

CARLOS III UNIVERSITY

DEPARTMENT OF SYSTEMS AND AUTOMATION ENGINEERING



**MASTER THESIS**

**A STUDY ON SIMPLIFIED AND COMPLETE  
DYNAMIC MODELS OF HUMANOID ROBOTS**

Author: Tamara Ramos Cambero

Director: Dr. Concepción Alicia Monje Micharet

OFFICIAL MASTER IN  
ELECTRICAL, ELECTRONIC AND AUTOMATION ENGINEERING

LEGANÉS, MADRID

MAY 2012





CARLOS III UNIVERSITY  
OFFICIAL MASTER IN ELECTRICAL, ELECTRONIC AND  
AUTOMATION ENGINEERING

This work was done wholly or mainly while in candidature for a research degree at this University. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated. Where it has been consulted the published work of others, this is always clearly attributed.

---

The court accepts this Master Thesis called "**A Study on Simplified and Complete Dynamic Models of Humanoid Robots**" written by **Tamara Ramos Cambero** as a requirement to obtain the Degree in the official Master in Electrical, Electronic and Automation Engineering.

Date: May 2012

Tribunal:

---

Dr. María Dolores Blanco Rojas

---

Dr. María Ángeles Malfaz Vázquez

---

Dr. Luis Mengibar Pozo



*To my grandparents*



# Contents

<b>List of Figures</b>	<b>x</b>
<b>Acknowledgements</b>	<b>xv</b>
<b>Abstract</b>	<b>xvii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 State of the art . . . . .	2
1.3 Objectives . . . . .	7
1.4 Structure of the document . . . . .	9
<b>2 Simplified Models of Humanoid Robots</b>	<b>11</b>
2.1 Introduction . . . . .	11
2.2 ZMP and COG concepts . . . . .	11
2.3 Single inverted pendulum model . . . . .	13
2.4 Double inverted pendulum model . . . . .	16
2.5 Cart-table model . . . . .	19
2.6 Chapter summary . . . . .	20

<b>3</b>	<b>Simulation of Simplified Models</b>	<b>21</b>
3.1	Introduction . . . . .	21
3.2	Software for simulation . . . . .	21
3.2.1	Simulink . . . . .	22
3.2.2	SimMechanics . . . . .	22
3.3	Simplified models in simulation . . . . .	22
3.3.1	Single inverted pendulum model . . . . .	23
3.3.1.1	M-File . . . . .	23
3.3.1.2	Simulink . . . . .	24
3.3.1.3	SimMechanics . . . . .	25
3.3.2	Control of the single inverted pendulum . . . . .	27
3.3.2.1	M-File . . . . .	27
3.3.2.2	Simulink . . . . .	29
3.3.2.3	Simulation results . . . . .	30
3.3.3	Double inverted pendulum model . . . . .	31
3.3.3.1	M-File . . . . .	32
3.3.3.2	Simulink . . . . .	32
3.3.3.3	SimMechanics . . . . .	34
3.3.4	Cart-table model . . . . .	36
3.3.4.1	M-File . . . . .	37
3.3.4.2	Simulink . . . . .	37
3.3.4.3	SimMechanics . . . . .	38
3.3.4.4	Simulation results . . . . .	41
3.4	Chapter summary . . . . .	43
<b>4</b>	<b>Ankle Prototype as Experimental Platform</b>	<b>45</b>
4.1	Introduction . . . . .	45
4.2	Hardware description . . . . .	47
4.3	Simulation results . . . . .	48
4.4	Experimental results . . . . .	51

4.5	Discussion of results . . . . .	51
4.6	Chapter summary . . . . .	52
<b>5</b>	<b>Humanoid Robot TEO as Experimental Platform</b>	<b>55</b>
5.1	Introduction . . . . .	55
5.2	Hardware description . . . . .	57
5.3	Simulation results . . . . .	60
5.4	Experimental results . . . . .	63
5.5	Discussion of results . . . . .	64
5.6	Chapter summary . . . . .	64
<b>6</b>	<b>OpenHRP3 Simulation Platform for Modelling and Test Validation of Humanoid Robots</b>	<b>67</b>
6.1	Introduction . . . . .	67
6.2	OpenHRP3 platform . . . . .	70
6.2.1	OpenHRP3 simulation interface . . . . .	72
6.3	Creating models in OpenHRP3 . . . . .	73
6.3.1	TEO model . . . . .	74
6.3.2	HOAP-3 model . . . . .	74
6.4	Simulation and experimental results . . . . .	74
6.4.1	Simulation and experimental results for TEO . . . . .	78
6.4.2	Simulation and experimental results for HOAP-3 . . . . .	78
6.5	Chapter summary . . . . .	84
<b>7</b>	<b>Conclusions and Future Works</b>	<b>85</b>
7.1	Conclusions . . . . .	85
7.2	Future works . . . . .	86
	<b>References</b>	<b>89</b>





# List of Figures

1.1	The humanoid robot RH-1 . . . . .	3
2.1	Forces acting on the foot of the bipedal mechanism . . . . .	12
2.2	Forces acting on a humanoid (sagittal plane) . . . . .	13
2.3	Single inverted pendulum model . . . . .	14
2.4	Double inverted pendulum model . . . . .	17
2.5	Cart-table model . . . . .	20
3.1	Single inverted pendulum m-file . . . . .	23
3.2	Single inverted pendulum Simulink model . . . . .	24
3.3	Subsystem . . . . .	25
3.4	Single inverted pendulum SimMechanics model . . . . .	25
3.5	Single inverted pendulum SimMechanics mechanical model . . . . .	26
3.6	Single inverted pendulum controlled m-file . . . . .	28
3.7	Motor model file . . . . .	28
3.8	Single inverted pendulum controlled Simulink model . . . . .	29
3.9	Subsystem . . . . .	30
3.10	Single inverted pendulum controlled Simulink model for simulation . . . . .	30
3.11	Simulation results in Simulink . . . . .	31

3.12	Double inverted pendulum m-file . . . . .	32
3.13	Double inverted pendulum Simulink model . . . . .	33
3.14	Subsystem . . . . .	33
3.15	Double inverted pendulum SimMechanics model . . . . .	34
3.16	Subsystem . . . . .	35
3.17	Double inverted pendulum SimMechanics mechanical model . .	36
3.18	Cart-table m-file . . . . .	36
3.19	Cart-table Simulink model . . . . .	37
3.20	Subsystem . . . . .	38
3.21	Cart-table SimMechanics model . . . . .	39
3.22	Subsystem . . . . .	39
3.23	Cart-table SimMechanics mechanical model . . . . .	40
3.24	Cart-table machine in simulation . . . . .	41
3.25	Obtained ZMP for X axis in Simulink and SimMechanics models	42
3.26	Obtained ZMP for Y axis in Simulink and SimMechanics models	42
4.1	Prototype of the ankle . . . . .	46
4.2	Motors and encoders of the prototype of the ankle . . . . .	47
4.3	Hardware structure of the prototype . . . . .	48
4.4	Simulation results of the single inverted pendulum . . . . .	49
4.5	Simulation of the SimMechanics mechanical model . . . . .	50
4.6	Test result of the ankle prototype . . . . .	50
4.7	Velocity control of the drivers of the ankle prototype . . . . .	52
4.8	Obtained results after a test . . . . .	53
5.1	Distribution of DOF of the robot TEO . . . . .	56
5.2	Lower part body of the humanoid robot TEO . . . . .	57
5.3	SBC Gemini 5.2" Embedded Intel Core 2 Duo of ARCOM micro-processor . . . . .	58
5.4	Maxon Brushless EC45 Flat 251601 motor . . . . .	58

5.5	ISCM8005 driver . . . . .	59
5.6	7500 AEAS absolute encoder . . . . .	59
5.7	JR3 force-torque sensor . . . . .	60
5.8	References for the simulation of the double inverted pendulum . . . . .	61
5.9	Simulation results of the DOF of the ankle in SimMechanics . . . . .	62
5.10	Simulation result of the DOF of the hip in SimMechanics . . . . .	62
5.11	Simulation of the SimMechanics mechanical model . . . . .	63
5.12	Sequence of motions of the robot TEO . . . . .	64
5.13	Positions of the ankle of the robot TEO . . . . .	65
5.14	Positions of the hip of the robot TEO . . . . .	65
6.1	OpenHRP3 functions . . . . .	71
6.2	OpenHRP3 interface . . . . .	73
6.3	VRML structure of TEO . . . . .	75
6.4	Humanoid robot HOAP-3 . . . . .	76
6.5	Distribution of DOF in HOAP-3 (up) and VRML structure (down) . . . . .	77
6.6	Trajectories for both right and left legs joints (12 DOF), in OpenHRP3 and real TEO . . . . .	79
6.7	Sequence of motions for TEO during the performance . . . . .	80
6.8	Trajectories for both right and left legs joints (12 DOF), in OpenHRP3 and real HOAP-3 . . . . .	81
6.9	Sequence of motions for HOAP-3 during the performance . . . . .	82
6.10	Sequence of motions for HOAP-3 touching a wall (up); Torques of right (up) and left (down) wrists during the touching motion (down) . . . . .	83
6.11	Sequence of motions for HOAP-3 in front of a wall . . . . .	83



# Acknowledgements

There are many people who helped me in producing this work. First of all, I would like to thank my director, Dr. Concepción Alicia Monje Micharet, for having faith in me. Thanks for being so patient with me and supporting me throughout. Without her guidance, this work would have never materialized and been completed. I really appreciate her help.

I extend my sincere gratitude to Prof. Carlos Balaguer for giving me the chance to continue my research at RoboticsLab.

Many thanks to all the colleagues at RoboticsLab, it has been a great pleasure to work with them. I would especially like to thank all my colleagues, who have given me all their support during the good and the bad days. I would also like to thank all my friends, for their understanding and encouragement, as well as for providing much entertainment.

Finally, I would like to thank my family, especially my grandparents. They never doubted me for a minute and their confidence helped me in rough times.

This work has been supported by the CYCIT Project PI2004-00325 and the European Project Robot@CWE FP6-2005-IST-5, both developed by the research team RoboticsLab at Carlos III University of Madrid.



# Abstract

Dynamics is a fundamental part on humanoid robotics. It relates torques with joint accelerations that appear in the system. Due to the high computational cost of obtaining the complete dynamic model of a humanoid robot, several simplified models have been developed.

The aim of this work is to study different dynamics models for humanoid robots, both simplified and complete ones.

First of all, several simplified models have been studied and implemented in simulation. On the other hand, a simulation platform for humanoid robots has been used for the creation of the complete models.

Finally, the behaviour of experimental platforms have been compared with the behaviour of simplified models in simulation. These experimental platforms have been the humanoid robot TEO (lower part body), the ankle prototype of TEO, and the miniature humanoid robot HOAP-3.





# Introduction

## 1.1 Motivation

Human evolution to bipedalism is an option for developing intelligent machines. Robotics researchers, in order to imitate human intelligence, have the way by developing biped robots, because they are following natural human evolution. The human tries to imitate it by developing humanoid robots with similar capabilities for working in its environments, to create an anthropomorphic partner or assistant.

To achieve this goal, the study of the forces and moments in the robot is very important in this field. The problem of the dynamics in a humanoid robot is a required topic and plays an important role in the control of the motion of the robot.

The dynamics is defined by the equation of motion, which relates torques with joint accelerations. Using the direct dynamic, the accelerations are obtained based on the torques. The inverse dynamics performs the opposite operation, and allows obtaining the torques based on the accelerations. This is usually used for generation of trajectories, control of stability, and mechanical design of the robot.

The aim of this work is to study different dynamic models for humanoid robots, starting with the simplest models (mass concentrated models), and finally using the complete models. For this purpose, a study of the main simplified models of humanoid robots is presented here, and they have been implemented for simulation. For the creation of the complete dynamic models, a simulation platform for humanoid robots has been used. Finally, the behaviour of experimental platforms have been compared with the behaviour of simplified models in simulation. These experimental platforms have been the humanoid robot TEO (lower part body), the ankle prototype of TEO, and the miniature humanoid robot HOAP-3.

## 1.2 State of the art

Humanoids have drawn a lot of attention from robotics researchers as their research target. Also, they made people imagine various dreamy applications. However, because of the difficulty of achieving a stable and reliable biped walking function, they were considered just a dreamy research target for researchers. Since Honda developed a reliable humanoid, the reputation of humanoids has been changing from just a research target to a machine which can be used for practical applications.

There are mainly two types of research on humanoids. The first one is related to scientific interests. In this research, humanoid is pursued to investigate what human intelligence is and to understand human behaviour in computational scientific ways. MIT, NASA, ERATO and the ATR groups are working on humanoids from this point of view (those kind of humanoids are not our reference).

The second type deals with developing a humanoid to use it practically. The new trend of humanoid research has been made from the second type of research. In such a trend, the Ministry of Economy, Trade and Industry (METI)



**Figure 1.1:** *The humanoid robot RH-1*

in Japan started the Humanoid Robotics Project (HRP) to find real practical applications (Tanie, 1998), after that, HONDA demonstrated the feasibility of this kind of robot, which is based on preliminary studies at Waseda University.

The robotics group RoboticsLab, at the Carlos III University of Madrid, has been working for many years on the RH-project, a robust and open humanoid platform for the research on biped walking, balancing control, sensor fusion, human-robot interaction (HRI) for collaborative tasks, and other related issues.

The first robot of the project is RH-1, shown in Figure 1.1. It is an anthropomorphic robot with 21 Degrees Of Freedom (DOF), a height of 1.5m, and a weight of about 50kg. The main research objectives of this platform have been

the stability of the robot (Kaynov, Soueres, Pierro, & Balaguer, 2009) and the gait generation (Arbulu & Balaguer, 2007), though other research activities have focused on human-robot interaction (Staroverov, Kaynov, Arbulú, Cabas, & Balaguer, 2007) and collaboration (Pierro, Monje, & Balaguer, 2008). In fact, this version, together with a previous one RH-0, aim at studying stable walking and do not consider upper part movements related to physical collaboration. Although stable walking has been achieved in this platform (Arbulu & Balaguer, 2007), several aspects have required an improvement. As a main problem, the mechanical structure of this prototype was not very robust and presented limitations such as the high flexibility of the whole body and the joints looseness, which not only complicated the stability control but also limited the range of applications to be performed with the robot, not to mention the increase of energy consumption.

However, the most relevant humanoid platforms came out as a step by step improvement of different versions, such as Asimo (Sakagami et al., 2002), HRP-3 (Kaneko, Harada, Kanehiro, Miyamori, & Akachi, 2008), Hubo (Park, Kim, Lee, & Oh, 2005), or Wabian-2 (Ogura et al., 2006). Other important platforms were specifically designed for peculiar applications: Jonnie (Pfeiffer, Loffler, & Gienger, 2002) was initially designed for fast-walking, and i-Cub (Metta, Sandini, Vernon, Natale, & Nori, 2008) for research on embodied cognition. In this line, the new prototype TEO (Task Environment Operator) appears as an improved version of its predecessor RH-1. The humanoid robot TEO (Pérez et al., 2009) addresses challenges in the fields of motion, safety, energy efficiency, and power autonomy performance. The mechatronics of this platform is inspired by the human natural and adaptive locomotion, and its design is oriented to achieve human physical capacities and performances.

The internal model that runs simultaneously with the robot takes the same information about the desired robot motion as the real hardware and based on the dynamical model of the system, it calculates the required nominal control

inputs that are needed to realize the desired joint trajectories. It is clear that the performance of the robot depends very much on the accuracy of the model, that is, how well the model represents the real-world system. This is the reason that many dynamical models have been developed for the bipedal robots.

Modelling of human body walk has evolved from simple models such as an inverted pendulum model (Kuo, 2007), (Garcia, Chatterjee, Ruina, & Coleman, 1998), (Kuo, 2001) and a mass-spring model (Geyer, Seyfarth, & Blickhan, 2005), (Whittington & Thelen, 2009), to relatively complicated models that include relatively high number of DOF (Shirata, Konno, & Uchiyama, 2004), (Yamaguchi & Takanishi, 1997), (Collins & Ruina, 2005), (Peasgood, Kubica, & McPhee, 2007). Primary goal of those models is to predict the internal and external forces during a regular walking cycle. Detailed human body models can include calculations of the most important muscle forces for particular types of motion. However, for a robot that is actuated via rotational motors, calculation of particular muscle forces is not needed, but the moments they produce about the corresponding joints. There are two main reasons for the calculation of the joint torques.

(a) Based on a walking model, one can predict maximum torques that are necessary to generate particular motion of a robotic structure. The maximum torque and maximum power are necessary data to choose joint actuators. The procedure of the actuators (motors) selection based on the robot dynamic model is inherently iterative, since the mass distribution and consequently, the static and dynamic forces are significantly dependent on the sizes and locations of the chosen actuators.

(b) Torques at relevant joints for particular motion, calculated based on a walking model, can be used to generate nominal control trajectories for a complex walking robotic architecture. Those nominal trajectories can be calculated offline or in real-time using an inverse plant model. The advantage of the last

one is that the reference trajectories can be changed online depending on the conditions imposed by environment rather than relaying on the previous calculated reference trajectories. This is essential for obstacle avoidance, which requires that the reference trajectories are adjusted accordingly. Inclusion of more degrees of freedom in a walking robot model normally leads to more precise results, but also it leads to more efforts needed to understand the process and what is happening with internal variables in the robotic structure.

A very important control concept in the field of bipedal robotics is the concept of Zero Moment Point (ZMP). This is a point at the foot/ground contact area where, for a dynamically stable walk, the sum of all moments about any axis laying in the tangential plane of the contact equals zero, including moments due to external forces and due to inertial forces acting on the walking robot architecture. The concept was introduced by Vukobratovic (Vukobratovic & Juricic, 1969). Although the concept has been in application for bipedal robots control tasks for more than 30 years, another concept related to the same problem has been introduced by Goswami (Goswami, 1999), so called, Foot Rotation Indicator (FRI), with intention of ZMP concept generalization and extension of its applicability. Foot Rotation Indicator is a point where the total ground reaction should act such that the supporting foot stays parallel to the ground, without any rotation. A bipedal robot loses stability if FRI point leaves the foot ground contact area and, similarly, if ZMP reaches the foot edge, the robot walk is marginally stable or unstable.

The models of humanoid robots can be organised into two categories; the simplified models (models with concentrated mass) or the complete models (models with distributed mass). A brief review of these models is given next.

#### **Models with Concentrated Mass.**

The group of relatively simple walking models is based on the inverted pendulum structure with variations combining a spring or two springs and dampers. The simple models include one or two variables with overall mass concentrated

into a point, a Centre Of Mass (COM). Six Determinants model (Kuo, 2007) has been used in the past to analyze the human gait cycle. Due to inconsistency with theoretical and experimental results, this model has been replaced by Inverted Pendulum model, which gives results much closer to those obtained experimentally. One of the major inconsistencies of the regular Inverted Pendulum model results with the experimental results is the zero energy input, which means that the model does not consider the energy lost during the gait cycle. A more advanced, but still one of the simplest models for human gait analysis is Mass and Spring Inverted Pendulum (MSIP) model (Geyer et al., 2005). The point mass is equal to the total mass of the body concentrated into the Centre Of Gravity (COG). The spring connects the ground Contact Point (CP) and the COG, and its deflections include all changes of the distance between the CP and the COG points due to flexions/extensions of the hip, knee, and ankle joints.

#### **Models with Distributed Mass and Multiple Degrees of Freedom.**

Complex bipedal walking models and their practical implementation bipedal robots are generally based on the human body anatomy (Shirata et al., 2004), (Yamaguchi & Takanishi, 1997). However, all those models and implementations include fewer DOF of motion than the DOF existing in the human body.

Another example is the methods proposed by (Albert & Gerth, 2003), the Two Masses Inverted Pendulum Mode (TMIPM) and the The Multiple Masses Inverted Pendulum Mode (MMIPM), which consist of simulating the performance of the biped walking motion by adding pendular motions on each biped link.

### **1.3 Objectives**

This work deals with the different dynamic models for humanoid robots. The simplified models for humanoid robots present some advantages, for instance,

the study and implementation can be easy to aim. On the other hand, the complete models for humanoids are complex to implement.

The principle objectives of this work are presented next:

- To study the main simplified models for humanoid robots and their motion equations.
- To develop the studied simplified models in simulation. They will be implemented using Matlab and two of its tools, Simulink and SimMechanics. For this purpose, a file which is based on the motion equations for every model will be created using Simulink. Furthermore, the mechanical model in simulation will be implemented for every model using SimMechanics.
- To compare the behaviour of experimental platforms with simplified models. The behaviour of some models in simulation will be compared with the obtained results in real platforms. To achieve this goal, the ankle prototype and the lower part body of the humanoid robot TEO will be used as test platforms.
- On the other hand, the creation of complete models for humanoid robots will be achieved. For this purpose, two robotics platforms will be used, the humanoid robot TEO and the miniature humanoid robot HOAP-3. The virtual robots will be modelled using a simulation platform. The selected platform is OpenHRP3, that is a simulation platform focused on humanoid robotics.
- Once the complete models are modelled and simulated in the OpenHRP3 platform, some tests will be accomplished using the experimental platforms. The simulation results and the real ones will be compared, and this will allow validating the complete models of humanoid robots created with OpenHRP3.



## 1.4 Structure of the document

This work is arranged into seven chapters as follows:

- In order to know the simplified models for humanoid robots used in this work, a brief explanation is presented in Chapter 2.
- The presented models in the previous chapter are implemented using Matlab as the software for simulation. Chapter 3 shows how the models have been created and explains the required files for simulation. In addition, the simulation results of two models are presented in this chapter.
- After the models have been implemented, they need to be tested in simulation. Chapter 4 shows the simulation results in the single inverted pendulum model. Moreover, the experimental results in the ankle prototype are presented in order to compare the behaviour of this platform with the single inverted pendulum model.
- Chapter 5 presents the humanoid robot TEO, which is used as the experimental platform. A test is performed using the lower part body of the robot. In order to know if the platform can behave as the double inverted pendulum model, the test results are compared with the results obtained in simulation using this simplified model.
- In order to develop complete models, the simulation platform OpenHRP3 has been used. Chapter 6 presents the created models in simulation for the humanoid robots TEO and HOAP-3. These robots are used as experimental platforms, and the simulation and experimental results are shown here.
- Finally, conclusions and future works are discussed in Chapter 7.



# Simplified Models of Humanoid Robots

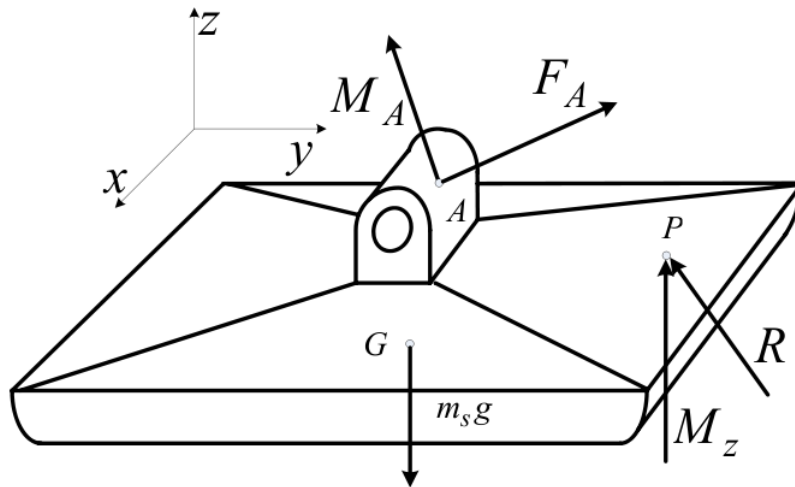
## 2.1 Introduction

This chapter presents a brief study of the main simplified models of humanoid robots. First of all, the concepts required to understand correctly the models are explained. In the following sections, the models and their motion equations are presented.

## 2.2 ZMP and COG concepts

Firstly, it is necessary to consider theoretical aspects of humanoid locomotion, the ZMP concept and COG. Several models are based on the measurement of the ZMP, which is the point with respect to which the dynamic reaction force at the contact of the foot with the ground does not produce any moment.

So as not to consider the complex dynamics of the entire upper body of the mechanism we can replace its influence by force  $F_A$  and moment  $M_A$  applied to the point  $A$  where the foot is connected with the leg. Also, the weight of the

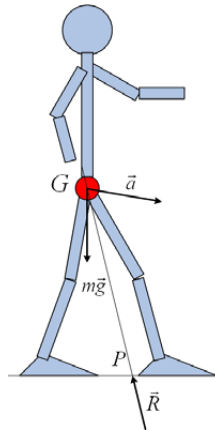


**Figure 2.1:** Forces acting on the foot of the bipedal mechanism

foot acts at its gravity centre (point  $G$ ). Finally, the contact of the foot with the ground produces its reaction force  $R$  at point  $P$  (Figure 2.1).

After the ZMP is clarified for the part below the ankle (foot) of the biped mechanism, it is necessary to consider how it can be interpreted for the entire robot. In addition to the forces which appeared at the contact point of the biped mechanism with the ground (reaction force and the friction), there are two other principal forces acting on the entire robot's body to take into account. They are gravity and inertial forces.

The human body can be considered as a chain of rigid links (hands, feet, body, etc.), joined by articulations with relative movements between them. Each link is subject to gravity and inertial forces. The best simplification that we can make is to consider that these forces act on the unique point - the COG.



**Figure 2.2:** Forces acting on a humanoid (sagittal plane)

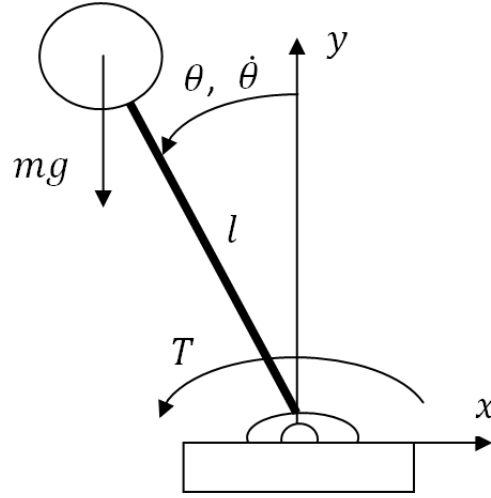
In Figure 2.2, point  $G$  is the Centre of Gravity of the humanoid,  $P$  is the point where the ground reaction force  $\vec{R}$  is applied,  $m\vec{g}$  is the gravity force and,  $\vec{a}$  is the inertial acceleration of the COG. It is important to mention that inertial forces are provoked by acceleration of the links and entire COG. Its analytical expression is:  $F_{inertia} = m\vec{a}$ .

As it has been discussed during the previous chapter, the mass concentrated models can be used to model the dynamics of a robot. So in this chapter, the different used models in this work will be explained.

## 2.3 Single inverted pendulum model

In a very simplified way, the dynamic model of the humanoid robot TEO can be considered similar to the inverted pendulum model, shown in Figure 2.3.

The similarity is established under the following assumptions. The mass of the humanoid ( $m$ ) is concentrated at its COG (tip of the pendulum), which is at a distance  $l$  from the floor. The mass of the rigid link is then considered negligible. Besides, the action (torque  $T$ ) that allows the mass  $m$  to move a specific angle  $\theta$  at a speed  $\dot{\theta}$  (movement of the COG during the walking action) is effected by a servomotor (ankle of the humanoid robot) fixed at the end of the link (floor).



**Figure 2.3:** Single inverted pendulum model

This servomotor performs the control action to ensure the stability of the system during the walking action

To write the equation of motion of the pendulum (Khalil & Grizzle, 1992), let us identify the forces acting on the tip. There is a downward gravitational force equal to  $m\vec{g}$ , where  $\vec{g}$  is the acceleration due to gravity. There is also a frictional force resisting the motion, which is assumed to be proportional to the speed of the tip with a coefficient of friction  $k$ . Using Newton's second law of motion, the equation of motion in the tangential direction can be written as

$$ml\ddot{\theta} = -m\vec{g}\sin\theta - kl\dot{\theta}. \quad (2.1)$$

Writing the equation of motion in this direction has the advantage that the link tension, which is in the normal direction, does not appear in the equation. To obtain a state model for the pendulum, let take the state variables as  $x_1 = \theta$  and  $x_2 = \dot{\theta}$ . Then, the state equations are

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -\frac{g}{l}\sin x_1 - \frac{k}{m}x_2. \end{aligned} \quad (2.2)$$

From the physical description of the pendulum, it is clear that it has only two equilibrium positions corresponding to the equilibrium points  $(0, 0)$  and  $(\pi, 0)$ . Physically, we can see that these two positions are quite distinct from each other. While the pendulum can indeed rest at the  $(0, 0)$  equilibrium point, it can hardly maintain at the  $(\pi, 0)$  point because infinitesimally small disturbance from that equilibrium will take the pendulum away. The difference between the two equilibrium points is in their stability properties.

Another version of the pendulum equations arises if we can apply a torque  $T$  to it. This torque is viewed in our case as a control input in the equation

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -\frac{g}{l} \sin x_1 - \frac{k}{m} x_2 + \frac{1}{ml^2} T. \end{aligned} \quad (2.3)$$

The model matching technique (Isidori, 1997) is used, based on the input-output linearization of the system. The equations obtained from the application of this technique are the ones presented next:

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} x_2 \\ -\frac{g}{l} \sin x_1 - \frac{k}{m} x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix} u, \\ y &= x_1. \end{aligned} \quad (2.4)$$

Therefore, the direct relation between the input and output of the system is given by

$$\ddot{y} = -\frac{g}{l} \sin x_1 - \frac{k}{m} x_2 + \frac{1}{ml^2} u, \quad (2.5)$$

with  $u = T$  and  $y = \theta$ . To simplify the use of the equation (2.5), constants have been defined as follows:

$$p_1 = -\frac{g}{l} \quad (2.6)$$

$$p_2 = -\frac{k}{m} \quad (2.7)$$

$$r_i = \frac{1}{ml^2} \quad (2.8)$$

Then, the result of (2.5) using these constants is:

$$\ddot{y} = p_1 \sin x_1 + p_2 x_2 + r_i u. \quad (2.9)$$

The purpose is to obtain the control law  $u$  so that  $y$  follows  $\theta_{ref}$  as  $\theta_r$  follows  $\theta_{ref}$ , that is, a control law that the whole dynamics matches the model  $M(s) = \frac{\theta_r}{\theta_{ref}}$  obtained previously by experimental identification. In order to do so, we define  $u$  as

$$u = ml^2 \left[ \left( \frac{g}{l} \sin x_1 + \frac{k}{m} x_2 \right) + v \right], \quad (2.10)$$

so that  $\ddot{y} = v$  (from (2.5)). Choosing

$$v = \dot{\theta}_r + a(\theta_r - y), \quad (2.11)$$

it is obtained that

$$\ddot{y} = \dot{\theta}_r + a(\theta_r - y). \quad (2.12)$$

The value of  $a$  is selected by trial-error process in order to obtain the minimum tracking error.

## 2.4 Double inverted pendulum model

Another dynamic model used in this work is the double inverted pendulum (Kaynov, 2008). It is used to implement simultaneously both the ZMP and the posture control acting on the ankle and hip joints to maintain the stability of a humanoid robot. A double pendulum consists of one pendulum attached to another. The mechanism is divided into two principal parts, the upper body and the bottom part with its own COM. The motion of the upper part controls the body posture and the motion of the bottom part of the pendulum controls the ZMP of the humanoid robot. A double inverted pendulum system is described in Figure 2.4.

Consider a double bob pendulum with masses  $m_1$  and  $m_2$ , the total mass of the humanoid robot ( $M$ ) is the sum of both masses. They are attached by rigid



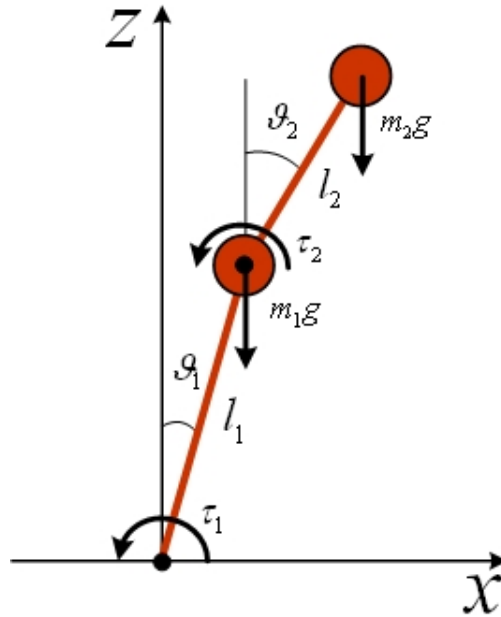


Figure 2.4: Double inverted pendulum model

massless wires of lengths  $l_1$  and  $l_2$ . Also, the angles the two wires make with the vertical be denoted  $\theta_1$  the ankle rotation and  $\theta_2$  the hip rotation.

To obtain the state representation of the double inverted pendulum system it is necessary to define state variables  $x_1, x_2, x_3$  and  $x_4$  as

$$x_1 = \theta_1 \quad (2.13)$$

$$x_2 = \dot{\theta}_1 \quad (2.14)$$

$$x_3 = \theta_2 \quad (2.15)$$

$$x_4 = \dot{\theta}_2 \quad (2.16)$$

The outputs of the system are  $\theta_1$  (rotation of the pendulum about the ankle) and  $\theta_2$  (rotation at the hip).

$$y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_3 \end{bmatrix}. \quad (2.17)$$

Then the state space representation of the system is

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{g(m_1+m_2)}{m_1 l_1} & 0 & -\frac{m_2 g}{m_1 l_1} & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{g(m_1+m_2)}{m_1 l_2} & 0 & \frac{g(m_1+m_2)}{m_1 l_2} & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{1}{m_1 l_1^2} & -\frac{1}{m_1 l_1 l_2} \\ 0 & 0 \\ -\frac{1}{m_1 l_1 l_2} & \frac{(m_1+m_2)}{m_1 m_2 l_2^2} \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad (2.18)$$

where  $u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$  is a vector consisting of control torques for both joints (hip and ankle motor's torques).

To simplify the equation (2.18), different constants have been defined:

$$p_1 = -\frac{g(m_1 + m_2)}{m_1 l_1} \quad (2.19)$$

$$p_2 = -\frac{m_2 g}{m_1 l_1} \quad (2.20)$$

$$r_{i1} = \frac{1}{m_1 l_1^2} \quad (2.21)$$

$$r_{i2} = -\frac{1}{m_1 l_1 l_2} \quad (2.22)$$

$$p_3 = -\frac{g(m_1 + m_2)}{m_1 l_2} \quad (2.23)$$

$$p_4 = \frac{g(m_1 + m_2)}{m_1 l_2} \quad (2.24)$$

$$r_{i3} = -\frac{1}{m_1 l_1 l_2} \quad (2.25)$$

$$r_{i4} = \frac{(m_1 + m_2)}{m_1 m_2 l_2^2} \quad (2.26)$$

The state space representation of the system using the constants above is

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ p_1 & 0 & p_2 & 0 \\ 0 & 0 & 0 & 1 \\ p_3 & 0 & p_4 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ r_{i1} & r_{i2} \\ 0 & 0 \\ r_{i3} & r_{i4} \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \quad (2.27)$$

Then the output equation is

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}. \quad (2.28)$$

## 2.5 Cart-table model

This model is based on the ZMP preview control scheme that obtain the COG trajectory from a defined ZMP trajectory (Kajita et al., 2003). The relationship between the ZMP trajectory and the COG trajectory is defined by the following equations:

$$p_x = x - \frac{\ddot{x}}{g} z_c, \quad (2.29)$$

$$p_y = y - \frac{\ddot{y}}{g} z_c, \quad (2.30)$$

where, in the sagittal plane,  $p_x$  is the ZMP reference,  $x$  is the COG trajectory,  $\ddot{x}$  is the COG acceleration,  $z_c$  is the COG height, and  $g$  is the gravity (Figure 2.5).

For the frontal plane the procedure is the same but using the  $y$  component of these terms. In the cart-table model, the cart mass corresponds to the COM of the robot. If the cart accelerates with 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.

$$\tau_{ZMP} = mg(x - p_x) - m\ddot{x}z_c. \quad (2.31)$$

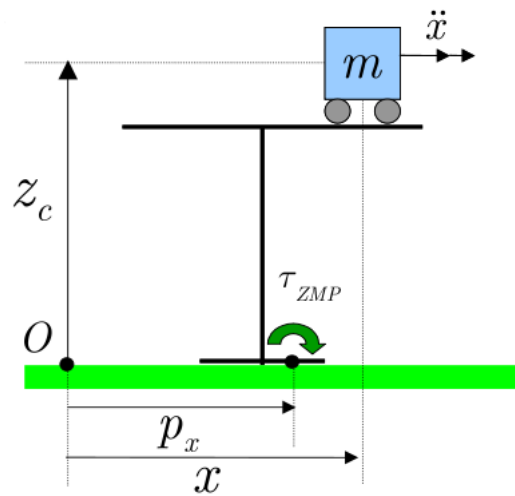


Figure 2.5: *Cart-table model*

## 2.6 Chapter summary

The main simplified models of humanoid robots have been presented in this chapter. Besides, some concepts have been explained in order to understand the models correctly. The presented models have been the single inverted pendulum, the double inverted pendulum, and the cart-table.

# Chapter 3

## Simulation of Simplified Models

### 3.1 Introduction

After the description of the models, this chapter details the procedure followed to analyse them. The models have been created using Matlab and Simulink as simulation platform. Moreover, the add-on of Simulink, SimMechanics, has been used to create the models.

### 3.2 Software for simulation

Simulating the dynamics of multibody systems is a common problem in engineering and science. Several programs are available for that task which are either symbolical computation programs to derive and solve the dynamic equations of motion, or numerical programs which compute the dynamics on the basis of a 3D-CAD model or by means of a more abstract representation, e.g. a block diagram.

### 3.2.1 Simulink

Simulink is an environment for multidomain simulation and Model-Based Design for dynamic and embedded systems. It provides an interactive graphical environment and a customizable set of block libraries that let you design, simulate, implement, and test a variety of time-varying systems, including communications, controls, signal processing, video processing, and image processing.

Simulating a dynamic system is a two-step process. First, the user creates a block diagram, using the Simulink model editor, that graphically depicts time-dependent mathematical relationships among the system's inputs, states, and outputs. The user then commands the Simulink software to simulate the system represented by the model from a specified start time to a specified stop time.

### 3.2.2 SimMechanics

SimMechanics is an add-on for the GUI-based simulation environment Simulink. Mechanical systems are represented by connected block diagrams. Unlike normal Simulink blocks, which represent mathematical operations, or operate on signals, Physical Modelling blocks represent physical components, and geometric and kinematic relationships directly. SimMechanics models, however, can be interfaced seamlessly with ordinary Simulink block diagrams. This enables the user to design e.g., the mechanical and the control system in one common environment. Various analysis modes and advanced visualization tools make the simulation of complex dynamical systems possible even for users with a limited background in mechanics.

## 3.3 Simplified models in simulation

The models have been modelled using the programs described previously. Each model has three files, one file for Simulink, one file for SimMechanics and another one m-file of Matlab. The last file includes the parameters of the model,

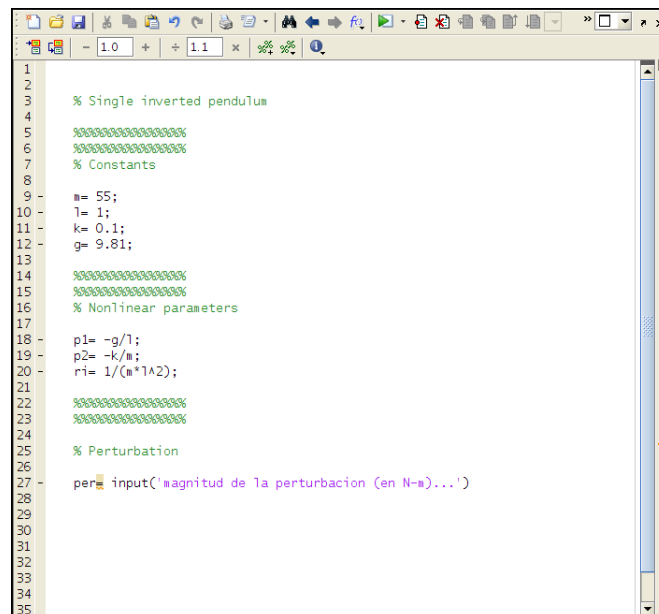
for instance, the weight and the length of the pendulum, and other information as the acceleration due to gravity or the coefficient of friction.

It is important to note that the simulation results of the explained models in Section 3.3.2 and Section 3.3.4 will be presented in this chapter.

On the other hand, the simulation results of the rest of the models will be presented in the next chapters, due to the fact that the behaviours of these simplified models will be compared with the behaviours of the experimental platforms.

### 3.3.1 Single inverted pendulum model

Using the description of the model in Section 2.3, the single inverted pendulum has been modelled with Simulink and SimMechanics.



```
1
2
3 % Single inverted pendulum
4
5 %~~~~~
6 %~~~~~
7 % Constants
8
9 m= 55;
10 l= 1;
11 k= 0.1;
12 g= 9.81;
13
14 %~~~~~
15 %~~~~~
16 % Nonlinear parameters
17
18 p1= -g/l;
19 p2= -k/m;
20 r1= 1/(m*l^2);
21
22 %~~~~~
23 %~~~~~
24
25 % Perturbation
26
27 per= input('magnitud de la perturbacion (en N-m)...')
```

Figure 3.1: Single inverted pendulum m-file

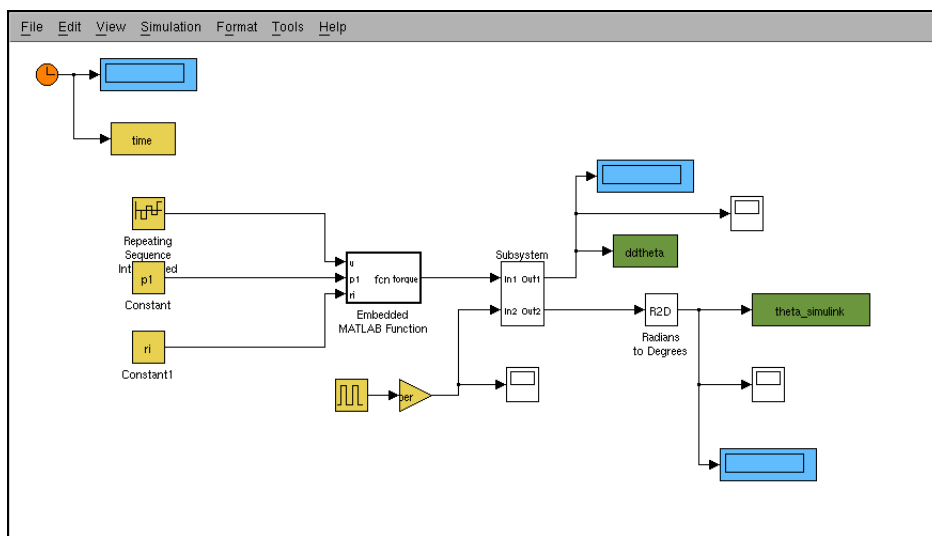
#### 3.3.1.1 M-File

This file is necessary to define the parameters and constants of the model. For instance, the mass and length of the tip of the pendulum. Furthermore, the

constants used and explained in Section 2.3 are in this file, see Figure 3.1. Finally, a perturbation of the system can be introduced by the user.

### 3.3.1.2 Simulink

The model created for simulation using Simulink is shown in Figure 3.2. The



**Figure 3.2:** Single inverted pendulum Simulink model

inputs of the model are on the left side, such as the sequence that will follow the pendulum, and the required constants. The embedded matlab function converts the values of the angles in the appropriate torque that moves the pendulum. Finally, the Subsystem implements the equations of the model, using as inputs the torque calculated before, and a perturbation that it can be selected by the user. Figure 3.3 shows the subsystem that implements the Equation (2.5). Its outputs are the acceleration and the position of the pendulum, as can be seen in Figure 3.3.



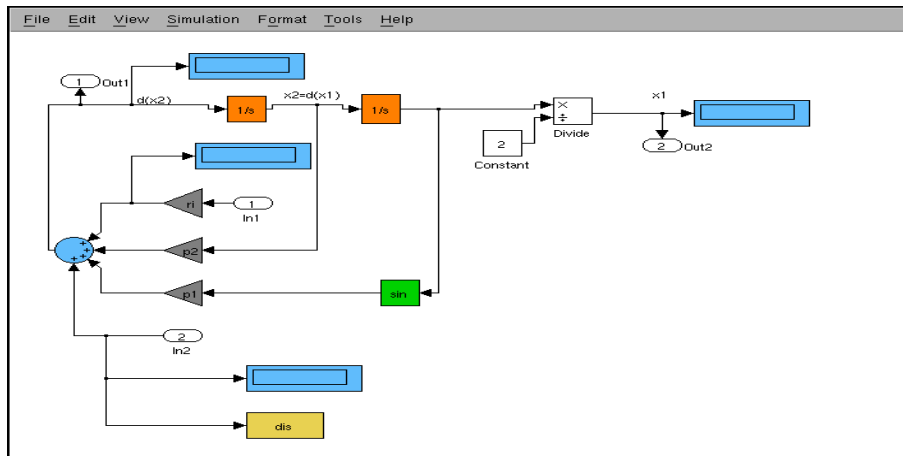


Figure 3.3: Subsystem

3.3.1.3 SimMechanics

This file represents the physical model of the single inverted pendulum (shown in Figure 3.4). The model is created using SimMechanics and is explained next.

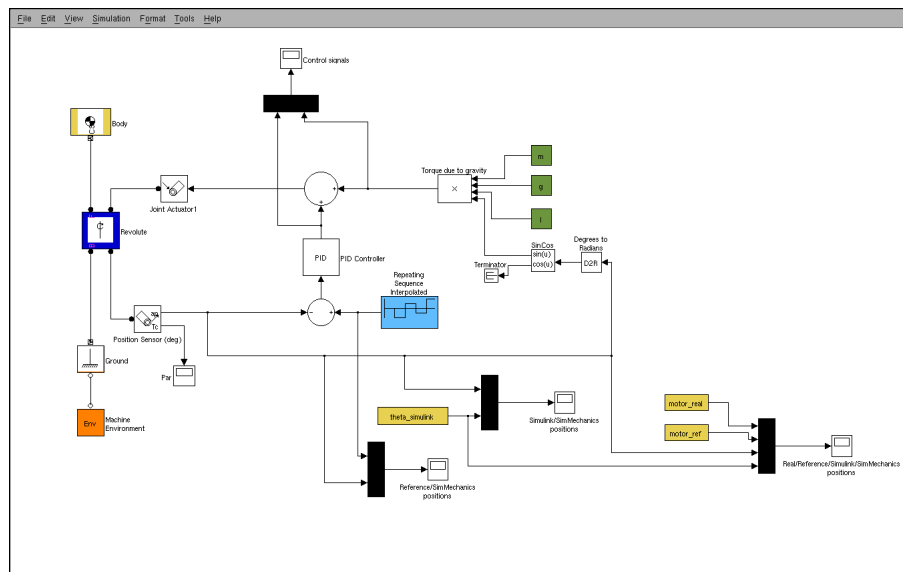
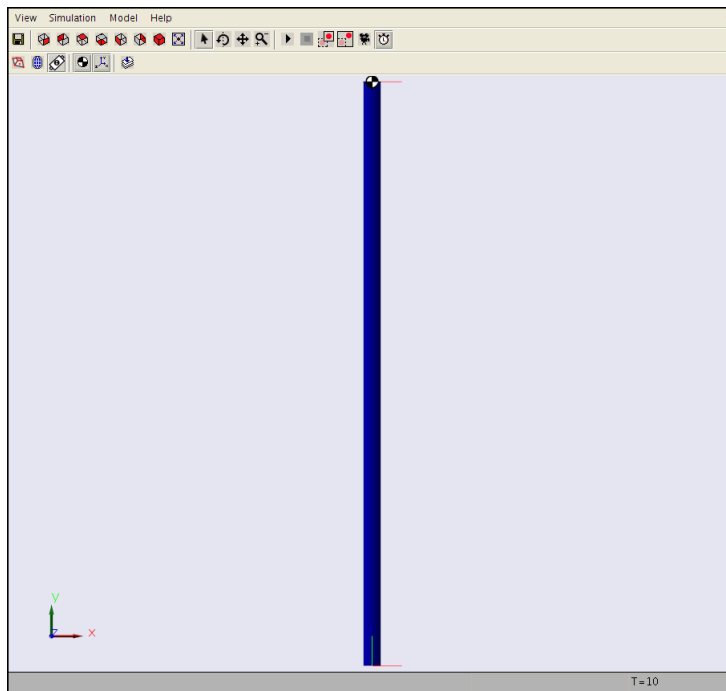


Figure 3.4: Single inverted pendulum SimMechanics model

The mechanical system composed of four blocks can be seen on the left side. Firstly, the machine environment block defines the mechanical simulation environment for the machine, the gravity, the analysis mode, the constraint solver type, visualization, etc.

Furthermore, the model has got a ground block that grounds one side of a joint to a fixed location in the world coordinate system. The revolute block is situated above and represents one rotational degree of freedom. It has got two ports, the follower (F) body rotates relative to the base (B) body about a single rotational axis. Base-follower sequence and axis direction determine sign of forward motion by the right-hand rule. Besides, sensor and actuator ports can be added to this block.



**Figure 3.5:** *Single inverted pendulum SimMechanics mechanical model*

Finally, the body block is used to represent a rigid body. The body is defined by mass, inertia tensor, and coordinate origins and axes for Centre of Gravity.

Moreover, the body initial position and orientation can be set, apart from the optional settings for customized the body geometry and colour. Figure 3.5 shows the machine modelled using the explained blocks.

As the Simulink model, this model is controlled through a torque control. The input torque is composed of the torque due to gravity and the equilibrium torque, which is calculated for every position given by the repeating sequence interpolated block. This signal is used as the input of the joint actuator that moves the pendulum.

The angle and the computed torque of the system can be measured using the position sensor, which is connected to the sensor port of the revolute block. The position is compared with the reference position, the obtained position of the Simulink model, and the given reference.

### **3.3.2 Control of the single inverted pendulum**

As the model explained in Section 3.3.1, this is a single inverted pendulum but the difference is that this file includes a control loop.

#### **3.3.2.1 M-File**

The main m-file (Figure 3.6) of this model is the same as the used file in Section 3.3.1. The user is asked to set the value of the angle rotation and the perturbation of the system.

Besides, this model runs another file that contents a motor model. This file is shown in Figure 3.7.

```

1 % Single Inverted Pendulum
2
3 %~~~~~
4 % Constants
5
6
7
8 m= 50;
9 l= 1;
10 k= 0.1;
11 g= 9.81;
12 %~~~~~
13 % Nonlinear parameters
14
15
16 p1= -g/l;
17 p2= -k/m;
18 r1= 1/(m*l^2);
19
20 %~~~~~
21 % Data input
22
23
24
25 theta= input('posicion de equilibrio (en grados)...');
26 theta_e= theta*(pi/180);
27
28 %~~~~~
29 % Perturbation
30
31
32
33
34
35
36
37
38 u_e= (-p1/r1)*sin(theta_e);
39
40 %~~~~~
41 %~~~~~
42
43 run motor_model
44
45
46

```

Figure 3.6: Single inverted pendulum controlled m-file

```

1 %
2 % Motor model
3 %
4 TO= 0.0144; %Sampling period
5 gmz= tf([0.6128 0.3194],[1 0.2315 -0.299],0.0144);
6 gms= d2c(gmz);
7 [ngm,dgm]= tfdata(gms,'v');
8 %
9 % Controllable realization
10 %
11 [Am,bm,cm,dm]= tf2ss(ngm,dgm);
12 Mss= ss(Am,bm,cm,dm);
13
14 Am2=Am'; bm2=cm'; cm2=bm'; dm2=dm;
15 Mss2= ss(Am2,bm2,cm2,dm2);
16 [num_m,den_m]=ss2tf(Am2,bm2,cm2,dm2);
17 gmz2= tf(num_m,den_m);
18
19 c3x3= [1 0 0; 0 1 0; 0 0 1];
20 d3= [0; 0; 0];
21
22
23
24 Am3=Am2; bm3=bm2; cm3=c3x3;
25 Mss3= ss(Am3,bm3,cm3,d3);
26
27
28
29
30
31
32
33
34
35
36

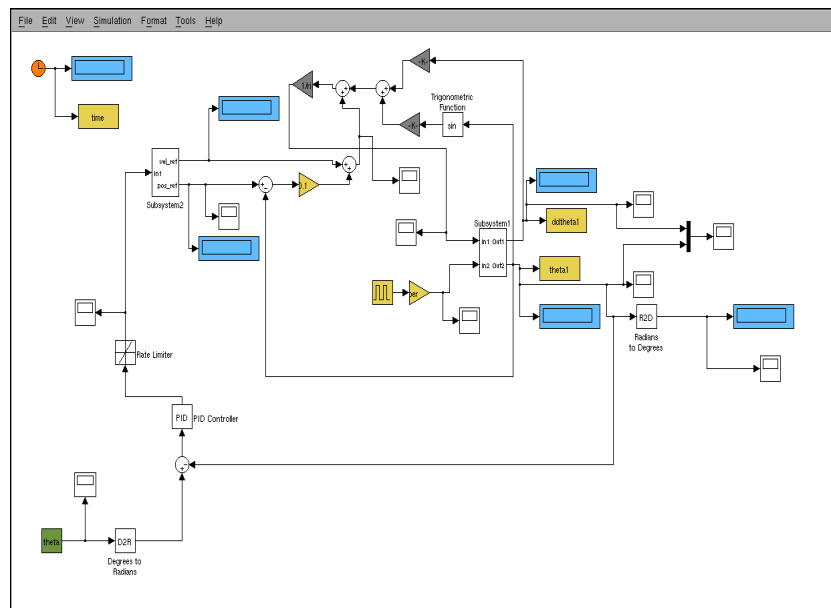
```

Figure 3.7: Motor model file

### 3.3.2.2 Simulink

The simulink file has been created using the equations of the single inverted pendulum and the model matching technique to obtain a linear system as has been explained in Section 2.3.

The model is shown in Figure 3.8 and the single inverted pendulum model created in Section 3.3.1.2 can be seen on the right side. The blocks that represent the Equation (2.10) are located below, and the calculated torque is the input of the pendulum. On the left side, there is a control loop that tries to minimize the error between the reference and the real position of the pendulum. The obtained signal is using to calculate the position and velocity references, this is calculated into the subsystem shown in Figure 3.9. The subsystem includes a state-space block and its parameters are the motor model parameters, which have been calculated previously in the motor model file. The outputs of the model are the angle rotation and the acceleration of the pendulum.



**Figure 3.8:** Single inverted pendulum controlled Simulink model

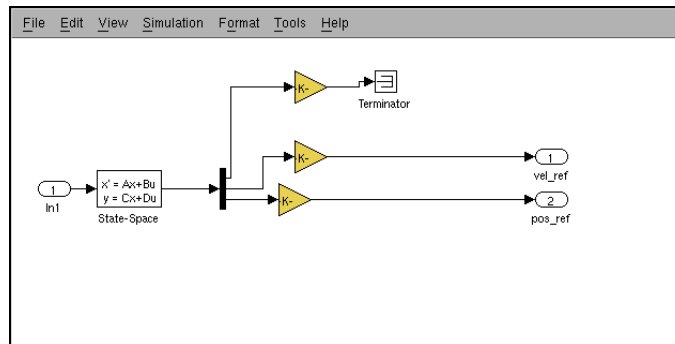


Figure 3.9: Subsystem

### 3.3.2.3 Simulation results

This model has been simulated in order to validate its correct implementation using Simulink.

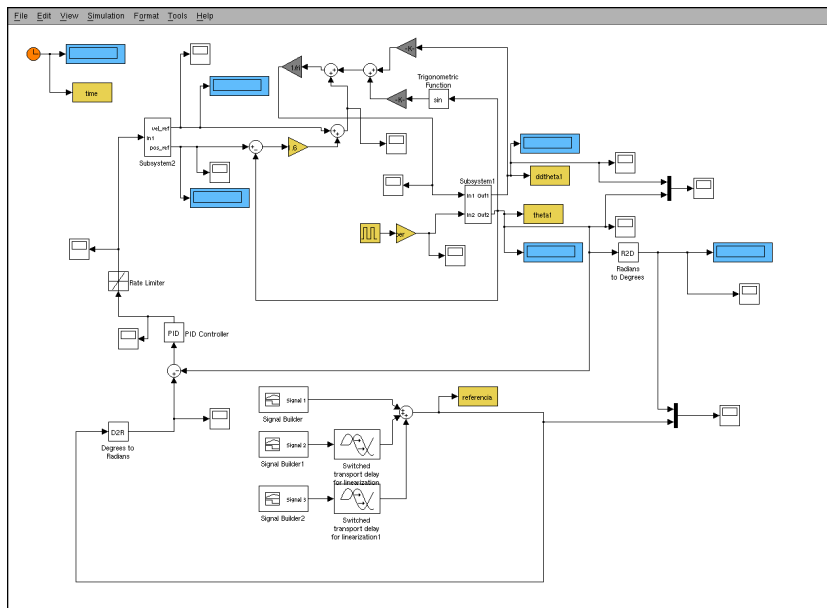
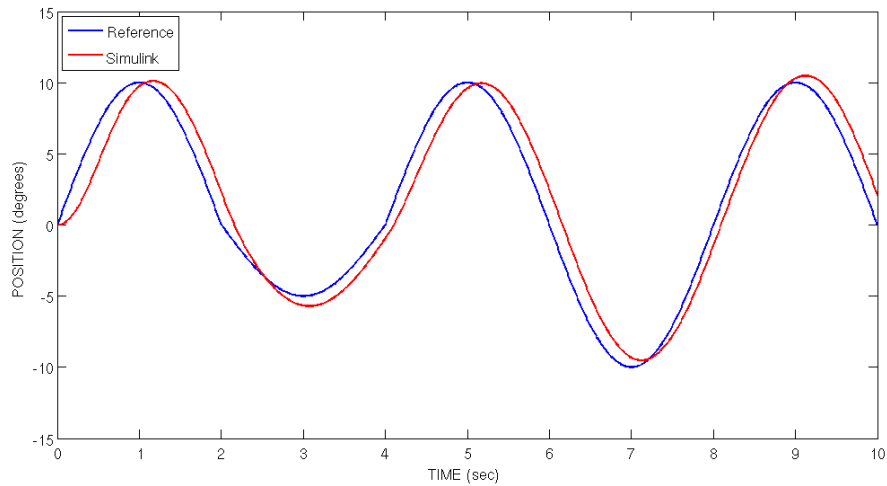


Figure 3.10: Single inverted pendulum controlled Simulink model for simulation

The given trajectory for simulation has been created using signal builder

blocks, as can be seen in the lower part of Figure 3.10. The reference takes different position values between  $10deg$  and  $-10deg$ , and the simulation time is  $10sec$ . Moreover, the parameters of the system are  $m = 10kg$ ,  $l = 1m$ ,  $g = 9.8m/sec^2$ , and  $a = 0.8$ .

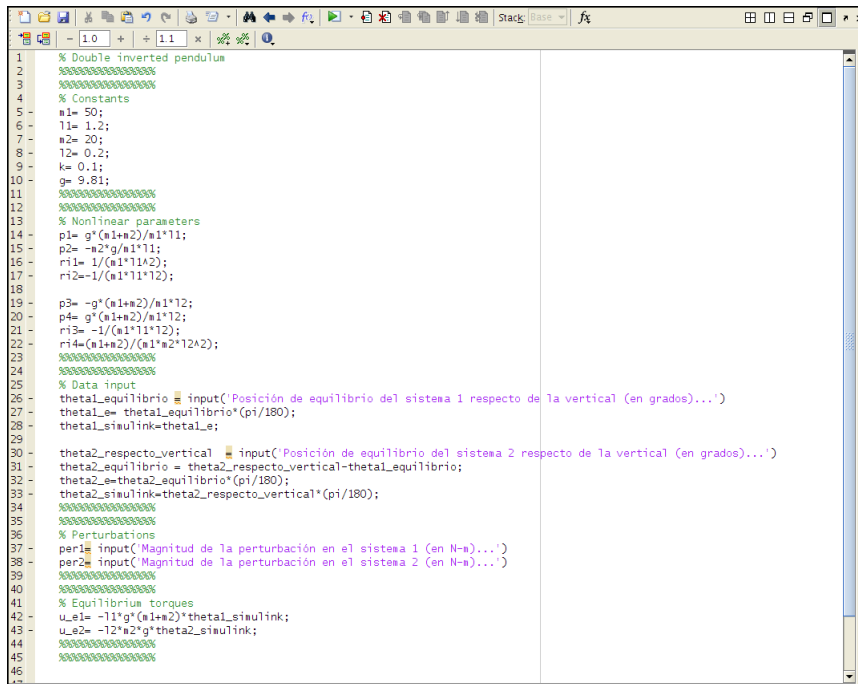


**Figure 3.11:** *Simulation results in Simulink*

After the simulation test, the results in Simulink are presented in Figure 3.11. The position reference and the obtained positions in simulation can be seen in the figure. The simulation model has followed the given reference, although its response has not been entirely good. It can be noticed that the model could not track correctly the trajectory, and a delay exists between the given trajectory and the obtained values. However, this test has been presented as a first approach, and the control loop could be adjusted in order to achieve a correct performance of the system.

### 3.3.3 Double inverted pendulum model

According to the description of the model in Section 2.4, the double inverted pendulum has been modelled with Simulink and SimMechanics.



```

1 % Double inverted pendulum
2 %~~~~~
3 % Constants
4 %~~~~~
5 m1= 50;
6 l1= 1.2;
7 m2= 20;
8 l2= 0.2;
9 k= 0.1;
10 g= 9.81;
11 %~~~~~
12 % Nonlinear parameters
13 %~~~~~
14 p1= g*(m1+m2)/m1*l1;
15 p2= -m2*g/m1*l1;
16 r1= 1/(m1*l1*l2);
17 r12=-1/(m1*l1*l2);
18
19 p3= -g*(m1+m2)/m1*l2;
20 p4= g*(m1+m2)/m1*l2;
21 r13= -1/(m1*l1*l2);
22 r14=(m1+m2)/(m1*m2*l2*l2);
23 %~~~~~
24 % Data input
25 %~~~~~
26 theta1_equilibrio = input('Posición de equilibrio del sistema 1 respecto de la vertical (en grados)...')
27 theta1_e= theta1_equilibrio*(pi/180);
28 theta1_simulink=theta1_e;
29
30 theta2_respecto_vertical = input('Posición de equilibrio del sistema 2 respecto de la vertical (en grados)...')
31 theta2_equilibrio = theta2_respecto_vertical-theta1_equilibrio;
32 theta2_e=theta2_equilibrio*(pi/180);
33 theta2_simulink=theta2_respecto_vertical*(pi/180);
34 %~~~~~
35 % Perturbations
36 %~~~~~
37 par1= input('Magnitud de la perturbación en el sistema 1 (en N-m)...')
38 par2= input('Magnitud de la perturbación en el sistema 2 (en N-m)...')
39 %~~~~~
40 % Equilibrium torques
41 %~~~~~
42 u_e1= -l1*g*(m1+m2)*theta1_simulink;
43 u_e2= -l2*m2*g*theta2_simulink;
44 %~~~~~
45 %~~~~~
46

```

Figure 3.12: Double inverted pendulum m-file

### 3.3.3.1 M-File

This file is necessary to define the parameters and constants of the model. For instance, the masses and lengths of the double pendulum. Moreover, the constants defined in Section 2.4 are in this file, see Figure 3.12. The user is asked to introduce the angles of rotation respect to the vertical for both pendulums. These values are used to calculate the required equilibrium torques that allow the system to reach to the position given by the user. Finally, a perturbation for each system can be introduced by the user.

### 3.3.3.2 Simulink

The created model with Simulink is shown in Figure 3.13. The inputs of the model are on the left side. They are the equilibrium torque for each pendulum calculated before in the m-file, and the perturbations.



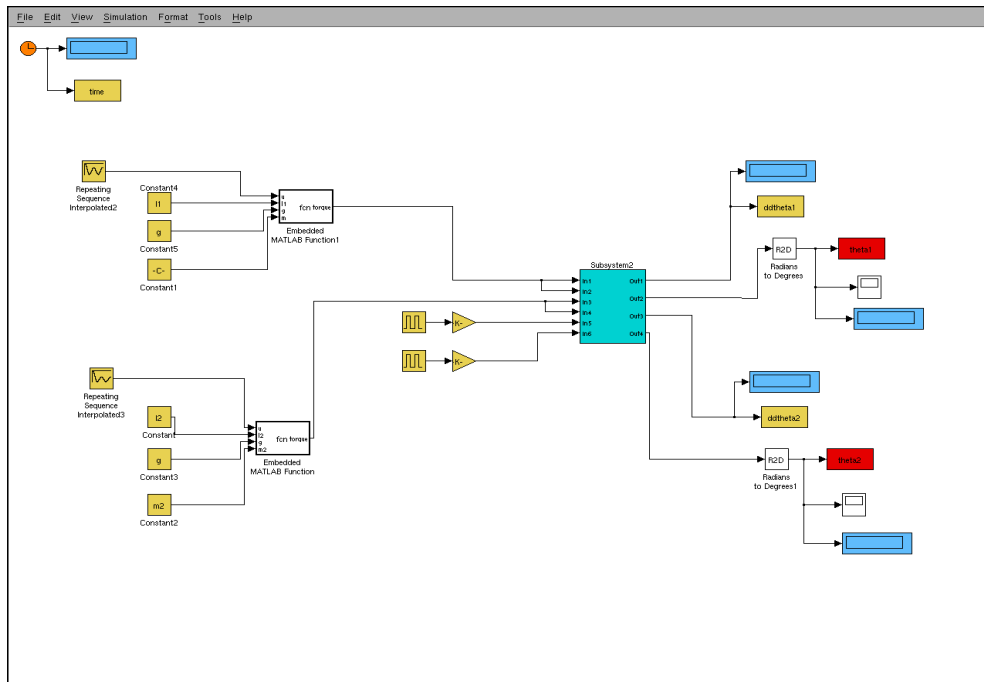


Figure 3.13: Double inverted pendulum Simulink model

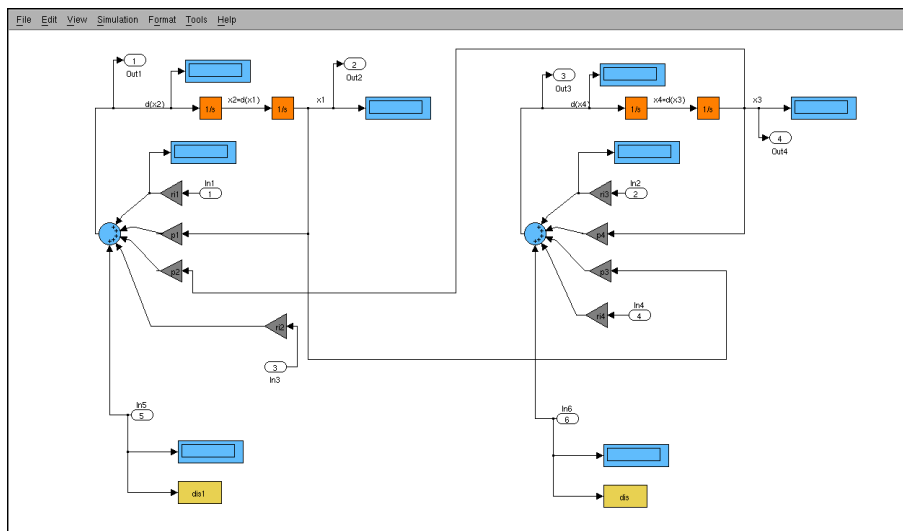


Figure 3.14: Subsystem

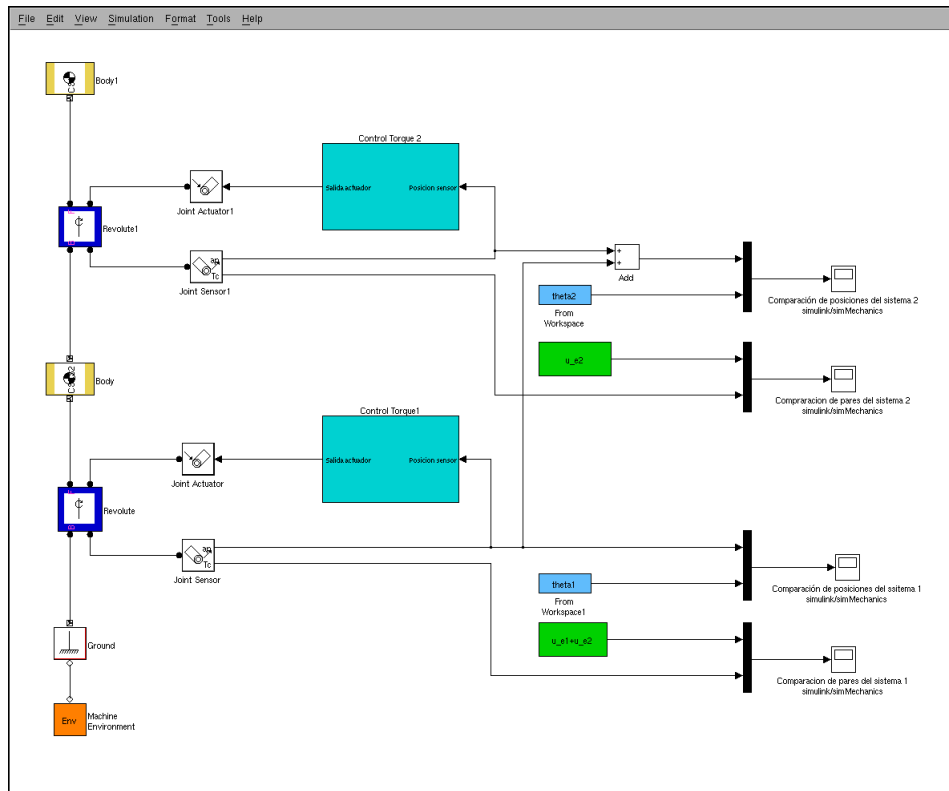


Figure 3.15: Double inverted pendulum SimMechanics model

The subsystem shown in Figure 3.14 has got the necessary blocks that implement correctly the Equations (2.18) explained in Section 2.4.

The first pendulum, which represents the lower part body, can be seen on the left side. Its outputs are the angle and acceleration of the ankle.

On the right side, the upper part body can be seen. Its outputs are the angle and acceleration of the hip, and they are calculated in this subsystem. Then, they are represented graphically on the right side of the main system.

### 3.3.3.3 SimMechanics

The SimMechanics file for this model is very similar to the created file for the previous model. The file contains one pendulum attached to another, as can be

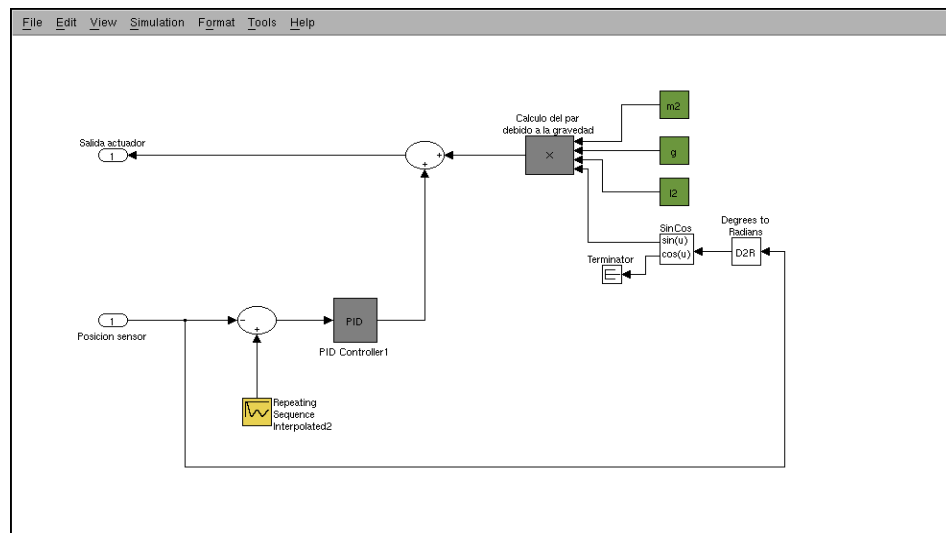


Figure 3.16: Subsystem

seen in Figure 3.15. These pendulums are located on the left side, and each one has got its own actuator and sensor.

The double inverted pendulum is controlled through a torque control, as well as the single inverted pendulum. In each pendulum, the input torque is composed of the torque due to gravity and the calculated equilibrium torque. These signals are used as the input of each joint actuator and have been calculated in the two subsystems located in the middle of the model. One of these subsystems is shown in Figure 3.16.

As the model for the single pendulum, the position sensors connected to the sensor ports in both revolute joints allow measuring the angle and the computed torque for every system. The angle is compared with the obtained angle of the Simulink model, and the torque is compared with the equilibrium torque used in both systems (Simulink and SimMechanics). Figure 3.17 shows the mechanical model of the double inverted pendulum. The green pendulum is the lower part body and the red one represents the upper part body.

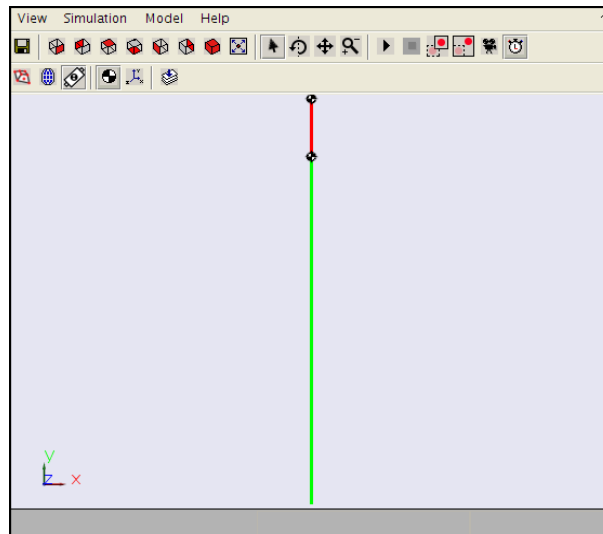


Figure 3.17: Double inverted pendulum SimMechanics mechanical model

### 3.3.4 Cart-table model

According to the description of the model in Section 2.5, the cart-table model

```

1
2 % Cart-table
3
4 %~~~~~
5 %~~~~~
6
7 % Physical constants
8
9 m= 55;
10 Zc= 1;
11 g= 9.81;
12
13
14 %~~~~~
15 %~~~~~
16
17 % Data input
18
19
20 acc_x= input('Aceleración (m/s²) en eje x...');
21 acc_y= input('Aceleración (m/s²) en eje y...');
22
23 T=input('Sampling time...');
24
25
26
27
28

```

Figure 3.18: Cart-table m-file

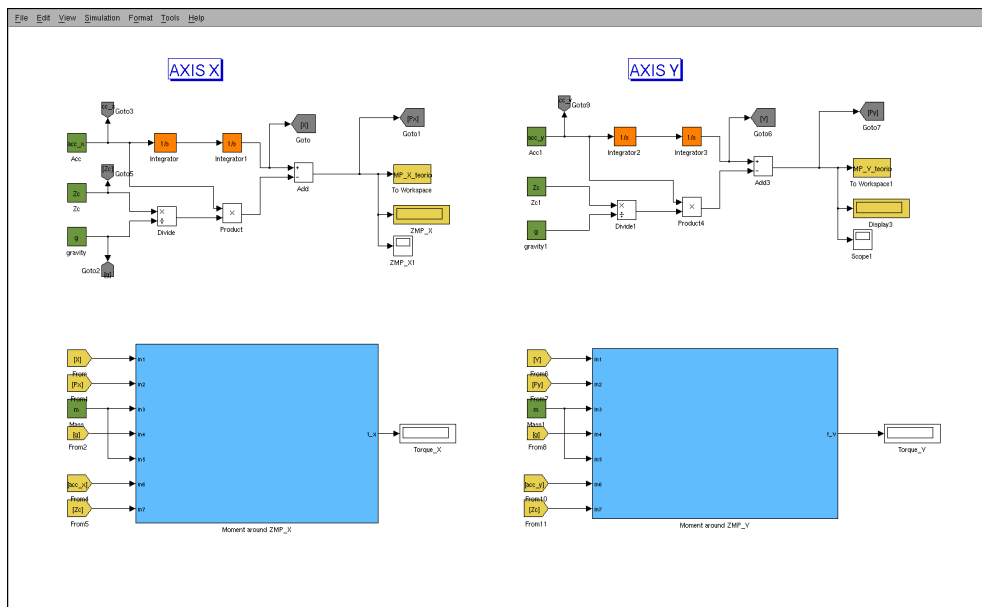


Figure 3.19: Cart-table Simulink model

has been modelled with Simulink and SimMechanics.

### 3.3.4.1 M-File

This file is necessary to define the constants of the model. For instance, the cart mass, the COG height and the gravity (Figure 3.18). Moreover, the user is asked to introduce the acceleration in the frontal and sagittal plane.

### 3.3.4.2 Simulink

The model shown in Figure 3.19 calculates the trajectory of the COG en both planes, around axis x (sagittal plane) and axis y (frontal plane). For this purpose, different blocks have been used to implement the Equations (2.29) and (2.30).

The inputs of both systems are the COG height, the mass of the cart and the respectively acceleration. The outputs are the ZMP reference in both axes, which are used to calculate the moments around these points. The moments

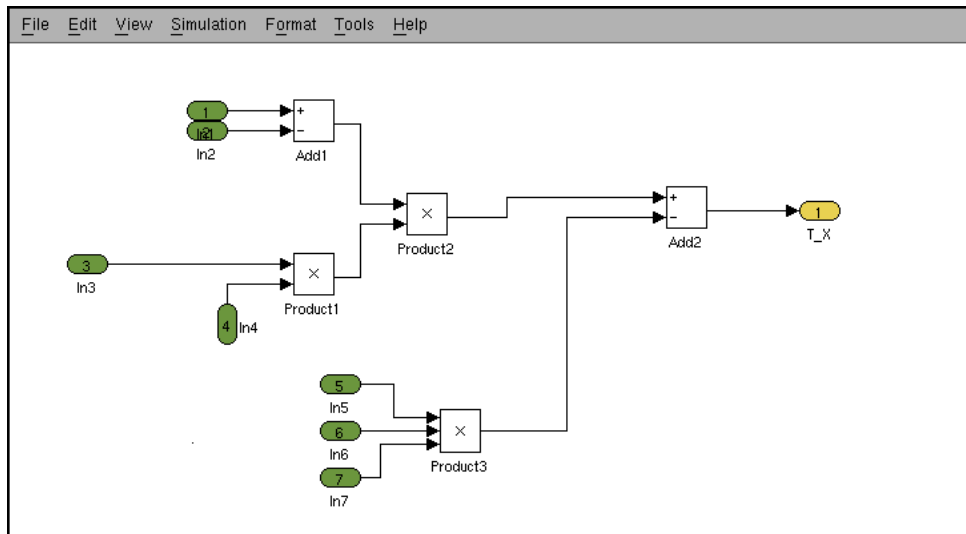


Figure 3.20: Subsystem

are calculated into two subsystems, which are located below the main model explained previously, using the Equation (2.31). One of these subsystems can be seen in Figure 3.20. Whether the values of the calculated moments are equal to zero, then the ZMP exists.

### 3.3.4.3 SimMechanics

Figure 3.21 shows the cart-table model in SimMechanics. To develop this model, new blocks have been used as the weld block that represents a joint with no degrees of freedom. These blocks have been used to create the lower part of the table, due to it is a rigid body with no relative motion respect to the ground.

As the cart motion is translational on the upper part of the table, it has been used a new block called in-plane. This block represents two translational degrees of freedom in two primitive prismatic axes. Sensor and actuator ports can be added to these blocks.

The SimMechanics model, as well as the Simulink model, is actuated using

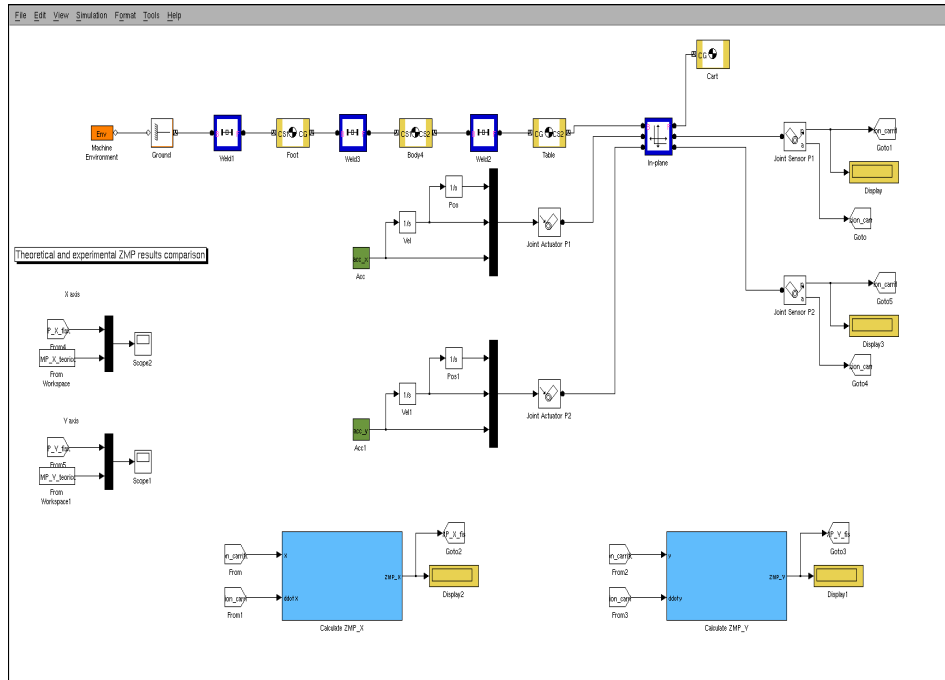


Figure 3.21: Cart-table SimMechanics model

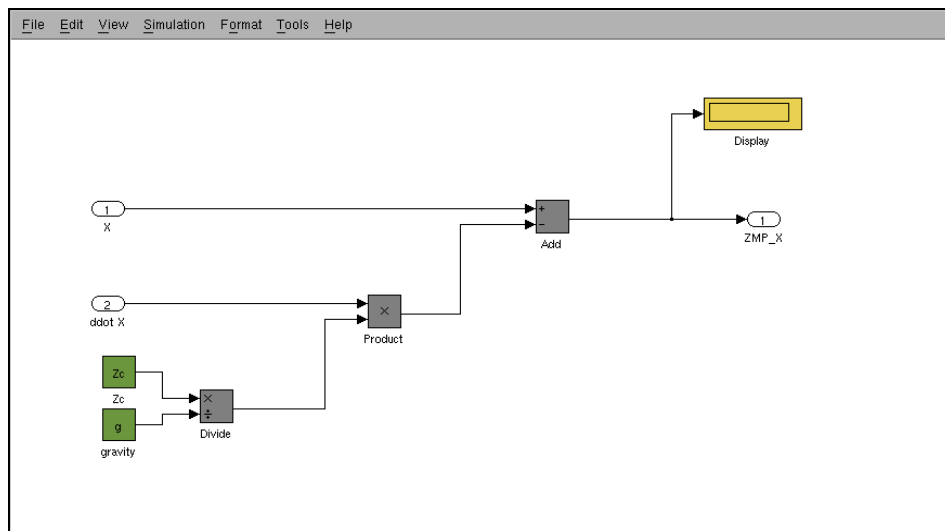
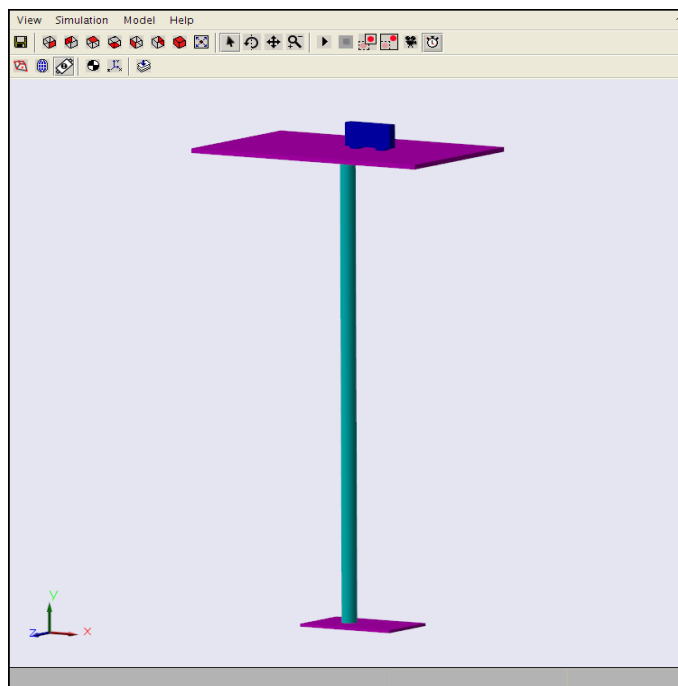


Figure 3.22: Subsystem

the acceleration that it has been obtained of the m-file. Apart from the acceleration, the position and velocity are required, and they have to be introduced as one signal in the actuator block. Two joint actuators are used to actuate the two different degrees of freedom of the cart. Once the model is actuated, two joint sensors allow measuring the ZMP reference and acceleration of the cart in both axes. This information is used to calculate the moment around ZMP into two subsystems, as can be seen in Figure 3.22. The moments are calculated using the Equation (2.31).

The machine of the cart-table model is shown in Figure 3.23. The table is green and violet, and the cart is over the table.



**Figure 3.23:** *Cart-table SimMechanics mechanical model*



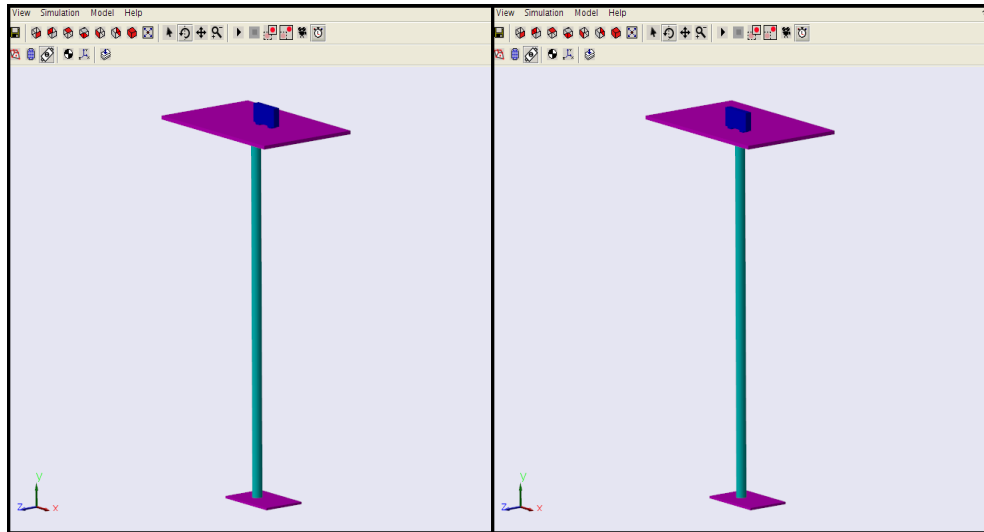
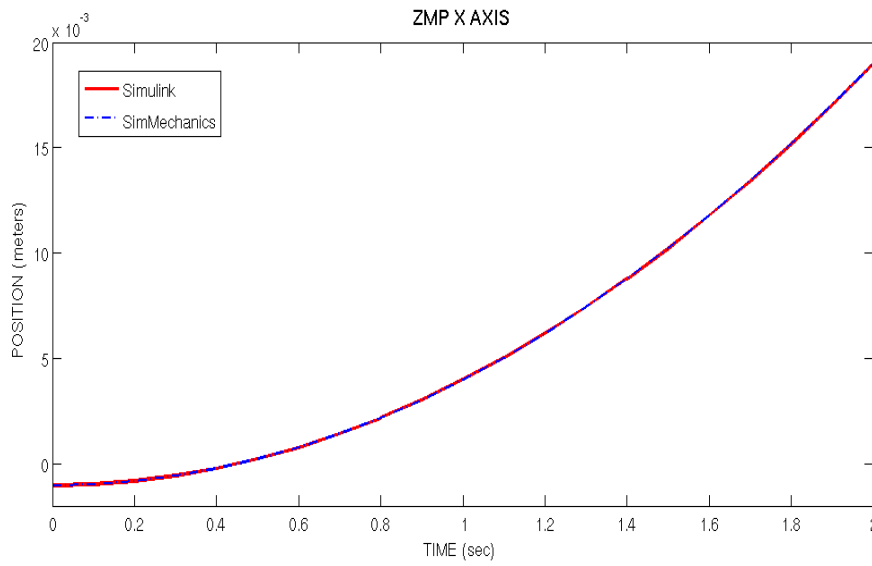


Figure 3.24: Cart-table machine in simulation

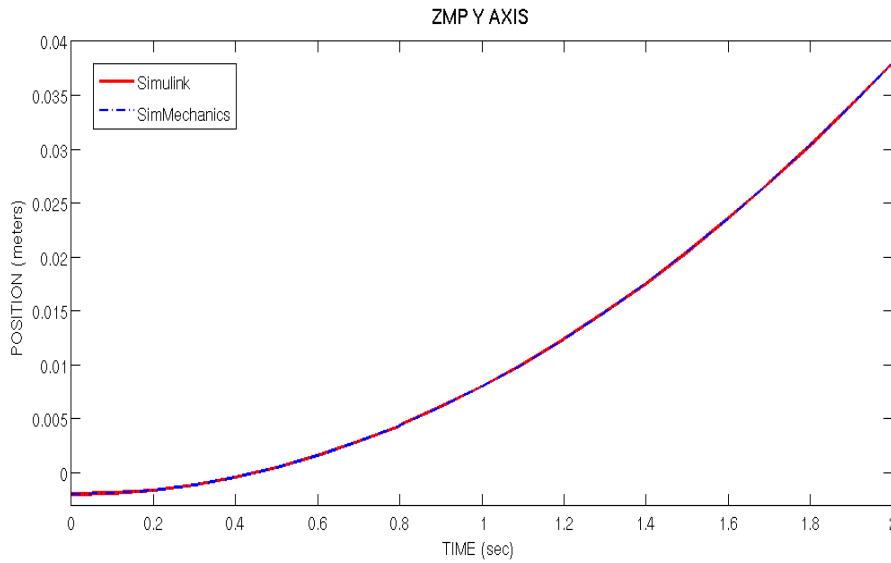
#### 3.3.4.4 Simulation results

The cart-table model has been simulated in order to validate its correct implementation using Simulink and SimMechanics. Before simulation, the user has been asked for the acceleration in both axes (X and Y). The acceleration values are  $0.01m/sec^2$  for the X axis and  $0.02m/sec^2$  for the Y axis, and the simulation time is  $2sec$ .

The simulation of the SimMechanics model is shown in Figure 3.24. At the beginning of the simulation, the cart is the centre of the table (shown on the left side). At the end of the simulation, the cart is located in another position due to the acceleration given to the cart, as can be seen on the right side. The Simulink model was also simulated, and the results are compared with the SimMechanics model. The results of the obtained ZMP for the X axis are shown in Figure 3.25, and the results of the obtained ZMP for the Y axis are shown in Figure 3.26.



**Figure 3.25:** Obtained ZMP for X axis in Simulink and SimMechanics models



**Figure 3.26:** Obtained ZMP for Y axis in Simulink and SimMechanics models

Making a comparison of the results for the given trajectory, it can be noticed that the physical model created using SimMechanics can work in a similar way

to the theoretical model created using the equations of motion of the cart-table model in Simulink. The obtained values are the same in both models and in both axes, as can be seen in Figure 3.25 and in Figure 3.26.

Due to the fact that good results have been achieved in the comparison between the Simulink and the SimMechanics model, in next the chapters the behaviour of the experimental platforms will be compared with the simulation results of the SimMechanics simplified models.

This choice has been made due to the creation of models in SimMechanics is simpler than in Simulink. SimMechanics allows modelling systems without knowing exactly their dynamics, since the models are created using blocks that imply directly the dynamics of the system.

On the other hand, another advantage of creating models using SimMechanics is that this tool offers a virtual representation of the systems. Moreover, to compare the behaviour of the experimental platforms with the models in simulation a control method is required, reason why, a control loop has been developed in every SimMechanics model.

### 3.4 Chapter summary

In this chapter the creation of the simplified models in simulation has been addressed. The models have been created from the motion equations of every model and using Simulink as software for simulation. The mechanical models have been implemented using SimMechanics.

The single inverted pendulum model explained in 3.3.2 has been simulated. The simulation results have shown that the model could not track correctly the trajectory, reason why, some adjustments are necessary.

Moreover, the cart-table model has been simulated, and the results of the simulation have been presented in this chapter. Once the comparison of the results is made, it can be concluded that the model created with SimMechanics

works similarly to the theoretical model(Simulink model) for the given trajectory.

Finally, SimMechanics becomes an easy tool for the creation of simplified models as has been said previously. The behaviour of the experimental platforms will be compared with the behaviour of the created models using SimMechanics in next chapters.

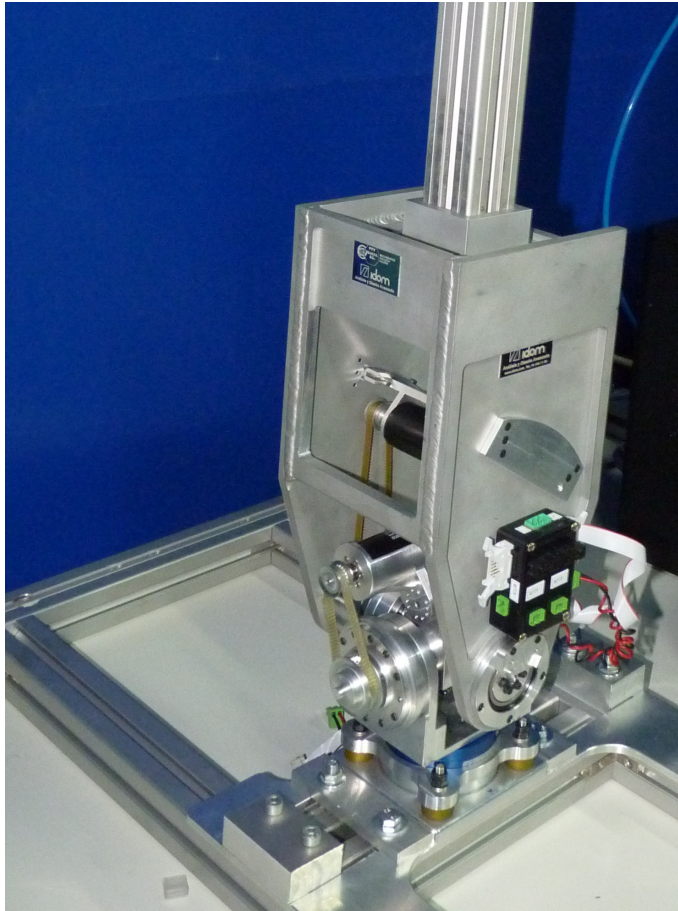
# Chapter 4

## Ankle Prototype as Experimental Platform

### 4.1 Introduction

Before creating the humanoid robot TEO, a test platform has been developed. This platform is the prototype of the ankle of this robot and is shown in Figure 4.1.

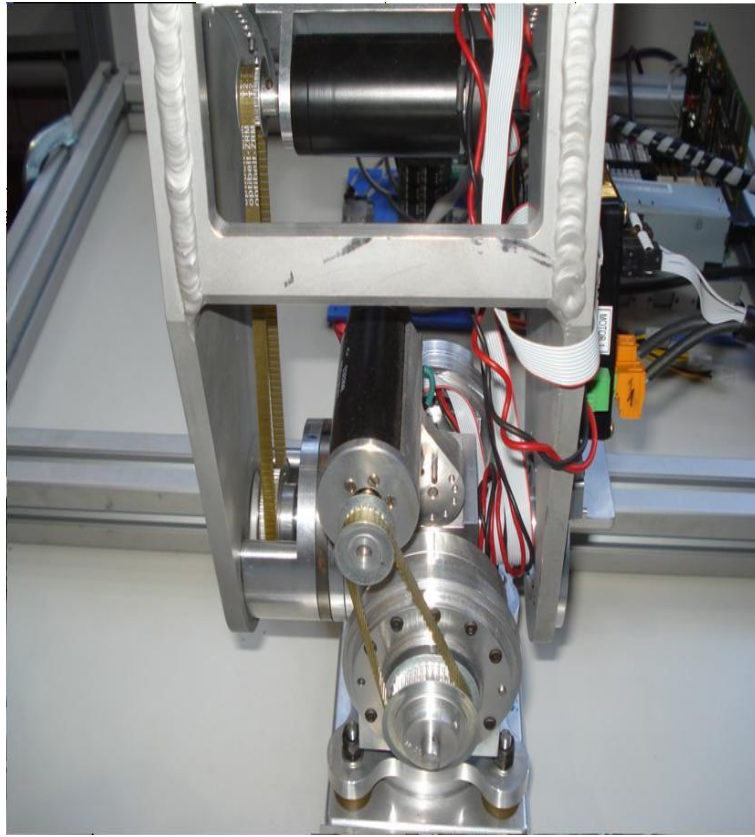
The ankle prototype includes two DOF; one on the sagittal plane and another one on the frontal plane. The height of the ankle is  $1m$  and has an approximate weight of  $10kg$ .



**Figure 4.1:** *Prototype of the ankle*

The prototype behaves as the single inverted pendulum and can perform two different motions due to two DOF situated at the frontal and sagittal planes.

So, we can make a comparison between the theoretical model of the single inverted pendulum model and the real platform, using the DOF in the sagittal plane. This allows us to know if the prototype works as the created models in Matlab.



**Figure 4.2:** *Motors and encoders of the prototype of the ankle*

## 4.2 Hardware description

As has been said previously, the prototype has two DOF and they are controlled individually through different motors. Each DOF has an integrated system called Maxon 315586 System that consists of a DC motor and a relative encoder shown in Figure 4.2.

There are other works as (Álvarez, 2011) and (Del Olmo, 2011), where it is explained how the prototype works. The hardware structure of the ankle can be seen in Figure 4.3, including the used elements and the communication between all of them.

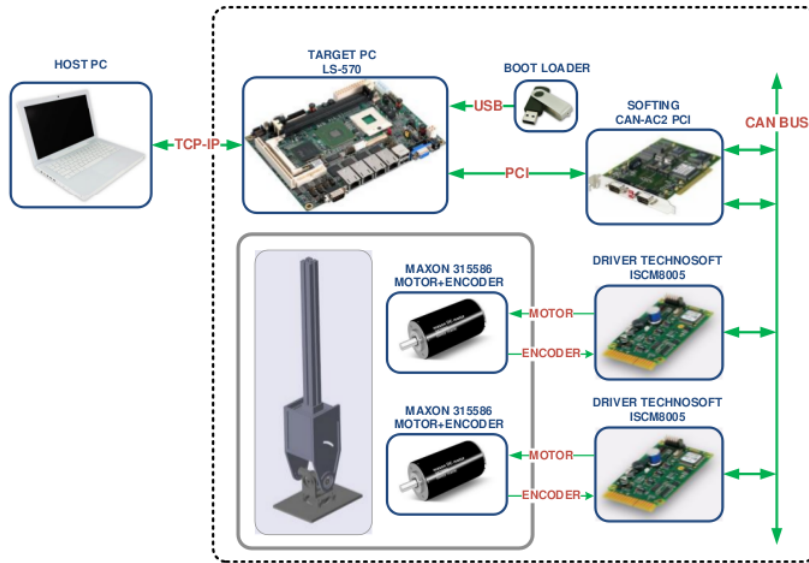


Figure 4.3: Hardware structure of the prototype

### 4.3 Simulation results

In order to test the correct performance of the single inverted pendulum in simulation, several tests have been done. As has been said in the previous chapter, the tested simulation model will be the one created using SimMechanics.

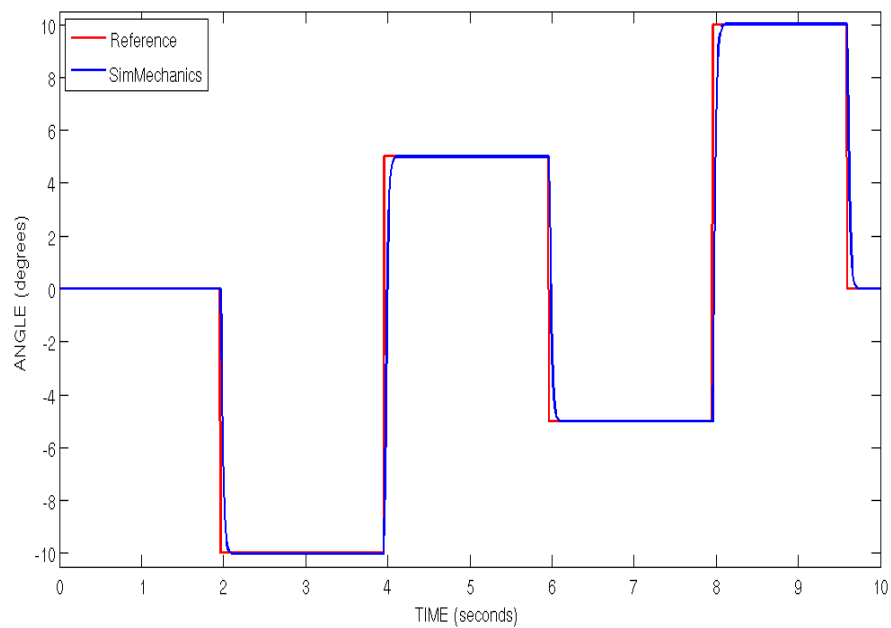
In order to achieve this goal, we have used a trajectory that is shown in Figure 4.4. The reference takes different position values between  $10deg$  and  $-10deg$ , and the simulation time is  $10sec$ . Moreover, the parameters of the system are  $m = 10kg$ ,  $l = 1m$ ,  $g = 9.8m/sec^2$ ,  $k = 0.1$ , and  $a = 0.8$ .

It is important to notice that this reference is not the most appropriate reference signal for simplified robot models tests, since it implies sharp changes in the position of the joint. The used reference is a step signal, which is used to characterize the system. This type of signal allows knowing how the system responds to a sudden input.

Figure 4.4 presents the simulation results in SimMechanics. The position



reference and the obtained positions in simulation can be seen in the figure. The simulation model has followed the given reference, and its response has been good, since the model uses a control loop that allows it to reach the required position values.



**Figure 4.4:** Simulation results of the single inverted pendulum

After the test is completed, a sequence of motions of the mechanical model during the simulation is shown in Figure 4.5.

The figure presents in the upper part from left to right the situation of the machine in 0sec, 3sec, 5sec, and the corresponding moments of 7sec, 9sec and 10sec can be seen in the lower part from left to right.

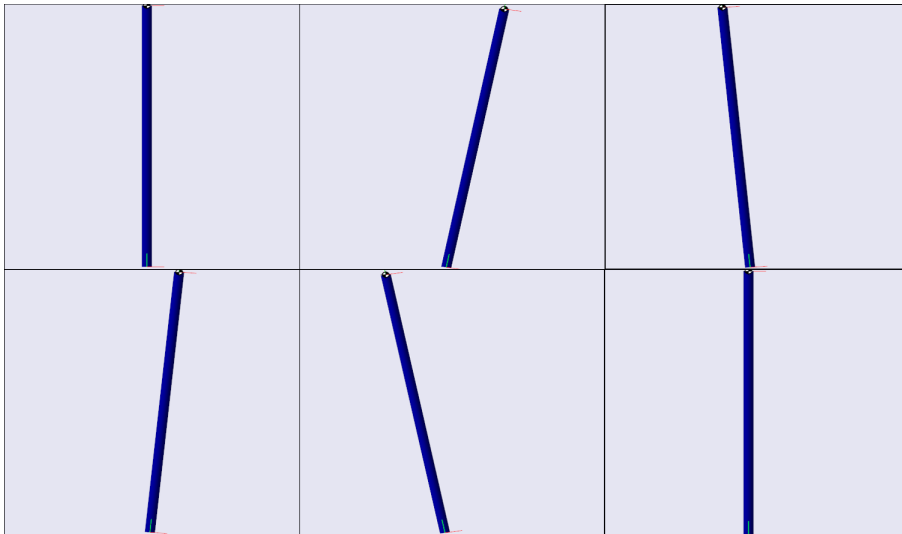


Figure 4.5: Simulation of the SimMechanics mechanical model

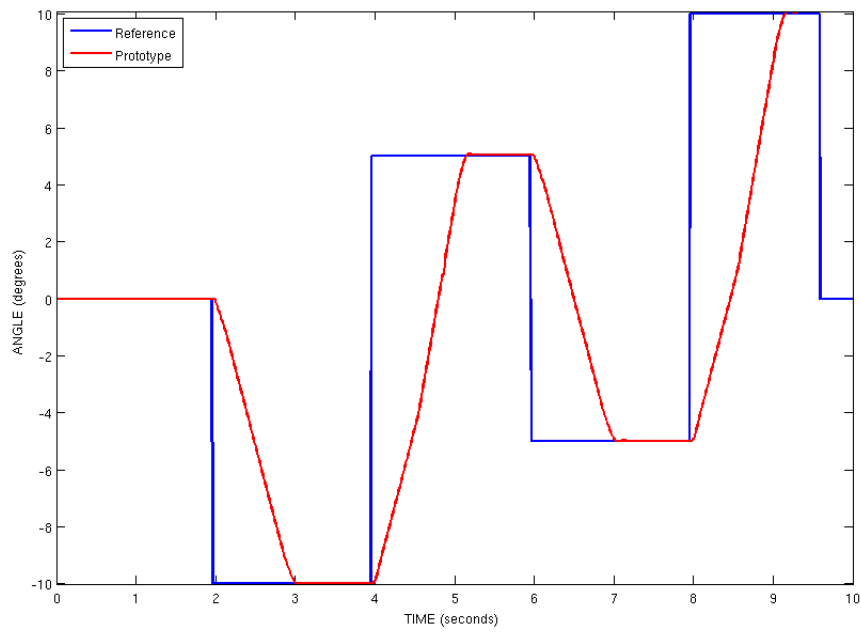


Figure 4.6: Test result of the ankle prototype

## 4.4 Experimental results

Once the Matlab model has been simulated, the behaviour of an experimental platform has been compared with the behaviour of the single inverted pendulum model. The used platform for this purpose has been the ankle prototype. A reference trajectory has been given, that has been used previously for the simulation of the SimMechanics model.

The prototype has been programmed using the method described in (Álvarez, 2011) and (Del Olmo, 2011).

Figure 4.6 shows the required position of the joint, and the obtained position of the ankle prototype after the test. The prototype can reach the required values, but the system response is not the expected one. The motor of the prototype includes an internal control loop, reason why, it can follow this reference doing slow motions. It is important to notice that the adjustment of this control loop is not one of the objectives of this work.

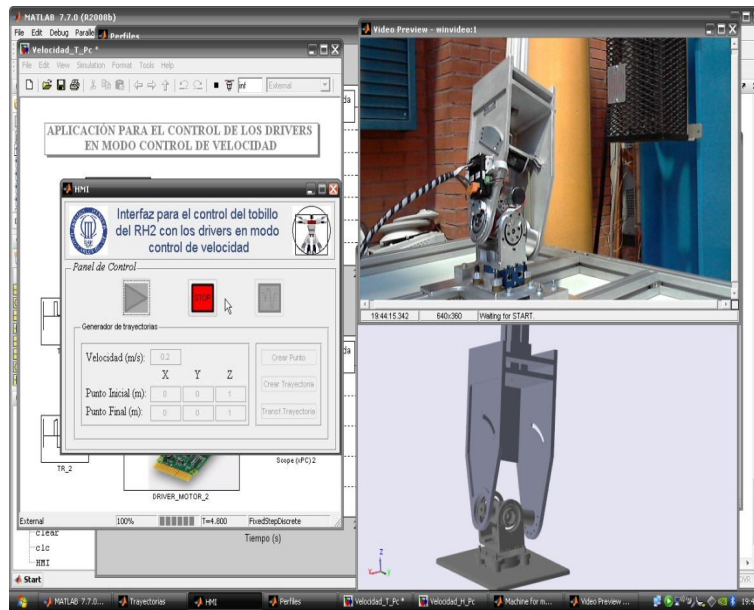
## 4.5 Discussion of results

As has been said previously, the given reference is not the most appropriate. This reference requires a fast response of the system, and it allows us to characterize the system.

First of all, the SimMechanics model has been simulated, and it could follow the reference due to the included control loop in the model.

Finally, the test of the ankle prototype has not been entirely bad. The prototype could achieve the required positions values through the included control loop of the motor.

Nevertheless, other tests have been achieved using the ankle prototype with good results. For instance, in (Álvarez, 2011) can be seen some tests using the experimental platform, and a simulation of a model based on the ankle and created using Matlab, as can be seen in Figure 4.7. This work deals with the design



**Figure 4.7:** Velocity control of the drivers of the ankle prototype

and implementation of a control architecture for the humanoid robot TEO. The used references in this work have been slow and smooth, and the ankle prototype could perform them successfully. An example of the given references is shown in Figure 4.8, the slopes of this trajectories are easy to achieve for the prototype.

## 4.6 Chapter summary

The ankle prototype has been used as an experimental platform in this chapter, and its behaviour has been compared with the behaviour of the single inverted pendulum model, that has been created using SimMechanics.

The test results have been presented and discussed in this chapter. Due to the given reference, the simulation and experimental results have not been entirely good. As has been explained previously, the reference signal implies a fast response of the system.

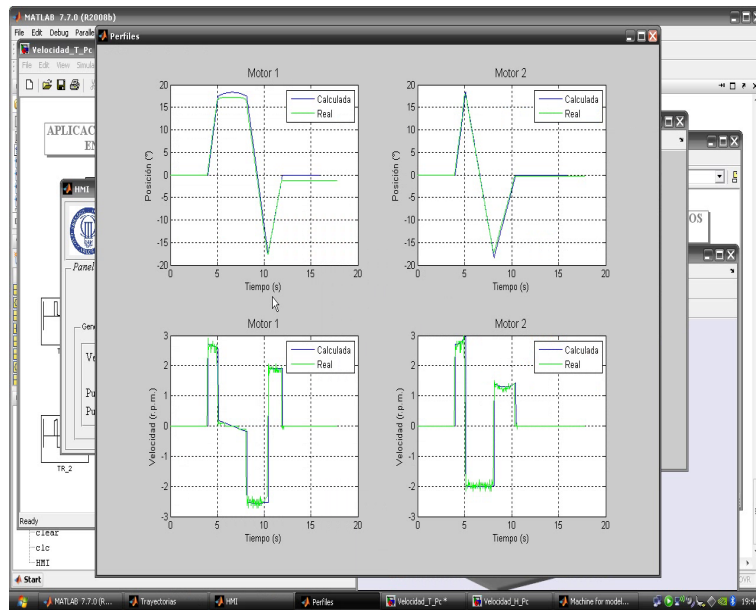


Figure 4.8: Obtained results after a test

The SimMechanics model could follow the required values in simulation, since it has its own control loop. On the other hand, the response of the ankle prototype was not so fast as it was required, since the given reference to the system has been a step signal. Nevertheless, other tests have been achieved using smooth trajectories as given references. The results of one of these test have been presented in this chapter.

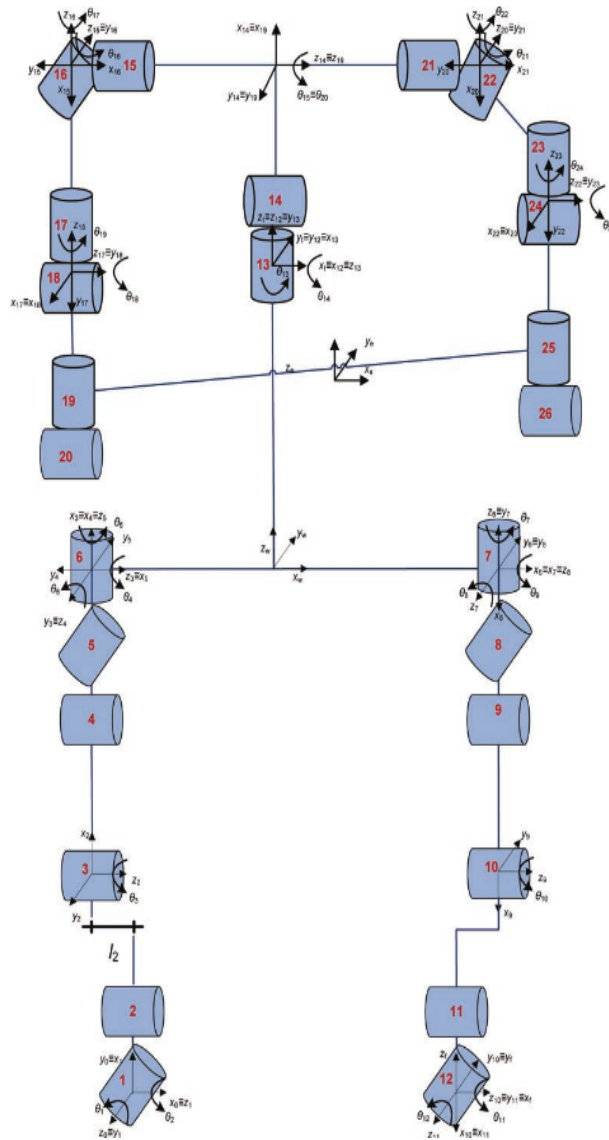


# Humanoid Robot TEO as Experimental Platform

## 5.1 Introduction

Humanoid robot TEO (Task Environment Operator) is the successful result of several years of research by the robotics group RoboticsLab at the Carlos III University of Madrid. It is an advanced version of RH-1, a prototype totally developed within the research team RoboticsLab.

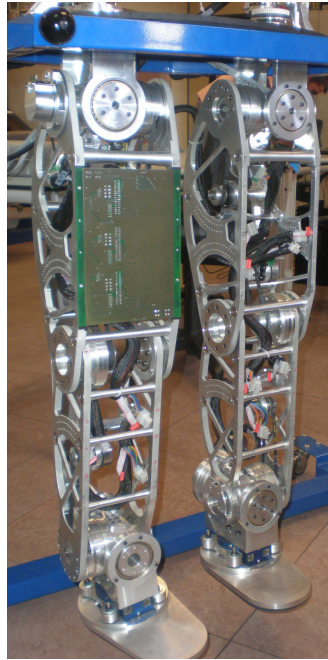
This robot has 28 DOF: six in each leg, six in both arms, two in the torso (waist), two in the chest (head), and two DOF due to the cameras. The distribution of DOF is shown in Figure 5.1. Its height is 1,65m and it has an estimated weight of 70kg.



**Figure 5.1:** Distribution of DOF of the robot TEO

The main objective is that the robot can perform different tasks in collaboration with humans in working environments, for instance, it will be able to carry an object with an approximate weight of  $2Kg$ . As a first approach, the lower part of its body has been assembled, as can be seen in Figure 5.2.





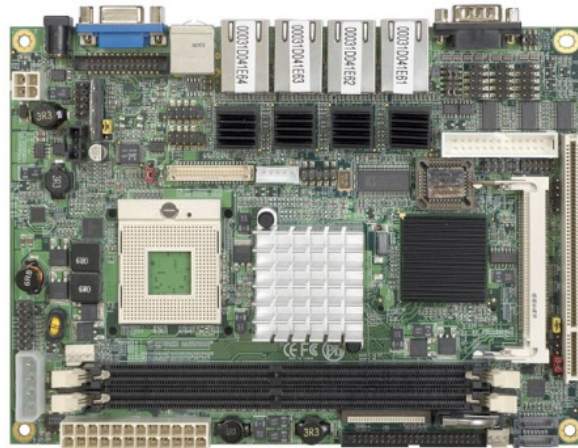
**Figure 5.2:** *Lower part body of the humanoid robot TEO*

## 5.2 Hardware description

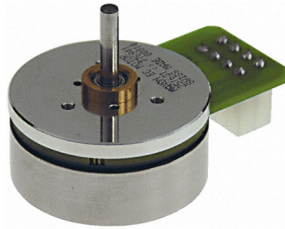
The hardware architecture proposed for the humanoid robot TEO is based in two microprocessors as computational system. The main microprocessor controls the lower part of the robot, and the other one the upper part. The chosen microprocessor is the model SBC Gemini 5.2" Embedded Intel Core 2 Duo of ARCOM (Figure 5.3).

The first microprocessor sends the signal of control to the motors located in the legs of the robot. Moreover, it uses the information obtained from the sensors in order to guarantee the stability of the robot during the walking action.

The other microprocessor controls the movements of the arms, and receives the environment information given by the sensors. Taking into account this information, the robot can perform several tasks. For instance, it can handle objects, avoid obstacles, etc.



**Figure 5.3:** *SBC Gemini 5.2'' Embedded Intel Core 2 Duo of ARCOM microprocessor*



**Figure 5.4:** *Maxon Brushless EC45 Flat 251601 motor*

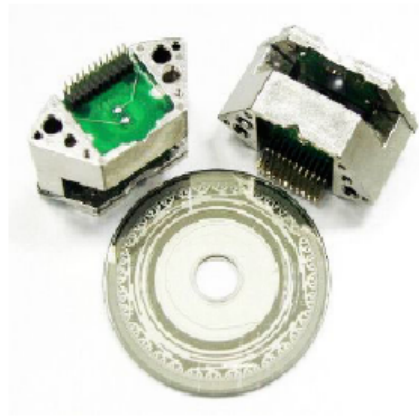
Using this architecture, a distributed system between two microprocessors is achieved. These microprocessors are connected to the microprocessors of the drivers and the encoders. The aim of this architecture is to obtain a high level of achievement. The motors that move the joints of the lower part of TEO are Brushless DC motors. The used motors are a Maxon Brushless EC45 Flat 251601 (Figure 5.4), for the sagittal joints, and a Maxon Brushless EC45 Flat 339287, for the frontal and axial joints. These motors include a Hall sensor to measure the relative position, and using a relative encoder coupled to the motor, the velocity of the motor can be measured.

These devices are controlled through the ISCM8005 driver of Technosoft, the device is shown in Figure 5.5.

A control position of the motors is required, to achieve this goal absolute



**Figure 5.5:** *ISCM8005 driver*



**Figure 5.6:** *7500 AEAS absolute encoder*

encoders are needed. They allow us to know the position of the motor, and then the absolute position of the joint. The selected absolute encoder is 7500 AEAS of Avago Technologies, shown in Figure 5.6.

The robot TEO needs to get information to work properly. Most of this information is obtained using force-torque sensors. The selected sensors are JR3 of JR3 Inc (in Figure 5.7), and they are located in the ankles and in the wrist of the robot. The aim of these sensors is measure the force and the torque applied in the contact point of the robot with other objects or the floor.



**Figure 5.7:** JR3 force-torque sensor

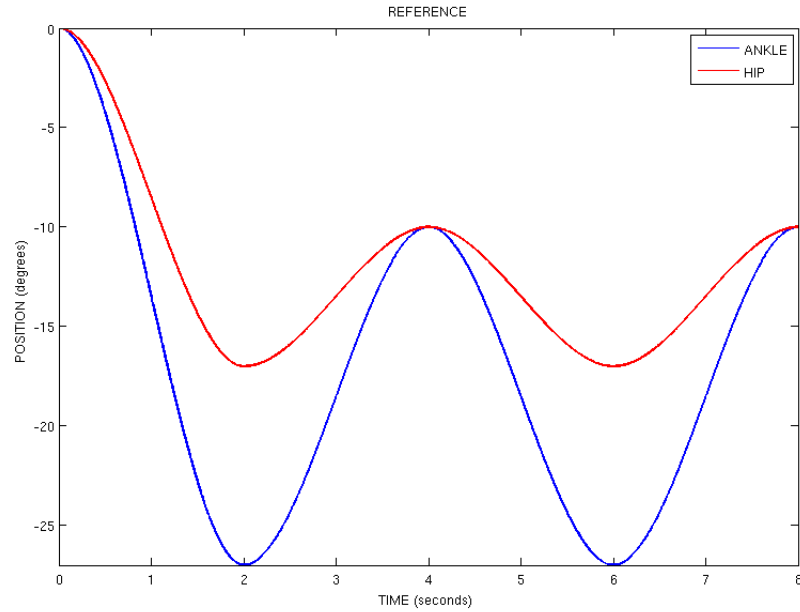
Furthermore, the robot TEO will include other type of sensors that allows it to work correctly depending on the environment and the required task.

Finally, it is worth mentioning the energy source for TEO (Monje et al., 2011). As a first approach, we have considered a fuel cell as a possible candidate to be the energy source for the robot. The key feature of small fuel cells to be used as battery replacements is the running time without recharging. We are currently working on the specific requirements regarding the whole fuel system installation.

### 5.3 Simulation results

In order to test the correct performance of the double inverted pendulum model using SimMechanics and shown in Section 3.3.3, several simulations have been done. A reference trajectory has been created for both DOF (Figure 5.8), and it is very smooth. The reference for the DOF of the ankle takes different values between  $-27deg$  and  $-10deg$ , and the values of the reference for the DOF of the hip are between  $-17deg$  and  $-10deg$ . The simulation time is  $8sec$  and the sample time is  $0.002$ . Moreover, the parameters of the system are  $m_1 = 50kg$ ,  $l_1 = 1m$ ,  $m_2 = 20kg$ ,  $l_2 = 0.2m$ ,  $g = 9.8m/sec^2$ , and  $k = 0.1$ .

First of all, the reference trajectories have been loaded and simulated using the SimMechanics model. The given references are the one presented previously



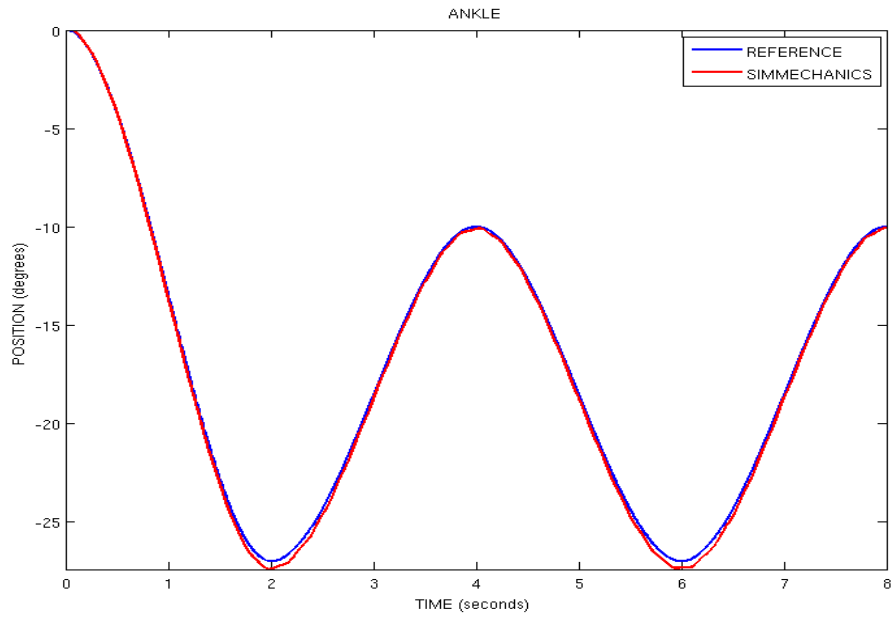
**Figure 5.8:** References for the simulation of the double inverted pendulum

in Figure 5.8. Once the simulation is completed, the simulation results of the DOF of the ankle are shown in Figure 5.9, and the obtained results of the DOF of the hip can be seen in Figure 5.10.

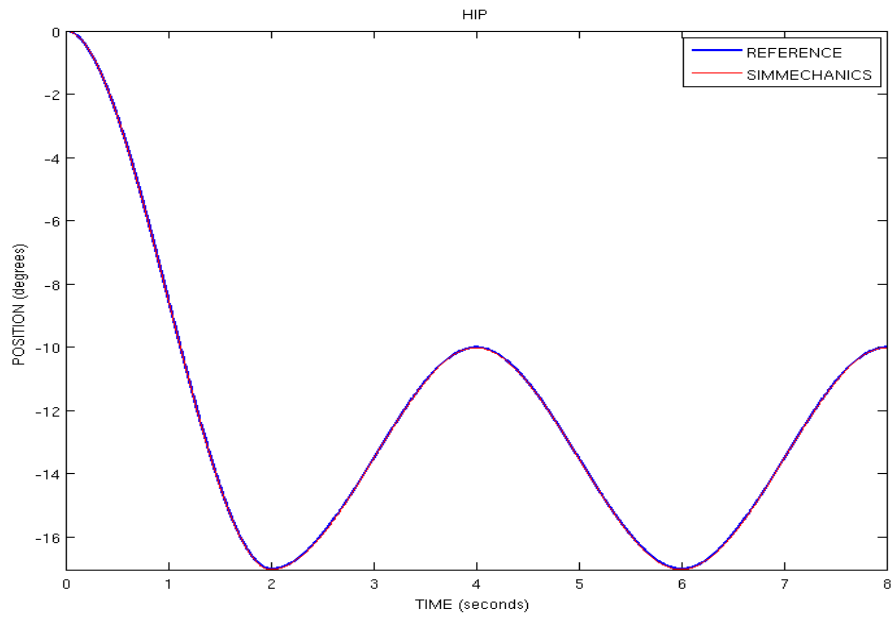
The response of the SimMechanics model is good, as can be seen in Figure 5.9 and Figure 5.10. Both joints have followed the given trajectories and reached most of the required values in simulation. However, it can be noticed that the ankle joint could not track the lowest values of the trajectory, and this could be fixed changing the PID controller parameters.

Moreover, a sequence of motions of the mechanical model during the simulation is shown in Figure 5.11.

The figure presents in the upper part from left to right the situation of the machine in  $0\text{sec}$ ,  $1\text{sec}$ ,  $2\text{sec}$ , and the corresponding moments of  $4\text{sec}$ ,  $6\text{sec}$  and  $8\text{sec}$  can be seen in the lower part from left to right.



**Figure 5.9:** Simulation results of the DOF of the ankle in SimMechanics



**Figure 5.10:** Simulation result of the DOF of the hip in SimMechanics

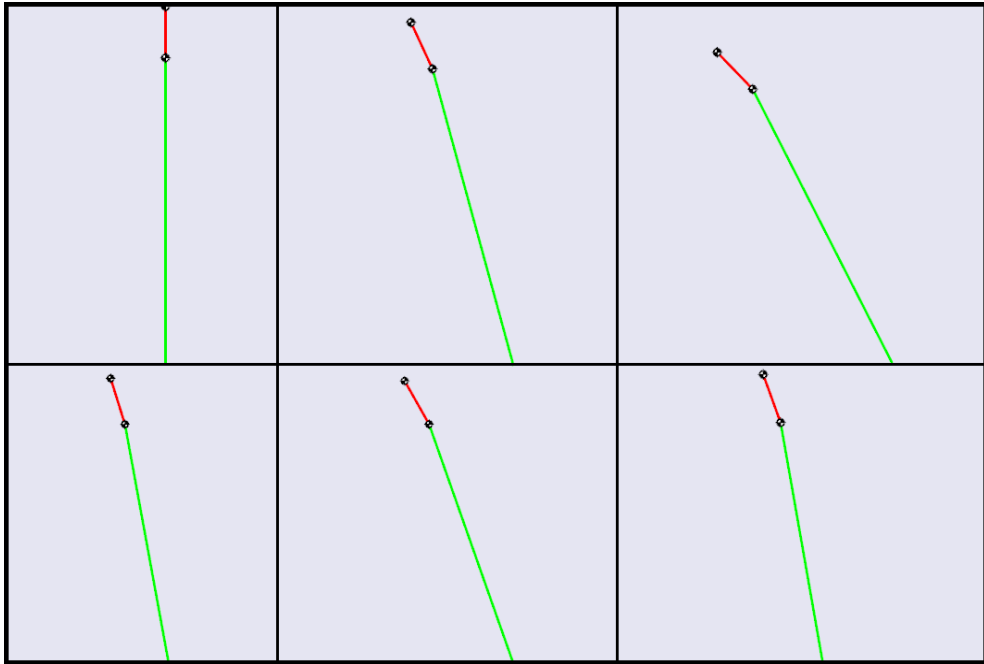


Figure 5.11: Simulation of the SimMechanics mechanical model

## 5.4 Experimental results

The experimental test has been done using the lower part of the humanoid robot TEO. For this purpose, we have used the sagittal DOF located at the ankle and the hip of the robot. It is important to notice that the DOF of the knee is also needed to perform a better motion, and it has been used in this test. The given reference trajectories to the robot joints have been used previously for the simulation of the SimMechanics model, and they have been shown in Figure 5.8.

A sequence of motions of the robot during the performance can be seen in Figure 5.12. Due to the fact that the reference is a smooth trajectory, the humanoid robot TEO has been able to perform correctly the required motions. Both joints have been reached the position values, as can be seen in Figure 5.13 and Figure 5.14.

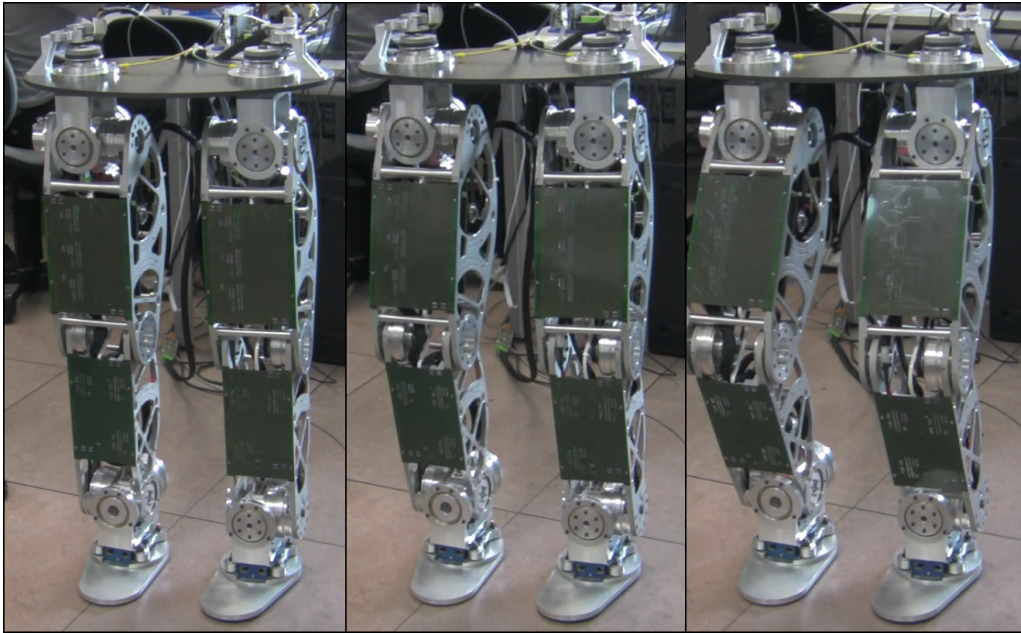


Figure 5.12: Sequence of motions of the robot TEO

## 5.5 Discussion of results

After the test has been done, good results have been achieved. Firstly, the Sim-Mechanics model has been simulated, and both joints (ankle and hip) could track correctly the required position values.

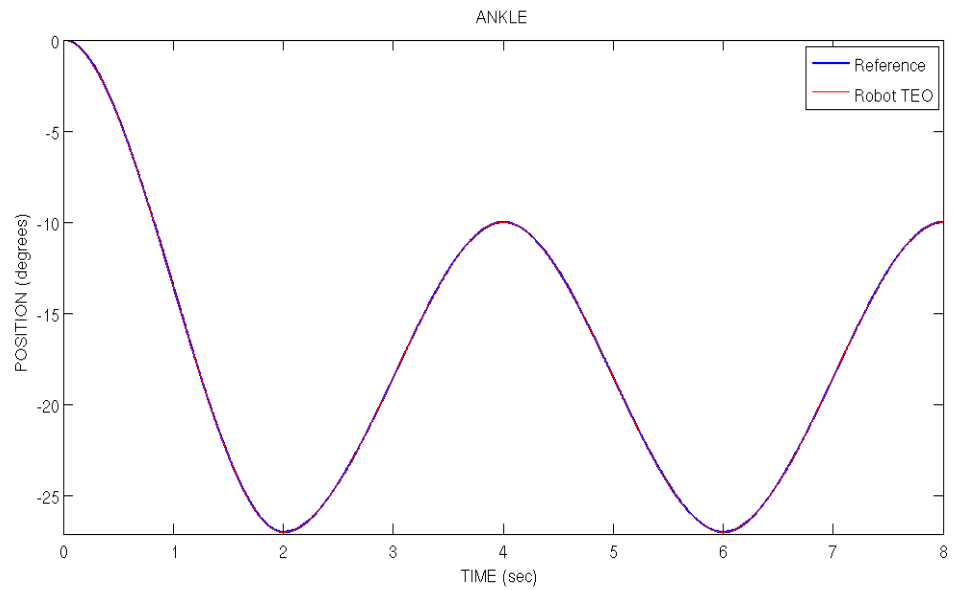
Once the trajectories have been validated in simulation, they have been tested in the real platform. The robot TEO could perform properly the given references for both joints.

As can be seen, the lower part of the humanoid robot can behave similarly as a double inverted pendulum.

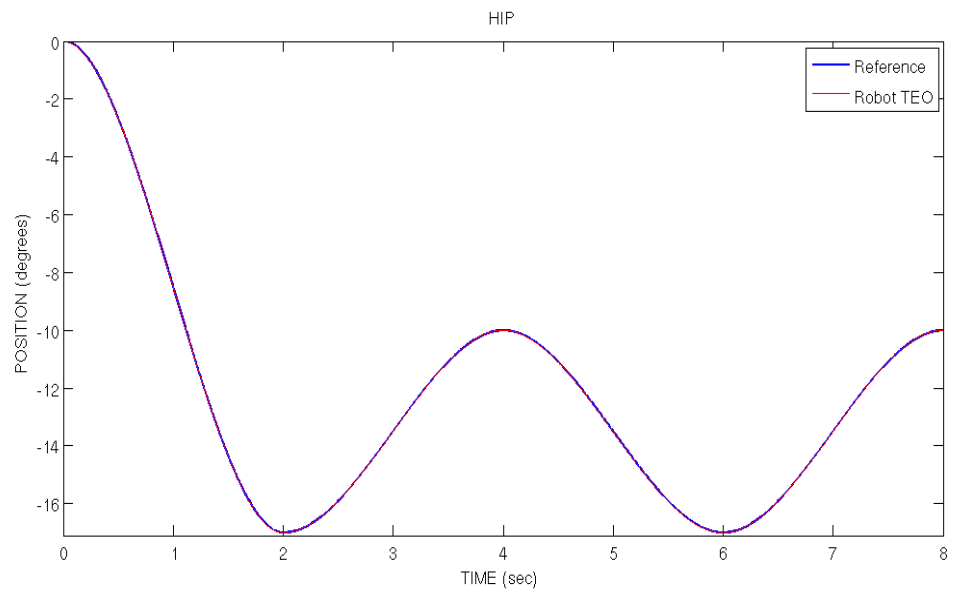
## 5.6 Chapter summary

The humanoid robot TEO has been presented in this chapter, and the behaviour of this platform has been compared with the behaviour of the double inverted





**Figure 5.13:** Positions of the ankle of the robot TEO



**Figure 5.14:** Positions of the hip of the robot TEO

pendulum model.

The test has been done using the lower part body of the robot (legs). Very good results have been achieved, and they have been presented and discussed in this chapter.

The SimMechanics model could follow the required values in simulation, and the performance of the robot TEO has been good.

To summarize, the lower part body of the robot TEO behaves in a similar way as the double inverted pendulum model.

# OpenHRP3 Simulation Platform for Modelling and Test Validation of Humanoid Robots

## 6.1 Introduction

The existence of computer simulation platforms is fundamental in robotics research, especially if it is with humanoid robots, since it allows us to develop the complete models for humanoid robots, the controllers and the necessary programming without compromising the complex and expensive mechanical system. In general, the objectives that the simulators allow us to approach are:

- To visualize three-dimensional work environment and the model of the robot in motion.
- To provide a test centre 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 behaviour of the virtual robot answers as closely as possible to the real robot, so the preparation work for a good 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.

One of the first principles and overviews of dynamic simulators was given in Baraff and Witkin (Witkin, 1997). The more advanced guidelines and the important problems that should be considered in humanoid dynamic simulations are mentioned in the General Human/Humanoid-Dynamics Simulator, proposed in Vukobratovic et al (Vukobratovic, Potkonjak, & Tzafestas, 2004). Although one can discuss the proposed implementations, e.g., the contact model, the paper provides some general guidance for the simulator design and the effects that should be taken into account in the dynamics simulations. Notably, they are a flexibility at the joints when the transmission between the motor and the corresponding joint is not completely rigid, but features some elasticity (one additional DOF) and a flexible (deformable) contact between the robot foot and the ground, i.e., elastodynamic contact (Fujimoto, Obata, & Kawamura, 1998)(Sugihara & Nakamura, 2003). One of the most widely used simulators, particularly for mobile robots, is the Player/Stage Open Source framework (Gerkey, Vaughan, & Howard, 2003). This consists of a Player robot device server and a 2D Stage multiple robot simulator. The main objective of the framework is research into multi-robot systems, with experiments and control of large population of robots without having to buy real hardware counterparts.

The Gazebo (Koenig & Howard, 2004) platform, an add-on to the Player/Stage framework, is based on OpenGL graphics, specifically on the GLUT toolkit, and Open Dynamics Engine. However, it should generally work in conjunction with Player software running on the robot, and it is mostly applied to mobile robot control. Some simulators, e.g., in Ponticelli and Armada (Ponticelli & Armada,

2006), are developed purely in Matlab using Simulink toolboxes, such as SimMechanics, and visualizing systems, such as VRML Viewer toolbox. This approach enables rapid controller design and testing, but it lacks some important features such as surface modeling, and consequently has no collision detection feature.

OpenRAVE (Diankov, 2010) (The Open Robotics Automation Virtual Environment) is a robotics planning and simulation tool. As OpenHRP3, it could almost be considered a complete software architecture, with certain limitations (in OpenRAVE, all component must run on the same machine). Its architecture is plug-in driven and supports the integration of custom functionalities, such as planning, control, or sensing modules, that are loaded at run time. The simulator is designed to be multi-platform, and many of the components are reusable. Interaction with the simulator can be performed through high level scripts in scripting environments such as Python, Octave or Matlab. Integration with ROS has also been achieved.

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), and Webots (a commercial software by Cyberbotics). Even Microsoft has developed a product named Microsoft Robotics Studio, which is primarily used for mobile robots.

It is important to mention the OpenHRP3 platform (Kanehiro, Miyata, & Kajita, 2001)(Kanehiro, Hirukawa, & Kajita, 2004) Open Architecture Humanoid Robotics Platform) as a simulator and motion control library for humanoid robots developed at NIAIST (National Institute of Advanced Industrial Science and Technology (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. This is a free solution that will be used in this work to model the humanoid robots TEO and HOAP-3 and simulate walking patterns before the final test in real time with the real robots.

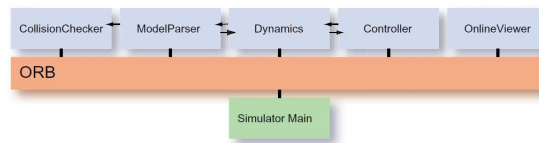
## 6.2 OpenHRP3 platform

OpenHRP3 (*OpenHRP3 official site*, 2008) (Open Architecture Humanoid Robotics Platform, version 3) is a simulation platform for humanoid robots and software development. It allows the users to inspect the original model of the robot and the control program across a dynamic simulation. In addition, OpenHRP3 provides several calculation software components and libraries that can be used to develop software related to robotics.

To use OpenHRP3 the following programs, libraries, and programming languages are needed (Yamane & Nakamura, 1999): Microsoft Visual Studio, BOOST, CLAPACK, and TVMET libraries, graphical environment OpenRT, Adaptive Communication Environment (ACE), ORB, Python, Java, and Jython. OpenHRP3 is currently supported on Ubuntu Linux platforms 7 or later and Windows XP and Vista (32bit/64bit). In our case we use Windows XP 64bit.

This virtual humanoid robot platform consists of a simulator of humanoid robots and motion control library for them, which can also be applied to a compatible humanoid robot as it is. OpenHRP also has a view simulator of humanoid robots on which humanoid robot vision can be studied. The consistency between the simulator and the robot is enhanced by introducing a new algorithm to simulate repulsive force and torque between contacting objects. OpenHRP is expected to initiate the exploration of humanoid robotics on an open architecture software and hardware, thanks to the unification of the controllers, and the examined consistency between the simulator and a real humanoid robot.

The configuration of OpenHRP is shown in Figure 6.1. OpenHRP can simulate the dynamics of structure-varying kinematic chains, both open chains and



**Figure 6.1:** *OpenHRP3 functions*

closed ones such as humanoid robots (*Object Management Group, 1997*). It can detect the collision between robots and their working environment (including other robots) very fast and precisely, computing the forward dynamics of the objects. It can also simulate the fields of vision of the robots, force/torque sensors, and gradient sensors according to the simulated motions. The sensor simulations are essential to develop the controllers of the robots. OpenHRP is implemented as a distributed object system on CORBA (Gottschalk, Lin, & Manocha, 1996). A user can implement a controller using an arbitrary language on an arbitrary operating system if it has a CORBA binding.

The dynamics simulator of OpenHRP consists of five kinds of CORBA servers (see Figure 6.1) and these servers can be distributed on the Internet and executed in parallel. Each server can be replaced with another implementation if it has the same interface defined by IDL (Interface Definition Language). Using the language independence feature of CORBA, ModelParser and OnlineViewer are implemented using Java and Java3D, other servers are implemented using C++. The functions of each server are as follows.

- **ModelParser.** This server loads a VRML file describing the geometrical models and dynamics parameters of robots and their working environment, and provides these data to other servers.
- **CollisionChecker.** The interference between two sets of triangles is inspected, and the position, normal vector, and the depth of each intersecting point are found. RAPID (Robotics Application Programming Interactive Dialogue) is enhanced to this end.

- Dynamics. The forward dynamics of the robots are computed.
- Controller. This server is the controller of a robot, which is usually developed by the users of OpenHRP.
- OnlineViewer. The simulation results are visualized by 3D graphics and recorded.

Using the servers, the forward dynamics of the robots are computed in the following procedure. The total control flow is shown in Figure 6.1.

- Setting up of the simulation environment. ModelParser reads a VRML file via HTTP protocol. The kinematics and dynamics parameters are sent to Dynamics and the geometric model is to CollisionChecker.
- Execution of the dynamics simulation. Controller reads the outputs of the simulated sensors while communicating with Dynamics. Controller and Dynamics execute the computations. Note that these computations can be run in parallel. The outputs of Controller are the torques of the actuators, and those of Dynamics are the updated states of the robot. While the forward dynamics is computed, CollisionChecker is called to find the position, normal vector, and the depth of each intersecting point. After these computations, Controller sends the control outputs to Dynamics.
- Visualization and recording. The current states of the world are sent from Dynamics to OnlineViewer, which visualizes the simulated world and records it.

### 6.2.1 OpenHRP3 simulation interface

The simulation interface of the OpenHRP3 platform is shown in Figure 6.2.

The interface of this simulator is basically divided into three parts. The first part (Figure 6.2, Part 1, left) shows the tree structure of the different modules



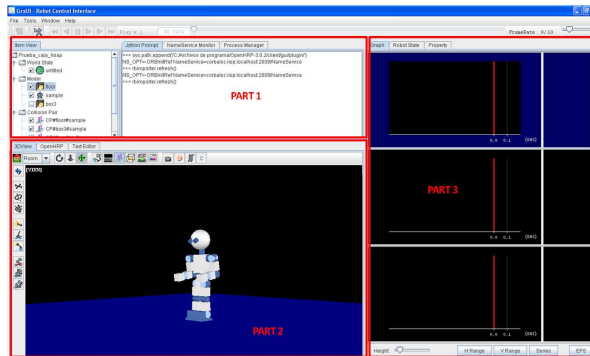


Figure 6.2: *OpenHRP3 interface*

loaded for the simulation. These modules can be the model of the robot, the model of the environment, the collision (Collision Pair) and control modules, and the graphical display package, among others. A command window is available at the right side of Part 1 to introduce command lines directly.

The second part is directed to display the robot model during the simulation (Figure 6.2, Part 2). In this area there are two toolbars, the vertical and the horizontal ones, which allow us to modify aspects of the display of the robot environment, take pictures, and record videos of tasks.

The third part is placed at the right of the interface (Figure 6.2, Part 3). It displays the signals that represent the evolution of the joint variables during the simulation (position, velocity, acceleration), as well as the information of the different sensors included along the mechanical structure of the robot (force/torque sensors, etc ).

### 6.3 Creating models in OpenHRP3

For modelling in OpenHRP, the VRML (*The Virtual Reality Model Language*, n.d.) file describing the geometrical model and dynamics parameters of the robot must be created. The main joint of the model is the WAIST, and the upper (arms) and lower (legs) bodies are defined from it, in this order.

The Humanoid node is the root node of the model. Along with it, Joint and Segment nodes (joint and link, respectively) are basic elements of VRML, and define the type of each joint (Joint) and its corresponding shape (Segment).

Once the VRML model of the robot is created according to the geometrical and dynamic parameters provided, the file can be loaded as a module in the tree structure of the simulator interface, as defined previously.

In order to reduce the computation cost and speed up the test procedure, it has been modelled a more basic shape of the robot instead of modelling its real appearance, which would imply a more complex VRML file to be computed.

### 6.3.1 TEO model

As has been said in Chapter 5, the humanoid robot TEO has 28 DOF, six in each leg, six in both arms, two in the torso (waist), and two in the chest (head). Its height is 1,65 m and it has an estimated weight of 60 kg. The VRML structure that realtes the different DOF of the TEO is shown in Figure 6.3.

### 6.3.2 HOAP-3 model

The miniature humanoid robot HOAP-3 (FUJITSU, 2005) (Humanoid for Open Architecture Platform) has been developed by Fujitsu in collaboration with Fujitsu Automation Lab (Figure 6.4).

The HOAP-3 robot has a height of 60 cm and an approximate weight of 8 Kg. It has 28 DOF distributed as shown in Figure 6.5 (up). The VRML structure that relates the different DOF of the robot is shown in Figure 6.5 (down).

## 6.4 Simulation and experimental results

Our purpose is to test the motion patterns obtained in the previous section in the real robots. Prior to this experimental test, the OpenHRP3 simulation platform is used to check the stability of the robots during the motion pattern in simulation.

```

+-humanoidBody
|
| # Upper part body
+-Joint WAIST : Segment WAIST_LINK0
|   Joint WAIST_P : Segment WAIST_LINK1
|   Joint WAIST_Y : Segment WAIST_LINK2
|   |
|   | # Cameras
+-VisionSensor VISION_SENSOR1
+-VisionSensor VISION_SENSOR2
|
|   # Left arm
+-Joint LARM_SHOULDER_P : Sejdghj
|   Joint LARM_SHOULDER_R : Segment LARM_LINK2
|   Joint LARM_SHOULDER_Y : Segment LARM_LINK3
|   Joint LARM_ELBOW : Segment LARM_LINK4
|   Joint LARM_WRIST_Y : Segment LARM_LINK5
|   Joint LARM_WRIST_P : Segment LARM_LINK6
|
|   # Right arm
+-Joint RARM_SHOULDER_P : Segment RARM_LINK1
|   Joint RARM_SHOULDER_R : Segment RARM_LINK2
|   Joint RARM_SHOULDER_Y : Segment RARM_LINK3
|   Joint RARM_ELBOW : Segment RARM_LINK4
|   Joint RARM_WRIST_Y : Segment RARM_LINK5
|   Joint RARM_WRIST_P : Segment RARM_LINK6
|
|   # Left leg
+-Joint LLEG_HIP_R : Segment LLEG_LINK1
|   Joint LLEG_HIP_P : Segment LLEG_LINK2
|   Joint LLEG_HIP_Y : Segment LLEG_LINK3
|   Joint LLEG_KNEE : Segment LLEG_LINK4
|   Joint LLEG_ANKLE_P : Segment LLEG_LINK5
|   Joint LLEG_ANKLE_R : Segment LLEG_LINK6
|
|   # Right leg
+-Joint RLEG_HIP_R : Segment RLEG_LINK1
|   Joint RLEG_HIP_P : Segment RLEG_LINK2
|   Joint RLEG_HIP_Y : Segment RLEG_LINK3
|   Joint RLEG_KNEE : Segment RLEG_LINK4
|   Joint RLEG_ANKLE_P : Segment RLEG_LINK5
|   Joint RLEG_ANKLE_R : Segment RLEG_LINK6

```

**Figure 6.3:** VRML structure of TEO



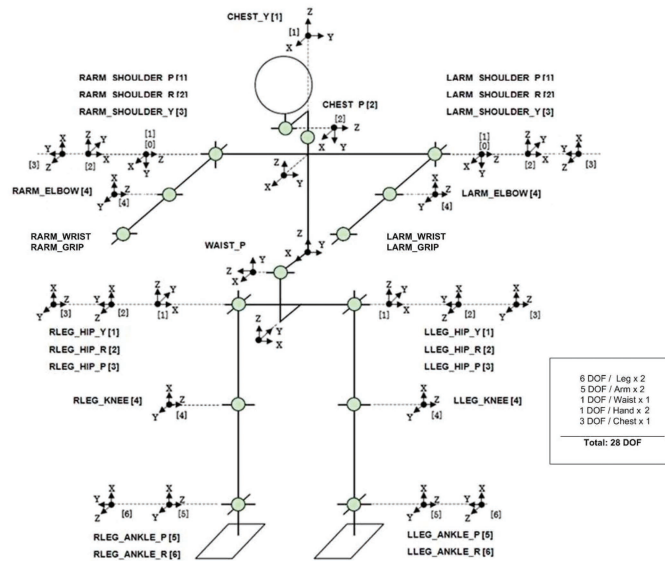
**Figure 6.4:** *Humanoid robot HOAP-3*

The specifications of the computer used include CPU: Intel Core Duo 2.4GHz, RAM Memory: 2GB, and OS: Windows XP 64 bits.

The VRML file of the corresponding robot model is loaded in the modules tree of the simulator, together with the CollisionChecker and Controller modules defined by OpenHRP3.

In order to generate stable walking patterns of the humanoid, we have used the cart-table model (Kajita et al., 2003). This model is based on the ZMP preview control scheme, that obtain the COG trajectory from a defined ZMP trajectory. The model has been studied in Section 2.5. Once the COG trajectory is obtained, the trajectories of the lower body joints are calculated applying inverse kinematics.

Finally, once the stability of the robots is guaranteed in simulation, the joints trajectories are loaded in the real platforms.



```

+-humanoidBody
|
| # Upper part body
+-Joint WAIST : Segment WAIST_LINK0
|
|   Joint WAIST_P : Segment WAIST_LINK1
|   Joint CHEST_Y : Segment WAIST_LINK2
|   Joint CHEST_R : Segment WAIST_LINK3
|   Joint CHEST_P : Segment WAIST_LINK4
|
|   | # Cameras
|   +-VisionSensor VISION_SENSOR1
|   +-VisionSensor VISION_SENSOR2
|
| # Left arm
+-Joint LARM_SHOULDER_P : Segment LARM_LINK1
|
|   Joint LARM_SHOULDER_R : Segment LARM_LINK2
|   Joint LARM_SHOULDER_Y : Segment LARM_LINK3
|   Joint LARM_ELBOW : Segment LARM_LINK4
|   Joint LARM_WRIST_Y : Segment LARM_LINK5
|
| # Right arm
+-Joint RARM_SHOULDER_P : Segment RARM_LINK1
|
|   Joint RARM_SHOULDER_R : Segment RARM_LINK2
|   Joint RARM_SHOULDER_Y : Segment RARM_LINK3
|   Joint RARM_ELBOW : Segment RARM_LINK4
|   Joint RARM_WRIST_Y : Segment RARM_LINK5
|
| # Left leg
+-Joint LLEG_HIP_R : Segment LLEG_LINK1
|
|   Joint LLEG_HIP_P : Segment LLEG_LINK2
|   Joint LLEG_HIP_Y : Segment LLEG_LINK3
|   Joint LLEG_KNEE : Segment LLEG_LINK4
|   Joint LLEG_ANKLE_P : Segment LLEG_LINK5
|   Joint LLEG_ANKLE_R : Segment LLEG_LINK6
|
|   | # Force sensor
|   +-ForceSensor LEFT_FORCE_SENSOR
|
| # Right leg
+-Joint RLEG_HIP_R : Segment RLEG_LINK1
|
|   Joint RLEG_HIP_P : Segment RLEG_LINK2
|   Joint RLEG_HIP_Y : Segment RLEG_LINK3
|   Joint RLEG_KNEE : Segment RLEG_LINK4
|   Joint RLEG_ANKLE_P : Segment RLEG_LINK5
|   Joint RLEG_ANKLE_R : Segment RLEG_LINK6
|
|   | # Force sensor
|   +-ForceSensor RIGHT_FORCE_SENSOR

```

Figure 6.5: Distribution of DOF in HOAP-3 (up) and VRML structure (down)

### 6.4.1 Simulation and experimental results for TEO

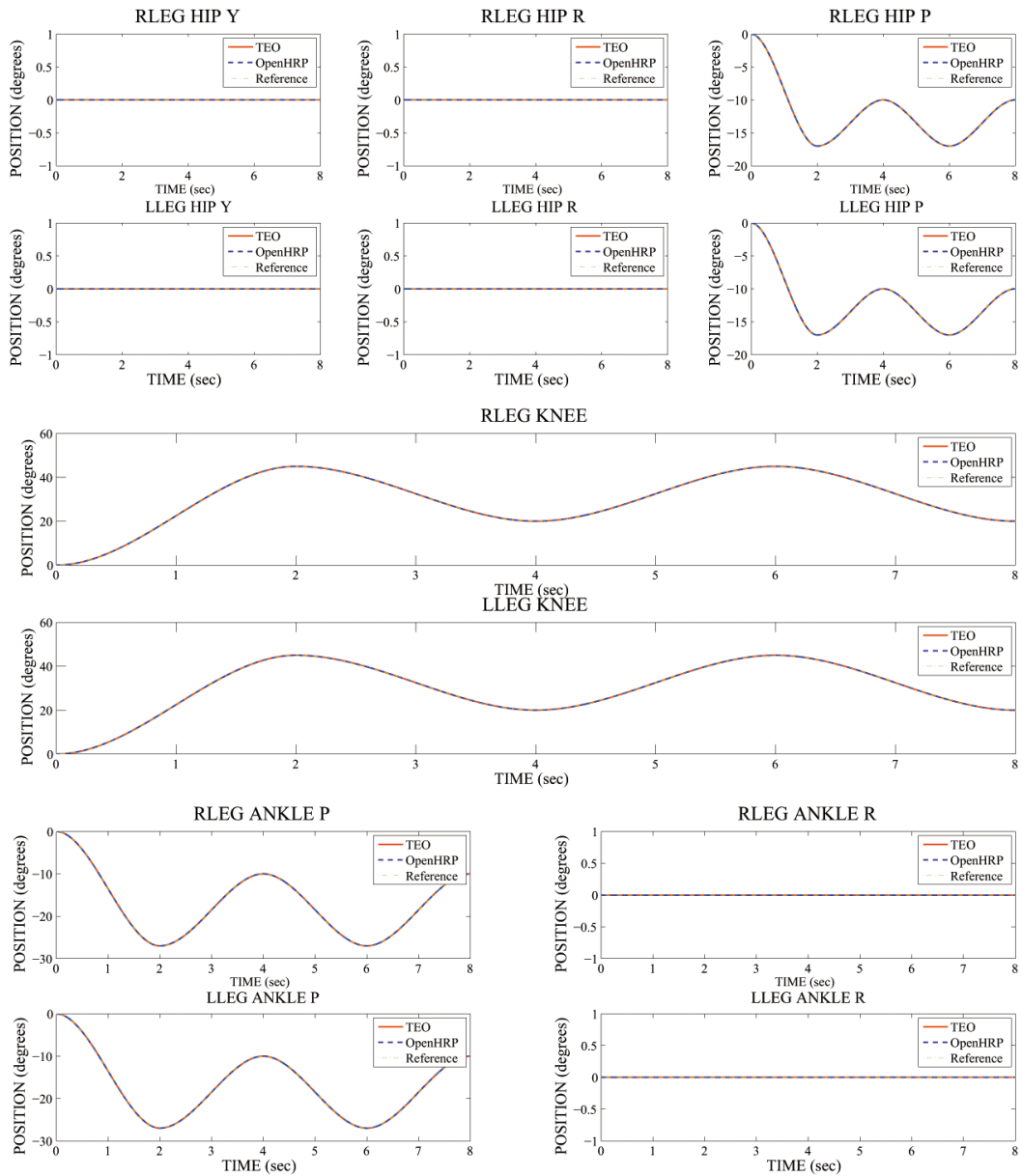
In the motion pattern simulated for the robot TEO, the robot bends its knees twice. The simulation time, in this case, is 8 seconds. After the simulation test, the stable sequence of motions are tested in the real platform (lower part body of the robot). Very good results are obtained, as can be seen in Figure 6.6 comparing the experimental and simulation joint angles measured by the robot encoders and simulator angular position sensors, respectively.

The whole task can be seen in the following video <http://www.youtube.com/watch?v=90JENgbeZvU>, the robot performs the motion pattern in real time, and the simulation in OpenHRP3 is shown in a left window, see Figure 6.7.

### 6.4.2 Simulation and experimental results for HOAP-3

In the motion pattern simulated for the robot HOAP-3, the robot walks 12 step forward. The step distance selected is 8 cm, a COG height of 32 cm, and a preview time of 0.75 sec. The simulation time in this case is 18 seconds. After the simulation test, the joints trajectories are loaded in the real HOAP-3 platform, and the walking test is performed. Very good results are obtained, as can be seen in Figure 6.8 comparing the experimental and simulation joint angles measured by the robot encoders, and simulator angular position sensors, respectively.

According to the good results obtained, a more complex and complete sequence of motions has been tested in HOAP-3 following the same procedure explained here. Several dance steps have been performed by HOAP-3, as can be seen in the following video showing the robot dancing in real time in an exhibition during 3 minutes: <http://www.youtube.com/watch?v=mu5psxG7bwA>. As can be seen in the video, the simulation in OpenHRP3 is run and shown in a top window during the performance. The simulation also includes the same environment as the real one (forest area), as can be seen in the sequence shown in Figure 6.9.



**Figure 6.6:** Trajectories for both right and left legs joints (12 DOF), in OpenHRP3 and real TEO

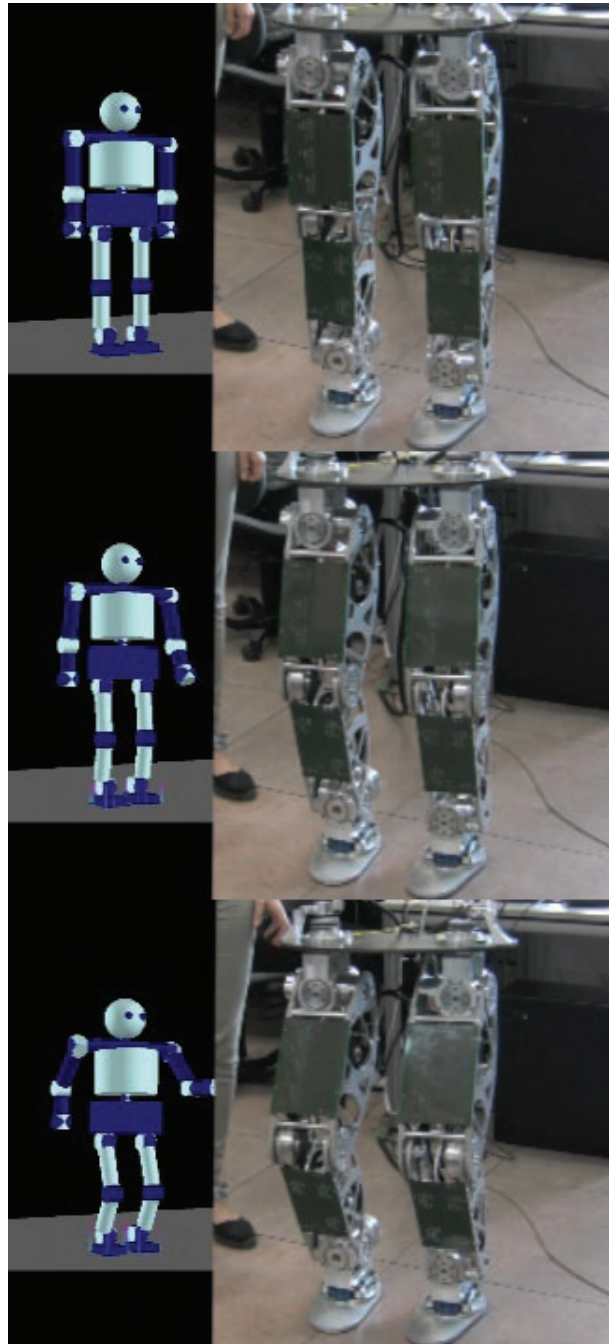
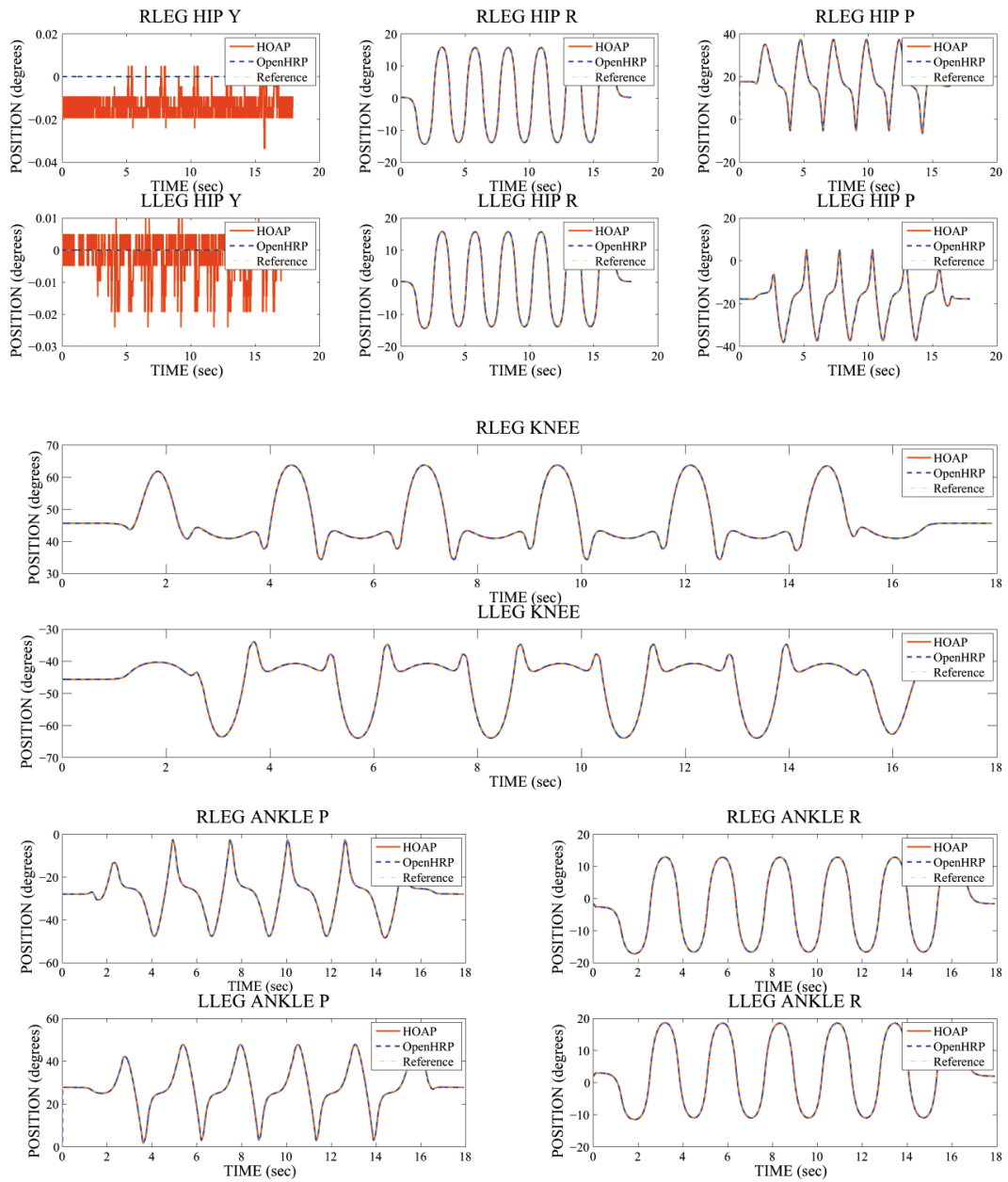


Figure 6.7: Sequence of motions for TEO during the performance





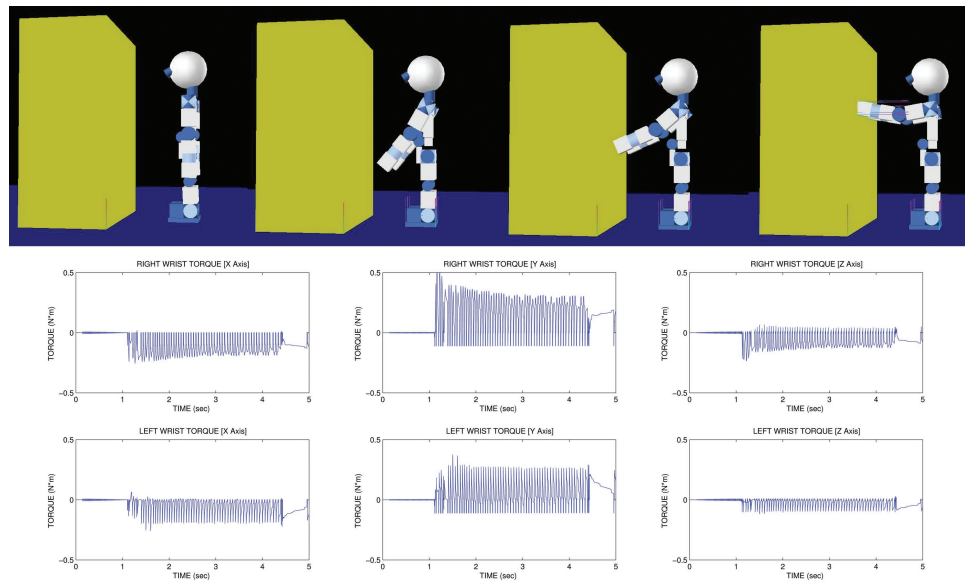
**Figure 6.8:** Trajectories for both right and left legs joints (12 DOF), in OpenHRP3 and real HOAP-3



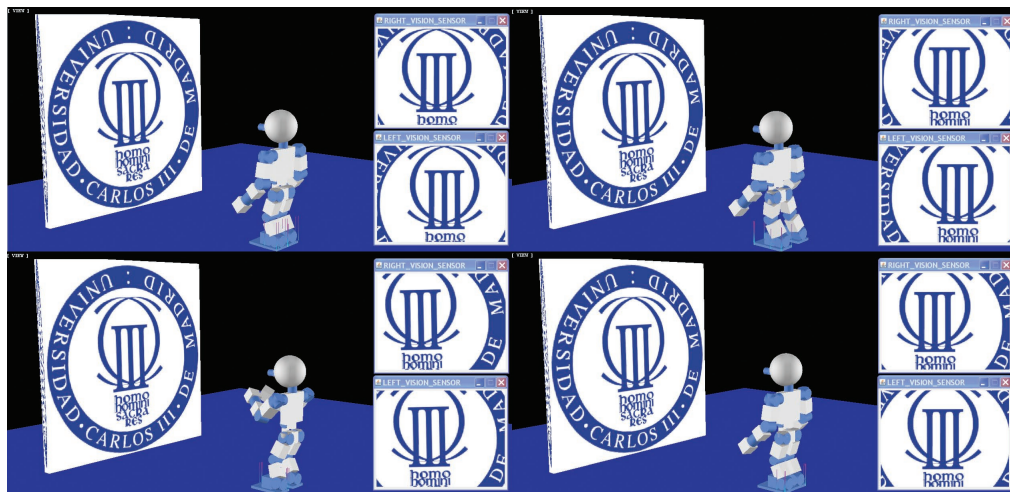
**Figure 6.9:** *Sequence of motions for HOAP-3 during the performance*

Another important approach is the manipulation of objects using HOAP-3. For that purpose, it is necessary to measure the torque applied to the robot wrists. The HOAP-3 platform has two force sensors on-board, one in each wrist. The objective is to introduce these torque measurements in simulation, and create an environment where the robot can operate freely, and interacting with objects in a stable way. As can be seen in the sequence shown in Figure 6.10 (up), the robot raises his arms until the hands touch the wall. The torques in each wrist are measured by two torque sensors included in the model of HOAP-3 in OpenHRP3, one in each wrist. The torque measurements for the right and left wrist are the ones shown in Figure 6.10 (down).

In addition to using the force sensors, the vision sensors can be included to improve the interaction with other robots or objects. These sensors allow the user to measure the depth and colour of the images captured by the robot during the simulation. In the sequence shown in Figure 6.11, the robot performs two right sidesteps, and the images received for the robot are located in the right part of every screenshot.



**Figure 6.10:** Sequence of motions for HOAP-3 touching a wall (up); Torques of right (up) and left (down) wrists during the touching motion (down)



**Figure 6.11:** Sequence of motions for HOAP-3 in front of a wall

## 6.5 Chapter summary

In this chapter, the potential of the OpenHRP3 platform has been used to create the virtual model of the humanoid robots TEO and HOAP-3, and simulate different motions.

The cart table model has been used to obtain stable motion patterns for the robots. These patterns have been tested in simulation in order to check their stability.

After this validation, experimental tests have been performed with the real robots with very good results, comparing simulation and experimental data and showing the efficient performance of the robot model in OpenHRP3.

# Conclusions and Future Works

## 7.1 Conclusions

As mentioned in the introduction, this workd has focused on the study and implementation of different dynamic models for humanoid robots in simulation. Some main conclusions derived from the work carried out here can be grouped into the following points:

- First of all, the main simplified models for humanoid robots have been presented. Moreover, the motion equations of every model have been studied.
- The selected simplified models in simulation have been implemented using Matlab and two of its tools, Simulink and SimMechanics. The required files for every model have been created and explained.
- The validation of the models in simulation has been achieved.
- The behaviour of the ankle prototype has been compared with the behaviour of the single inverted pendulum model in SimMechanics. The simulation and experimental results have not been entirely good, since the given reference has been the most appropriate for simplified robot models tests.

- The behaviour of the humanoid robot TEO has been compared with the behaviour of the double inverted pendulum model in SimMechanics. The test has been done using the lower part body of the robot (legs), and very good results have been achieved in simulation and in the experimental test.
- On the other hand, the complete models for humanoid robots in simulation have been created. To get this goal, the humanoid robot TEO and the miniature humanoid robot HOAP-3 have been used as experimental platforms. The virtual models have been modelled using the simulation platform OpenHRP3.
- Once the complete models have been modelled and simulated using the OpenHRP3 platform, some tests have been accomplished on the experimental platforms in order to compare the results in simulation and the real ones. This has been allowed to validating the complete models of humanoid robots created with OpenHRP3.

## 7.2 Future works

There are some interesting lines of research opened in this work which can be followed up in the future into the research group RoboticsLab at the Carlos III University of Madrid.

It was found that a control loop is needed in some simplified models in simulation. This is the case of the created models using Simulink, and a control loop could allow them to work properly. The control method will be selected depending on the requirements of the system. This will lead to a better control of the experimental platforms.

Once the humanoid robot TEO is fully assembled, its complete model in simulation will be tested and validated using the simulation platform OpenHRP3.

---

New steps are also being taken towards the simulation of manipulation tasks with these robots. For this purpose, a wall has been integrated in the HOAP-3 simulation environment, and the torques in the robot wrists when they get in contact with the wall are measured by torque sensors included in the model of the robot in OpenHRP3. This type of test will also be performed with TEO once the whole robot assembly process is finished.

Apart from OpenHRP3, another simulation platform could be used. Nowadays, OpenRAVE is being used as a robotic simulation platform. It is designed to be multi-platform, and many of the components are reusable. Interaction with the simulator can be performed through high level scripts in scripting environments such as Python, Octave or Matlab. The complete models of the humanoid robot TEO and the HOAP-3 could be developed and simulated using OpenRAVE, and the simulation results could be compared with the obtained ones in OpenHRP3.





# References

Albert, A., & Gerth, W. (2003). Analytic path planning algorithms for bipedal robots without a trunk. *Journal of Intelligent & Robotic Systems*, 36(2), 109–127.

Álvarez, J. (2011). *Diseño e implementación de una arquitectura para el control del robot humanoide RH-2*. Unpublished master's thesis, Carlos III University of Madrid.

Arbulu, M., & Balaguer, C. (2007). Real-time gait planning for rh-1 humanoid robot, using local axis gait algorithm. In *7th IEEE-RAS International Conference on Humanoid Robots* (pp. 563–568).

Collins, S., & Ruina, A. (2005). A bipedal walking robot with efficient and human-like gait. In *IEEE International Conference on Robotics and Automation, ICRA'05* (pp. 1983–1988).

Del Olmo, D. (2011). *Diseño e implementación de una arquitectura de control mediante lógica fuzzy para uso sobre robot humanoide RH-2*. Unpublished master's thesis, Carlos III University of Madrid.

Diankov, R. (2010). *Automated construction of robotic manipulation programs*. Unpublished doctoral dissertation, Carnegie Mellon University, Robotics Institute. Available from <http://www.programmingvision.com/rosen>

\_diankov\_thesis.pdf

Fujimoto, Y., Obata, S., & Kawamura, A. (1998). Robust biped walking with active interaction control between foot and ground. In *IEEE International Conference on Robotics and Automation, ICRA'98* (Vol. 3, pp. 2030–2035).

FUJITSU. (2005). Hoap-3 instruction manual [Computer software manual].

Garcia, M., Chatterjee, A., Ruina, A., & Coleman, M. (1998). The simplest walking model: Stability, complexity, and scaling. *Journal of Biomechanical Engineering*, 120, 281.

Gerkey, B., Vaughan, R., & Howard, A. (2003). The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the 11th International Conference on Advanced Robotics, ICAR'03* (pp. 317–323).

Geyer, H., Seyfarth, A., & Blickhan, R. (2005). Spring-mass running: simple approximate solution and application to gait stability. *Journal of Theoretical Biology*, 232(3), 315–328.

Goswami, A. (1999). Postural stability of biped robots and the foot-rotation indicator (fri) point. *The International Journal of Robotics Research*, 18(6), 523–533.

Gottschalk, S., Lin, M., & Manocha, D. (1996). Obb-tree: A hierarchical structure for rapid interference detection. In *Proceedings of 23rd Annual Conference on Computer Graphics and Interactive Techniques SIGGRAPH* (pp. 171–180).

Isidori, A. (1997). *Nonlinear control systems*. Springer-Verlag New York, Inc.

Kajita, S., Kanehiro, F., Keneko, K., Fujiwara, K., Harada, K., & Yokoi, K. (2003). Biped walking pattern generation by using preview control of zero-moment point. In *IEEE International Conference on Robotics and Automation, ICRA'03* (pp. 1620–1626).

- Kanehiro, F., Hirukawa, H., & Kajita, S. (2004). OpenHRP: Open architecture humanoid robotics platform. *International Journal of Robotics Research*, 23(2), 155–165.
- Kanehiro, F., Miyata, N., & Kajita, S. (2001). Virtual humanoid robot platform to develop controllers of real humanoid robots without porting. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'01* (pp. 1093–1099).
- Kaneko, K., Harada, K., Kanehiro, F., Miyamori, G., & Akachi, K. (2008). Humanoid robot hrp-3. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'08* (pp. 2471–2478).
- Kaynov, D. (2008). *Open motion control architecture for humanoid robots*. Unpublished doctoral dissertation, Carlos III University of Madrid.
- Kaynov, D., Soueres, P., Pierro, P., & Balaguer, C. (2009). A practical decoupled stabilizer for joint-position controlled humanoid robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'09* (pp. 3392–3397).
- Khalil, H., & Grizzle, J. (1992). *Nonlinear systems* (Vol. 3). Prentice hall.
- Koenig, N., & Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'04* (Vol. 3, pp. 2149–2154).
- Kuo, A. (2001). A simple model of bipedal walking predicts the preferred speed–step length relationship. *Journal of Biomechanical Engineering*, 123, 264.
- Kuo, A. (2007). The six determinants of gait and the inverted pendulum analogy: A dynamic walking perspective. *Human Movement Science*, 26(4), 617–656.
- Metta, G., Sandini, G., Vernon, D., Natale, L., & Nori, F. (2008). The icub humanoid robot: an open platform for research in embodied cognition. In *Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems* (pp. 50–56).

Monje, C., Martínez, S., Jardón, A., Pierro, P., Balaguer, C., & Muñoz, D. (2011). Full-size humanoid robot TEO: Design attending mechanical robustness and energy consumption. In *11th IEEE-RAS International Conference on Humanoid Robots* (pp. 714–721).

*Object management group*. (1997). Available from <http://www.omg.org>

Ogura, Y., Aikawa, H., Shimomura, K., Morishima, A., Lim, H., & Takanishi, A. (2006). Development of a new humanoid robot wabian-2. In *IEEE International Conference on Robotics and Automation, ICRA'06* (pp. 76–81).

*Openhrp3 official site*. (2008). Available from <http://www.openrtp.jp/openhrp3/en/index.html>

Park, I., Kim, J., Lee, J., & Oh, J. (2005). Mechanical design of humanoid robot platform khr-3 (kaist humanoid robot 3: Hubo). In *5th IEEE-RAS International Conference on Humanoid Robots* (pp. 321–326).

Peasgood, M., Kubica, E., & McPhee, J. (2007). Stabilization of a dynamic walking gait simulation. *Journal of Computational and Nonlinear Dynamics*, 2(1), 65–72. Available from <http://link.aip.org/link/?CND/2/65/1>

Pérez, C., Pierro, P., Martínez, S., Pabón, L., Arbulú, M., & Balaguer, C. (2009, Sep.). Rh-2 an upgraded full-size humanoid platform. In *12th International Conference on Climbing and Walking Robots, CLAWAR'09*.

Pfeiffer, F., Loffler, K., & Gienger, M. (2002). The concept of jogging johnnie. In *IEEE International Conference on Robotics and Automation, ICRA'02* (Vol. 3, pp. 3129–3135).

Pierro, P., Monje, C., & Balaguer, C. (2008). Modelling and control of the humanoid robot rh-1 for collaborative tasks. In *8th IEEE-RAS International Conference on Humanoid Robots* (pp. 125–131).

Ponticelli, R., & Armada, M. (2006). Vrsilo2: dynamic simulation system for the biped robot silo2. In *Proceedings of the 9th International Conference on Climbing and Walking Robots, CLAWAR'06*.

Sakagami, Y., Watanabe, R., Aoyama, C., Matsunaga, S., Higaki, N., & Fujimura, K. (2002). The intelligent asimo: System overview and integration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'02* (Vol. 3, pp. 2478–2483).

Shirata, S., Konno, A., & Uchiyama, M. (2004). Design and development of a light-weight biped humanoid robot saika-4. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'04* (Vol. 1, pp. 148–153).

Staroverov, P., Kaynov, D., Arbulú, M., Cabas, L., & Balaguer, C. (2007). Creating a gesture recognition system based on shirt shapes. In *International Conference on Climbing and Walking Robots, CLAWAR'07*.

Sugihara, T., & Nakamura, Y. (2003). Contact phase invariant control for humanoid robot based on variable impedant inverted pendulum model. In *IEEE International Conference on Robotics and Automation, ICRA'03* (Vol. 1, pp. 51–56).

Tanie, K. (1998). Fnr: Toward a platform based humanoid project. In *International Symposium on Robotics Research, ISRR* (Vol. 8, pp. 439–445).

*The virtual reality model language*. (n.d.). Available from <http://www.web3d.org/x3d/specifications/vrml/VRML1.0/index.html>

Vukobratovic, M., & Juricic, D. (1969). Contribution to the synthesis of biped gait. *IEEE Transactions on Biomedical Engineering*(1), 1–6.

Vukobratovic, M., Potkonjak, V., & Tzafestas, S. (2004). Human and humanoid dynamics. *Journal of Intelligent and Robotic Systems*, 41(1), 65–84.

- Whittington, B., & Thelen, D. (2009). A simple mass-spring model with roller feet can induce the ground reactions observed in human walking. *Journal of Biomechanical Engineering*, 131, 011013.
- Witkin, A. (1997). Physically based modeling: Principles and practice constrained dynamics. *SIGGRAPH Course notes*, 11–21.
- Yamaguchi, J., & Takanishi, A. (1997). Development of a leg part of a humanoid robot - development of a biped walking robot adapting to the humans' normal living floor. *Autonomous Robots*, 4(4), 369–385.
- Yamane, K., & Nakamura, Y. (1999). Dynamics computation of structure-varying kinematic chains for motion synthesis of humanoid. In *IEEE International Conference on Robotics and Automation, ICRA'99* (pp. 714–721).