Universidad Carlos III de Madrid

Departament of Systems Engineering and Automation

Ph.D. Thesis

Humanoid Robot Control of Complex Postural Tasks based on Learning from Demonstration

Miguel González-Fierro

Advisers:

Prof. Carlos Balaguer
Universidad Carlos III

Dr. Thrishantha Nanayakkara
King's College London

Leganés, October 2014

Ph.D. Thesis

Humanoid Robot Control of Complex Postural Tasks based on Learning from Demonstration

Candidate:
Miguel González-Fierro

Advisers:
Prof. Carlos Balaguer
(Universidad Carlos III de Madrid)
Dr. Thrishantha Nanayakkara
(King's College London)

Review Committee

Signature

Chair: Dr. Manuel Armada _____

Member: Dr. Sylvain Calinon _____

Secretary: Dr. Santiago Garrido _____

Degree: Doctorado en Ingeniería Eléctrica, Electrónica y Automática
Grade: _____

Leganés, 7 October 2014

A mi mujer y a mi familia

"Well, I feel bad to lose you, but we did some nice work when we were together. Always remember that we are a lucky lot. We explore nature looking for her secret laws that govern our lives. [...] Our thirst is to uncover the secrets. That is exciting. Each finding we make changes the way people feel about their world. You must continue that. The human brain is an amazing thing. [...] Our brains are designed to think. They are thinking machines. The more you imagine, the more you enjoy. Never be followers. They are next to dead bodies. All I want you to do is to continue challenging life."

— Thrishantha Nanayakkara in *Devi*

Acknowledgements

I started a Ph.D. because when I finished my degree I felt that I had learned very few things about Engineering. It is funny that now that I finished the thesis, I feel kind of the same way. Even worse, the more I study, the more I understand that I have many lessons to learn. I realised that there are two kinds of people, those who don't know and those who don't know they don't know. Those who study all their life and those who think they know everything. When you are a scientist, you surely are one of the students.

I am a privileged person, I am very lucky. I was granted with the opportunity to put a brick in the wall of science. I stood on the shoulders of giants as many people did before me. It is even more exciting to participate in the growing of robotics, a very young science in comparison with others like physics or mathematics. In the next decades there will be robots collaborating with us and this future will have a piece of my work and effort.

I would like to express my deepest gratitude to everyone that have accompanied me in all this years of learning. Therefore, I would like to propose a toast:

To Carlos Balaguer. For his leadership and kindness. For giving me the opportunity to learn at his side. For all the time he has given me, which is more valuable knowing he does not have much. And for his guidance and help in all the decisions I have made.

To Thrishantha Nanayakkara. For accepting me as his Ph.D student and giving me the opportunity of being part of his research group in London. For everything that I have learned from him and all the things I will learn. For his wisdom and knowledge. And for being so close even though we were so far.

To Luis Moreno and Santiago Garrido. For being my Ph.D. advisors in the shadows. For all that have learned from them and for answering all my questions.

To Juan González Víctores, Daniel Hernández, Martin Stoelen and Víctor González Pacheco. For all the ideas and experiences we have shared.

To Jorge García Bueno and Alejandro Martín. For being my partners in the difficult adventure of creating a company.

To Concha Monje, Dolores Blanco, Paolo Pierro, Alberto Jardón, Javier Gorostiza, Ramón Barber and Santiago Martínez de la Casa. For helping me every time I have needed them and for all the funny moments we have lived together.

To Raúl Perula, Juanmi García Haro, Santi Morante, Javier Quijano and every member of RoboticsLab. For all the jokes and the good experiences.

To Sonia Mata and Eduardo Silles. For solving many problems and being always ready to help.

To Juan Domínguez. For being the one to encourage me to study Industrial Engineer.

To David López del Moral, Mª Jesús Gómez and Eduardo Corral. For being as crazy as I am to do a Ph.D. and for all the afternoons we have spent together.

To Miguel Maldonado. For being my brother. For supporting me, working with me and helping me. And for being at my side so many time.

To my father, sister, grandparents and the rest of my family. For their support and love.

To my mother. For being the best mother anyone can desire. For his wisdom and dedication.

To my wife. For being everything to me.

Gracias a todos

Miguel

Abstract

This thesis addresses the problem of planning and controlling complex tasks in a humanoid robot from a postural point of view. It is motivated by the growth of robotics in our current society, where simple robots are being integrated. Its objective is to make an advancement in the development of complex behaviors in humanoid robots, in order to allow them to share our environment in the future.

The work presents different contributions in the areas of humanoid robot postural control, behavior planning, non-linear control, learning from demonstration and reinforcement learning. First, as an introduction of the thesis, a group of methods and mathematical formulations are presented, describing concepts such as humanoid robot modelling, generation of locomotion trajectories and generation of whole-body trajectories.

Next, the process of human learning is studied in order to develop a novel method of postural task transference between a human and a robot. It uses the demonstrated action goal as a metrics of comparison, which is codified using the reward associated to the task execution.

As an evolution of the previous study, this process is generalized to a set of sequential behaviors, which are executed by the robot based on human demonstrations.

Afterwards, the execution of postural movements using a robust control approach is proposed. This method allows to control the desired trajectory even with mismatches in the robot model.

Finally, an architecture that encompasses all methods of postural planning and control is presented. It is complemented by an environment recognition module that identifies the free space in order to perform path planning and generate safe movements for the robot.

The experimental justification of this thesis was developed using the humanoid robot HOAP-3. Tasks such as walking, standing up from a chair, dancing or opening a door have been implemented using the techniques proposed in this work.

Resumen

Esta tesis aborda el problema de la planificación y control de tareas complejas de un robot humanoide desde el punto de vista postural. Viene motivada por el auge de la robótica en la sociedad actual, donde ya se están incorporando robots sencillos y su objetivo es avanzar en el desarrollo de comportamientos complejos en robots humanoides, para que en el futuro sean capaces de compartir nuestro entorno.

El trabajo presenta diferentes contribuciones en las áreas de control postural de robots humanoides, planificación de comportamientos, control no lineal, aprendizaje por demostración y aprendizaje por refuerzo. En primer lugar se desarrollan un conjunto de métodos y formulaciones matemáticas sobre los que se sustenta la tesis, describiendo conceptos de modelado de robots humanoides, generación de trayectorias de locomoción y generación de trayectorias del cuerpo completo.

A continuación se estudia el proceso de aprendizaje humano, para desarrollar un novedoso método de transferencia de una tarea postural de un humano a un robot, usando como métrica de comparación el objetivo de la acción demostrada, que es codificada a través del refuerzo asociado a la ejecución de dicha tarea.

Como evolución del trabajo anterior, se generaliza este proceso para la realización de un conjunto de comportamientos secuenciales, que son de nuevo realizados por el robot basándose en las demostraciones de un ser humano.

Seguidamente se estudia la ejecución de movimientos posturales utilizando un método de control robusto ante imprecisiones en el modelado del robot.

Para finalizar, se presenta una arquitectura que aglutina los métodos de planificación y el control postural desarrollados en los capítulos anteriores. Esto se complementa con un módulo de reconocimiento del entorno y extracción del espacio libre para poder planificar y generar movimientos seguros en dicho entorno.

La justificación experimental de la tesis se ha desarrollado con el robot humanoide HOAP-3. En este robot se han implementado tareas como caminar, levantarse de una silla, bailar o abrir una puerta. Todo ello haciendo uso de las técnicas propuestas en este trabajo.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

This chapter deals with the initial introduction, motivation and presentation of this thesis. The thesis addresses aspects related to planning and control of complex postural tasks for humanoid robots using human learning by demonstration. The thesis proposes a specific situation that has to be solved by the robot. It is the order "stand up from a chair, walk to the door and open it" that represents a complex task which involves the development of several skills and gives rise to a broad set of approaches to solve this problem. The present work is focused on discussing and studying methods that involves the generation and execution of humanoid motions in terms of postural body transitions. These postural body transitions are optimized using an index that we called the reward profile. The reward profile is a multi-modal time-dependent function that encodes the skill goal that the robot has to perform. At the same time, it is a measurement of the skill performance in terms of different aspects like stability, softness or human likeliness. Furthermore, this thesis addresses the modelling and control of the humanoid robot and presents a wide variety of simulated and experimental results. The objective behind this work is to make humanoid robots more intelligent and autonomous, by allowing them to follow complex orders safely and precisely.

1.1 Motivation

There has been a recent surge of interest to understand how humans manage some metastable maneuvers such as walking on stochastically rough terrain, stand up from seated postures, and walk while manipulating objects with such elegance. Even though industrial robots are widely used around the world in factories or even hospitals, complex machines like humanoid robots are not currently used in the human environment due to limitations such as robustness, security, level of autonomy or level of cognition. Many years may pass before we have humanoid robots collaborating with us in our homes or in our working space. However, in recent years we are feeling a state of excitement that make us think that a humanoid robot living in our environment may soon became a reality.

It is remarkable to analyze the opinion of Bill Gates about this matter (Gates, 2007). In 2007, he published an article called *"A robot in every home"*, where he foresees the emergence of the robotic industry in the following years. He compares the present situation with the revolution of the computer industry that took place in the mid-1970s, where researchers around the world were creating the basic components that were used to construct the first computers and some tech companies like Intel, Atari or Microsoft (and some others like IBM and Apple) were making products for the young personal computer market. The level of excitement is similar to what happened 30 years ago. There are some amazing advances happening in robotics that have the potential to significantly improve the current relationship between humans and machines.

One of the most interesting events that is happening right now is the DARPA Robotic Challenge[1]. It is a challenge where research groups and spinoffs at top Universities compete to be able to create a humanoid robot that have to pass a set of trials. The nuclear disaster of Fukushima in 2011 made people around the world realize the importance of making robots capable of assisting us in natural and man-made disasters. With that in mind, DARPA created a competition with the objective to boost the advancement and development of hardware, software, sensors and human-machine interfaces and complete a series of tasks, selected by them for their relevance in preventing another similar disaster.

The challenge consists on three phases:

- The Virtual Robotics Challenge occurred in June 2013 and tested software teams' ability to effectively guide a simulated robot through three sample tasks in a virtual environment. Here teams of Universidad Carlos III and King's College London were selected to participate, even though they did a good job, they did not pass to the next phase.

- The DRC Trials occurred in December 2013, where teams guided their robots

[1]More information at www.theroboticschallenge.org

Figure 1.1: *SCHAFT team robot, winner of the second phase of the DARPA Robotic Challenge in December 2013.*

through eight individual, physical tasks that tested mobility, manipulation, dexterity, perception and operator control mechanisms. The winner was SCHAFT team that obtained 27 points over 32, successfully completing almost all tasks. The robot constructed by SCHAFT team can be seen in Figure 1.1.

- The DRC Finals (occurring before mid 2015) will require robots to attempt a circuit of consecutive physical tasks, with degraded communications between the robots and their operators. The winning team will receive a 2 million dollars prize.

Another important event in the field of robotics took place in 2013. Google bought 8 robotics companies in just 6 months. Schaft, which was the winner of the second phase of DARPA Challenge; Industrial Perceptions, experts on 3D vision robotic guidance; Redwood Robotics and Meka Robotics, both robotic manufacturers; Holomini, that designs omnidirectional wheels; Bot & Dolly, an artistic robotic design studio; Boston Dynamics, the creators of some of the most advanced robots to date, like the Big Dog, PETMAN or Atlas and finally, Deep Mind, a machine

learning company. It is not surprising that Google, a company founded by two scientists, realized the importance of robotics. Other tech companies like Facebook, Apple or Amazon are starting to follow Google's example by acquiring companies that have a deep relationship with robotics.

With all this happening right now it is very exciting to be part of the robotics scientific community. The future development of robotics strongly depends on the discoveries and contributions that we make today. In particular, this thesis contributes in fields like control, machine learning, modeling and behavior planning of humanoid robots.

We are living in exponential times and it is fortunate to be able to add a brick to the most amazing building the humanity has ever made: science.

1.2 Objectives and Approach

The main objective of this thesis is to make a humanoid robot capable of learning and successfully executing complex skills based on human demonstrations. As a framework of development, there is a high level order that the robot must execute, which is *"stand up from a chair, walk to the door and open it"* that constitutes the screenplay of the experiments.

The problem is approached from the point of view of the postural body transitions that the robot has to acquire to learn the selected skills. It is also important to be able to measure the level of performance of the humanoid postural trajectory. It has to be measured in terms of effort, stability, internal robot constraints and similarity with the human behavior.

Therefore, there are three consecutive objectives that are pursued in this thesis:

1. Postural planning and control of a single skill based on human demonstrations.

2. Postural planning and control of a series of consecutive skills based on human demonstrations.

3. Execution of the high level order *"stand up from a chair, walk to the door and open it"* in a cluttered environment.

The objectives previously mentioned are experimentally validated through its implementation in the real humanoid robot HOAP-3 (see Appendix A) and discussed in the following chapters.

1.3 Document Organization

The document is ordered as follows:

Chapter 2 introduces basic concepts and tools applied during the thesis development. First, there is an overview of different humanoid models, second, there is a brief study of locomotion methods for humanoid robots and finally, there is an explanation of a method of full body trajectory mimicking.

Chapter 3 presents a methodology for imitation and innovation of a single humanoid skill based on human demonstrations. The main characteristic of the approach is the application of a multi-modal reward profile as a common basis of comparison between the human and the robot. The reward profile represents a temporal measurement of the good or bad robot performance in terms of postural motion. Defining a Markov Transition Matrix of the reward, the method encodes in one single metrics the postural strategy of the human, which can be transferred to the robot to imitate the showed skill. In addition, the robot can search in the neighborhood of the solution space for an improved behavior, which innovates a new skill that fits better with the robot structure and constraints. Experiments with the real robot are presented performing the skill of standing up from a chair.

Chapter 4 discusses a generalization of the method presented in the previous chapter for linking a set of sequential skills executed by the robot. In this method there is a local optimization of every individual skill to match the reward profile of the human demonstrator, and in addition, there is also a global optimization that includes all sequential skills together. The experiment developed in this chapter is a real humanoid robot that walks to a door and opens it.

Chapter 5 discusses the advantages and disadvantages of using concentrated mass models or distributed models to represent a humanoid robot. It proposes an alternative method to model and control humanoid robots based on simple concentrated mass models and fractional order controllers. The method has the advantage of having a simple mathematical formulation, while at the same time, is able to cope with model imprecisions and disturbances.

Chapter 6 presents the complete architecture of the thesis. It consists on four modules. First, the definition of a high level order like *"stand up from a chair, walk to the door and open it"* that the robot has to perform. Second, an environment analysis, distinguishing between free space and obstacles. The third module is the postural planning, which is an offline planning of the postural motion sequence needed to perform the order. Finally, the last module is an online postural control which computes the control orders to follow the desired postural trajectory.

Chapter 7 contains the concluding remarks, key contributions and recommendations for future improvements of this work.

Chapter 2

Basic Representations for Postural Control in Humanoids

This chapter deals with some basic concepts that are repeatably used in this thesis. There are some basic tasks that have to be done in advance in order to conduct many of the experiments. They are related to modeling, kinematics, control and dynamics. There are many aspects of the humanoid set up that are not explicitly described in the thesis. This chapter acts as the basis of some of the algorithms that are explained in the next chapters. First, a study of different humanoid robot models is presented. It includes simple models to represent the robot, like the inverted pendulum, and complex models like the mass distributed model. The equation of motion of each of them is obtained. These models are included in some motion generation algorithms and balance maintenance methods. Afterwards, the most famous stability criterion, the Zero Moment Point (ZMP), is explained in detail. Furthermore, two common methods of biped locomotion generation are studied, the 3D Linear Inverted Pendulum Model and the Cart Table model. The latter was used to generate stable biped locomotion in the real humanoid. Finally, a simple method of whole body imitation is presented and a dance performance is obtained as an experiment.

2.1 Introduction

It has been one of the dreams of human kind to create an artificial partner in our image and our likeness. And it is the duty of robotic researchers to make this dream came true.

The history of humanoid robotics starts in 1973 at Waseda University, Japan, with the creation of the first full size humanoid robot by the team of Professor Ichiro Kato, called WABOT-1 (Kato, 1973, 1974). The robot integrated different capabilities as visual object recognition, speech generation, speech recognition, bimanual object manipulation and bipedal walking. This pioneer development was followed by the WABOT-2 (Ogura et al., 2006), whose ability to play the piano drew significant public interest in 1984. Further on, different improvements resulted in the series of humanoids WABIAN.

In 1986, Honda began the ASIMO project (Hirai, 1997; Hirai et al., 1998) with the thought of creating a machine "to coexist and collaborate with a human, to perform things that a human is unable to do and to create a mobility which brings additional value to human society". In 1996, they presented the Honda Humanoid P2. It was the first humanoid to be able to perform stable bipedal walking with onboard power and processing. In 2000, the company unveiled ASIMO (Sakagami et al., 2002; Hirose and Takenaka, 2001), which could climb stairs and recognize faces. Further on, in 2005, they presented the new ASIMO, which could run up to $6Km/h$ and is today one of the most advanced humanoids in the world.

In 1998, the AIST (Advance Institute of Science and Technology) and Kawada Industries, with the economic support of the Japanese government, started the Humanoid Robot Project. It led to the development of the HRP-2L, HRP-2P, HRP-2, HRP-3P, HRP-3, HRP-4C and HRP-4 humanoid robots (Kaneko et al., 2002, 2004; Akachi et al., 2005; Kaneko et al., 2008, 2009, 2011). The robot can walk (Kajita et al., 2003a; Hirukawa et al., 2006; Arbulu et al., 2008; Bouyarmane and Kheddar, 2012), lie down and stand up (Hirukawa et al., 2005), dance (Nakaoka et al., 2005) and collaborate with humans (Harada et al., 2007).

In the early 21^{th} century, many other companies and research groups are following the trend to build full size humanoid robots. Some well known humanoids are HUBO from KAIST (Oh et al., 2006), Johnnie from TUM (Löffler et al., 2003), Robonaut from NASA (Diftler et al., 2011) or PETMAN and Atlas from Boston Dynamics (Nelson et al., 2012).

In the last ten years, the Robotics Lab research group from Universidad Carlos III de Madrid has leaded the humanoid robot development in Spain, under the direction of Prof. Balaguer. It begun with the first prototype named RH-0, who was capable of stable walking (Arbulú et al., 2005; Kaynov et al., 2006). The next prototype, RH-1, incorporated new features like voice and gesture recognition (Staroverov et al., 2007a,b) and new gait generation methods (Arbulú et al., 2009; Arbulú and Balaguer, 2009; Monje et al., 2009b; Pierro et al., 2008; Kaynov et al.,

2009a). The new prototype, TEO (Task Environment Operator), was designed carrying through the lessons learned from the previous robots and incorporates new advances in mechatronics and control techniques (Monje et al., 2011a; Martínez et al., 2012).

Equation of motion

The first step towards controlling a humanoid robot is obtaining its dynamic model. The dynamic model of a robot is used to know the relationship between the robot motion and the forces involved in this motion (Barrientos et al., 2007). Obtaining the dynamic model of the robot is essential to achieve different purposes as movement simulation, design of the mechanical structure and actuators or movement control. The movement of the robot is defined by its equation of motion.

$$\tau = H(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) + G(\mathbf{q}) \tag{2.1}$$

where H is the inertia matrix, C is the coriolis and centripetal matrix and G is the gravity matrix. To be more precise, these matrices does not only depend on $\mathbf{q}$ and $\dot{\mathbf{q}}$, but also on the model, so it would be more correct to write these matrices as $H(model, \mathbf{q})$, $C(model, \mathbf{q}, \dot{\mathbf{q}})$ and $G(model, \mathbf{q})$ where *model* refers to the rigid body system including the number of bodies and joints, the kind of joint (prismatic, rotational or spherical), masses, inertias and the way they are connected. $\mathbf{q}$, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ are the generalized joint position, velocity and acceleration vectors and τ is the generalized joint torque. It has to be remarked that the generalized torque vector should include all torques that intervene in the movement, i.e.:

$$\tau = \tau_m - \tau_p - \tau_f \tag{2.2}$$

where τ_m is the actuated motor torque, τ_p is the perturbation torque and τ_f is the friction torque.

The objective of the robot dynamics is to find the values of joint acceleration $\ddot{\mathbf{q}}$ given the joint torques τ, which is named forward dynamics, or to find the values of joint torques τ, given the accelerations $\ddot{\mathbf{q}}$, which is named inverse dynamics.

$$\ddot{\mathbf{q}} = FD(model, \mathbf{q}, \dot{\mathbf{q}}, \tau)$$

$$\tau = ID(model, \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$$

Dynamic formulation and algorithms

There are classically two approaches to solve the dynamics of a rigid body system, the Lagrangian formulation and the Newton-Euler formulation. Some classical reference books are (Featherstone, 1987; Hollerbach et al., 1982) or more recently (Sciavicco et al., 2009).

Lagrangian approach solves the dynamics of a rigid body system tacking into account the energy of the system. This formulation allows to derive the equation of motion in a systematic way, independently of the reference frame. It requires the generalized positions $\mathbf{q}$ and the generalized forces τ.

The Lagrangian equation or Lagrangian $\mathcal{L}$ depends on the generalized position, generalized velocities and time and is the difference between the kinetic $\mathcal{T}$ and potential energy $\mathcal{V}$.

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = \mathcal{T}(\mathbf{q}, \dot{\mathbf{q}}) - \mathcal{V}(\mathbf{q}) \tag{2.3}$$

The equation of motion using the Lagrange theory is enunciated as:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = \tau \tag{2.4}$$

The algorithms based on this approach are slower than those based on Newton-Euler Formulation. They have the advantage that they only need to compute the kinetic and potential energy, so they reduce the number of equations to derive and are less prone to errors. Some classical examples of algorithms based on the Lagrange formulation can be found in (Uicker, 1967; Kahn and Roth, 1971; Hollerbach, 1980; Book, 1984).

Newton-Euler formulation is based on the balance of all the forces acting on the robot links. This implies that the equations can be expressed in a recursive way, which produces a big advantage, the algorithms based on this formulation are faster than non recursive ones. Newton-Euler method is described by two equations, the first one is related to the translational movement of the center of mass.

$$f_i - f_{i+1} = m_i \ddot{r}_{CM} - m_i g \tag{2.5}$$

where f is the force passing through the link, $\ddot{r}_{CM}$ is the center of mass acceleration, m is the link mass and g is the gravity acceleration. The second equation is based on the rotative movement of the link.

$$\tau_i - \tau_{i+1} = I_i \alpha_i + \omega_i \times (I_i \omega_i) \tag{2.6}$$

where τ is the torque produced by the link, I is the inertia tensor of the link, α is the angular acceleration and ω the angular velocity.

Some algorithms based on the Newton-Euler formulation are presented in (Stepanenko and Vukobratovic, 1976; Orin et al., 1979; Luh et al., 1980; Featherstone, 1999a,b).

2.2 Humanoid Robot Models

Researchers proposed a wide variety of kinematic and dynamic models to represent humanoid robots. They range from concentrated models, where all the humanoid

mass is concentrated in the center of gravity, to distributed models, where the masses and inertias of every link are described as close as possible.

Mass concentrated models assume that the whole body dynamics of the robot is concentrated on its center of mass or on different points. These models are usually task oriented, then a specific model is designed for a specific task. Their main advantage is the reduction of complexity from a mathematical point of view and the reduction of computation time. They usually rely on feedback control and movement generation can be computed online. Experimentally, it has been proved that it is not necessary to model the robot as a complex dynamic system to generate bipedal walking (Kajita et al., 2003a,b; Komura et al., 2005; Arbulú and Balaguer, 2009; González-Fierro et al., 2013b) or other tasks (Kaynov et al., 2009b; González-Fierro et al., 2013a).

Many examples of concentrated models can be found in the literature. The most famous for humanoid locomotion are the inverted pendulum model (Kajita and Tani, 1991; Kajita et al., 2001) and the cart-table model (Kajita et al., 2003a).

On the other hand there are the distributed models. These models assume that all link masses, lengths and inertia tensors, as well as kinematic and dynamic transformations of each joint, are known. The computation of the dynamic parameters are a key feature of theses models, which rely on identification procedures more than feedback control (Ayusawa et al., 2008; Mistry et al., 2009; Arbulú et al., 2010). Their mathematical complexity and the high number of operations they incorporate make movement generation to be usually computed off-line.

It is important to highlight that a complete dynamic model of a robot should include not only the dynamics of its elements, like links or joints, but also the dynamics of transmission system, actuators and electronic drivers. These elements add new inertias, frictions and design errors to the model, which increments its complexity. Besides, even though in most robotic applications loads and inertias are not high enough to generate deformations, in some cases it is not like that, and it is precise to consider the robot as a non rigid group of links.

Many researchers prefer to use distributed models to represent a humanoid robot. Hirukawa et al. (2006, 2007) generate whole-body motions taking into account a strong stability criteria called Contact Wrench Cone (CWC). It implies that the sum of all wrenches should be inside the CWC. Another known approach is based on Khatib's operational space (Khatib, 1987). These methods compute whole-body motions by hierarchically select between operational space tasks (Sentis and Khatib, 2004, 2005; Khatib et al., 2008; Sentis et al., 2010). Other examples of distributed models are (Nagasaka et al., 1999; Yamane and Nakamura, 2003).

2.2.1 Inverted Pendulum Model

A well known approach to represent a humanoid robot is the inverted pendulum model. Some examples are the 2D inverted pendulum model (Kajita and Tani,

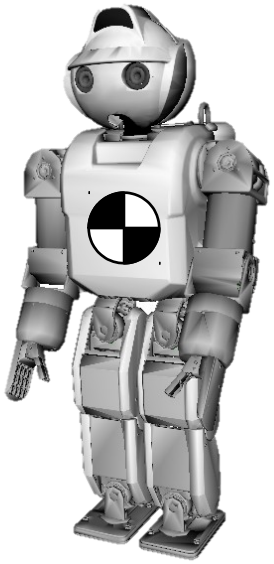


Figure 2.1: *Snapshot of a concentrated mass model of the humanoid HOAP-3.*

1991) or the 3D inverted pendulum mode (Kajita et al., 2001).

Let the position of the mass $p = (x, y, z)$ be uniquely defined by $q = (\theta_r, \theta_p, r)$, then:

$$x = r \sin \theta_p \tag{2.7}$$

$$y = -r \sin \theta_r \tag{2.8}$$

$$z = r\sqrt{1 - \sin^2 \theta_r - \sin^2 \theta_p} \tag{2.9}$$

The equation of motion of is given by:

$$\begin{pmatrix} \tau_r \\ \tau_p \\ f \end{pmatrix} = m \begin{pmatrix} 0 & -r\cos\theta_r & -\frac{r\cos\theta_r \sin\theta_r}{\sqrt{1-\sin^2\theta_r-\sin^2\theta_p}} \\ r\cos\theta_p & 0 & -\frac{r\cos\theta_p \sin\theta_p}{\sqrt{1-\sin^2\theta_r-\sin^2\theta_p}} \\ \sin\theta_p & -\sin\theta_r & \sqrt{1-\sin^2\theta_r-\sin^2\theta_p} \end{pmatrix} \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} + \\ + mg \begin{pmatrix} -\frac{r\cos\theta_r \sin\theta_r}{\sqrt{1-\sin^2\theta_r-\sin^2\theta_p}} \\ -\frac{r\cos\theta_p \sin\theta_p}{\sqrt{1-\sin^2\theta_r-\sin^2\theta_p}} \\ \sqrt{1-\sin^2\theta_r-\sin^2\theta_p} \end{pmatrix} \tag{2.10}$$

2.2.2 Double Inverted Pendulum Model

The double inverted pendulum model can be used, among other applications, as a model for stability control (Stephens, 2007; Kaynov, 2008). The first link represents

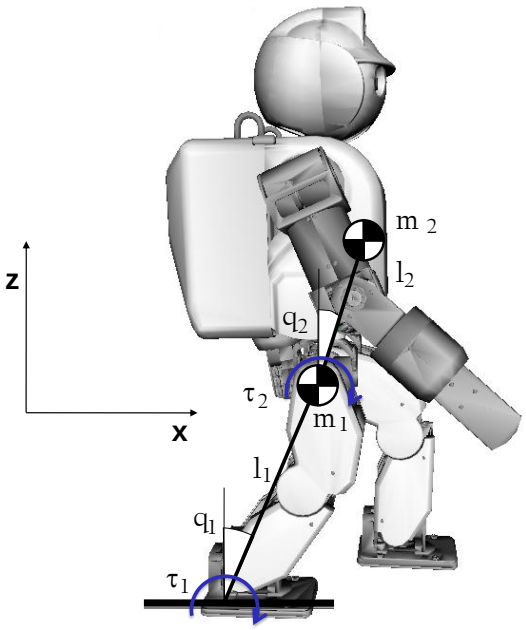


Figure 2.2: *Double inverted pendulum*

the supporting leg and the second link represents the robot chest. Using this model, the ankle and hip positions can be controlled to avoid the loss of stability, in the sagittal and frontal plane (Figure 2.2).

A double pendulum (Figure 2.2) consist in two punctual masses, m_1, located in the hip, and m_2, located in the robot center of mass (COM), whose sum is the total mass of the robot. They are joined by a massless link of length l_1 and l_2 and actuated by torques τ_1 and τ_2. Taking into account the ankle and hip angle (q_1 and q_2), the position and acceleration of the two masses can be obtained as:

$$x_1 = l_1 \sin q_1 \Rightarrow \dot{x_1} = l_1 \cos q_1 \dot{q_1} \tag{2.11}$$

$$z_1 = l_1 \cos q_1 \Rightarrow \dot{z_1} = -l_1 \sin q_1 \dot{q_1} \tag{2.12}$$

$$x_2 = l_1 \sin q_1 + l_2 \sin q_2 \Rightarrow \dot{x_2} = l_1 \cos q_1 \dot{q_1} + l_2 \cos q_2 \dot{q_2} \tag{2.13}$$

$$z_2 = l_1 \cos q_1 + l_2 \cos q_2 \Rightarrow \dot{z_2} = -l_1 \sin q_1 \dot{q_1} - l_2 \sin q_2 \dot{q_2} \tag{2.14}$$

We can obtain the equation of motion using the Lagrange theory. The Lagrangian, stated in (2.31) is composed by potential and kinetic energy

$$\mathcal{V} = m_1 g z_1 + m_2 g z_2 \tag{2.15}$$

$$\mathcal{T} = \frac{1}{2} m_1 v_1^2 + \frac{1}{2} m_2 v_2^2 \tag{2.16}$$

substituting (2.11),(2.12),(2.13),(2.14) into (2.15), (2.16) and then into (2.31) the Lagrangian is calculated:

$$L = \frac{1}{2}m_1 l_1^2 \dot{q}_1^2 + \frac{1}{2}m_2 l_2^2 \dot{q}_2^2 + m_2 l_1 l_2 \dot{q}_1 \dot{q}_2 \cos(q_1 - q_2) - \\ - (m_1 + m_2) g l_1 \cos q_1 + m_2 g l_2 \cos q_2 \quad (2.17)$$

Using (2.17) and substituting q_1 y q_2 the joint torques are obtained:

$$\tau_1 = (m_1 + m_2) l_1^2 \ddot{q}_1 + m_2 l_1 l_2 \ddot{q}_2 \cos(q_1 - q_2) + \\ + m_2 l_1 l_2 \dot{q}_2^2 \sin(q_1 - q_2) - (m_1 + m_2) g l_1 \sin q_1 \quad (2.18)$$

$$\tau_2 = m_2 l_2^2 \ddot{q}_2 + m_2 l_1 l_2 \ddot{q}_1 \cos(q_1 - q_2) - \\ - m_2 l_1 l_2 \dot{q}_1^2 \sin(q_1 - q_2) - m_2 g l_2 \sin q_2 \quad (2.19)$$

2.2.3 Triple Inverted Pendulum Model

In a very simplified way, a humanoid robot can be dynamically modeled as a triple inverted pendulum. As it can be seen in Figure 2.3(a), HOAP humanoid robot is modeled as a triple pendulum, where the ankle joint of the robot corresponds to the first pendulum joint, the knee joint corresponds to the second one, and the hip joint corresponds to the third one (see Figure 2.3(b)).

The similarity is stated under the assumptions that the pendulum masses are concentrated at the tip of every link and the link masses are negligible. The control action that allows every mass m_i to move a position q_i is the torque τ_i.

Triple Pendulum Equations

To obtain the triple pendulum equations let us define the position and velocity of every link.

$$x_1 = l_1 \sin q_1, \quad \dot{x}_1 = l_1 \cos q_1 \dot{q}_1 \quad (2.20)$$

$$z_1 = l_1 \cos q_1, \quad \dot{z}_1 = -l_1 \sin q_1 \dot{q}_1 \quad (2.21)$$

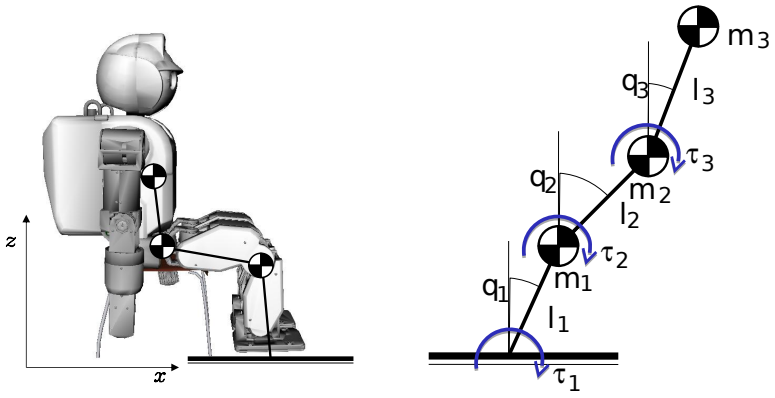
$$x_2 = l_1 \sin q_1 + l_2 \sin q_2 \quad (2.22)$$

$$\dot{x}_2 = l_1 \cos q_1 \dot{q}_1 + l_2 \cos q_2 \dot{q}_2 \quad (2.23)$$

$$z_2 = l_1 \cos q_1 + l_2 \cos q_2 \quad (2.24)$$

$$\dot{z}_2 = -l_1 \sin q_1 \dot{q}_1 - l_2 \sin q_2 \dot{q}_2 \quad (2.25)$$

$$x_3 = l_1 \sin q_1 + l_2 \sin q_2 + l_3 \sin q_3 \quad (2.26)$$



(a) (b)

Figure 2.3: *(a) Reduced model of HOAP humanoid robot seated on a chair. The proposed model is a two dimensional triple inverted pendulum with massless links and the center of mass at the tip of the pendulum. (b) Triple inverted pendulum with masses, lengths, torques and positions.*

$$\dot{x}_3 = l_1 \cos q_1 \dot{q}_1 + l_2 \cos q_2 \dot{q}_2 + l_3 \cos q_3 \dot{q}_3 \tag{2.27}$$

$$z_3 = l_1 \cos q_1 + l_2 \cos q_2 + l_3 \cos q_3 \tag{2.28}$$

$$\dot{z}_3 = -l_1 \sin q_1 \dot{q}_1 - l_2 \sin q_2 \dot{q}_2 - l_3 \sin q_3 \dot{q}_3 \tag{2.29}$$

Articulated torques can be derived using the lagrangian equation:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = \tau \tag{2.30}$$

where the Lagrangian is the difference between kinetic and potential energy.

$$\mathcal{L} = \mathcal{T} - \mathcal{V} \tag{2.31}$$

$$\mathcal{V} = m_1 g z_1 + m_2 g z_2 + m_3 g z_3 \tag{2.32}$$

$$\mathcal{T} = \frac{1}{2} m_1 v_1^2 + \frac{1}{2} m_2 v_2^2 + \frac{1}{2} m_3 v_3^2 \tag{2.33}$$

where v_1, v_2 and v_3 are the speed of the centers of mass of the inverted pendulum, $v_i^2 = \dot{x}_i^2 + \dot{z}_i^2$. Substituting (2.20,...,2.29) into (2.32) and (2.33) and then into (2.31), we obtain the equation of motion of the triple pendulum, whose compact form is stated as follows.

$$\tau = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) \tag{2.34}$$

where $\mathbf{H} \in \mathbb{R}^{3\times3}$ is the inertia matrix, $\mathbf{C} \in \mathbb{R}^{3\times3}$ is the matrix of centrifugal and coriolis forces and $\mathbf{G} \in \mathbb{R}^{3\times1}$ is the gravity matrix. The components of every matrix

can be expressed as:

$$\begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{pmatrix} + \\ + \begin{pmatrix} 0 & c_{12} & c_{13} \\ c_{21} & 0 & c_{23} \\ c_{31} & c_{32} & 0 \end{pmatrix} \begin{pmatrix} \dot{q}_1^2 \\ \dot{q}_2^2 \\ \dot{q}_3^2 \end{pmatrix} + \begin{pmatrix} g_1 \\ g_2 \\ g_3 \end{pmatrix} \quad (2.35)$$

$$h_{11} = {l_1}^2 (m_1 + m_2 + m_3) \quad (2.36)$$

$$h_{22} = {l_2}^2 (m_2 + m_3) \quad (2.37)$$

$$h_{33} = {l_3}^2 m_3 \quad (2.38)$$

$$h_{12} = h_{21} = (m_2 + m_3) l_1 l_2 cos(q_1 - q_2) \quad (2.39)$$

$$h_{13} = h_{31} = m_3 l_1 l_3 cos(q_1 - q_3) \quad (2.40)$$

$$h_{23} = h_{32} = m_3 l_2 l_3 cos(q_2 - q_3) \quad (2.41)$$

$$c_{12} = -c_{21} = -(m_2 + m_3) l_1 l_2 sin(q_2 - q_1) \quad (2.42)$$

$$c_{13} = -c_{31} = -m_3 l_1 l_3 sin(q_3 - q_1) \quad (2.43)$$

$$c_{23} = -c_{32} = -m_3 l_2 l_3 sin(q_3 - q_2) \quad (2.44)$$

$$g_1 = -g l_1 (m_1 + m_2 + m_3) \sin(q_1) \quad (2.45)$$

$$g_2 = -g l_2 (m_2 + m_3) \sin(q_2) \quad (2.46)$$

$$g_3 = -g l_3 m_3 \sin(q_3) \quad (2.47)$$

2.2.4 Mass Distributed Model

Many authors prefer to model the humanoid as a more accurate multiple mass system, as it is represented in Figure 2.4. This model requires the computation of the mass and inertia tensor, as well as the kinematic and dynamic transformations between joints. It consumes many computer resources and it is usually compute off-line.

An example of mass distributed model is the work of Hirukawa et al. (2006, 2007) that developed a gait generation method taking into account the whole body dynamics using a stability criterion named Contact Wrench Cone (CWC). CWC criteria check is the sum of the gravity and inertia wrench applied to the COM is inside the convex hull. Even though this algorithm has been proved to be strongly stable even under rough terrain, its high computation cost does not make possible for the gait pattern to be generated in real time.

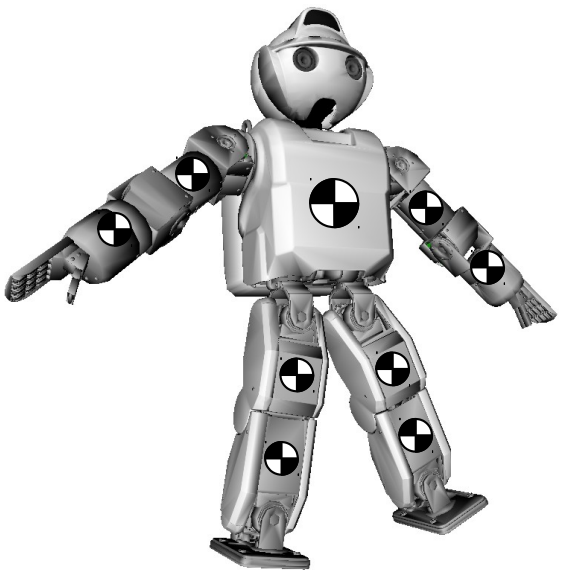


Figure 2.4: *Snapshot of a distributed mass model of the humanoid HOAP-3.*

Khatib and his team work with a complete model of a simulated humanoid robot (Khatib et al., 2004a, 2008; Sentis and Khatib, 2005; Sentis et al., 2010). They defined the equation of motion of the complete humanoid, taking into account the contacts and creating hierarchical tasks using the Operational Space (Khatib, 1987). They made force control for task oriented postures.

Spatial Equation of Motion

Building the dynamic model of a high degree of freedom robot can be tedious. If we are working with a humanoid robot, the problem is more difficult due to the numerous joints and the closed kinematic chains. Spatial formulation of dynamics provides a compact and easy way to implement the notation. This formulation make use of $6D$ vector and tensors to describe velocity, acceleration, inertia and force. Using these components, a set of dynamic algorithms can be developed (Featherstone, 2008).

The equation of motion of a rigid body system (see Figure 2.5) defined using the spatial notation can be expressed as:

$$\mathbf{f} = \frac{d}{dt}(\mathbf{Iv}) = \mathbf{Ia} + \mathbf{v} \times \mathbf{Iv} \tag{2.48}$$

with

$$\mathbf{f} = \begin{pmatrix} n \\ f \end{pmatrix} \in \mathbb{F}^6 \tag{2.49}$$

$$\mathbf{v} = \begin{pmatrix} \omega \\ v \end{pmatrix} \in \mathbb{M}^6 \tag{2.50}$$

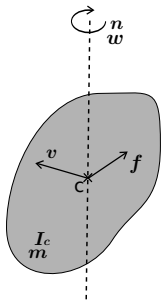


Figure 2.5: *Forces and velocities acting on a rigid body*

$$\mathbf{a} = \begin{pmatrix} \dot{\omega} \\ \ddot{c} - v \times \omega \end{pmatrix} \in \mathbb{M}^6 \tag{2.51}$$

$$\mathbf{I} = \begin{pmatrix} Ic & 0 \\ 0 & m \end{pmatrix} \in \mathbb{M}^{6\times 6} \tag{2.52}$$

where $\mathbf{f}$ is the net spatial force applied in the body, which is composed by $3D$ vectors force f and torque n, $\mathbf{v}$ is the spatial velocity, composed by the linear v and angular velocity ω of the body center of mass, $\mathbf{a}$ is the spatial acceleration and $\mathbf{I}$ is the spatial inertia, composed by the inertia tensor I_c and the mass m.

Spatial Inverse and Forward Dynamics

Inverse dynamics deals with the problem of obtaining the torques applied in every joint given the acceleration of the rigid body system. The generic formula can be expressed as:

$$\tau = ID(model, \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$$

The most used algorithm to calculate inverse dynamic is the Recursive Newton Euler Algorithm (RNEA) presented by Luh et al. (1980), whose spatial formulation can be found in (Featherstone, 2008). This algorithm has a complexity of $O(n)$, where n is the number of degrees of freedom.

RNEA has two phases. First, it calculates recursively the velocity and acceleration of every joint, and then, using (2.48), it calculates the force transmitted in every joint. In a second stage, it computes the joint forces starting at the terminal links and working towards the base.

Forward dynamics consists on determining the accelerations that appears in the actuated joints as a function of the torques applied. The general formulation can be expressed as:

$$\ddot{\mathbf{q}} = FD(model, \mathbf{q}, \dot{\mathbf{q}}, \tau)$$

One of the most cited algorithms for forward dynamics is the Composite Rigid Body algorithm (CRBA), first developed by Walker and Orin (1982). This algorithm computes the inertia matrix of a set of composite rigid bodies and then solve for every

joint acceleration. This matrix can be computed efficiently by applying successively inverse dynamics with joint velocity and acceleration set to zero and it depends on the connectivity of the kinematic chain.

Another approach for solving the forward dynamic is the Articulated Body Algorithm (ABA), developed by Featherstone (1983). It is based on the propagation of the equations in an articulated body. Forward dynamic problem presents two set of unknowns, joint accelerations and joint forces. ABA calculates the coefficients of this equation locally, taking into account one joint at every step. It calculates the acceleration that appears in a joint formed by two bodies, one is the parent body, the second one is an articulated body formed by all the other links of the kinematic chain. It computes this equation recursively, until it finds a local solution (usually at the terminal link) and propagates backwards to obtain a global solution.

Both CRBA and ABA are algorithms to compute forward dynamics. Generally speaking, CRBA is faster than ABA, but ABA is more precise (Asher et al., 1997). Also, they have a spatial formulation than can be found in (Featherstone, 2008).

2.3 Gait Generation

The motion equations for the biped locomotion of a humanoid robot are very complex. Several facets have to be taken into account, first the equations of all degrees of freedom for the kinematic chains, second, the equations of the floating base, which encloses the stability of the movement, and finally, the constraints derived of the contact with the floor and the closed kinematic chains. The result is a dynamic algebraic differential system.

One of the main features of the biped locomotion is the movement periodicity, whose period is a step. This assumes that the joint position and velocity at the beginning and at the end of the movement should be the same.

Another fundamental detail is the continuous change in the support foot. There exist two states in the locomotion process, the double support phase, which represents the 20% of the cycle time and the simple support phase, which represents the 80% of the cycle time. In the first situation, the movement is intrinsically stable, however, in the second situation it is not. A statically unstable situation appears when there is a foot in contact with the ground and the other is moving forward, producing accelerations that affects the mechanism.

In Figure 2.6 a scheme of the biped locomotion cycle is shown. The cycle is divided in two phases, the right step phase and the left step phase. The left step begin when the right foot touches the floor. At that very moment, the weight is transmitted to the right foot, therefore the lateral accelerations that appear have to be controlled through stabilization algorithms. Next, the left foot moves forward during the single support phase, i.e., with only the right feet in contact with the floor. The weight is transmitted forward reaching the double support state, where

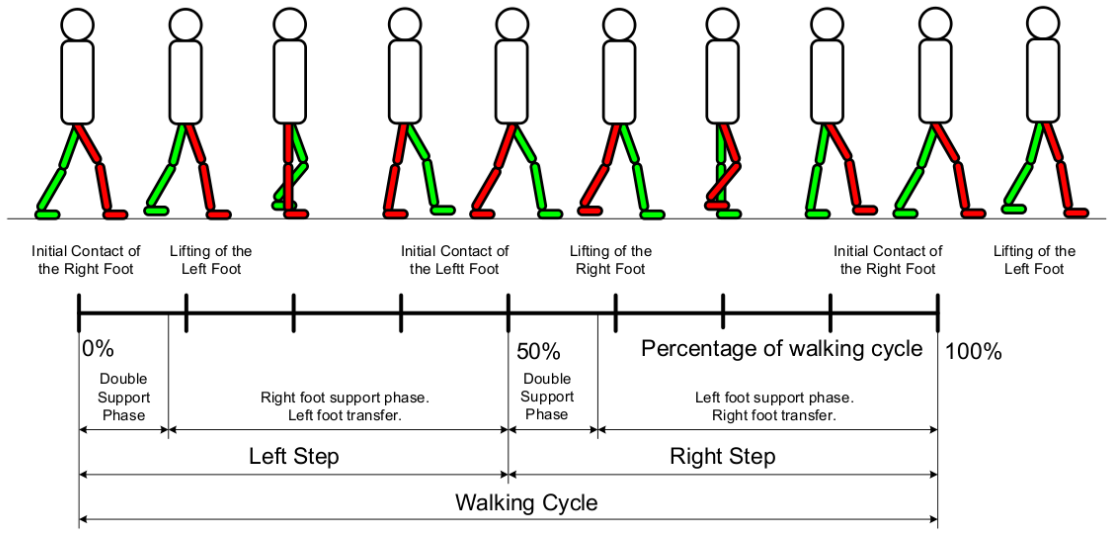


Figure 2.6: *Walking cycle phases ©(Kaynov, 2008)*

both feet touch the floor. Finally, the cycle is repeated for the other foot.

2.3.1 The Concept of Zero Moment Point

The Zero Moment Point (ZMP) is a dynamic balance criterion for humanoid robots established in a series of articles by Vukobratovic and Juricic (1968, 1969). However, the term Zero Moment Point was presented in the subsequent papers (Vukobratovic et al., 1970; Vukobratović and Stepanenko, 1972; Vukobratovic and Borovac, 2004).

To achieve stable locomotion, it is necessary to satisfy some balance criterion in the trajectory planning. The ZMP provides a powerful approach to determine biped locomotion trajectories and any other trajectory which involves loss of balance. The concept of ZMP is one of the most used methods due to its simplicity and effectiveness.

The ZMP is defined as the point in which the moment created by the inertia and gravity forces do not have components in the horizontal axis (Dasgupta and Nakamura, 1999). This theory assumes that the contact surface is flat and the feet do not slip. It is therefore a balance indicator, if the ZMP is inside the support polygon, the movement is dynamically stable, otherwise it is unstable.

To compute the ZMP position it is necessary to calculate the force equilibrium in the support foot (Figure 2.7). The projection of these equations in the horizontal plane is the base of the computation of point P, which is the application point of the reaction force with the floor R.

$$R + F_A + m_s g = 0 \tag{2.53}$$

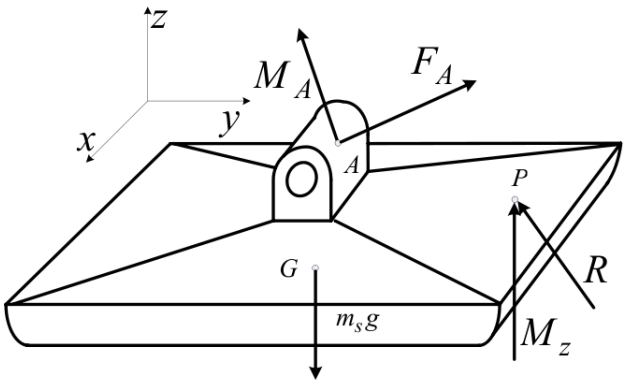


Figure 2.7: *Forces acting in a humanoid foot ©(Vukobratovic and Borovac, 2004).*

$$\overline{OP} \times R + \overline{OG} \times m_s g + M_A + M_z + \overline{OA} \times F_A = 0 \tag{2.54}$$

In general, the resulting reaction force and momentum with the floor have three components $R(R_x, R_y, R_z)$ and $M(M_x, M_y, M_z)$. Assuming that there is no slipping, the reaction force horizontal components, R_x and R_y, will be cancelled by the horizontal components of F_A. In a similar way, the momentum generated by the vertical reaction forces M_z will be cancelled by the vertical component of the momentum acting on the body M_A and the momentum created by the horizontal components of F_A. R_z is the force that cancels the vertical forces of the body.

To compensate the horizontal components of M_A produced by a movement or by an additional load, it is necessary to change the application reaction force R_z (Figure 2.8). If this point is inside the area covered by the foot sole, then the system is in equilibrium because the horizontal components of the reaction momentum are cancelled, $M_X = 0$ and $M_Y = 0$. As the momentum M_{Ax} increases, the application point of force R_z moves towards the foot edge. If this point leaves the foot sole, there will appear momentums $M_X \neq 0$ and $M_Y \neq 0$, which will cause the mechanism to rotate around the foot edge.

Therefore, the ZMP is defined as the point $P_{ZMP} = (x_{ZMP}, y_{ZMP}, 0)$ in which the momentum of the reaction force has no horizontal components. This can be understood as a dynamic balance criterion, given that when the ZMP is inside the support polygon, the robot is stable, otherwise it is not.

The ZMP of a multi-body rigid system can be stated as:

$$x_{ZMP} = \frac{\sum_{i=1}^{n} m_i x_i(\ddot{z}_i + g) - \sum_{i=1}^{n} m_i \ddot{x}_i z_i - \sum_{i=1}^{n} I_{iy}\alpha_{iy}}{\sum_{i=1}^{n} m_i(\ddot{z}_i + g)} \tag{2.55}$$

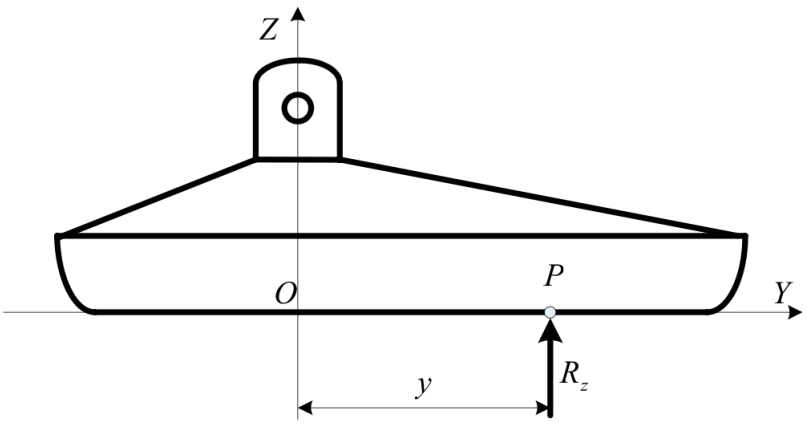


Figure 2.8: *ZMP compensation ©(Vukobratovic and Borovac, 2004).*

$$y_{ZMP} = \frac{\sum_{i=1}^{n} m_i y_i(\ddot{z}_i + g) - \sum_{i=1}^{n} m_i \ddot{y}_i z_i + \sum_{i=1}^{n} I_{ix}\alpha_{ix}}{\sum_{i=1}^{n} m_i(\ddot{z}_i + g)} \tag{2.56}$$

In order to compute the ZMP, it is necessary to know all forces and momentums acting in the structure in real time, as well as the reaction forces and momentums. The computation of these momentums and forces are usually slow due to the high number of degrees of freedom that a humanoid has. Therefore, researchers use different simplified models, that allow to generate stable walking trajectories in real time. An example is the 3D Linear Inverted Pendulum Mode that will be explained in detail next.

2.3.2 The 3D Linear Inverted Pendulum Mode

The 3D Linear Inverted Pendulum Mode (3D-LIPM) models the humanoid as a single mass inverted pendulum, where all the humanoid masses are concentrated in its COM (Kajita et al., 2001). The pendulum connects the supporting foot with the COM with a massless telescopic link.

To generate a gait pattern, the 3D-LIPM constraints the COM to move in an horizontal plane and computes a smooth trajectory, which follows the pendulum law motion under a gravity field and ensures that the ZMP is inside the support polygon.

Given (2.10), the dynamics along the x axis is given by

$$m(z\ddot{x} - x\ddot{z}) = \frac{\sqrt{1 - \sin^2\theta_r - \sin^2\theta_p}}{\cos\theta_p}\tau_p + mgx \tag{2.57}$$

The dynamics along the y axis is given by

$$m(z\ddot{y} - y\ddot{z}) = \frac{\sqrt{1 - \sin^2\theta_r - \sin^2\theta_p}}{\cos\theta_r}\tau_r - mgy \tag{2.58}$$

One assumption of the LIPM is that the motion is constrained to move in a plane:

$$z = k_x x + k_y y + z_c \tag{2.59}$$

where z_c is the height of the robot COM. Replacing (2.59) into (2.57) and (2.58), we obtain:

$$\ddot{x} = \frac{g}{z_c}x + \frac{k_y}{z_c}(x\ddot{y} - y\ddot{x}) + \frac{1}{mz_c}\frac{\sqrt{1 - \sin^2\theta_r - \sin^2\theta_p}}{\cos\theta_p}\tau_p \tag{2.60}$$

$$\ddot{y} = \frac{g}{z_c}y - \frac{k_x}{z_c}(x\ddot{y} - y\ddot{x}) - \frac{1}{mz_c}\frac{\sqrt{1 - \sin^2\theta_r - \sin^2\theta_p}}{\cos\theta_r}\tau_r \tag{2.61}$$

Assuming the motion is constrained to a flat plane ($k_x = 0$ and $k_y = 0$) and there is no input torques ($\tau_p = 0$ and $\tau_r = 0$), we can simplify (2.60) and (2.61) to

$$\ddot{x} = \frac{g}{z_c}x \tag{2.62}$$

$$\ddot{y} = \frac{g}{z_c}y \tag{2.63}$$

2.3.3 The Cart-Table Model

Other common method of gait generation is the cart-table model (Kajita et al., 2003a). This model is based on a preview control scheme to obtain the COM trajectory from a defined ZMP trajectory. This method generates a dynamically stable gait trajectory using the 3D Linear Inverted Pendulum Model (Kajita et al., 2001) to approximate the humanoid dynamics. The relationship between ZMP trajectory and COM trajectory is defined by the following equations:

$$p_x = x - \frac{\ddot{x}}{g}z_c \tag{2.64}$$

$$p_y = y - \frac{\ddot{y}}{g}z_c \tag{2.65}$$

where p_x is the ZMP reference, x is the COM trajectory, $\ddot{x}$ the COM acceleration, z_c is the COM height and g is the gravity. In cart table model (Figure 2.9), the cart mass corresponds to the center of mass of the robot. If the cart accelerates at

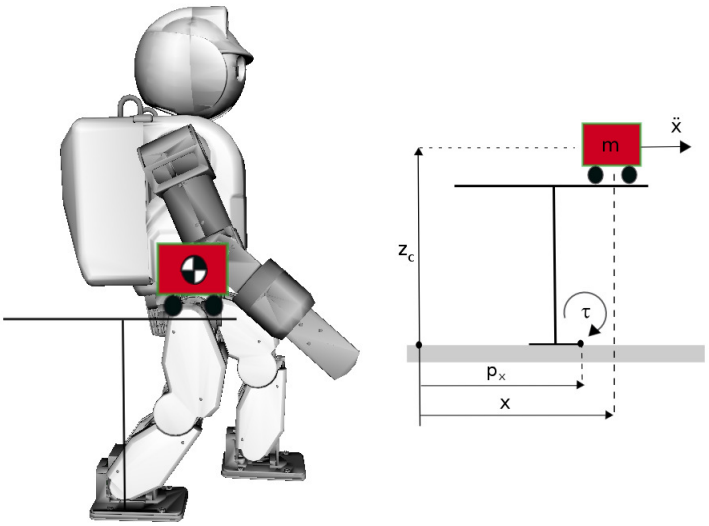


Figure 2.9: *Cart table model in sagital plane in the humanoid HOAP-3.*

a proper rate, the table can be upright for a while. At this moment, the moment around p_x is equal to zero, so the ZMP exists (eq. (2.66)).

$$\tau_{ZMP} = mg(x - p_x) - m\ddot{x}z_c = 0 \tag{2.66}$$

In order to obtain the COM trajectory, the ZMP control is defined as a servo problem. Using the optimal preview servo controller technique proposed by Katayama et al. (1985), the COM trajectory can be obtained from a ZMP reference.

Let the derivative of the COM acceleration be defined as:

$$u_x = \frac{d}{dt}\ddot{x} \tag{2.67}$$

Using u_x as the input of the eq. (2.64), the ZMP equations can be stated in the form of a variable state problem.

$$\frac{d}{dt}\begin{pmatrix} x \\ \dot{x} \\ \ddot{x} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} x \\ \dot{x} \\ \ddot{x} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u_x \tag{2.68}$$

$$p_x = \begin{pmatrix} 1 & 0 & z_c/g \end{pmatrix}\begin{pmatrix} x \\ \dot{x} \\ \ddot{x} \end{pmatrix} \tag{2.69}$$

The COM trajectory is discretized as a piecewise cubic polynomial at intervals of constant time T. Using the notation:

$$\hat{x}_k = \begin{pmatrix} x(kT) \\ \dot{x}(kT) \\ \ddot{x}(kT) \end{pmatrix}, u_k = u_x(kT), p_k = p_x(kT) \tag{2.70}$$

(2.68) and (2.69) can be transformed into:

$$\hat{x}_{k+1} = \begin{pmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{pmatrix} \hat{x}_k + \begin{pmatrix} T^3/6 \\ T^2/2 \\ \ddot{x} \end{pmatrix} u_k \tag{2.71}$$

$$p_k = \begin{pmatrix} 1 & 0 & z_c/g \end{pmatrix} \hat{x}_k \tag{2.72}$$

The COM constraints are defined by:

$$p_k^{min} \leq p_k \leq p_k^{max} \tag{2.73}$$

where the maximal and minimal value are defined by the edge of the feet.

In order to design the optimal servo controller the performance index can be expressed as:

$$J = \sum_{i=k}^{\infty} \left\{ Q_e e(i)^2 + \Delta x^T(i) Q_x \Delta x(i) + R \Delta u^2 \right\} \tag{2.74}$$

where $e(i) = p(i) - p^{ref}(i)$ is the servo error, Q_e, $R > 0$, Q_x is a symmetric non-negative definite matrix, $\Delta x = x(k) - x(k-1)$ is the incremental state vector and $\Delta u = u(k) - u(k-1)$ is the incremental input.

The optimal controller that minimizes the index in (2.74) is given by:

$$u(k) = -G_i \sum_{i=0}^{k} e(k) - G_x \hat{x}(k) - \sum_{j=1}^{N_L} G_p(j) p^{ref}(k+j) \tag{2.75}$$

where G_i, G_x and G_p are the controller gains and N_L is the preview period time.

Equations (2.74) and (2.75) give the original solution to cart-table method proposed by Kajita et al. (2003a). However, we used the implementation of Wieber (2006) to perform a stable walking pattern in the HOAP humanoid.

In Figure 2.10, the COM trajectory of the sagittal plane is obtained from the ZMP reference. The humanoid walks 12 steps forward, with $z_c = 32$cm, $T = 5$ms, $N = 300$ and $R/Q = 10^{-5}$.

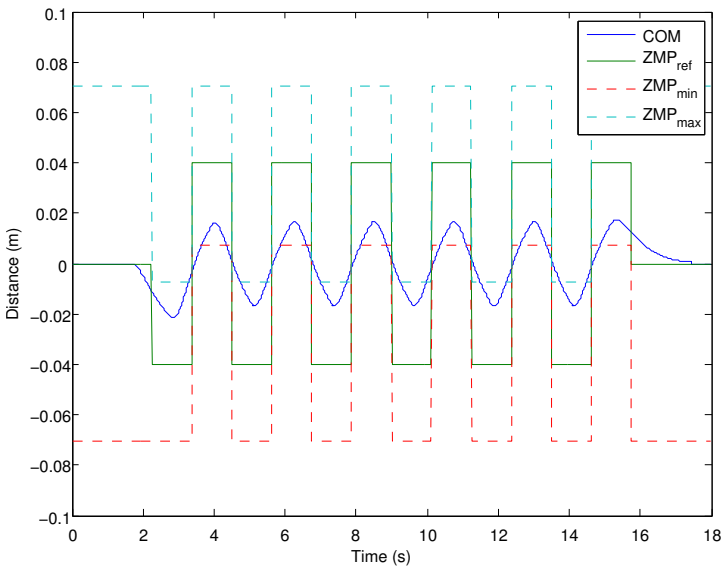


Figure 2.10: *Trajectory of the COM (in blue) and the ZMP reference (in green). The dotted lines are the ZMP limits.*

2.3.4 Experiment: a Humanoid Walking

Once the COM trajectory is obtained using the cart-table method, the trajectories of the lower body joints are calculated applying inverse kinematics.

Simulation of a Walking Pattern

Simulation platforms are fundamental in robotics research, especially with humanoid robots, since they allow to develop the controllers and the necessary programming tools without compromising the complex and expensive mechanical system. In general, the purposes of simulators are:

- To visualize three-dimensional work environment and the model of the robot in motion.
- To provide a test center for the development and evaluation of controls and software of the robot.
- To serve as a graphical user interface, which can even be interactive in real time with the robot.

A necessary requirement for really effective simulations is that the mechanical behavior of the virtual robot answers as closely as possible to the real robot, so the set-up of a virtual reality simulation platform turns out to be crucial. Thereby,

the programming developed over the simulator will be able to be inherited by real applications.

Existing robotics simulators include, among others, Honda and Sony simulators (proprietary for ASIMO and the QRIO), the Fujitsu HOAP simulator (Fujitsu sells HOAP with a basic simulation software), RoboWorks (a commercial software developed by Newtonium), SD/FAST (by Symbolic Dynamics, which provides nonlinear equations of motion from a description of an articulated system of rigid bodies), Webots (a commercial software by Cyberbotics) and Gazebo (developed by Koenig and Howard (2004) and integrated in ROS, the Robotic Operating System). Even Microsoft has developed a product named Microsoft Robotics Studio, which is primarily used for mobile robots.

It is important to mention the OpenHRP platform (Open Architecture Humanoid Robotics Platform) (Kanehiro et al., 2001, 2004) as a simulator and motion control library for humanoid robots developed at AIST (National Institute of Advanced Industrial Science and Technology of Japan). This is a distributed framework based on CORBA (Common Object Request Broker Architecture), created with the idea of sharing a code between real and virtual robots, and ultimately of developing identical controllers for real and virtual robots.

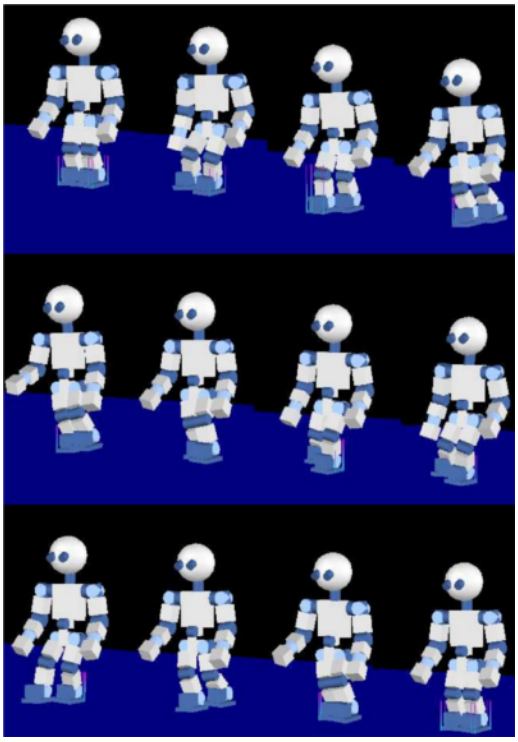


Figure 2.11: *Simulation of humanoid robot HOAP-3 in OpenHRP while performing a gait pattern.*

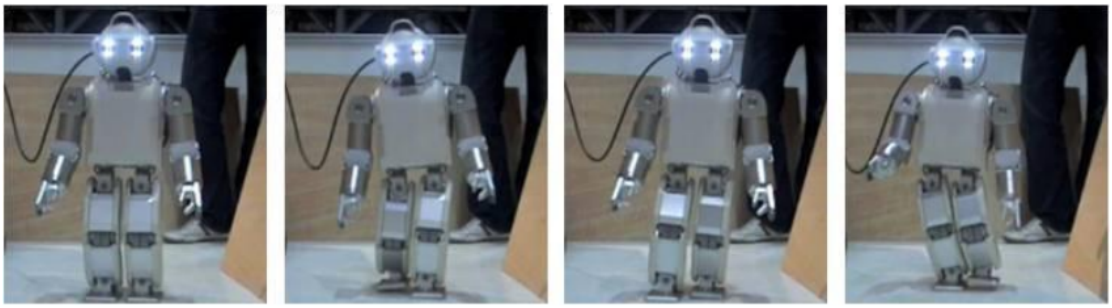


Figure 2.12: *Biped locomotion in the real robot.*

Another used platform is OpenRAVE (Open Robotics Automation Virtual Environment) created by Diankov (2010). The main focus is on simulation and analysis of kinematic and geometric information related to motion planning. OpenRAVE's stand-alone nature allows it to be easily integrated into existing robotics systems. It provides many command-line tools to work with robots and planners, and the run-time core is small enough to be used inside controllers and bigger frameworks. It also includes several dynamic libraries ODE libraries (Open Dynamic Engine) or Bullet Physic Libraries.

OpenHRP and OpenRAVE simulators have been used in this thesis to simulate different motion patterns before the final real-time test with the robot (Monje et al., 2011b; González-Fierro et al., 2013b). In Figure 2.11 some snapshots of a simulation of the humanoid HOAP in OpenHRP are presented.

Implementation of a Locomotion Routine in a Humanoid

Once the stability of the robot is guaranteed in simulation, the joint trajectories are loaded in the real HOAP-3 platform and the walking test is executed experimentally. One of the main problems that the programmer faces is the mismatch between the simulation and the real world. The contact of the robot feet with the floor can create in some cases undesired reactions that perturbs the movement. This reactions may not be detected in simulation. In Figure 2.12 it is shown a set of snapshots where the HOAP-3 robot is walking forward.

One of the most important tests that needs to be performed before implementing any kind of trajectory in the robot is the inverse dynamics. We need to be sure that the trajectory does not produce excessive torques. In Figure 2.13 and Figure 2.14 the inverse dynamics of the locomotion trajectory is presented. Taking the right leg as a support, the left leg swings forward, developing small torques in comparison with the right leg ones, which supports almost all humanoid weight. The joints that suffer a higher variation are the 3^{rd}, 4^{th} and 5^{th}, which are the ones whose axis is perpendicular to the sagittal plane, and therefore, the ones that produce the movement.

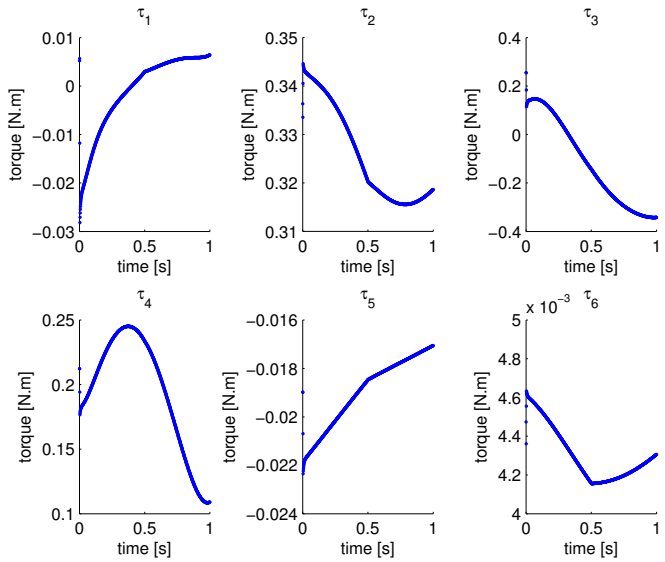


Figure 2.13: *Torques in the flying leg.*

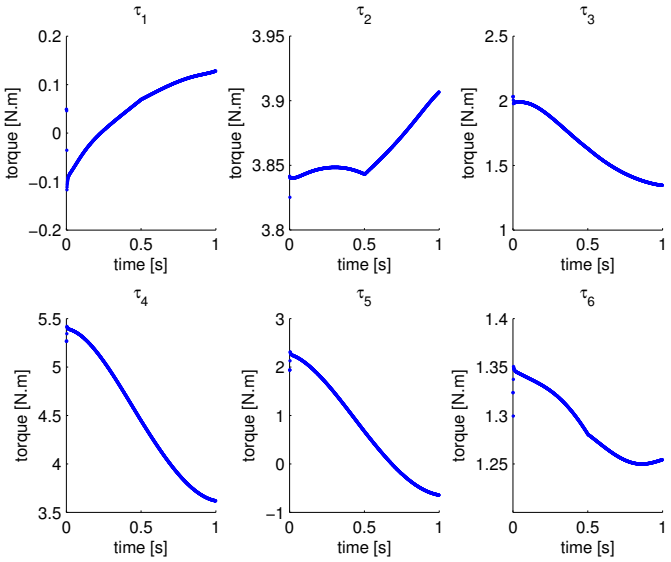


Figure 2.14: *Torques in the supporting leg.*

2.4 Full Body Trajectory Generation through Mimicking

The computation of a full body trajectory for a humanoid is a complex problem. In some cases the movement is generated taking the body as a whole (Nagasaka et al.,

1999; Yamane and Nakamura, 2003; Khatib et al., 2004a, 2008; Hirukawa et al., 2006, 2007).

Another approach is to accomplish whole body robot movement through imitating a human. Usually, the problem is divided in two. The upper part of the body, which can be imitated without taking into account the stability of the robot and the lower part, where the ZMP criteria is used. The study of Nakaoka et al. (2005, 2007) is an example of this. Based on that idea, we developed an approach to perform imitation routines in the HOAP humanoid (González-Fierro et al., 2012; Monje et al., 2013).

2.4.1 Mimicking and Movement Adaptation of a Dancing Routine

There are many works regarding dance performance imitation in humanoid robots. In (Pollard et al., 2002) a method to scale human upper body motion capture data to a humanoid robot is proposed. Other example of upper body motion imitation can be found in (Shiratori et al., 2007), where the dance performance speed is taken into account to control the robot. An approach similar to ours is based on acquiring human motion from a motion capture system and then convert the motion into primitives (Nakazawa et al., 2002; Nakaoka et al., 2005, 2007).

We created a dance trajectory imitating the dance performance of a professional dancer (González-Fierro et al., 2012). To simplify the dance adaptation to the robot, we constructed the set of motion primitives performed by a dancer and imitated by the robot. The complete dance routine consist of 12 different motion primitives, which combines arms and legs movement.

First, a tracking vision system to capture the movement of the dancer was developed (see Figure 2.15). Using a set of 3 tags placed at the shoulder, elbow and hand, the movement of the dancer arms were tracked. To track the $3D$ trajectories of the tags, we used a color histogram segmentation and a Kalman filter. The noisy trajectory obtained is smoothed using a third order spline and some via points are manually adapted to obtain a more coordinated movement.

In such way, it is possible to use Inverse Kinematics algorithms in order to get the joint angles of the robot arm. In Figure 2.16 the algorithm is presented.

The reference position and velocities of the human arm are used as input. The human arm angle velocities can be calculated using the equation

$$\dot{\mathbf{q}}_H = \mathbf{J}_H^{\dagger} \left[\dot{\mathbf{x}}_H + \mathbf{K} \left(\mathbf{x}_H - k(\mathbf{q}) \right) \right] \tag{2.76}$$

where the pseudo-inverse of the Jacobian Matrix is used since only the position of the arm is considered. The remaining degrees of freedom can be used in order to adapt the different range of movements of the HOAP-3 robot with respect to the

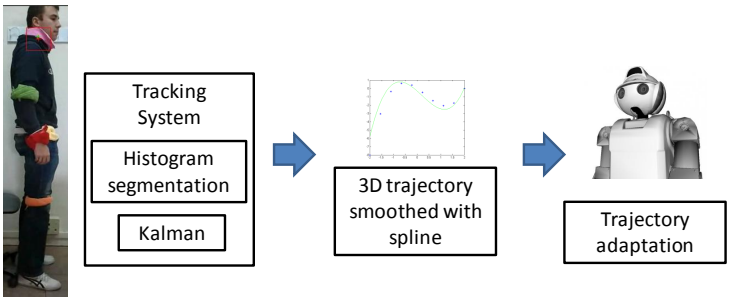


Figure 2.15: *Vision tracking system and adaptation of 3D trajectory for the humanoid robot.*

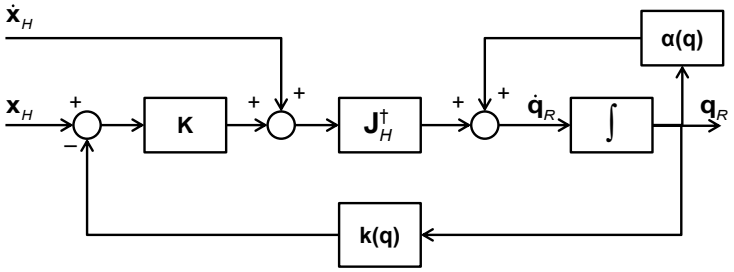


Figure 2.16: *Adaptation of the inverse kinematics for the human arm to the robot arm.*

human arm. So, the velocity of robot arms are calculated as:

$$\dot{\mathbf{q}}_R = \mathbf{J}_H^\dagger \left[\dot{\mathbf{x}}_H + \mathbf{K}\left(\mathbf{x}_H - k(\mathbf{q})\right)\right] + \alpha(\mathbf{q}_R) \tag{2.77}$$

where

$$\alpha(\mathbf{q}_R) = \left[\mathbf{I} - \mathbf{J}_H^\dagger \mathbf{J}_H\right] \dot{\mathbf{q}}_0 \tag{2.78}$$

The vector $\dot{\mathbf{q}}_0$ can be calculated in order to get a solution of joint angles being far from the HOAP-3 joints limits, while getting the same end-effector trajectory:

$$\dot{\mathbf{q}}_{0,i} = -k_l \frac{\mathbf{q}_{H,i} - \overline{\mathbf{q}}_{R,i}}{(\mathbf{q}_{R,i,M} - \mathbf{q}_{R,i,m})^2} \tag{2.79}$$

with $k_l > 0$.

At the same time, if the robot moves the lower part of the body, it is necessary to maintain the ZMP inside the support polygon. The leg motion is calculated independently from the arm motion and they are constrained to maintain stability. The movement of the legs is computed tracking the 3D position of both feet and hip of the dancer. These three points are adapted to the size of the robot and the ZMP

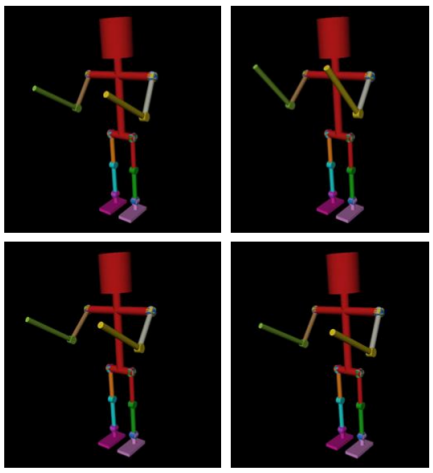


Figure 2.17: *Simulated humanoid robot performing a circular trajectory primitive with the arms.*

trajectory is computed using (2.55) and (2.56) to ensure the dynamic balance and stability.

As an example, the forward and inverse dynamics of a circular trajectory like the one observed in Figure 2.17 are presented next. The torques that are observed are very small (Figure 2.19). Joint 1 has a higher torque due to its wide movement (Figure 2.18). Finally, in Figure 2.20 the joint accelerations are obtained using two different methods of forward dynamics, CRBA algorithm and ABA algorithm, and compared with the real acceleration.

This set of dynamic algorithms is obtained and applied with the objective of setting a base for the developments that are performed in the thesis. It has a fundamental importance to know the torques that appears in the robot to secure its correct operation.

2.4.2 Experiment: A Humanoid Dancing

Before implementing the trajectories in the real robot we test them in simulation. Using OpenInventor libraries, a simple robot model can be simulated. It can be used to see the robot trajectories (see Figure 2.21).

Simulation of a Humanoid Dancing

The complete performance, that includes both upper part and lower part movement, was implemented in OpenRAVE simulator which can be seen in Figure 2.22.

OpenRAVE is able to reproduce a very realistic simulation of the robot. The model includes all masses and inertia tensors of the real robot. Since it incorporates a dynamic engine, it also takes into account the friction with the floor and the

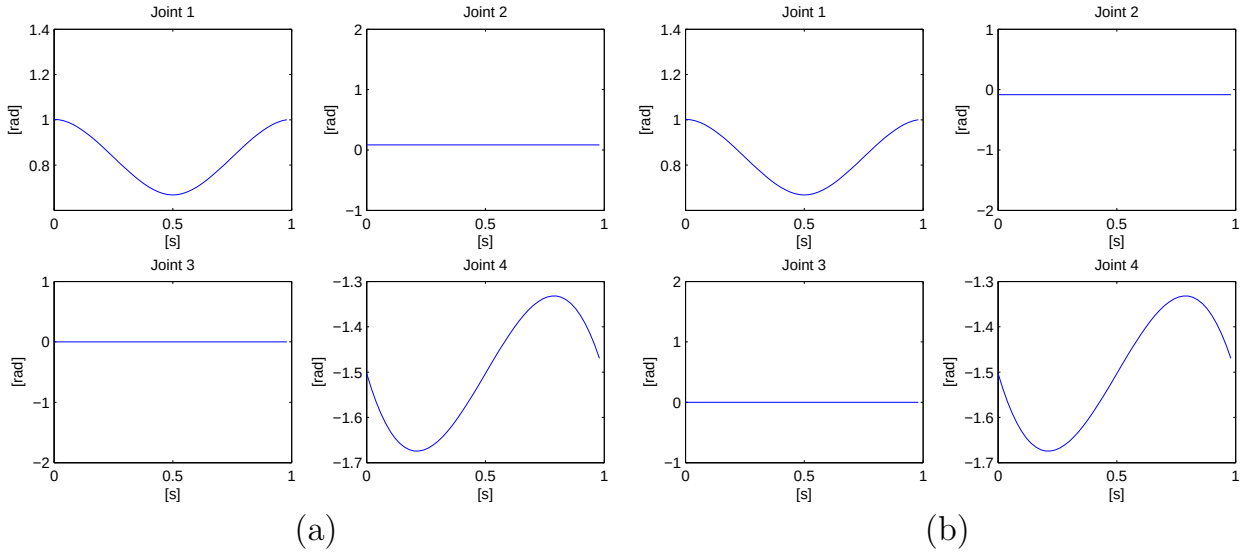


Figure 2.18: *(a) Joint position trajectory for left arm. (b) Joint position trajectory for right arm.*

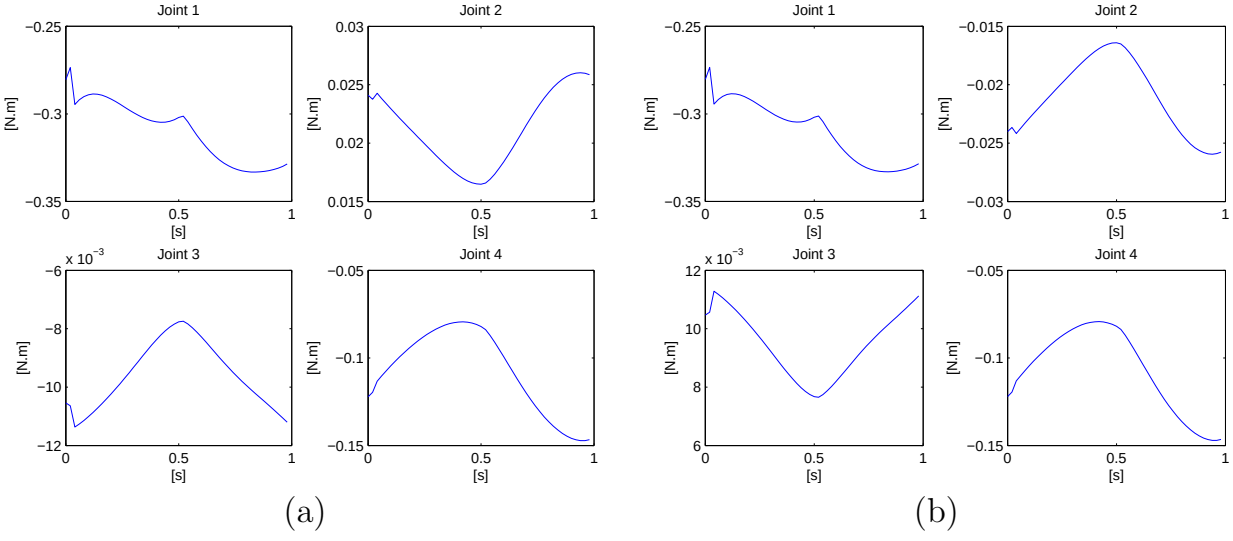


Figure 2.19: *(a) Joint torque trajectory for left arm. (b) Joint torque trajectory for right arm.*

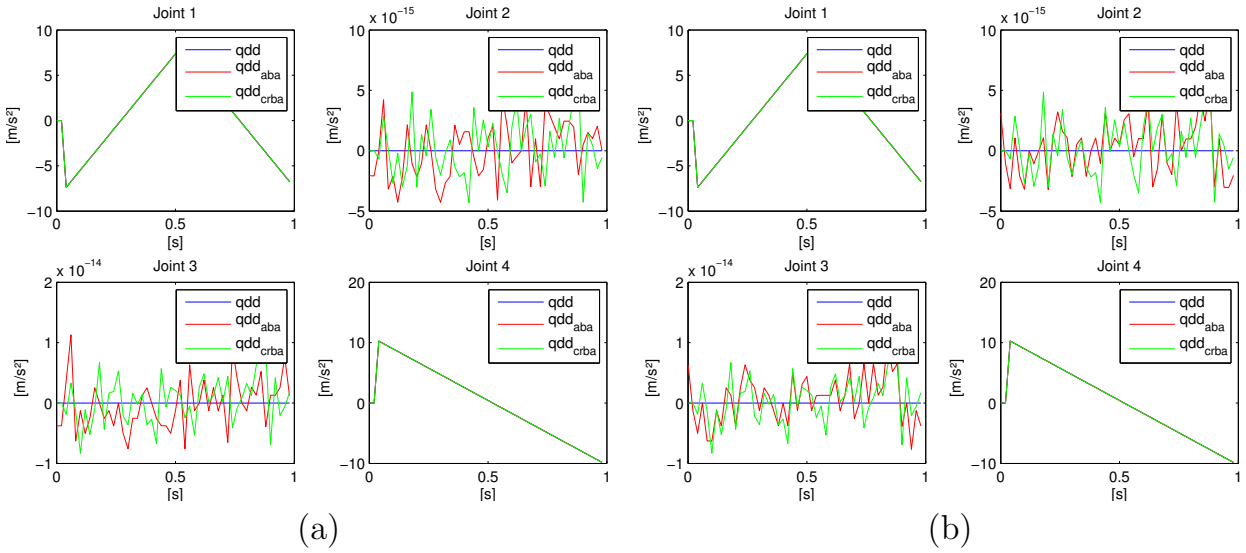


Figure 2.20: *(a) Joint acceleration trajectory for left arm. (b) Joint acceleration trajectory for right arm.*

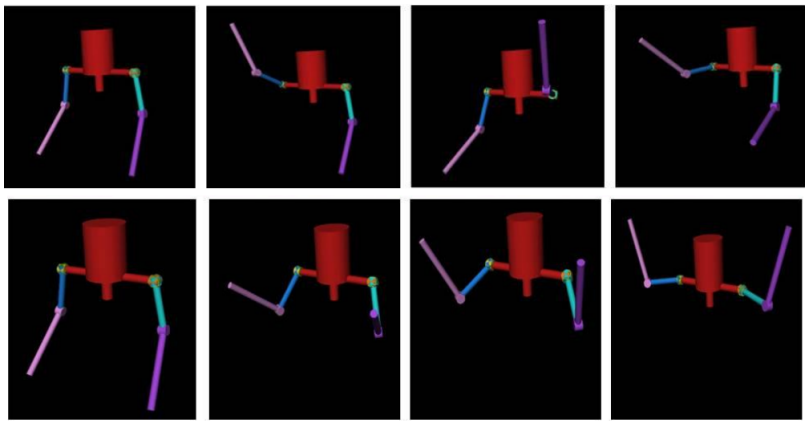


Figure 2.21: *Simulation of two different dancing primitives of the upper body.*

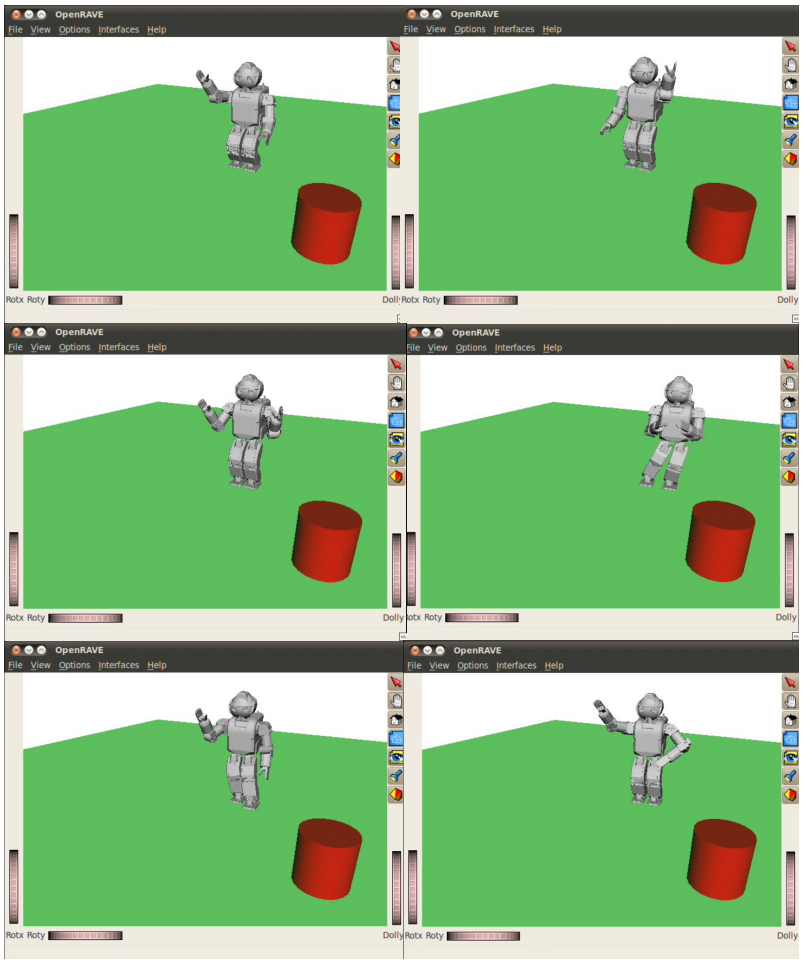


Figure 2.22: *Simulation of the dance routine in the OpenRAVE simulator.*

gravity force. Therefore, the robot can move in the environment, there are contacts with the floor and the accelerations produced by the body masses and inertias are simulated. Even though it is not perfect, because there are always errors in the robot model parameters and in the simulator itself, OpenRAVE allows to have a good first approximation of the robot performance.

The dance routine was implemented as well in OpenHRP simulator. Both simulators have its advantages and disadvantages. An interested feature of OpenHRP is the possibility to obtain the forces and torques that appears in the humanoid model. This have to be done manually in OpenRAVE. Even though OpenHRP simulates very closely the AIST series of robots HRP, it is difficult to create a model of another robot and tune its parameters. On the contrary, OpenRAVE has a simpler and friendlier SDK and the construction of the model is easier. We obtained more realistic simulations when we were simulating the HOAP-3 robot in OpenRAVE

Figure 2.23: *Dance performance of the humanoid robot HOAP-3 in the robotics show. In the image it can be seen a synchronized simulation of the dance in OpenHRP simulator.*

than in OpenHRP.

Implementation of a Dancing Routine in a Humanoid

Once the routine has been tested in simulation, it is implemented in the real humanoid.

We also obtained the torque trajectory of the dance performance to validate the movement. The computation of the robot dynamic model constitutes an essential start point to the development of any kind of research which involves the study of movement. The dance routine was analysed to ensure that no joint surpass the allowed torques.

The complete routine consisted on a combination of 9 different primitives that were mixed in 17 dance periods. The performance included different arm motions, forward steps and lateral steps. The robot maintained in all moment the balance.

The complete dance performance was presented in a robotics show to all Spanish robotic community (see Figure 2.23). The robot danced in a stage accompanied by a musical improvisation. It can be seen in this video[1].

2.5 Discussion and Conclusions

This chapter deals with two basic concepts, robot modeling and motion generation. These are concepts widely discussed and studied in related works. We presented

[1] http://www.youtube.com/watch?v=mu5psxG7bwA (last visit March 2014)

different approaches of humanoid robot modeling, ranging from the simple concentrated model of the inverted pendulum to a much more complex models like the distributed mass model. They both have their advantages and disadvantages. Furthermore, we presented some motion generation methods that are used throughout the thesis. Some were related to biped locomotion and some related to a general whole-body motion, like a dancing routine. The most used balance criterion for humanoids, the ZMP, was also presented.

Now, the discussion that scientific community set is: do we really need models? The answer to this question could be found in the human being. Does the human being have an internal model of himself that applies in a way to generate a behavior and does he has a model of the environment? There are many scientist who aims to find a set of generalized equations that were able to model both robot and environment to achieve a unified representation. On the other hand, there are colleges that believe that the robot should learn everything, whether it is learned from humans or robot demonstrations or it is learned by auto exploration.

In this thesis there is a positioning for the learning approach, but at the same time, it relies a little bit on models. It can be presented as a mixed approach. In the following chapters we will discuss methods to generate robot skills using human demonstrations as a start point.

Chapter 3

Imitation Learning and Skill Innovation in Humanoids through Reward Templates

The imitation of a skill and its improvement by innovating new solutions are the key steps to human learning. The process of human learning has been widely studied in psychology, sociology and neuroscience. Terms such as emulation or imitation are common in these sciences and widely discussed. The behavior transference from a human to a robot is the content of this chapter. Standing up from a chair to a stable upright posture causes conventional Zero Moment Point (ZMP) based controllers of humanoid robots to produce excessive joint torques. Humans, however, are known to manage this challenging dynamic posture control task very elegantly. This chapter proposes a novel method for humanoid robots to acquire optimal standing up behaviors based on human demonstrations. We collected 3D motion data of a group of human subjects standing up from a chair. We solve the correspondence problem by making comparisons in a common reward space defined by a multi-objective reward function. We fitted a fully actuated triple inverted pendulum model to both human and robot motion data in order to compute a reward profile for stability and effort. Afterwards, we used Differential Evolution optimizer to obtain a trajectory that minimizes the Kullback-Liebler divergence between the reward of the human and that of the robot, subject to constraints of ZMP, joint torques, and joint rotation limits of the robot. This chapter presents an advancement in how a humanoid robot can learn to imitate and innovate motor skills from demonstrations of human teachers of larger kinematic structures and different actuator constraints.

3.1 Introduction

Consider a child learning motor skills based on demonstrations performed by his parent. In this case, the problem of relating demonstrations performed by the parent to the child's own kinematic scale, weight and height, known as the *correspondence problem*, would be one of the complex challenges that should be solved first. The correspondence problem is one of the crucial problems of imitation and can be stated as the mapping of action sequences between the demonstrator and the imitator (Schaal et al., 2003; Alissandrakis et al., 2002). This problem can be solved by mapping movements made in a different kinematic scale to a common domain, such as a set of optimality criteria.

From that perspective, the child could find a solution which fits his own muscular strength, size, reachable space and kinematic characteristics which somehow matches the level of optimality of demonstrations performed by the parent. Moreover, if comparisons are made in an optimality domain, the child could even innovate solutions that can be more relevant to his kinematic structure, but closely follow the optimal solution demonstrated by the parent. This comparison is best done in a common reward landscape, specified by a set of reward functions rather than in the kinematic domain or in the muscle effort domain, since similar behavioral goals should give similar trajectories in a common reward landscape subject to a set of constraints.

We present experimental results for the task of standing up from a chair to a stable upright posture, where the robot has to transit from one stable posture to another via a set of unstable states. The results were published in (González-Fierro et al., 2013a, 2014a).

3.1.1 Foundations of Imitation and Innovation in Humans

A wide range of work has been done in the area of *observational learning* from a psychological point of view (Bandura, 1986; Taylor et al., 2013). It ranges from how children learn from demonstrations (Thompson and Russell, 2004) to how apes learn certain motor skills based on demonstrations (Tomasello, 1996).

Thompson and Russell (2004) suggested that children learn not only by imitating, but also by understanding how the process works, what is known as *emulation learning*. For example, to understand that a doorknob twist will open a door will help to learn how to leave a room. Even, the high predisposition of children to learn from observation suggested a more appropriate name for the human species: *homo imitans*, which means "man who imitates" (Meltzoff, 1988). There are also many experiments conducted with apes (Whiten et al., 2004; Hopper et al., 2008) that support the argument that learning based on demonstrations can happen in intrinsic domains to do with the context of the kinematic domain.

Results recently presented point out that two individuals performing the same

action do not share the same muscle control sequence or visual perception of the action, but the goal and strategy to perform that action (Metta et al., 2006; Craighero et al., 2007). A class of visuomotor neurons called *mirror neurons*, first discovered by Rizzolatti, that are activated when a similar goal-directed action is observed, also suggest that goals of the task mediate the recruitment of computational elements of learning based on demonstrations instead of just mere kinematics of the movement (Rizzolatti et al., 2002).

It has been demonstrated that even animals can outperform the optimality of the demonstrated behavior in certain contexts. In a task of pushing a lever to obtain a food reward, rats finally associated the amount of food to the rate of pushing the lever, which was not demonstrated at the beginning (Heyes and Dawson, 1990; Heyes et al., 1994). Similar observations have been made in other experiments with birds (Nguyen et al., 2005; Akins and Zentall, 1998) and apes (Whiten et al., 1996). Even Piaget, the father of the constructivist theory of knowing, hypothesized that the likelihood of matching a response may depend on the expected outcome for the observer (Piaget, 1962).

The phenomenon known as *goal emulation* shows that the observer can reproduce the result of a behavior with a method slightly different from that of the demonstrator (Whiten and Ham, 1992; Gergely et al., 2002; Metta et al., 2006; Craighero et al., 2007). This is similar to what Mitchell (1987) calls *fourth-level imitation*, where an individual tends to reproduce a model understanding the consequences of that behavior, and performs a different behavior maintaining what they called *intentionality*.

Recent work in emulation show that apes are more suitable to emulate while children show more tendency to *over-imitate*, in the sense that children make an attempt to improve the optimality of the learnt skills. In that sense, skill innovation is therefore an essential part of the human behavior (Whiten et al., 2009).

Skill innovation among humans is also studied in business theory and management (Amabile, 1996; Burns and Stalker, 2009), sociology (West, 1990), biology (Love, 2003) and political science (Nelson, 1993; Ball et al., 1989).

A good review of the process of learning from a biological and psychological point of view can be found in (Zentall, 2006).

3.1.2 Learning from Demonstration and Skill Innovation in Robots

A humanoid robot sharing tools and space in a human society can benefit from a sound framework of learning based on demonstrations, that can vary across trials (Argall et al., 2009). Despite the challenges to solve the correspondence problem (Schaal et al., 2003; Alissandrakis et al., 2002; Byrne, 2003; Billard et al., 2004; Peters and Schaal, 2008c), there has been a growing interest in this area due to

several advantages such as the simplification of communicating a complex behavior through a demonstration (Jeannerod, 1988), the absence of the need to have complex mathematical models of the dynamical system to learn an optimal behavior, and the fact that it does not require an expert teacher to perform the demonstrations, which simplifies the information gathering process (Schaal, 1999).

Similar to the biological world, robotic imitation is achieved under a set of schemes, as stated by (Schaal et al., 2003):

- Determine what to imitate, inferring the goal of the demonstrator.
- Establish a metric for imitation.
- Map between dissimilar bodies.
- Compute the control commands to perform the imitation behavior.

The problem of skill transfer and whole body motion transfer has been an interesting area of research in recent years. Some studies addressed the problem of manipulating the COM angular momentum (Naksuk et al., 2005), using graphs and Markov chains (Kulić et al., 2008), defining a spatio-temporal models based on movement primitives (Ilg et al., 2004) or encoding and organizing learned skills (Lin and Lee, 2008).

Different approaches have been presented to address the problem of robot programming from demonstration, that ranges from standing up while holding onto a support object (Inaba et al., 1996), using a graph of stable state transitions (Hirukawa et al., 2005), to imitation of human demonstrations (Mistry et al., 2010b; Suleiman et al., 2008). In (Kormushev et al., 2011) a humanoid learns to write on a wall mounted board while maintaining balance. In (Morimoto and Doya, 2001) a three link simulated pendulum learns to stand up using a hierarchical Reinforcement Learning (RL) method.

Several studies have been conducted in the area of Learning from Demonstration (LfD) using Gaussian Mixture Models (GMM), that encode a set of trajectories, and Gaussian Mixture Regression (GMR), to obtain a generalized version of these trajectories to perform a robot movement (Calinon and Billard, 2007; Calinon et al., 2007; Calinon, 2009; Khansari-Zadeh and Billard, 2011). Calinon et al. (2010) present a combination of Hidden Markov Models (HMM) with GMR that is used to perform imitation handling partial demonstrations.

LfD can also be used to study the physical collaboration and interaction between a robot and a human (Calinon et al., 2009; Calinon, 2009; Evrard et al., 2009; Gribovskaya et al., 2011).

The framework called *incremental learning* uses a few demonstrations to perform a task which is incrementally improved with the aid of verbal or non-verbal guidance. In the work of Saunders et al. (2006) a human guides a robot to hierarchically

construct memory models of the desired task. This incremental learning method, inspired on the behavior of social animals, allows to combine different competences to create complex tasks. Some approaches (Pardowitz et al., 2006, 2005) are based on constructing a task graph that leads to more general behaviors. Kulic generated whole-body motion using factorial HMM, that encodes and clusters a set of incrementally learned movement primitives that can be combined to generate different behaviors (Kulić et al., 2008; Kulic and Nakamura, 2009).

In the process of learning, as psychological and biological studies support for animals and humans, robots should be able to innovate new behavioral solutions, that fit their constraints, to behave more efficiently(Whiten et al., 2009). In this regard, RL is a good framework to innovate behaviors, since we can construct a reward landscape such that some search mechanism could explore for better actions in the neighborhood of demonstrations (Sutton and Barto, 1998; Peters and Schaal, 2008a,c). However, a framework that allows a robot to use a set of reward functions to acquire skills demonstrated by a human with a different kinematic structure and physical constraints has not yet been proposed (Sutton and Barto, 1998).

Mixing imitation learning with RL produces a set of benefits as Barrios claims (Barrios-Aranibar et al., 2008). It diminishes the computational time of convergence, since the search space is reduced. The innovation is based on actions that the robot has observed, so it is easier to improve this behavior. Furthermore, RL algorithms do not need to have the states and actions defined a priori. Some works combine LfD with RL, to teach the humanoid robot a constrained task of placing a cylinder in a box (Guenter et al., 2007) or to teach a robot how to hit a baseball (Peters and Schaal, 2008c). Once the skill was learnt based on kinesthetic demonstrations, the robot was able to learn to avoid an obstacle in the environment using an actor critic algorithm (Peters and Schaal, 2008b).

There are many approaches ranging from hard control methods to those relying on learning. The work of Sentis (Sentis and Khatib, 2005; Sentis et al., 2010) is centered in the operational space approach (Khatib, 1987) to hierarchically control a constrained humanoid robot. Other approaches are the Optimal Gradient Method (Nagasaka et al., 1999), the Dynamics Filter (Yamane and Nakamura, 2003) or the Cart-Table (Kajita et al., 2003a). Furthermore, there are other learning approaches like (Inaba et al., 1996).

In some sense our work is similar to that of (Morimoto and Doya, 2001), where a inverted triple pendulum learns to stand up using TD-learning. Instead of using TD-learning, we used a combination of LfD and Skill Innovation.

We performed the same task as Mistry et al. (2010b) but in a completely different way. In their work a full-size humanoid robot stand up from a chair using different strategies, imitating a young and an elder person. Their approach is based on mimicking, they adapt the human trajectories to the robot structure. By contrast our approach is more general. We are able to transfer the stand up behavior to a

robot much smaller than the human with a kinematic structure, weight and height completely different.

Our work is somewhat similar to (Billard et al., 2006; Calinon, 2009) that used HMM to recognize and generate motion patterns. They addressed the question of what to imitate and how to imitate. First, they encoded the demonstration in a HMM that are treated with Bayesian Information Criterion (BIC) to optimize the number of model states. They defined a metric in the form of a cost function to evaluate the robot's performance. Finally, they optimized the reproduction of the task in another context. The key differences between Billard and our work are that they used kinesthetic information, instead of transferring the behavior from several humans to a small robot. Moreover, the behavior is more complex in our case, since it has the problem of handling stability. However, the clearest difference is the selection of the cost function. They used a cost function that takes into account the joint trajectories. On the contrary, we constructed our reward function taking into account the stability and the effort, so that the robot and the human have different joint trajectories in their successful standing up behaviors.

An interesting approach that have synergies with our work is the concept of goal oriented behavior understanding (Schaal et al., 2003; Billard et al., 2004; Jamone et al., 2012). This field studies the recognition and posterior imitation of other agent's behavior. Aksoy et al. (2011) present a method of understanding a manipulation behavior using graphs. Similarly to us, they define a transition matrix of semantic events that allows to understand a behavior and reproduce it under different conditions. Takahashi et al. (2010) present a multi-agent behavior imitation procedure based on RL. Their method can be divided in two phases. First, they recognize an observed behavior through the estimation of the state and action reward, encoding it as a state value function. Afterwards, the imitator develops a similar behavior optimizing a reward function which is a weighted combination of the imitator reward and the teacher reward. This work is similar to ours in the sense that they used a reward profile as a basis of behavior comparison.

Argall et al. (2008) presented a combination of LfD and teacher advice that is used to improve the policy in the continuous space. Similarly, Bentivegna et al. (2004) made a humanoid robot to learn from observation a set of tasks using a library of manually predefined primitives. The performance of the robot is improved through practice based on observations of the teacher's outcomes.

A number of other approaches use a framework based on Ask for Help to speed up and enhance learning. Here, an agent request advice for other similar agents which are combined with information of the environment (Oliveira and Nunes, 2004; Alissandrakis et al., 2004). A similar approach called Active Learning has drawn attention lately in the research community (Lopes et al., 2009; Cakmak and Thomaz, 2012; Gonzalez-Pacheco et al., 2013, 2014). The idea of Active Learning is to improve learning rates by giving the learner more control over what examples it receives.

Our work is partially inspired by a framework to perform imitation by solving the correspondence problem (Alissandrakis et al., 2002, 2007, 2011). They defined three metrics for imitation: *end-point level*, where only the overall goal is considered, *trajectory level*, where the imitator considers a set of subgoals that are sequentially reached, and finally *path level*, where the imitator attempts to replicate the teacher's trajectory as closely as possible. Trajectory level and path level metric is similar to program level and action level of Byrne (2003). The method for imitation we present in this chapter is based on the trajectory level of Alissandrakis et al. (2002). Instead of using a set of sub-states as the metric of the imitation, we used the reward of the state, which is our basis of evaluation. Furthermore, our method not only imitates but innovates new behaviors, which are evaluated producing an improvement of the demonstrator behavior.

Another solution to the problem of what to imitate is presented by Billard et al. (2004). Similar to our approach, they establish a metric to evaluate the performance of the imitation process, paying attention to the manipulation task of writing the letters A, B, C and D. This metric is divided into three levels of imitation of Alissandrakis and a mimic metric which reproduces the exact trajectory of the robot. They optimized the robot control signal to minimize this four metrics which are expressed as costs functions (Alissandrakis et al., 2002).

A recent approach addressed how to obtain a model of the locomotion behavior that can be transferred form a human demonstrator to a robot, called *inverse optimal control* (Mombaur et al., 2010). The authors selected an objective function which is a linear combination of position, velocity and other features of the movement as the metric. The parameters are obtained through optimization. This model can be transfered to the robot to produce a similar behavior. The differences with our approach are: first, the selection of the metrics to optimize, that in our case is a combination of a reward function of stability and effort, and second, they used the model of the human to compute the humanoid locomotion, which produce a similar movement, whereas in our case, we use two 3-link kinematic chain models of different dimensions which perform drastically different trajectories to render in similar optimality of standing up behaviors. Other similar approaches can be found in (Ratliff et al., 2009; Kalakrishnan et al., 2013).

Our approach is based on the work of Nanayakkara where a policy improvement method is used over a large number of machine operators to improve their expertise and enhance their skills in a global way (Nanayakkara et al., 2009, 2007). He demonstrated that individuals innovate better skills while mixing their behavior with that of an elite individual, producing new elite members with better skills.

3.2 Imitation of a Single Human Behavior

In this section we explain the process of how the robot performs a meta-stable postural movement based on human demonstrations.

Since a human and a humanoid robot are morphologically similar, the optimality criteria guiding the behavioral policies should be comparable. In (González-Fierro et al., 2013a, 2014a), we used this premise to make a humanoid robot learn from an array of demonstrations performed by a group of 8 human participants, in the task of standing up from a seated posture to an upright posture. In the case of the robot, both postures were manually selected. The initial posture was achieved by manually placing the robot on a small chair. The height of the chair was selected to make sure the robot does not exceed the maximum torques and the seated posture is such that the ZMP is a little outside the sole of the feet. The final posture was a stable upright stand-up position. These postures can be changed as long as the initial posture do not surpass the maximum torque and the final posture is stable.

To perform the robot standing up maneuver and solve the correspondence problem, we cannot compare directly the position or the torques of the human, since in our case the anthropomorphic difference between them is significant (the humanoid height is 60cm). Instead, we defined a reward function as a metrics that evaluates the optimality of the overall goal, similarly to the trajectory level of Alissandrakis et al. (2002). This reward function takes sense given the initial and final conditions of the movement, which are known a priori.

3.2.1 Imitation through Reward Profile Comparison

Robot Learning from Demonstration (LfD), also called Robot Programming by Demonstration or Imitation Learning, is a powerful method to reduce the space of solutions and accelerate the learning process. LfD is a natural way to interact with the robot since it does not require an expert teacher. Furthermore, in contrast to slow reinforcement learning or trial-and-error learning methods, it can easily find a good solution from the observed demonstrations (local optima). Further information on LfD can be found in (Argall et al., 2009; Billard et al., 2008; Schaal and Atkeson, 2010).

We address two main challenges when using human demonstrations to train a humanoid robot. First, the robot is of a much smaller kinematic structure compared to the human demonstrators, while being limited by different actuator constraints, causing a correspondence problem (Schaal et al., 2003). Second, we noticed that the demonstrations performed by a group of 8 human participants were variable across trials of a given participant and across the average behaviors of individuals.

In (González-Fierro et al., 2013a), we proposed a novel method where a humanoid robot learns to stand up from human demonstrations. The human and the robot may be very different in terms of height and weight. However, since they are

anthropomorphically similar, their behavior must be the same even if they perform the same movement. Recent advances in neuroscience (Metta et al., 2006) suggest that what humans copy when imitating is the overall goal of the behavior, not just the trajectories of the movement.

In that sense, we propose a method where a multi-objective reward function is used to transfer a behavior between a human and a robot. This reward function is the basis of comparison between them. We used the Kullback-Liebler difference between the human reward and the robot reward, while taking into account the ZMP, torque and joint limit constraints.

We defined a fitness function to minimize, as the summation in every time step k of the Kullback-Liebler divergence between the mean reward profile of all the human participants $p(i)$ and the reward profile of the robot $q(i)$. Furthermore, we added as a constraints the ZMP limits, torque limits and joint limits.

$$\min \sum_{k} \sum_{i} p(i) log \frac{p(i)}{q(i)} \tag{3.1}$$

subject to

$$\theta_{min} \leq \theta \leq \theta_{max} \tag{3.2}$$

where θ represents ZMP, torque or joint position.

We demonstrated that the answer to *what to imitate* question (Schaal et al., 2003) may be to imitate the overall goal of the behavior, defined as a reward profile.

3.2.2 Behavior Prediction through Reward Transition Probability Matrix

In (González-Fierro et al., 2014a), which is an evolution of the previous work (González-Fierro et al., 2013a), we addressed the problem of imitation and innovation learning in a task of standing up from a chair. This particular posture control task takes the human body from one stable posture (seated) to the other (standing) through a series of unstable postures. In this task, a robot would benefit from demonstrations to avoid excessive activation of joints to reach the second stable posture.

We used the idea of solving the correspondence problem by finding a common constrained reward domain for the behavior of both the humans and the robot. Once a set of reward functions have been identified, we approach to solve the problem by taking a stochastic approach to model the reward transition probability distribution for all demonstrated trials by all participants. More specifically, we construct a Markov chain using the discretized reward profiles for each demonstrated trial.

The advantage of a Markov chain to model human demonstrations in this manner is that we can use a particular reward value, obtained by the robot at any given time,

to predict the future reward it would obtain if it follows a policy similar to humans in a qualitative sense. Errors of such predictions can be used as feedback signals to update the policy of the robot. Another advantage of the Markov chain is that we can find a single state transition probability distribution, or a transition matrix, that summarizes the dominant intrinsic structure of demonstrations performed by many individuals subject to variability.

The main difference between the work presented in (González-Fierro et al., 2013a) and the work presented in (González-Fierro et al., 2014a) is the way we used the reward. In (González-Fierro et al., 2013a) we compare the robot reward with the average human reward. However, in (González-Fierro et al., 2013a), a step forward was performed. The stochasticity of the human demonstrations is encoded in a Reward Transition Probability Matrix (RTPM) in terms of the reward. The RTPM encodes the transition in the pursuit of the behavior goal, which is equivalent to say that RTPM is the strategy the human use to perform the behavior. We assume the strategy of the human and the robot is the same, since they are performing the same behavior. So using the RTPM the robot can predict its future reward like if it behaves like a human.

We captured data from 8 human participants performing 20 consecutive demonstrations each, using a Qualisys motion capturing system. Modeling the humans as an actuated 3-link kinematic chain, we computed a reward function of stability and effort which we used as a common basis of comparison with the humanoid robot. Using Markov chains theory, we summarized the performance of all human participants in a transition probability matrix of the scalar reward that defines the optimality of the behavior. Using Differential Evolution (DE) (Storn and Price, 1997), we optimized the robot joint trajectory to minimize the difference between the predicted reward and the real reward achieved by human demonstrators, subject to the constraints of ZMP limits, maximum and minimum torque, and joint limits. Once this first stage of imitative learning is accomplished, we proceed to explore new solutions with better rewards in the neighborhood of human-like movement subject to constraints, which we call skill innovation. Figure 3.1 shows the summary of the algorithm discussed in this paper.

Markov Decision Process

A learning process that consider the Markov property to predict actions and states is called Markov Decision Process (MDP) (Nanayakkara et al., 2009).

The learning process is defined as a sequence of finite states s and actions a pairs that produce an associate reward $r : s \to \mathbb{R}$. The agent, starting from a state $s(t)$ will compute an action $a(t)$ to reach a future state $s(t+1)$, obtaining a reward $r(t)$, which can be defined as a set of values or as a mathematical function, it is usually called *the reward function*.

The key feature of the rewarded based learning process is the estimation and

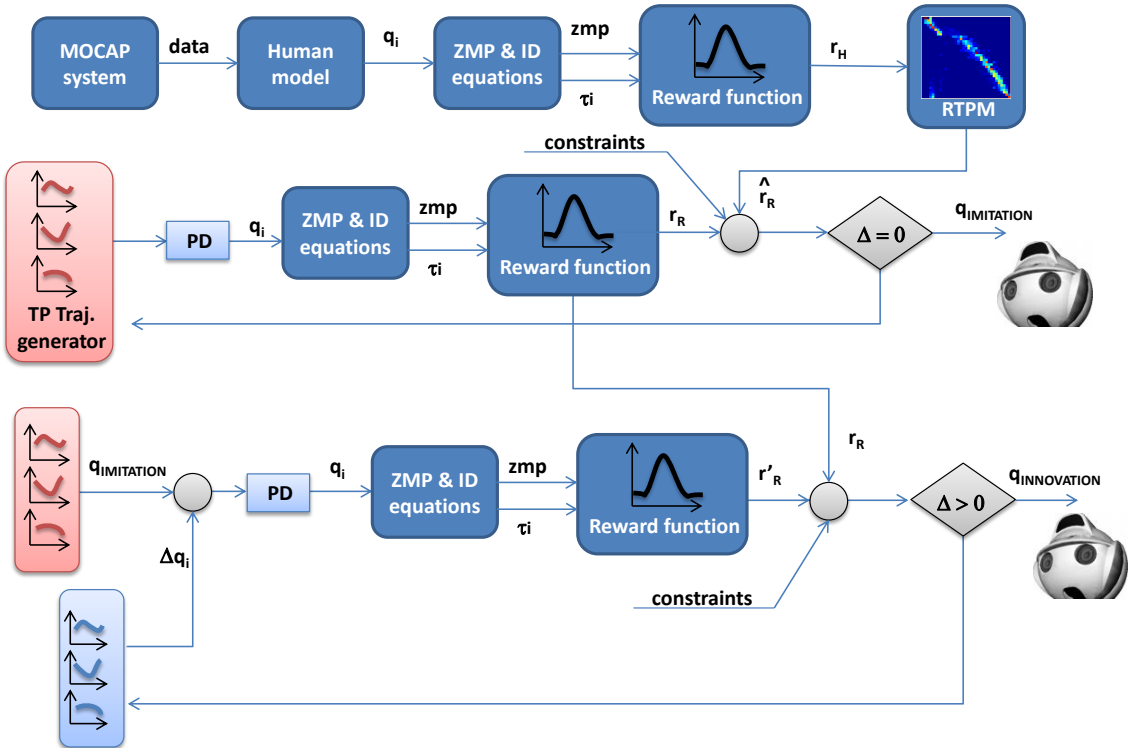


Figure 3.1: *Overview of the algorithm. We collected data from a MOCAP system and model the human as an actuated 3-link kinematic chain, where q_i are the joint positions. After computing the ZMP and joint torque τ_i, we define a reward function for the human r_H. This is done for all 160 demonstrations of standing up from a chair. Then, the Reward Transition Probability Matrix (RTPM) is obtained. Using Differential Evolution, we generate a new triple pendulum articular trajectory and obtain the reward profile r_R. This profile is compared with the predicted reward profile if the robot behaves like a human $\hat{r}_R$. The optimization process ends when the difference is small producing the imitation solution. Furthermore, we perturb the imitation solution Δq_i and compute a new reward profile r'_R that is compared with the imitation reward (r_R). The optimization process ends when the imitation reward is higher than the innovation reward, producing the innovation solution.*

maximization of the total future expected reward, given a policy $\pi : s \rightarrow a$ that maps states to actions. This estimation is known as the *state-value function* or simply *value function*, given by

$$V^{\pi}(s) = E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^{k} r(t+k) | s(t) = s\right\} \tag{3.3}$$

where $\gamma \in (0, 1)$ is called the discount factor.

Similarly, we can define an *action-value function*. In the state-value function, we assume that at each state we choose the action that will maximize the total future reward, because we know the policy π. Here, we consider the expected total future reward for each action given a state. The action-value function is defined as

$$Q^{\pi}(s, a) = E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^{k} r(t+k) | s(t) = s, a(t) = a\right\} \tag{3.4}$$

There are two key issues that are addressed from this point forward. The first one is the maximization of the value function, finding an optimal policy that solves the Bellman equation (Sutton and Barto, 1998). Many solutions have been proposed to solve the problem of reinforcement learning. Temporal difference learning (Sutton, 1988), actor-critic methods (Peters et al., 2003; Peters and Schaal, 2008b), Q-learning (Watkins and Dayan, 1992), Monte Carlo methods (Michie and Chambers, 1968) or, as our approach, using genetic algorithms (Goldberg, 1989). For further information, please refer to (Sutton and Barto, 1998; Schaal and Atkeson, 2010).

The second key issue is the selection of the reward function (Mataric, 1994), that can be selected by the teacher (Bentivegna et al., 2004), mathematically defined (Nanayakkara et al., 2007; González-Fierro et al., 2014a) or directly learned from demonstrations (Abbeel and Ng, 2004; Lopes et al., 2009).

Markov Chains and Transition Matrix

A Markov chain is a random process that can define the behavior of a dynamical system under the Markov property. This property assumes that a state has all the required information to make a decision about the future.

A first order Markov chain is defined as a series of random variables or states $s_1, ..., s_N$ where

$$P(s_{N+1} | s_1, ..., s_N) = P(s_{N+1} | s_N) \tag{3.5}$$

By using a Markov chain to explain a behavior, the next state in a sequence can be predicted. The state distribution will depend only on the previous state and will be independent of all early states. In other words, the defining characteristic of a

Markov chain is that its future trajectories depend on its present and its past only through the current value (Medhi, 2010).

For nth and *(n+1)*st trials, if a state s_N has an outcome j (i.e. $s_N = j$) and $s_{N+1} = k$, the transition probability associated with both trials is $P(s_{N+1} = k|s_N = j) = p_{jk}$.

We can specify a Markov chain given the initial probability distribution $P(s_1)$ and the conditional probabilities in the form of a Transition Probability Matrix or *Markov Transition Matrix* T, where T is a stochastic matrix, i. e. it satisfies that every row is a probability distribution and it is a square matrix with non-negative elements[1].

$$p_{jk} \geq 0, \ \sum_j p_{jk} = 1 \text{ for all } j \tag{3.6}$$

This matrix may be written in the form

$$T = \begin{pmatrix} p_{11} & p_{12} & p_{13} & \cdots \\ p_{21} & p_{22} & p_{23} & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{pmatrix} \tag{3.7}$$

The transition probabilities and the Transition Matrix are defined for a unit-step transition, however, we can define a m-step transition in the future. The m-step transition probability is defined by

$$P(s_{N+M} = k|s_N = j) = p_{jk}^M \tag{3.8}$$

and the m-step Transition Matrix is denoted by T^M.

Thus, the probability of a state with m-steps in the future can be denoted by

$$P(s_{N+M}) = P(s_N)T^M \tag{3.9}$$

In Algorithm 1 the computation of the transition matrix is presented.

Extracting a Stochastic Template from Human Demonstrations

This section presents the process of extracting a stochastic model of the human behavior, that will be transferred to the robot. This behavior is presented in the form of a Reward Transition Probability Matrix, that can be used to compute an optimal robotic behavior learned based on human demonstrations.

Using the transition matrix we can predict the probability of the future reward of the human.

$$P_{human}(k+n) = P_{human}(k)T_{human}^n \tag{3.10}$$

[1]The proof of the general existence of the Markov chain can be found in (Chung, 1967)

Algorithm 1 TRANSITION PROBABILITY MATRIX COMPUTATION

```
Create a 1-D array R of size 1 by N from the minimum to maximum possible value in the trajectory
Create an N by N null matrix T.
for all trial = 1 : t do
    for all SamplingStep = 1 : s do
        Calculate the trajectory and find its bin in the 1-D array R.
        Store the bin number in the array B(SamplingStep)
        if SamplingStep > 1 then
            T(B(SamplingStep − 1), B(SamplingStep))+ = 1
        end if
    end for
end for
Normalize each raw of T so that the sum of each raw adds up to 1.
```

where $P_{human}(k)$ is the probability in step k, $P_{human}(k+n)$ is the probability in N steps in the future and T_{human} is the transition matrix.

If the robot is going to behave like the human, we can suppose that their transition matrices are the same:

$$T_{human} = T_{robot} = T \tag{3.11}$$

So if we know the reward probability in step k, we can predict the future probability in step $k+n$.

$$P_{robot}(k+n) = P_{robot}(k)T^n_{robot} = P_{robot}(k)T^n \tag{3.12}$$

The fitness function is defined as the difference between the predicted reward of the robot if it behaves like a human and the actual reward, under some constraints. The predicted reward is obtained as the expected value of the probability in (3.10), and it is given by

$$\hat{r}_R(k+n) = E[P_{robot}(k)T^n] \tag{3.13}$$

Furthermore, we added the ZMP limits, torque limits and joint limits as constraints.

We defined the fitness function as:

$$\min \sum_{k=1}^{N-1} (\hat{r}_R(k+1) - r_R(k+1))^2 \tag{3.14}$$

$$\theta_{min} \leq \theta \leq \theta_{max} \tag{3.15}$$

where θ represents ZMP, torque or joint position.

In Algorithm 2, an outline of the imitation process is presented. This algorithm can be easily implemented minimizing the fitness function (3.14) using DE.

Algorithm 2 IMITATION LEARNING

Initialize: Identification of human and robot model
Initialize: K_p, K_d
Compute ZMP_H for all trials of every human
Compute τ_H for all trials of every human
Compute r_H for all trials of every human
Calculate the RTPM T
while $F \neq 0$ **do**
    New q_{m1}, q_{m2}, q_{m3} using Differential Evolution
    Compute piecewise polynomial
    Simulate system using K_p, K_d
    Compute $ZMP(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$
    Compute $ID(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$
    Compute the reward $\hat{r}_R$
    Compute the estimated reward r_R using (3.13)
    Compute fitness function F using (3.14)
end while

3.3 Innovation of a Single Human Behavior

Similar to how a child would try to improve learned behaviors based on demonstrations, often known as *over-imitation*, a robot could use heuristic search algorithms to explore the reward landscape for better behavioral policies in the neighborhood of the policies acquired based on demonstrations (Whiten et al., 2009; Nielsen et al., 2014). In this section, we will discuss how we achieved this, based on the demonstrations of the 8 human participants.

3.3.1 Innovative Solution to Stand Up Process

We can compute a new desired joint trajectory as a perturbation of the imitation trajectory $\mathbf{q}_{innovation} = \mathbf{q}_{imitation} + \Delta\mathbf{q}$ and maximize the difference between the innovation reward and imitation reward, while fitting the constraints.

The new fitness function maximizes the positive difference between the innovation reward r'_R and imitation reward r_R. It is given by

$$\min \sum_{k=1}^{N} e^{-\mu(r'_R(k) - r_R(k))} \tag{3.16}$$

$$\theta_{min} \leq \theta \leq \theta_{max} \tag{3.17}$$

where θ represents ZMP, torque or joint position.

The Algorithm 3 presents the innovation behavior.

Algorithm 3 SKILL INNOVATION

Initialize: Identification of human and robot model
Initialize: K_p, K_d
Initialize: $\mathbf{q}_{imitation}$
Initialize: Compute the imitation reward r_R
1: **while** $f \neq 0$ **do**
2: New $\Delta q_{m1}, \Delta q_{m2}, \Delta q_{m3}$ using DE
3: Compute piecewise polynomial
4: Simulate system using K_p, K_d
5: Compute $ZMP(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$
6: Compute $ID(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$
7: Compute the reward r'_R
8: Compute fitness function f using (3.16)
9: **end while**

3.3.2 Imitative and Innovative Learning

Imitation and innovation process during learning is a complex process that can be formulated together using a simple variation of (3.14) and (3.16).

$$\min\ (1-\rho)\sum_{k=1}^{N-1}(\hat{r}_R(k+1) - r_R(k+1))^2 + \rho\sum_{k=1}^{N}\frac{1}{r_R(k)} \tag{3.18}$$

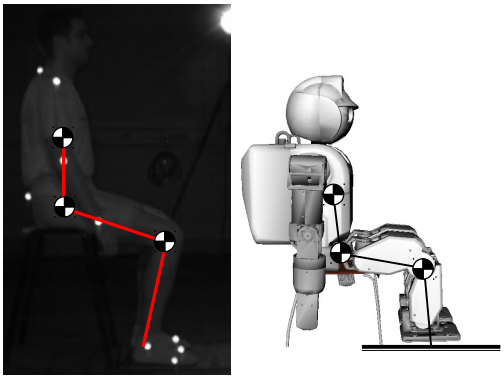
$$\theta_{min} \leq \theta \leq \theta_{max} \tag{3.19}$$

where θ represents ZMP, torque or joint position.

The first term of (3.18) represents the imitation part, the second part represents the innovation, tuned by the term $\rho \in (0, 1)$. The third term corresponds to the ZMP, torque and joint limits constraints. If we are looking for more innovation, we just need to adjust the term ρ.

3.4 Behavior Representation in the Reward Domain

In this section we develop the mathematical tools that allow to transfer the behavior from the human to the robot. We modeled both human and robot as an actuated 3-link kinematic chain. Next, we obtain the ZMP and torque of every link trajectory to compute a common basis of comparison, the reward domain.



(a) (b)

Figure 3.2: *(a) Snapshot of the high frequency camera of the MOCAP system with a subject seated on a chair and markers on his body. Actuated 3-link kinematic chain is overlaid. (b) A simulation of the humanoid HOAP-3 seated on a chair and the 3-link kinematic chain.*

3.4.1 Kinematic Model

We approximated both the humans and the robot using an actuated 3-link kinematic chain in the sagital plane, to represent the third scheme of robotic imitation, the mapping between dissimilar bodies (Schaal et al., 2003), since the human standing up movement occurs in that plane without relative movement of legs. It should be noted that this does not account for the role of the swing of hands during standing up.

Figure 3.2(a) shows a snapshot of the high frequency camera of the MOCAP system, where a human is seated on a chair with all the markers on his body. An actuated 3-link kinematic chain is overlaid. In Figure 3.2(b) we show the position of the 3-link actuated kinematic chain over the humanoid robot. The center of mass of each link in the kinematic chain is located at its tip. The first joint of the kinematic chain corresponds to the ankle joint in both human and humanoid, the second joint of the kinematic chain corresponds to the knee and the third one corresponds to the hip.

It is clear that a 3-link kinematic chain does not represent completely the behavior of a humanoid robot in every situation, however the model has some advantages that make it suitable for the task of standing up form a chair. We chose a 3-link kinematic chain with continuous boundary conditions (starting from static torques needed to keep balance soon after lift-off from the chair) to represent a human and robot standing up due to the model suitability for this task. It is a simple model with easy to solve equations, with low computational cost, which is an advantage to use an optimization process like is our case. The task involves relatively low velocity

and acceleration profiles. Motor tasks like standing up in healthy adults is a subconscious process with minimum cognitive inhibition. Furthermore, the movement is symmetric (legs do not move relative to each other). Therefore a kinematic chain in the sagittal plane is suitable to represent the movement. Barin (1989) demonstrated the relevance of using an inverted pendulum structure to model the human postural dynamics in the sagittal plane.

In a previous work (González-Fierro et al., 2013e), we discussed the advantages and limitations of using a reduced model and how the performance can be improved using a robust control technique like Fractional Calculus. It should be noted that using a more complete model like in the work of Sentis and Khatib (2005) would reduce real time feedback control effort but it would not affect the application or results of our method. A complete model would require a more complex formulation with higher computational cost and room for errors due to wrong estimation of an increased number of parameters.

Our motion data shows that there is no slip between the foot and the ground in the human demonstrations, therefore we assume that the friction coefficient was high enough for the reaction force vector to stay within the friction cone with no slip. Furthermore, for simplicity we did not model the contact with the chair when the human is seated.

Human Kinematic Model and Simplifications

To calculate the masses of the actuated 3-link kinematic chain for the human, we took into account the total weight of the subject and a estimate of the mean distribution of human body parts presented by NASA (1995). The mass of the first link is composed by the masses of the foot and calf, the mass of the second link is composed by the masses of the thighs, hip flap and pelvis and finally, the mass of the third link is composed by the rest of the body masses. In B the mass distribution of an average human body is shown.

The length of the links is estimated using the distance between markers (see Figure 3.10). For the first link, the length is the distance between ankle and knee, for the second one, the distance between knee and hip and for the third one, the distance between the hip and the middle of the chest.

An actuated 3-link kinematic chain was computed for every subject, so we obtained a total of 8 kinematic chains and 20 trajectories of standing up for every human demonstrator.

Robot Kinematics Identification

To identify the actuated 3-link kinematic chain parameters of the Fujitsu HOAP-3 humanoid robot, i.e. the length and mass of every link, we used Differential

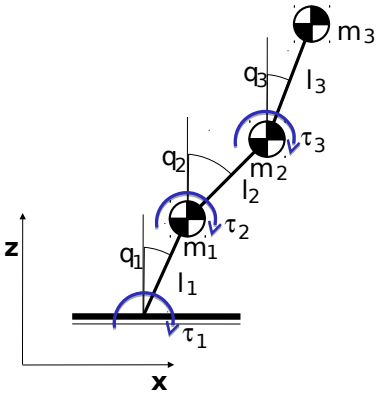


Figure 3.3: *An actuated 3-link kinematic chain in the sagittal plane.*

Evolution (Storn and Price, 1997) and data of the robot sensors. This method is based on the work of Tang et al. (2008).

We manually planned several stand up trajectories for the robot and obtained the ZMP measurement of the force sensors in the feet. Later, we used the ZMP multi-body equation (2.55) to obtain the theoretical ZMP trajectory in the saggital plane.

To identify the system we optimized the kinematic chain parameters minimizing the difference between the theoretical ZMP and the real ZMP. The results are shown in the table 3.1.

Table 3.1: *Parameter identification of the 3-link kinematic chain for the robot.*

	Mass (g)	Lenght (m)
Link 1	505	0.167
Link 2	500	0.260
Link 3	3900	0.264

The trajectories to identify the system were planned performing a stand up movement starting form seated (Figure 3.4). At first the robot seems unstable because its ZMP is slightly outside the limits. Actually it is not, the robot is seated on a small chair. Since we do not take into account the contacts with this chair, both theoretical and real ZMP are outside the limits. However, the robot at this moment has three contacts, the chair and both feet. When the movement starts, the robot rapidly reaches stability.

3.4.2 Trajectory Generation

The use of cubic splines to define the desired trajectory is very common in robotics (Lin et al., 1983). Usually, the spline is optimized to avoid roughness and to pass a

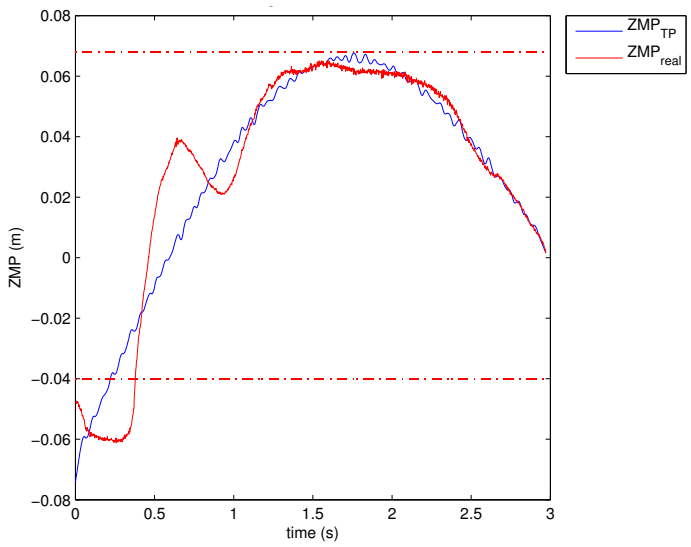


Figure 3.4: *An example of a theoretical vs real ZMP trajectory used in the parameter identification. The limits of the ZMP is showed in dotted red.*

set of constraints.

The cubic spline is defined as a piecewise polynomial fitted to a set of via points

$$(t_0, q_0^*), (t_1, q_1^*)...(t_k, q_k^*) \tag{3.20}$$

where $q_i^* \in \mathbb{R}^N$ is the joint via points at time $t_i \in \mathbb{R}$.

Given these via points, there is a cubic trajectory that passes through these points and satisfy a smooth criteria.

$$q_i(t) = a_i(t - t_i)^3 + b_i(t - t_i)^2 + c_i(t - t_i) + d_i \tag{3.21}$$

where a_i, b_i, c_i, d_i are the polynomial coefficients optimized. The complete joint trajectory $q(t) \in \mathbb{R}^N$ is a concatenation of (4.24) over the time intervals.

$$q(t) = \begin{cases} q_0(t) & \text{if } t_0 \leq t < t_1 \\ \vdots & \\ q_k(t) & \text{if } t_{k-1} \leq t < t_k \end{cases} \tag{3.22}$$

To perform the imitation, a desired joint trajectory for the robot is defined and computed as a cubic spline with one via point with an initial, middle and final point. The initial and final points correspond to the static postures of being seated and being standing up and are known. The middle point is optimized using DE to obtain the imitative and innovative behavior, using (3.14) and (3.16) respectively.

It is important to highlight that there is no dynamic control. We use the data obtained via the MOCAP system to compute a stochastic model of the human

behavior. The model is transferred to the humanoid robot by computing an optimal trajectory, which imitates the human fitting all humanoid constraints.

The trajectory optimization is computed offline at the moment, because the robot has to learn the average behavior demonstrated by a group of humans. However, Differential Evolution can use individual demonstrations in a pool of references in an online learning framework.

3.4.3 Definition of Reward Profile

Through human demonstrations, the humanoid robot learns how to imitate the human performance, taking into account the robot constraints. Furthermore, it is able to improve the imitation, obtaining a better solution than the one demonstrated by the human.

For this purpose, we defined a reward function of stability and effort for all human participants, which are modeled as 3-link kinematic chains. To check the stability, we used the ZMP and to check the effort, the torques of the three joints.

To compute the inverse dynamics we used (2.34). To compute the ZMP for the 3-link kinematic chain we used the equation of multibody ZMP in the sagital plane (2.55).

In Figure 3.5 is plotted the ZMP profile for all 20 demonstrations of one human participant, whose weight is 68,3 Kg and height is 179,6 cm. As it can be observed, at the beginning the ZMP is outside the limits because we do not model the contact with the chair. The ZMP limits are obtained by measuring the feet size of all subjects with the data provided in the MOCAP system. We took as the stable zone the mean of the feet measurements in every demonstration. As it can be seen, not all trajectories have the same duration, since not every demonstration is equal. To solve this problem we took the slowest movement as the basis and stretch the other trajectories as if the human is still.

Figure 3.6 shows the joint torques for the same participant. Since we can not measure the maximum torque that the muscles support, for simplicity, we took as the maximum value the maximum torque of the 20 demonstrations. This value will be used in the definition of the reward function.

Selection of the Reward Function

We used two different functions to evaluate the human behavior in the reward space: a polynomial and a gaussian-like function. Every function is used to obtain the ZMP reward profile and the torque reward profile. θ_j represents the ZMP minimum, medium and maximum in the case of the ZMP reward function and similarly with the torque reward function. The torque reward function is the normalized mean of the 3-link kinematic chain's joint torques.

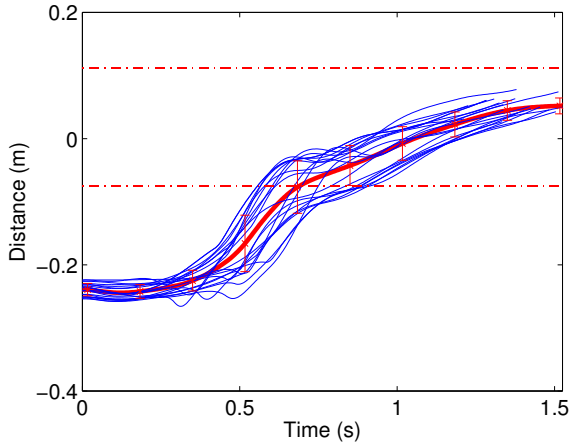


Figure 3.5: *Actuated 3-link kinematic chain's ZMP profile for the 20 demonstrations of one of the subjects standing up. In red are the mean and standard deviation. In dotted red the ZMP limits.*

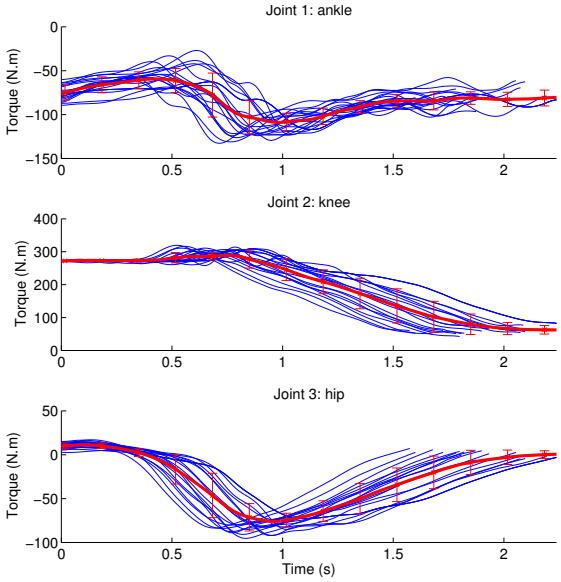


Figure 3.6: *Actuated 3-link kinematic chain's torques of the 20 demonstrations of one of the subjects standing up. The first joint of the 3-link kinematic chain corresponds to the human ankle, the second one to the knee and the third one to the hip. In red are the mean and standard deviation.*

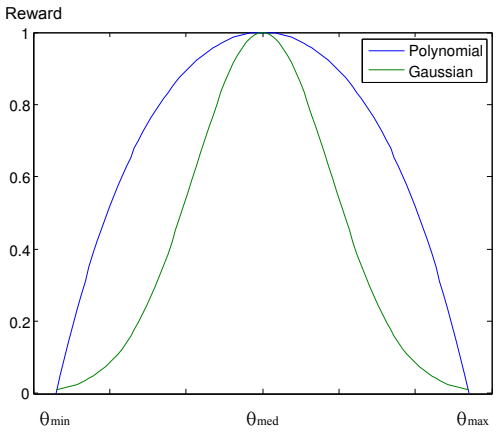


Figure 3.7: *Two reward functions. The blue one is a polynomial function of 4th degree, the green one is a gaussian-like function whose maximum is one. The parameter θ represent the ZMP or the torque of the actuated 3-link kinematic chain.*

The polynomial reward function has order 4 and is centered in θ_{med}. Values outside the limits have zero reward. It is given by

$$f_1(x) = ax^4 + bx^3 + cx^2 + dx + e \tag{3.23}$$

where x can be ZMP or joint torque trajectory and the coefficients a, b, c, d, e, are obtained solving

$$f(\theta_{min}) = 0;\ f(\theta_{max}) = 0;\ f(\theta_{med}) = 1;$$
$$f(\theta_{med}/2) = 0.8;\ f(3\theta_{med}/4) = 0.8 \tag{3.24}$$

The gaussian-like function follows the next equation

$$f_2(x) = \exp \frac{-36(x - \theta_{med})^2}{2(\theta_{max} - \theta_{min})^2} \tag{3.25}$$

These functions allow the mapping from the ZMP or torque space to the reward space (Figure 3.7).

The ZMP stability region and torque for the robot is obtained using the manual provided by the manufacturer. Measuring from the heel of the foot, $ZMP_{max} = 108mm$ and $ZMP_{min} = 0mm$. The maximum and minimum torque for the motors of ankle, knee and hip is $\pm 4.5N.m$, so the range of the pendulum is $\pm 9N.m$. For the human, the limits are set for simplicity as the minimum and maximum values of torque and ZMP measured in the MOCAP system.

The total reward profile for the human is the sum of stability and effort functions:

$$r_H(t) = \frac{w_{zmp}(t)r_{zmp}(t) + w_\tau(t)r_\tau(t)}{2} \tag{3.26}$$

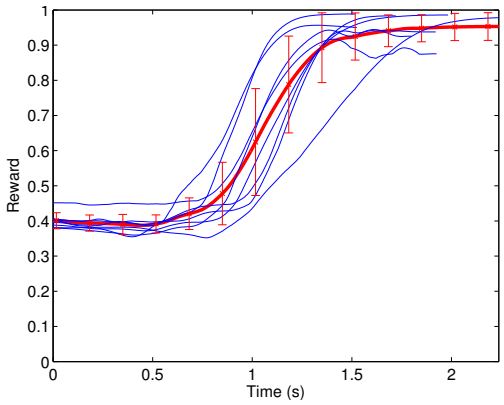


Figure 3.8: *In blue the mean reward of the 20 demonstrations of each human participant. In red are the mean and standard deviation of all demonstrations of the 8 human participants.*

where w_{zmp} and w_τ are weights of ZMP and torque respectively, that can vary from 0 to 1, r_{zmp} is the reward of the zmp and r_τ is the reward of the torque, which is the sum of the reward of every joint torque divided by 3.

Figure 3.8 shows the mean reward of the 20 demonstrations of every participant standing up (in blue). In red is plotted the mean of all 160 demonstrations and the standard deviation. For this profile we chose the following weights:

$$w_{zmp}(t) = at^3 + bt^2 + c \tag{3.27}$$

$$a = \frac{-2(\phi_1 - \phi_0)}{T^3}; \quad b = \frac{3(\phi_1 - \phi_0)}{T^2}; \quad c = \phi_0 \tag{3.28}$$

where $\phi_1 = 1$, $\phi_0 = 0$ and $T = t_{max}$

$$w_\tau(t) = 1; \tag{3.29}$$

Equations (3.27) and(3.28) represent a third order polynomial, that starts in 0 and finishes in 1, meaning that at the beginning of the stand up motion we do not care if the ZMP is outside the limits but we care about the torques (3.29).

Analyzing Figure 3.8, the predominantly subconscious operation of motor programs to execute standing up show some stereotypical pattern across all subjects irrespective of their variability in terms of weight and height.

3.4.4 Generalization and Discussion of the Reward Profile

The shape of the reward functions are selected in accordance to the task. The final posture or goal posture of standing up has the feature of being stable and of minimum effort, as we showed in a previous work(González-Fierro et al., 2013a). In that case, the ZMP is almost in the middle of the feet and the torque is minimum.

Therefore, the reward function is selected to be an attractor to these conditions. That is why the middle point of the reward function is the mean ZMP in the case of stability function and zero in the case of effort function (Figure 3.7).

The selection of a suitable reward function has been discussed extensively (Mataric, 1994). Sometimes, it is the observer who will manually set the reward value (Alissandrakis et al., 2002; Bentivegna et al., 2004); it can be defined as a mathematical function that maps from states and actions to rewards (Billard et al., 2004, 2006; Takahashi et al., 2010; Nanayakkara et al., 2007; Guenter et al., 2007) or the reward function can be learned from the demonstration set, what is called inverse reinforcement learning (Ng and Russell, 2000; Abbeel and Ng, 2004).

We used stability as a criterion because this task involves moving from a statically stable posture (seated) to an unstable fixed point (upright posture) through a process of dynamic stabilizing. This can be achieved by humanoid robots with regulator type feedback control, that leads to high peak torques at the start of the movement. In our experiments, we show that humans do not use such a regulator type feedback control. In contrast, humans use an optimum strategy in terms of effort minimization. Therefore, it is more meaningful to combine the intention to maintain stability while minimizing effort in a learning based on demonstration framework.

Our method needs to define a reward function for each task. In a complete different task, as for example opening a door, the reward function will have to account for a complete different shape. This has all the sense since the goal is completely different. In the case of opening the door, the goal is grabbing the knob successfully and pulling the door until it is open, then the reward function has to be selected to take that into account. The goal and reward function are completely different in the case of standing up from a chair where it is important to maintain stability and minimize the effort.

3.5 Experimental Results

We used the humanoid HOAP-3 to show the robustness of our method and present the experimental results.

3.5.1 Experimental Setup

We collected data from 8 human participants of age between 20 to 40 years, weights between 60 and 99 Kg, and heights between 1.68m and 1.88 m. For this task, we recruited adult healthy participants with no known history of motor dysfunction. The experimental protocol was approved by the ethics committee on using human participants in experiments of Kingston University of London. The height and weight of all human participant is presented in 3.5.4.

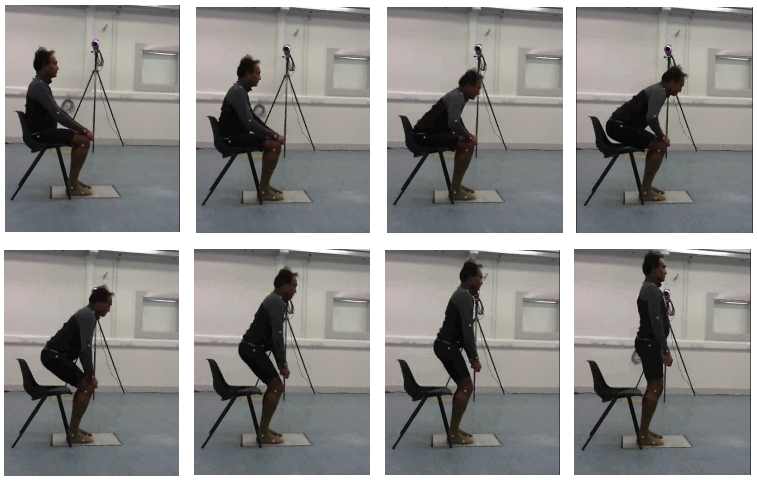


Figure 3.9: *Snapshots of one of the human participants standing up from a chair in the MOCAP environment.*

Every participant performed 20 consecutive demonstrations of standing up from a chair. There were no special training for the participants, since it is a simple task, only a few instructions like do not stand up very fast or put your feet straight. A 6-camera Oqus motion capturing system made by Qualisys, Sweden, collected position data of 21 markers attached to the subject's body at 240Hz sampling rate.

In Figure 3.9 the experimental procedure is shown. The participant is seated on a chair and performs the movement of standing up.

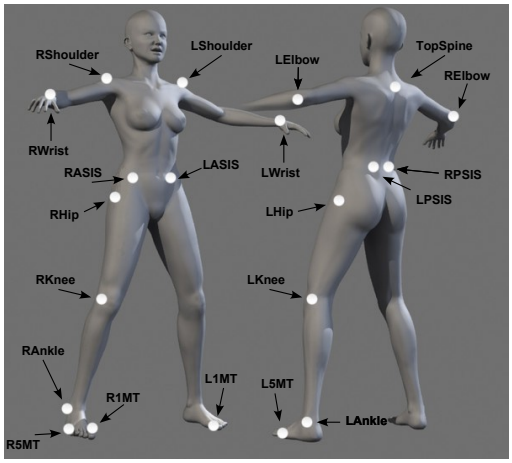


Figure 3.10: *Distribution of the 21 markers in a human body. R stands for right and L stands for left. 1MT and 5MT stands for first and fifth metatarsi respectively.*

The markers were distributed as follows: first and fifth metatarsi, lateral malleolus (ankle), lateral epicondyle of the femur (knee), greater trochanter (hip), anterior superior iliac spine (ASIS), posterior superior iliac spine (PSIS), Seventh cervical vertebra (top of spine), acromion process (shoulder), lateral epicondyle of the humerus (elbow) and lateral styloid process (wrist). All markers are bilateral, they were located on both sides of the body, except the seventh cervical vertebra. In Figure 3.10 the position of the markers is shown.

3.5.2 Extraction of Human Behavior

After computing the reward function for every trial of every human participant, and assuming that the reward fits the Markov property, we computed the Reward Transition Probability Matrix (RTPM) (Nanayakkara et al., 2009). This matrix summarizes, in one singular metric, the behavior of several human participants doing the action of standing up from a chair. Its computation is presented in Algorithm 1.

This matrix changes with the reward function we select. Therefore, we computed several RTPM depending on the reward function selected, if it is the polynomial or the gaussian, and depending of the weights selected in equation (3.26).

Figure 3.11(a) represents the RTPM using the polynomial function and the weights (3.27) and (3.29). Figure 3.11(b) represents the RTPM using the gaussian-like function (3.25) and $w_{zmp} = w_{\tau} = 1$. These matrices, Figure 3.11, represent the behavior of the human standing up taking into consideration the stability and the torques, and of course, it strongly depends on the selection of the reward function.

3.5.3 Humanoid Standing Up from a Chair

In Figure 3.12 a simulation of an actuated 3-link kinematic chain is shown performing a stand up movement in the imitation learning approach (upper snapshots) and in the innovation learning approach (lower snapshots).

As it can be seen in the Figure 3.12, the trajectory of the imitation performance has more variability than the trajectory of the innovation performance. This small variability in the innovation movement produces a smaller variation in the ZMP and a lower torque profile.

Figure 3.13(a) and Figure 3.13(b) shows the theoretical ZMP, calculated using (2.55) and the ZMP measured from the robot sensors for both imitation and innovation. This measurement is the mean of the ZMP trajectory of both feet. As it can be seen, initially the ZMP is outside the stability region. This happens because at that time the robot is slightly leaned on the chair. The ZMP in the innovation motion goes straighter to the middle, which is translated in a higher reward. The imitation ZMP profile stays also near the middle value, but not as much as the innovation profile. The explanation is simple, in the case of the imitation, the solution

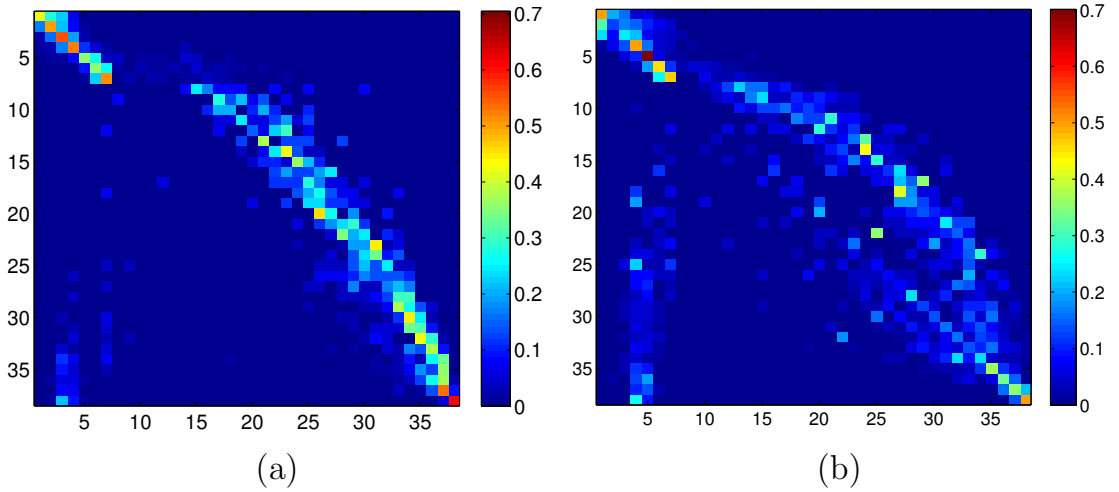


Figure 3.11: *(a) Normalized Reward Transition Probability Matrix (RTPM) for all human participants using the polynomial function* (3.23)*. (b) Normalized Reward Transition Probability Matrix (RTPM) for all human participants using the gaussian-like function* (3.25)*.*

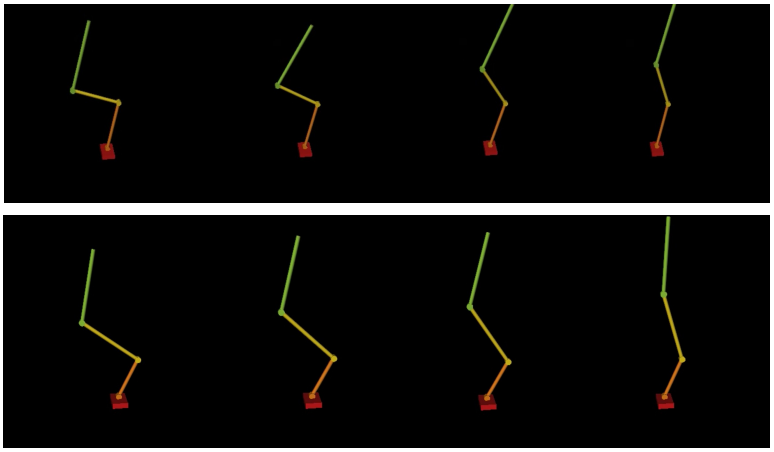


Figure 3.12: *Several snapshots of an actuated 3-link kinematic chain simulating the movement of standing up. The upper part simulates the imitation behavior while the lower part simulates the innovation. In the imitation line it can be seen that the kinematic chain lean forward while in the innovation part the movement is straighter.*

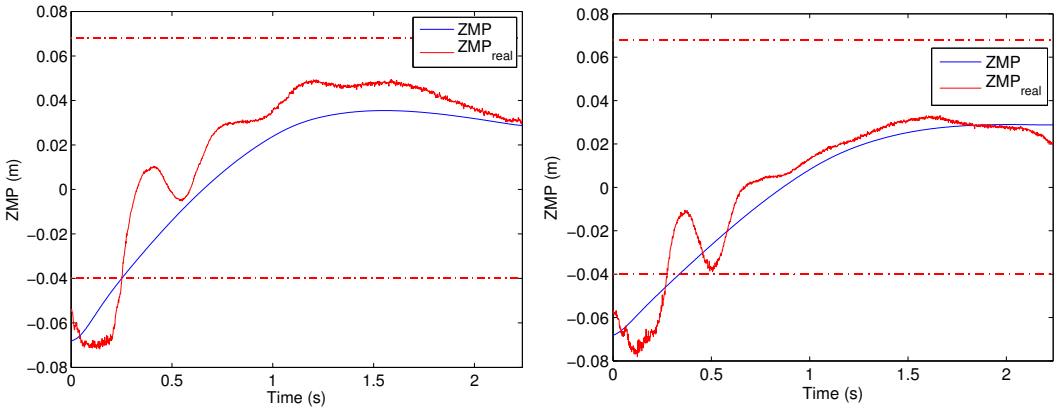


Figure 3.13: *Computed ZMP of the actuated 3-link kinematic chain approximation and that for the real robot for the imitation behavior (a) and for the innovation behavior (b).*

minimizes the difference between the predicted reward if the robot behaves like a human and the actual reward, instead, in the case of the innovation, the solution maximizes the reward, always fitting the constraints. Furthermore, as it can be noted in Figure 3.5, the ZMP of the human and those of the robot is not the same, which is obvious as their sizes and kinematic structure are diferent.

Figure 3.14(a) and Figure 3.14(b) plots the 3-link kinematic chain torques. As it can be seen, they are between the limits. It is remarkable that the second joint has the higher value, because it supports the heaviest part of the robot. Again, if we analyze the imitation and innovation torque profiles, we observe that in the imitation movement, the knee joint stays near the limit almost until the second 1. However, in the innovation movement, the reward is higher, and the torque decrease faster to a comfortable posture.

In the movement of standing up to an upright posture the torque limits play an important role. They define the initial posture. It is the same when a human stands up. If the torque that our legs have to create is too much high, we help ourselves with our hands, finding another contact or a different stand up strategy. Therefore, our method as we presented it, can cope with postural movements starting from a safe and feasible initial posture.

In Figure 3.15 the computed reward profiles for imitation and innovation behavior are plotted. Blue line represents the imitation reward, red dots represent the predicted reward if the robot behaves like a human and the green line represents the innovation reward. Comparing the robot reward with the human participants reward in Figure 3.8 we observe that they are very similar, since the predicted robot

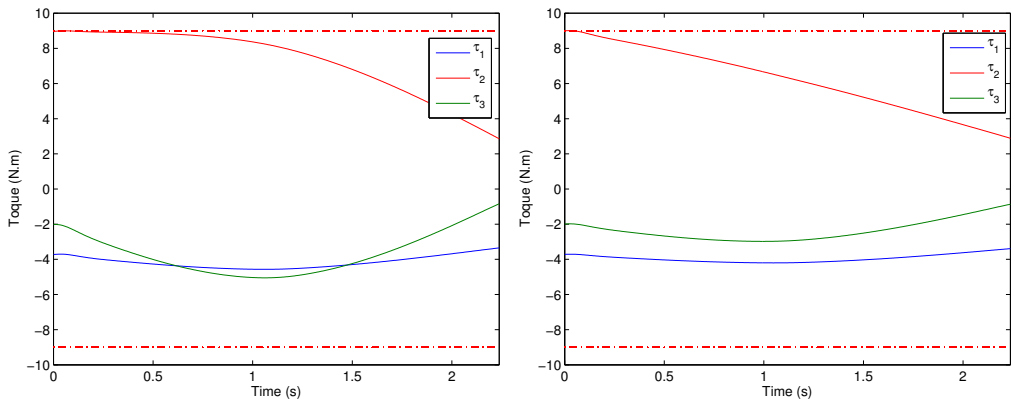


Figure 3.14: *3-link kinematic chain torques in the imitation trajectory (a) and in the skill innovation trajectory (b).*

reward is related to human reward. However it is not the same, because the variability of the different demonstrator performances is encoded in the RTPM, which is the key element to transfer a behavior.

The results presented in this section were obtained using the polynomial function that maps from ZMP and torque space to reward space. The results using the gaussian-like function (3.25) were not showed here, for reasons of space and that they would be very similar to the results of the polynomial function.

In the imitation approach (see snapshots in Figure 3.16(a)), the kinematic chain lean forward producing a movement very similar to that of the human demonstrations. In that case, the optimizer minimize the difference between the actual reward and the predicted reward if the robot behaves like a human.

In the innovation approach (see snapshots in Figure 3.16(b)), we obtained a new reward which is greater than the imitation reward, fitting all the constraints. In that case, the movement of the 3-link kinematic chain is straighter, and it is more adequate to the robot structure. The ratio between the sole of human's foot and human's height is around 0.14. The ratio for the robot is 0.18. Then, the robot's feet are greater in relation with its height than the human's feet. As the robot has a wider surface, its ZMP is wider, in relation with its height. Therefore, the robot does not need to lean forward so much as when it is imitating the human performance, instead, it can go straighter, obtaining a better reward for the movement.

3.5.4 Hypothesis Testing and Generality

To prove the generality of our method, we generate up to 35 experimental trajectories of the robot's standing up behavior. This trajectories have different initial and final

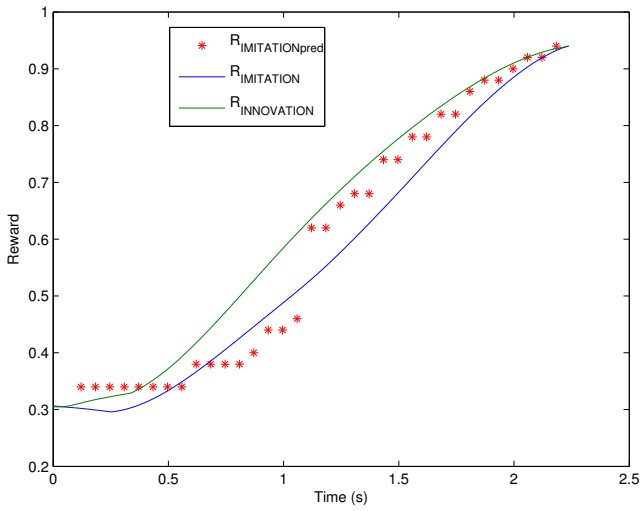
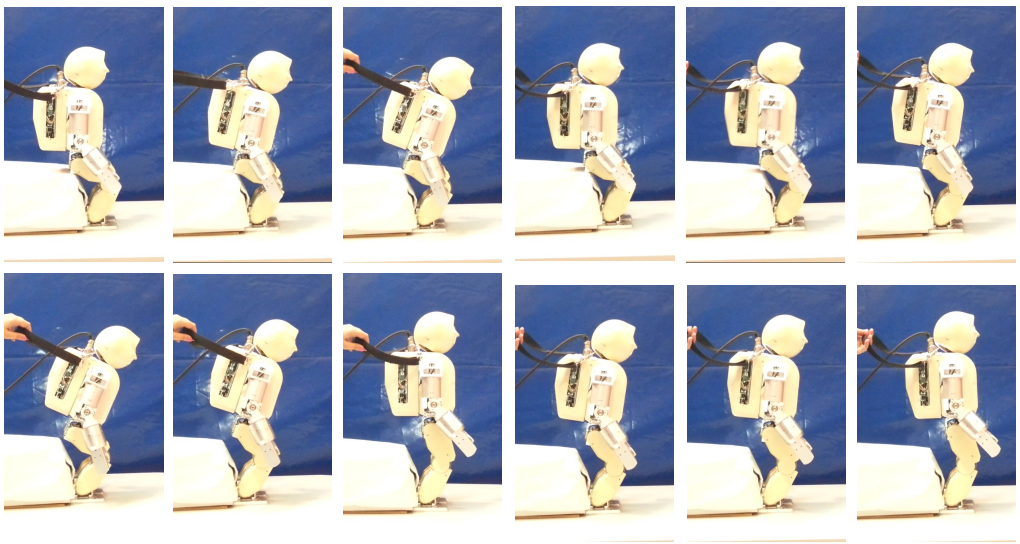


Figure 3.15: *Reward profiles for imitation and innovation robot behaviors. In blue the imitation reward, in red the predicted reward imitating a human and in green the innovation reward.*



(a) (b)

Figure 3.16: *(a) Snapshots of the robot standing up in the imitation process. (b) Snapshots of the robot standing up in the innovation process. In the imitation process the robot lean forward too much, attempting to follow the strategy of the human. However, in the innovation process the robot stands up more straightly, since it is maximizing its reward. This behavior is logical because the feet size in the case of the robot is larger in relation with the feet size of the human. Therefore, the robot does not really have to lean forward so much.*

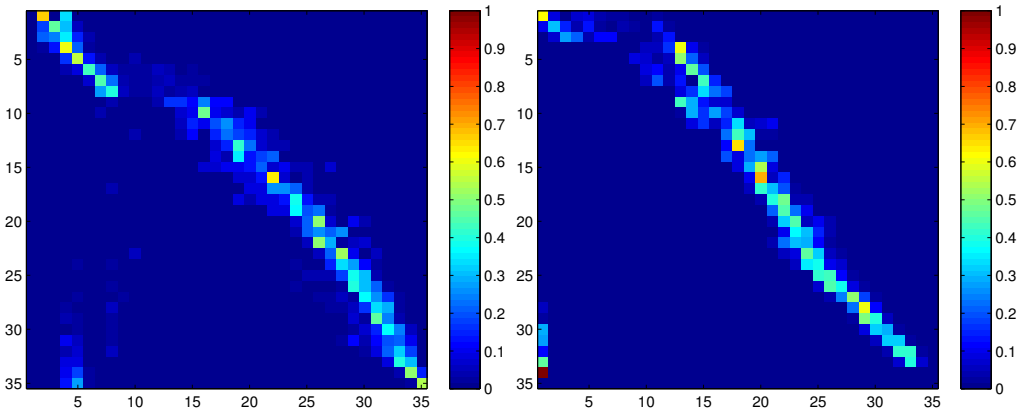


Figure 3.17: *(a) Normalized Transition Matrix of the Reward for the human using the polynomial function* (3.23)*. (b) Normalized Transition Matrix of the Reward for the robot using the polynomial function* (3.23) *and the experimental solutions.*

postures. Our method allows to robustly transit from a seated posture to a stand up posture. The initial posture is selected to not surpass the maximum torque and the final posture is stable.

We used all these trajectories to compute the RTPM for the robot as shown in Figure 3.17. This matrix represents the real behavior of the humanoid when it imitates the repertoire of human demonstrations. The human and the robot are morphologically similar though the exact scales are different. Therefore, we hypothesized that their stand up strategies should be similar. In order to test this, we can compare the human RTPM ((Figure 3.17(a))) with the robot RTPM (Figure 3.17(b)) as similar strategies should result in similar probability transitions in the reward space (see (3.11)).

To compare the matrices we compute the mean square error, e, given by

$$e = \sqrt{(T_{human}(i,j) - T_{robot}(i,j))^2} = 0.0395 \tag{3.30}$$

and then obtain the average probability error of a cell, P_e, given by

$$P_e = eP(s = s_i) = 0.1128\% \tag{3.31}$$

where $P(s = s_i)$ is the probability of staying in state s_i, which in the case of this RTPM is $1/35$. For a more detailed discussion of the human demonstration consistency please refer to 3.5.4.

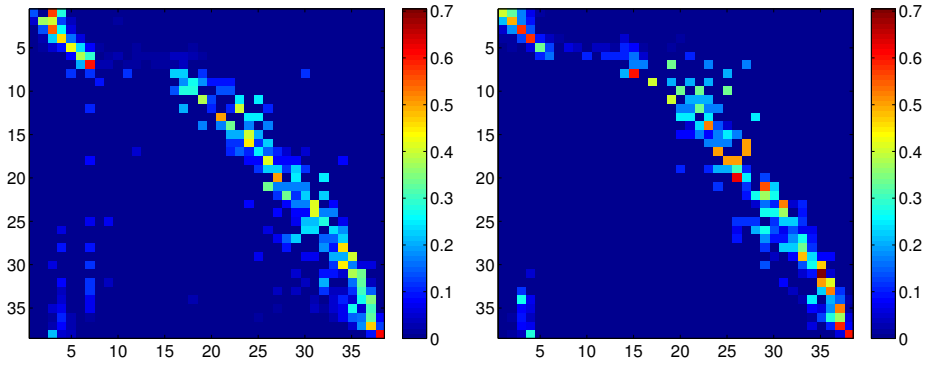


Figure 3.18: *(a) Normalized Reward Transition Probability Matrix for the first group of human participants using the polynomial function* (3.23)*. (b) Normalized Reward Transition Probability Matrix for the second group of human participants using the polynomial function* (3.23)*.*

Consistency of Human Demonstrations

Our participants varied in physical characteristics in terms of their weight, height and limb kinematics (see Table 3.2). We wanted to test the consistency of the demonstrations and therefore the reward profile among different groups of humans. For that purpose we divided the demonstrators in two groups. The group number 1 is composed by people with high height and high weight, they are the participants number 3, 5, 6 and 7. The group number 2 is composed by people with low height and low weight, they are the participants number 1, 2, 4 and 8.

We computed the RTPM for both groups obtaining Figure 3.18. To see the difference between them we compute the mean square error (3.30) which is $e = 0.0264$ and the average probability error of a cell (3.31) which is $P_e = 0.0694\%$.

Seeing the results we can conclude that there is no significant difference in the reward profiles between the two groups. Therefore there seems to be a reward profile that is independent of the human body size and can define the behavior of

Table 3.2: *Weight and height of the human participants.*

	Weight (Kg)	Height (m)
Participant 1	68.3	1.79
Participant 2	60.9	1.68
Participant 3	78.1	1.82
Participant 4	75,2	1.75
Participant 5	83.4	1.84
Participant 6	99.0	1.88
Participant 7	81.0	1.85
Participant 8	67.8	1.71

standing up. Furthermore this is equivalent to say that all humans, no matter size and weight share the same strategy to accomplish a task, which can be defined as a reward profile and transmitted to a robot.

Our results are in accordance to those saying that what should be imitated is the goal of the action, not just the movements (Thompson and Russell, 2004; Metta et al., 2006; Craighero et al., 2007; Whiten et al., 2009).

3.6 Discussion and Conclusions

In this chapter we presented an original method to obtain imitative and innovative postural behaviors in a humanoid robot through human demonstrations.

We collected data from a group of 8 human participants standing up from a chair. We modeled both human and humanoid using an actuated 3-link kinematic chain approximation and computed a reward profile in terms of ZMP and inverse dynamics. We used 20 demonstrations each from 8 participants to obtain the Markov probability transition matrix of the compound reward for the human demonstrations.

Provided the humanoid robot should follow the same optimality criteria and profile as the human if it were to imitate the human in a qualitative sense, we can use the Markov chain obtained for human demonstrations to predict the future humanoid reward starting from any state of the humanoid robot. We then optimized a joint trajectory to obtain imitation, where the robot reward is equal to the predicted human-like reward along the whole posture control period. Having achieved imitation, we proceeded to achieve robotic skill innovation where the average reward profile of the humanoid is higher than that of the average human demonstrations.

The approach discussed in this paper emphasizes the fact that intelligent behavior of an embodied agent is in the eyes of the observer (Brooks, 1991). Therefore, different observers can use different criteria to compare two embodied agents attempting to achieve a given goal. Here, we propose that the observer can compare a behavior enacted by two different embodiments in a common reward space. This paper considers the case where one multimodal reward function is used throughout the standing up behavior. However, it should be noted that there can exist state dependent heterogeneous reward functions in more complex cases. An example is to consider acceleration and joint torque optimization at the start and ZMP variability minimization in the neighborhood of the standing posture. Well established techniques of Gaussian Mixture Models can be a good technique to model such reward landscapes.

The developed algorithm produces a dynamic posture (standing up), which is the transition between the static posture of being seated and the static posture of being standing up. Both initial and final static postures are calculated in advance.

The main features of our method are discussed here:

- Our method allows to transfer a stand up behavior from a human teacher to a robot learner, even if there is a wide mismatch in their kinematic structures.
- The robot learns to perform smooth and stable standing up movements based on human demonstrations.
- The robot does not simply imitate the human movement, rather learns an optimal behavior subject to a set of internal constraints.
- It takes into account the ZMP, torque, and joint limits of the robot, so the trajectory is always executable.
- We defined a multi-objective reward profile of ZMP and joint torques and encoded the demonstrating trials of the human in a reward transition probability matrix.
- Based on neuroscientific theories that suggest that human skill transfer is achieved by imitating the goal of the action, we suppose that the reward transition probabilities of the robot show the same structure of that in the human demonstrations. Thus, we computed a constrained policy that minimizes the predicted error in the reward profile (Metta et al., 2006).
- After the imitation is achieved so that the robots reward transition probability matrix is statistically significantly equal to that of the human demonstrations, we moved on to find a new policy that improves the robot reward profile leading to skill innovation.

Chapter 4

Learning and Improving a Sequence of Goal Directed Skills

As it was discussed in the previous chapter, there are evidences that justify that the imitation between humans are goal-directed. We proposed there a new method to acquire a single skill from human demonstrations. However, it is quite common for a human being to perform several skills sequentially, for example, to walk to a door and open it. Therefore, when performing multiple skills, we internally define an unknown optimal policy to satisfy multiple goals. This chapter presents a method to transfer a complex behavior composed by multiple skills from a human demonstrator to a humanoid robot. We defined a multi-objective reward function as a measurement of the goal optimality for both human and robot, which is defined in each subtask of the global behavior. We optimized a hierarchical policy to generate whole-body movements for the robot that produces a reward profile that is compared and matched with the human reward profile, producing an imitative behavior. Furthermore, we can search in the proximity of the solution space to improve the reward profile and innovate a new solution, which is more beneficial for the humanoid. Experiments were carried out in a real humanoid robot.

4.1 Introduction

When a human performs a high level task like *"stand up from a chair, walk to the door and open it"* there is a sequence of skills that take place to optimally perform the task, such as approaching the door in a manner that the location of the body makes it easier to reach the knob, grasping the knob, performing the movement that activates the mechanism of opening the door, going backwards while holding the knob, detecting that the door is open in a way that it can be overpassed, and finally, going through the opened doorway. All these skills are automatically selected to optimize, in some manner, the high level strategy of performing this task. Figure 4.1 shows a detail of the high level task of opening the door.

As it was stated in the previous chapter, recent psychology studies suggest that when a human reproduces a learned task, he understands the consequences of this behavior and attempt to emulate the overall goal (Whiten and Ham, 1992; Gergely et al., 2002; Metta et al., 2006; Craighero et al., 2007). Even recent studies demonstrate that the main difference between apes and humans is our capability to *over-imitate*, or find newer and better solutions to accomplish optimal actions (Whiten et al., 2009; Nielsen et al., 2014). In that sense, innovation is an essential feature of the human behavior.

Robots need to be able to handle similar situations, finding an optimal way to successfully complete these tasks, while maintaining the balance and moving in a safe and smooth manner. In recent years, researchers have made a significant effort to cope with this problem and Learning from Demonstration (LfD) (Argall et al., 2009; Gribovskaya et al., 2010; Guenter et al., 2007; Khansari-Zadeh and Billard, 2011; Calinon et al., 2010; Ariki et al., 2013; Billard et al., 2006) has became one of

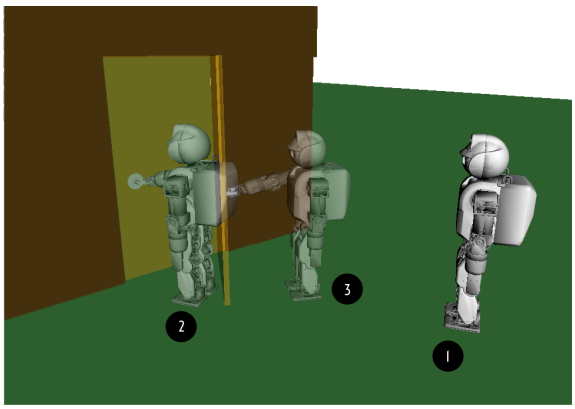


Figure 4.1: *Behavior sequence detail of the high-level task of opening a door by a simulated HOAP-3 robot. The robot starts at point 1, it approaches to point 2 near the door in a way it can reach the knob, after grasping the door, it pulls back the door to point 3. Finally, it releases the knob.*

the most popular ways to create motor skills in a robot. One of the key questions to be solved is *what to imitate* (Argall et al., 2009; Alissandrakis et al., 2002).

Minsky (2006) suggested that the way to create a machine that imitates the human behavior is not by constructing an unified compact theory of artificial intelligence. On the contrary, he argues that our brain contains resources that compete between each other to satisfy different goals at the same moment. A similar view is shared by Brooks (1990, 1991). Starting from that idea, our approach is based on computing different reward profiles for different behaviors, which sequentially optimizes different goals.

In this chapter a sequential method to learn concurrent behaviors from a human demonstrator is presented. The behavior is adapted to the robot embodiment and refined to successfully accomplish the desired task. We collected data from several human demonstrators performing a complex task composed by a set of sequential behaviors. Extracting determined features of every behavior, like COM position, human orientation, hand trajectory, etc., and encoding them using Gaussian Mixture Models (GMM), we defined a multi-objective reward function which represents the overall goal, which is used as a basis of comparison or metrics with the robot. The reward is used to solve the *correspondence problem*, which is defined as the action mapping between the demonstrator and the imitator (Alissandrakis et al., 2002). In this regard, we mapped movements performed in a different kinematic domain and at a different scale to a common domain, defined as the goal domain and expressed mathematically as a reward profile, formed by a multimodal landscape of movement features.

In a previous work, we addressed the problem of mapping a behavior from a group of unexperienced workers to match and even surpass the expert behavior of an elite individual (Nanayakkara et al., 2007). Using that idea, we proposed a method of imitation learning of a single behavior in a small humanoid robot using the reward as a common space of comparison (González-Fierro et al., 2013a). Later in (González-Fierro et al., 2014a), we extended this work by not only imitating but innovating new behaviors using a Markov Transition Matrix to encode the reward variability and represent the behavior strategy when performing an action while fitting the robot's internal constraints and kinematic structure.

We define a sequential policy for the robot that allows to find in which behavior the robot is and computes a constrained whole-body movement pattern that optimizes the reward in order to be as close as possible to the human's reward. Then, we refine the policy by innovating a new solution which improves the current robot reward.

4.1.1 Behavioral Planning

There are many studies conducted in the area of LfD using GMM and Gaussian Mixture Regression (GMR) to encode kinesthetic trajectories and generalize them to

perform a robot movement (Gribovskaya et al., 2010; Guenter et al., 2007; Calinon, 2009; Khansari-Zadeh and Billard, 2011) or based on Hidden Markov Models (HMM) to encode the human demonstrations so they can be transferred to the robot Calinon et al. (2010); Ariki et al. (2013); Billard et al. (2006). In our work, instead of learning the robot movement we learn a reward function, which is the basis of comparison between the human and the robot.

Our work takes some ideas from (Guenter et al., 2007) where a humanoid robot use LfD to initially learn a pick and place task. Later, if an obstacle interrupts the movement, a new movement is computed using Reinforcement Learning to avoid the obstacle. This gave us the idea of not only using the reward profile as the space of imitation but as a metrics of the behavior's performance. Therefore, improving the imitation reward by searching in the neighbourhood of the reward space, the robot can obtain a better reward which by innovating new behaviors which are not those learned from imitation.

The authors in (Muhlig et al., 2009) address the problem of *what to imitate* in a similar way to our proposal. They identified several possibilities of task space methods to imitate, what they called task space pool. Next, they defined several criteria, like an attention criterion or an effort criterion, to choose the optimal task space to imitate. In our case, instead of defining a pool of criteria, we learn a probabilistic behavior selector matrix from human demonstrations. It defines the probability of being in a behavior given a set of states. Another interesting approach is presented in (Grollman and Billard, 2011), where instead of the usual LfD approach, the robot learns from failed demonstrations.

There are many works related to policy learning like (Matsubara et al., 2008; Yamaguchi et al., 2010; Yi et al., 2011). In (Daniel et al., 2012) a policy search method is applied to optimally select between several solutions of the same task, initially learned from demonstrations. Our work shares a similar idea, but instead of selecting solutions of the same task, we select between sub behaviors of a complex task and attempt to find an optimal policy that produces a similar reward to that of the human.

The problem of skill transfer and whole body motion transfer has been an interesting area of research in recent years. Some studies addressed the problem manipulating the angular momentum of the COM (Naksuk et al., 2005; Matsubara et al., 2008), using graphs and Markov chains (Kulić et al., 2008), imitation of movement using Neural Networks (Yokoya et al., 2006) or Bayesian Networks (Grimes et al., 2006), sequencing multicontact postures (Bouyarmane and Kheddar, 2012) or encoding and organizing learned skills (Lin and Lee, 2008).

The framework called *incremental learning* uses a few demonstrations to perform a task which is incrementally improved with the aid of verbal or non-verbal guidance. In (Saunders et al., 2006) a human guided a robot to sequentially construct memory

models of the desired task. This incremental learning method, inspired on the behavior of social animals, allows to combine different competences to create complex tasks. Some approaches like (Pardowitz et al., 2006) are based on constructing a task graph that leads to more general behaviors. Kulic et al. (Kulić et al., 2008; Kulic and Nakamura, 2009) generated whole-body motion using factorial HMM, that encodes and clusters a set of incrementally learned movement primitives that can be combined to generate different behaviors.

Our work has points in common with (Stulp et al., 2012) in the sense that they proposed a RL algorithm for robot manipulation that simultaneously optimizes the shape of the movement and the sequential subgoals between motion primitives. In contrast, we established a set of behaviors, each of them with a different goal, and also a reward profile that represents that goal.

Our approach shares many similarities with *inverse reinforcement learning* (IRL) (Ng and Russell, 2000; Abbeel and Ng, 2004; Ziebart et al., 2008; Lopes et al., 2009) and *inverse optimal control* (IOC) (Ratliff et al., 2009; Mombaur et al., 2010; Kalakrishnan et al., 2013). IRL is initially presented in (Ng and Russell, 2000; Abbeel and Ng, 2004) as the problem of extracting a reward function given an optimal behavior. The reward is extracted as a lineal combination of basis features of the behavior. It can be obtained using support vector machines (Abbeel and Ng, 2004), methods based on maximum entropy (Ziebart et al., 2008) or active learning (Lopes et al., 2009). Similarly, IOC aims to determine the optimization criterion that produced a demonstrated dynamic process. It was succesfully applied to locomotion (Mombaur et al., 2010), pedestrian detection (Ratliff et al., 2009) and manipulation (Kalakrishnan et al., 2013). In contrast with the commented approaches that attempt to explain the observations with rewards functions defined for the complete behavior, our method relies on context-dependent goal oriented reward functions that are selected depending on which task the robot is executing.

Recently, two articles with a similar approach to our work appeared (Malekzadeh et al., 2013; Calinon et al., 2014). They were coauthored by Nanayakkara, one of the advisors of this thesis. In their work a surgical robot learns several tasks demonstrated by a surgeon, who selects a set of critical points that the robot's end effector has to touch. They proposed a LfD and skill innovation method based on the reward. One of the main differences with the work proposed in this thesis is the way they select the basis reward functions and how they relate to each other. Instead of defining a fixed reward function for each task or goal, the robot is provided with a set of candidate reward functions. The optimal combination of these basis functions and in which proportion they are relevant to different parts of the task, are learned by demonstrations. In our work a fixed reward function is defined for each part of the behavior. The use of a specific reward function is decided by a selector matrix, learned from the human, that predicts the current state of the robot behavior and allow to apply the associated reward function.

4.1.2 A Humanoid Opening a Door

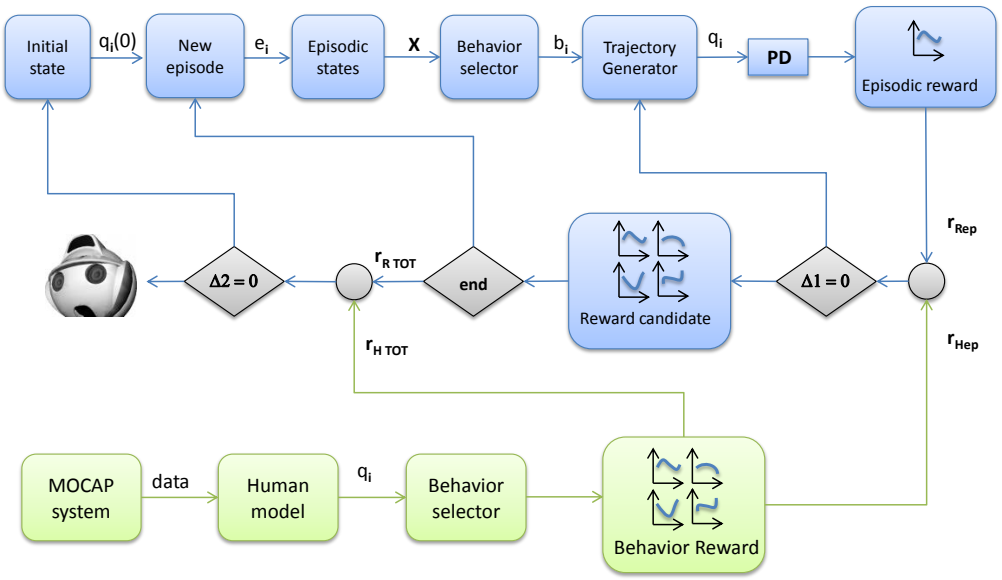


Figure 4.2: *Overview of the imitation system. Using a MOCAP system, the movement of the human demonstrator is obtained and a reward profile for every behavior is computed. On the other hand, the robot starts in an initial state $q_i(0)$. A new episode e_i is defined, which is a pair of initial and final states $\mathbf{X} = (\xi_{ini}, \xi_{final})$. Then the behavior selector decides in which behavior the humanoid is. The trajectory generator produces a stable trajectory within the pair of states. The episodic reward is optimized until the difference Δ_1 between the robot reward r_{Rep} and the human reward r_{Hep} is small or it reaches a number of iterations. This process is repeated until all behaviors are completed. Then, there is a comparison between the complete reward profile of the robot r_{RTOT} and the complete reward profile of the human r_{HTOT}, which is the index $\Delta_2 = J$. If this index is close to zero it means that the imitation is completed.*

Figure 4.2 and Figure 4.3 show the complete architecture of both sequential imitation learning and sequential innovation learning. There are two optimizations in every architecture. A local optimization between behavior episodes and a global optimization of the complete behavior. Therefore, the system does not only obtain a local stable movement but it takes into consideration the complete shape of the action movement.

Figure 4.2 shows the imitation learning process. The human data is acquired using a MOCAP system, which in our experiments is a Kinect camera. This data is used to obtain a model of the human, which generates a joint trajectory q_i. This trajectory is used to compute the behavior selector matrix and the human reward profile. The behavior selector matrix indicates the probability distribution of being

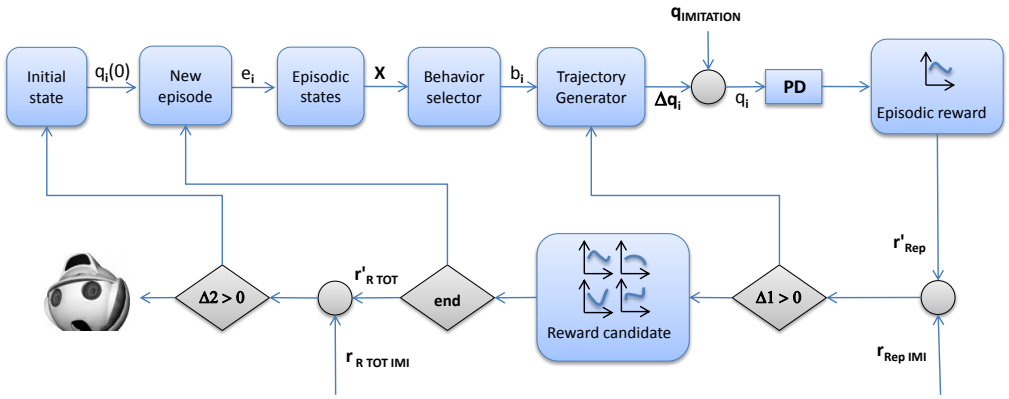


Figure 4.3: *Overview of the innovation system. The process is very similar to Figure 4.2. The main difference appears in the trajectory generation. The generator perturbs the imitation trajectory $q_{IMITATION}$ in an amount Δq_i, to generate a new trajectory q_i which is evaluated in terms of the episodic reward. The other difference is in the reward comparison Δ_1 and Δ_2. In this case the objective is to maximize the difference. If the robot gets a better episodic reward r'_{Rep} than the reward obtained in the imitation r_{RepIMI}, then $\Delta_1 > 0$ and the local optimization ends. If the robot gets a better reward r'_{RTOT} than the reward obtained in the imitation $r_{RTOTimi}$, then $\Delta_2 > 0$ and the innovation is completed.*

in a determined behavior given a state. The human reward is compared globally and locally with the robot reward.

The robot imitation process begins by knowing its initial joint values $q_i(0)$. At this point a new episode e_i begins. An episode is a transition between a pair of initial and final states $\mathbf{X} = (\xi_{ini}, \xi_{final})$, which depends on the current behavior b_i, the generated trajectory q_i, the controller PD and the episodic reward r_{Rep}. Then a local optimization takes place.

The robot reward r_{Rep} is compared with the human reward r_{Hep}. When its difference Δ_1 is a small number or the maximum number of iterations have passed, the robot satisfactorily imitates the human and the reward candidate for this episode is saved. This process is repeated until all episodes have been computed.

When this loop finishes it means that all episodes for all behaviors have been computed and a complete movement candidate is available. At this point, a global optimization process takes place to minimize the difference between the total robot reward r_{RTOT} and the total human reward r_{HTOT}, denoted by Δ_2. If they are similar, the process stops and we can conclude that the imitation process is not only successfully achieved locally but globally, taking into account the complete movement.

Figure 4.3 shows the innovation learning process. It is very similar to the imitation process but this time, instead of comparing with the human reward, it compares

with the best reward of the imitations process. Therefore, in this case the robot is not just imitating the human but generating an innovative behavior which is better, since the performance can be measurement with the reward profile.

4.2 Sequential Policy Definition

A learning process that considers the Markov property to predict actions and states is called a Markov Decision Process (MDP) (Nanayakkara et al., 2009).

The learning process is defined as a sequence of finite states $s \in S$ and actions $a \in A$ pairs, that produces an associate reward $r \in R$. The agent, starting from a state $s(t)$ computes an action $a(t)$ to reach a future state $s(t+1)$, obtaining a reward $r(t)$, which can be defined as a set of values or as a mathematical function, it is usually called *the reward function*.

Let $b \in B$ be a set of behaviors or skills that compose the full high level strategy of performing a task. An example of a behavior or a skill can be approaching the door in a manner that the location of the body allow to reach the knob, grasping the knob, performing the movement that activate the mechanism to open the door, going backwards while holding the knob, realizing that the door is open in a way that it can be overpassed, and finally, passing through the doorway.

Let it be noted that the states and actions of every individual behavior can determinate the better or worse performance of the following ones. For example, if we analyze only the behavior of grasping the knob, it could be beneficial if the robot is in a frontal position in relation to the plane of the door, however, if we want to concatenate this with the behavior of going backwards opening the door, maybe it is better first to locate the robot with a determined angle to then perform an easier trajectory of going backwards.

The objective is to determine a policy $\pi(a|s)$ in the form

$$\pi(a|s) = \sum_{b} \pi(a|s,b)\pi(b|s) \tag{4.1}$$

where $\pi(b|s)$ is the selector of behavior b given a state s, and the policy $\pi(a|s,b)$ to select the action a, given a behavior b.

We consider an episodic learning strategy to generate a policy inside every behavior. At the beginning of the episode, starting from a state s, we compute a parametrized postural primitive that takes into account the whole body movement, while maintaining the stability. The parametrized postural primitive can be defined in several ways, in some works like (Daniel et al., 2012) the movement is computed as a Dynamic Movement Primitive (DMP) (Schaal et al., 2005). For more complex trajectories that implies displacement and manipulation at the same time, it is easier to define trajectories in the task space (Sentis et al., 2010).

For each episode, we only consider one action a that determines the parameters of the postural primitive, which for instance defines the movement plan for the complete episode. The states are defined as the via points of the primitive and the reward profile is computed from the reward function $r^{\pi}(s, a, b)$. The reward is constructed as a metrics to measure the overall goal performance and it depends on the behavior, the initial state of the episode and the action that takes place in this episode.

Figure 4.4 represents a diagram of an episode. It shows how situations branch off to behaviors and then to actions. Given a situation in the state space, there can be many behaviors according to human demonstrations. Given a behavior, the action to change over time. We choose a branch in the tree using a probability distribution derived from the demonstrations.

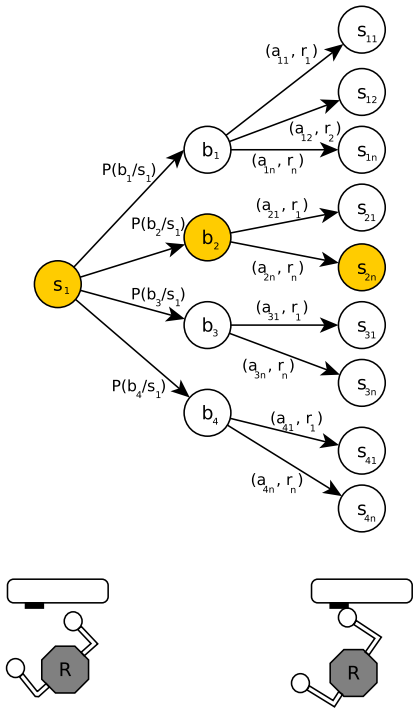


Figure 4.4: *Diagram explaining one episode. The robot, represented in the lower part of the diagram, performs a transition from state s_1 to s_2. In the upper part of the diagram there is a tree representing the complete process. Given a state s_1, the behavior selector $\pi(b|s)$ computes the probability of being in a behavior $P(b_i/s_1)$. Then $\pi(a|s, b)$ generates an action a_{ij}, which retrieves a reward r_j. The generated action takes the robot to a state s_{ij}. The selection of one branch, in yellow, is determined by the most probable behavior and by the best reward.*

4.2.1 Encoding and Generalizing Demonstrations

The probability distribution space of the human demonstrations is approximated using GMM. A time independent model of the motion dynamics is estimated through a set of first order non-linear multivariate dynamical systems. Schaal et al. (2007) proposed an approach based on imitation learning and on-line trajectory modification, by representing movement plans based on a set of non-linear differential equations with well-defined attractor dynamics. We follow a framework presented on (Gribovskaya et al., 2010) allowing learning non-linear dynamics of motions and generating dynamical laws for control.

A variable ξ is defined describing the state of the robot. Let the set $\mathbf{M}$ of N-dimensional demonstrate data points $\{\xi_i, \dot{\xi}_i\}_{i=0}^{M}$ be instances of a global motion governed by a first order autonomous ordinary differential equation (ODE):

$$\dot{\xi}(t)^M = f(\xi(t)^M), \tag{4.2}$$

where $\xi^M \in R^n$, and its time derivative $\dot{\xi}^M \in R^n$ are vectors that describe the robot motion. The problem then consists on building a stable estimate $\hat{f}$ of f based on the set of demonstrations. Without loss of generality, we can transfer the attractor $\bar{\xi}$ to the origin, $\bar{\xi} = 0$, so that $f(\bar{\xi}) = f(0) = 0$ and by extension $\hat{f}(\bar{\xi}) = \hat{f}(0) = 0$.

To build the estimate $\hat{f}$ from the set of demonstrated data points $\{\xi_i, \dot{\xi}_i\}_{i=0}^{M}$ we follow a statistical approach and define $\hat{f}$ through a GMM.

Gaussian Mixture Models

The GMMs define a probability distribution $p(\xi^i, \dot{\xi}^i)$ of the training set of demonstrated trajectories as a mixture of the $\boldsymbol{K}$ Gaussian multivariate distributions $\mathbf{N^k}$

$$p(\xi^i, \dot{\xi}^i) = \frac{1}{K} \sum_{k=1}^{K} \pi^k N^k(\xi^i, \dot{\xi}^i; \mu^k, \Sigma^k) \tag{4.3}$$

Where π^k is the prior probability; $\mu^k = \{\mu_{\xi}^k; \mu_{\dot{\xi}}^k\}$ is the mean value; and $\Sigma^k = \begin{bmatrix} \Sigma_{\xi}^k & \Sigma_{\xi\dot{\xi}}^k \\ \Sigma_{\dot{\xi}\xi}^k & \Sigma_{\dot{\xi}}^k \end{bmatrix}$ is the covariance matrix of a Gaussian distribution $\mathbf{N^k}$. The theoretical analysis of *GMMs* can be found on (Dasgupta and Schulman, 2000).

The probability density function of the model $N^k(\xi^i, \dot{\xi}^i; \mu^k, \Sigma^k)$ is then given by:

$$N^k(\xi^i, \dot{\xi}^i; \mu^k, \Sigma^k) = \frac{1}{\sqrt{(2\pi)^{2d}|\Sigma^k|}} e^{\frac{-1}{2}([\xi^i, \dot{\xi}^i] - \mu^k)^T (\Sigma^k)^{-1} ([\xi^i, \dot{\xi}^i] - \mu^k)} \tag{4.4}$$

By considering an adequate number of Guassians, and adjusting their means and covariance matrix parameters, almost any continuous density can be approximated

to arbitrary accuracy. The form of the Gaussian mixture distribution is governed by the parameters π^k, μ^k, Σ^k. The model is initialized using the k-means clustering algorithm starting from a uniform mesh and is refined iteratively through *Expectation-Maximization (EM)* for finding the maximum likelihood function of (4.3).

$$\ln p(\xi^i, \dot{\xi}^i) = \sum_{n=1}^{N} \ln\{\sum_{k=1}^{K} \pi^k N(\xi_n^i, \dot{\xi}_n^i | \mu^k, \Sigma^k)\} \tag{4.5}$$

Figure 4.5(a) illustrates the encoding of a training data set $\{\xi_i, \dot{\xi}_i\}_{i=0}^{M}$ into a model of mixtures of Guassians (Figure 4.5(b)).

In this work we used the Binary Merging (BM) algorithm, (Khansari-Zadeh and Billard, 2010), to build the GMM. BM determines an optimal minimum number of Gaussian functions to employ, while satisfying the stability criteria and also keeping the error of the estimates under a threshold. To generate a new trajectory from the GMM, one then can sample from the probability distribution function $p(\xi^i, \dot{\xi}^i)$, this process is called Gaussian Mixture Regression (GMR).

Gaussian Mixture Regression

The GMM computes a joint probability density function for the input and the output so that the probability of the output conditioned on the input are a Mixture of Gaussians. So it is possible after training, to recover the expected output variable $\hat{\xi}$, given the observed input ξ. Taking the conditional mean estimate of $p(\dot{\xi}|\xi)$, the estimate of our function $\hat{\dot{\xi}} = \hat{f}(\xi)$ can be expressed as a non-linear sum of linear dynamical systems, given by:

$$\hat{\dot{\xi}} = \sum_{k=1}^{K} h_k(\xi)(\Sigma_{\dot{\xi}\xi}^k (\Sigma_{\xi}^k)^{-1}(\xi - \mu_{\xi}^k) + \mu_{\dot{\xi}}^k) \tag{4.6}$$

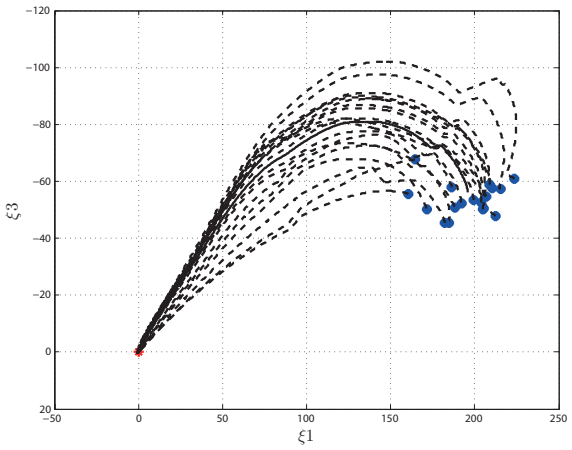
where

$$h_k(\xi) = \frac{p(\xi; \mu_{\xi}^k, \Sigma_{\xi}^k)}{\sum_{k=1}^{K} P(\xi; \mu_{\xi}^k, \Sigma_{\xi}^k)}, h_k(\xi) > 0 \tag{4.7}$$

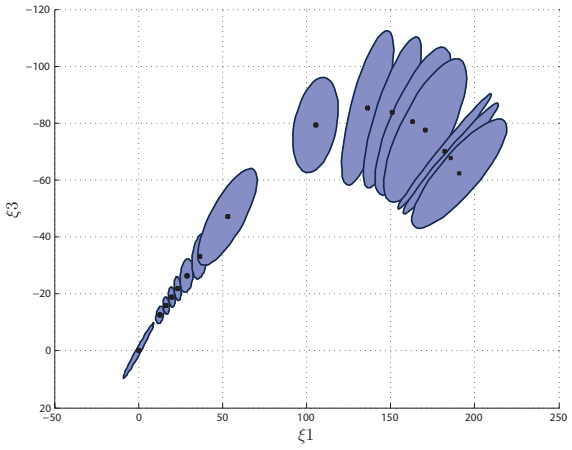
and $\sum_{k=1}^{K} h_k(\xi) = 1$

This process is called Gaussian Mixture Regression *GMR*. A review of GMR can be found in (Sung, 2004).

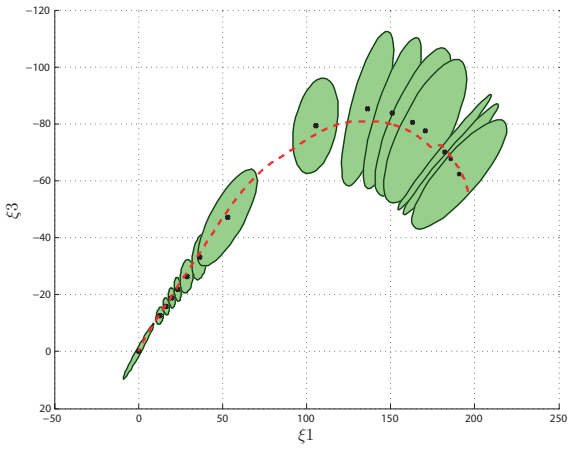
Figure 4.5(c) illustrates the GMR as a reproduction of the learned motions. To learn the model of the trajectories, first several demonstration of the task are presented and them the trajectory is encoded as a mixture of Gaussian distributions. To reproduce the trajectories one sample from the probability distribution of the GMM trough the GMR process. The GMR approximates the dynamical systems through a non-linear weighted sum of local linear models.



(a)



(b)



(c)

Figure 4.5: *Illustration of the learning process of grasping a door knob from a top view. (a) Training data of the task. (b) GMM of the learned motion. (c) Reproduction of the GMR.*

4.2.2 Parametrized Postural Primitives

Based on the demonstrations encoded as GMM, we can compute a parametrized postural primitive for each episode, which defines a complete motion of the humanoid in the task space. To simplify the process of generating a whole body motion taking into account contacts and stability, we decoupled the robot in two modules or tasks, the locomotion task and the grasping task. This means that when the robot is performing a locomotion task, the module in charge of computing the grasping task is stopped. In a similar way, when the robot is performing a grasping operation the locomotion module is stopped.

There is a moment when the robot is moving backwards and, at the same time, is grasping the knob. At this moment, the only active module is the locomotion one. The robot arm is idle to decouple the robot from the door dynamics. We assumed that the door weight is small in comparison with the robot weight and the resistive torque of the hinge is negligible.

For the locomotion task of the humanoid, the postural primitive can easily be computed using the cart-table model (Kajita et al., 2003a). This model is based on ZMP a preview control scheme to obtain the COM trajectory from a defined ZMP trajectory. This method generates a dynamically stable gait trajectory using the Inverted Pendulum Model to approximate the dynamics of the humanoid.

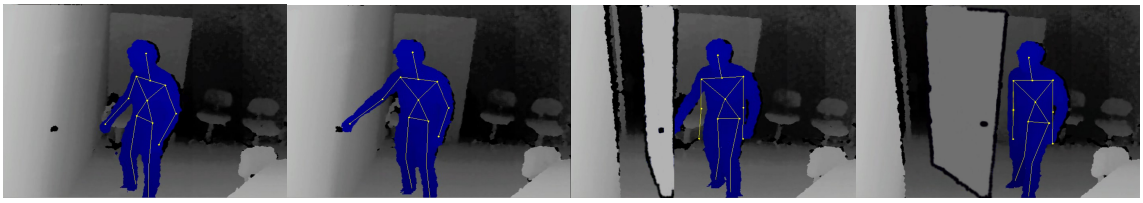
Regarding the grasping, we can use GMR to define a desired trajectory for the hands and add a modulation term that improves the reward index, similarly to (Guenter et al., 2007).

4.2.3 Sequential Policy Search

The sequential policy search problem is defined as an optimization problem where we used the reward framework as a basis of comparison between the human and the robot (González-Fierro et al., 2013a, 2014a). The objective is to find a policy for the robot that, in an initial moment, imitates the behavior of the human, by producing a similar reward profile, and later improves the robot performance, by auto exploring new solutions that return a better reward.

Taking that into account, we can define an imitation index J, which is defined as the optimization problem of minimizing the episodic difference of the human and robot reward profile (4.8), and the innovation index J', which is defined as the optimization problem of maximizing the positive difference between the episodic imitation reward profile and the new innovation profile (4.9). To compare between reward profiles we make use of the Kullback-Liebler divergence, which can be stated as a directional information transfer.

$$\min\ J = \sum_{b} \sum_{e} r^{h}(s,a,b) \log \frac{r^{h}(s,a,b)}{r^{r}(s,a,b)} \tag{4.8}$$



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

Figure 4.6: *Snapshots of one human demonstrator performing the task of opening the door using the Kinect camera. Each snapshot corresponds to a different behavior. a) Behavior b_1: approaching the door, b) Behavior b_2: grasping the knob, c) Behavior b_3: pulling back the door and d) Behavior b_4: releasing the knob.*

where $e \in E$ is the episode, r^h is the human reward profile and r^r is the robot reward profile.

$$\max J' = \sum_b \sum_e r_i^r(s,a,b) \log \frac{r_i^r(s,a,b)}{r^r(s,a,b)} \tag{4.9}$$

subject to

$$\mu_i^r(e) \geq \mu^r(e) \tag{4.10}$$

where r_i^r is the innovation reward profile of the robot, $\mu^r(e)$ is the mean of the imitation reward profile in episode e and $\mu_i^r(e)$ is the mean of the innovation reward profile in episode e. The optimization process is performed using Differential Evolution optimizer (Storn and Price, 1997).

4.3 Experimental Results

The task chosen for testing our method is to make a humanoid robot approach a door, grasp the knob and open the door while maintaining the balance. The robot used is the middle sized humanoid HOAP-3 of Fujitsu.

4.3.1 Acquiring Behaviors from Human Demonstrations

The experimental setup consist of a Kinect camera recording 9 human participants opening a door 10 times each (see Figure 4.6). The API of the Kinect allows to perform an accurate tracking of the human body, which is improved using a Kalman Filter.

The complete task is segmented into several behaviors $b \in B$. The first behavior b_1 consists on approaching the door to a place where the knob can be reached; then grasping the knob b_2; going backwards leaving the arm passive, but without releasing the knob b_3 and finally, releasing the knob b_4.

The selected states for the task are position and orientation of the COM, $\xi_{com} = \{x_{com}, y_{com}, \theta_{com}\}$ and the position of the grasping hand, $\xi_{hand} = \{x_{hand}, y_{hand}, z_{hand}\}$. All states are measured with respect to the Kinect position.

Let it be noted that the identification, and therefore the segmentation, of a behavior depends on the perspective of the observer (Brooks, 1990, 1991; Minsky, 2006). As Brooks (1990) suggested, behavior representation has to be grounded to the physical world and it relies on the feedback received through the interaction with the environment. We divided the task of opening a door into 4 behaviors, however another observer could define a different set of behaviors or it can be done techniques like in Grimes et al. (2006); Chalodhorn et al. (2009); Aleotti and Caselli (2012).

For each human demonstration, a temporal state trajectory $\xi = [\xi_{com}, \xi_{hand}]$ is obtained using the Kinect API. After a filtering, the trajectory is automatically classified into the four behaviors. For b_1, approaching the door, ξ_{com} approaches to the door, whose position with respect to the Kinect reference system is known. In b_2, grasping the knob, ξ_{hand} goes up until it touches the knob, whose position with respect to the Kinect reference system is also known. b_3 starts when the hand grasps the knob and ξ_{com} moves backwards. Finally, in b_4 the hand release the knob and ξ_{hand} goes down to a rest position.

Let it be noted that the demonstrations performed by all subjects are in some sense artificial. In order to make the automatic behavior segmentation easier, the subjects are told to perform each behavior separately, i.e., they first approach the door, then move his hand to grasp the knob, then pull the door and finally release the knob. A human opening a door in a real environment would perform several of these behaviors at the same time, smoothly and elegantly.

4.3.2 Learning the Behavior Selector from Human Demonstrations

By observing the human demonstrations, we can construct the behavior selector $\pi(b|s)$ in (4.1), by obtaining the probability of being in a determined behavior given a combination of states.

Figure 4.7 represents the mean and standard deviation of all human demonstrations segmentated by behaviors.

In order to compute the behavior selector matrix of Figure 4.8, we first divide each state length into z substates, where the length is $l_i = s_{imax} - s_{imin}$ and the step is $\Delta s_i = l_1/z$. Therefore, each state is composed by a number of substates $[s_{ia}, s_{ib}, s_{ic}, ..., s_{iz}]$. Next, for each human demonstration in each behavior, we perform a mapping from trajectories to substates, obtaining the probability matrix of Figure 4.8.

In order to compute the probability of being in behavior b_i given a combination

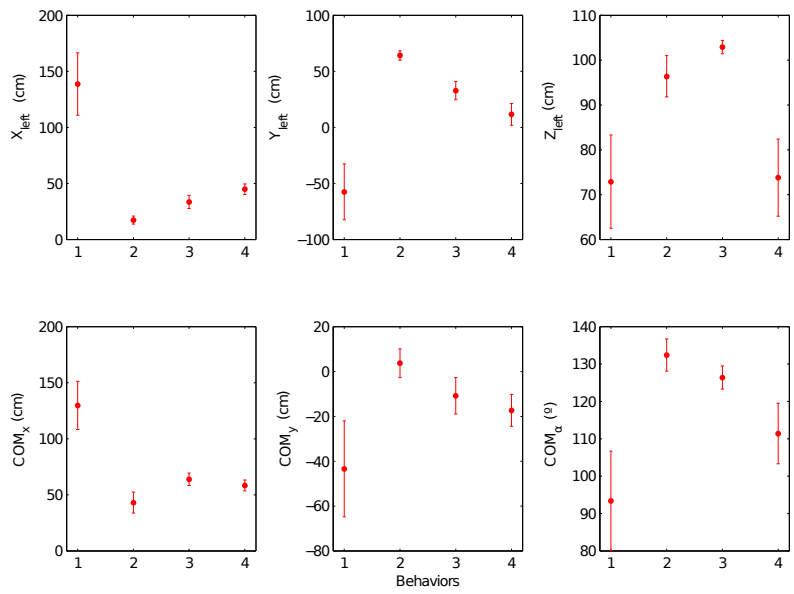


Figure 4.7: *Mean and standard deviation of the human demonstrated states.*

of states $\hat{s} = \{s_{1,a}, s_{2,b}, ..., s_{n,z},\}$, where n is the number of states and $a, b, ..., z$ corresponds to an arbitrary substate inside a state:

$$P(b_i|\hat{s}) = P(b_i|s_{1,a}).P(b_i|s_{2,b})...P(b_i|s_{n,z}) \tag{4.11}$$

Finally the selector of behavior can be stated as

$$\pi(b|s) = b_i \text{ with } b_i = \text{argmax}_i(P(b_i|\hat{s})) \quad i \text{ from 1 to } m \tag{4.12}$$

where m is the total number of behaviors.

Once the behavior selector matrix is obtained, it can be used to predict the current robot behavior, given a combination of states. In the case of the robot we applied a scale factor ρ to obtain the length $l'_i = l_i/\rho$ and the step $\Delta s'_i = l'_1/z$.

4.3.3 Definition of the Reward Profile

The reward function $r^{\pi}(s, a, b)$ varies depending on what behavior is being performed. Let define d_i as the quadratic difference of the actual state ξ_i and ξ^*_i, defined as the GMR of the human demonstrations (4.6) in the case of the human and an adapted trajectory for the robot based on the GMR human trajectory.

$$d_i = (\xi_i - \xi^*_i)^T W (\xi_i - \xi^*_i) \tag{4.13}$$

with W a weight matrix.

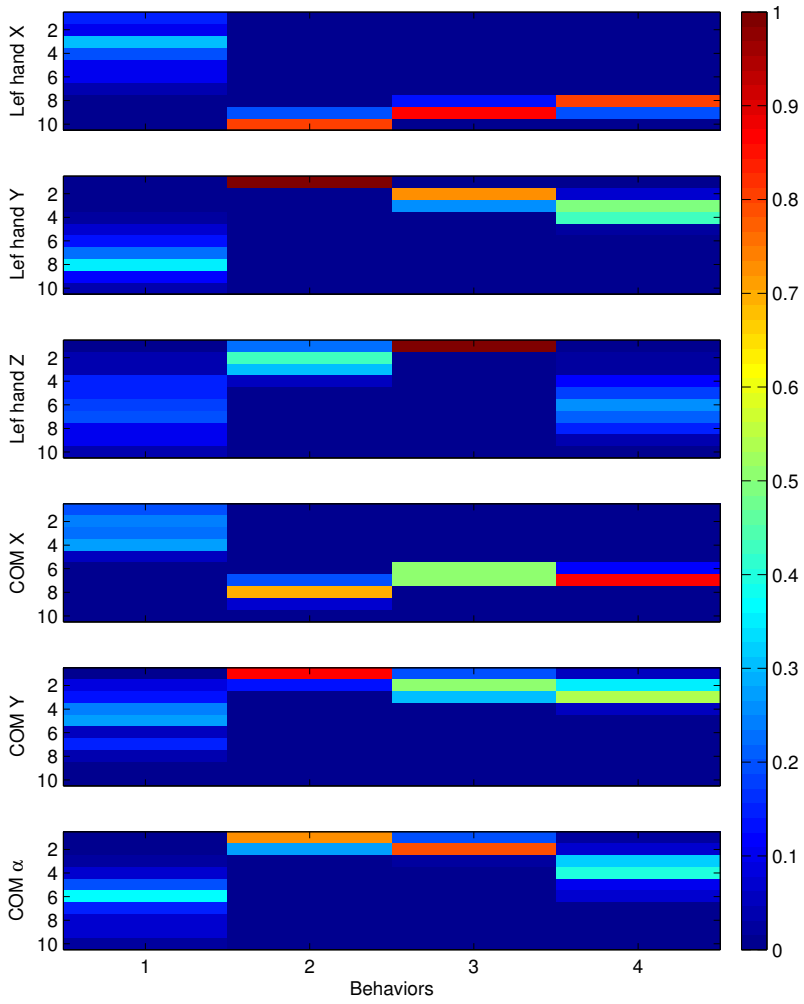


Figure 4.8: *Behavior selector matrix. The columns represent the behaviors and the rows represent the substates in a state. For each substate the color represents the probability of being in a behavior.*

We also define the reward i as a Cauchy distribution in the form

$$r_i = \frac{1}{\epsilon + d_i} \tag{4.14}$$

with a small ϵ.

Let be defined the reward for each behavior.

$$r^{\pi}(s, a, b_1) = \frac{1}{2} \sum_{e} r_{com} + r_{door} \tag{4.15}$$

$$r^{\pi}(s, a, b_2) = \frac{1}{2} \sum_{e} r_{hand} + r_{knob} \tag{4.16}$$

$$r^{\pi}(s, a, b_3) = \frac{1}{2} \sum_{e} r_{com} + \hat{r}_{\alpha} \tag{4.17}$$

$$r^{\pi}(s, a, b_4) = \frac{1}{2} \sum_{e} r_{hand} + r_{antiknob} \tag{4.18}$$

and

$$\hat{r}_{\alpha} = \frac{\alpha}{\alpha_{max}} \tag{4.19}$$

where r_{hand} and r_{com} represent the reward when the hand and COM trajectory of robot and human is close to the trajectory defined by the GMR of the human demonstrations. Both terms represent a direct imitative behavior. The closer the actual trajectory is to the desired trajectory, the higher the reward. The term r_{door} is the reward obtained for locating in a point near the door where the knob can be reached, the closer the point the higher the reward. r_{knob} is the reward obtained by the difference between the hand and the knob position, the closer the hand to the knob, the higher the reward. The term $\hat{r}_{\alpha}$ represents the reward given for the achievement of the hight level task, which is to open the door. Finally $r_{antiknob}$ is a reward that penalizes to have the hand close to the knob and follows a sigmoid function that starts on zero and finishes on 1. α_{max} is the maximum angle that the door opens and α is the actual door angle computed as:

$$\alpha = \frac{1}{2}(\alpha_x + \alpha_y) \tag{4.20}$$

and

$$\alpha_x = \arccos \frac{x_{hand}}{l} \tag{4.21}$$

$$\alpha_y = \arcsin \frac{y_{hand}}{l} \tag{4.22}$$

where $l$is the door length.

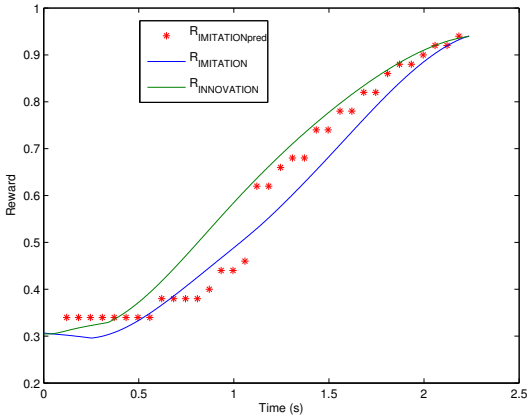


Figure 4.9: *Reward profiles of the complete action of opening a door. The blue line represents the reward profile for the human demonstrator. Black boxes represent the means of the reward for the imitative behavior. The red crosses represent the means of the reward profile for the innovative behavior. The dotted vertical lines represents the changes between behaviors. The vertical axis represent the reward value and the horizontal axis represents the states in which is divided each behavior. As it can be appreciated in the figure, the imitative behavior produces rewards similar to the human's and the innovative behavior produces rewards slightly higher.*

The robot can find a way to obtain a better total reward than the human if it is able to improve r_{door}, r_{knob}, $\hat{r}_{\alpha}$ and $r_{antiknob}$. Those terms represent the possibility of innovation. Please note that all $r^{\pi}(s,a,b)$ functions have to be normalized so its integration sums to 1 in order to be used with the Kulback-Leibler distance in (4.8) and (4.9). In Figure 4.9 the resulting rewards are plotted.

Discussion on the Reward Profile

As (Whiten et al., 2009) suggested, both children and chimpanzees attempt to emulate the goal of the action when imitating a behavior. Furthermore, some recent studies suggested that the main difference between humans and chimpanzees is the ability of over-imitation (Nielsen et al., 2014). Our proposal of using a reward profile to solve the correspondence problem in order to transfer a complex behavior from a human to a humanoid, is based on these previous neuroscience works and previous experiments performed in a humanoid standing up from a chair (González-Fierro et al., 2014a). However, we are not sure of what is the internal objective function that the brain is optimizing when performing a complex task sequence. We proposed a compound reward function that takes into account the position. Position in terms of closeness to the door, position in terms of distance from the hand to the door knob, position in terms of COM trajectory. However, the human brain may use also velocity, acceleration, jerk or even other factors we are not taking into account.

Although at first sight the proposed model of imitation and innovation may seem task dependent it is not. The generality comes from the definition of the reward profile. In fact, any behavior can be modeled, from simple ones to complex behaviors. The preference in the selection of a predefined reward function over a learned function like in inverse reinforcement learning (Abbeel and Ng, 2004; Ng and Russell, 2000) does not affect the general idea of comparing the behavior of a human and a robot in a common domain, which is the reward domain.

Moreover, it can be noted from Figure 4.9 that the innovative process does not improve the performance radically. The importance of the innovative behavior is not in the improvement quantity. It lies in the fact that the reward profile represents the behavior goal and, at the same time, a metrics to measure its performance. Therefore, since it is a behavior metrics, we can generate different movements, not only imitating the human but innovating a new behavior, which is better than the behavior demonstrated by the human.

4.3.4 Trajectory generation and optimization

Given a behavior and a state, a candidate state space trajectory $\xi_i = [\xi_{com}, \xi_{hand}]$ is computed as a cubic spline. A generalized cubic spline is defined as a piecewise polynomial fitted to a set of via points.

$$(t_0, \xi_0^*), (t_1, \xi_1^*)...(t_k, \xi_k^*) \tag{4.23}$$

where $\xi_i^* \in \mathbb{R}^N$ is the joint via points at time $t_i \in \mathbb{R}$.

Given these via points, there is a cubic trajectory that passes through these points and satisfy a smooth criteria.

$$\xi_i(t) = a_i(t - t_i)^3 + b_i(t - t_i)^2 + c_i(t - t_i) + d_i \tag{4.24}$$

where a_i, b_i, c_i, d_i are the polynomial coefficients optimized. The complete joint trajectory $q(t) \in \mathbb{R}^N$ is a concatenation of (4.24) over the time intervals.

$$q(t) = \begin{cases} \xi_0(t) & \text{if } t_0 \leq t < t_1 \\ \vdots & \\ \xi_k(t) & \text{if } t_{k-1} \leq t < t_k \end{cases} \tag{4.25}$$

Once the candidate trajectory is generated and the behavior that the robot should use is known using (4.12), the associated reward is computed using (4.14). The optimization process is performed using Differential Evolution algorithm (Storn and Price, 1997) with (4.8) and (4.9) as cost functions.

From the candidate state space trajectory, both locomotion and grasping patterns are obtained using the parametrized postural primitives.

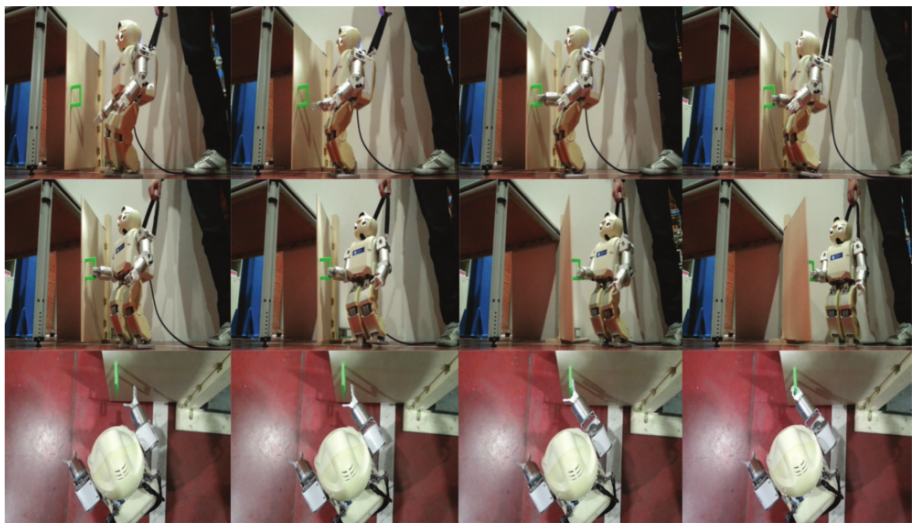


Figure 4.10: *Snapshots of the humanoid robot performing the task of opening a door from different views.*

For the locomotion pattern, ξ_{com} is used to calculate the ZMP reference, which is the input of the cart-table algorithm (Kajita et al., 2003a). For each episode, ξ_{com} is in fact a spline that connects two states, that in the case of the locomotion pattern, corresponds to one step. The location of this step is the ZMP reference. Therefore the original ξ_{com} trajectory is not the one followed by the robot COM. The real COM trajectory is generated by cart-table algorithm, and later, a kinematic inversion is used to compute the joint trajectory.

The grasping pattern is much more easy to implement in the robot. The desired trajectory ξ_{hand} corresponds to the robot end effector. The joint trajectory is computed using the humanoid Jacobian.

The humanoid initially detects the 3D position of the knob using the stereo cameras integrated in the robot. The knob is located using a simple color filter. Since the initial position of the robot and the door position is known, we compute the optimization process and generate the desired state space trajectories. This process is computed offline since the genetic algorithm consumes substantial computing resources. Once the desired trajectory is known, both locomotion and grasping trajectories are computed online inside the robot. The door angle is estimated by knowing the location of the robot with respect to the door hinge.

Some snapshots of the implementation with the real robot are shown in Figure 4.10.

Limitations and considerations

Regarding the implementation of our method in the humanoid robot, some considerations and limitations have to be taken into account. The first difference between the human and the robot performance is the smoothness of the walking pattern. In the case of the human, the COM barely swings when going backwards and the GMR output of the COM is almost a straight line. However, in the robot, the swing is much greater. This produces undesirable effects. The swing may produce a crash of the robot body with the door. Furthermore, it produces a back and forth movement of the door while the robot is moving backwards. To solve this problem, we simplify the computation by allowing the robot to decouple itself from the momentum of the door by relaxing the arm stiffness and having compliance along the plane of the door, meaning the hand can passively move along the plane of the door. For instance, when defining the behaviors, we select b_3 to be the moment when the robot is opening the door with the movement of its body, turning off the arm motors. For simplicity, we do not consider for the robot the case when the human is moving backward and pulling the door at the same time.

4.4 Discussion and Conclusions

The present chapter address one of the biggest questions of LfD, what to imitate (Argall et al., 2009). As some studies reveal (Gergely et al., 2002), the human brain understands the final goal of the action and reproduce it optimizing some kind of metrics, allowing to succesfully and elegantly reach the goal. Our proposal is to define this metrics as a reward profile which can be used as a basis of comparison between the human demonstrator and the robot. But an important feature of us humans is the ability to innovate new behaviors (Whiten et al., 2009; Nielsen et al., 2014). Therefore, we propose a reward base optimization process where the robot explores the neighbor solution space to come up with new behaviors which produce a better reward. Our framework allows a robot to create complex sequential behaviors taking into account the whole body movement.

We define a sequential multi-objective reward function for every sub behavior of the complete task. The optimization problem consist on generating a policy for the robot to obtain an episodic reward similar to the human's, achieving an imitative behavior. Refining this policy, we can generate new solutions which improves the reward profile to achieve an innovative behavior, more relevant to the robot circumstances. The result is a framework to generate whole-body motions for the robot which can be generalized to any movement that can be learned from demonstrations.

We carried out experiments in a real humanoid robot performing the task of opening a door to test our method.

The main contribution of this work is the solution to the correspondence problem

between a human and a robot in a common space, which represents a metrics to achieve the task goal, the reward space, and its application in a complex behavior formed by a sequence of actions. The reward space is formed by different components, depending on the objective of the action in every moment. This agrees with the theory of Minsky that proposes that our brain manages different resources that compete between each other to fulfil different goals (Minsky, 2006).

Chapter 5

Robust Control of Humanoid Models through Fractional Calculus

There is an open discussion between those who defend mass distributed models for humanoid robots and those in favor of simple concentrated models. Even though each of them has its advantages and disadvantages, little research has been conducted analyzing the control performance due to the mismatch between the model and the real robot, and how the simplifications affect the controller's output. In this chapter we address this problem by combining a reduced model of the humanoid robot, which has an easy mathematical formulation and implementation, with a fractional order controller, which is robust to changes in the model parameters. This controller is a generalization of the well-known PID structure obtained from the application of Fractional Calculus to control. This control strategy guarantees the robustness of the system, minimizing the effects from the assumption that the robot has a simple mass distribution. The humanoid robot is modeled and identified as a triple inverted pendulum and, using a gain scheduling strategy, the performances of a classical PID controller and a fractional order PID controller are compared, tuning the controller parameters with a genetic algorithm.

5.1 Introduction

In recent years, there have been a strong discussion between researchers on favor of using mass distributed models to model a humanoid robot, where the mass and inertia of every link is known, and those who prefer to use a simplified or concentrated mass model, where all robot dynamics are simplified and concentrated in the center of gravity (Arbulu, 2009).

Those who prefer a complete dynamic representation defend that it allows a more complex behavior, the model is more accurate and therefore the behavior of the model corresponds better to reality. In (Khatib et al., 2008) the authors performed a whole-body motion hierarchically dividing the control in tasks. Arbulú et al. (2010) used Lie algebra to obtain the humanoid whole-body dynamics and reduce the computation time. In (Kajita et al., 2003b), humanoid motion is accomplished by controlling the complete body model momentum. Hirukawa et al. (2006) presented an approach to control the humanoid gait in rough terrain through the Contact Wrench Cone. Other examples of complete dynamic representation are (Nagasaka et al., 1999; Suleiman et al., 2008; Yoshikawa and Khatib, 2009; Mistry et al., 2009, 2010a).

Many researchers make use of reduced dynamic models to control humanoids, some examples are the 2D and 3D linear inverted pendulum (LIPM) (Kajita and Tani, 1991; Kajita et al., 2001), cart-table (Kajita et al., 2003a) or the angular momentum pendulum model (Komura et al., 2005). A reduced model does not cover all dynamic behavior and non linearities of the real model, however, they are commonly used and many researchers have obtained good experimental results. In (Kaynov et al., 2009b) a humanoid robot is modeled as a double inverted pendulum and a stabilizer is studied. Mistry et al. (2010b) modeled a humanoid as an inverted pendulum of five links and a stand up task is performed. Other examples can be found in (Kim et al., 2007). In (Pan et al., 2004) a triple inverted pendulum is controlled using an evolutionary approach. In (Arisumi et al., 2008) a humanoid robot modeled as a 8 link manipulator is able to lift a box. Another examples of triple pendulum control use H_∞ (Tsachouridis, 1999) or fuzzy methods (Xiaofeng et al., 2009). The literature on humanoid reduced models is extensive (Sugihara et al., 2002; Sugihara and Nakamura, 2003; So et al., 2005; Kaynov et al., 2009b; Monje et al., 2009b, 2011a; González-Fierro et al., 2014a).

In this chapter an intermediate approach is proposed in order to obtain an advantageous control framework for a humanoid robot, which is to combine a humanoid reduced model with a Fractional Order Controller (FOC). The advantage is twofold. The use of a reduced model makes the formulation simple and fast to implement. The disadvantage of using a reduced model in comparison with a complete model is canceled by the use of a fractional controller. The fractional controller is able to cope with mismatches in the robot model, so even though the reduced model does not represent the robot accurately, the fractional controller makes the robot to

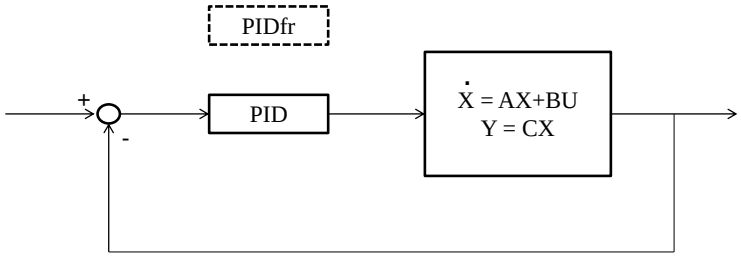


Figure 5.1: *Control system. The block PID is changed for the block PIDfr when the fractional order control strategy is used.*

behave successfully.

Specifically, we applied a FOC strategy for the control of a humanoid robot modeled as a triple inverted pendulum, in order to improve the system performance and overtake the mismatches produced between the simplified and real models of the robot (see Figure 5.1). This work were published in (González-Fierro et al., 2013e,d).

To test the robustness of our controller, we compared a classical PID controller with a fractional controller when the humanoid follows a trajectory of standing up from a chair. We overloaded the system adding 1 Kg. to every pendulum link, with the objective of evaluating the robot performance when there is a change in the mass of the model. The controller gains were optimized with Differential Evolution algorithm (Storn and Price, 1997).

5.1.1 Fractional Control in Robotics

The interest of fractional order control (FOC) is growing nowadays due to its applicability to engineering and science. Its practical applications make this field of research an emerging new topic. Even if they can be thought of as somehow ideal, they are, in fact, useful tools for both the description of a more complex reality and the enlargement of the practical applicability of the common integer order operators. Among these fractional order operators and operations, the fractional integro-differential operators (fractional calculus) are specially interesting in automatic control and robotics, among others.

The first mention to fractional control can be found in the work of Bode (Bode, 1940, 1945). He studied the design of an amplifier with the idea of producing a performance in the closed loop that were invariant to gain changes. The first application of fractional calculus in control led to its implementation in the frequency domain as a non-integer integral (Manabe, 1961).

Oustaloup et al. (1995, 1996, 1999, 2000) studied the fractional order algorithms for the control of dynamic systems and demonstrated the superior performance of the $CRONE$ (Commande Robuste d'Ordre Non Entier) method over the PID

controller. Podlubny (1999) proposed a generalization of the PID controller, namely the $PI^{\lambda}D^{\mu}$ controller, involving an integrator of order λ and a differentiator of order μ. He also demonstrated the better response of this type of controller, in comparison with the classical PID controller, when used for the control of fractional order systems. A frequency domain approach by using fractional order PID controllers has also been studied in (Monje et al., 2010).

There was an extensive effort in determinate the best tuning and auto-tuning methods for FOC. A relevant work in this area was performed by Monje (Monje et al., 2004, 2005; Monje et al., 2009a; Monje et al., 2010). Some other related publications are (Petráš et al., 2001; Lanusse et al., 2003; Caponetto et al., 2004; Barbosa et al., 2004b,a; Valério and da Costa, 2006; Padula and Visioli, 2011).

Fractional control is used extensively in control. Some examples that can be found in the literature are flexible transmissions (Valério, 2001; Monje et al., 2007; Delavari et al., 2013), active suspensions (Lanusse et al., 2003), heat control (Petráš and Vinagre, 2002; Petráš et al., 2002) or hydraulic actuators (Pommier et al., 2002; Chen et al., 2014).

Regarding humanoid robotics, it is not common to find this kind of controllers. In (Wen et al., 2014) a method to reduce the error in the localization of a humanoid robot is presented. Silva and Santos (2005) presented a method to control the gait of a small humanoid. They used FOC to track the interaction forces between the feet and the ground. Similarly to us, they optimized the controller gains using a genetic algorithm. Puga et al. (2006) proposed a humanoid control technique that uses force and position control with the integration of a fractional controller.

Fractional calculus also extends to other kinds of control strategies different from PID ones, but in the case study presented in this chapter we propose the use of the fractional order $PI^{\lambda}D^{\mu}$ controller as a robust alternative for the control of a humanoid robot simplified model based on the triple inverted pendulum.

5.2 Fractional order controllers

Fractional calculus is a generalization of the integration and differentiation to the non-integer (fractional) order fundamental operator ${}_a\mathrm{D}_t^{\alpha}$, where a and t are the limits and α ($\alpha \in \mathbb{R}$) is the order of the operation. Among many different definitions, two commonly used for the general fractional integro-differential operation are the Grünwald-Letnikov (GL) definition and the Riemann-Liouville (RL) definition (I. Podlubny, 1999). The GL definition is

$$ {}_a\mathrm{D}_t^{\alpha} f(t) = \lim_{h \to 0} h^{-\alpha} \sum_{j=0}^{[\frac{t-a}{h}]} (-1)^j \binom{\alpha}{j} f(t - jh), \tag{5.1} $$

where $[\cdot]$ means the integer part, while the RL definition is

$$ {}_a\mathrm{D}_t^{\alpha} f(t) = \frac{1}{\Gamma(n-\alpha)} \frac{\mathrm{d}^n}{\mathrm{d}t^n} \int_a^t \frac{f(\tau)}{(t-\tau)^{\alpha-n+1}} \mathrm{d}\tau, \tag{5.2} $$

for $(n-1 < \alpha < n)$ and where $\Gamma(\cdot)$ is the *Euler's gamma* function.

For convenience, Laplace domain notion is commonly used to describe the fractional integro-differential operation. The Laplace transform of the RL fractional derivative/integral (5.2) under zero initial conditions for order α $(0 < \alpha < 1)$ is given by

$$ \mathcal{L}\{{}_a\mathrm{D}_t^{\pm\alpha} f(t)\} = s^{\pm\alpha} F(s). \tag{5.3} $$

In theory, control systems can include both the fractional order dynamic system to be controlled and the fractional order controller. However, in control practice, more common is to consider the fractional order controller. This is due to the fact that the system model may have been already obtained as an integer order model in the classical sense.

5.3 Robust Postural Control of Humanoid Models

In a very simplified way, a humanoid robot can be dynamically modeled as a triple inverted pendulum. We modeled the HOAP humanoid robot as a triple pendulum, where the ankle joint of the robot corresponds to the first pendulum joint, the knee joint corresponds to the second one, and the hip joint corresponds to the third one (see Figure 5.2).

As it is represented in Figure 5.2, the pendulum masses are concentrated at the tip of every link and the link masses are negligible. The control action that allows every mass m_i to move a position q_i is the torque τ_i.

Since the task we wanted to simulate is a robot standing up from a chair, we have chosen a triple pendulum to model the humanoid. The reason why we decided this is because there is a direct mapping between the pendulum joints and the joints needed for the robot to stand up. It is a good trade between selecting a simple inverted pendulum model and a complete model.

5.3.1 State Space Representation of the Humanoid Model

In chapter 2 the equation of motion of the inverted triple pendulum (see Figure 5.2) was obtained, the general form is stated as:

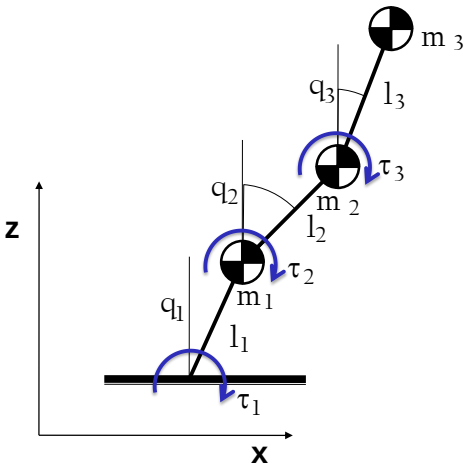


Figure 5.2: *An inverted triple pendulum.*

$$\begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{pmatrix} + \\ + \begin{pmatrix} 0 & c_{12} & c_{13} \\ c_{21} & 0 & c_{23} \\ c_{31} & c_{32} & 0 \end{pmatrix} \begin{pmatrix} \dot{q}_1^2 \\ \dot{q}_2^2 \\ \dot{q}_3^2 \end{pmatrix} + \begin{pmatrix} g_1 \\ g_2 \\ g_3 \end{pmatrix} \quad (5.4)$$

where the parameters of the matrices are explained in chapter 2.

The inverted triple pendulum of Figure 5.2 can be expressed as a dynamical system in the standard form:

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} \quad (5.5)$$

$$\mathbf{y} = C\mathbf{x} \quad (5.6)$$

where $\mathbf{x}$ is the state vector, $\mathbf{u}$ is the control vector and $\mathbf{y}$ is the output vector.

To obtain the representation of the triple pendulum system let us define the following state variables:

$$\mathbf{x_1} = q_1 \quad (5.7)$$

$$\mathbf{x_2} = \dot{q}_1 \quad (5.8)$$

$$\mathbf{x_3} = q_2 \quad (5.9)$$

$$\mathbf{x_4} = \dot{q}_2 \quad (5.10)$$

$$\mathbf{x_5} = q_3 \quad (5.11)$$

$$\mathbf{x_6} = \dot{q}_3 \quad (5.12)$$

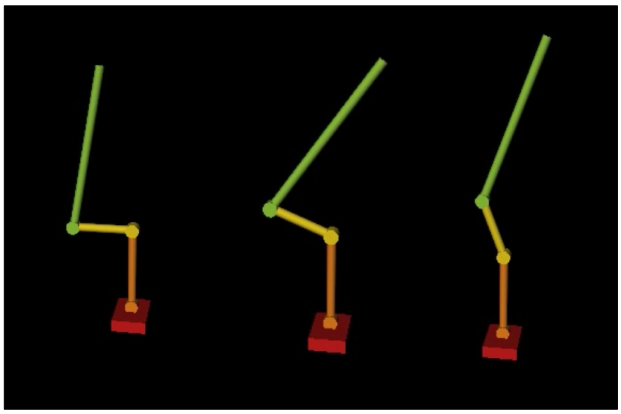


Figure 5.3: *The three positions of the system linearization. Every position is a point of linearization and defines a linear system.*

Taking this into account, and reordering (2.34), the matrices A, B and C can be obtained knowing that:

$$\dot{\mathbf{x}}_1 = \mathbf{x}_2, \quad \dot{\mathbf{x}}_3 = \mathbf{x}_4, \quad \dot{\mathbf{x}}_5 = \mathbf{x}_6 \tag{5.13}$$

$$\begin{pmatrix} \dot{\mathbf{x}}_2 \\ \dot{\mathbf{x}}_4 \\ \dot{\mathbf{x}}_6 \end{pmatrix} = \hat{f}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6) \tag{5.14}$$

where $\hat{f}$ contains non-linear terms of the state variables.

To get rid of the nonlinear terms, we linearized over the point of maximum acceleration, $\mathbf{x}_{i0}$ and $\mathbf{u}_{i0}$, using a Taylor expansion.

$$\dot{\widetilde{\mathbf{x}}} = \mathbf{A}\widetilde{\mathbf{x}} + \mathbf{B}\widetilde{\mathbf{u}} \tag{5.15}$$

where

$$A = \frac{\partial f}{\partial \mathbf{x}}\Big|_{\substack{\mathbf{x}=\mathbf{x_0} \\ \mathbf{u}=\mathbf{u_0}}} \quad ; \quad B = \frac{\partial f}{\partial \mathbf{u}}\Big|_{\substack{\mathbf{x}=\mathbf{x_0} \\ \mathbf{u}=\mathbf{u_0}}} \tag{5.16}$$

and $\widetilde{\mathbf{x}}_\mathbf{i} = \mathbf{x}_\mathbf{i} - \mathbf{x}_{\mathbf{i0}}$, $\widetilde{\mathbf{u}}_\mathbf{i} = \mathbf{u}_\mathbf{i} - \mathbf{u}_{\mathbf{i0}}$.

Since the desired trajectory has a wide variation, we selected three regions of linearization, obtaining three subsystems. We divided the desired trajectory in three regions and we chose the middle point of every region as the linearization point. In Figure 5.3 the selected linearization positions are shown. The result is three linear systems that are going to be controlled with standard and fractional order PID controllers using Differential Evolution to tune the controller parameters.

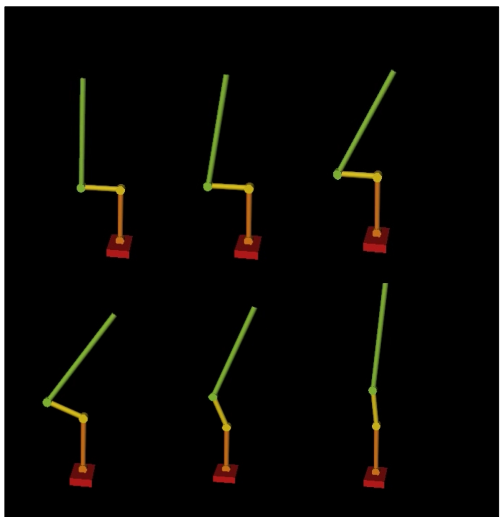


Figure 5.4: *Simulation of triple inverted pendulum trajectory.*

5.3.2 Selection of the Control Strategy

The first step to be able to control a humanoid robot model is the identification of this model. For this purpose we used Differential Evolution optimizer, computing a triple pendulum's Zero Moment Point (ZMP) trajectory and comparing it with the real ZMP measurement of the robot feet force sensors, minimizing the quadratic difference. The identification is based on the work of Tang et al. (2008).

The identification of the inverted triple pendulum has been already discussed in this thesis in chapter 3. The results are shown in the Table 3.1.

Taking these parameters into account and the three operating points previously stated (Figure 5.3), we obtained three linearized subsystems using equation (5.15). Each subsystem was controlled using an standard and a fractional order PID controller, whose gains $k_p, k_i, k_d \in R^{3\times3}$ and fractional orders λ, μ, were obtained using Differential Evolution. To change between systems, we used a Gain Scheduling strategy.

The desired trajectory was manually defined using three order splines and it simulates a stand up trajectory. The trajectory has been divided into three regions of two seconds, corresponding to the three subsystems each. In Figure 5.4 the simulated trajectory is shown.

Furthermore, to estimate the controller robustness, we overloaded the pendulum masses, adding 1 Kg to each link and comparing the new responses with those obtained from the nominal system.

5.4 Experimental Results

Following, the experimental results with the humanoid reduced model are shown and analyzed. It is also discussed the implementation of the FOC, which is approximated to a rational equation that behaves similarly to the controller.

5.4.1 Implementation of Fractional Controllers in a Humanoid Robot Model

Before introducing the method used for the tuning of the different controllers proposed in this work, some considerations on the implementation of the FOC $PI^{\lambda}D^{\mu}$ have to be taken into account. A good review regarding this topic is given by Monje et al. (2010).

The generalized transfer function of this controller is given by

$$c(s) = k_p + \frac{k_i}{s^{\lambda}} + k_d s^{\mu} \tag{5.17}$$

In order to implement a FOC in a real robot or in a simulation, fractional transfer functions are usually replaced by integer transfer functions with a behavior close enough to the one desired, but much easier to handle. There are many different ways of finding such approximations but unfortunately it is not possible to say that one of them is the best, because even though some of them are better than others in regard to certain characteristics, the relative merits of each approximation depend on the differentiation order, on whether one is more interested in an accurate frequency behavior or in accurate time responses, on how large admissible transfer functions may be, and other factors like these (Monje et al., 2010).

In this work, a frequency identification method performed by the MATLAB function *invfreqs* was used. With this method, a rational transfer function is obtained whose frequency response fits the frequency response of the original irrational transfer function within a selected frequency range. This method is chosen due to its accuracy in the frequency range of interest, which can be adjusted by selecting the number of poles/zeros of the rational transfer function.

5.4.2 Comparison between Classical and Fractional Controllers

In Figure 5.5 and Figure 5.6 the control system with the PID and the fractional PID are shown.

Differential Evolution algorithm produces random values of the controller gains that are used to simulate the system in Figure 5.5 and Figure 5.6. The fitness function to minimize is the quadratic difference between the system output and

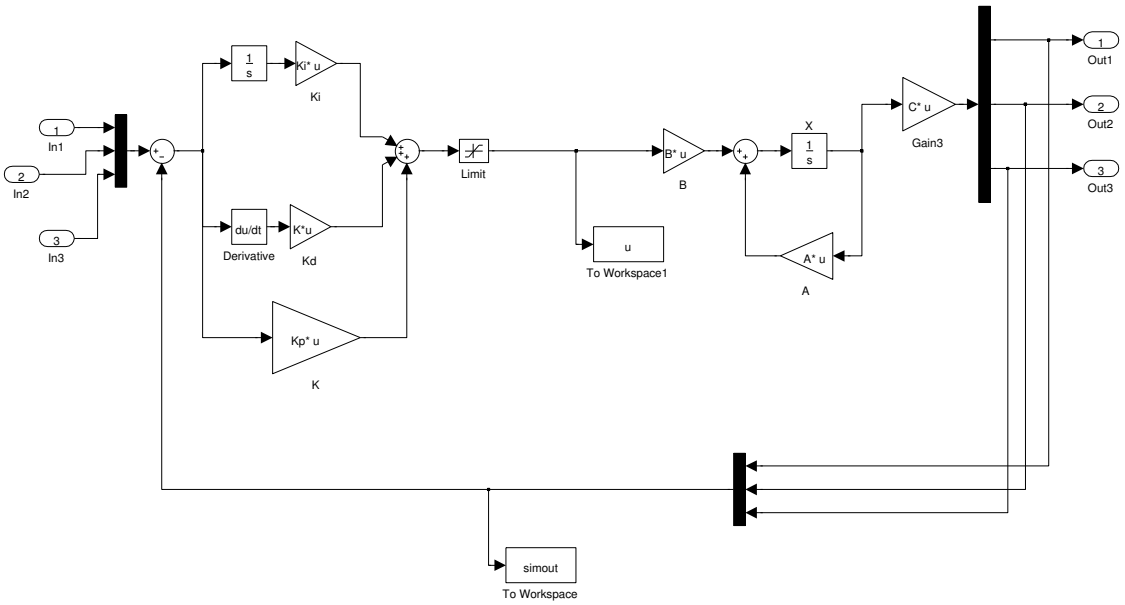


Figure 5.5: *Control system for the triple inverted pendulum using a PID.*

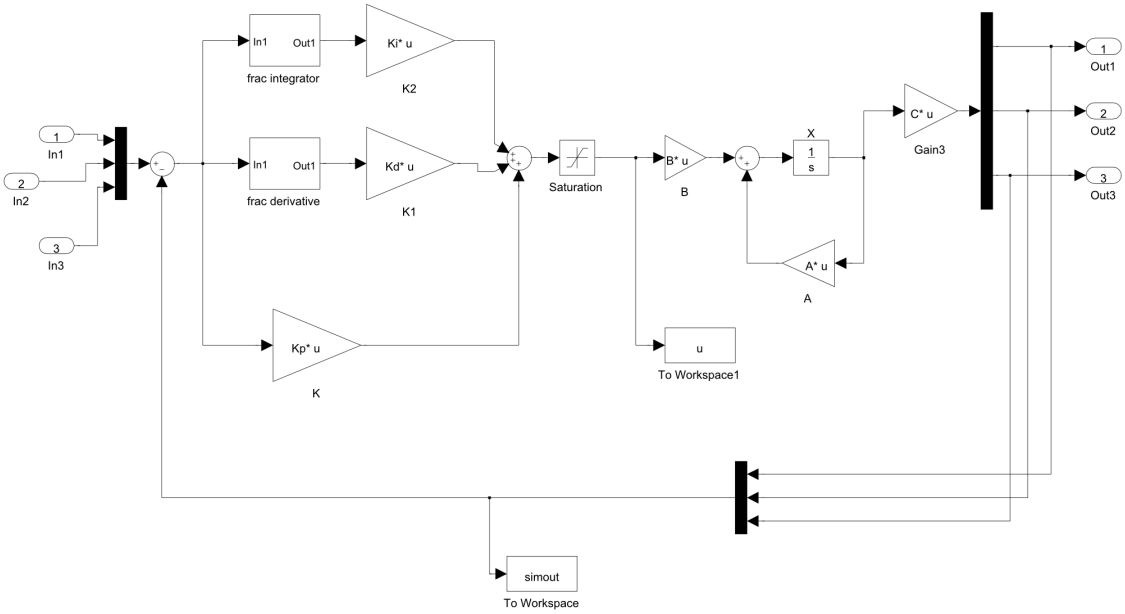


Figure 5.6: *Control system for the triple inverted pendulum using a fractional order PID.*

the reference. The best member of every iteration is mutated and evaluated again until a final value of the fitness function is reached or a total number of iteration is passed. In our case, the final value is 0 and the maximum number of iterations is 50. All simulations have been performed in MATLAB, using Runge-Kutta solver and a sampling time of 1 ms.

This is done for every subsystem with the standard PID gains and with the fractional order PID gains and λ and μ orders.

To approximate the behavior of a fractional controller, we used the frequency identification method *invfreqs* provided by MATLAB. The chosen crossover frequency was 0.001 rad/s and we approximated the behavior of the fractional controller for 4 decades.

The approximation of the FOC is a rational expression of order 8. This expression is evaluated in MATLAB and substituted in the block $PI^{\lambda}D^{\mu}$ of Figure 5.1.

Next the results obtained are presented. For the sake of space, we are just presenting the parameters of the classical PID and the fractional order PID controller for the first region, similarly obtaining the corresponding controllers for the other two regions. Next the parameters of the classical PID controller are presented:

$$k_{p1} = \begin{pmatrix} 1205.9 & 7266.4 & 26866 \\ 5567.1 & 1029.2 & -7054.1 \\ 6222.7 & 1181.5 & -620.23 \end{pmatrix}$$

$$k_{i1} = \begin{pmatrix} -260.06 & 9.552 & 43.524 \\ -338.64 & -3.1912 & -394.46 \\ 330.96 & 8.3799 & 156.98 \end{pmatrix}$$

$$k_{d1} = \begin{pmatrix} -249.66 & 1325.5 & -956.91 \\ 1072.9 & -319.45 & 1259.6 \\ -344.21 & 541.02 & 379.57 \end{pmatrix}$$

Next the parameters of the FOC for the first region are presented:

$$k_{p1}^{fr} = \begin{pmatrix} 404.7 & -305.2 & -782.6 \\ 1887.7 & -102.1 & -6281.7 \\ 1097.3 & -13.2 & 417.5 \end{pmatrix}$$

$$k_{i1}^{fr} = \begin{pmatrix} -13129.1 & 13074.1 & -5581.2 \\ -1185.4 & -118.5 & 1321.5 \\ -1971.7 & -933.6 & 12007.2 \end{pmatrix}$$

$$k_{d1}^{fr} = \begin{pmatrix} 10891.5 & 6320.6 & 1687.9 \\ -3646.4 & 1252.1 & 7721.2 \\ 1025.5 & -943.7 & -1851.3 \end{pmatrix}$$

$$\lambda_1 = 0.595 \;\; \mu_1 = 0.432$$

The results obtained for the three regions are presented in Figure 5.7, 5.8 and 5.9, respectively. As it can be seen, the FOC maintains the system stability even when a significant mass mismatch appears in the model. On the contrary, the responses with the standard PID controller are unstable for some of the joins when the system is overloaded. As a consequence, we can guarantee the robustness of the control system to uncertainties in the model, compensating this way the effects of using a reduced robot model for control purposes. Using this technique the modeling of the humanoid robot takes a secondary importance, since the system relies on robust control to cope with the model mismatches.

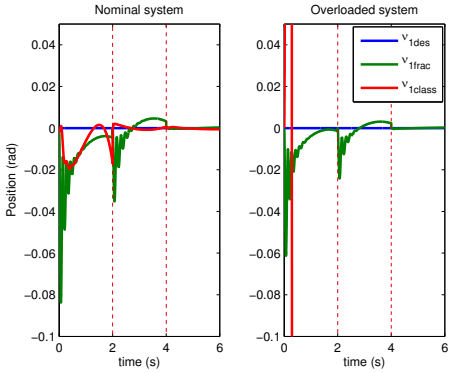


Figure 5.7: *System response for joint 1 for the nominal (left) and overloaded (right) subsystem. In blue is the desired trajectory, in green the trajectory with the fractional order controller and in green the trajectory with the standard PID. In dotted red the limits of the three linearization regions.*

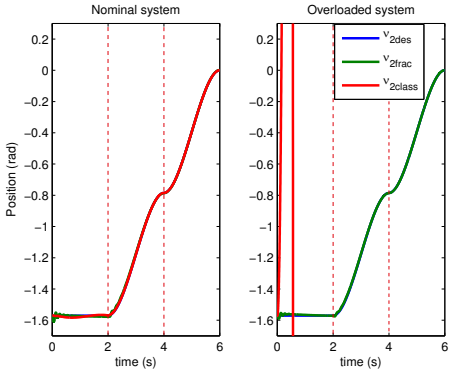


Figure 5.8: *System response for joint 2 for the nominal (left) and overloaded (right) subsystem. In blue is the desired trajectory, in green the trajectory with the fractional order controller and in green the trajectory with the standard PID. In dotted red the limits of the three linearization regions.*

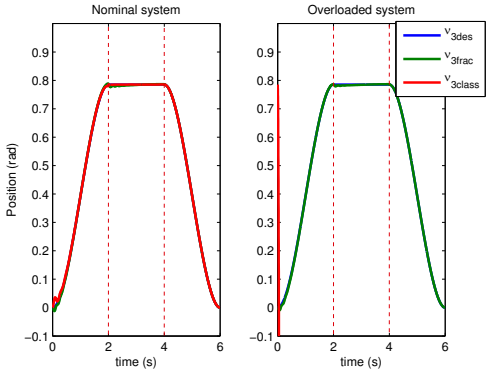


Figure 5.9: *System response for joint 3 for the nominal (left) and overloaded (right) subsystem. In blue is the desired trajectory, in green the trajectory with the fractional order controller and in green the trajectory with the standard PID. In dotted red the limits of the three linearization regions.*

5.5 Discussion and Conclusions

This chapter addresses the problem of modeling and controlling a humanoid robot model. Some authors prefer to use mass distributed humanoid models which involve complex mathematical equations, time consuming computation and difficult identification methods (Hirukawa et al., 2006; Khatib et al., 2008; Mistry et al., 2010a). However, they have the advantage to be more precise and they can be used for different tasks.

On the other hand there are many scientists that prefer mass concentrated humanoid models, which have easy mathematical models and are faster to compute (Kajita and Tani, 1991; Kajita et al., 2001, 2003a; Komura et al., 2005; Monje et al., 2011a). As a disadvantage, they are not so accurate as the complete models. However, to a greater or lesser extent, both distributed and concentrated approaches have model parameter mismatches between the real robot and the model.

The work presented in (González-Fierro et al., 2013e,d) proposes an alternative to this problem. Combining a reduced model with a robust control method like a fractional order controller, benefits from an easy mathematical framework and a fast computation. At the same time, the controller absorbs the errors between the model and the real robot. The effect of mass mismatches between the real and the simplified model of the humanoid is compensated, to a significant extent, by the fractional order PID controller, which ensures the robust response of the whole system during a motion when a mass increase of 1 Kg is considered in each tip.

After comparing the behavior of the humanoid model when performing a standing up movement using the standard PID controller and the fractional order one, it is concluded that, using Differential Evolution as gain optimizer, both controllers track the reference satisfactorily for the nominal case. However, when the robot is

overloaded, only the fractional order controller guarantee the stability of the system.

Chapter 6

Control of Humanoid Robots Executing Complex Tasks

This chapter deals with the planning and execution of a complex task ordered to a humanoid robot. The robot has to be able to execute high level postural tasks in the presence of a cluttered environment. The motion execution must be soft and stable and, at the same time, the robot has to be able to successfully avoid obstacles in the environment. First, the robot has to identify the environment and the obstacles. Second, the robot has to be able to move from the initial point to the final point performing a set of postural movements. The postural task is performed in two levels, a postural planning, which off-line computes the safe and stable postural movement that allows the robot to navigate through the environment, and an online postural control, which has to do with the execution, control and disturbance rejection that makes possible the fulfillment of the task. This chapter encompasses the learning strategies explained in chapters 3 and 4, the control method of chapter 5, while using methodologies of chapter 2. The task selected as an example is a robot that starts seated on a chair, stands up, walks avoiding obstacles until it reaches a door, opens the door and leaves the room. This chapter also gives a practical significance to this thesis.

6.1 Introduction

The future robots will need to understand the order that is given to them, where are the obstacles in the environment it has to avoid to perform that order and what are the movements or actions it has to generate to successfully complete the task.

It all requires a set of multiples components like robot architectures, control strategies, computer vision algorithms, motion planning methods, learning methodologies, etc.

This chapter does not intend to give an answer to everything the robot needs to perform in a high level order like *"stand up from a chair, walk to the door and open it"* or in any other high level order. The purpose of this chapter is to give an ultimate significance to this thesis by proposing a method to plan and control a humanoid robot from a postural point of view. This is not a definitive architecture for postural control in humanoids, but a framework where almost all pieces of the work developed in this thesis are put together.

6.1.1 Biological Foundation of Postural Planning and Control

Human posture control is governed by the Central Nervous System (CNS) and it is developed during the firsts years after birth. Toddlers learn to maintain balance by means of the experience acquired by trial and error.

The CNS is composed by four groups of elements: receptors, processors, effectors and communication channels. The receptors captures the sensorial information by means of the visual, vestibular and articular nervous systems. The information transmitted through neurons of the peripheral nervous system arrives at the CNS where is processed. In the information processing intervenes the spinal cord, the cerebral trunk and the motor cortex depending on what reaction has to be triggered. The resulting action is transmitted to the effector system by means of the efferent neurons and finally the musculo-skeletal system generates the body movement.

There are several levels of nuero-musculo integration when a movement is produced. Furthermore, at any given moment, individual groups of muscles may have different simultaneous goals to accomplish (Winter, 2009). Regarding balance maintenance, there have been advances in the understanding of balance control during walking (MacKinnon and Winter, 1993), body response to arm voluntary movements (Eng et al., 1992) and body balance recovery mechanisms (Rietdyk et al., 1999).

For neuroscientists like Prof. Horak, human postural control is intimately related to body orientation and equilibrium maintenance (Horak, 1987; Horak and Macpherson, 1996; Horak, 2006). Postural orientation involves the active alignment of the trunk and head with respect to gravity, support surfaces, the visual surround and internal references. Sensory information from somatosensory, vestibular and

visual systems is integrated, and the relative weights placed on each of these inputs are dependent on the goals of the movement task and the environment. Postural equilibrium involves the coordination of movement strategies to stabilize the center of mass during disturbances of stability. When the body reacts to external perturbations, there are two postural activities at the muscular activation level that occurs, as it was observed by (Nashner and McCollum, 1985). The balance is maintained by an ankle activity and a hip activity that coordinate the body to control the posture.

Generally, a human postural task is composed by two main types of control loops. The first one is a feedback control loop that follows the postural motion reference planned by the CNS and reacts to disturbances during slow motions. Furthermore, there is a complementary open loop control system that allow fast reflexes and corrective reactions. These postural strategies have led to the development of bio-inspired planning and control strategies of postural motion for robots.

6.1.2 Postural Tasks in Humanoids Robots

An intense research work has been done since the beginning of robotics science to be able to model and control a humanoid robot (Schaal, 1999; Katic and Vukobratovic, 2003; Peters et al., 2003; Kajita et al., 2003b; Calinon et al., 2007; Sentis, 2007; Khatib et al., 2008; Siciliano and Khatib, 2008; Argall et al., 2009). A humanoid robot is a complex machine which tends to have large degrees of freedom and is characterized with a high capability of movement.

One basic characteristic that humanoid robot have is the detachment to the ground. It is indeed the robot's contact to the ground which makes the gait possible. Therefore, robot movement is not only determined by the position and orientation of its kinematic chains but also by the position and orientation of its body with respect to an inertial frame. The position and orientation of the humanoid is usually represented as a moving frame with respect to the inertial frame, usually addressed as floating base. A humanoid robot can be treated as a platform composed by 4 actuated kinematic chains of n DoF each, representing both arms and legs, and a virtual 6 DoF joint representing its floating base.

There is an intense literature in this area (Kuffner Jr et al., 2002; Kuffner et al., 2005; Yoshida et al., 2005). Prof. Sentis and colleagues performed an intense work studying postural control in humanoid robots (Khatib et al., 2004a, 2008; Sentis and Khatib, 2005; Sentis, 2007; Sentis et al., 2010). Basing their research in the operational control of Khatib (1987), they dig into the problem of performing locomotion and manipulation tasks while hierarchically performing different postures and maintaining the balance. Their proposed controller is able to perform the desired task, maintain balance and, at the same time, produce postures which implies a minimal effort for the robot.

A similar approach is presented in (Monje et al., 2009b, 2011a) where a humanoid robot collaborates with a human. The authors proposed a system based

on two control modules, a collaborative control loop devoted to the generation of stable motion patterns for a robot, given a specific manipulation task, and a posture stability control loop, which guarantees the stability of humanoid for different poses determined by motion patterns.

Martínez de la Casa et al. (2013) studied the postural control problem in terms of similarity with the postural control system of human beings. They analyzed the human postural control system (Martínez de la Casa, 2012), which is composed by a closed loop system in charge of reacting to posture disturbances and an anticipative open loop system able to preview the consequences of undesirable events and, at the same time, trigger corrective actions. They proposed an architecture able to predict postural corrections using neuro-fuzzy algorithms which are implemented in the full-size humanoid TEO.

In the work of Arbulu et al. (2008), the authors proposed a method to generate a dynamic motion from a initial static posture to a final static posture without losing stability.

There is an intense work related to the problem of rejecting perturbations to maintain the humanoid's balance. Some works used inverted pendulum structures to model the robot and proposed different control strategies (Stephens, 2007; Bonnet et al., 2009; Kaynov et al., 2009a; Kajita et al., 2010). Other approaches takes into account the whole-body structure of the robot (Stephens and Atkeson, 2010; Sentis and Khatib, 2004; Khatib et al., 2004a, 2008).

6.1.3 Proposed Architecture

In Figure 6.1 the proposed architecture is presented. The first step is the generation of the high level order: *"stand up from a chair, walk to the door and open it"* which defines a complex task for the robot to achieve. This is the main objective that, in the case of a humanoid robot, is a postural task.

The next step is an environment analysis. This analysis allows the robot to know where are the obstacles located and where is the free space, in order to navigate through it safely.

Then a postural planning is needed. Our proposal consist on 3 steps: a global path planning, which gives an initial approximation of the humanoid's COM path; a generator of postural skills, which defines the posture primitives (initial, intermediate and final postures of the complete action) and the skills (transition between two posture primitives); and finally a postural motion planning, which computes the desired movement that the robot needs to perform to achieve the goal, optimizing a certain performance metrics (the reward).

Finally, the robot executes the movement in the postural control phase. It includes a whole-body postural control, which defines the control strategy to follow the desired trajectory, and a correction of postural disturbances, which corrects the movement in the case of perturbations.

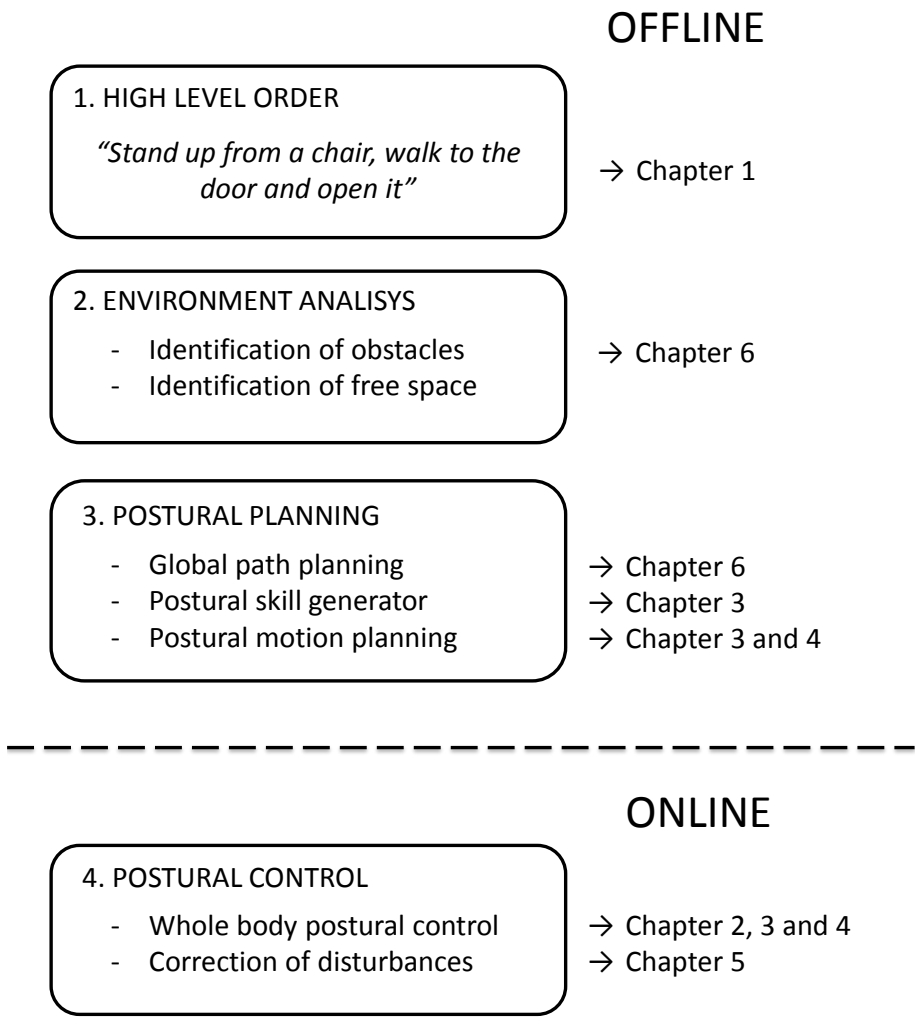


Figure 6.1: *Complete proposed architecture for the humanoid behavior. It consist on four modules. The high level task, an environment analysis, an offline postural planning and an online postural control.*

There is a close relationship between the architecture of Figure 6.1 and the development of this thesis. The part of environment analysis is presented in this chapter 6, as well as the global path planning. The generator of postural skills is mentioned in chapter 3. The development of the postural motion planning is deeply studied in chapters 3 and 4. The whole-body postural control is discussed in chapter 2, and finally the correction of postural disturbances is addressed in chapter 5. This chapter gives an ultimate sense to this thesis and, at the same time, encompasses all techniques developed during this work in one single common thread.

6.2 Environment Analysis

If robots are going to share our living space, they have be able to move in clustered environments with reliability, avoiding fixed and moving obstacles. Smart environments are an easy solution to this matter (Coradeschi and Saffiotti, 2006; Pierro et al., 2009c; Jardón et al., 2011; Pierro et al., 2012; González-Fierro et al., 2013b).

The main objective of a smart environment is to reduce the complexity of a determined task and to help the robot to perform this task. First, by installing external cameras or sensors, a global and more accurate view of the environment can be perceived. It can help the robot to free computational resources by eliminating the need of performing costly computational methods like SLAM, while at the same time, it reduces the dependence of inner robot sensors. In addition, a smarter high-level action planning can be performed by anticipating the location of fixed or moving obstacles.

Next, all the theoretical principles used to integrate the perception feature into the humanoid architecture will be explained.

6.2.1 Environment Perception

The very first goal of the robot is to understand its surroundings in order to construct a valid path and therefore to achieve the desired goal successfully. For this study, a RGB-D Asus Xtion Pro Live camera has been installed in the upper part of a room with a certain angle. The role of the camera is to acquire a valid model of the supporting plane (floor) and subtract all the possible obstacles that could interfere during the path planning.

The sensor provided by the RGB-D manufacturer is ready to detect indoor 3D points where errors increase quadratically from a few millimeters at 0.5 m to about 4 cm at the maximum sensor range (Khoshelham and Elberink, 2012). The camera resolution is 640 x 480 pixels and the output video frame rate may vary between 25 and 30 Hz. The camera provides for each frame a depth map, color map and infrared map. Infrared information has been used on previous researches to fix lens distortions and other optical aberrations by means of a chessboard borders

detection, but for this project the standard distortion matrix will be used to simplify the problem and therefore only color and depth sources are used to determine the architecture of the surroundings.

The complete camera flow acquisition is programmed using OpenNI library (OpenNI, 2012) while cloud operations are done using Point Cloud Library (Rusu, 2010) with adhoc interface for the visualization. All the architecture follows a modular structure to facilitate the addition of new filters or improvements.

6.2.2 Downsampling the Data

The camera provides approximately 8M points per second that have to be managed somehow. Due to the significant amount of data that has to be interpreted, it is necessary to reduce the order of magnitude rejecting points in order to let the robot react in real time. In this case, a VoxelGrid filter approach is used. A VoxelGrid represents a small 3D box in space. All the points inside a VoxelGrid are approximated to the centroid reducing the amount of data. The bigger the VoxelGrid, the smaller the data obtained and so the faster the operations (see Figure 6.2).

6.2.3 Supporting Plane Extraction

In order to determine the obstacles, the robot has to be able to differentiate between floor and non-floor objects. This distinction can be obtained using depth map perception and 3D constraints. In this study, normals extraction is first performed and then the supporting plane is acquired.

Normals Computation

Surface normals are very useful to understand the geometry surfaces and also to reconstruct and understand the point cloud. Normal extraction means estimating the normal of a plane tangent to the surface. This process can be as simply as computing the cross product between a pair of nearest points for any query point. This technique is not recommended for noisy point clouds because results are inconsistent and extremely variable (Dey et al., 2005).

The normal extraction was stated as a least-square plane fitting estimation problem. With the analysis of the eigenvectors and eigenvalues of a covariance matrix C created from the nearest neighbors to the query point, it is possible to determine the surface normal

$$C = \frac{1}{k}\sum_{i=1}^{k}(p_i - \bar{p})(p_i - \bar{p})^T, \quad C \cdot \vec{v_j} = \lambda_j, \quad j \in \{0, 1, 2\} \tag{6.1}$$

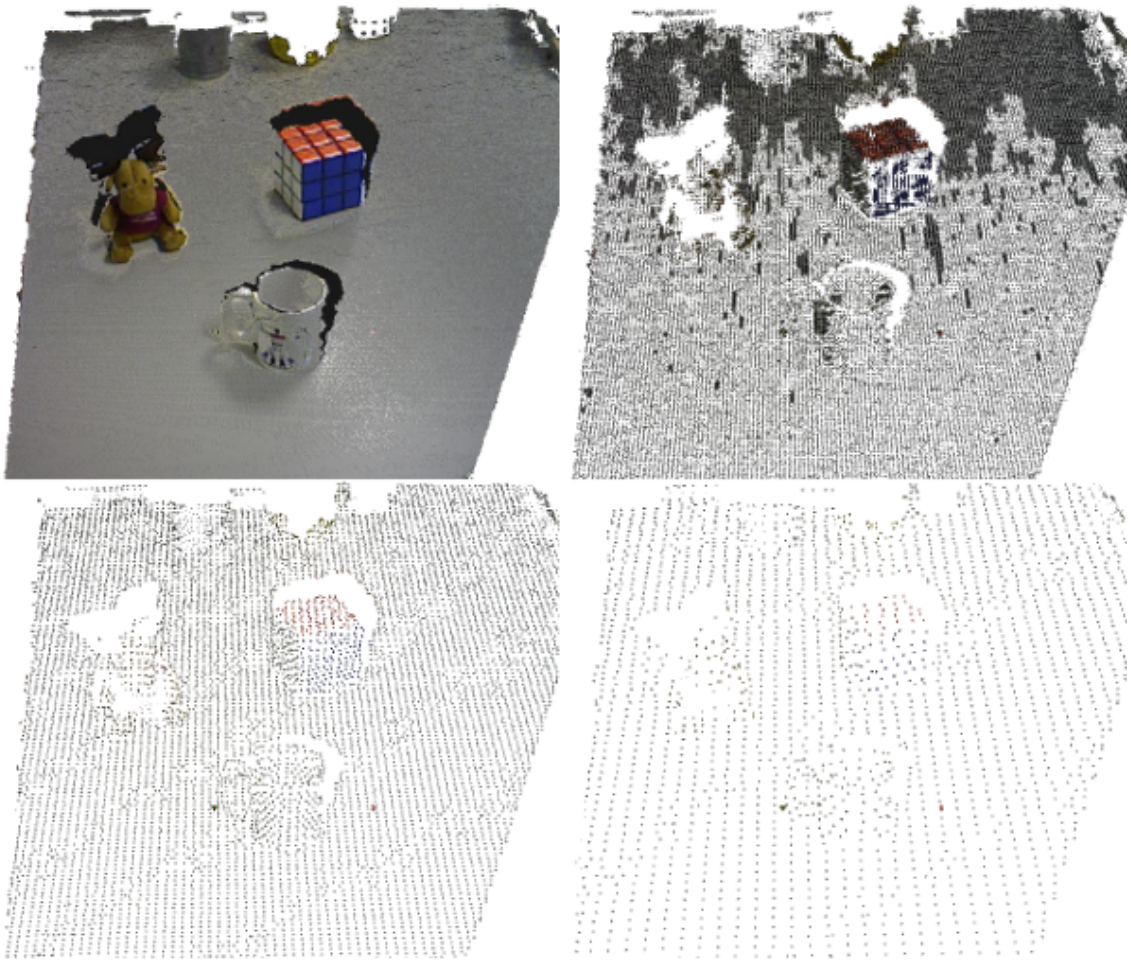


Figure 6.2: *Down-sampling the point cloud using voxelgrids at different sizes. Leaf sizes are 0.005m, 0.01m, 0.05m and 0.08m respectively.*

where λ_j represents the j^{th} eigenvalue and $\vec{v_j}$ is the j^{th} eigenvector of C. Normals orientation is solved applying the viewpoint constraint given by

$$\vec{n_i} \cdot (v_p - p_i) > 0 \tag{6.2}$$

knowing that the primary surface direction is given by the third eigenvector

$$\vec{n_i} = \pm\vec{v_3} \tag{6.3}$$

Plane Estimation

Once the normal point cloud is computed for each frame, the next step is to determine the plane which fits best in the point cloud. To perform this operation RANSAC algorithm (Random Sample Consensus) is applied with a geometric model of a plane. RANSAC iterates over the whole point population and estimates the parameters of the most suitable plane. To do this, the method distinguishes between inliers (points that fit the model) and outliers (points outside the model) iteratively. For this study, 70% of points are required to be part of the floor and the rest are supposed to be obstacles.

The output of the plane estimation are the classical four parameters that generates a plane equation

$$Ax + By + Cz + D = 0 \tag{6.4}$$

With this information in mind, it becomes straightforward to split up the original depth map into two maps: the first one containing points inside the supporting plane and the other one gathering all the obstacles around the robot.

6.2.4 Obstacle Clustering

The last step in the perception architecture consists on converting the outliers of the previous section into bounding boxes. Those groups are then introduced into the path planning algorithm as geometric constraints. In order to track each obstacle, a clustering process is performed.

Euclidean Clustering algorithm is based on K-means using a KD-tree to boost the performance of the process. It is similar to flood-fill algorithm in image processing. The followed steps are stated in the Algorithm 4.

6.3 Postural Planning

The postural planning involves the postural strategy the robot has to follow to successfully achieve the desired behavior or the high level order given by the human. These postures can be seen as the postural configurations the robot uses to achieve a specific goal. Regarding displacement activities they include sitting, standing up,

Algorithm 4 EUCLIDEAN CLUSTERING

```
1: input point cloud data set P
2: build KD-tree (P)
3: empty list of clusters C
4: empty queue of points to be checked Q
5: for all p_i in P do
6:     add p_i to Q
7:     for all p_i in Q do
8:         P_i^k = search p_i^k of neighbours of p_i with r < r_threshold
9:         for all p_i^k in P_i^k do
10:            if Q not contains p_i^k then
11:                Q ← p_i^k
12:            end if
13:        end for
14:    end for
15:    C ← Q
16: end for
17: check p_i in C
```

standing still, walking, running, jumping, lying down, and bending down. Regarding manipulation activities they include grasping, pushing, pulling and manipulation. All these activities can be seen as postural activities if they are analyzed from the postural point of view, we are not only interested in the activity per se, but in the posture the robot needs to acquire to complete the activity with a good performance. This good or bad performance can be measured by means of the reward, as it was stated in chapter 4 of this thesis. The reward can integrate in a single metric stability, human movement likeliness, robot safety, effort, movement continuity or movement feasibility.

The whole body postural planning of a humanoid robot follows three steps: first there is a global path planning that gives an initial approximation of the COM path the robot has to follow to successfully achieve the desired goal. Second, to define the correct set of postural skills is needed, which includes the skills the robot has to generate to advance through the complex task in a proper way. The final step is the computation of a complete postural motion planning, meaning the interpolation between postural primitives while fulfilling the goal, satisfying the robot constraints and optimizing a behavior reward.

6.3.1 Global Path Planning

Two of the most successful path planning algorithms include Probabilistic Roadmap Method (PRM) (Kavraki et al., 1996) and Rapidly-exploring Random Tree (RRT)

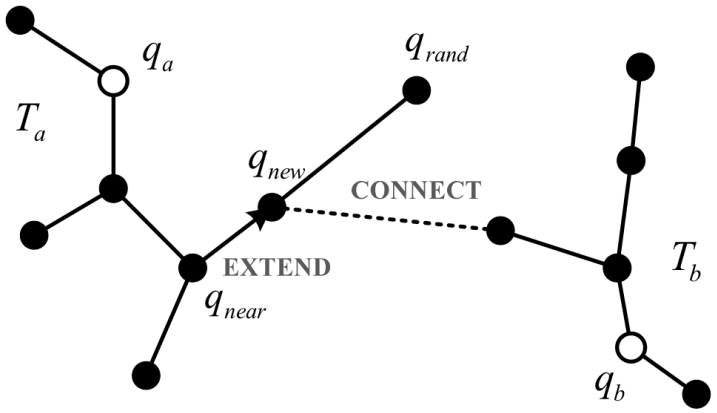


Figure 6.3: *Process of RRT-Connect merging two RRT trees.*

(LaValle, 1998). We used the Bi-directional Rapidly-Exploring Random Trees Algorithm (Bi-RRT) (Kuffner and LaValle, 2000). A reference book regarding this matter is (Sciavicco et al., 2009).

The Rapidly-exploring Random Tree (RRT) was introduced in (LaValle, 1998) as an efficient data structure and sampling scheme to quickly search high-dimensional spaces that have both algebraic constraints (arising from obstacles) and differential constraints (arising from non-holonomy and dynamics).

The key idea is to bias the exploration towards unexplored portions of the space. In the paper (González-Fierro et al., 2013b), we presented an approach that is tailored to problems in which there are no differential constraints, and the problem can be expressed in *C-Space*. If there are external sensors that help the robot to make a map of the environment, it is not necessary to do time consuming and complex SLAM. All it is needed is a simple path planning.

As illustrated in Figure 6.3, the RRT-Connect works by extending and connecting two trees towards each other. Two trees (T_a and T_b), rooted at two different milestones (q_a and q_b), either be local tree or global tree, are maintained at all times until they are connected to each other and merged into one single RRT.

At every time step, a random configuration q_{rand} is sampled inside the free space C_{free}. The **EXTEND** function determines the nearest configuration of q_{rand} in the current tree T_a, denoted as q_{near}. After that, T_a extends in the direction of q_{near} for one step, generating a new configuration q_{new}, using a fixed incremental distance.

From this point three situations can occur:

1. Reached: $q_{new} = q_{rand}$ and the configuration is directly added to the tree because it already contains a vertex within a fixed distance of q_{rand}.

2. Advanced: $q_{new} \neq q_{rand}$ and it is added to the tree.

3. Trapped: the new configuration is rejected because it does not belong to the free space C_{free}.

At the same time, a tree T_b starts to grow from the goal configuration q_b. The other tree T_b uses another procedure called **CONNECT** to extend towards q_{new} as much as possible. The **CONNECT** procedure is a greedy function that can be considered as an extension of the **EXTEND** function. Instead of attempting to grow towards the sample q_{rand}, it iterates the **EXTEND** function towards q_{new}, until a configuration is reached or there is an obstacle.

If T_b can successfully reach q_{new}, the two trees are connected and merged into one single RRT. It is all explained in Algorithms 5, 6, 7 and 8.

Algorithm 5 EXTEND(T,q)

```
1: q_near ← Nearest-Neighbour(q,T)
2: if new_config(q, q_near, q_new) then
3:     add_vertex(q_new)
4:     add_edge(q_near, q_new)
5:     if q_new = q_rand then
6:         return reached
7:     else
8:         return advanced
9:     end if
10: else
11:     return trapped
12: end if
```

Algorithm 6 BUILD RTT

```
1: init T(q_init)
2: for all k = 1:K do
3:     q_rand ← Rnd.Cfg
4:     EXTEND(T, q_rand)
5: end for
6: return T
```

In every iteration, one tree is extended to the new configuration and the other attempts to connect its nearest branch to the other tree reaching. Then, the roles are reversed by swapping the trees. This causes both trees to explore the free space while attempting to connect each other.

Algorithm 7 CONNECT(T,q)

1: **do**
2: $S \leftarrow$ **EXTEND**(T, q)
3: **until not** $(S = advanced)$
4: **return** S

Algorithm 8 RRT CONECT PLANNER(*init,goal*)

1: **init** $T_a(q_{init}), T_b(q_{init}, q_{goal})$
2: **for all** k = 1:K **do**
3: $q_{rand} \leftarrow Rnd.Cfg$
4: **if not EXTEND**$(T_a, q_{rand}) = trapped$ **then**
5: **if CONNECT**$(T_b, q_{new}) = reached$ **then**
6: **return path**(T_a, T_b)
7: **end if**
8: **end if**
9: **swap**(T_a, T_b)
10: **end for**
11: **return** *failure*

6.3.2 Postural Skill Generator

This module generates the skills the robot needs to follow to be able to plan the postural behavior.

Primitive postures can be defined as postures that does not imply a movement. Some examples can be standing still, seated, laid down, bended down, etc. The robot can acquire any of these postures with different body configurations and with different purposes. The robot will not have the same posture when it is going to open a door than when it is going to grasp an object from a table.

The transition between two postures can be defined as a skill. Some examples of skills can be sitting down, standing up, walking, running, jumping, lying down, etc.

The main difference between posture skills and normal skills is the focus on the posture. There are thousand of ways of walking, in some cases the robot may walk imitating the human postural behavior, it can walk lowering the COM to be more stable or it can even walk while optimizing some index that requires some postural configuration.

6.3.3 Postural Motion Planning

The postural motion planning can be understood as the computation of the postural trajectories generated in a robot to successfully achieve a skill. This term shares a significant similarity with the term humanoid motion planning or simply

motion planning (Kuffner Jr et al., 2002; Kuffner et al., 2005; Yoshida et al., 2005). Motion planning is defined as the necessary plan that enable the robot to execute the assigned task without colliding with the environment obstacles (Sciavicco et al., 2009).

However, usually motion planning does not pay attention to the postural configuration of the robot. The postural motion planning is deeply involved not only with the postural transition that the robot achieves when performing a behavior, but also with the level of performance, that in this work is represented by means of a reward profile. Therefore, the postural motion planning is indeed a type of motion planning but with an emphasis in the postures.

During the development of this thesis, several methods of postural motion planning have been proposed (González-Fierro et al., 2013a, 2014a). These methods are applied to only one dynamic posture that can be any general posture, but in these works is the task of standing up from a chair. They are widely discussed in chapter 3.

Besides, it was also developed a generalization of the previous methods to a set of sequential behaviors. They included walking forward and backwards, turning, grasping a door knob and open a door. They are addressed in chapter 4 as a general formulation and also in chapter 2.

There is also a common thread in all postural motion planning methods proposed during the realization of this work. It is the satisfaction of an optimality criteria which guides the postural motion planning. In traditional learning from demonstration is the metrics of imitation as stated in (Schaal et al., 2003), it can be a the joint trajectory (González-Fierro et al., 2012), it can be the cartesian trajectory or it can be an effort function (Anderson and Pandy, 2001; Khatib et al., 2004b; Crowninshield and Brand, 1981). This metric can be a reward that measures the goal performance, as we proposed (González-Fierro et al., 2013a, 2014a) or it can be any other metric.

6.4 Postural Control

Postural control can be defined as controlling the body's position in space for the purpose of stability and orientation for the robot to move from one static posture to another (Shumway, 2000). Postural control is an integral component of redundant manipulators and multi-legged robots where the goal is to enhance the execution of manipulation and locomotion behaviors (Sentis, 2007). It plays an important role in the optimization of these behavior performance.

This section presents some postural control methods that were developed during this thesis. Some of them are original and some of them are taken from the state of the art and implemented in the robot. Furthermore, the addition of a robust control method is discussed to improve the overall system performance and help to correct

disturbances.

6.4.1 Whole Body Postural Control

Whole body postural control has to do with all control techniques the robot precise to follow the desired postural trajectory generated in the postural planning module.

Until now, all developments were performed off-line and the output was a task space trajectory that is transformed in a joint trajectory using the robot's Jacobian. There is a control part that has not been addressed yet but it is inherent to all robot movements.

In chapter 3, where a method for imitating and innovating human behaviors was developed, the robot follows a postural trajectory which is different in the case of the imitation behavior than in the case of the innovation behavior. The objective in the imitation part is to perform a movement that imitates the human strategy when standing up, and at the same time fits the humanoid constraints. On the other hand, in the innovation part, the objective is to perturb the previous initial solution to find a new movement which improves the imitation behavior from the humanoid's perspective.

In Figure 6.4(a) it is shown the desired joint trajectory of the actuated 3-link kinematic chain and the output of the PD controller in the imitation process. Figure 6.4(b) shows the innovation one. It is also represented the initial, middle and final points for the piecewise polynomial. The middle point is the result of the minimization of the fitness function (3.14).

Another example of whole body postural control is the walking pattern routine computed in (González-Fierro et al., 2013b). We selected the initial and final position of the walking trajectory, which in addition to the 3D model of the environment, was the input to the Bi-RRT algorithm. The result of this path planning was a preliminary cartesian COM trajectory. The output of the Bi-RRT algorithm can be seen in the following Figure 6.5.

The initial point is located near the bottom of the scenario and the final point is near the front door. To configure the free space, the Bi-RRT algorithm does not only take into account the data of the xml file, but also the dimensions of the robot plus a security quantity. Finally, from the COM trajectory, the joint trajectory is generated using a kinematic inversion.

6.4.2 Correction of Postural Disturbances

There are many works related to the posture correction of humanoid robots (Stephens, 2007; Kaynov et al., 2009a; Kajita et al., 2010; Stephens and Atkeson, 2010; Pierro, 2012; Martínez de la Casa et al., 2013), others related to the computation of postures as a reaction to some input (Bueno et al., 2012, 2013b) or works that maintain a

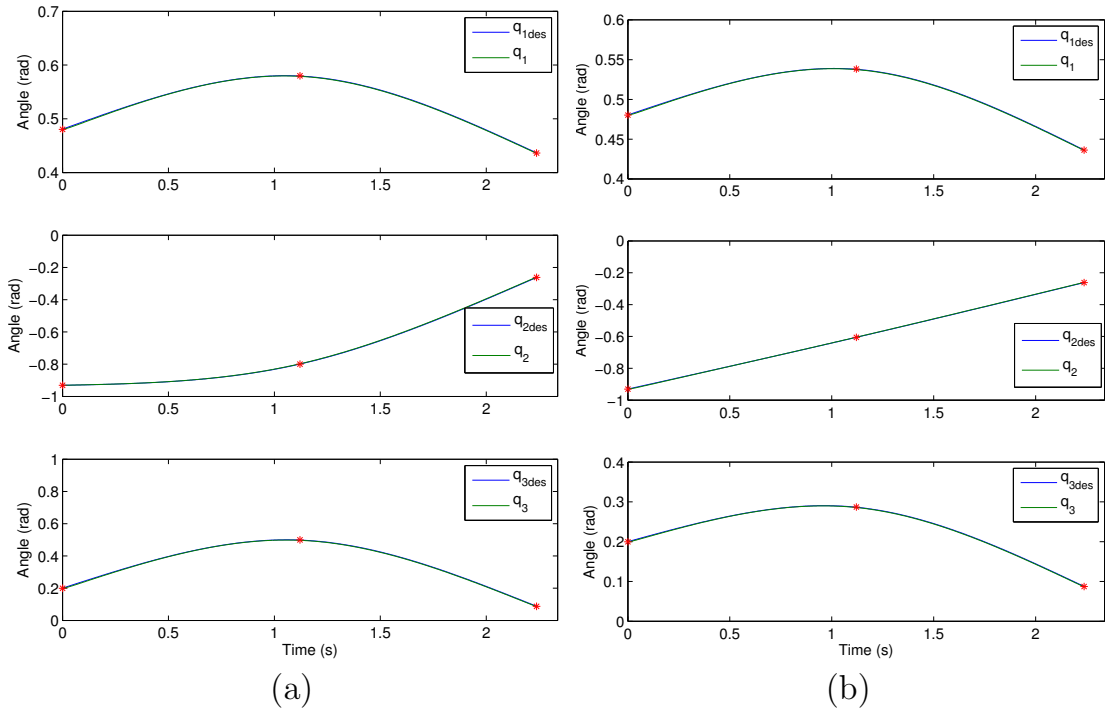


Figure 6.4: *Desired joint trajectory of the three actuated joints and output of the PD controller in the imitation process (a) and in the innovation process (b) of a humanoid robot standing up from a chair. The red dot represents the initial, middle and final points of the piecewise polynomial of* (4.24).

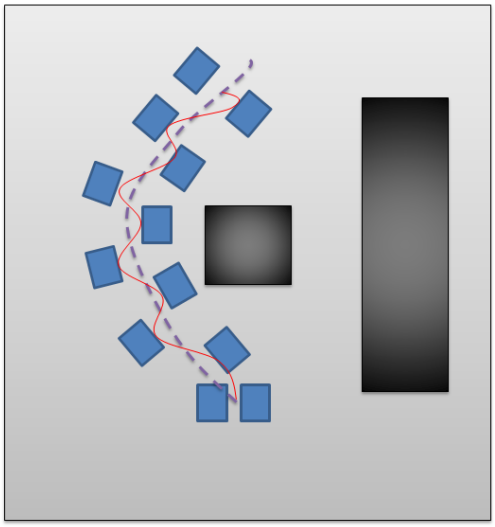


Figure 6.5: *Top-view schema of the complete scenario with obstacles in dark gray, global path planning represented by the dotted line, steps plotted as blue boxes and finally COM path shown with a red thin line.*

determined posture while walking (Choi et al., 2007) or performing a task (Khatib et al., 2004a, 2008; Sentis and Khatib, 2004, 2005; Sentis et al., 2010).

Our proposal in this matter is using a robust control technique like fractional control (González-Fierro et al., 2013e). Fractional controllers are able to reject disturbances and absorb mismatches in the robot model even with a clear difference between the real robot and the robot model. Therefore, the use of a robust controller like a fractional $PI^{\lambda}D^{\mu}$ minimizes the necessity of computing a complete and accurate multibody robot model, reducing the mathematical complexity and computation time. An extension of this idea was developed in chapter 5.

6.5 Experimental Results

In this section the experimental results are presented and discussed. The experiment carried out is the high level order *"stand up from a chair, walk to the door and open it"* which is executed in a cluttered environment by the humanoid HOAP-3.

6.5.1 Real Environment Analysis

The first step is to extract the supporting plane equation from the point cloud. Afterwards, the objects are identified using Euclidean Clustering Algorithm, and the equation of the bounding box containing each object is obtained. The plane coefficients and the bounding box of every object and its position are written to a xml file, which defines the 3D model of the environment.

In Figure 6.6 a simulation of the environment is shown using OpenRAVE (Diankov, 2010). Objects identified by the Asus camera are represented as light grey boxes. The first one is ahead the robot, and the second one is to its right. The floor is represented in dark grey. The rest of the environment was manually introduced and is irrelevant to the experiment. The yellow dotted line represents the field of view of the RGB-D camera installed in the upper part of the scenario.

With the help of the camera, the system identifies and clusters the obstacles as it is showed in Figure 6.7. It also identifies the floor using the supporting plane estimation. When both obstacles and floor are identified and their position obtained, a file is generated. This files has the information that is included in the simulation environment OpenRAVE to perform the path planning.

6.5.2 Postural Planning in a Humanoid

The result of the global path planning algorithm, integrating the data obtained with the perception sensor, is represented in Figure 6.8. The HOAP's head represents the COM of the complete robot and the plotted trajectory the final COM path estimation. The algorithm is able to obtain a safe trajectory for the robot. Even

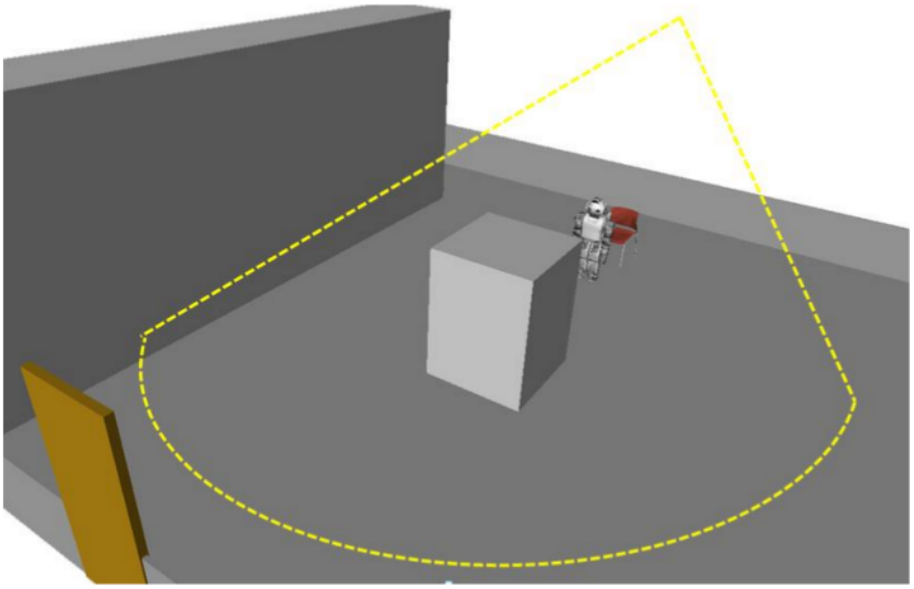


Figure 6.6: *Simulation of the environment with bounding boxes representing obstacles detected by the RGB-D sensor. Yellow dotted line represents the field of view of the RGB-D sensor.*

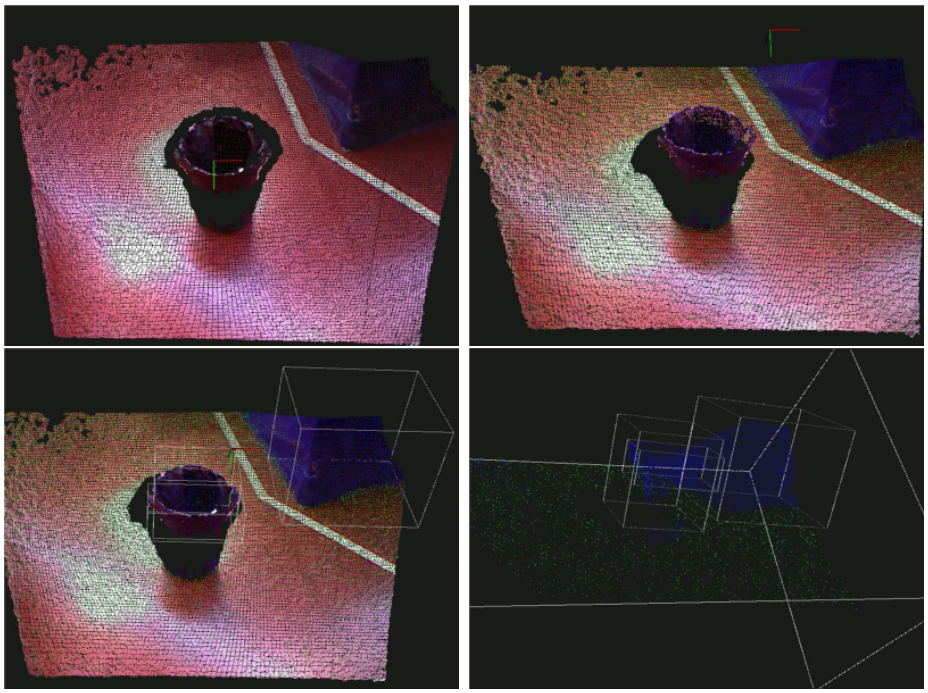


Figure 6.7: *In the snapshot the environment recognition is performed. The green dots represents the floor and the blue dots represent the obstacles. Finally a bounding box of the obstacles is extracted together with their dimensions and their positions.*

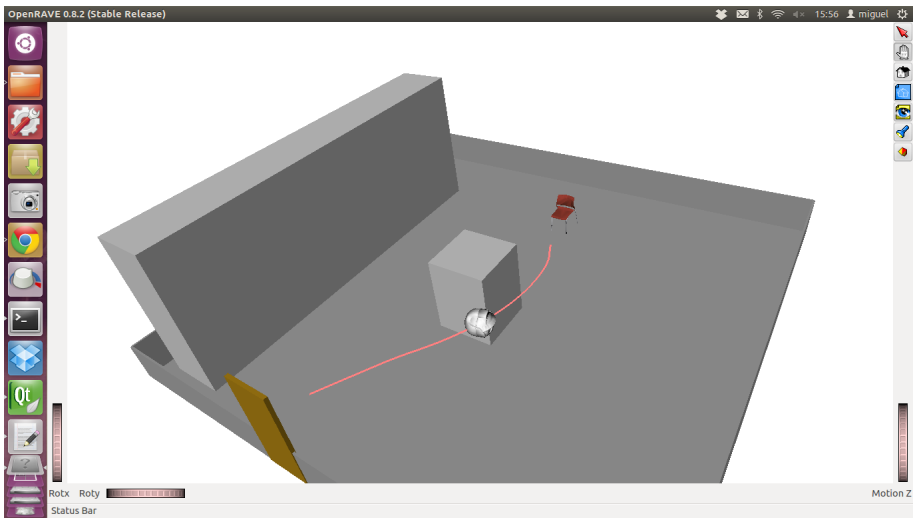


Figure 6.8: *Resulting path planning of the COM plotted in OpenRave. HOAP head represents the COM of the complete robot. It is also represented the obstacles as grey boxes.*

though there are other methods more precise like (Garrido et al., 2009), the selected method has a good trade between precision and computational time.

Once the global path planning is computed, the postural skill generator plans which are the postural primitives and skill sequence the robot has to follow to complete the task. In Figure 6.9 the sequence of generating a complete task of standing up and leave the room is showed. The postures defined are four, the robot is seated, then standing near the chair, then standing near the door and finally standing with the door opened. The transition among this postures are the skills the robot needs to generate to complete the task. The skills are standing up, walking and opening a door.

In this thesis is not included an automatic generator of postural skills. Right now the postures and skills are selected by the programmer, not by the robot. It is proposed as a future work a decision process where the robot is able to understand what is the objective of the order, what is its current postural state and what are the sequence of skills that the robot needs to follow to complete the order.

The final step is the computation of the postural motion planning. For the first skill, standing up from a chair, we used the method proposed by González-Fierro et al. (2013a, 2014a) and detailed in chapter 3. For the second skill, walking, we used the cart-table model proposed by Kajita et al. (2003a). Finally, for the last skill, opening a door, we used the method detailed in chapter 4.

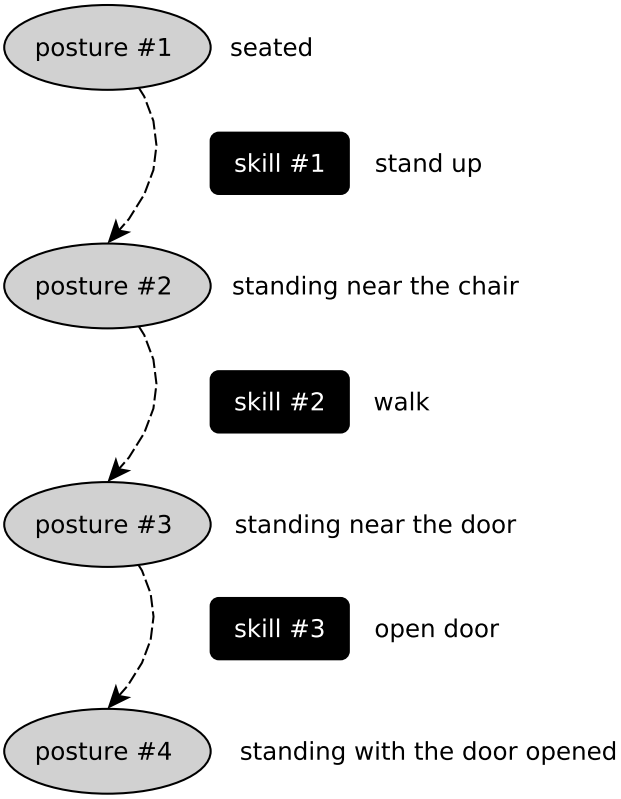


Figure 6.9: *Postural skill generation process for fulfilling the order "stand up from a chair, walk to the door and open it" in a humanoid robot. It consists on 4 postures, which are the initial and final states of the movement, and three skills, which corresponds to the dynamical transition between postural primitives.*

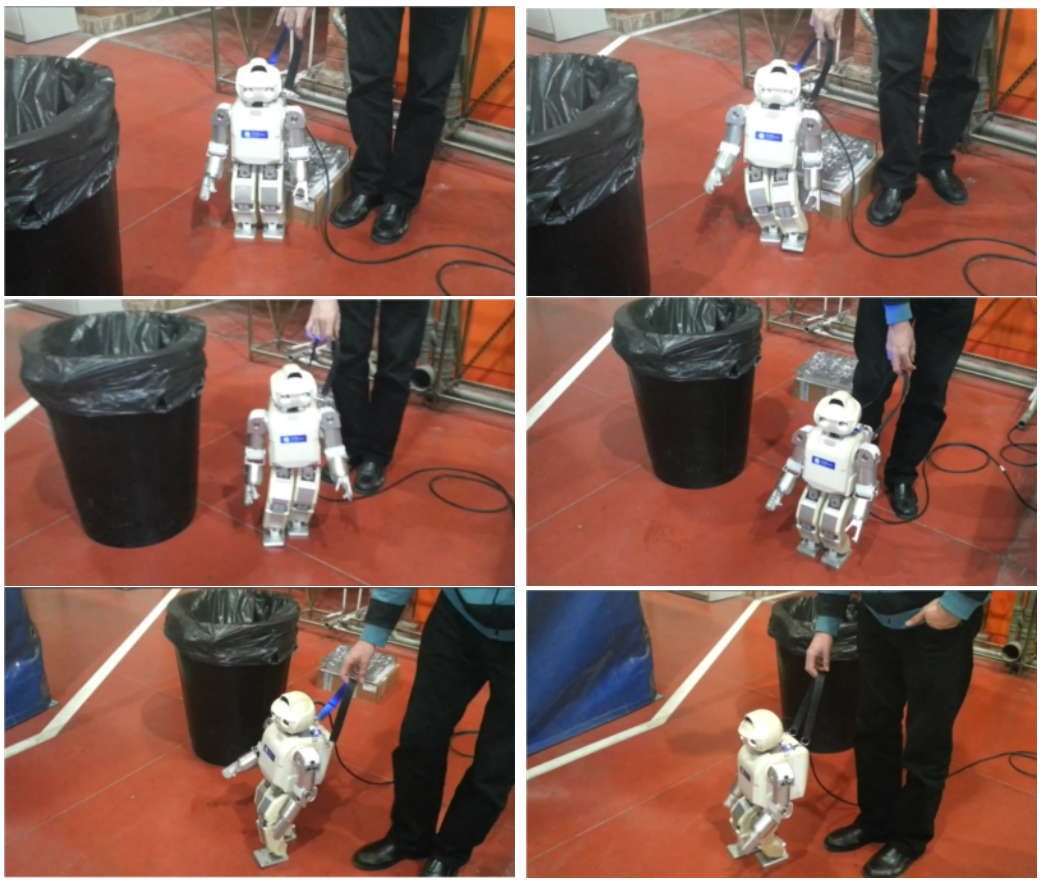


Figure 6.10: *Several frames representing the real performance of the robot path planning in the laboratory. The black bucket is one of the obstacles avoided.*

6.5.3 Postural Control in a Humanoid

The last step towards the execution of the complete task is the postural control of the robot, starting with the module whole body postural control. It includes the control strategy in the standing up, in the walking pattern and in the opening door behavior.

In Figure 6.10 a group of snapshots of the real humanoid performing the navigation is shown. The robot performs a safe locomotion trajectory avoiding the obstacles in the environment.

The last module, the correction of postural disturbances using fractional controllers (González-Fierro et al., 2013e), was not implemented in the real robot, just in simulation. The implementation of this method in the real robot is proposed as future works.

6.6 Discussion and Conclusions

This chapter introduced the architecture that supports this thesis. It is based on four steps, first the high level order given by the human, second an analysis of the environment to distinguish between the free space and the obstacles. The next step is the postural planning. It involves all postural movements that needs to be computed to allow the robot to plan the successful sequence of skills. They are the reference that has to be followed with the help of the postural control, which produces the robot movement in real time.

It has to be highlighted that the complete architecture is centered in the robot's movement from a postural perspective. It is not our interest the skills executed by the robot but what is the postural sequence the robot follows and what is the performance of this sequence. Is it a performance which imitates the human performance or is it even better for the robot body? They are questions that we addressed during this chapter and throughout this thesis.

Chapter 7

Conclusions and Future Works

This thesis has attempted to contribute in some areas related to posture behavior of humanoid robots. From the beginning, this work has had a differential feature if it is compared with a usual Ph.D. thesis. It has an initial framework which is the center of all developed studies. The framework is the high level order "stand up from a chair, walk to the door and open it" that the robot needs to acomplish. To execute that order, the humanoid robot needs to perform a set of movements, it needs to avoid a series of obstacles and it needs to adopt a determined posture or set of postures. All needed procedures to successfully achieve this order are gathered in this thesis. The amount of contributions that this thesis has generated are related to learning from demonstration, reinforcement learning, non-linear control and motion planning. In this chapter the conclusions of this thesis are summarized and a discussion of the positive as well as the negative aspects of the developed work is presented. There are also proposed some future lines of development that can be used to improve the current work or to serve as a start point for new developments.

7.1 Conclusions

The problem of postural behavior in humanoids has not being solved so far and it is being addressed by the scientific community with great interest. The advancement in control and learning methods, together with the attention that robotic science is recently drawing in areas not related directly to science, makes this matter of a significant importance.

If future robots are going to share our living space, they will need to be able to move safely, understand our orders, collaborate with us and have a high level of autonomy. Every chapter of this thesis attempts to give a step forward in the development of a humanoid robot able to really be our partner. Next, the summary and discussion of every chapter is presented.

In **Chapter 2** an introduction to the basic mathematical tools used in this thesis is presented. The equation of motion of different humanoid models is developed, the equation for the triple inverted pendulum takes a superior relevance since it is the most used model in the thesis. Then, the most used method for dynamic balance is explained, the ZMP, which is in deep relation with the locomotion methods presented afterwards. A simulated and a real walking pattern generation is presented. Finally, a simple method for whole body humanoid movement generation is studied. The method allows the robot to mimic a human movement and adapt it to its size. This method is the first step towards a most advanced postural behavior.

Chapter 3 introduces the concept of the reward as the basis of comparison between the postural behavior of a human and a robot. The reward is defined as a metrics of the action goal, which is evaluated in terms of stability and effort and is shared between any actor that is imitating an action. It also encodes the behavior in a single magnitude, therefore a robot can use it to imitate the behavior of a human but it can even improve it, generating a new behavior which is more suitable to the robot's body configuration and constraints.

A generalization of the previous idea is presented in **Chapter 4**. It takes into account not only one action but a set of different actions that are sequentially executed by the robot. The behavior transference between the human and the robot is performed through a context-based reward profile, which is sequentially selected based on the current behavior goal.

Chapter 5 presents a robust control method for controlling humanoid reduced models based on fractional order controllers. The presented method addresses the question of what kind of humanoid robot model should be used, whether a complete mass distributed model or a simple mass concentrated model. The solution presented is a combination of fractional order controllers and reduced robot models, which cancels the need of having a distributed model by absorbing disturbances in the environment and mismatches in the model.

Finally, **Chapter 6** presents the complete architecture of the thesis, discussing concepts like postural planning and postural control. It also proposes a closed

solution to the high level order *"stand up from a chair, walk to the door and open it"* which applies methods from the state of the art and some others developed in the thesis.

7.2 Key Contributions

The main contributions presented in this thesis are summarized as follows:

1. Regarding Postural Motion Planning and Control:

- We proposed to use a multimodal reward profile as a measurement of the good or bad performance of the action goal. The original idea comes from the work of Alissandrakis et al. (2002, 2007, 2009) where the behavior is accomplished at three levels. In (González-Fierro et al., 2013a, 2014a) we extended the idea of *trajectory level* where the behavior is defined as a set of subgoals that are sequentially reached. We used a temporal reward trajectory to represent the behavior.

- We proposed a new behavior representation using a Markov Transition Matrix named Reward Transition Probability Matrix (RTPM), that summarizes the state transition probabilities in the reward space. It represents the motion strategy in terms of the task goal, which we assumed is common for a human and a humanoid. This generic method can be extended to other movements like sitting down, crouching or grasping an object subject to a set of robot constraints.

- In Chapter 6 an architecture for the planning and control of complex postural tasks is proposed. It combines offline planning and online control, always from the point of view of the postural behavior of a humanoid robot.

2. Regarding Learning from Demonstration:

- We presented a new skill transfer method of stand up behaviors from human demonstrators to humanoid robots, that involves comparing temporal transition in a common multi-objective reward landscape. The main advantage is that we could accommodate the behavior even if the human and the robot have a mismatch in their kinematic structures, weights, and heights (González-Fierro et al., 2013a).

- In a first approach (González-Fierro et al., 2013a), we proposed to use the average multimodal reward profile as the basis of comparison between the human and the robot. Since the reward defines the behavior performance, we assumed that we can compare the human and robot performance by optimizing a policy that minimizes the error between the human's mean reward profile and the robot's mean reward profile.

- In a second proposal (González-Fierro et al., 2014a), we achieved imitation learning by finding a policy that minimizes the error between the predicted robot reward profile, if it behaves like a human which is computed using the RTPM, and the actual reward profile. The consequence is a trajectory that fits stability, torque and joint limit constraints while producing a movement that imitates the human behavior.

- The idea of imitation learning through a reward profile is extended to a set of sequential skills, computing a complex task. The work is presented in Chapter 4.

3. Regarding Skill Innovation:

- We refined the robot behavior by maximizing the positive difference between a new generated reward and the imitation reward, producing the innovation of new postural motions that are translated in a more suitable behavior of the humanoid robot (González-Fierro et al., 2014a).

- In Chapter 4 the same idea of skill innovation is applied to a set of sequential skills. The method is able to generate, not only a better performance for the robot in a single skill, but a better performance in the complete motion, which includes all sequential skills together.

4. Regarding Modelling and Control:

- In (González-Fierro et al., 2013e) we proposed a method for modelling and control of humanoid robot based on fractional order controllers and mass concentrated models. Our method proposes an alternative to mass concentrated models like (Kajita et al., 2001, 2003a; Komura et al., 2005; Kaynov et al., 2009b), which have an easy mathematical formulation, are fast in terms of computational cost but have the disadvantage that they have mismatches in the model, which can produce undesired behaviors. It also proposes an alternative to mass distributed models like (Kajita et al., 2003b; Khatib et al., 2008; Arbulú et al., 2010), that have a complex formulation, are slow but use a precise humanoid model. A combination of a robust control method plus a mass concentrated model has the advantage of having a simple mathematical formulation, of being fast in terms of computational cost and also of absorbing the mismatches in the model due to the fractional order controller.

7.3 Future Works

There is a long path before creating a truly human-like skilled humanoid robot. Here, I would like to encourage researchers to extend and enhance the work presented in this thesis.

Robotics is expected to grow into a wide range of robots with different embodiments. Therefore, the correspondence problem is likely to become one of the bottlenecks in the robotics development. It will be necessary to develop new approaches that explore intention understanding and goal emulation techniques.

There is a broad spectrum of research in the way the reward profile is selected. We do not really know what is the process that appears in the brain that optimizes so gracefully and smoothly the human motion. It will require an effort in neuroscience studying the brain and how infants learn. It will be necessary as well to explore the hidden objectives that the user is not aware of. A promising technique that is currently overpassing all machine learning records is deep learning (Hinton et al., 2006; Lee et al., 2009; Ngiam et al., 2011; Le, 2013). It has the advantage that the method performance improves as it grows the number of learning examples.

Another open path is the transference of behaviors from robot to robot or from human to human. Using some of the methods proposed in this thesis, and taking ideas from previous approaches like (Nanayakkara et al., 2007), it may be possible to transfer new skills or improve the previously learned by using a teacher-learner approach, where the teacher is the skillful individual or robot that is imitated by a group of amateur individuals or robots. It can even lead to the rising of newer and better skills than the ones taught, producing a new teacher that becomes the new reference for the group.

It is also proposed as a future work the implementation of the fractional controller in the real humanoid. Furthermore, it would be interested to compare the performance of a combination of a reduced model and a fractional controller with the combination of a complete model and a fractional controller. Even though a mass distributed model is much more precise than a concentrated model, there are always mismatches between the real world and the model, which can be set aside with the integration of a fractional controller. It will be interesting to perform a benchmark of both methods.

Finally, it will be usefull to incorporate a decision module where the robot is able to autonomously understand a complex order, know what is its current postural configuration and generate the sequence of skills needed to complete the order.

7.4 List of Publications

Here are listed the publications produced during the Ph.D. development.

Journal Papers

1. (González-Fierro et al., 2014a) **González-Fierro, M.**, Balaguer, C., Swann, N., and Nanayakkara, T. (2014). Full-Body Postural Control of a Humanoid

Robot with Both Imitation Learning and Skill Innovation. *International Journal of Humanoid Robotics*, 11(2):1450012.

2. (Monje et al., 2013) Monje, C.A., Pierro, P., Ramos, T., **González-Fierro, M.**, and Balaguer, C. (2013). Modelling and Simulation of the Humanoid Robot HOAP-3 in the OpenHRP3 Platform *Cybernetics and Systems*, 44(8):663–680.

3. (Bueno et al., 2013b) Bueno, J. G., **González-Fierro, M.**, Moreno, L., and Balaguer, C. (2013). Facial Emotion Recognition and Adaptative Postural Reaction by a Humanoid based on Neural Evolution *International Journal of Advanced Computer Science*, 3(10):481–493.

4. (González-Fierro et al., 2014b) **González-Fierro, M.**, Hernández, D., Nanayakkara, T., and Balaguer, C. (2014). Behavior Sequencing Based on Demonstrations - a Case of a Humanoid Opening a Door while Walking. *submitted to Advanced Robotics*.

5. (González-Fierro et al., 2014c) **González-Fierro, M.**, Monje, C. A., and Balaguer, C. (2014). Fractional Control of a Humanoid Robot Reduced Model with Model Disturbances. *submitted to Cybernetics and Systems.*

Conference/Symposiums Papers

1. (González-Fierro et al., 2013a) **González-Fierro, M.**, Balaguer, C., Swann, N., and Nanayakkara, T. (2013). A Humanoid Robot Standing Up Through Learning from Demonstration Using a Multimodal Reward Function. In *IEEE-RAS International Conference on Humanoid Robots, 2013. Humanoids 2013.* IEEE.

2. (González-Fierro et al., 2013b) **González-Fierro, M.**, Bueno, J., Balaguer, C., and Moreno, L. (2013). A Complete 3D Perception and Path Planning Architecture for a Humanoid. In *Proceedings of Robocity2030 11th Workshop: Robots Sociales*, pages 167–184.

3. (González-Fierro et al., 2013c) **González-Fierro, M.**, Maldonado, M. A., Víctores, J. G., Morante, S., and Balaguer, C. (2013). Object Tagging for Human-Robot Interaction by Recolorization using Gaussian Mixture Models. In *Proceedings of Robocity2030 12th Workshop: Robótica Cognitiva*, pages 67–76.

4. (González-Fierro et al., 2013d) **González-Fierro, M.**, Monje, C., González, V., and Balaguer, C. (2013). Evolutionary Fractional Order Control of a Humanoid Robot Modeled as a Triple Inverted Pendulum. In *Proceedings of Robocity2030 11th Workshop: Robots Sociales*, pages 245–263.

5. (González-Fierro et al., 2013e) **González-Fierro, M.**, Monje, C. A., and Balaguer, C. (2013). Robust Control of a Reduced Humanoid Robot Model using Genetic Algorithms and Fractional Calculus. In *Mathematical Methods in Engineering International Conference MME2013*, pages 183–194.

6. (Bueno et al., 2013a) Bueno, J., Martín, A., **González-Fierro, M.**, Moreno, L., and Balaguer, C. (2013). Distinguishing between Similar Objects based on Geometrical Features in 3D Perception. In *Proceedings of Robocity2030 12th Workshop: Robótica Cognitiva*, pages 77–92.

7. (Víctores et al., 2013) Víctores, J. G., Morante, S., **González-Fierro, M.**, and Balaguer, C. (2013). Augmented Reality and Social Interaction platform through Multirobot Design. In *Proceedings of Robocity2030 11th Workshop: Robots Sociales*, pages 131–143.

8. (González-Fierro et al., 2012) **González-Fierro, M.**, Hernández, D., Pierro, P., and Balaguer, C. (2012). Dynamic Modelling of Humanoid Robots Using Spatial Algebra. In *XXXIII Jornadas de Automática*. CEA.

9. (Pierro et al., 2012) Pierro, P., Hernández, D., Herrero, D., **González-Fierro, M.**, and Balaguer, C. (2012). Perception System for Working with Humanoid Robots in Unstructured Collaborative Scenarios. In *Proceedings of the 2012 International IEEE Intelligent Vehicles Symposium. Workshops V Perception in Robotics*. IEEE.

10. (Bueno et al., 2012) Bueno, J. G., **González-Fierro, M.**, L.Moreno, and C.Balaguer (2012). Facial Gesture Recognition using Active Appearance Models based on Neural Evolution. In *2012 International Conference on Human-Robot Interaction (HRI 2012)*, pages 133–134. IEEE.

11. (Monje et al., 2011b) Monje, C. A., Pierro, P., Ramos, T., **González-Fierro, M.**, and Balaguer, C. (2011). Modelling and Simulation of the Humanoid Robot HOAP-3 in the OpenHRP3 Platform. In *Proceedings of Robot 2011. Workshop Robots Humanoides*.

12. (Bueno et al., 2011) Bueno, J. G., **González-Fierro, M.**, Moreno, L., and Balaguer, C. (2011). Facial Gesture Recognition and Postural Interaction using Neural Evolution Algorithm and Active Appearance Models. In *Proceedings of Robocity2030 9th Workshop: Robots colaborativos e interacción humano-robot*, pages 145–159.

13. (González-Fierro et al., 2010) **González-Fierro, M.**, Jardón, A., Martínez de la Casa, S., Stoelen, M. F., Víctores, J. G., and Balaguer, C. (2010). Educational Initiatives Related with the CEABOT Contest. *Proceedings of SIMPAR*, pages 649–658.

14. (Mateo et al., 2010) Mateo, A. P., **González-Fierro, M.**, Hernández, D., Pierro, P., and Balaguer, C. (2010). Robust Real Time Stabilization: Estabilización de la Imagen con Aplicación en el Robot Humanoide HOAP-3. In *Proceedings of Robocity2030 7th Workshop: Visión en Robótica*.

15. (Peña et al., 2010) Peña, A., Hernández, D., **González-Fierro, M.**, Pierro, P., and Balaguer, C. (2010). Sistema de Visión del Humanoide HOAP-3 para la Detección e Identificación de Objetos Mediante Librerías OpenCV. In *Proceedings of Robocity2030 7th Workshop: Visión en Robótica*.

16. (Pierro et al., 2009a) Pierro, P., **González-Fierro, M.**, and Balaguer, C. (2009). El Proyecto Europeo ROBOT@CWE: Advanced Robotic Systems in Future Collaborative Working Environments. In *Proceedings of Robot 2009. II Workshop de Robotica (ROBOT 2009)*.

17. (Pierro et al., 2009b) Pierro, P., Hernández, D., **González-Fierro, M.**, Blasi, L., Milani, A., and Balaguer, C. (2009). A Human-Humanoid Interface for Collaborative Tasks. In *Proceedings on the Second workshop for young researchers on Human-friendly robotics*, Sestri Levante, Italy.

18. (Pierro et al., 2009c) Pierro, P., Hernández, D., **González-Fierro, M.**, Blasi, L., Milani, A., and Balaguer, C. (2009). Humanoid Teleoperation System for Space Environments. In *Advanced Robotics, 2009. ICAR 2009. International Conference on*, pages 1–6. IEEE.

19. (González-Fierro et al., 2009) **González-Fierro, M.**, Pierro, P., Jardón, A., Herrero, D., and Balaguer, C. (2009). Realización de Tareas Colaborativas entre Robots Humanoides. Experimentación con dos Robots Robonova. In *Proceedings of Robocity2030 5th Workshop: Cooperación en Robótica*.

Appendix A

HOAP-3 Robot

In this thesis we used the humanoid HOAP-3 as the experimental platform (Figure A.1). It is a robot of small size and height, 60 cm and 9 Kg approximately, designed and manufactured by the Japanese company Fujitsu. HOAP stands for "Humanoid for Open Architecture Platform" and this model is the evolution of the previous versions, HOAP and HOAP-2. In 2001, Fujitsu released its first commercial humanoid robot named HOAP. The HOAP–2 was released in 2003, followed by the HOAP–3 in 2005. Sadly, the company closed the funding for robotics development so in the short term there will not be a new version of the HOAP series.

HOAP-3 robot has 28 degrees of freedom (DoF) distributed as it is shown in Figure A.2. It possesses 6 DoF in each leg, 6 in each arm, 3 in the head and 1 in the hip. Even though the humanoid has 28 DoF, it has only 23 servomotors. The first 21 control legs, arms and hip as it is showed in Figure A.3. These motors incorporate relative enconders and can be controlled in position and velocity.

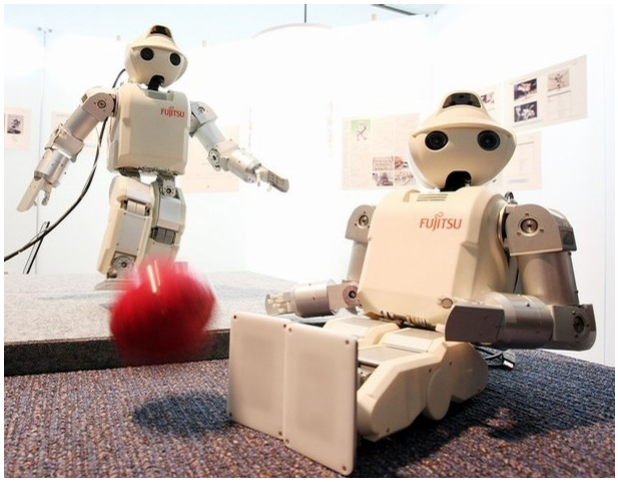


Figure A.1: *Humanoid robot HOAP-3.*

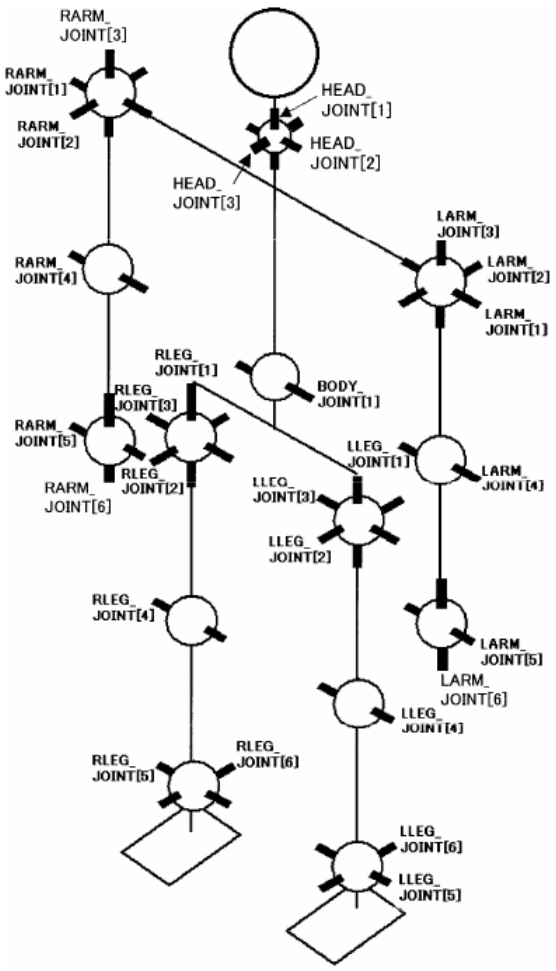


Figure A.2: *Degrees of freedom of HOAP-3 robot.*

Motors number 22 and 23 does not have encoders. Motor 22 controls the 3 DoF of the head (pitch, yaw and roll) and motor 23 controls the hand rotation and the griper.

The robot incorporates an embedded PC-104 inside the back pack (Figure A.4). It is a Pentium of 1.1 GHz with 512 Mb of RAM and a Compact Flash memory of 1 Gb. It includes a WIFI connection IEEE802.11g and 4 USB ports. The processing capacity of the robot is low. Some algorithms that require a high computational capacity, like computer vision algorithms, have to be computed outside the humanoid CPU. Those related to low level control are usually computed inside the robot.

The operating system is a Real Time Linux, based on Fedora Core 1 with the kernel 2.4. In order to control the robot there are two connection methods, the first is wireless through telnet, the second one is a direct connection by means of a USB cable that connects the external PC directly with the motors and drivers of the

Joint name	Flexibility	Device ID
HEAD_JOINT[1]	Head torsion	22
HEAD_JOINT[2]	Head pitch	22
HEAD_JOINT[3]	Head roll	22
BODY_JOINT[1]	Waist pitch	21
RLEG_JOINT[1]	Right hip joint torsion	1
RLEG_JOINT[2]	Right hip joint roll	2
RLEG_JOINT[3]	Right hip joint pitch	3
RLEG_JOINT[4]	Right knee	4
RLEG_JOINT[5]	Right Ankle pitch	5
RLEG_JOINT[6]	Right Ankle roll	6
RARM_JOINT[1]	Right shoulder Pitch	7
RARM_JOINT[2]	Right shoulder roll	8
RARM_JOINT[3]	Right shoulder torsion	9
RARM_JOINT[4]	Right elbow	10
RARM_JOINT[5]	Right fingers open/close	23
RARM_JOINT[6]	Right hand torsion	23
LLEG_JOINT[1]	Left hip joint torsion	11
LLEG_JOINT[2]	Left hip joint roll	12
LLEG_JOINT[3]	Left hip joint pitch	13
LLEG_JOINT[4]	Left knee	14
LLEG_JOINT[5]	Left Ankle pitch	15
LLEG_JOINT[6]	Left Ankle roll	16
LARM_JOINT[1]	Left shoulder Pitch	17
LARM_JOINT[2]	Left shoulder roll	18
LARM_JOINT[3]	Left shoulder torsion	19
LARM_JOINT[4]	Left elbow	20
LARM_JOINT[5]	Left fingers open/close	23
LARM_JOINT[6]	Left hand torsion	23

Figure A.3: *HOAP joints and the associated motor.*

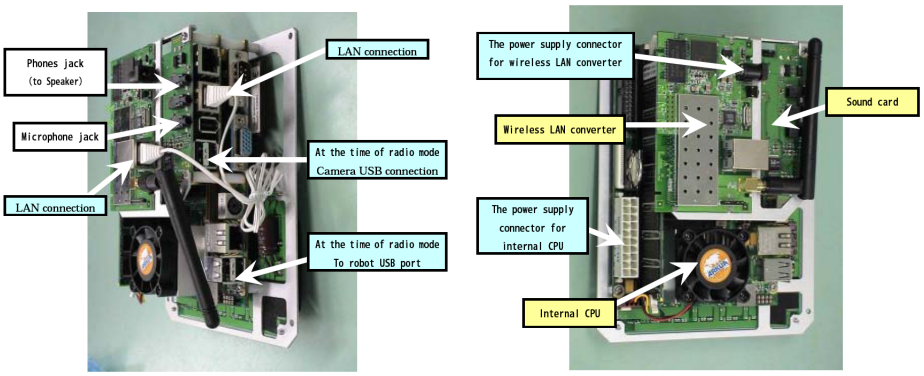


Figure A.4: *Embeded PC of the humanoid HOAP-3.*

robot. Besides, the robot can be powered directly by cable or through a battery of 24V. The robot currently uses YARP (**?**) as a communication framework.

To complete the humanoid functionality a set of sensors were added. It has two stereo cameras for stereoscopy vision, a microphone, a speaker, distance infrared sensors, FSR force sensors, gyroscopes and accelerometers in the 3 axis.

We can not doubt that the HOAP-3 is a very complete and versatile platform, which includes all necessary features to develop any type of robot based research.

Appendix B

Mass Distribution of an Average Human

In Figure B.1 and Table B.1 the distribution of an average human body is shown along with the corresponding percentage for each link.NASA (1995)

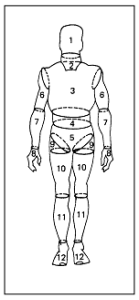


Figure B.1: *Body segments of a United State male crew member.*

Table B.1: *Mass distribution of the body segments of a United State male crew member and distribution of the actuated 3-link kinematic chain's mass for the human.*

Segments (see Figure B.1)	Mass (g)	Percentage (%)
Head (1)	4550	5.78
Neck (2)	1270	1.61
Thorax (3)	31760	40.32
Abdomen (4)	2960	3.76
Pelvis (5)	15150	19.24
Upper arm (6)	2500	3.17
Forearm (7)	1720	2.18
Hand (8)	610	0.77
Hip flap (9)	4380	5.56
Thigh minus flap (10)	7920	10.06
Calf (11)	4760	6.04
Foot (12)	1180	1.50
Torso (3 + 4 + 5)	49870	63.32
Thigh (9 + 10)	12300	15.62
Forearm plus hand (7 + 8)	2320	2.95
Total weight	78760	100
Link 1 (m_1)		7.54
Link 2 (m_2)		34.86
Link 3 (m_3)		57.60

Bibliography

Abbeel, P. and Ng, A. (2004). Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM.

Akachi, K., Kaneko, K., Kanehira, N., Ota, S., Miyamori, G., Hirata, M., Kajita, S., and Kanehiro, F. (2005). Development of humanoid robot hrp-3p. In *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, pages 50–55. IEEE.

Akins, C. and Zentall, T. (1998). Imitation in Japanese quail: The role of reinforcement of demonstrator responding. *Psychonomic Bulletin & Review*, 5(4):694–697.

Aksoy, E., Abramov, A., Dörr, J., Ning, K., Dellen, B., and Wörgötter, F. (2011). Learning the semantics of object–action relations by observation. *The International Journal of Robotics Research*, 30(10):1229–1249.

Aleotti, J. and Caselli, S. (2012). Learning manipulation tasks from human demonstration and 3d shape segmentation. *Advanced Robotics*, 26(16):1863–1884.

Alissandrakis, A., Nehaniv, C., and Dautenhahn, K. (2002). Imitation with ALICE: Learning to imitate corresponding actions across dissimilar embodiments. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 32(4):482–496.

Alissandrakis, A., Nehaniv, C., and Dautenhahn, K. (2004). Towards robot cultures?: Learning to imitate in a robotic arm test-bed with dissimilarly embodied agents. *Interaction Studies*, 5(1):3–44.

Alissandrakis, A., Nehaniv, C., and Dautenhahn, K. (2007). Correspondence mapping induced state and action metrics for robotic imitation. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 37(2):299–307.

Alissandrakis, A., Otero, N., and Saunders, J. (2009). Helping robots imitate: Metrics and technological solutions inspired by human behaviour. In *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, pages 135–140. IEEE.

Alissandrakis, A., Syrdal, D., and Miyake, Y. (2011). Helping robots imitate: Acknowledgement of, and adaptation to, the robot's feedback to a human task demonstration. In Dautenhahn, K. and Saunders, J., editors, *New Frontiers in Human-Robot Interaction*, pages 9–33. John Benjamins Publishing Company.

Amabile, T. (1996). Creativity and innovation in organizations. *Harvard Business School*, 5(9):396–239.

Anderson, F. C. and Pandy, M. G. (2001). Static and dynamic optimization solutions for gait are practically equivalent. *Journal of biomechanics*, 34(2):153–161.

Arbulu, M. (2009). *Stable locomotion of humanoid robots based on mass concentrated model*. PhD thesis, Universidad Carlos III de Madrid.

Arbulú, M. and Balaguer, C. (2009). Real-Time Gait Planning for Rh-1 Humanoid Robot Using Local Axis Gait Algorithm. *International Journal of Humanoid Robotics*, 6(1):71–91.

Arbulú, M., Balaguer, C., Monge, C., Martínez, S., and Jardon, A. (2010). Aiming for multibody dynamics on stable humanoid motion with special euclideans groups. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 691–697. IEEE.

Arbulú, M., Kaynov, D., Cabas, L. M., and Balaguer, C. (2009). The Rh-1 full-size humanoid robot: design, walking pattern generation and control. *Journal of Applied Bionics and Biomechanics.*, 6(3):301–344.

Arbulú, M., Pardos, J. M., Cabas, L., Staroverov, P., Kaynov, D., Pérez, C., Rodríguez, M., and Balaguer, C. (2005). Rh-0 humanoid full size robot's control strategy based on the lie logic technique. In *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, pages 271–276. IEEE.

Arbulu, M., Yokoi, K., Kheddar, A., and Balaguer, C. (2008). Dynamic acyclic motion from a planar contact-stance to another. In *Intelligent Robots and Systems, 2008.(IROS 2008). Proceedings. 2008 IEEE/RSJ International Conference on*, pages 3440–3445.

Argall, B., Browning, B., and Veloso, M. (2008). Learning robot motion control with demonstration and advice-operators. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 399–404. IEEE.

Argall, B., Chernova, S., Veloso, M., and Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483.

Ariki, Y., Hyon, S.-H., and Morimoto, J. (2013). Extraction of primitive representation from captured human movements and measured ground reaction force to generate physically consistent imitated behaviors. *Neural Networks*, 40:32–43.

Arisumi, H., Miossec, S., Chardonnet, J.-R., and Yokoi, K. (2008). Dynamic lifting by whole body motion of humanoid robots. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 668–675. IEEE.

Asher, U. M., Pai, D., and Cloutier, B. P. (1997). Forward dynamics, elimination methods, and formulation stiffness in robot simulation. *International Journal of Robotic Research*, 6(6):749–758.

Ayusawa, K., Venture, G., and Nakamura, Y. (2008). Identification of humanoid robots dynamics using floating-base motion dynamics. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2854–2859. IEEE.

Ball, T., Farr, J., and Hanson, R. (1989). *Political innovation and conceptual change*, volume 11. Cambridge Univ Pr.

Bandura, A. (1986). Social foundations of thought and action. *Englewood Cliffs, NJ*.

Barbosa, R., Tenreiro, J., and Ferreira, I. (2004a). PID Controller Tuning Using Fractional Calculus Concepts. *Fractional Calculus and Applied Analysis*, 7(2):119–134.

Barbosa, R., Tenreiro, J. A., and Ferreira, I. M. (2004b). Tuning of PID Controllers based on Bode's Ideal Transfer Function. *Nonlinear Dynamics*, 38:305–321.

Barin, K. (1989). Evaluation of a generalized model of human postural dynamics and control in the sagittal plane. *Biological Cybernetics*, 61(1):37–50.

Barrientos, A., Peñín, L. F., Balaguer, C., and Aracil, R. (2007). *Fundamentos de Robótica*. Mc Graw-Hill, 2ª edition.

Barrios-Aranibar, D., Gonçalves, L., and Alsina, P. (2008). Learning by Experience and by Imitation in Multi-Robot Systems. *International Journal of Advanced Robotic Systems.(Org.). Livro: Frontiers in Evolutionary Robotics. Viena: Aleksandar Lazinica*.

Bentivegna, D., Atkeson, C., and Cheng, G. (2004). Learning tasks from observation and practice. *Robotics and Autonomous Systems*, 47(2):163–169.

Billard, A., Calinon, S., Dillmann, R., and Schall, S. (2008). *Robot programming by demonstration*, volume 1, chapter 59. Eds. Cambridge. MIT Press.

Billard, A., Calinon, S., and Guenter, F. (2006). Discriminative and adaptive imitation in uni-manual and bi-manual tasks. *Robotics and Autonomous Systems*, 54(5):370–384.

Billard, A., Epars, Y., Calinon, S., Schaal, S., and Cheng, G. (2004). Discovering optimal imitation strategies. *Robotics and autonomous systems*, 47(2):69–77.

Bode, H. W. (1940). Relations between attenuation and phase in feedback amplifier design. *Bell System Technical Journal*, 19(3):421–454.

Bode, H. W. (1945). Network analysis and feedback amplifier design.

Bonnet, V., Fraisse, P., Ramdani, N., Lagarde, J., Ramdani, S., and Bardy, B. G. (2009). A robotic closed-loop scheme to model human postural coordination. In *Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems*, pages 2525–2530. IEEE Press.

Book, W. J. (1984). Recursive lagrangian dynamics of flexible manipulator arms. *International Journal of Robotics Research*, 3:87–101.

Bouyarmane, K. and Kheddar, A. (2012). Humanoid robot locomotion and manipulation step planning. *Advanced Robotics*, 26(10):1099–1126.

Brooks, R. A. (1990). Elephants don't play chess. *Robotics and autonomous systems*, 6(1):3–15.

Brooks, R. A. (1991). Intelligence without representation. *Artificial intelligence*, 47(1):139–159.

Bueno, J., Martín, A., González-Fierro, M., Moreno, L., and Balaguer, C. (2013a). Distinguishing between similar objects based on geometrical features in 3d perception. In *Proceedings of Robocity2030 12th Workshop: Robótica Cognitiva*, pages 77–92.

Bueno, J. G., González-Fierro, M., L.Moreno, and C.Balaguer (2012). Facial Gesture Recognition using Active Appearance Models based on Neural Evolution. In *2012 International Conference on Human-Robot Interaction (HRI 2012)*, pages 133–134. IEEE.

Bueno, J. G., González-Fierro, M., Moreno, L., and Balaguer, C. (2011). Facial gesture recognition and postural interaction using neural evolution algorithm and active appearance models. In *Proceedings of Robocity2030 9th Workshop: Robots colaborativos e interacción humano-robot*, pages 145–159.

Bueno, J. G., González-Fierro, M., Moreno, L., and Balaguer, C. (2013b). Facial emotion recognition and adaptative postural reaction by a humanoid based on neural evolution. *International Journal of Advanced Computer Science*, 3(10):481–493.

Burns, T. and Stalker, G. (2009). The management of innovation.

Byrne, R. (2003). Imitation as behaviour parsing. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 358(1431):529–536.

Cakmak, M. and Thomaz, A. L. (2012). Designing robot learners that ask good questions. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pages 17–24. ACM.

Calinon, S. (2009). *Robot programming by demonstration: a probabilistic approach*. EPFL Press.

Calinon, S. and Billard, A. (2007). What is the teacher's role in robot programming by demonstration? Toward benchmarks for improved learning. *Interaction Studies*, 8(3).

Calinon, S., Bruno, D., Malekzadeh, M. S., Nanayakkara, T., and Caldwell, D. G. (2014). Human–robot skills transfer interfaces for a flexible surgical robot. *Computer methods and programs in biomedicine*.

Calinon, S., D'halluin, F., Sauser, E., Caldwell, D., and Billard, A. (2010). Learning and reproduction of gestures by imitation. *Robotics & Automation Magazine, IEEE*, 17(2):44–54.

Calinon, S., Evrard, P., Gribovskaya, E., Billard, A., and Kheddar, A. (2009). Learning collaborative manipulation tasks by demonstration using a haptic interface. In *Advanced Robotics, 2009. ICAR 2009. International Conference on*, pages 1–6. IEEE.

Calinon, S., Guenter, F., and Billard, A. (2007). On learning, representing, and generalizing a task in a humanoid robot. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 37(2):286–298.

Caponetto, R., Fortuna, L., and Porto, D. (2004). A New Tuning Strategy for a Non Integer Order PID Controller. In *First IFAC Workshop on Fractional Differentiation and Its Application*, pages 168–173, ENSEIRB, Bordeaux, France.

Chalodhorn, R., MacDorman, K. F., and Asada, M. (2009). Humanoid robot motion recognition and reproduction. *Advanced Robotics*, 23(3):349–366.

Chen, Z., Yuan, X., Ji, B., Wang, P., and Tian, H. (2014). Design of a fractional order pid controller for hydraulic turbine regulating system using chaotic non-dominated sorting genetic algorithm ii. *Energy Conversion and Management*, 84:390–404.

Choi, Y., Kim, D., Oh, Y., and You, B.-J. (2007). Posture/walking control for humanoid robot based on kinematic resolution of com jacobian with embedded motion. *Robotics, IEEE Transactions on*, 23(6):1285–1293.

Chung, K. (1967). *Markov chains with stationary transition probabilities*, volume 104. Springer Berlin.

Coradeschi, S. and Saffiotti, A. (2006). Symbiotic robotic systems: Humans, robots, and smart environments. *Intelligent Systems, IEEE*, 21(3):82–84.

Craighero, L., Metta, G., Sandini, G., and Fadiga, L. (2007). The mirror-neurons system: data and models. *Progress in brain research*, 164:39–59.

Crowninshield, R. D. and Brand, R. A. (1981). A physiologically based criterion of muscle force prediction in locomotion. *Journal of biomechanics*, 14(11):793–801.

Daniel, C., Neumann, G., and Peters, J. (2012). Learning Concurrent Motor Skills in Versatile Solution Spaces. pages 3591–3597. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

Dasgupta, A. and Nakamura, Y. (1999). Making feasible walking motion of humanoids. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1044–1049.

Dasgupta, S. and Schulman, L. J. (2000). A Two-Round Variant of EM for Gaussian Mixtures. In *UAI '00: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 152–159, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Delavari, H., Lanusse, P., and Sabatier, J. (2013). Fractional order controller design for a flexible link manipulator robot. *Asian Journal of Control*, 15(3):783–795.

Dey, T. K., Li, G., and Sun, J. (2005). Normal estimation for point clouds: A comparison study for a voronoi based method. In *Point-Based Graphics, 2005. Eurographics/IEEE VGTC Symposium Proceedings*, pages 39–46. IEEE.

Diankov, R. (2010). *Automated Construction of Robotic Manipulation Programs*. PhD thesis, Carnegie Mellon University, Robotics Institute.

Diftler, M. A., Mehling, J., Abdallah, M. E., Radford, N. A., Bridgwater, L. B., Sanders, A. M., Askew, R. S., Linn, D. M., Yamokoski, J. D., Permenter, F., et al. (2011). Robonaut 2-the first humanoid robot in space. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2178–2183. IEEE.

Eng, J., Winter, D., MacKinnon, D., and Patla, A. (1992). Interaction of the reactive moments and centre of mass displacement for postural control during voluntary arm movements. *Neuroscience Research Communications*, 11(2):73–80.

Evrard, P., Gribovskaya, E., Calinon, S., Billard, A., and Kheddar, A. (2009). Teaching physical collaborative tasks: Object-lifting case study with a humanoid. In *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, pages 399–404. IEEE.

Featherstone, R. (1983). The Calculation of Robot Dynamics Using Articulated-Body inertias. *International J. of Robotics, 2(1)*, pages 13–29.

Featherstone, R. (1987). *Robot Dynamics Algorithms*. Kluwer Academic Publishers.

Featherstone, R. (1999a). A Divide-and-Conquer Articulated-Body Algorithm for Parallel O(log(n)) Calculation of Rigid-Body Dynamics. Part 1: Basic Algorithm. *I. J. Robotic Res.*, 18(9):867–875.

Featherstone, R. (1999b). A Divide-and-Conquer Articulated-Body Algorithm for Parallel O(log(n)) Calculation of Rigid-Body Dynamics. Part 2: Trees, Loops, and Accuracy. *I. J. Robotic Res.*, 18(9):876–892.

Featherstone, R. (2008). *Rigid Body Dynamics Algorithms*. Springer.

Garrido, S., Moreno, L., Abderrahim, M., and Blanco, D. (2009). Fm2: a real-time sensor-based feedback controller for mobile robots. *International Journal of Robotics & Automation*, 24(1):48.

Gates, B. (2007). A robot in every home. *Scientific American*, 296(1):58–65.

Gergely, G., Bekkering, H., Király, I., et al. (2002). Rational imitation in preverbal infants. *Nature*, 415(6873):755.

Goldberg, D. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-wesley.

González-Fierro, M., Balaguer, C., Swann, N., and Nanayakkara, T. (2013a). A Humanoid Robot Standing Up Through Learning from Demonstration Using a Multimodal Reward Function. In *IEEE-RAS International Conference on Humanoid Robots, 2013. Humanoids 2013.* . IEEE.

González-Fierro, M., Balaguer, C., Swann, N., and Nanayakkara, T. (2014a). Full-body postural control of a humanoid robot with both imitation learning and skill innovation. *International Journal of Humanoid Robotics*, 11(2):1450012.

González-Fierro, M., Bueno, J., Balaguer, C., and Moreno, L. (2013b). A complete 3d perception and path planning architecture for a humanoid. In *Proceedings of Robocity2030 11th Workshop: Robots Sociales*, pages 167–184.

González-Fierro, M., Hernández, D., Nanayakkara, T., and Balaguer, C. (2014b). Behavior sequencing based on demonstrations - a case of a humanoid opening a door while walking. *submitted to Advanced Robotics*.

González-Fierro, M., Hernández, D., Pierro, P., and Balaguer, C. (2012). Dynamic Modelling of Humanoid Robots Using Spatial Algebra. In *XXXIII Jornadas de Automática*. CEA.

González-Fierro, M., Jardón, A., Martínez de la Casa, S., Stoelen, M. F., Víctores, J. G., and Balaguer, C. (2010). Educational initiatives related with the ceabot contest. *Proceedings of SIMPAR*, pages 649–658.

González-Fierro, M., Maldonado, M. A., Víctores, J. G., Morante, S., and Balaguer, C. (2013c). Object tagging for human-robot interaction by recolorization using gaussian mixture models. In *Proceedings of Robocity2030 12th Workshop: Robótica Cognitiva*, pages 67–76.

González-Fierro, M., Monje, C., González, V., and Balaguer, C. (2013d). Evolutionary fractional order control of a humanoid robot modeled as a triple inverted pendulum. In *Proceedings of Robocity2030 11th Workshop: Robots Sociales*, pages 245–263.

González-Fierro, M., Monje, C. A., and Balaguer, C. (2013e). Robust control of a reduced humanoid robot model using genetic algorithms and fractional calculus. In *Mathematical Methods in Engineering International Conference MME2013*, pages 183–194.

González-Fierro, M., Monje, C. A., and Balaguer, C. (2014c). Fractional control of a humanoid robot reduced model with model disturbances. *submitted to Cybernetics and Systems*.

González-Fierro, M., Pierro, P., Jardón, A., Herrero, D., and Balaguer, C. (2009). Realización de tareas colaborativas entre robots humanoides. experimentación con dos robots robonova. In *Proceedings of Robocity2030 5th Workshop: Cooperación en Robótica*.

Gonzalez-Pacheco, V., Malfaz, M., Fernandez, F., and Salichs, M. A. (2013). Teaching human poses interactively to a social robot. *Sensors*, 13(9):12406–12430.

Gonzalez-Pacheco, V., Malfaz, M., and Salichs, M. A. (2014). Asking rank queries in pose learning. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 164–165. ACM.

Gribovskaya, E., Kheddar, A., and Billard, A. (2011). Motion learning and adaptive impedance for robot control during physical interaction with humans. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4326–4332. IEEE.

Gribovskaya, E., Zadeh, K., Mohammad, S., and Billard, A. (2010). Learning Nonlinear Multivariate Dynamics of Motion in Robotic Manipulators. *International Journal of Robotics Research*.

Grimes, D. B., Chalodhorn, R., and Rao, R. P. (2006). Dynamic imitation in a humanoid robot through nonparametric probabilistic inference. In *Robotics: science and systems*, pages 199–206. Cambridge, MA.

Grollman, D. H. and Billard, A. (2011). Donut as I do: Learning from failed demonstrations. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3804–3809. IEEE.

Guenter, F., Hersch, M., Calinon, S., and Billard, A. (2007). Reinforcement learning for imitating constrained reaching movements. *Advanced Robotics*, 21(13):1521–1544.

Harada, K., Kajita, S., Kanehiro, F., Fujiwara, K., Kaneko, K., Yokoi, K., and Hirukawa, H. (2007). Real-time planning of humanoid robot's gait for force-controlled manipulation. *Mechatronics, IEEE/ASME Transactions on*, 12(1):53–62.

Heyes, C. and Dawson, G. (1990). A demonstration of observational learning in rats using a bidirectional control. *The Quarterly Journal of Experimental Psychology*, 42(1):59–71.

Heyes, C., Jaldow, E., and Dawson, G. (1994). Imitation in rats: Conditions of occurrence in a bidirectional control procedure. *Learning and Motivation*, 25(3):276–287.

Hinton, G., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554.

Hirai, K. (1997). Current and future perspective of honda humamoid robot. In *Intelligent Robots and Systems, 1997. IROS'97., Proceedings of the 1997 IEEE/RSJ International Conference on*, volume 2, pages 500–508. IEEE.

Hirai, K., Hirose, M., Haikawa, Y., and Takenaka, T. (1998). The development of the honda humanoid robot. pages 1321–1326. Proceedings of IEEE International Conference on Robotics and Automation.

Hirose, R. and Takenaka, T. (2001). Development of the humanoid robot asimo. *Honda R&D Technical Review*, 13(1):1–6.

Hirukawa, H., Hattori, S., Harada, K., Kajita, S., Kaneko, K., Kanehiro, F., Fujiwara, K., and Morisawa, M. (2006). A universal stability criterion of the foot contact of legged robots-adios zmp. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 1976–1983. IEEE.

Hirukawa, H., Hattori, S., Kajita, S., Harada, K., Kaneko, K., Kanehiro, F., Morisawa, M., and Nakaoka, S. (2007). A pattern generator of humanoid robots walking on a rough terrain. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 2181–2187. IEEE.

Hirukawa, H., Kajita, S., Kanehiro, F., Kaneko, K., and Isozumi, T. (2005). The human-size humanoid robot that can walk, lie down and get up. *The International Journal of Robotics Research*, 24(9):755.

Hollerbach, J. M. (1980). A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity. *IEEE Trans. Syst., Man, and Cybern.*, pages 730–736.

Hollerbach, J. M., Johnson, T. L., Lozano-Perez, T., Mason, M. T., and Brady, M. (1982). *Robot Motion: Planning and Control*. MIT Press.

Hopper, L., Lambeth, S., Schapiro, S., and Whiten, A. (2008). Observational learning in chimpanzees and children studied through 'ghost'conditions. *Proceedings of the Royal Society B: Biological Sciences*, 275(1636):835–840.

Horak, F. B. (1987). Clinical measurement of postural control in adults. *Physical therapy*, 67(12):1881–1885.

Horak, F. B. (2006). Postural orientation and equilibrium: what do we need to know about neural control of balance to prevent falls? *Age and ageing*, 35(suppl 2):ii7–ii11.

Horak, F. B. and Macpherson, J. M. (1996). Postural orientation and equilibrium. *Comprehensive Physiology*.

I. Podlubny (1999). Fractional-Order Systems and PID Controllers. *IEEE Transactions on Automatic Control*, 44(1):208–214.

Ilg, W., Bakir, G. H., Mezger, J., and Giese, M. A. (2004). On the representation, learning and transfer of spatio-temporal movement characteristics. *International Journal of Humanoid Robotics*, 1(04):613–636.

Inaba, M., Igarashi, T., Kagami, S., and Inoue, H. (1996). A 35 DOF humanoid that can coordinate arms and legs in standing up, reaching and grasping an object. In *Intelligent Robots and Systems' 96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*, volume 1, pages 29–36. IEEE.

Jamone, L., Natale, L., Nori, F., Metta, G., and Sandini, G. (2012). Autonomous online learning of reaching behavior in a humanoid robot. *International Journal of Humanoid Robotics*, 9(03).

Jardón, A., Victores, J. G., Martínez, S., Giménez, A., and Balaguer, C. (2011). Personal autonomy rehabilitation in home environments by a portable assistive robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part C Applications and Reviews*, 42(4):561–570.

Jeannerod, M. (1988). *The neural and behavioural organization of goal-directed movements.* Clarendon Press/Oxford University Press.

Kahn, M. E. and Roth, B. (1971). The Near-Minimum-Time Control of Open-Loop Articulated Kinematic Chains. *J. Dynamic Systems, Measurement and Control*, (93):164–172.

Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., and Hirukawa, H. (2003a). Biped walking pattern generation by using preview control of zero-moment point. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 2, pages 1620–1626. Ieee.

Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., and Hirukawa, H. (2003b). Resolved momentum control: Humanoid motion planning based on the linear and angular momentum. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 2, pages 1644–1650. IEEE.

Kajita, S., Kanehiro, F., Kaneko, K., Yokoi, K., and Hirukawa, H. (2001). The 3D Linear Inverted Pendulum Mode: A simple Modelling for a biped walking pattern generation. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 1, pages 239–246. IEEE.

Kajita, S., Morisawa, M., Miura, K., Nakaoka, S., Harada, K., Kaneko, K., Kanehiro, F., and Yokoi, K. (2010). Biped walking stabilization based on linear inverted pendulum tracking. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4489–4496. IEEE.

Kajita, S. and Tani, K. (1991). Study of dynamic biped locomotion on rugged terrain-theory and basic experiment. In *Advanced Robotics, 1991.'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, pages 741–746. IEEE.

Kalakrishnan, M., Pastor, P., Righetti, L., and Schaal, S. (2013). Learning objective functions for manipulation. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 1331–1336. IEEE.

Kanehiro, F., Hirukawa, H., and Kajita, S. (2004). Openhrp: Open architecture humanoid robotics platform. *The International Journal of Robotics Research*, 23(2):155–165.

Kanehiro, F., Miyata, N., Kajita, S., Fujiwara, K., Hirukawa, H., Nakamura, Y., Yamane, K., Kohara, I., Kawamura, Y., and Sankai, Y. (2001). Virtual humanoid robot platform to develop controllers of real humanoid robots without porting. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 2, pages 1093–1099. IEEE.

Kaneko, K., Harada, K., Kanehiro, F., Miyamori, G., and Akachi, K. (2008). Humanoid robot hrp-3. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2471–2478. IEEE.

Kaneko, K., Kanehiro, F., Kajita, S., Hirukawa, H., Kawasaki, T., Hirata, M., Akachi, K., and Isozumi, T. (2004). Humanoid robot HRP-2. pages 1083–1090. Proceedings of IEEE International Conference on Robotics and Automation.

Kaneko, K., Kanehiro, F., Kajita, S., Yokoyama, K., Akachi, K., Kawasaki, T., Ota, S., and Isozumi, T. (2002). Design of prototype humanoid robotics platform for hrp. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 3, pages 2431–2436. IEEE.

Kaneko, K., Kanehiro, F., Morisawa, M., Akachi, K., Miyamori, G., Hayashi, A., and Kanehira, N. (2011). Humanoid robot hrp-4-humanoid robotics platform with lightweight and slim body. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 4400–4407. IEEE.

Kaneko, K., Kanehiro, F., Morisawa, M., Miura, K., Nakaoka, S., and Kajita, S. (2009). Cybernetic human hrp-4c. In *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, pages 7–14. IEEE.

Katayama, T., Ohki, T., Inoue, T., and Kato, T. (1985). Design of an optimal controller for a discrete time system subject to previewable demand. *Int. J. Control*, 41(3):677–699.

Katic, D. and Vukobratovic, M. (2003). Survey of Intelligent Control Techniques for Humanoid Robots. *Jounal of Intelligent and Robotics Systems*, (37):117–141.

Kato, I. (1973). The wabot-1. *Bulletin of Science and engineering Research Laboratory, Waseda University*, (62).

Kato, I. (1974). Information-power machine with senses and limbs (wabot 1). In *First CISM-IFToMM Symp. on Theory and Practice of Robots and Manipulators*, volume 1, pages 11–24. Springer-Verlag.

Kavraki, L. E., Svestka, P., Latombe, J.-C., and Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *Robotics and Automation, IEEE Transactions on*, 12(4):566–580.

Kaynov, D. (2008). *Open motion control architecture for humanoid robots*. PhD thesis, Universidad Carlos III de Madrid.

Kaynov, D., Rodríguez, M., Arbulú, M., Staroverov, P., Cabas, L., and Balaguer, C. (2006). Advanced motion control system for the humanoid robot rh-0. In *Climbing and Walking Robots*, pages 449–456. Springer.

Kaynov, D., Souères, P., Pierro, P., and Balaguer, C. (2009a). A practical decoupled stabilizer for joint-position controlled humanoid robots. In *IROS'09: Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems*, pages 3392–3397, Piscataway, NJ, USA. IEEE Press.

Kaynov, D., Soueres, P., Pierro, P., and Balaguer, C. (2009b). A practical decoupled stabilizer for joint-position controlled humanoid robots. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 3392–3397. IEEE.

Khansari-Zadeh, S. and Billard, A. (2011). Learning Stable Nonlinear Dynamical Systems With Gaussian Mixture Models. *Robotics, IEEE Transactions on*, 27(5):943–957.

Khansari-Zadeh, S. M. and Billard, A. (2010). BM: An Iterative Algorithm to Learn Stable Non-Linear Dynamical Systems with Gaussian Mixture Models. In *Proceeding of the International Conference on Robotics and Automation (ICRA)*, pages 2381–2388.

Khatib, O. (1987). A unified approach for motion and force control of robot manipulators: The operational space formulation. *Robotics and Automation, IEEE Journal of*, 3(1):43–53.

Khatib, O., Sentis, L., and Park, J. (2008). A unified framework for whole-body humanoid robot control with multiple constraints and contacts. In *European Robotics Symposium 2008*, pages 303–312. Springer.

Khatib, O., Sentis, L., Park, J., and Warren, J. (2004a). Whole-body dynamic behavior and control of human-like robots. *Int. J. of Humanoid Robotics*, 1(1):29–43.

Khatib, O., Warren, J., De Sapio, V., and Sentis, L. (2004b). Human-like motion from physiologically-based potential energies. In *On advances in robot kinematics*, pages 145–154. Springer.

Khoshelham, K. and Elberink, S. O. (2012). Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454.

Kim, J., Park, I., and Oh, J. (2007). Walking control algorithm of biped humanoid robot on uneven and inclined floor. *Journal of Intelligent & Robotic Systems*, 48(4):457–484.

Koenig, N. and Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2149–2154. IEEE.

Komura, T., Leung, H., Kudoh, S., and Kuffner, J. (2005). A feedback controller for biped humanoids that can counteract large perturbations during gait. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 1989–1995. IEEE.

Kormushev, P., Nenchev, D., Calinon, S., and Caldwell, D. (2011). Upper-body kinesthetic teaching of a free-standing humanoid robot. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3970–3975. IEEE.

Kuffner, J., Nishiwaki, K., Kagami, S., Inaba, M., and Inoue, H. (2005). Motion planning for humanoid robots. In *Robotics Research*, pages 365–374. Springer.

Kuffner, J. J. and LaValle, S. M. (2000). Rrt-connect: An efficient approach to single-query path planning. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 2, pages 995–1001. IEEE.

Kuffner Jr, J. J., Kagami, S., Nishiwaki, K., Inaba, M., and Inoue, H. (2002). Dynamically-stable motion planning for humanoid robots. *Autonomous Robots*, 12(1):105–118.

Kulic, D. and Nakamura, Y. (2009). Comparative study of representations for segmentation of whole body human motion data. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 4300–4305. IEEE.

Kulić, D., Takano, W., and Nakamura, Y. (2008). Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains. *The International Journal of Robotics Research*, 27(7):761–784.

Lanusse, P., Poinot, T., Cois, O., Oustaloup, A., and Trigeassou, J. (2003). Tuning of an Active Suspension System Using a Fractional Controller and a Closed-Loop Tuning. In *11th International Conference on Advanced Robotics*, pages 258–263, Coimbra, Portugal.

LaValle, S. M. (1998). Rapidly-exploring random trees a new tool for path planning.

Le, Q. V. (2013). Building high-level features using large scale unsupervised learning. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8595–8598. IEEE.

Lee, H., Grosse, R., Ranganath, R., and Ng, A. Y. (2009). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 609–616. ACM.

Lin, C., Chang, P., and Luh, J. (1983). Formulation and optimization of cubic polynomial joint trajectories for industrial robots. *Automatic Control, IEEE Transactions on*, 28(12):1066–1074.

Lin, H. and Lee, C. (2008). Self-organizing skill synthesis. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 828–833. IEEE.

Löffler, K., Gienger, M., and Pfeiffer, F. (2003). Sensors and control concept of walking "johnnie". *The International Journal of Robotics Research*, 22(3-4):229–239.

Lopes, M., Melo, F., and Montesano, L. (2009). Active learning for reward estimation in inverse reinforcement learning. In *Machine Learning and Knowledge Discovery in Databases*, pages 31–46. Springer.

Love, A. (2003). Evolutionary morphology, innovation, and the synthesis of evolutionary and developmental biology. *Biology and Philosophy*, 18(2):309–345.

Luh, J. Y. S., Walker, M. W., and Paul, R. P. C. (1980). On-line computational scheme for mechanical manipulators. *J. Dynamic Systems, Meas. and Control*, 102:69–76.

MacKinnon, C. D. and Winter, D. A. (1993). Control of whole body balance in the frontal plane during human walking. *Journal of Biomechanics*, 26(6):633 – 644.

Malekzadeh, M. S., Bruno, D., Calinon, S., Nanayakkara, T., and Caldwell, D. G. (2013). Skills transfer across dissimilar robots by learning context-dependent rewards. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1746–1751. IEEE.

Manabe, S. (1961). The Non-Integer Integral and its Application to Control System. *ETJ of Japan*, 6(3/4):83–87.

Martínez, S., Monje, C., Jardón, A., Pierro, P., Balaguer, C., and Muñoz, D. (2012). Teo: Full-size humanoid robot design powered by a fuel cell system. *Cybernetics and Systems*, 43(3):163–180.

Martínez de la Casa, S. (2012). *Human inspired humanoid robots control architecture*. PhD thesis, Universidad Carlos III of Madrid.

Martínez de la Casa, S., Jardon, A., and Balaguer, C. (2013). Human-inspired anticipative postural control architecture for humanoid robots. In *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, pages 4825–4830. IEEE.

Mataric, M. (1994). Reward functions for accelerated learning. In *Proceedings of the Eleventh International Conference on Machine Learning*, volume 189. San Francisco.

Mateo, A. P., González-Fierro, M., Hernández, D., Pierro, P., and Balaguer, C. (2010). Robust real time stabilization: Estabilización de la imagen con aplicación en el robot humanoide hoap-3. In *Proceedings of Robocity2030 7th Workshop: Visión en Robótica*.

Matsubara, T., Morimoto, J., Nakanishi, J., Hyon, S.-H., Hale, J. G., and Cheng, G. (2008). Learning to acquire whole-body humanoid center of mass movements to achieve dynamic tasks. *Advanced Robotics*, 22(10):1125–1142.

Medhi, J. (2010). *Stochastic processes*. New Age International, 3er edition.

Meltzoff, A. (1988). The human infant as Homo imitans. *Social learning: Psychological and biological perspectives*, pages 319–341.

Metta, G., Sandini, G., Natale, L., Craighero, L., and Fadiga, L. (2006). Understanding mirror neurons: a bio-robotic approach. *Interaction studies*, 7(2):197–232.

Michie, D. and Chambers, R. (1968). BOXES: An experiment in adaptive control. *Machine intelligence*, 2(2):137–152.

Minsky, M. (2006). The emotion machine. *New York: Pantheon*.

Mistry, M., Buchli, J., and Schaal, S. (2010a). Inverse dynamics control of floating base systems using orthogonal decomposition. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3406–3412. IEEE.

Mistry, M., Murai, A., Yamane, K., and Hodgins, J. (2010b). Sit-to-stand task on a humanoid robot from human demonstration. In *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, pages 218–223. IEEE.

Mistry, M., Schaal, S., and Yamane, K. (2009). Inertial parameter estimation of floating base humanoid systems using partial force sensing. In *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, pages 492–497. IEEE.

Mitchell, R. (1987). A comparative-developmental approach to understanding imitation. *Perspectives in ethology*, 7:183–215.

Mombaur, K., Truong, A., and Laumond, J. (2010). From human to humanoid locomotion—an inverse optimal control approach. *Autonomous robots*, 28(3):369–383.

Monje, C., Pierro, P., and Balaguer, C. (2011a). A new approach on human-robot collaboration with humanoid robot rh-2. *Robotica*, 29(6):949–957.

Monje, C., Ramos, F., Feliu, V., and Vinagre, B. (2007). Tip position control of a lightweight flexible manipulator using a fractional order controller. *Control Theory Applications, IET*, 1(5):1451–1460.

Monje, C., Vinagre, B., Santamaria, G., and Tejado, I. (2009a). Auto-tuning of fractional order pid controllers using a plc. In *Emerging Technologies Factory Automation, 2009. ETFA 2009. IEEE Conference on*, pages 1–7.

Monje, C. A., Calderón, A. J., Vinagre, B. M., Chen, Y. Q., and Feliu, V. (2004). On Fractional PID Controllers: Some Tuning Rules for Robustness to Plant Uncertainties. *Nonlinear Dynamics*, 38(1-4):369–381.

Monje, C. A., Chen, Y. Q., Xue, D., Vinagre, B. M., and Feliu, V. (2010). *Fractional-order Systems and Controls. Fundamentals and Applications*. Springer.

Monje, C. A., Pierro, P., and Balaguer, C. (2009b). Humanoid robot rh-1 for collaborative tasks: a control architecture for human-robot cooperation. *Applied Bionics and Biomechanics*, 5(4):225–234.

Monje, C. A., Pierro, P., Ramos, T., González-Fierro, M., and Balaguer, C. (2011b). Modelling and simulation of the humanoid robot hoap-3 in the openhrp3 platform. In *Proceedings of Robot 2011. Workshop Robots Humanoides*.

Monje, C. A., Pierro, P., Ramos, T., González-Fierro, M., and Balaguer, C. (2013). Modelling and simulation of the humanoid robot hoap-3 in the openhrp3 platform. *Cybernetics and Systems*, 44(8):663–680.

Monje, C. A., Vinagre, B. M., Calderón, A. J., Feliu, V., and Chen, Y. Q. (2005). Auto-Tuning of Fractional Lead-Lag Compensators. In *16th IFAC World Congress*, Prague.

Morimoto, J. and Doya, K. (2001). Acquisition of stand-up behavior by a real robot using hierarchical reinforcement learning. *Robotics and Autonomous Systems*, 36(1):37–51.

Muhlig, M., Gienger, M., Steil, J. J., and Goerick, C. (2009). Automatic selection of task spaces for imitation learning. *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4996–5002.

Nagasaka, K., Inoue, H., and Inaba, M. (1999). Dynamic walking pattern generation for a humanoid robot based on optimal gradient method. In *Systems, Man, and Cybernetics, 1999. IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on*, volume 6, pages 908–913. IEEE.

Nakaoka, S., Nakazawa, A., Kanehiro, F., Kaneko, K., Morisawa, M., Hirukawa, H., and Ikeuchi, K. (2007). Learning from observation paradigm: Leg task models for enabling a biped humanoid robot to imitate human dances. *The International Journal of Robotics Research*, 26(8):829–844.

Nakaoka, S., Nakazawa, A., Kanehiro, F., Kaneko, K., Morisawa, M., and Ikeuchi, K. (2005). Task model of lower body motion for a biped humanoid robot to imitate human dances. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3157–3162. IEEE.

Nakazawa, A., Nakaoka, S., Ikeuchi, K., and Yokoi, K. (2002). Imitating human dance motions through motion structure analysis. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 3, pages 2539–2544. IEEE.

Naksuk, N., Lee, C., and Rietdyk, S. (2005). Whole-body human-to-humanoid motion transfer. In *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, pages 104–109. IEEE.

Nanayakkara, T., Piyathilaka, C., Subasingha, A., and Jamshidi, M. (2007). Development of Advanced Motor Skills in a Group of Humans Through an Elitist Visual Feedback Mechanism. In *IEEE International Conference on Systems of Systems Engineering, San Antonio*.

Nanayakkara, T., Sahin, F., and Jamshidi, M. (2009). *Intelligent Control Systems with an Introduction to System of Systems Engineering*. CRC Press.

NASA (1995). Man-systems integration standards. Technical report, National Aeronautics and Space Administration.

Nashner, L. M. and McCollum, G. (1985). The organization of human postural movements: a formal basis and experimental synthesis. *Behavioral and brain sciences*, 8(01):135–150.

Nelson, G., Saunders, A., Swilling, B., Bondaryk, J., Billings, D., Lee, C., Playter, R., and Raibert, M. (2012). Petman: A humanoid robot for testing chemical protective clothing. *Journal of the Robotics Society of Japan*, 30:372–377.

Nelson, R. (1993). *National innovation systems: a comparative analysis*. Oxford University Press, USA.

Ng, A. and Russell, S. (2000). Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 663–670.

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. Y. (2011). Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 689–696.

Nguyen, N., Klein, E., and Zentall, T. (2005). Imitation of a two-action sequence by pigeons. *Psychonomic bulletin & review*, 12(3):514–518.

Nielsen, M., Mushin, I., Tomaselli, K., and Whiten, A. (2014). Where culture takes hold:"overimitation" and its flexible deployment in western, aboriginal, and bushmen children. *Child development*.

Ogura, Y., Aikawa, H., Kondo, H., Morishima, A., ok Lim, H., and Takanishi, A. (2006). Development of a new humanoid robot WABIAN-2. pages 76–81. International Conference on Proceedings of IEEE Robotics and Automation.

Oh, J.-H., Hanson, D., Kim, W.-S., Han, Y., Kim, J.-Y., and Park, I.-W. (2006). Design of android type humanoid robot albert hubo. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 1428–1433. IEEE.

Oliveira, E. and Nunes, L. (2004). Learning by exchanging Advice. *Studies in Fuzziness and Soft Computing*, 162:279–314.

OpenNI (2012). The largest 3D sensing development framework and community. http://www.openni.org [Online; accessed Apr-2014].

Orin, D. E., McGhee, R. B., Vukobratovic, M., and Hartoch, G. (1979). Kinematic and kinetic analysis of open-chain linkages utilizing newton-euler methods. *Mathematical Biosciences*, 43:107–130.

Oustaloup, A., Levron, F., Nanot, F., and Mathieu, B. (2000). Frequency-Band Complex Noninteger Differentiator: Characterization and Synthesis. *IEEE Transaction on Circuits and Systems I: Fundamental Theory and Applications*, 47(1):25–40.

Oustaloup, A., Mathieu, B., and Lanusse, P. (1995). The CRONE Control of Resonant Plants: Application to a Flexible Transmission. *European Journal of Control*, 1(2):113–121.

Oustaloup, A., Moreau, X., and Nouillant, M. (1996). The CRONE Suspension. *IFAC Journal of Control Engineering Practice*, 4(8):1101–1108.

Oustaloup, A., Sabatier, J., and Lanusse, P. (1999). From Fractal Robustness to the CRONE Control. *Fractional Calculus and Applied Analysis: An International Journal for Theory and Applications*, 2(1):1–30.

Padula, F. and Visioli, A. (2011). Tuning rules for optimal pid and fractional-order pid controllers. *Journal of Process Control*, 21(1):69–81.

Pan, W., Chengzhi, X., and Zhun, F. (2004). Evolutionary linear control strategies of triple inverted pendulums and simulation studies. In *Intelligent Control and Automation, 2004. WCICA 2004. Fifth World Congress on*, volume 3, pages 2365–2368. IEEE.

Pardowitz, M., Zollner, R., and Dillmann, R. (2005). Learning sequential constraints of tasks from user demonstrations. In *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, pages 424–429. Ieee.

Pardowitz, M., Zöllner, R., and Dillmann, R. (2006). Incremental learning of task sequences with information-theoretic metrics. In *European Robotics Symposium 2006*, pages 51–63. Springer.

Peña, A., Hernández, D., González-Fierro, M., Pierro, P., and Balaguer, C. (2010). Sistema de visión del humanoide hoap-3 para la detección e identificación de objetos mediante librerías opencv. In *Proceedings of Robocity2030 7th Workshop: Visión en Robótica*.

Peters, J. and Schaal, S. (2008a). Learning to control in operational space. *The International Journal of Robotics Research*, 27(2):197–212.

Peters, J. and Schaal, S. (2008b). Natural actor-critic. *Neurocomputing*, 71(7):1180–1190.

Peters, J. and Schaal, S. (2008c). Reinforcement learning of motor skills with policy gradients. *Neural networks*, 21(4):682–697.

Peters, J., Vijayakumar, S., and Schaal, S. (2003). Reinforcement learning for humanoid robotics. In *Proceedings of the third IEEE-RAS international conference on humanoid robots*, pages 1–20.

Petráš, I., Podlubny, I., O'Leary, P., and Dorčák, L. (2001). Analogue Fractional-Order Controllers: Realization, Tuning and Implementation. In *Proceedings of the ICCC'2001*, pages 9–14, Krynica, Poland.

Petráš, I. and Vinagre, B. M. (2002). Practical Application of Digital Fractional-Order Controller to Temperature Control. *Acta Montanistica Slovaca*, 7(2):131–137.

Petráš, I., Vinagre, B. M., Dorčák, L., and Feliu, V. (2002). Fractional Digital Control of a Heat Solid: Experimental Results. In *International Carpathian Control Conference*, pages 365–370, Malenovice, Czech Republic.

Piaget, J. (1962). *Play, dreams and imitation*, volume 24. New York: Norton.

Pierro, P. (2012). *Stabilizer architecture for humanoid robots collaborating with humans*. PhD thesis, Universidad Carlos III de Madrid.

Pierro, P., González-Fierro, M., and Balaguer, C. (2009a). El proyecto europeo robot@cwe: Advanced robotic systems in future collaborative working environments. In *Proceedings of Robot 2009. II Workshop de Robotica (ROBOT 2009)*.

Pierro, P., Hernández, D., González-Fierro, M., Blasi, L., Milani, A., and Balaguer, C. (2009b). A human-humanoid interface for collaborative tasks. In *Proceedings on the Second workshop for young researchers on Human-friendly robotics*, Sestri Levante, Italy.

Pierro, P., Hernández, D., González-Fierro, M., Blasi, L., Milani, A., and Balaguer, C. (2009c). Humanoid teleoperation system for space environments. In *Advanced Robotics, 2009. ICAR 2009. International Conference on*, pages 1–6. IEEE.

Pierro, P., Hernández, D., Herrero, D., González-Fierro, M., and Balaguer, C. (2012). Perception system for working with humanoid robots in unstructured collaborative scenarios. In *Proceedings of the 2012 International IEEE Intelligent Vehicles Symposium. Workshops V Perception in Robotics*. IEEE.

Pierro, P., Monje, C. A., and Balaguer, C. (2008). Modelling and control of the humanoid robot rh-1 for collaborative tasks. In *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on*, pages 125–131. IEEE.

Podlubny, I. (1999). *Fractional Differential Equations*. Academic Press.

Pollard, N., Hodgins, J., Riley, M., and Atkeson, C. (2002). Adapting human motion for the control of a humanoid robot. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 2, pages 1390–1397. IEEE.

Pommier, V., Sabatier, J., Lanusse, P., and Oustaloup, A. (2002). Crone Control of a Nonlinear Hydraulic Actuator. *Control Engineering Practice*, 10:391–402.

Puga, J. R., Silva, F. M., and Santos, V. M. (2006). Motion planning and control strategies for a distributed architecture humanoid robot. In *Robot Control*, volume 8, pages 773–778.

Ratliff, N., Ziebart, B., Peterson, K., Bagnell, J. A., Hebert, M., Dey, A. K., and Srinivasa, S. (2009). Inverse optimal heuristic control for imitation learning. In *International Conference on Artificial Intelligence and Statistics*. AISTATS.

Rietdyk, S., Patla, A., Winter, D., Ishac, M., and Little, C. (1999). Balance recovery from medio-lateral perturbations of the upper body during standing. *Journal of Biomechanics*, 32(11):1149 – 1158.

Rizzolatti, G., Fadiga, L., Fogassi, L., and Gallese, V. (2002). From mirror neurons to imitation: Facts and speculations. *The imitative mind: Development, evolution, and brain bases*, 6:247.

Rusu, R. B. (2010). Semantic 3d object maps for everyday manipulation in human living environments. *KI-Künstliche Intelligenz*, 24(4):345–348.

Sakagami, Y., Watanabe, R., Aoyama, C., Matsunaga, S., Higaki, N., and Fujimura, K. (2002). The intelligent ASIMO: System overview and integration. pages 2478–2483. IEEE/RSJ international conference on intelligent robots and systems.

Saunders, J., Nehaniv, C., and Dautenhahn, K. (2006). Teaching robots by moulding behavior and scaffolding the environment. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 118–125. ACM.

Schaal, S. (1999). Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3(6):233–242.

Schaal, S. and Atkeson, C. (2010). Learning control in robotics. *Robotics & Automation Magazine, IEEE*, 17(2):20–29.

Schaal, S., Ijspeert, A., and Billard, A. (2003). Computational approaches to motor learning by imitation. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 358(1431):537–547.

Schaal, S., Mohajerian, P., and Ijspeert, A. (2007). Dynamics systems vs. optimal control - a unifying view. In Paul Cisek, T. D. and Kalaska, J. F., editors, *Computational Neuroscience: Theoretical Insights into Brain Function*, volume 165 of *Progress in Brain Research*, pages 425–445. Elsevier.

Schaal, S., Peters, J., Nakanishi, J., and Ijspeert, A. (2005). Learning movement primitives. *Robotics Research*, pages 561–572.

Sciavicco, L., Villani, L., Oriolo, G., and Siciliano, B. (2009). *Robotics. Modelling, Planning and Control*. Springer.

Sentis, L. (2007). *Synthesis and control of whole-body behaviors in humanoid systems*. PhD thesis, Stanford University.

Sentis, L. and Khatib, O. (2004). Task-oriented control of humanoid robots through prioritization. *International Journal of Humanoid Robotics*.

Sentis, L. and Khatib, O. (2005). Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *International Journal of Humanoid Robotics*, 2(4):505–518.

Sentis, L., Park, J., and Khatib, O. (2010). Compliant control of multicontact and center-of-mass behaviors in humanoid robots. *Robotics, IEEE Transactions on*, 26(3):483–501.

Shiratori, T., Kudoh, S., Nakaoka, S., and Ikeuchi, K. (2007). Temporal scaling of upper body motion for sound feedback system of a dancing humanoid robot. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 3251–3257. IEEE.

Shumway, C. (2000). Motor control: theory and practical applications. *Recherche*, 67:02.

Siciliano, B. and Khatib, O. (2008). *Springer handbook of robotics*. Springer-Verlag New York Inc.

Silva, F. and Santos, V. (2005). Towards an autonomous small-size humanoid robot: design issues and control strategies. In *Computational Intelligence in Robotics and Automation, 2005. CIRA 2005. Proceedings. 2005 IEEE International Symposium on*, pages 87–92. IEEE.

So, B. R., Choi, J. Y., Yi, B.-J., and Kim, W. (2005). A new zmp constraint equation with application to motion planning of humanoid using kinematic redundancy. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 4021–4027. IEEE.

Staroverov, P., Kaynov, D., Arbulú, M., Cabas, L., and Balaguer, C. (2007a). Creating a gesture recognition system based on shirt shapes. In *Proc. 8th International Conference on Climbing and Walking Robots (Clawar 2007), Singapore*. World Scientific.

Staroverov, P., Martínez, R., Kaynov, D., Arbulú, M., Cabas, L., and Balaguer, C. (2007b). Detecting sound sources with the humanoid robot rh-1. In *Proceedings of 10th International Conference (CLAWAR 2007)*, volume 16, page 18. World Scientific.

Stepanenko, Y. and Vukobratovic, M. (1976). Dynamics of articulated open-chain active mechanisms. *Mathematical Biosciences*, 28:137–170.

Stephens, B. (2007). Integral control of humanoid balance. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 4020–4027. IEEE.

Stephens, B. J. and Atkeson, C. G. (2010). Dynamic balance force control for compliant humanoid robots. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1248–1255. IEEE.

Storn, R. and Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359.

Stulp, F., Evangelos, T., Stefan, S., et al. (2012). Reinforcement learning with sequences of motion primitives for robust manipulation. *IEEE Transactions on Robotics*, 28(6):1360–1370.

Sugihara, T. and Nakamura, Y. (2003). Variable impedant inverted pendulum model control for a seamless contact phase transition on humanoid robot. In *IEEE International Conference on Humanoid Robots (Humanoids2003)*, pages 0–0.

Sugihara, T., Nakamura, Y., and Inoue, H. (2002). Real-time humanoid motion generation through zmp manipulation based on inverted pendulum control. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 2, pages 1404–1409. IEEE.

Suleiman, W., Yoshida, E., Kanehiro, F., Laumond, J., and Monin, A. (2008). On human motion imitation by humanoid robot. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 2697–2704. IEEE.

Sung, H. G. (2004). *Gaussian mixture regression and classification*. PhD thesis, Rice University.

Sutton, R. (1988). Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44.

Sutton, R. and Barto, A. (1998). *Reinforcement learning: An introduction*, volume 1. Cambridge Univ Press.

Takahashi, Y., Tamura, Y., Asada, M., and Negrello, M. (2010). Emulation and behavior understanding through shared values. *Robotics and Autonomous Systems*, 58(7):855–865.

Tang, H., Xue, S., and Fan, C. (2008). Differential evolution strategy for structural system identification. *Computers & Structures*, 86(21-22):2004–2012.

Taylor, J. G., Cutsuridis, V., Hartley, M., Althoefer, K., and Nanayakkara, T. (2013). Observational learning: basis, experimental results and models, and implications for robotics. *Cognitive Computation*, 5(3):340–354.

Thompson, D. and Russell, J. (2004). The ghost condition: imitation versus emulation in young children's observational learning. *Developmental Psychology*, 40(5):882.

Tomasello, M. (1996). Do apes ape?

Tsachouridis, V. (1999). Robust control of a triple inverted pendulum. In *Control Applications, 1999. Proceedings of the 1999 IEEE International Conference on*, volume 2, pages 1235–1240. IEEE.

Uicker, J. J. (1967). Dynamic Force analysis of spatial linkages. *J. Appl. Mechanics*, (34):418–424.

Valério, D. (2001). Fractional Order Robust Control: An Application. In *Student Forum*, pages 25–28, Porto.

Valério, D. and da Costa, J. S. (2006). Tuning of fractional pid controllers with ziegler–nichols-type rules. *Signal Processing*, 86(10):2771–2784.

Víctores, J. G., Morante, S., González-Fierro, M., and Balaguer, C. (2013). Augmented reality and social interaction platform through multirobot design. In *Proceedings of Robocity2030 11th Workshop: Robots Sociales*, pages 131–143.

Vukobratovic, M. and Borovac, B. (2004). Zero-moment point - Thirty five years of its life. *International Journal of Humanoid Robotics*, 1(1):157–173.

Vukobratovic, M., Frank, A., and Juricic, D. (1970). On the stability of biped locomotion. *Biomedical Engineering, IEEE Transactions on*, (1):25–36.

Vukobratovic, M. and Juricic, D. (1968). Contribution to the synthesis of biped gait. In *Proceeding of the I.F.A.C. International Symposium on Technical and Biological Problem of Control*. IEEE.

Vukobratovic, M. and Juricic, D. (1969). Contribution to the synthesis of biped gait. *Biomedical Engineering, IEEE Transactions on*, (1):1–6.

Vukobratović, M. and Stepanenko, J. (1972). On the stability of anthropomorphic systems. *Mathematical Biosciences*, 15(1):1–37.

Walker, M. W. and Orin, D. E. (1982). Efficient dynamic computer simulation of robotic mechanisms. *J. Dyn. Systems, Meas. and Control, ASME Trans*, 104(3):205–211.

Watkins, C. and Dayan, P. (1992). Q-learning. *Machine learning*, 8(3):279–292.

Wen, S., Chen, X., Zhao, Y., Rad, A. B., Othman, K. M., and Zhang, E. (2014). The study of fractional order controller with slam in the humanoid robot. *Advances in Mathematical Physics*, 2014.

West, M. (1990). The social psychology of innovation in groups.

Whiten, A., Custance, D., Gomez, J., Teixidor, P., and Bard, K. (1996). Imitative learning of artificial fruit processing in children (Homo sapiens) and chimpanzees (Pan troglodytes). *Journal of Comparative Psychology*, 110(1):3–14.

Whiten, A. and Ham, R. (1992). On the nature and evolution of imitation in the animal kingdom: reappraisal of a century of research. *Advances in the Study of Behavior*, 21:239–283.

Whiten, A., Horner, V., Litchfield, C., and Marshall-Pescini, S. (2004). How do apes ape? *Learning & Behavior*, 32(1):36–52.

Whiten, A., McGuigan, N., Marshall-Pescini, S., and Hopper, L. (2009). Emulation, imitation, over-imitation and the scope of culture for child and chimpanzee. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1528):2417–2428.

Wieber, P.-B. (2006). Trajectory free linear model predictive control for stable walking in the presence of strong perturbations. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 137–142. IEEE.

Winter, D. A. (2009). *Biomechanics and motor control of human movement*. John Wiley & Sons.

Xiaofeng, G., Hongxing, L., Guannan, D., and Haigang, G. (2009). Variable universe adaptive fuzzy control on the triple inverted pendulum and choosing contraction-expansion factor. In *Artificial Intelligence and Computational Intelligence, 2009. AICI'09. International Conference on*, volume 4, pages 63–67. IEEE.

Yamaguchi, A., Hyon, S., and Ogasawara, T. (2010). Reinforcement learning for balancer embedded humanoid locomotion. In *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, pages 308–313. IEEE.

Yamane, K. and Nakamura, Y. (2003). Dynamics filter-concept and implementation of online motion generator for human figures. *Robotics and Automation, IEEE Transactions on*, 19(3):421–432.

Yi, S.-J., Zhang, B.-T., Hong, D., and Lee, D. D. (2011). Online learning of a full body push recovery controller for omnidirectional walking. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages 1–6. IEEE.

Yokoya, R., Ogata, T., Tani, J., Komatani, K., and Okuno, H. G. (2006). Experience based imitation using rnnpb. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 3669–3674. IEEE.

Yoshida, E., Belousov, I., Esteves, C., and Laumond, J.-P. (2005). Humanoid motion planning for dynamic tasks. In *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, pages 1–6. IEEE.

Yoshikawa, T. and Khatib, O. (2009). Compliant humanoid robot control by the torque transformer. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 3011–3018. IEEE.

Zentall, T. (2006). Imitation: definitions, evidence, and mechanisms. *Animal cognition*, 9(4):335–353.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., and Dey, A. K. (2008). Maximum entropy inverse reinforcement learning. In *AAAI*, pages 1433–1438.