{k+1} \\ \ddot{x}^c_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^c_k \\ \dot{x}^c_k \\ \ddot{x}^c_k \end{bmatrix} + \begin{bmatrix} \frac{T^3}{6} \\ \frac{T^2}{2} \\ T \end{bmatrix} \dddot{x}^c_k$$

$$x^z = \begin{bmatrix} 1 & 0 & -\frac{z^c}{g} \end{bmatrix} \begin{bmatrix} x^c_k \\ \dot{x}^c_k \\ \ddot{x}^c_k \end{bmatrix}.$$

B.2 Recursive Model

Model predictive control (MPC) generates a control based on the predicted future states using a prediction horizon. Using the previous dynamics recursively, and considering a prediction

horizon of N samples, the velocity of the CoM from time t_{k+1} to t_{k+N} can be expressed as [Wieber 06b]:

$$\dot{X}^c_{k+1} = \begin{bmatrix} \dot{x}^c_{k+1} \\ \vdots \\ \dot{x}^c_{k+N} \end{bmatrix} = P_{vs}\ \hat{x}_k + P_{vu}\dddot{X}^c_k$$

where $\dddot{X}^c_k = (\dddot{x}^c_k, \cdots, \dddot{x}^c_{k+N-1})$ is the jerk of the CoM from time t_k to t_{k+N-1}, and the matrices P_{vs} and P_{vu} are given by

$$P_{vs} = \begin{bmatrix} 0 & 1 & T \\ \vdots & \vdots & \vdots \\ 0 & 1 & NT \end{bmatrix} \qquad P_{vu} = \begin{bmatrix} \frac{T^2}{2} & 0 & 0 \\ \vdots & \ddots & 0 \\ (1+2N)\frac{T^2}{2} & \cdots & \frac{T^2}{2} \end{bmatrix}.$$

In the same time horizon, the ZMP from t_{k+1} to t_{k+N} is expressed as:

$$X^z_{k+1} = \begin{bmatrix} x^z_{k+1} \\ \vdots \\ x^z_{k+N} \end{bmatrix} = P_{zs}\ \hat{x}_k + P_{zu}\dddot{X}^c_k$$

where the matrices P_{zs} and P_{zu} are

$$P_{zs} = \begin{bmatrix} 1 & T & \frac{T^2}{2} - \frac{z^c}{g} \\ \vdots & \vdots & \vdots \\ 1 & NT & \frac{N^2T^2}{2} - \frac{z^c}{g} \end{bmatrix}$$

$$P_{zu} = \begin{bmatrix} \frac{T^3}{6} - \frac{Tz^c}{g} & & 0 & 0 \\ \vdots & & \ddots & \vdots \\ [1+3(N-1)+3(N-1)^2]\frac{T^3}{6} - \frac{Tz^c}{g} & \cdots & \frac{T^3}{6} - \frac{Tz^c}{g} \end{bmatrix}.$$

B.3 MPC based Pattern Generator

The first approach that used optimal control to compute the desired trajectory for the CoM based on the system dynamics was proposed by [Kajita 03a] and uses the jerk of the CoM as control signal: $u_k = (\dddot{X}^c_k, \dddot{Y}^c_k)$. Using both horizontal components $(x,\ y)$ at the same time, and considering the preview horizon from $k+1$ to $k+N$, this approach can be written as the following MPC problem [Wieber 06b]:

$$\min_{u_k} \frac{1}{2} q \left(\|X^z_{k+1} - X^{z_r}_{k+1}\|^2 + \|Y^z_{k+1} - Y^{z_r}_{k+1}\|^2 \right) + \frac{1}{2} r \left(\|\dddot{X}^c_k\|^2 + \|\dddot{Y}^c_k\|^2 \right)$$

where $X^{z_r}_{k+1}$, $Y^{z_r}_{k+1}$ are the desired references for the ZMP, and $q, r \in \mathbb{R}^+$ are positive constants used to balance the minimization of the jerks with the tracking of the reference ZMP. The solution to this problem provides the optimal pattern for the motion of the robot modeled as an inverse pendulum and is, thus, usually referred to as a *pattern generator*.

To automatically decide the foot placement for the steps, the scheme can be modified by regulating the speed of the CoM. The optimization variable in this case is expanded to

$u_k = (\dddot{X}^c_k, \dddot{Y}^c_k, X^f_k, Y^f_k)$, and (X^f_k, Y^f_k) represent the positions on the ground of the following m foot steps such that $X^f_k, Y^f_k \in \mathbb{R}^m$. The resulting optimization problem [Herdt 10], which is used for Section 4.4 in this thesis, is stated as:

$$\begin{aligned}\min_{u_k} \quad & \frac{1}{2}p\left(\|\dot{X}^c_{k+1} - \dot{X}^{c_r}_{k+1}\|^2 + \|\dot{Y}^c_{k+1} - \dot{Y}^{c_r}_{k+1}\|^2\right) \\ & + \frac{1}{2}q\left(\|X^z_{k+1} - X^{z_r}_{k+1}\|^2 + \|Y^z_{k+1} - Y^{z_r}_{k+1}\|^2\right) \\ & + \frac{1}{2}r\left(\|\ddot{X}^c_k\|^2 + \|\ddot{Y}^c_k\|^2\right)\end{aligned}$$

where $\dot{X}^{c_r}_{k+1}$ and $\dot{Y}^{c_r}_{k+1}$ are the desired mean values for the speed of the CoM, and $p \in \mathbb{R}^+$ is a constant value. In this work, the value of the following 2 steps are computed so that $m = 2$. For this system, the ZMP references are not fixed in advance but are permanently recomputed from the feet position decided by the algorithm so that the ZMP lies in the middle of the foot (or of the feet, when there is double support). Let (X^{fc}_k, Y^{fc}_k) be the current position of the foot on the ground. Then, the ZMP references are given by:

$$\begin{aligned}X^{z_r}_{k+1} &= U^c_{k+1}X^{fc}_k + U_{k+1}X^f_k \\ Y^{z_r}_{k+1} &= U^c_{k+1}Y^{fc}_k + U_{k+1}Y^f_k\end{aligned}$$

where U^c_{k+1} and U_{k+1} are binary matrices describing the time at which the feet are in contact along the preview window. The components of these matrices are equal to 1 if the corresponding foot is in contact with the ground (it is a support foot), and are equal to 0 otherwise. U^c_{k+1} is always equal to 1 at the beginning of the preview window because it is the current robot support foot. More concretely, if n_r is the remaining number of iterations for the current robot support foot, and n_s is the necessary number of iterations to realize a step, then

$$U^c_{k+1} = \begin{bmatrix}\mathbf{1}_{n_r} \\ \mathbf{0}_{N-n_r}\end{bmatrix}, \qquad U_{k+1} = \begin{bmatrix}\mathbf{0}_{n_r} & \mathbf{0}_{n_r} \\ \mathbf{1}_{n_s} & \mathbf{0}_{n_s} \\ \mathbf{0}_{N-n_s-n_r} & \mathbf{1}_{N-n_s-n_r}\end{bmatrix}$$

where $\mathbf{1}_n \in \mathbb{R}^n$ and $\mathbf{0}_n \in \mathbb{R}^n$ are column vectors containing only 1's or 0's, respectively. It should be noted that U_{k+1} contains only two columns since only the following 2 steps are considered.

The constraints that ensure that the ZMP remains inside the support polygon, when the robot is in single support phase, can be expressed as:

$$\begin{bmatrix}n_{x_1}(\theta) & n_{y_1}(\theta) \\ n_{x_2}(\theta) & n_{y_2}(\theta) \\ \vdots & \vdots \\ n_{x_E}(\theta) & n_{y_E}(\theta)\end{bmatrix}\begin{bmatrix}x^z - x^f \\ y^z - y^f\end{bmatrix} \leq \begin{bmatrix}\delta_1(\theta) \\ \delta_2(\theta) \\ \vdots \\ \delta_E(\theta)\end{bmatrix}$$

where (x^f, y^f) is the foot position, θ is its orientation, $(n_{x_i}(\theta), n_{y_i}(\theta))$ is the normal vector to the i-th foot edge, $b_i(\theta)$ is a security margin, and it is assumed that there are E edges for the foot. Similar constraints can be formulated for the double support phase [Herdt 10].

Appendix C

Capture Point

This Appendix gives a brief review of the derivation of the Capture Point, which is used for the Capture Point task throughout this document.

C.1 Linear Inverted Pendulum (LIP)

The *linear inverted pendulum* (LIP), described in [Kajita 01], can be used as a model for the analysis of a humanoid robot in the space. It consists in approximating the dynamics of the robot by an inverted pendulum according to some assumptions, which are the following.

1. The robot is represented by a punctual mass m located at its center of mass (CoM).
2. The legs of the robot are massless and the extreme in contact with the ground can be freely moved (it is not externally actuated).
3. The height of the CoM is kept constant throughout the motion.

Consider the LIP model shown in Fig. C.1. Let the position of the CoM be represented by $r_c = (x_c, y_c, z_c)$, the point of the pendulum in contact with the ground by $r_z = (x_z, y_z, 0)$, and the force that acts along the pendulum by $\mathcal{F} = (f_x, f_y, f_z)$. For a robot with feet, the point r_z is equivalent to the Center-of-Pressure (CoP), which is equivalent to the ZMP of the robot [Sardain 04]. With this notation, the equations of motion for the system are:

$$m\ddot{r}_c = \mathcal{F} + m\mathcal{G} \tag{C.1}$$

$$(r_c - r_z) \times \mathcal{F} = 0 \tag{C.2}$$

where $\mathcal{G} = (0, 0, -g)$ is the gravity vector.

Since the third assumptions of the model implies a continuous constant height for the CoM ($\dot{z}_c = 0$), then, both the velocity and the acceleration of the CoM will have a null third component: $\dot{z}_c = \ddot{z}_c = 0$. With this consideration, the components of (C.1) can be written as

$$f_x = m\ddot{x}_c, \qquad f_y = m\ddot{y}_c, \qquad f_z = mg. \tag{C.3}$$

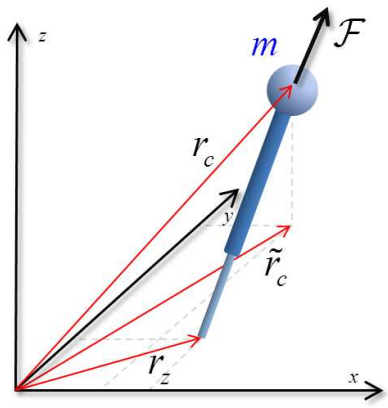


Figure C.1 – 3D Linear Inverted Pendulum (LIP)

Using the equivalence for f_z in (C.3), the first two components of (C.2) can be written as

$$f_x = \frac{mg}{z_c}(x_c - x_z), \qquad f_y = \frac{mg}{z_c}(y_c - y_z)$$

which, using (C.3) for the forces, leads to the equation for the CoM dynamics modeled as a LIP:

$$\ddot{\tilde{r}}_c = \omega^2(\tilde{r}_c - \tilde{r}_z) \tag{C.4}$$

where $\omega = \sqrt{\frac{g}{z_c}}$ is the eigenfrequency of the pendulum, $\tilde{r}_c = (x_c, y_c)$ is the vector containing the horizontal components of the CoM position, and $\tilde{r}_z = (x_z, y_z)$ is the ZMP without considering the third null component.

The solution to the differential equation representing the LIP dynamics (C.4) leads to the temporal evolutions of the position of the CoM horizontal components as:

$$\tilde{r}_c(t) = \frac{1}{2}\left(\tilde{r}_c(0) - \tilde{r}_z + \frac{\dot{\tilde{r}}_c(0)}{\omega}\right)e^{\omega t} + \frac{1}{2}\left(\tilde{r}_c(0) - \tilde{r}_z - \frac{\dot{\tilde{r}}_c(0)}{\omega}\right)e^{-\omega t} + \tilde{r}_z \tag{C.5}$$

where $\tilde{r}_c(0)$ and $\dot{\tilde{r}}_c(0)$ are the initial conditions. It can be seen that in general the trajectory for the center of mass diverges due to the exponential term.

C.2 Capture Point Dynamics

The *Capture Point* (CP), introduced in [Pratt 06b] and also referred to as the *Extrapolated Center of Mass* [Hof 05], is the point $\xi = (\xi_x, \xi_y)$ on the ground where the robot should step on to be able to come to a complete rest.

Modeling the robot as a LIP, the contact of the robot with the ground $\tilde{r}_z$ is not fixed but can move. Moreover, with a robot that has feet, the point $\tilde{r}_z$ corresponding to the CoP can move inside the polygon defined by the supporting foot. If the CoM is moving, the only way for it to stop is by achieving a constant value for $\tilde{r}_c(t)$ as time approaches infinity. In (C.5), as $t \to \infty$, there is a divergence of $\tilde{r}_c(t)$ since $e^{\omega t} \to \infty$. Then, the condition to avoid divergence

is that the coefficient of the exponential be null, that is:

$$\tilde{r}_c(0) + \frac{\dot{\tilde{r}}_c(0)}{\omega} = \tilde{r}_z. \tag{C.6}$$

Using the condition (C.6), the limit for the CoM position (C.5) and for its velocity, which can be easily shown, as time approaches infinity is

$$\lim_{t\to\infty}\{\tilde{r}_c(t)\} = \tilde{r}_z = \xi$$
$$\lim_{t\to\infty}\{\dot{\tilde{r}}_c(t)\} = 0$$

where ξ is called the Capture Point since it is the only point where the CoM can come to a rest. Considering this equivalence for ξ, the condition (C.6) for a general position of the CoM gives the expression for the instantaneous capture point as:

$$\xi = \tilde{r}_c + \frac{\dot{\tilde{r}}_c}{\omega} \tag{C.7}$$

It is also possible to find the expression of the instantaneous capture point by analyzing the orbital energy of the pendulum [Pratt 06a].

The velocity of the capture point can be obtained by differentiating (C.7) and replacing both (C.4) and the expression for the velocity of the CoM from (C.7), and is given by:

$$\dot{\xi} = \omega(\xi - r_z) \tag{C.8}$$

This expression represents the first order dynamics of the CP. Solving the differential equation (C.8), the explicit formulation of the instantaneous capture point trajectory as a function of time is

$$\xi(t) = (\xi_0 - r_z)e^{\omega t} + r_z \tag{C.9}$$

where ξ_0 is ξ at the initial time t_0. The computation of (C.9) allows a prediction of the future position of the capture point.

Bibliography

[Abe 07] Y. Abe, M. da Silva, J. Popović. – Multiobjective control with frictional contacts. – *ACM SIGGRAPH/Eurographics symposium on Computer Animation*, 2007.

[Abry 97] P. Abry. – *Ondelettes et turbulence, Multirésolutions, algorithmes de décomposition, invariance d'échelles*. – Diderot Editeur, Paris, 1997.

[Aghili 05] F. Aghili. – A unified approach for inverse and direct dynamics of constrained multibody systems based on linear projection operator: applications to control and simulation. *IEEE Transactions on Robotics (TRO)*, 21(5):834–849, 2005.

[AlAzemi 13] H. Al-Azemi. – Fire fighting robot, February 2013. US Patent 8,381,826.

[AlBorno 13] M. Al Borno, M. de Lasa, A. Hertzmann. – Trajectory optimization for full-body movements with complex contacts. *IEEE Transactions on Visualization and Computer Graphics*, 19(8):1405–1414, 2013.

[AlBorno 14] M. Al Borno, E. Fiume, A. Hertzmann, M. de Lasa. – Feedback control for rotational movements in feature space. *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2014.

[Ali 10] M. Ali, H. Park, C.S. Lee. – Closed-form inverse kinematic joint solution for humanoid robots. – *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[Ambrose 04] R. O Ambrose, R. T Savely, S. Goza, P. Strawser, M. Diftler, I. Spain, N. Radford. – Mobile manipulation using nasa's robonaut. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2004.

[Arbulu 07] M. Arbulu, C. Balaguer. – Real-time gait planning for rh-1 humanoid robot, using local axis gait algorithm. – *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2007.

[Arimoto 83] S. Arimoto, F. Miyazaki. – Stability and robustness of pid feedback control for robot manipulators of sensory capability. – *Robotics Research, 1st International Symposium*, 1983.

[Arisumi 08] H. Arisumi, S. Miossec, J-R Chardonnet, K. Yokoi. – Dynamic lifting by whole body motion of humanoid robots. – *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008.

[Arnol'd 89] V.I. Arnol'd. – *Mathematical methods of classical mechanics*. – Springer, vol. 60, 1989.

[Asfour 06] T. Asfour, K. Regenstein, P. Azad, J. Schroder, A. Bierbaum, N. Vahrenkamp, R. Dillmann. – Armar-iii: An integrated humanoid platform for sensory-motor control. – *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2006.

[Atkeson 94] C.G. Atkeson. – Using local trajectory optimizers to speed up global optimization in dynamic programming. *Advances in Neural Information Processing Systems*, pp. 663–670, 1994.

[Atkeson 00] C.G. Atkeson, J. Hale, F. Pollick, M. Riley, S. Kotosaka, S. Schaul, T. Shibata, G. Tevatia, A. Ude, S. Vijayakumar. – Using humanoid robots to study human behavior. *IEEE Intelligent Systems and their applications*, 15(4):46–56, 2000.

[Atkeson 03] C.G. Atkeson, J. Morimoto. – Nonparametric representation of policies and value functions: A trajectory-based approach. *Advances in Neural Information Processing Systems*, 15, 2003.

[Ayaz 07] Y. Ayaz, K. Munawar, M. Malik, A. Konno, M. Uchiyama. – Human-like approach to footstep planning among obstacles for humanoid robots. *International Journal of Humanoid Robotics (IJHR)*, 4(01):125–149, 2007.

[Baerlocher 98] P. Baerlocher, R. Boulic. – Task-priority formulations for the kinematic control of highly redundant articulated structures. – *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1998.

[Baerlocher 04] P. Baerlocher, R. Boulic. – An inverse kinematics architecture enforcing an arbitrary number of strict priority levels. *The visual computer*, 20(6):402–417, 2004.

[Baillieul 85] J. Baillieul. – Kinematic programming alternatives for redundant manipulators. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 1985.

[Baillieul 90] J. Baillieul, D.P. Martin. – Resolution of kinematic redundancy. – *Proceedings of Symposia in Applied Mathematics*, 1990.

[Bejczy 74] A.K. Bejczy. – Robot arm dynamics and control, 1974.

[Bellman 56] R. Bellman. – Dynamic programming and lagrange multipliers. *Proceedings of the National Academy of Sciences of the United States of America*, 42(10):767, 1956.

[BenIsrael 03] A. Ben-Israel, T.N.E. Greville. – *Generalized inverses: Theory and applications*. – Springer, 2003.

[Berenson 11] D. Berenson, S.S. Srinivasa, J. Kuffner. – Task Space Regions: A framework for pose-constrained manipulation planning. *The International Journal of Robotics Research (IJRR)*, 2011.

[Bouyarmane 11] K. Bouyarmane, A. Kheddar. – Using a multi-objective controller to synthesize simulated humanoid robot motion with changing contact configurations. – *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[Bouyarmane 12a] K. Bouyarmane, A. Kheddar. – Humanoid robot locomotion and manipulation step planning. *Advanced Robotics*, 26(10):1099–1126, 2012.

[Bouyarmane 12b] K. Bouyarmane, A. Kheddar. – On the dynamics modeling of free-floating-base articulated mechanisms and applications to humanoid whole-body dynamics and control. – *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2012.

[Bretl 06] T. Bretl. – Motion planning of multi-limbed robots subject to equilibrium constraints: The free-climbing robot problem. *The International Journal of Robotics Research (IJRR)*, 25(4):317–342, 2006.

[Brown 13] D. Brown, A. Macchietto, K. Yin, V. Zordan. – Control of rotational dynamics for ground behaviors. – *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2013.

[Caccavale 98] F. Caccavale, C. Natale, B. Siciliano, L. Villani. – Resolved-acceleration control of robot manipulators: A critical review with experiments. *Robotica*, 16(05):565–573, 1998.

[Calinon 07] S. Calinon, F. Guenter, A. Billard. – On learning, representing, and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man and Cybernetics*, 37(2):286–298, 2007.

[Canny 88] J. Canny. – *The complexity of robot motion planning*. – The MIT Press, 1988.

[Capek 20] K. Capek, N. Playfair, P. Selver, WA. Landes. – *Rossum's universal robots*. – Prague, CZ, 1920.

[Chang 08] Y. Chang, Y. Oh, D. Kim, S. Hong. – Balance control in whole body coordination framework for biped humanoid robot mahru-r. – *The 17th IEEE International Symposium on Robot and Human Interactive Communication*, 2008.

[Chaumette 06] F. Chaumette, S. Hutchinson. – Visual servo control. I. Basic approaches. *IEEE Robotics & Automation Magazine (RAM)*, 13(4):82–90, 2006.

[Cheng 07] G. Cheng, S. Hyon, J. Morimoto, A. Ude, J. Hale, G. Colvin, W. Scroggin, S. Jacobsen. – Cb: A humanoid research platform for exploring neuroscience. *Advanced Robotics*, 21(10):1097–1114, 2007.

[Chestnutt 05] J. Chestnutt, M. Lau, G. Cheung, J. Kuffner, J. Hodgins, T. Kanade. – Footstep planning for the honda asimo humanoid. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2005.

[Choset 05] H. Choset, K.M. Lynch, S. Hutchinson, G.A. Kantor, W. Burgard, L.E. Kavraki, S. Thrun. – *Principles of Robot Motion: theory, algorithms, and implementation.* – MIT Press, 2005.

[Coleman 00] R. Coleman. – Multi-feature automated wheelchair, November 2000. US Patent 6,154,690.

[Collette 08] C. Collette, A. Micaelli, C. Andriot, P. Lemerle. – Robust balance optimization control of humanoid robots with multiple non coplanar grasps and frictional contacts. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2008.

[Craig 89] J.J. Craig. – *Introduction to robotics.* – Addison-Wesley Reading, MA, vol. 7, 1989.

[Dalibard 13] S. Dalibard, A. El Khoury, F. Lamiraux, A. Nakhaei, M. Taïx, J-P. Laumond. – Dynamic walking and whole-body motion planning for humanoid robots: an integrated approach. *The International Journal of Robotics Research (IJRR)*, 2013.

[Dariush 10] B. Dariush, G. Hammam, D. Orin. – Constrained resolved acceleration control for humanoids. – *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[DaSilva 08] M. Da Silva, Y. Abe, J. Popović. – Simulation of human motion data using short-horizon model-predictive control. – *Computer Graphics Forum*, 2008.

[deLasa 10] M. de Lasa, I. Mordatch, A. Hertzmann. – Feature-based locomotion controllers. *ACM Transactions on Graphics (SIGGRAPH)*, 29(3), 2010.

[Dellin 12] C. Dellin, S. Srinivasa. – A framework for extreme locomotion planning. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2012.

[Dellon 07] B. Dellon, Y. Matsuoka. – Prosthetics, exoskeletons, and rehabilitation. *IEEE Robotics & Automation Magazine (RAM)*, 14(1):30–34, 2007.

[DelPrete 14a] A. Del Prete, F. Nori, G. Metta, L. Natale. – Partial force control of constrained floating-base robots. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2014.

[DelPrete 14b] A. Del Prete, F. Romano, F. Nori, G. Metta, G. Sandini, L. Natale. – Prioritized optimal control. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2014.

[Denavit 55] J. Denavit, R. S. Hartenberg. – A kinematic notation for lower-pair mechanisms based on matrices. *ASME Journal of Applied Mechanics*, 22:215–221, 1955.

[Dimitrov 11] D. Dimitrov, A. Sherikov, P-B Wieber. – A sparse model predictive control formulation for walking motion generation. – *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[Doty 93] K.L. Doty, C. Melchiorri, C. Bonivento. – A theory of generalized inverses applied to robotics. *The International Journal of Robotics Research (IJRR)*, 12(1):1, 1993.

[Dubowsky 79] S. Dubowsky, DT DesForges. – The application of model-referenced adaptive control to robotic manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 101(3):193–200, 1979.

[Durán 12] B. Durán, S. Thill. – Rob's robot: Current and future challenges for humanoid robots. *In The Future of Humanoid Robots-Research and Applications*, 2012.

[ElKhoury 13] A. El Khoury, F. Lamiraux, M. Taix. – Optimal motion planning for humanoid robots. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2013.

[Endo 08] N. Endo, S. Momoki, M. Zecca, M. Saito, Y. Mizoguchi, K. Itoh, A. Takanishi. – Development of whole-body emotion expression humanoid robot. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2008.

[Englsberger 11] J. Englsberger, C. Ott, M. Roa, A. Albu-Schaffer, G. Hirzinger. – Bipedal walking control based on capture point dynamics. – *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[Escande 13] A. Escande, A. Kheddar, S. Miossec. – Planning contact points for humanoid robots. *Robotics and Autonomous Systems*, 2013.

[Escande 14] A. Escande, N. Mansard, P-B. Wieber. – Hierarchical quadratic programming. *The International Journal of Robotics Research (IJRR)*, 2014.

[Everett 87] L. Everett, M. Driels, B.W. Mooring. – Kinematic modelling for robot calibration. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 1987.

[Featherstone 00] R. Featherstone, D. Orin. – Robot dynamics: Equations and algorithms. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2000.

[Featherstone 08] R. Featherstone. – *Rigid body dynamics algorithms*. – Springer New York, vol. 49, 2008.

[Fink 12] J. Fink. – Anthropomorphism and human likeness in the design of robots and human-robot interaction. *Social Robotics*, pp. 199–208. – Springer, 2012.

[Galdeano 14] D. Galdeano, A. Chemori, S. Krut, P. Fraisse. – Task-based whole-body control of humanoid robots with zmp regulation, real-time application to a squat-like motion. – *SSD SAC 2014: 11th International Conference on Systems, Analysis and Automatic Control*, 2014.

[Geiger 10] M. Geiger, M. Roser, R. Urtasun. – Efficient large-scale stereo matching. – *Asian Conference on Computer Vision*, 2010.

[Geoffroy 14] P. Geoffroy, N. Mansard, M. Raison, S. Achiche, Y. Tassa, E. Todorov. – From inverse kinematics to optimal control. *Advances in Robot Kinematics*, 2014.

[Geppert 04] L. Geppert. – Qrio, the robot that could. *IEEE Spectrum*, 41(5):34–37, 2004.

[Gienger 05] M. Gienger, H. Janssen, C. Goerick. – Task-oriented whole body motion for humanoid robots. – *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2005.

[Gleicher 98] M. Gleicher. – Retargetting motion to new characters. – *ACM Transactions on Graphics (SIGGRAPH)*, 1998.

[Gogu 05] G. Gogu. – Chebychev–grübler–kutzbach's criterion for mobility calculation of multi-loop mechanisms revisited via theory of linear transformations. *European Journal of Mechanics-A/Solids*, 24(3):427–441, 2005.

[Goodman 04] J.E. Goodman, J. O'Rourke. – *Handbook of discrete and computational geometry*. – CRC press, 2004.

[Goswami 99] A. Goswami. – Foot rotation indicator (fri) point: A new gait planning tool to evaluate postural stability of biped robots. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 1999.

[Gottschalk 96] S. Gottschalk, M.C. Lin, D. Manocha. – Obbtree: a hierarchical structure for rapid interference detection. – *Conference on Computer Graphics and Interactive Techniques*, 1996.

[Gouaillier 09] D. Gouaillier, V. Hugel, P. Blazevic, Ch. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, B. Maisonnier. – Mechatronic design of nao humanoid. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, Kobe, Japan, 2009.

[Grey 13] M. Grey, N. Dantam, D. Lofaro, A. Bobick, M. Egerstedt, P. Oh, M. Stilman. – Multi-process control software for hubo2 plus robot. – *IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, 2013.

[Guigon 07] E. Guigon, P. Baraduc, M. Desmurget. – Computational motor control: redundancy and invariance. *Journal of neurophysiology*, 97(1):331–347, 2007.

[H. 11] Sovannara H. – *Reconnaissance de Tâches par Commande Inverse*. – PhD. Thesis, University of Toulouse (INSA Toulouse) and LAAS-CNRS, November 2011.

[Hak 12] S. Hak, N. Mansard, O. Stasse, J-P. Laumond. – Reverse control for humanoid robot task recognition. *IEEE Transactions on Systems, Man and Cybernetics*, 42(6):1524–1537, 2012.

[Harmeyer 04] S. Harmeyer, A. Bowling. – Dynamic performance as a criterion for redundant manipulator control. – *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.

[Hauser 10] K. Hauser, J.C. Latombe. – Multi-modal motion planning in non-expansive spaces. *The International Journal of Robotics Research (IJRR)*, 29(7):897–915, 2010.

[Hayati 85] S. Hayati, M. Mirmirani. – Improving the absolute positioning accuracy of robot manipulators. *Journal of Robotic Systems*, 2(4):397–413, 1985.

[Herdt 10] A. Herdt, H. Diedam, P.B. Wieber, D. Dimitrov, K. Mombaur, M. Diehl. – Online walking motion generation with automatic footstep placement. *Advanced Robotics, 24*, 5(6):719–737, 2010.

[Herzog 13] A. Herzog, L. Righetti, F. Grimminger, P. Pastor, S. Schaal. – Experiments with a hierarchical inverse dynamics controller on a torque-controlled humanoid. *arXiv preprint*, 2013.

[Hirai 98] K. Hirai, M. Hirose, Y. Haikawa, T. Takenaka. – The development of Honda humanoid robot. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 1998.

[Hirose 01] R. Hirose, T. Takenaka. – Development of the humanoid robot ASIMO. *Honda R&D Technical Review*, 13(1):1–6, 2001.

[Hockstein 07] N. Hockstein, C. Gourin, R. Faust, D. Terris. – A history of robots: from science fiction to surgical robotics. *Journal of robotic surgery*, 1(2):113–118, 2007.

[Hof 05] A.L. Hof, M. Gazendam, W. Sinke. – The condition for dynamic stability. *Journal of biomechanics*, 38(1):1–8, 2005.

[Hofmann 09] A. Hofmann, M. Popovic, H. Herr. – Exploiting angular momentum to enhance bipedal center-of-mass control. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2009.

[Hollerbach 80] J.M. Hollerbach. – A recursive lagrangian formulation of maniputator dynamics and a comparative study of dynamics formulation complexity. *IEEE Transactions on Systems, Man and Cybernetics*, 10(11):730–736, 1980.

[Hornby 00] G. Hornby, S. Takamura, J. Yokono, O. Hanagata, T. Yamamoto, M. Fujita. – Evolving robust gaits with aibo. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2000.

[Hornung 13] A. Hornung, D. Maier, M. Bennewitz. – Search-based footstep planning. – *ICRA - Workshop on Progress and Open Problems in Motion Planning and Navigation for Humanoids*, Karlsruhe, Germany, May 2013.

[Hsu 97] D. Hsu, J-C. Latombe, R. Motwani. – Path planning in expansive configuration spaces. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 1997.

[Hudson 97] T.C. Hudson, M.C. Lin, J. Cohen, S. Gottschalk, D. Manocha. – V-collide: accelerated collision detection for vrml. – *Symposium on Virtual Reality Modeling Language*, 1997.

[Hunt 67] KH Hunt. – Prismatic pairs in spatial linkages. *Journal of Mechanisms*, 2(2):213–230, 1967.

[Idel 90] M. Idel. – *Golem: Jewish magical and mystical traditions on the artificial anthropoid*. – SUNY Press, 1990.

[Inaba 03] M. Inaba, I. Mizuuchi, R. Tajima, T. Yoshikai, D. Sato, K. Nagashima, H. Inoue. – Building spined muscle-tendon humanoid. *Robotics Research*, pp. 113–127. – 2003.

[Indyk 04] P. Indyk, J. Matousek. – *Low-distorsion embeddings of finite metric spaces, Handbook of Discrete and Computational Geometry*. – MIT Press, Cambridge, MA, USA, 2004.

[Jacobson 68] D. Jacobson. – New second-order and first-order algorithms for determining optimal control: A differential dynamic programming approach. *Journal of Optimization Theory and Applications*, 2(6):411–440, 1968.

[Jaillet 08] L. Jaillet, J. Cortés, T. Siméon. – Transition-based rrt for path planning in continuous cost spaces. – *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008.

[Jarquin 13] G. Jarquin, A. Escande, G. Arechavaleta, T. Moulard, E. Yoshida, V. Parra-Vega. – Real-time smooth task transitions for hierarchical inverse kinematics. – *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2013.

[Jones 06] J. Jones. – Robots at the tipping point: the road to irobot roomba. *IEEE Robotics & Automation Magazine (RAM)*, 13(1):76–78, 2006.

[Kahan 83] W. Kahan. – Lectures on computational aspects of geometry, 1983.

[Kahn 71] M. Kahn, B. Roth. – The near-minimum-time control of open-loop articulated kinematic chains. *Journal of Dynamic Systems, Measurement, and Control*, 93(3):164–172, 1971.

[Kajita 01] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, H. Hirukawa. – The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation. – *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2001.

[Kajita 03a] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, H. Hirukawa. – Biped walking pattern generation by using preview control of zero-moment point. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, Taipei, Taiwan, 2003.

[Kajita 03b] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, H. Hirukawa. – Resolved momentum control: Humanoid motion planning based on the linear and angular momentum. – *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003.

[Kajita 05] S. Kajita, H. Hirukawa, K. Harada, K. Yokoi. – *Humanoid Robot*. – Omsha, Tokyo, 2005. in Japanese.

[Kalakrishnan 11] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, S. Schaal. – Stomp: Stochastic trajectory optimization for motion planning. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2011.

[Kalman 60] R. Kalman. – Contributions to the theory of optimal control. *Boletin de la Sociedad Matematica Mexicana*, 5(2):102–119, 1960.

[Kaneko 02] K. Kaneko, F. Kanehiro, S. Kajita, K. Yokoyama, K. Akachi, T. Kawasaki, S. Ota, T. Isozumi. – Design of prototype humanoid robotics platform for hrp. – *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2002.

[Kaneko 04] K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukawa, T. Kawasaki, M. Hirata, K. Akachi, T. Isozumi. – Humanoid robot HRP-2. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2004.

[Kaneko 08] K. Kaneko, K. Harada, F. Kanehiro, G. Miyamori, K. Akachi. – Humanoid robot hrp-3. – *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008.

[Kaneko 09] K. Kaneko, F. Kanehiro, M. Morisawa, K. Miura, S. Nakaoka, S. Kajita. – Cybernetic human HRP-4C. – *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2009.

[Kaneko 11] K. Kaneko, F. Kanehiro, M. Morisawa, K. Akachi, G. Miyamori, A. Hayashi, N. Kanehira. – Humanoid robot hrp-4-humanoid robotics platform with lightweight and slim body. – *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[Kanoun 09] O. Kanoun, F. Lamiraux, P.B. Wieber, F. Kanehiro, E. Yoshida, J.P. Laumond. – Prioritizing linear equality and inequality systems: application to local motion planning for redundant robots. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2009.

[Karaman 11] S. Karaman, E. Frazzoli. – Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research (IJRR)*, 30(7):846–894, 2011.

[Kato 73] I. Kato. – Development of WABOT 1. *Biomechanism*, 2:173–214, 1973.

[Kavraki 96] L.E. Kavraki, P. Svestka, J.-C. Latombe, M.H. Overmars. – Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566 –580, 1996.

[Khalil 86] W. Khalil, J. Kleinfinger. – A new geometric notation for open and closed-loop robots. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 1986.

[Khalil 04] W. Khalil, E. Dombre. – *Modeling, identification and control of robots*. – Butterworth-Heinemann, 2004.

[Khatib 80] O. Khatib. – *Commande dynamique dans l'espace opérationnel des robots manipulateurs en présence d'obstacles*. – PhD. Thesis, L'École Nationale Supérieure de l'Aéronautique et de l'Espace, 1980.

[Khatib 86] O. Khatib. – Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research (IJRR)*, 5(1):90–98, 1986.

[Khatib 87] O. Khatib. – A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal of Robotics and Automation*, 3(1):43–53, 1987.

[Khatib 04a] O. Khatib. – Advanced robotic manipulation, 2004.

[Khatib 04b] O. Khatib, L. Sentis, J. Park, J. Warren. – Whole-body dynamic behavior and control of human-like robots. *International Journal of Humanoid Robotics (IJHR)*, 1(01):29–43, 2004.

[Khatib 04c] O. Khatib, L. Sentis, J. Park, J. Warren. – Whole body dynamic behavior and control of human-like robots. *The International Journal of Robotics Research (IJRR)*, 1(1):29–44, 2004.

[Kim 02] J. Kim, S. Park, I. Park, J. Oh. – Development of a humanoid biped walking robot platform khr-1-initial design and its performance evaluation. *Feedback*, 1:1–12, 2002.

[Kim 09] J Kim, S. Choi, D. Kim, J. Kim, M. Cho. – Animal-robot interaction for pet caring. – *IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, 2009.

[Kim 11] K. Kim, Y. Cha, J. Park, J. Lee, B. You. – Providing services using network-based humanoids in a home environment. *IEEE Transactions on Consumer Electronics*, 57(4):1628–1636, 2011.

[Kim 12] J. Kim, Y. Lee, S. Kwon, K. Seo, H. Kwak, H. Lee, K. Roh. – Development of the lower limbs for a humanoid robot. – *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[Kitano 98] H. Kitano, M. Asada. – They robocup humanoid challenge as the millennium challenge for advanced robotics. *Advanced Robotics*, 13(8):723–736, 1998.

[Klarbring 97] A. Klarbring. – Steady sliding and linear complementarity. *Complementarity and Variational Problems: State of the Art*, pp. 132–147, 1997.

[Knabe 13] C. Knabe, M. Hopkins, D. Hong. – Team charli: Robocup 2012 humanoid adultsize league winner. *RoboCup 2012: Robot Soccer World Cup XVI*, pp. 59–64. – 2013.

[Koch 12] K. Koch, K. Mombaur, P. Souères. – Studying the effect of different optimization criteria on humanoid walking motions. *Simulation, Modeling, and Programming for Autonomous Robots*, pp. 221–236. – Springer, 2012.

[Kuffner 00] Jr. Kuffner, J.J., S.M. LaValle. – Rrt-connect: An efficient approach to single-query path planning. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2000.

[Kuindersma 14] S. Kuindersma, F. Permenter, R. Tedrake. – An efficiently solvable quadratic program for stabilizing dynamic locomotion. *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2014.

[Kusuda 08] Y. Kusuda. – Toyota's violin-playing robot. *Industrial Robot: An International Journal*, 35(6):504–506, 2008.

[Kwon 07] W. Kwon, H. Kim, J. Park, Ch. Roh, J. Lee, J. Park, W. Kim, K. Roh. – Biped humanoid robot mahru iii. – *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2007.

[Latombe 91] J-C. Latombe. – *Robot Motion Planning*. – Kluwer Academic Publishers, Norwell, MA, USA, 1991.

[Laumond 12] J-P. Laumond. – *La robotique: une récidive d'Héphaïstos*. – Collège de France/Fayard (In French), April 2012.

[LaValle 06] S.M. LaValle. – *Planning Algorithms*. – Cambridge University Press, Cambridge, U.K., 2006.

[Lee 99] J. Lee, S.Y. Shin. – A hierarchical approach to interactive motion editing for human-like figures. *ACM Transactions on Graphics (SIGGRAPH)*, pp. 39–48, 1999.

[Lee 12] K. Lee, K. Oh. – Ubiquitous robot and its realization. *Applied Mathematics and Information Sciences*, 6(1S):311S–321S, 2012.

[Lengagne 13] S. Lengagne, J. Vaillant, E. Yoshida, A. Kheddar. – Generation of whole-body optimal dynamic multi-contact motions. *The International Journal of Robotics Research (IJRR)*, 2013.

[Levine 13] S. Levine, V. Koltun. – Guided policy search. – *International Conference on Machine Learning*, 2013.

[Liegeois 77] A. Liegeois. – Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Transactions on Systems, Man, and Cybernetics*, 7(12):868–871, 1977.

[Lohmeier 09] S. Lohmeier, T. Buschmann, H. Ulbrich. – Humanoid robot lola. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2009.

[Lozano 92] R Lozano, B Brogliato, ID Landau. – Passivity and global stabilization of cascaded nonlinear systems. *IEEE Transactions on Automatic Control*, 37(9):1386–1388, 1992.

[LozanoPerez 83] T. Lozano-Perez. – Spatial planning: A configuration space approach. *IEEE Transactions on Computers*, 100(2):108–120, 1983.

[Luh 80] J. Luh, M. Walker, R. Paul. – Resolved-acceleration control of mechanical manipulators. *IEEE Transactions on Automatic Control*, 25(3):468–474, 1980.

[Macchietto 09] A. Macchietto, V. Zordan, C. Shelton. – Momentum control for balance. *ACM Transactions on Graphics (SIGGRAPH)*, 28(3), 2009.

[Maier 13] D. Maier, C. Lutz, M. Bennewitz. – Integrated perception, mapping, and footstep planning for humanoid navigation among 3d obstacles. – *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[Mansard 07] N. Mansard, F. Chaumette. – Task sequencing for high-level sensor-based control. *IEEE Transactions on Robotics (TRO)*, 23(1):60–72, 2007.

[Mansard 12] N. Mansard. – A dedicated solver for fast operational-space inverse dynamics. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, St Paul, MN, USA, 2012.

[Marchionni 13] L. Marchionni, J. Pages, J. Adell, J. Capriles, H. Tomé. – Reem service robot: How may i help you? *Natural and Artificial Models in Computation and Biology*, pp. 121–130. – 2013.

[Markiewicz 73] BR Markiewicz. – Analysis of the computed torque drive method and comparison with conventional position servo for a computer-controlled manipulator, 1973.

[Metta 10] G. Metta, L. Natale, F. Nori, G. Sandini, D. Vernon, L. Fadiga, C. von Hofsten, K. Rosander, M. Lopes, J. Santos-Victor. – The iCub humanoid robot: An open-systems platform for research in cognitive development. *Neural Networks*, 23(8-9):1125–1134, 2010.

[Miossec 06] S. Miossec, K. Yokoi, A. Kheddar. – Development of a software for motion optimization of robots-application to the kick motion of the hrp-2 robot. – *IEEE International Conference on Robotics and Biomimetics*, 2006.

[Mistry 10] M. Mistry, J. Buchli, S. Schaal. – Inverse dynamics control of floating base systems using orthogonal decomposition. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2010.

[Mizuuchi 06] I. Mizuuchi, T. Yoshikai, Y. Sodeyama, Y. Nakanishi, A. Miyadera, T. Yamamoto, T. Niemela, M. Hayashi, J. Urata, Y Namiki. – Development of musculoskeletal humanoid kotaro. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2006.

[Mizuuchi 07] I. Mizuuchi, Y. Nakanishi, Y. Sodeyama, Y. Namiki, T. Nishino, N. Muramatsu, J. Urata, K. Hongo, T. Yoshikai, M. Inaba. – An advanced musculoskeletal humanoid kojiro. – *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2007.

[Mombaur 10] K. Mombaur, A. Truong, JP Laumond. – From human to humanoid locomotion an inverse optimal control approach. *Autonomous robots*, 28(3):369–383, 2010.

[Monje 11] C. Monje, S. Martinez, A. Jardón, P. Pierro, C. Balaguer, D. Munoz. – Full-size humanoid robot teo: Design attending mechanical robustness and energy consumption. – *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2011.

[Morisawa 11] M. Morisawa, F. Kanehiro, K. Kaneko, S. Kajita, K. Yokoi. – Reactive biped walking control for a collision of a swinging foot on uneven terrain. – *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2011.

[Moro 13] F. Moro, M. Gienger, A. Goswami, N. Tsagarakis, D.G. Caldwell. – An attractor-based whole-body motion control (wbmc) system for humanoid robots. – *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2013.

[Murray 94] R. Murray, Z. Li, S. Sastry. – *A mathematical introduction to robotic manipulation*. – CRC press, 1994.

[Nakamura 86] Y. Nakamura, H. Hanafusa. – Inverse kinematic solutions with singularity robustness for robot manipulator control. *Journal of Dynamic Systems, Measurement and Control*, 108(3):163–171, 1986.

[Nakamura 87] Y. Nakamura, H. Hanafusa, T. Yoshikawa. – Task-priority based redundancy control of robot manipulators. *The International Journal of Robotics Research (IJRR)*, 6(2):3–15, 1987.

[Nakamura 90] Y. Nakamura. – *Advanced robotics: redundancy and optimization*. – Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1990.

[Nakanishi 08] J. Nakanishi, R. Cory, M. Mistry, J. Peters, S. Schaal. – Operational space control: A theoretical and empirical comparison. *The International Journal of Robotics Research (IJRR)*, 27(6):737–757, 2008.

[Nakanishi 12] Y. Nakanishi, Y. Asano, T. Kozuki, H. Mizoguchi, Y. Motegi, M. Osada, T. Shirai, J. Urata, K. Okada, M. Inaba. – Design concept of detail musculoskeletal humanoid kenshiro - toward a real human body musculoskeletal simulator. – *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2012.

[Nelson 12] G. Nelson, A. Saunders, N. Neville, B. Swilling, J. Bondaryk, D. Billings, C. Lee, R. Playter, M. Raibert. – Petman: A humanoid robot for testing chemical protective clothing. *Journal of the Robotics Society of Japan*, 30(4):372–377, 2012.

[Neo 07] E. Neo, K. Yokoi, S. Kajita, K. Tanie. – Whole-body motion generation integrating operator's intention and robot's autonomy in controlling humanoid robots. *IEEE Transactions on Robotics (TRO)*, 23(4):763–775, 2007.

[Newcombe 11] R.A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A.J. Davison, P. Kohli, J. Shotton, S. Hodges, A. Fitzgibbon. – KinectFusion: Real-time dense surface mapping and tracking. – *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Washington, DC, USA, 2011.

[Niechwiadowicz 08] K. Niechwiadowicz, Z. Khan. – Robot based logistics system for hospitals-survey. – *IDT Workshop on Interesting Results in Computer Science and Engineering*, 2008.

[Nilsson 84] N.J. Nilsson. – Shakey the robot, 1984.

[Nishiwaki 07] K. Nishiwaki, J. Kuffner, S. Kagami, M. Inaba, H. Inoue. – The experimental humanoid robot h7: a research platform for autonomous behaviour. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1850):79, 2007.

[Nishiwaki 12] K. Nishiwaki, J. Chestnutt, S. Kagami. – Autonomous navigation of a humanoid robot over unknown rough terrain using a laser range sensor. *The International Journal of Robotics Research (IJRR)*, 31(11):1251–1262, 2012.

[Noda 14] S. Noda, M. Murooka, S. Nozawa, Y. Kakiuchi, K. Okada, M. Inaba. – Generating whole-body motion keep away from joint torque, contact force, contact moment limitations enabling steep climbing with a real humanoid robot. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, Hong Kong, China, 2014.

[Nunez 12] J.V. Nunez, A. Briseno, D. A Rodriguez, J.M. Ibarra, V.M. Rodriguez. – Explicit analytic solution for inverse kinematics of bioloid humanoid robot. – *Brazilian Robotics Symposium and Latin American Robotics Symposium (SBR-LARS)*, 2012.

[Oberman 07] L. M Oberman, J. McCleery, V. Ramachandran, J. Pineda. – Eeg evidence for mirror neuron activity during the observation of human and robot actions: Toward an analysis of the human qualities of interactive robots. *Neurocomputing*, 70(13):2194–2203, 2007.

[Ogura 06] Y. Ogura, H. Aikawa, K. Shimomura, A. Morishima, H. Lim, A. Takanishi. – Development of a new humanoid robot WABIAN-2. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2006.

[Oh 06] J.H. Oh, D. Hanson, W. Kim, Y. Han, J. Kim, I. Park. – Design of android type humanoid robot albert hubo. – *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006.

[Orin 79] D. Orin, R. McGhee, M. Vukobratović, G. Hartoch. – Kinematic and kinetic analysis of open-chain linkages utilizing newton-euler methods. *Mathematical Biosciences*, 43(1):107–130, 1979.

[Orin 08] D. E Orin, A. Goswami. – Centroidal momentum matrix of a humanoid robot: Structure and properties. – *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008.

[Orin 13] D. Orin, A. Goswami, S. Lee. – Centroidal dynamics of a humanoid robot. *Autonomous Robots*, 35(2-3):161–176, 2013.

[Ott 10] C. Ott, C. Baumgartner, J. Mayr, M. Fuchs, R. Burger, D. Lee, O. Eiberger, A. Albu-Schaffer, M. Grebenstein, G. Hirzinger. – Development of a biped robot with torque controlled joints. – *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2010.

[Ott 11] C. Ott, M. A. Roa, G. Hirzinger. – Posture and balance control for biped robots based on contact force optimization. – *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2011.

[Paden 86] B.E. Paden. – *Kinematics and Control of Robot Manipulators*. – PhD. Thesis, University of California, Berkeley, 1986.

[Paipetis 10] SA Paipetis. – The robots of hephaestus. *The Unknown Technology in Homer*, pp. 107–111. – Springer, 2010.

[Park 94] F.C. Park. – Computational aspects of the product-of-exponentials formula for robot kinematics. *IEEE Transactions on Automatic Control*, 39(3):643–647, 1994.

[Park 01] K. Park, P. Chang, S. Lee. – Analysis and control of redundant manipulator dynamics based on an extended operational space. *Robotica*, 19(6):649–662, 2001.

[Park 05a] I. Park, J. Kim, S. Park, J. Oh. – Development of humanoid robot platform khr-2 (kaist humanoid robot 2). *International Journal of Humanoid Robotics (IJHR)*, 2(04):519–536, 2005.

[Park 05b] I.W. Park, J.Y. Kim, J. Lee, J.H. Oh. – Mechanical design of humanoid robot platform KHR-3 (KAIST humanoid robot 3: HUBO). – *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2005.

[Park 13] T. Park, S. Levine. – Inverse optimal control for humanoid locomotion. – *Robotics Science and Systems - Workshop on Inverse Optimal Control and Robotic Learning from Demonstration*, 2013.

[Park 14] C. Park, J. Pan, D. Manocha. – High-dof robots in dynamic environments using incremental trajectory optimization. *International Journal of Humanoid Robotics (IJHR)*, 2014.

[Paul 81] R.P. Paul. – *Robot manipulators: mathematics, programming, and control*. – MIT Press, 1981.

[Peiper 68] D.L. Peiper. – *The kinematics of manipulators under computer control*. – PhD. Thesis, Stanford University, 1968.

[Penrose 55] R. Penrose. – A generalized inverse for matrices. – *Mathematical proceedings of the Cambridge philosophical society*, 1955.

[Perez 12] A. Perez, R. Platt, G. Konidaris, L. Kaelbling, T. Lozano-Perez. – Lqr-rrt*: Optimal sampling-based motion planning with automatically derived extension heuristics. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2012.

[Perrin 12] N. Perrin, O. Stasse, L. Baudouin, F. Lamiraux, E. Yoshida. – Fast humanoid robot collision-free footstep planning using swept volume approximations. *IEEE Transactions on Robotics (TRO)*, 28(2):427–439, 2012.

[Peters 08] J. Peters, M. Mistry, F. Udwadia, J. Naka, S. Schaal. – A unifying framework for robot control with redundant DOFs. *Autonomous Robots*, 24(1):1–12, 2008.

[Pontryagin 62] L. Pontryagin, V. Boltyanskii, R. Gamkrelidze. – *The mathematical theory of optimal processes*. – New York, 1962.

[Popovic 00] Z. Popovic. – Editing dynamicproperties of captured human motion. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, San Francisco, CA, USA, 2000.

[Popovic 04] M. Popovic, A. Hofmann, H. Herr. – Angular momentum regulation during human walking: biomechanics and control. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2004.

[Pratt 06a] J. Pratt, J. Carff, S. Drakunov, A. Goswami. – Capture point: A step toward humanoid push recovery. – *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2006.

[Pratt 06b] J. E Pratt, R. Tedrake. – Velocity-based stability margins for fast bipedal walking. *Fast Motions in Biomechanics and Robotics*, pp. 299–324. – Springer, 2006.

[Raghavan 93] M. Raghavan, B. Roth. – Inverse kinematics of the general 6r manipulator and related linkages. *Journal of Mechanical Design*, 115(3):502–508, 1993.

[Ramos 11] O. Ramos, L. Saab, S. Hak, N. Mansard. – Dynamic motion capture and edition using a stack of tasks. – *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, pp. 224–230, Bled, Slovenia, October 2011.

[Ramos 12] O. Ramos, N. Mansard, O. Stasse, P. Soueres. – Walking on non-planar surfaces using an inverse dynamic stack of tasks. – *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, Osaka, Japan, November 2012.

[Ramos 13] O. Ramos, N. Mansard, O. Stasse, P. Souères. – An advanced robotics motion generation framework for inferring the organisation of human movements. *Computer methods in biomechanics and biomedical engineering*, 16(sup1):177–178, 2013.

[Ramos 14a] O. Ramos, M. Garcia, N. Mansard, O. Stasse, J-B Hayet, P. Souères. – Towards reactive vision-guided walking on rough terrain: An inverse-dynamics based approach. *International Journal of Humanoid Robotics (IJHR)*, 11(2), 2014.

[Ramos 14b] O. Ramos, N. Mansard, P. Souères. – Whole-body motion integrating the capture point in the operational space inverse dynamics control. – *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, Madrid, Spain, November 2014.

[Ramos 15] O. Ramos, N. Mansard, O. Stasse, S. Hak, L. Saab, C. Benazeth. – Dynamic whole body motion generation for the dance of a humanoid robot. *IEEE Robotics & Automation Magazine (RAM)*, 2015. – in press.

[Raunhardt 07] D. Raunhardt, R. Boulic. – Progressive clamping. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, Roma, Italy, 2007.

[Rebula 07] J. Rebula, F. Canas, J. Pratt, A. Goswami. – Learning capture points for humanoid push recovery. – *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2007.

[Richalet 78] J. Richalet, A. Rault, J. Testud, J. Papon. – Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14(5):413–428, 1978.

[Righetti 11a] L. Righetti, J. Buchli, M. Mistry, S. Schaal. – Control of legged robots with optimal distribution of contact forces. – *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2011.

[Righetti 11b] L. Righetti, J. Buchli, M. Mistry, S. Schaal. – Inverse dynamics control of floating-base robots with external constraints: A unified view. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2011.

[Roberts 88] K.S. Roberts. – A new representation for a line. – *Computer Society Conference on Computer Vision and Pattern Recognition*, 1988.

[Rusu 11] R.B. Rusu, S. Cousins. – 3d is here: Point cloud library (PCL). – *IEEE International Confenrence on Robotics and Automation (ICRA)*, Shanghai, China, 2011.

[Saab 11] L. Saab, O. Ramos, N. Mansard, P. Souères, J-Y. Fourquet. – Generic dynamic motion generation with multiple unilateral constraints. – *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, USA, September 2011.

[Saab 13] L. Saab, O. Ramos, N. Mansard, P. Souères, J-Y. Fourquet. – Dynamic whole-body motion generation under rigid contacts and other unilateral constraints. *IEEE Transactions on Robotics (TRO)*, 29(2):346–362, April 2013.

[Salini 09] J. Salini, S. Barthelemy, P. Bidaud. – Lqp controller design for generic whole body motion. *Climbing and Walking Robots and the Support Technologies for Mobile Machines*, 2009.

[Salini 11] J. Salini, V. Padois, P. Bidaud. – Synthesis of complex humanoid whole-body behavior: a focus on sequencing and tasks transitions. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2011.

[Samson 91] C. Samson, B. Espiau, M. Le Borgne. – *Robot control: the task function approach.* – Oxford University Press, 1991.

[Sardain 04] P. Sardain, G. Bessonnet. – Forces acting on a biped robot. center of pressure-zero moment point. *IEEE Transactions on Systems, Man and Cybernetics (I-TSMC)*, 34(5):630–637, 2004.

[Schultz 10] G. Schultz, K. Mombaur. – Modeling and optimal control of human-like running. *IEEE/ASME Transactions on Mechatronics*, 15(5):783–792, 2010.

[Sciavicco 87] L. Sciavicco, B. Siciliano. – Solving the inverse kinematic problem for robotic manipulators. *RoManSy 6*, pp. 107–114. – Springer, 1987.

[Selig 07] J.M. Selig. – *Geometric fundamentals of robotics.* – Springer, 2007.

[Sentis 05] L. Sentis, O. Khatib. – Control of free-floating humanoid robots through task prioritization. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2005.

[Shan 02] J. Shan, F. Nagashima. – Neural locomotion controller design and implementation for humanoid robot HOAP-1. – *20th Annual Conference of the Robotics Society of Japan*, 2002.

[Shibata 01] T. Shibata, T. Mitsui, K. Wada, A. Touda, T. Kumasaka, K. Tagami, K. Tanie. – Mental commit robot and its application to therapy of children. – *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2001.

[Shin 01] H. Shin, J. Lee, S. Shin, M. Gleicher. – Computer puppetry: an importance-based approach. *ACM Transactions on Graphics (SIGGRAPH)*, 20(2):67–94, 2001.

[Siciliano 91] B. Siciliano, J.J. Slotine. – A general framework for managing multiple tasks in highly redundant robotic systems. – *Fifth International Conference on Advanced Robotics (ICAR)*, 1991.

[Siciliano 09] B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo. – Robotics: Modelling, planning and control. *Advanced Textbooks in Control and Signal Processing (*, 2009.

[Slotine 87] J-J Slotine, W. Li. – On the adaptive control of robot manipulators. *The International Journal of Robotics Research (IJRR)*, 6(3):49–59, 1987.

[Soroka 12] A. Soroka, R. Qiu, A. Noyvirt, Z. Ji. – Challenges for service robots operating in non-industrial environments. – *IEEE International Conference on Industrial Informatics (INDIN)*, 2012.

[Spenko 06] M. Spenko, H. Yu, S. Dubowsky. – Robotic personal aids for mobility and monitoring for the elderly. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(3):344–351, 2006.

[Spong 92] M.W. Spong. – Remarks on robot dynamics: canonical transformations and riemannian geometry. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 1992.

[Spong 06] M.W. Spong, S. Hutchinson, M. Vidyasagar. – *Robot modeling and control*. – John Wiley & Sons, 2006.

[Stentz 15] T. Stentz, H. Herman, A. Kelly, E. Meyhofer, G.C. Haynes, D. Stager, B. Zajac, J.A. Bagnell, J. Brindza, C. Dellin, M. George, J. Gonzalez-Mora, S. Hyde, M. Jones, M. Laverne, M. Likhachev, L. Lister, M. Powers, O. Ramos, J. Ray, D. Rice, J. Scheifflee, R. Sidki, S. Srinivasa, K. Strabala, J-P Tardif, J-S Valois, J-M Vande-Weghe, M. Wagner, C. Wellington. – Chimp, the cmu highly intelligent mobile platform. *Journal of Field Robotics*, 2015. – in press.

[Sugano 87] S. Sugano, I. Kato. – WABOT-2: Autonomous robot with dexterous finger-arm–Finger-arm coordination control in keyboard performance. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 1987.

[Sussmann 97] H. Sussmann, J. Willems. – 300 years of optimal control: from the brachystochrone to the maximum principle. *IEEE Control Systems*, 17(3):32–44, 1997.

[Tassa 12] Y. Tassa, T. Erez, E. Todorov. – Synthesis and stabilization of complex behaviors through online trajectory optimization. – *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[Tassa 14] Y. Tassa, N. Mansard, E. Todorov. – Control-limited differential dynamic programming. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2014.

[Tedrake 10] R. Tedrake, I. Manchester, M. Tobenkin, J. Roberts. – Lqr-trees: Feedback motion planning via sums-of-squares verification. *The International Journal of Robotics Research (IJRR)*, 2010.

[Tellez 08] R. Tellez, F. Ferro, S. Garcia, E. Gomez, E. Jorge, D. Mora, D. Pinyol, J. Oliver, O. Torres, J. Velazquez. – Reem-B: An autonomous lightweight human-size humanoid robot. – *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2008.

[Tevatia 00] G. Tevatia, S. Schaal. – Inverse kinematics for humanoid robots. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2000.

[Todorov 12] E. Todorov, T. Erez, Y. Tassa. – Mujoco: A physics engine for model-based control. – *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[Tomei 91] P. Tomei. – Adaptive pd controller for robot manipulators. *IEEE Transactions on Robotics and Automation*, 7(4):565–570, 1991.

[Trinkle 97] J. Trinkle, J-S Pang, S. Sudarsky, G. Lo. – On dynamic multi-rigid-body contact problems with coulomb friction. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 77(4):267–279, 1997.

[Tsagarakis 13] N. Tsagarakis, S. Morfey, Medrano C., L. Zhibin, D. Caldwell. – Compliant humanoid coman: Optimal joint stiffness tuning for modal frequency control. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2013.

[Tsai 99] L.W. Tsai. – *Robot analysis: the mechanics of serial and parallel manipulators.* – John Wiley & Sons, 1999.

[Tzu 11] Lieh Tzu. – *The book of Master Lie.* – Translated by T. Clearly, August 2011.

[Uicker 67] JJ Uicker. – Dynamic force analysis of spatial linkages. *Journal of applied mechanics*, 34(2):418–424, 1967.

[Vukobratovic 69] M. Vukobratovic, D. Juricic. – Contribution to the synthesis of biped gait. *IEEE Transactions on Biomedical Engineering*, pp. 1–6, 1969.

[Vukobratović 04] M. Vukobratović, B. Borovac. – Zero-moment point-thirty five years of its life. *International Journal of Humanoid Robotics (IJHR)*, 1(01):157–173, 2004.

[Wampler 86] C.W. Wampler. – Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Transactions on Systems, Man and Cybernetics*, 16(1):93 – 101, 1986.

[Webb 13] D. Webb, J. van den Berg. – Kinodynamic rrt*: Asymptotically optimal motion planning for robots with linear dynamics. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2013.

[Wensing 13] P. Wensing, D. Orin. – Generation of dynamic humanoid behaviors through task-space control with conic optimization. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2013.

[Whitman 13] E. Whitman. – *Coordination of Multiple Dynamic Programming Policies for Control of Bipedal Walking.* – PhD. Thesis, Carnegie Mellon University, 2013.

[Whitney 69] D.E. Whitney. – Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on man-machine systems*, 1969.

[Wieber 02] PB. Wieber. – On the stability of walking systems. – *International workshop on humanoid and human friendly robotics*, 2002.

[Wieber 06a] PB. Wieber. – Holonomy and nonholonomy in the dynamics of articulated motion. *Fast motions in biomechanics and robotics*, pp. 411–425. – Springer, 2006.

[Wieber 06b] PB. Wieber. – Trajectory free linear model predictive control for stable walking in the presence of strong perturbations. – *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2006.

[Wiener 49] N. Wiener. – Extrapolation, interpolation and smoothing of stationary time series. *Technology Press*, 1949.

[Wimbock 07] T. Wimbock, C. Ott, G. Hirzinger. – Impedance behaviors for two-handed manipulation: Design and experiments. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, 2007.

[Xia 09] Z. Xia, G. Chen, J. Xiong, Q. Zhao, K. Chen. – A random sampling-based approach to goal-directed footstep planning for humanoid robots. – *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2009.

[Yamane 03] K. Yamane, Y. Nakamura. – Dynamics filter-concept and implementation of online motion generator for human figures. *IEEE Transactions on Robotics and Automation*, 19(3):421–432, 2003.

[Yanai 11] H. Yanai, K. Takeuchi, Y. Takane. – *Projection Matrices.* – Springer, 2011.

[Yoshida 06] E. Yoshida, O. Kanoun, C. Esteves, J.P. Laumond. – Task-driven support polygon reshaping for humanoids. – *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2006.

[Yoshida 08] E. Yoshida, C. Esteves, I. Belousov, J.-P. Laumond, T. Sakaguchi, K. Yokoi. – Planning 3-d collision-free dynamic robotic motion through iterative reshaping. *IEEE Transactions on Robotics (TRO)*, 24(5):1186 –1198, 2008.

[You 05] B. You, Y. Choi, M. Jeong, D. Kim, Y. Oh, C. Kim, J. Cho, M. Park, S. Oh. – Network-based humanoids mahru and ahra. – *International Conference on Ubiquitous Robots and Ambient Intelligence*, 2005.

[Zhang 09] L. Zhang, J. Pia, D. Manocha. – Motion planning of human-like robots using constrained coordination. – *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2009.

[Zhang 14] Y. Zhang, J. Luo, K. Hauser, A. Park, M. Paldhe, G. Lee, R. Ellenberg, B. Killen, P. Oh, J. Oh, J. Lee, I. Kim. – Motion planning and control of ladder climbing on drc-hubo for darpa robotics challenge. – *IEEE International Confenrence on Robotics and Automation (ICRA)*, Hong Kong, China, 2014.

[Zoss 06] A. Zoss, H. Kazerooni, A. Chu. – Biomechanical design of the berkeley lower extremity exoskeleton (bleex). *IEEE/ASME Transactions on Mechatronics*, 11(2):128–138, 2006.

[Zucker 13] M. Zucker, N. Ratliff, A. Dragan, M. Pivtoraiko, M. Klingensmith, C. Dellin, J. Bagnell, S. Srinivasa. – Chomp: Covariant hamiltonian optimization for motion planning. *The International Journal of Robotics Research (IJRR)*, 32(9-10):1164–1193, 2013.

Abstract

This thesis aims at providing a solution to the problem of motion generation for humanoid robots. The proposed framework generates whole-body motion using the complete robot dynamics in the task space satisfying contact constraints. This approach is known as operational-space inverse-dynamics control. The specification of the movements is done through objectives in the task space, and the high redundancy of the system is handled with a prioritized stack of tasks where lower priority tasks are only achieved if they do not interfere with higher priority ones. To this end, a hierarchical quadratic program is used, with the advantage of being able to specify tasks as equalities or inequalities at any level of the hierarchy. Motions where the robot sits down in an armchair and climbs a ladder show the capability to handle multiple non-coplanar contacts.

The generic motion generation framework is then applied to some case studies using HRP-2 and Romeo. Complex and human-like movements are achieved using human motion imitation where the acquired motion passes through a kinematic and then dynamic retargeting processes. To deal with the instantaneous nature of inverse dynamics, a walking pattern generator is used as an input for the stack of tasks which makes a local correction of the feet position based on the contact points allowing to walk on non-planar surfaces. Visual feedback is also introduced to aid in the walking process. Alternatively, for a fast balance recovery, the capture point is introduced in the framework as a task and it is controlled within a desired region of space. Also, motion generation is presented for CHIMP which is a robot that needs a particular treatment.

Keywords: Humanoid robotics, whole-body motion, inverse dynamics, task space, motion imitation, dynamic walking

Résumé

Cette thèse propose une solution au problème de la génération de mouvements pour les robots humanoïdes. Le cadre qui est proposé dans cette thèse génère des mouvements corps-complet en utilisant la dynamique inverse avec l'espace des tâches et en satisfaisant toutes les contraintes de contact. La spécification des mouvements se fait à travers objectifs dans l'espace des tâches et la grande redondance du système est gérée avec une pile de tâches où les tâches moins prioritaires sont atteintes seulement si elles n'interfèrent pas avec celles de plus haute priorité. À cette fin, un QP hiérarchique est utilisé, avec l'avantage d'être en mesure de préciser tâches d'égalité ou d'inégalité à tous les niveaux de la hiérarchie. La capacité de traiter plusieurs contacts non-coplanaires est montrée par des mouvements où le robot s'assoit sur une chaise et monte une échelle.

Le cadre générique de génération de mouvements est ensuite appliqué à des études de cas à l'aide de HRP-2 et Romeo. Les mouvements complexes et similaires à l'humain sont obtenus en utilisant l'imitation du mouvement humain où le mouvement acquis passe par un processus cinématique et dynamique. Pour faire face à la nature instantanée de la dynamique inverse, un générateur de cycle de marche est utilisé comme entrée pour la pile de tâches qui effectue une correction locale de la position des pieds sur la base des points de contact permettant de marcher sur un terrain accidenté. La vision stéréo est également introduite pour aider dans le processus de marche. Pour une récupération rapide d'équilibre, le capture point est utilisé comme une tâche contrôlée dans une région désirée de l'espace. En outre, la génération de mouvements est présentée pour CHIMP, qui a besoin d'un traitement particulier.

Mots-clefs : Robotique humanoïde, mouvement corps-complet, dynamique inverse, espace des tâches, imitation du mouvement, marche dynamique