Thesis

Paulina Aguirre

October 2010

Contents

List of Figures

List of Tables

Knowlegements

Abstract

Chapter 1

Introduccion

1.1 History of mobile legged Robots

Despite of the obvious benefits legged machines, their development in practical and industrial areas has been almost insignificant. The lack of legged machines employed to perform real work is not due to a lack of prototype development. Is quite surprising the fact that for the past 40 years there have been hundreds constructed, from lumbering polypeds to hopping monopeds. The General Electric Walking Truck constructed by Mosher[16] in 1968, was a polyped machine which carried an operator responsible of controlling twelve servo loops that controlled the legs. Weighing 1400kg, required an external power source for its hydraulic actuation. In the early 1980s Odex Inc. constructed the Odex-1, one of the largest legged machines in Ohio State's hexapedal, hydraulicly actuated Adaptive Suspetion Vehicle.

In contrast to the Mosher vehicle, it used a digital feedback control to avoid the need of an operator. Finally, another pioneering machine is the Raibert's dynamically balancing monoped hopper. Built on the early 1980s at the MIT leg lab the prototype, it was the first machine that exhibited a dynamic balance.[20]. Raibert showed that certain legged machines dynamically stable locomotion could be achieved with simple control actions. Figure 1.1 shows these tree polypedal robots.

Even though of all this success in legged machines, they still are not common in industrial applications and every day's life. The main factor contributing to its slow development is the difficulty for achieving energy efficiency and stability, both basic attributes for an automous vehicle.[26]. Automous vehicle energy efficiency is traduced in the capability for traveling further and longer with a low consuming rate of energy. In the modern automobile industry this can be achieved by better aerodynamic design, lighter materials for its structure, and better controllers for the combustion of their engines. For legged robots energy efficiency could be achieved by improved mechanical design, lighter materials and a strong control law that designs a gait that assurances an efficient locomotion.

As Efficiency is important, Stability is too. When a vehicle overturns may damage itself and whatever it falls onto. To maximize stability, means minimize the chance of overturning by using suspension components that maintain the

Figure 1.1: Pioneering legged machines. General Electric Walking Truck, Odex-1 and Raibert's hopper

wheels in contact with the driving surface, break system to prevent side-skidding and wheel slippage. Analogous, for legged machines may be designed to have morphologies that improve stability, bigger feet and number of legs increased. As well, the role of control algorithms for designing gaits that could be stable by slowing the motion minimizing inertial effects.[14]

Biped Robots, over the last years have rapidly developed, including prototypes, control algorithms and analysis of gaits. In this section a review of biped robot prototypes and controlling techniques will be given. All the literature will be divided into categories: analysis of passive walking-when gravity alone powers the walking motion- and the analysis of non-passive walking-walking that requires an external power source.

Passive walking is motivated to by the drive for energy efficiency, where dissipation due to impacts or damping is offset by the use of potential energy supplied by walking down a slope.[26] Research in passive waking stars with McGeer in the late 1980s [18]. In his work, McGeer built a four-link planar passive walker and performed a detailed parameter variation and stability analysis. The mechanism was designed in such way, that from preventing leg collapse he locked knees and for imposing rolling ground contact he added circular feet to the model. It weighed 3.5 kg, was 0.5m tall, and could stably walk down a 1.4 degree slope at about 0.4 m/s. In the late 1990s Goswami, Espiau and Keramane[2] showed the so-called compass gait walker, a two-link planar passive walker with prismatic legs.

At the end of the 1990s, Collins built a three-dimensional version of McGeer's passive walker. The walker with carefully designed feet and pendular arms was able to walk down a 3.1 degree slope at about 0.5 m/s. It weighed 4.8 kg and measured 0.85 m in height, see Fig 1.2 [1]. Recently Adolfsson, Dankowicz and Nordmark [11] studied a passive, three-dimensional model which started with McGeer's planar model and gradually became in a ten DOF, three-dimensional model. By this work, stable gaits of the three-dimensional model where found.

In the recent years Japanese have lead the effort for in developing non-passive

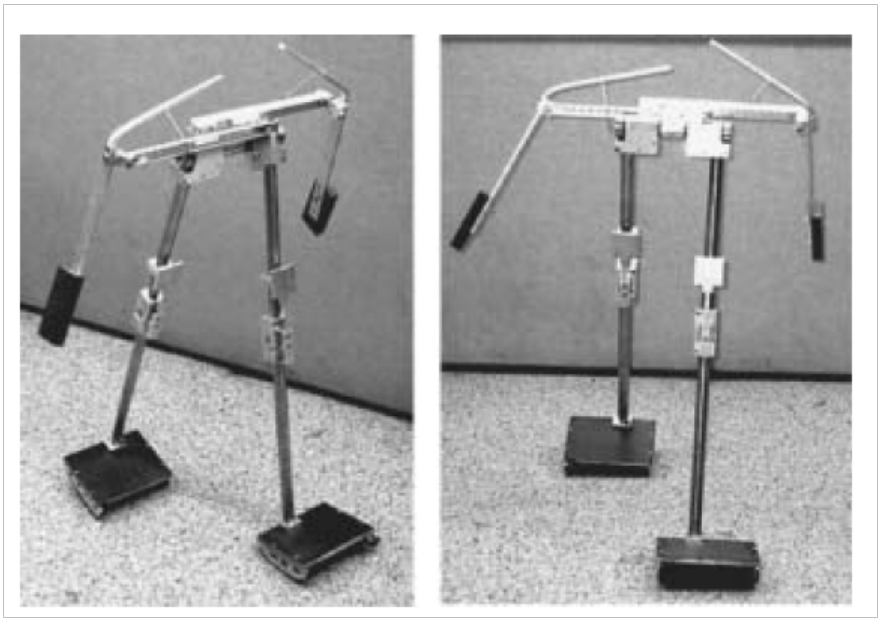

Figure 1.2: Collins two legged, kneed, passive dynamic walking robot

biped robots. The first reported biped capable of walking is the WL-5, a three dimensional, 11-DOF walker constructed at the Waseda University in Japan in 1972. By the mid-1980s the same laboratory developed WL-10RD, a three-dimensional 12-DOF walker. In the late 1990s, Pratt, at the MIT Leg Lab, built a planar, seven-link walker with no feet named Spring Flamingo. It weighed 14 kg and measured 1.2m in height, being capable of walking 1.2 m/s, transversing sloped terrain.[19] In the Technical University of Munich a three-dimensional walker named Jonnie, has been developed. Able to walk 0.4 m/s, weighs 40 kg and has 1.8m in height [8].

In late 1990s, the French National Research Council constructed RABBIT, a planar walker weighing 32kg and measuring 1.2m in height. Rabbit is the prototype on which this thesis is based. Other prototypes inspired on the design and the general morphology of Rabbit and its control theory are MABEL assembled on spring 2008 from the University of Michigan, capable of walking uneven terrains and ERNIE from the Ohio State University. ERNIE was built during the period of September 2005 to January 2006. ERNIE's legs are modular, which means the leg lengths, the leg ends, and the joint offsets may be changed with minimal redesign.[14].

Finally the advanced Japanese prototypes are presented. The Waseda University in 2006 has developed WABIAN-2R. WABIAN-2R is a 41 DOF, weighing 64 kg, measuring 1,5, capable of walking at 0.21 m/s.[25]. One of the more famous bipeds of history and to the date is ASIMO (standing for Advanced Step in Innovation Mobility) developed by the Honda Corporation. ASIMO is an autonomous three-dimensional walker with 26-DOF weighing 43 kg and measuring 1.2m in height and its capable of walking 0.3 m/s on the level ground and climbing and descending stairs. The development has involved ten generations of prototypes, named E0 through E6 and P1 through P3, and has cost hundreds

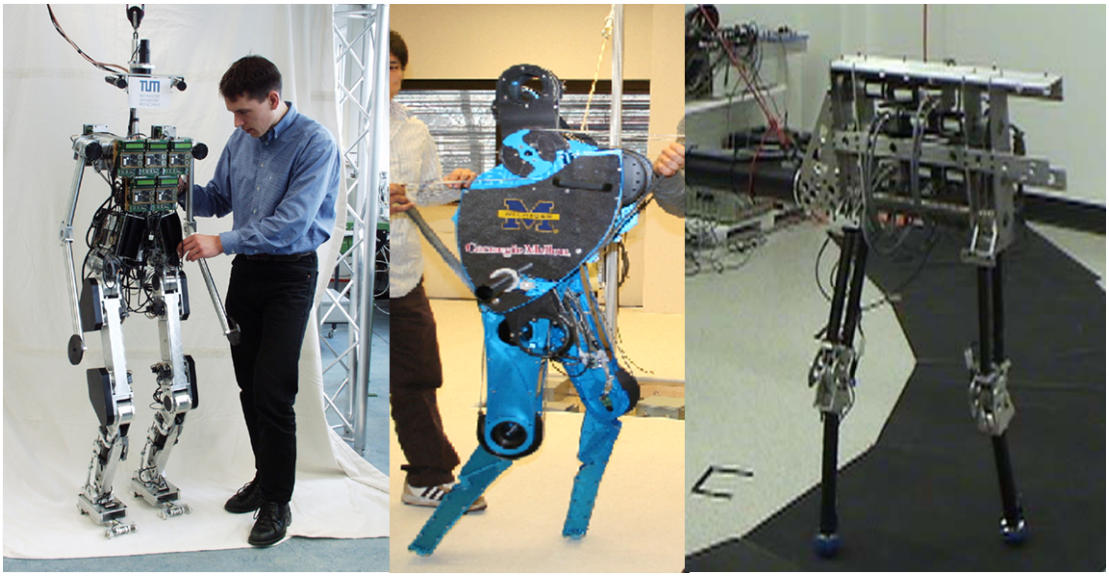

Figure 1.3: Three Biped Robots, Jonnie, MABEL, ERNIE

of millions of dollars. [4]

Following ASIMO, the Japanese government began the Humanoid Robot Project (HRP). In 2002 the project produced HRP-2, a three-dimensional, 30-DOF biped weighing 58kg and measuring 1.54 m in height. Figure 1.4 shows these three impressing prototypes.[HRP-2]

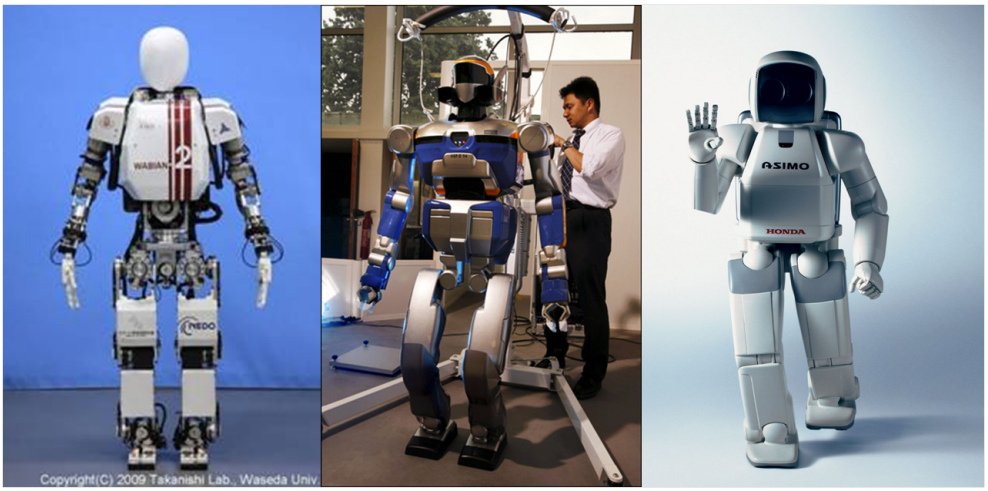

Figure 1.4: Japanesse Robots: WABIAN 2R, HRP-2,ASIMO

1.2 Introduction

Locomotion is not only the ability that has a body to move from one place to another, but also an essential characteristic of animal life. The lack of it represents a incapacity for nourishing, avoiding predators and mating. Animal locomotion ranges from swimming, crawling and flying, to walking and running. The means of locomotion are the result of a successful adaptation of an organism's morphology and scale to its environment, thanks to evolution.

The same idea could be applied to the non-biological scheme. Human beings have created numerous machines that are perfectly adapted to its environment, such as airplanes for the air, submarines and boats for water, cars and tanks for uneven surfaces on the earth. In these machines, robots are included as well. They have been developed to achieve different tasks that often depend on their interactions with the environment for which they had been designed. Bipedal, quadrupedal and other walkers, compose a class of mobile robots. The superiority of these robots over robots with tires and chains deals with the fact that legs enable the avoidance of support of discontinuities like a rocky slope or a flight of stairs, just by stepping over them.

The flight phase is the period of time while a robot has one leg on the air; this allows it to overcome obstacles, such as stairs and insurances it's ability to deal with irregular terrain. In addition to this, walking robots have as well a double support phase the same that occurs when all of their feet are in contact with the floor. It is important to note that the robot has a different nominal operation for walking, running and jumping. [21] Since their locomotion is composed of the alternating of these two phases, their dynamic behavior is very complex that leads to differences between running, jumping and walking.

Despite these difficulties, walking robots arouse a lot of interest in the research of robotics, modelization problems, trajectory generation and new control methods. Moreover, their application fields include biomechanical and medical fields holding a potential interest to the restoration of motion in the disabled (dynamical controlled lower-limb prostheses, rehabilitation robotics and functional neuronal stimulation), as well as in the domain of dangerous activities such as nuclear task and space exploration.

1.3 Biped Basics

A biped is an open kinematic chain consisting of two sub chains called *legs* and often, a sub chain called the *torso*, all connected at a common point called the *hip*. The end of each leg will be referred as a FOOT, even if it doesn't have a link that actually constitutes a foot. During the gait it is possible that one or both legs are in contact with the ground, when one leg is in contact with the ground it could be referred as *stance leg*, while the other that is not under contact is called the *swing leg*. During the gait, two locomotion phases are present. The *single support* or *swing phase* when just one foot is on the ground and the *double support phase*, where both feet are in the ground. Finally *walking* is defined as alternating phases of single and double support, assuming that the feet are not slipping when in contact with the ground. [14]

At figure 1.5 the single support phase is shown in (a) and (b), while the double support is depicted in (c). Assuming that all the joints are fully actuated

and there is no slipping in the floor and comparing the number of degrees of freedom to the number of independent actuators in the same locomotion phase, reveals that during the phase (a) the robot is fully actuated, under actuated in (b) and over actuated in (c).

As shown in 1.6 the bipedal walking with point feet has a single support phase is shown in (a), while the double support is depicted in (b). If the feet are not slipping and the joints are fully actuated, then the robot is unactuated in (a) and overactuated (b).

At figure 1.5 the single support phase is shown in (a) and (b), while the double support is depicted in (c). Assuming that all the joints are fully actuated and there is no slipping in the floor and comparing the number of degrees of freedom to the number of independent actuators in the same locomotion phase, reveals that during the phase (a) the robot is fully actuated, under actuated in (b) and over actuated in (c).

As shown in 1.6 the bipedal walking with point feet has a single support phase is shown in (a), while the double support is depicted in (b). If the feet are not slipping and the joints are fully actuated, then the robot is unactuated in (a) and overactuated (b). For explaining what a planar biped like Rabbit is, we first will introduce basic notions about the human body planes. Indeed the human body could be divided 3 planes of section which are:

- The *sagittal plane* which is the longitudinal plane that divides the human body into a left and right section
- The *frontal plane* perpendicular to the sagital plane, separates the body into front and back portions.
- The *transverse plane* Perpendicular to both sagital and frontal plane.

See Fig 1.7 for an illustration of these planes of section. According to these definitions, a *planar biped* could be defined as a biped which has motions just in the sagital plane. Our subject of study, Rabbit is a planar biped since its motions are constrained to the sagital plane by a boom which is attached to its hip.

On the other hand a *three dimensional walker* has motions taking place in both the sagital and frontal planes. A mechanical system is *fully actuated* when the number of independent actuators equals the number of degrees of freedom.

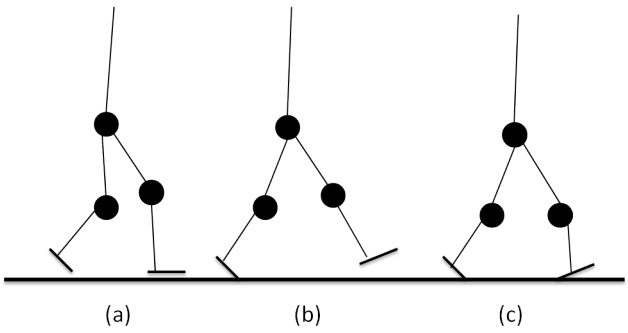

Figure 1.5: Various phases of bipedal walking with non point feet

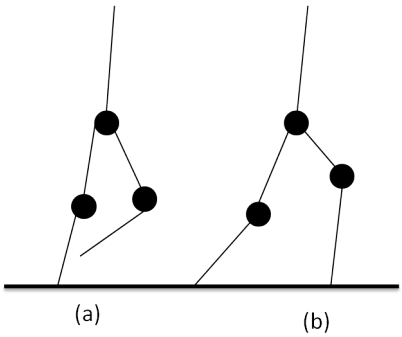

Figure 1.6: Phases of bipedal walking with point feet

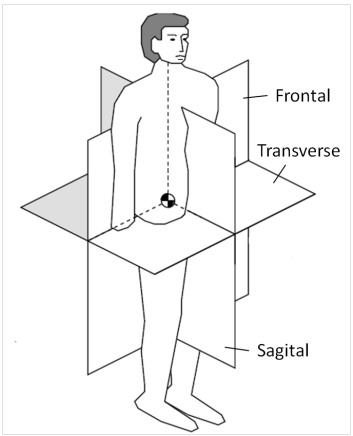

Figure 1.7: Planes of the human body

If there are fewer actuators than degrees of freedom then it is *underactuated* , and if there are more actuators than degrees of freedom, it is *overactuated* [6]. For a planar robot like Rabbit, for the single support phase (a) of **??** it has 4 active actuators and 7 DOF (the possible movements of the robot), which means is underactuated. During the double support phase (b) the robot has the same number of active actuators and 4 DOF, this means that a fully actuated system .

A *statically stable gait* is a periodic locomotion in which the center of mass of polypedal mechanism COM (center of Mass) does not leave the *the support polygon*, see Fig 1.9 for an illustration a support polygon. A support polygon is a convex frame formed by all the contact points in the ground. The *center of pressure* (CoP) is defined as the point on the ground where the resultant of the ground-reaction force acts[2]. Bipeds require a dynamically stable gait, since a *dynamically stable gait* is a periodic gait assures the biped's CoP is on the boundary of the support polygon for at least part of the cycle and yet the biped does not overturn.[14]

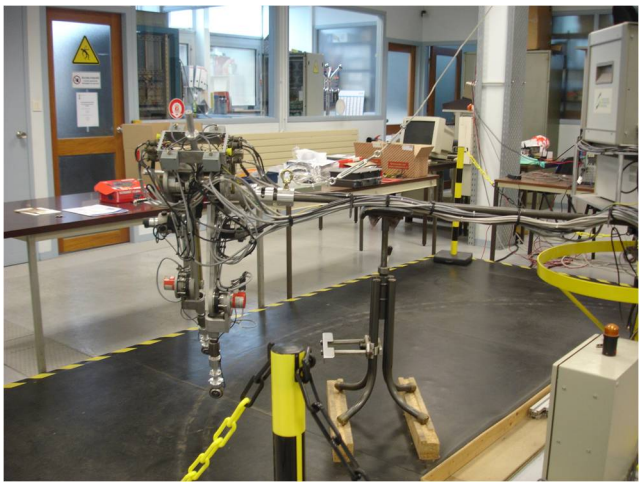

Figure 1.8: Rabbit, a planar biped

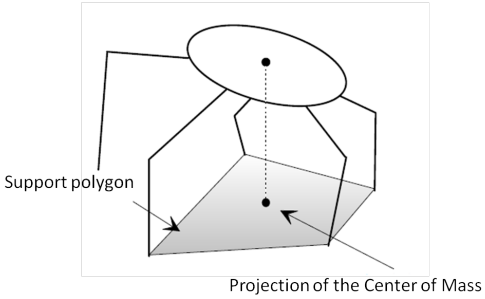

Figure 1.9: Support polygon for a polypedal mechanism

1.4 Challenges of Bipedal Locomotion

One of the most difficult taks inherit to bipedal locomotion is designing a control algorithms capable of coping with underactuation of the biped systems, limb coordination and common difficulties for bipedal walking.

1.4.1 Common Difficulties

Limb coordination: Despite that the task of walking is inherently a low DOF task, which is the transportation of the robot's center of mass from one point to another, bipeds are typical high degree of freedom (DOF) mechanism. Consequently, walking does not only specify how the limbs must be coordinated in order to achieve a desired displacement of the robot's center of mass.

Hybrid dynamics: The presence of impacts and the varying nature of the contact conditions of the leg ends through a walking cycle, causes that models have multiple phases and hence are hybrid. In this dissertation, the studied biped is a hybrid. Control theory for hybrid systems is just

now being developed[14].

Effective underactuation: As presented in the previous section, during the phase of single support or swing phase, the robot is underactuated. Unlike most of robot manipulator that are fixed to the ground, a biped is designed to move with respect to the environment. Because of finite foot size, a large torque supplied at the ankle joint may result in foot rollover; in which case the robot is underactuated.

Unilateral Constraints: Since the ends of the robot's legs whether they are terminated with feet or points are not attached to the walking surface, normal forces at the contact points can only act in one direction and hence are unilateral. As well in order for the foot not to slip the ground reaction forces must satisfy $\left|F^T\right| \leq \mu_s \left|F^N\right|$, where μ_s is a given coefficient of static friction, F^N the normal force and F^T the tangential force. This can be expressed in multiple unilateral constraints. Moreover, if there is a foot that must remain flat in the ground and not rotate about its extremities, there must be a point between the heel and toe where the moment on the foot is zero. [14]

1.5 Contributions

This dissertation tries to build a simulator of a biped Robot, entirely based on the prototype Rabbit. The main contributions for this thesis are the work of Eric Westervelt and his design of a feedback controller that achieves exponentially stable walking motions in Rabbit. His work was proved and validated in the physical prototype of Rabbit which is located in the Laboratory of Automatic of Grenoble, as his PHD thesis work. Moreover, the work of Laurence Roussel inspired the implementation of the model in Multibody representation.

The main objective of this dissertation is contributing to the long and existing project of Rabbit, with a Multibody model of Rabbit that is suitable for simulation and controlling purposes. This model would be implemented under mathlab syntax allowing it to be charged in a Simulink block for simulation. Since a biped robot, needs the adequate applied torques on its actuators for reaching an stable gait, the trajectory generation and control blocks of the project were provided under permission of the GIPSA Lab (Grenoble Image Parole Signal and Automatique Laboratory) for this project, for testing the liability and performance of our Multibody model. The control and trajectory generation blocks or Rabbit, were implemented under mathlab and Simulink syntax, and these are currently successfully operating on the real prototype, since the main objective this dissertation is proving the proposed Multibody model of Rabbit, this blocks were included in the simulator with its respective modifications for working in a software mode instead of the a hardware mode which they were initially implemented for.

1.6 Organization of the dissertation

In this section we are presenting a structure of this dissertation. The first chapter is the Introduction, which has been already presented. This chapter includes

the history of legged robots, biped basics, challenges of biped locomotion and contributions. Chapter 2, main objective is presenting a description of the biped robot Rabbit, the working prototype on which this dissertation was based. This chapter includes Rabbit's history, a mechanical description of the prototype, including its former actuators and sensors, its walking surface and its graphical user interface. Chapter 3 describes the general background theory of Multibody Systems and the Multibody model of Rabbit as a 7 DOF system. Chapter 4 describes the external forces acting in the feet of Rabbit. The chapter main objective is to describe the non-linear model of normal force and the dynamical friction coefficient that composes the tangential force, acting at every moment of the impact over Rabbit's feet. Chapter 5 describes a general overview of the control system of Rabbit and its background theory. This includes virtual constraints, zero dynamics, feedback controllers and the trajectory generation for the physical prototype. Chapter 6 presents the results obtained in this dissertation and Chapter 7 presents the conclusion and further work of this dissertation. In addition to this, Appendix A gives the practical implementation of the model in Robotran and the computational generation of the kinematic and dynamical equation of the model in mathlab syntax. Finally, Appendix B describes the implementation of Rabbit's model in Simulink.

Chapter 2

The Prototype Rabbit

Rabbit is a five-link planar prototype, which is located at the Laboratoire D'Automatique de Grenoble in Grenoble, France. It was constructed jointly by several French research laboratories, spanning Mechanical Engineering, Automatic Control, and Robotics. The RABBIT project was initiated in 1997 and is funded by the French CNRS (Centre National de Recherché Scientifique) and the French National Research Council. The central mission of the project, was to build a prototype for studying truly dynamic motion control [7]. The mechanism was designed to allow for high-speed walking and running.

2.1 Description of the prototype

The prototype is composed of five links which are connected by revolute joints that form two symmetric legs and torso; see figure1 1

Figure 2.1: Knee Actuators

Actuators supply torque between each of the four internal joints: one at each knee and one between the torso and each femur. All actuators are dc motors with Samarium Cobalt magnets, capable of producing a peak torque of 150 Nm. A gear reducer and belt were used to connect the motors to each of the four actuated joints. The motors of the knees were mounted as close as possible to the hips to minimize the inertia of the legs; this decreases the coupling in the dynamic model as well as th require motor torques. [6] To obtain configuration information, encoders are located at each internal joint giving the robot's shape, between the boom and hip giving the robot's orientation with

Figure 2.2: Binary contact switches

respect to a world frame. Binary contact switches located at the leg ends are used to detect whether or not a leg is in contact with the walking surface. See figure 2.2

For a real-time control platform, RABBIT uses a dSPACE DS1103 system. With the DS1103 system, run-time software is created by automatic translation and cross-compiling of Simulink diagrams for the system's 400 Mhz PowerPC 604e DSP, allowing real-time controller software to be developed in a high-level language.

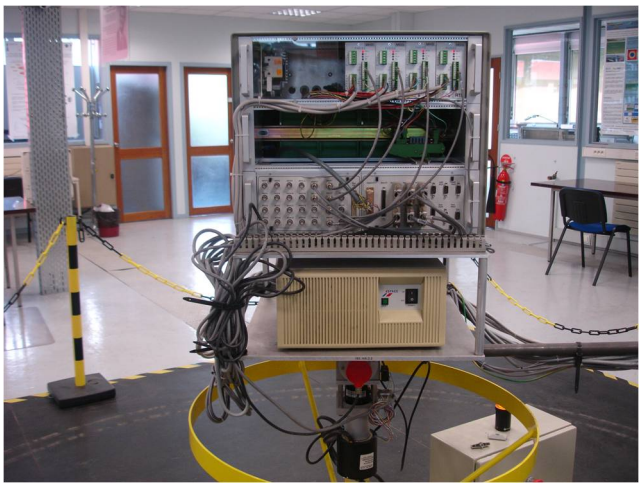

Figure 2.3: dSPACE DS1103 system

This enables debugging facilities and obviates the need for low-level I/O programming. In addition to this, the system provides digital-to-analog and analog-to-digital conversion as well as a user interface. See 2.3

Rabbit is made up of two legs and a trunk. Both legs are composed of a

femur and tibia respectively, having the same lengths each one. This fact makes the model symmetrical with respect to the torso. In table 2.1 gives the model parameters of each part of the robot. In addition to this, figure 2.4 and 2.5 shows the measurement conventions of the torso and the a leg the legs respectively, where p_*^M, L_*, u_1, u_2. are the position of the center of mass, the length of each articulation and coordinates in the Cartesian plane of the center of mass of each articulation.

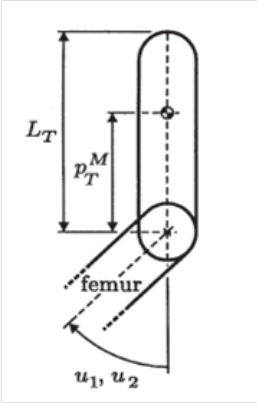

Figure 2.4: Schematic Torso with measurement conventions

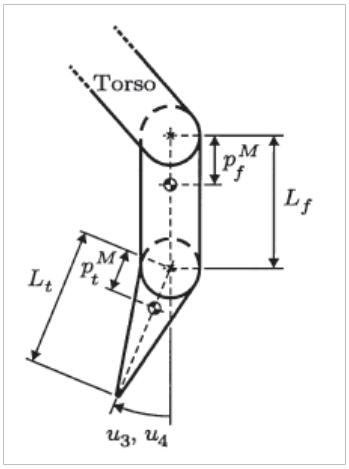

Figure 2.5: Schematic of leg with measurement conventions

2.2 Constraining RABBIT to be Planar

The boom attached to Rabbit's hip constraints Rabbits's motion to the sagittal plane and constraints the sagittal plane to be tangent to a sphere centered at the universal joint that connects the boom to the center stand (see figure

The typical means for constraining a biped robot's motion to be planar is through the use of wide feet[7], but the advantage of a boom system over wide

Model Parameters	Torso (T)	Femur (f)	Tibias (t)
Mass, M_*(kg)	20	6.8	3.2
Length, L_*(m)	0.625	0.4	0.4
Inertia, $I_*(m^2kg)$	2.22	1.08	0.93
Mass center, p_*^M(m)	0.2	0.163	0.128

Table 2.1: Model Parameteres

feet is that a boom is able to constrain the robot's motion even when none of the feet is on the ground. This is important in the case of Rabbit, because one of its purposes is to study running, which necessarily has a flight phase (when no feet are in contact with the ground) [7]. For connecting power and communications cabling to the experimental setup, a slip ring was installing for avoiding the cabling connected to the experimental setup will become twisted or wound as the robot makes laps.

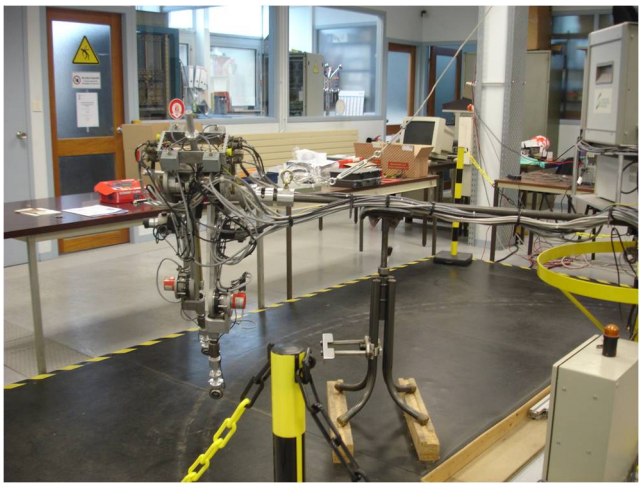

Figure 2.6: Experimental setup of Rabbit

Model Parameter	Unit	Label	Value
Constraint boom length	m	l_b	1.5
Hip to stand distance	m	$l_{b,1}$	1.4
Stand height	m	l_s	1.4
Constraint boom mass	kg	m_b	5.0
Cable mass	kg	m_c	2.0
Counterbalance mass	kg	m_w	0.0
Support electronics mass	kg	m_e	20.0

Table 2.2: Rabbit's Experimental Platform Parameters

The inertia of the boom system used to constraint Rabbit's motion to be planar results into an additional inertia, which is significant important for including its effects on the model. The inertia has four components due to (1)

the boom connecting Rabbit, (2) the counterbalance, (3) the cabling connecting Rabbit to the support electronics, and (4) the support electronics. The inertia can be approximated as:

$$I_s = \frac{1}{3} m_b l_b \left(l_{b,1}^3 + l_{b,2}^3 \right) + m_w l_{b,2}^2 + 13 m_c l_{b,1}^2 \tag{2.1}$$

$$I_e = \frac{1}{12} m_e l_e^2 \tag{2.2}$$

This results in additional kinetic energy:

$$K_{support} = \frac{1}{2} I_{support} \left(\dot{\phi_h^2} + \dot{\phi_v^2} \right) + \frac{1}{2} I_{electronics} \dot{\phi_h^2} \tag{2.3}$$

where ϕ_h and ϕ_v are the horizontal and vertical angular displacement of Rabbit about the center stand, these angles can be approximated like:

$$\phi_h \approx \frac{p_H^h(q) - p_H^h(q_0)}{l_{b,1}} \tag{2.4}$$

$$\phi_v \approx \frac{p_H^v(q) - p_H^v(q_0)}{l_{b,1}} \tag{2.5}$$

Where q_0 is Rabbit's configuration at the beginning of the step and p_H^h and p_H^v are the horizontal and vertical position of the hip.

The potential energy due to the boom, the counterbalance and the cabling is:

$$V_a = \frac{1}{2} g_0 \frac{m_b}{l_b} \left(l_{b,1}^2 + l_{b,2}^2 \right) - g_0 m_w l_{b,2}^2 \sin(\phi_v) + \frac{1}{2} g_0 m_c l_{b,1}^2 \sin(\phi_v) \tag{2.6}$$

The counterbalance mass was chosen to negate the potential energy due to the boom and cabling. In all of the experiments no counterbalance was used, the required counterbalance could not be securely fastened to the boom because of the short length of $l_{b,2}$.

The length of the boom determines the ability to counterbalance the boom. Moreover, the longer the boom, the better the approximation of Rabbit as a planar mechanical system, however, the longer the boom, the greater the dynamic effects of the additional kinetic and potential energies[7].

2.3 Gears Reducers and Joint Friction

For allowing lighter weight motors use, Rabbit has gear reducers between its motor and links. The gear reduces have two important effects over Rabbit's dynamics. The first effect is to add significant joint friction and the second effect is to approximately decouple the robot's dynamics leaving the only significant inertial load on the motor [14]. They were considered in the control implementation. The joint friction was modeled by viscous and static friction terms,

$$F(q, \dot{q}) := F_v \dot{q} + F_s sgn(\dot{q}) \tag{2.7}$$

where $F_v = (F_{v,H}, F_{v,H}, F_{v,K}, F_{v,K},)$ and $F_s = (F_{s,H}, F_{s,H}, F_{s,K}, F_{s,K},)$. The frictional parameters of Rabbit are given in table 2.3.

Model Parameter	Unit	Label	Value
Viscous friction	Ns	$F_{v,K}$	5.48
		$F_{s,H}$	15.0
Static friction	Nm	$F_{s,K}$	8.84
Gear Ratio	—	n_g	50.0
Motor rotor inertia	m^2kg	I_a	0.83

Table 2.3: Frictional parameters

2.4 Walking Surface

The floor on which Rabbit walks is concrete with 30 cm wide cabling access trenches covered with 4mm steel plates. To help make the walking surface uniform, the floor was covered with 1.5 cm particle board, which was then covered with a layer of 3 mm rubber. Besides making the walking surface uniform, the rubber layer was added in the hope of extending the life of Rabbit by providing a modest amount of compliance.

Chapter 3

Multibody Approach of Rabbit

3.1 Main Definitions and conventions

A *mutibody system* (MBS) is a set of N^{body} rigid bodies interconected by joints.[13]. *Joints* are physical devices that connect two contiguous and rigid bodies at *connecting points*, also called *attachment points*. In addition to this, at least one joint of the system should be connected to an inertial fixed body called the base.[13] Figure 3.1 shows as multibody system and its joints.

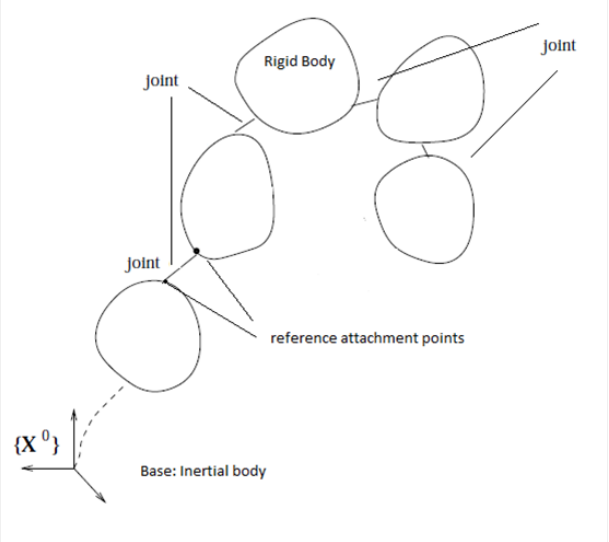

Figure 3.1: A Multibody System.

Concerning system topology, a multibody system could have a *tree-like structure* (see 3.4(a)) and *closed-loop structures* (see 3.4(b)). For instance in a tree structure, independent movements are possible in every joint. On the other hand in closed-loop structures, the presence of loops causes that the movement

in the involved joints are restricted. This is caused by the assumption that bodies in the structures are considered as rigid and therefore non deformable.

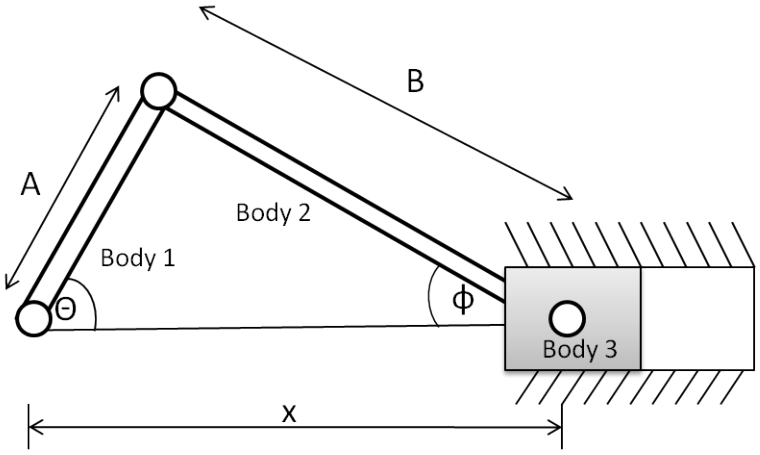

Figure 3.2: The slider-crank mechanism

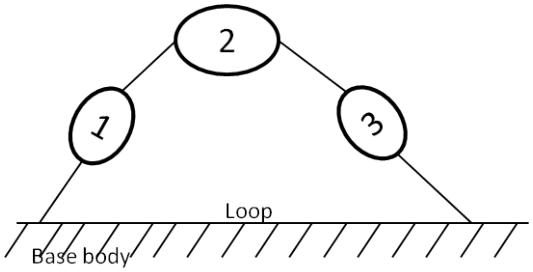

Figure 3.3: Reference Configuration of the slider-crank mechanism

The slider-crank mechanism is an example of this kind of structure(see 3.2 and 3.3 for its reference configuration). It presents a mathematical relation that expresses this movement limitation, which is known as a restriction [1]. For example, the restrictions for the slider-crank mechanism are:

$$x = A\cos\theta + B\cos\phi \tag{3.1}$$

$$0 = A\sin\theta - B\sin\phi \tag{3.2}$$

A common approach for close-loop structures, consist in considering them as an open-loop structures with additional restrictions. Therefore, a multibody will be always expressed like a tree, with its bodies as *nodes*, its joints as *branches* and the base body as the root of tree. Concerning a *tree-like structure*, denoting N^{body} the number of bodies and N^{joint} the number joints, only if all joints have 1 DOF, it is possible to have $N^{body} = N^{joint}$. Moreover, there is only one *path* (sequence of branches) to go from the base to any other body [22].

[1] A restriction is a mathematical relationship between variables, which characterizes the system behavior

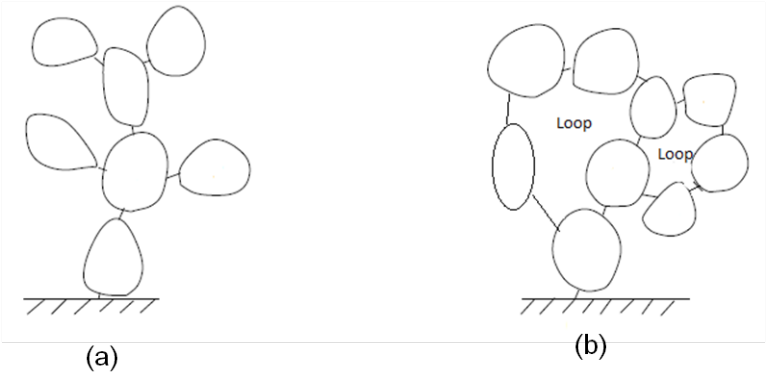

Figure 3.4: Tree-like structure and closed-loop structure

In this dissertation just tree-like structures will be considered, due to the fact that Rabbit has a tree-like structure.

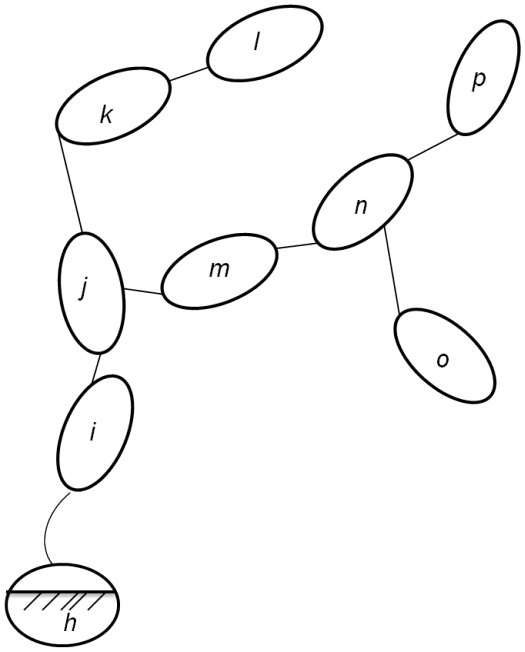

Figure 3.5: Illustration of filiation concepts

A *kinematic chain* is an ordered set of several interconnected bodies without making a loop, which usually stars from at a reference body to end at a terminal body. Being precise the reference body can be any body of the structure and the terminal body doesn't needs to be a leaf body of the structures. A leaf body is a terminal body of the tree-like structures, which has no children. In figure 3.5 $\{l, p, o\}$ are leaf bodies of the system. In figure 3.5 $\{i, j, k\}$, $\{l, k, j, m, n\}$ represent kinematic chains. Referring to the previous definition, it is possible to use some filiations concepts inherited from the family tree representation:

- *Base body :* an inertial reference fixed body
- *Ancestor :* the base is the only possible ancestor of all the bodies belonging to the structure. Figure 3.5 shows that the body i is an ancestor of the body k at the kinematic chain that starts from the base to the body k (excluded) and contains body i.
- *Descendant :* Since i is an ancestor of body k, then k is the descendant of body i. This is possible only if the kinematic chain going from the base to the body k (excluded) contains body i.
- *Parent or direct ancestor :* Among of all the possible ancestors of body k, j is the only possible direct ancestor or parent of body k. As shown in Fig 3.5, body j is the only body directly connected to body k. Moreover, in a tree structure connected to a base each body can have only one parent.
- *Child or direct descendant :* body j is called child of body i, if body$i$ is the parent of body j. In a tree-like structure a child has only one parent as shown in Fig 3.5, but a parent can have more than one child.
- *Leaf body :* since it is the terminal body of the tree-like structure; a leaf body has no children. In the figure 3.5 bodies *l,p,o* are leaf bodies.

3.1.1 Topology

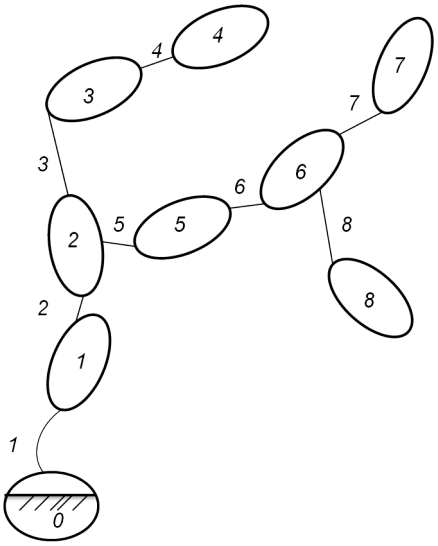

Figure 3.6: Illustration of filiations concepts

For computer implementation it is easier to refer to bodies and joints by means of indexes [22]. For this reason bodies and joints can be numbered in ascending order, starting from the base (index 0) to the leaf bodies. Moreover, joint which precedes a body receives the same index as this body in the tree structure. This is illustrated in figure 3.6.

A topological vector or *inbody vector* is defined as the vector whose i^{th} element contains the index of the parent of body i. For example, in figure 3.6, *inbody=[0 1 2 3 2 5 6 6]*.

3.1.2 Frames and vectors

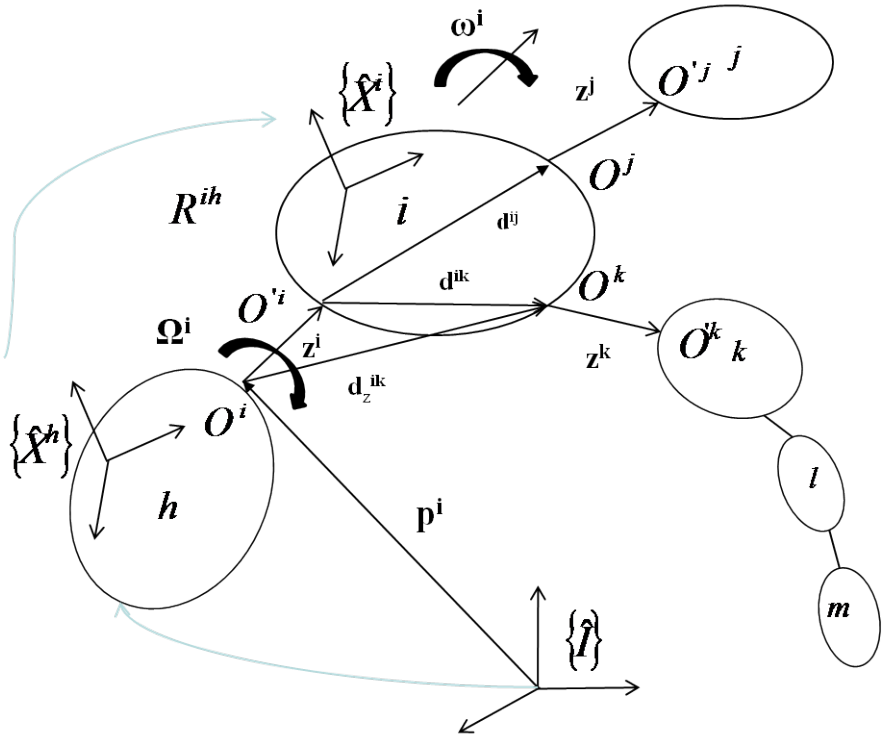

Figure 3.7: Frames and vectors definitions

Using figure 3.7 important frames and vectors can be defined below:

- O^i, O'^i, are the reference attachment points of the joint i with its parent body h, and its child i, respectively.
- $\mathbf{z}^i$, the relative position vector $O^i\vec{O}'^i$, represents the relative displacement in joint i.
- $\mathbf{d}^{ik}$ is the position vector of the attach point O^k of joint k with respect to O'^i, where body k is the child of body i. This vector on body i, represents the contribution of body i to the kinematic chain which links body k to the base.
- $\mathbf{p}^i$, is the absolute vector of the attach point O^i
- Finally, the extended position vector of the joint k can be defined like this:

$$\mathbf{d}_z^{ik} \equiv \mathbf{z}^i + \mathbf{d}^{ik} \tag{3.3}$$

The orientation of the bodies is described by the following quantities:

- $\left\{\hat{I}\right\}$, the inertial frame which is fixed to the base(body 0). It is composed by three base vectors $\left\{\hat{I}\right\} = \left\{\hat{I}_1, \hat{I}_2, \hat{I}_3\right\}$
- $\left\{\hat{X}^i\right\}$ is the moving frame rigidly attached to the body i and located in the center of mass G^i. When expressed in this frame, the geometrical vectors $\mathbf{d}^{ij} = \left[\hat{X}^i\right]^T d^{ij}$ and $\mathbf{d}^{ik} = \left[\hat{X}^i\right]^T d^{ik}$ have constant components d^{ij} and d^{ik} since the bodies are rigid.
- $R^{i,h}$ is the rotation matrix, such that $\left[\hat{X}^i\right] = R^{i,h}\left[\hat{X}^h\right]$
- $\mathbf{\Omega}^i$, the relative angular velocity vector of the body i with respect to frame $\left\{\hat{X}^i\right\}$ fixed on body i.
- $\left[\hat{X}^i\right]$ refers to a column array containing the unit vectors of the frame

$$\left[\hat{X}^i\right] = \begin{bmatrix} \hat{X}_1^i \\ \hat{X}_2^i \\ \hat{X}_3^i \end{bmatrix} \tag{3.4}$$

- A vector $\mathbf{v}$ can be expressed in the body-fixed frame as

$$v = [\hat{X}^i]^T \upsilon = [\hat{X}^i]^T \begin{pmatrix} \upsilon 1 \\ \upsilon_2 \\ \upsilon_3 \end{pmatrix} \tag{3.5}$$

$$v = [\hat{X}^i]^T \upsilon = \upsilon_1[\hat{X}_1^i] + \upsilon_2[\hat{X}_2^i] + \upsilon_3[\hat{X}_3^i] \tag{3.6}$$

3.1.3 Joints

Joints are mechanical connection devices that connect two rigid bodies of a multibody structure. The relative motions allowed by a joint define its number of *relative degrees of freedom* (relative d.o.f.)[22].

The set of physical joints can have two elemental cases: the *prismatic* and *revolute* single-degree-of-freedom, as shown in figure 3.8. Therefore, any joint that has two degrees or more degrees of freedom could be modeled as an appropriate succession of prismatic and revolute joints. [13]

Generalized coordinates (q) are variables which describe the relative motion of the joints (linear and angular displacements, respectively for prismatic and revolute joints). Hence, for each joint i $\left(i = 1, ..., N^{body}\right)$ a generalized coordinate q^i will be associated with it and will represent:

- The amplitude of the *relative displacement* ξ^i of O^i with respect to O'^i measured along the unit joint vector $\hat{\mathbf{e}}^i$, if joint i is prismatic.

  See figure 3.8(a)

$$\boldsymbol{\xi}^i = O^i\vec{O}'^i = q^i\hat{\mathbf{e}}^i \tag{3.7}$$

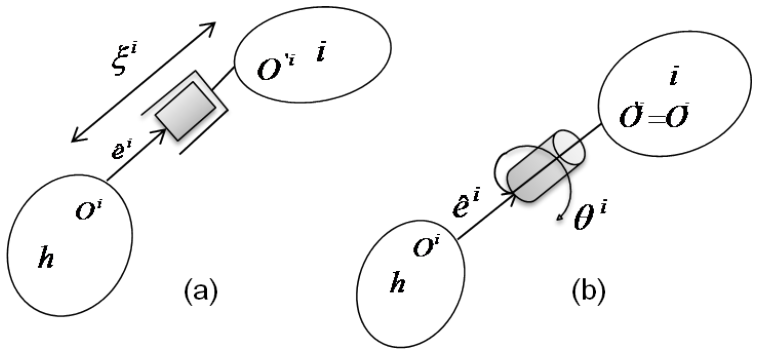

Figure 3.8: Prismatic joint (a) and revolute joint (b)

- The relative rotation angle Θ^i of body i with respect to its parent around the unit joint vector $\hat{\mathbf{e}}^i$, if joint i is revolute. See figure 3.8 (b).

$$\mathbf{\Omega}^i = \dot{q}^i \hat{\mathbf{e}}^i \tag{3.8}$$

For a tree-like multibody system with N^{body} bodies and joints, the system configuration can be fully determined by $N^{joint} = N^{body}$ generalized coordinates q^i representing the relative motion in the joint[22]. Thus, the relative motion of a body i with respect to its parent h can be characterized by the following joint vectors and have the following characteristics:

$$\mathbf{z}^i = q_i \boldsymbol{\psi}^i \tag{3.9}$$

$$\mathbf{\Omega}^i = \dot{q}_i \boldsymbol{\varphi}^i \tag{3.10}$$

3.1.4 Dynamics quantities

Figure 3.9 illustrates the main quantities, which will be useful for the characterization of the dynamics of a multibody system:

- m^i and $\mathbf{I}^i = \left[\hat{X}^i\right]^T I^i \left[\hat{X}^i\right]$, are the mass and the inertia tensor of the body i with respect to the center of mass G^i
- $\mathbf{d}^{ii} = \left[\hat{X}^i\right]^T d^{ii}$, the position vector of the center of mass G^i with respect to O'^i.
- $\mathbf{x}^i = \left[\hat{\mathbf{I}}\right]^T x^i$, the absolute position vector of the center of mass G^i
- $\mathbf{g} = \left[\hat{I}\right] g$, the gravity vector
- $\mathbf{F}^i, \mathbf{L}^i$ are respectively the internal resultant force and torque applied to body **i** by its child **h** through the joint i, the same that must be evaluated at point O^i.As well, reactions **-F**^{i} and **-L**^{i} are applied on body i. In figure 3.9 body i bears the reactions produced by its children joints, $\mathbf{F}^j, \mathbf{F}^k$ and $\mathbf{L}^j, \mathbf{L}^k$

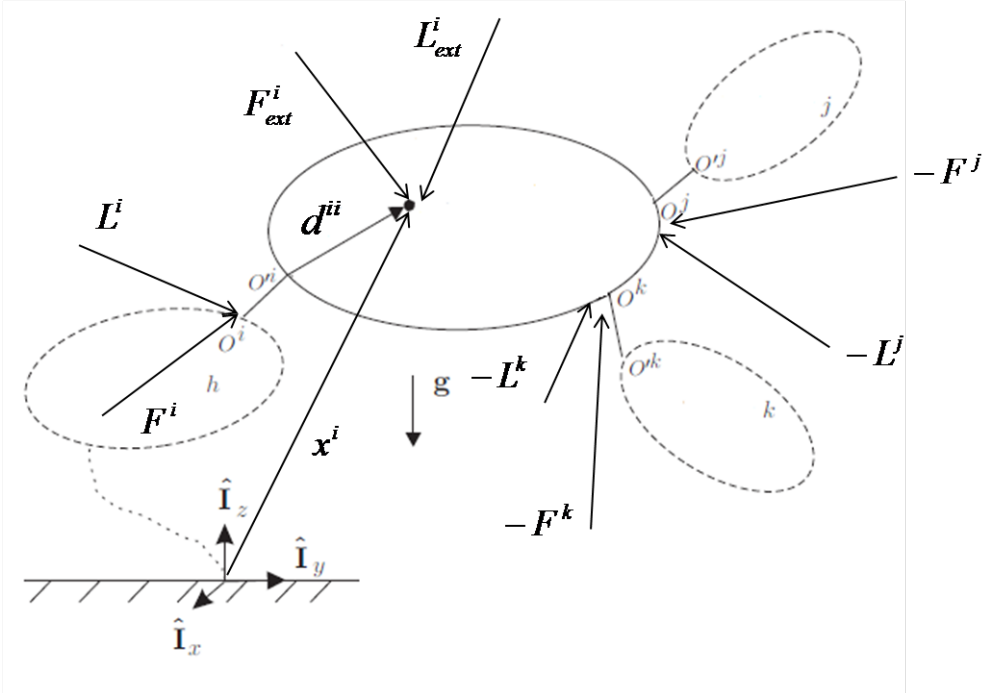

Figure 3.9: Main dynamical notations

- $\mathbf{F}^i_{ext}, \mathbf{L}^i_{ext}$ are the external loads acting on the body i as an equivalent resultant force $\mathbf{F}^i_{ext}$ applied to the center of mass G^i and a resultant torque $\mathbf{L}^i_{ext}$ with respect to the same point. Moreover, for locating the center of mass it is convenient to define an augmented vector $\mathbf{d}^{ii}_z$ as:

$$\mathbf{d}^{ii}_z \triangleq \mathbf{z}^i + \mathbf{d}^{ii} \tag{3.11}$$

- The total resultant force F^i_{tot} and torque applied on body i are given by:

$$\mathbf{F}^i_{tot} = \mathbf{F}^i - \sum_{h \in \bar{i}} \mathbf{F}^h + \mathbf{F}^i_{ext} + m^i \mathbf{g} \tag{3.12}$$

$$\mathbf{L}^i_{tot} = \mathbf{L}^i - \sum_{h \in \bar{i}} \left(\mathbf{L}^h + \mathbf{d}^{ih} \times \mathbf{F}^h - \mathbf{d}^{ii} \times \mathbf{F}^h \right) + \mathbf{L}^i_{ext} + \mathbf{d}^{ii}_z \times \mathbf{F}^i \tag{3.13}$$

  Where $\sum_{h \in \bar{i}}$ represents that$h$is ancestor of body i.

- Using the definitions from the previous section, the absolute position vector of the center of mass G^i of body i can be obtained from the relative displacement vectors as:

$$\mathbf{x}^i = \sum_{h<i} \left(\mathbf{z}^h + \mathbf{d}^{hi} \right) = \sum_{h<i} \mathbf{d}^{hi}_z \tag{3.14}$$

Where $h \rfloor i$ states that the body h is the direct ancestor of body i. Obtaining the derivates from the expression from above, the velocity vector of the center of mass G^i can be defined as:

$$\dot{\mathbf{x}}^i = \sum_{h<i} \left(\overset{\circ}{\mathbf{z}}^h + \widetilde{\boldsymbol{\omega}}^h \mathbf{d}_z^{hi} \right) \tag{3.15}$$

And its absolute acceleration vector as:

$$\ddot{\mathbf{x}}^i = \sum_{h<i} \left(\overset{\circ\circ}{\mathbf{z}}^h + 2\widetilde{\boldsymbol{\omega}}^h \overset{\circ}{\mathbf{z}}^h + \dot{\widetilde{\boldsymbol{\omega}}}^h \mathbf{d}_z^{hi} + \widetilde{\boldsymbol{\omega}}^h \widetilde{\boldsymbol{\omega}}^h \mathbf{d}_z^{hi} \right) \tag{3.16}$$

- Since the bodies are considered rigid, if the position of the reference material point is know and the orientation of the body as well, is possible determining the position of any point of the body. Taking the center of mass G^i as reference point, the position of any point vector of any point P on the body is:

$$\mathbf{v}^P = \mathbf{x}^i + \mathbf{r}^P \tag{3.17}$$

  Where r^P represents the relative position vector of P with respect to G^i. This vector is constant in the body fixed frame $\left\{ \hat{X} \right\}$ and $\overset{\circ}{\mathbf{r}}^P = 0$. The velocity vector of point P is:

$$\dot{\mathbf{v}}^P = \dot{\mathbf{x}}^i + \dot{\mathbf{r}}^P \tag{3.18}$$

- The *linear momentum* N^i of body i is defined as:

$$\mathbf{N}^i \triangleq m^i \dot{\mathbf{x}}^i \tag{3.19}$$

- The *angular momentum* of body i with respect the center of mass is defined as:

$$\begin{array}{rcl} \mathbf{H}^i & \triangleq & \int_i (\mathbf{r} \times \dot{\mathbf{r}})\, dm \\ & = & \int_i \left(\mathbf{r} \times \left(\boldsymbol{\omega}^i \times \dot{\mathbf{r}} \right) \right) dm \\ & = & -\int_i \left(\mathbf{r} \times \left(\dot{\mathbf{r}} \times \boldsymbol{\omega}^i \right) \right) dm \\ & = & \left[\hat{\mathbf{X}}^i \right]^T \left(-\int_i \widetilde{r} \widetilde{r} dm \right) \omega^i \\ \mathbf{H}^i & = & \left[\hat{\mathbf{X}}^i \right]^T I^i \omega^i = \mathbf{I}^i \cdot \boldsymbol{\omega}^i \end{array} \tag{3.20}$$

where $I^i \triangleq -\int_i \widetilde{r} \widetilde{r} dm$ is the *inertia matrix* with respect to G^i [13].

$$m^i = \int_i dm \tag{3.21}$$

3.2 Newton-Euler equations of motion

According to Newton's second law, the motion of the center of mass of body i subjected to forces F_{tot}^i whatever their origins is given by:

$$m^i \ddot{\mathbf{x}}^i = \mathbf{F}_{tot}^i \tag{3.22}$$

where the definition 3.19 of the linear momentum is used.

The Euler equation is the equivalent to Newton's second law for the rotation motion of a body i and it is defined as:

$$\dot{\mathbf{H}}^i = \mathbf{L}^i_{tot} \tag{3.23}$$

Using the definition of the angular momentum 3.19, equation 3.23 becomes:

$$\mathbf{I}^i \cdot \dot{\boldsymbol{\omega}}^i + \widetilde{\boldsymbol{\omega}}^i \cdot \mathbf{I}^i \cdot \boldsymbol{\omega}^i = \mathbf{L}^i_{tot} \tag{3.24}$$

3.3 Newton-Euler Recursive Formalism

The Newton-Euler formalism, is an important tool for obtaining equations of motion that allow the prediction a system behavior starting from a given initial state.

Furthermore, the Newton-Euler formalism obtains recursively the *semi-explicit* form of the dynamical equations of tree-like multibody systems[13].

$$M\left(q\right)\ddot{q} + c\left(q, \dot{q}, F_{ext}, L_{ext}, g\right) = Q \tag{3.25}$$

Where M is the generalized mass matrix, c containts the Coriolis, centrifugal and gravity terms as well as the external forces and torques. The equation 3.25 is exactly the same as the equation obtained from the Virtual Work Principle[2]

The structure of these equations will also show which terms correspond to inertia, Coriolis and centrifugal effects. The resultant equations of motion formulated in matrix form are:

$$M\left(q\right)\ddot{q} + c\left(q, \dot{q}, F_{ext}, L_{ext}, g\right) = Q \tag{3.26}$$

for unsconstrained system[13].

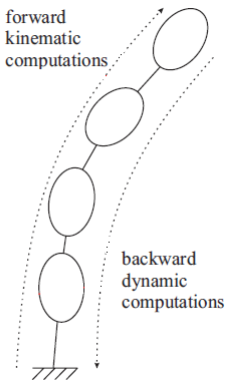

Figure 3.10: Foward kinematics and backward dynamics

The formalism uses forward and backward recursive computations as illustrates in figure 3.10:

[2]The Virtual Work Principle is a formalism for generating dynamical equations of tree-like multibody systems. The generated equations will show explicity dependence of the different terms of the dynamical equations on accelerations, the velocities and generalized coordinates.

- a foward kinematics computation of the position, velocity and aceleration vectors, is conducted from the root of the tree to the leaf bodies.
- a backward dynamics computation of the forces and torques on each body, is conducted from the leaf bodies to the root.

3.4 Rabbit Multibody Model

After reviewing basic concepts of the Modeling of Multibody Systems, it is possible to present a multybody approach of the Rabbit's kinematic and dynamic model. Thus, is very important to note that the generation of kinematic and dynamical equations of a dynamic system is a very long and tedious task, due to the huge amount of equations and complexity of a dynamic model. For solving this issue, the symbolic approach is a powerful tool that simplifies mathematical expressions and makes the set of equations highly portable for further applications such as control, optimization and dimensioning of a model[13].

For this important reason a symbolic approach has been chosen to model Rabbit, as an open tree-like structure with 7 rigid bodies. The children body of the base body (body *2*) is the trunk (body *2*). The trunk would be the parent the bodies *3* and*5* respectively, the left femur and the right femur. Furthermore, the body *5* is the parent of the body *6*, which is represents the left tibia of Rabbit. In the model the bodies *6* and *4*are the leaf bodies, since they are terminal bodies of the tree-like structure.

Figure 3.11 shows the multibody model of Rabbit.

3.4.1 Inertial frames and directions of movement

As exposed in previous in chapter 1 and 2, Rabbit is a planar biped and its movement is constrained to the plane formed by the y and z directions of the inertial frame $\left\{\hat{I}\right\}$. The base is attached to the body 2 by two prismatic articulations and a rotational articulation that enables the rotation of the around the axe x and the translation in the y and z directions of the inertial frame $\left\{\hat{I}\right\}$. Thus for implementing the three degrees of freedom joint, this can be modeled as a succession of two single-degree-of-freedom translational joints and one single-degree-of-freedom rotational joint.

The rest of bodies that are linked to the trunk, bodies *3*and *5*, are attached to the anchor points of trunk by revolute joints, which have an rotation axe around the x direction of the inertial frame. The ax of rotation of all the revolute joints is the axe x.

Moreover, each tibia, bodies *4* and *6* are linked to its respective parent bodies by revolute joints, that have the same rotation axe. Figure 3.11 shows a schematic representation of the 2D movement and the axes of rotation, the inertial frame is represented like $\left[\hat{I}\right]$. The center of mass of the body *2* (the trunk) is translated by the absolute vector $\mathbf{z}^i=\left\{\hat{X}^1\right\}^T d^{12} = 1.40\hat{X}_3^1$ from the fixed base.

Furthermore, the kinematic quantities that appear in the model are:

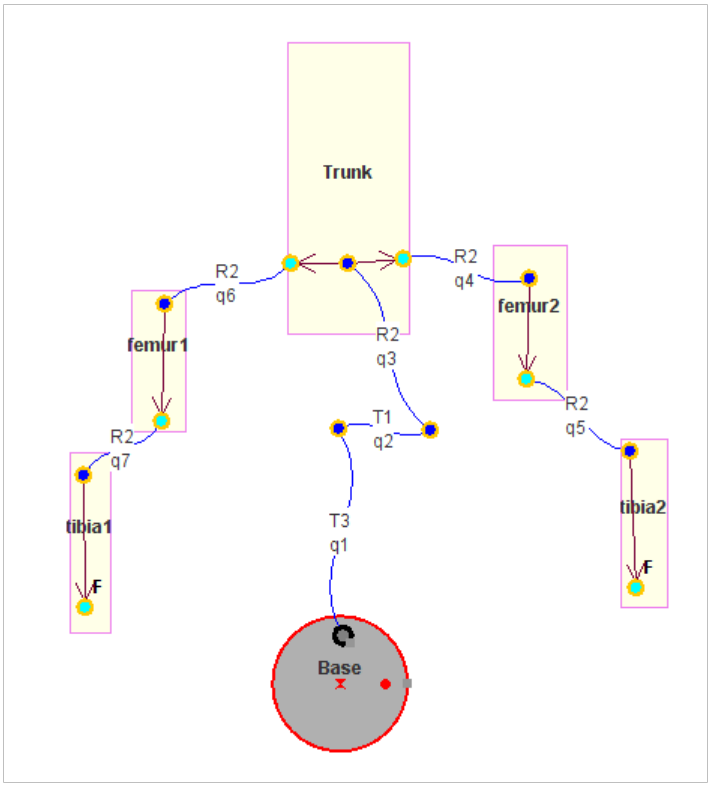

Figure 3.11: Multibody model of Rabbit

- The reference attachment points $O^i, O'i$ of the joint i, where $i = \{1, 2, 3, 4, 5, 6\}$, the same that determine the position vectors $\mathbf{d}^{ik}$, where $k = i + 1$. In the model with a i=3,5, $\mathbf{d}^{34}$ and $\mathbf{d}^{56}$ are the length of the right and left femur, respectively.

- The position vector of the center of mass G^i is $\mathbf{d}^{ii} = \vec{O'_i G^i} = \left[\hat{X}^i\right]^T \mathbf{d}^{ii}$ for each body. Furthermore; for all the bodies the vector of position of the center of mass is in the direction $\left[\hat{X}_i^z\right]$. In figure 3.11 the mobile reference frame of each body would be named $\left[\hat{X}_i\right]$, the centers of mass, the points of attachment of each articulation, the vectors of distance of each articulation and the vectors for the center of mass of each body.

Since the aim of this work is presenting a multibody representation of the Rabbit, an existing biped robot, this model will use the kinematic quantities measured like the robot:

- Length of the trunk, l_2= 0.63 [m].
- Length of each femur, right and left, l_3,l_5=0.40 [m].
- Length of the tibia, right and left, l_4,l_6=0.40 [m].

The positions vectors of each body i, with respect to the moving frame attached to each body i are:

- $\mathbf{d}^{23} = \left[\hat{X}^2\right]^T d^{23} = 0$
- $\mathbf{d}^{56} = \left[\hat{X}^5\right]^T d^{56} = -0.4\hat{X}_3^5$
- $\mathbf{d}^{25} = \left[\hat{X}^2\right]^T d^{25} = 0$
- $\mathbf{d}^{34} = \left[\hat{X}^3\right]^T d^{34} = -0.4\hat{X}_3^3$

Then, the position vector of the center of mass G^i of each body i are:

- $\mathbf{d}^{11} = \left[\hat{X}^1\right]^T d^{11} = 0$
- $\mathbf{d}^{22} = \left[\hat{X}^2\right]^T d^{22} = 0.315\hat{X}_3^2$
- $\mathbf{d}^{33} = \left[\hat{X}^3\right]^T d^{33} = -0.20\hat{X}_3^3$
- $\mathbf{d}^{44} = \left[\hat{X}^4\right]^T d^{44} = -0.20\hat{X}_3^4$
- $\mathbf{d}^{55} = \left[\hat{X}^5\right]^T d^{55} = -0.20\hat{X}_3^5$
- $\mathbf{d}^{66} = \left[\hat{X}^6\right]^T d^{66} = -0.20\hat{X}_3^6$

The inertial matrices are:

- For the trunk:

$$I^2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1.33 & 0 \\ 0 & 0 & 1.33 \end{pmatrix} \tag{3.27}$$

- For both femurs:

$$I^5 = I^3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0.47 & 0 \\ 0 & 0 & 0.47 \end{pmatrix} \tag{3.28}$$

- For both tibias

$$I^4 = I^6 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0.0.2 & 0 \\ 0 & 0 & 0.20 \end{pmatrix} \tag{3.29}$$

In 3.11 the external resultant forces and moments applied to each body are shown. As well, the tangential forces on contact over the feet or in the ends of bodies 4 and 6, respectively. The normal force F_n is the z direction of the inertial frame and the tangential force F_{tan} in the z direction. External moments should be applied to the bodies *3, 4, 5, 6*, in order to simulate actuators that give movement to the model, specially for obtaining a constant gait and observe a dynamical behavior of the model.

Chapter 4

External Forces acting over Rabbit

The dynamic model of Rabbit is subject to the influence of external forces that affect the dynamics of the model. Furthermore, the external forces present in the dynamic model result from the contact of the feet and the floor. In this section, an analysis of the non-linear contact model and dynamic friction model that characterizes these forces will be done.

4.1 Foot contact modelization

Foot contact modelization obeys a non linear model that tries to reproduce the physical behavior of the contact of the feet with the floor. It should be done through the assumption that the feet and the floor are both rigid bodies and the ground-feet contact is modeled as an instantaneous event that involves i contact forces. Because of this assumption, the normal force at the moment of the impact depends on the behavior of a non linear model and the tangential force, as well.

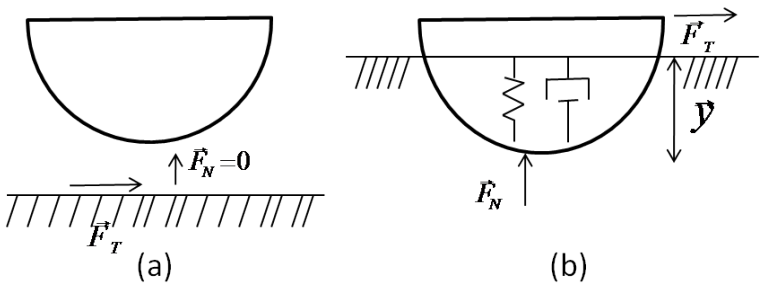

Figure 4.1: Model of Contact. Before the impact (a), during the contact (b)

The contact model used in this dissertation could be resumed as follows:

- Before the contact:

$$F_n = 0 \tag{4.1}$$
$$F_T = 0 \tag{4.2}$$

  Where the F_n is the normal force and F_T is the tangential force.

- When the foot is in contact with the ground:

$$F_n = -\lambda |y|^n \dot{y} - k |y|^n \tag{4.3}$$
$$F_T = \mu F_n \tag{4.4}$$

  where y is the penetration as shown in figure 4.1. In addition to this $\dot{y}$ is defined as the penetration speed, λ is the coefficient of restitution,k is the spring constant,n is a number that equals to $\frac{2}{3}$ and μ is the dynamic friction coeficient for the tangential force. In the following sections, this model will be explained in detail.

4.1.1 Linear spring-damp model

At first instance a spring-damp linear model for modelizing the contact of two rigid models seems quite accurate for modelizing two rigid bodies in contact, since it takes into account possible deformations of the bodies during the contact. The interaction force for the contacting objects is then given by the equation:

$$F = -\lambda_c \dot{y} - k\dot{y} \tag{4.5}$$

Where k is the spring constant in $[N.m]$, b is the damping coefficient in $[N.s.m^-1]$, y the penetration in *[m]* and $\dot{y}$ the penetration speed in $[m.s^-1]$.[17] Even thought this model accomplishes the assumption of occurring deformations, a discontinuity during time is present the same that may cause computational shortcomings during simulation. It may be caused by the following reason: In the moment just before the conctact, the contact is zero. In addition to this, the immediate moment upon the conctact the force will remain zero, since the spring force before the moment of the moment of the contact is zero.

Despite of the fact that the force due the spring is still zero just upon the contact, the applied damping force is instantaneously causing that the force steps from 0 to $-b\dot{y}$ and physically the interaction forces should star at zero and build up over time. For this reason, is necessary replacing the linear spring/damper parallel combination with a nonlinear one for solving this inconvenience .

4.2 Non-Linear Model

In this model the contact forces evolve continuously upon contact, despite the spring force is permitted to be nonlinear, the damping depends on the penetration depth. This makes physical sense because the damping increases with the deep of penetration as more area of the bodies comes into contact. Moreover, the contact forces evolve continuously upon contact . The dependence of the damping term in y (as well the spring term) causes the force to build up from zero upon contact and return to zero as separation approaches. [17]

$$F = -\lambda |y|^n \dot{y} - k |y|^n \tag{4.6}$$

Where the power n is close to one and depends on the surface geometry. For simulation purposes it will be set to n=3/2 , because in the nonlinear model it will simulate two contacting spheres under static conditions. The coefficient λ is coefficient of restitution, y is the penetration depth and $\dot{y}$ is the velocity of penetration of the foot into the ground [10].

4.2.1 Setting Parameteres and Coefficient of Restitution

The spring coefficient k is a fixed as a function of the mass, if the floor is considered rigid and y_{ss} is the admissible stationary deformation of the sole of the foot. In order to obtain this coefficient for the case of Rabbit, its average weight (m=*39.2kg*) and the admissible deformation of the rubber sole its foot ($y_{ss} = 0.15mm$) should be related by the following equation:

$$k = -\frac{mg}{y_{ss}} \tag{4.7}$$

For Rabbit's contact model the spring coefficient k will be$k$=*2500000*.

For choosing an adequate damping coefficient we are going to start explaining that the coefficient of restitution (e) for a linear model . For impact of an object of a mass m with a massive object such as the ground, the coefficient of restitution, e, generated by the linear model is given by:

$$e = e^{-b\pi/\sqrt{4mk-b^2}} \tag{4.8}$$

Where b is the damping coefficient (a constant) of the previous presented linear model of contact.

The equation 4.8 can be obtained assuming a damped sinusoidal response of contact model and non zero tensile contact forces. [17]The coefficient of restitution is an intrinsic property of the material which should not general depend on the mass. Even though it should depend on impact velocity. At low contact velocities and for most materials with a linear elastic range [17], the coefficient of restitution can be approximated by the equation:

$$e = 1 - \alpha v_i \tag{4.9}$$

Where v_i is the contact velocity and α in $\left[\frac{s}{m}\right]$ is a constant of proportionality of the damping term of the model, consequently the larger the value of α the more would be the damping. It usually has small values, rangin from 0.008 to 0.35 $\frac{s}{m}$.

Examining equation 4.9 the coefficient of restitution of a simple linear spring/damper model, shows no dependence of e on v_i, the contact velocity. For solving this problem, for the non-linear model previous work of Hunt and Crossley [10] proposes a new definition for the coefficient of restitution. This shows that the coefficient of restitution for a sufficiently small α and v_i can be giving by:

$$\lambda = \frac{3}{2}\alpha k \tag{4.10}$$

For the case of Rabbit contact model the parameters λ and α will have the following values: $\lambda = 900000$ and $\alpha = 2,4$

4.3 Application of the Normal Force non-linear model to 1D jumping model

The following section describes the performance characteristics of the a non-linear model of the normal force present during the contact of one foot with the ground. In addition to this, for analyzing its characteristics the model will be applied to the most basic case of a walker, a jumping robot of one degree of freedom the same that is shown in figure 4.2.[21].

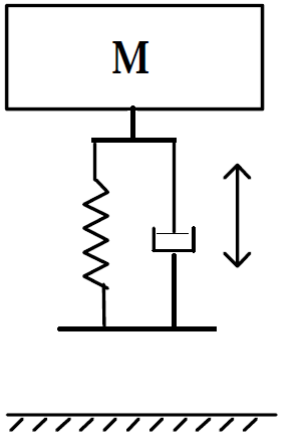

Figure 4.2: One degree of freedom Jumping Robot with a rigid ground

The normal force acting over the mass m depends either over the ground or it is in contact with the ground, as shown in the following equation:

$$\begin{array}{lll} F_n = & 0 & y > 0 \\ F_n = & -\lambda \left|y\right|^n \dot{y} + k \left|y\right|^n & y < 0 \end{array} \tag{4.11}$$

where $y > 0$ is the heigth over the ground and $y < 0$ is the height below the ground or the penetration. The movement could be divided in three phases, the first phase one that caracterizes the moment before the contact, when the mass is above the ground D_{nc}. The second phase D_{c1} the moment on which the impact takes place and the mass hits the ground with a velocity v_i and finally, D_{c2} the phase just after the contact, when the mass m returns from the ground with a velocity $-v_i$ under the condition the mass rebounds to the same height.

$$D_{nc} = (y, v) : y > 0 \tag{4.12}$$

$$D_{c1} = (y, v) : y \leq 0, v < 0 \tag{4.13}$$

$$D_{c2} = (y, v) : y \leq 0, v \geq 0 \tag{4.14}$$

The system becomes:

$$\begin{array}{lll} F_n = & 0 & \text{if (y,v)} \in D_{nc} \\ F_n = & -\lambda \left|y\right| \dot{y} + k \left|y\right|^n & \text{if (y,v)} \in D_{c1} \\ F_n = & \lambda \left|y\right|^n \dot{y} + k \left|y\right|^n & \text{if (y,v)} \in D_{c2} \end{array} \tag{4.15}$$

4.3.1 The Tangential Force model

In the previous section the characteristics and performance of the normal force where analyzed. Though for completely describing the model of feet contact of Rabbit and extending the previous model to a two degree of freedom case, an analysis and description of the tangential force model will be done. For this purpouse the normal force F_n will be studied as a non-linear model applied in the normal sense of the contact and the same non-linear model with a dynamic friction coefficient for the tangential sense of the contact.

During the contact of the foot with the ground, the phenomenon of Stick-Slip motion is present in the system, the same that is a typical behavior for systems with friction and it is caused by the fact that friction is larger at rest than during motion [12] .

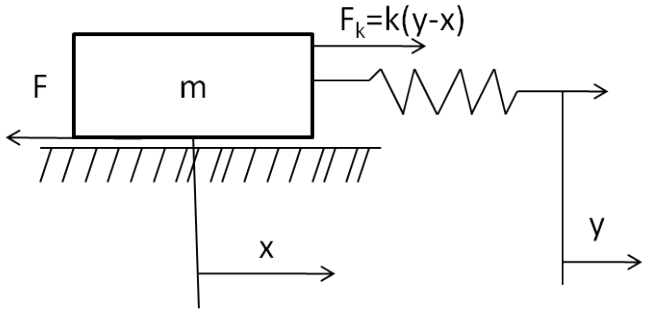

Figure 4.3: Experimental setup for stick-slip motion

A typical experiment that may give stick-slip motion is shown in Fig 4.3. A unit mass is attached to a spring with stiffness k=*2* N/m. The end of the spring is pulled with constant velocity, $dy/dt = 0.1m/s$. The mass is originally at rest and the force from the spring increases linearly. The friction force counteracts the spring force $F_k > \mu mg$, and there no displacement. When the applied force reaches break-away force, the mass stars to slide and the friction decreases rapidly due to the Stribeck effect. The spring contracts and the spring force decreases. The mass slows down and the friction force increases because of the Stricbeck effect and the motion stops. The phenomenon then repeats itself[17].

For the 2D extension of the model there are three possible states:

- the mass is over the floor and there is no contact, $y > 0$:

$$F_n = 0 \tag{4.16}$$
$$F_t = 0 \tag{4.17}$$

- In the second state the mass and the ground are in contact. The friction force counteracts the tangential force that is acting over the mass. This state is characterized by $y \leq 0$ and $|F_t| < \alpha_0 |F_n|$

$$F_n = -\lambda |y|^n \dot{y} + k |y|^n \tag{4.18}$$
$$F_t \leq -\lambda |x - x_c|^n \dot{y} - sign(x - x_c)\, k |x - x_c|^n \tag{4.19}$$

Where x_c represents abscise with the floor, where the deformation of the spring-damp stars. See figure 4.3.

- the mass and the floor are in contact and the mass stars sliding. This state stars with $y \leq 0$ and $|F_t| \geq \alpha_0 |F_n|$. In this case the tangential force is bounded by $\alpha_0 |F_n|$

$$F_n = -\lambda |y|^n \dot{y} + k |y|^n \tag{4.20}$$
$$F_t = -sign(\dot{x})\alpha_0 |-\lambda |y|^n \dot{y} + k |y|^n| \tag{4.21}$$

Is important to add that the introduction of a model of viscous friction, posses several limitations. First, once the mass is sliding the tangential force becomes saturated $|Ft_{sat}| = \alpha_0 |F_n|$. Furthermore, the switching between the state of sliding and stick takes place when $|F_t| < \alpha_0 |F_n|$. This fact posses the difficulty of detecting the instant when the system reaches the stick state which is necessary for evaluating the stick state abscise x_c for the reinitialization of the spring-damp deformations in the tangential force model [21].

Finally, this model doesn't generate the switching between the slip and stick states, what is more, previous testing should be done for estimating the current state of the system. For all the reasons exposed above, is necessary the introduction of a dynamic friction model that avoids these problems and is able of sustaining a numerical precision, even if the detection of the moment of the contact is not accurate.

4.4 Model of dynamic friction

In the tangential sense an important phenomena, should be considered: friction. For its vital role into the gait generation this section will be devoted for describing the model of dynamic friction.

4.4.1 Dynamic Friction

A suitable model for friction is important for predicting its effects over a system find controller gains and perform simulations. A better description of the friction phenomena for low velocities and especially when crossing zero velocity, is an improvement over previous classical friction models that try to describe relations between velocity and the friction force. These previous, typical examples are different combinations of Coulomb friction, viscous friction, and the Stribeck effect. The latter is recognized to produce a unstable effects at very low velocities.[3]. During this dissertation the model of LuGre will be used. This model tries to prove that contact surfaces are very irregular; so they can be considered as two rigid bodies that make contact through elastic bristles.

4.4.2 The model of LuGre

This model is based on the assumption that the surfaces are very irregular at the microscopic level and two surfaces therefore make contact at a number

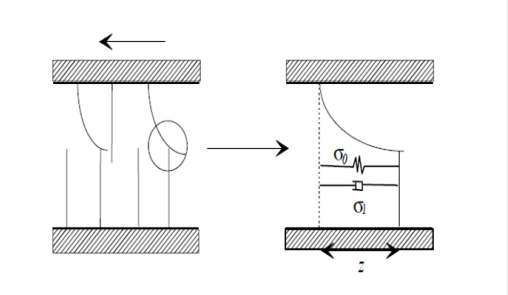

Figure 4.4: The friction interface between two surfaces is thought of as contact between bristles

of asperities [5]. This idea could be visualized as two rigid bodies that make contact through elastic bristles. When a tangential force is applied, the bristles will deflect like springs that will produce a friction force. See figure 4.4 for a graphical interpretation.

$$F_t = \mu(z)F_n \tag{4.22}$$

Depending on the applied force the bristles will deflect, but if the force is sufficiently large the bristles will deflect so much that they will slip. Hence, the phenomenon is highly random due to irregular forms of the surfaces. For this reason, the random behavior can be captured by a simple reset-integrator model which describes the aggregated behavior of the bristles.[9] . This model is proposed in the average behavior of the bristles. The average deflection of the bristles is denoted by z. The following equation models the average deflection of the bristles:

$$\frac{dz}{dt} = v - \frac{|v|}{g\,(v)}z \tag{4.23}$$

Where v is the relative velocity of the two surfaces on contact, in this case is the relative velocity between the feet and the walking surface of the robot. Analizing the last equation, the second term asserts that the deflection z aproches to the value:

$$z_{ss} = \frac{v}{|v|}g\,(v) = g\,(v)\,sgn\,(v) \tag{4.24}$$

in steady state, for example when v is constant.The function $g\,(v)$ is a part of the static characteristic of friction. Moreover, is positive and depends on many factors such as material properties, lubrication, and temperature. For typical bearing friction, $g(v)$ will decrease monotonically from $g\,(0)$ when v increase. This is known as the Stribeck effect[5].

$$g\,(v) = \frac{\alpha_0 z + \alpha_1 \exp^{\left(-\left(\frac{v}{v_s}\right)^2\right)}}{\sigma_0} \tag{4.25}$$

Where σ_0 is the stiffness and v_s is the Stribeck velocity and σ_0[3].

Neglecting the second term of equation 4.25 [21] and replacing it on equation 4.23, we obtain a direct expresion of the average deflection of the bristles, in function of the speed and already known parameters:

$$\dot{z} = v - |v|\,\frac{\sigma_0}{\alpha_0}z \tag{4.26}$$

The friction force generated from the bending of the bristles is described as:

$$\mu(z) = \sigma_0 z + \sigma_1 \frac{dz}{dt} \tag{4.27}$$

Where σ_1 a damping coefficient. A term proportional to the relative velocity could be added to the friction force to account for viscous friction (α_2), so that:

$$\mu(z) = \sigma_0 z + \sigma_1 \frac{dz}{dt} + \alpha_2 \tag{4.28}$$

Where σ_0 is a stiffness coeficient, σ_1 a damping coefficient and α_2 is a term proportional to the relative velocity that will account as the viscous friction.

In conclusion the dynamic model could be resumed in these two equations:

$$\dot{z} = v - |v| \frac{\sigma_0}{\alpha_0} z \tag{4.29}$$

$$\mu(z) = \sigma_0 z + \sigma_1 \dot{z} + \alpha_2 v \tag{4.30}$$

From these equations, their former parameters are:

- α_0 in m, the static parameter of friction.
- α_2 in $s.m^{-1}$, the viscous friction.
- σ_0 in m^{-1}, stiffness coefficient
- σ_1 in $s.m^{-1}$, the damping coefficient.

Where α_0 and α_2 are static parameters of friction, moreover σ_0 and σ_1 are the dynamic parameters that determine the transitory response of the friction model depending on variations of the speed[21]

4.4.3 Simulation

The numeric values of these parameters were obtained after several simulations for finding the best ones in order to optimize the model. The values used there were:

- α_0 : 0.2850
- α_1: 0 (The Stribeck effect; that was neglected)
- α_2: 0.180
- k : 2500000
- λ_c: 9000000
- σ_0 : 260
- σ_1 : 0.6
- $Ncontact$: 1.5

Chapter 5

Trajectory Generation and Control Introduction

5.1 Introduction

The main objective of this section is to briefly explain the control law that generates Rabbit's trajectories which keep a dynamical stable gait in the whole system. Despite the accuracy of the dynamical model is important for achieving stable gait, a control law that is capable of compensating direct perturbations of the system for keeping a stable behavior of the system. From the literature, several categories of control algorithms appear which can be classified into categories: time-dependent and time-invariant algorithms. By far, the most popular algorithms are time-dependent and involve the tracking to precomputed trajectories, on which the periodic walking motion must be supplied by an external trajectory planner, usually in the form of desired joint trajectories.[14]

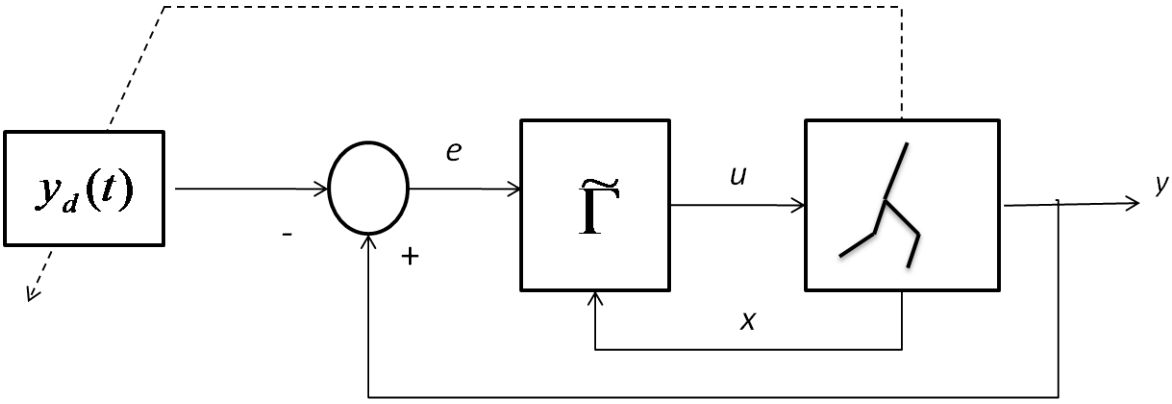

Figure 5.1: Block diagram of a trajectory tracking controller.

It is challenging to design the trajectories in such a way that the resulting nonlinear, time-varying, closed-loop system is stable. Figure 5.1 shows the block diagram of a trajectory tracking controller, the controller Γ force the error $e = y - y_d$ to zero so that the output y tracks the desired trajectory $y_d(t)$. The

dashed line indicates that the trajectories $y_d(t)$ may be modified on the basis of the robot's state.

On the other hand, time-invariant controller which does not involve trajectory tracking is able to cope with external disturbances better than a controller based upon tracking. For a controller based upon tracking, if a disturbance affects the robot and causes its motion to be retarded with respect to the planned motion, the feedback system is obliged to play catch up in order to regain synchrony with the reference trajectory. What is more important is the orbit of the robot's motion that is the path in state space traced out by the robot, and not the slavish notion of time imposed by a reference trajectory. A time-invariant controller causes the robot response to a disturbance to converge back to the period orbit, but not to attempt otherwise re-synchronizing itself with time. [6] One way to achieve this is by parametrizing the orbit (the walking motion) with respect to (a scalar-valued function of) the state of the robot, instead of time [15]. In this way, when a disturbance perturbs the motion of the robot, the feedback controller can focus solely on maintaining appropriate limb positions and velocities for that point of the orbit, without re-synchronizing with an external clock.

In fig 5.2

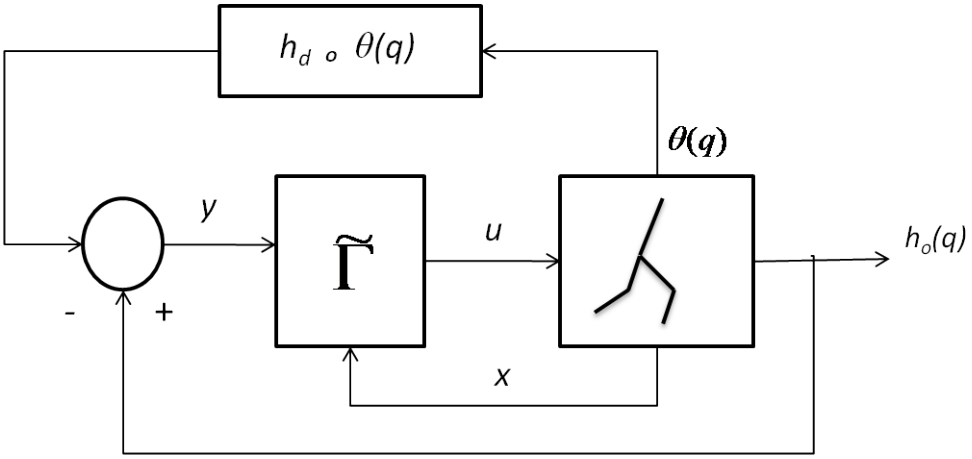

Figure 5.2: Block diagram of time-invariant controller

In Figure 5.2 the controller Γ forces the signal $y = h_0(q) - h_d \circ \theta(q)$ to zero so that the signal $h_0(q)$ tracks the function $h_d \circ \theta(q)$. In this way, the control is action is clocked to events on the robot's path and not to an externally supplied time-based trajectory. With proper design of $h_0(q)$ and $h_d \circ \theta(q)$, a self-generated limit cycle exists through the combined actions of the controller and the environment on the robot.

5.2 Gait hypotheses and Robot assumptions

In the first section concepts like the *swing phase*, *double support* and *walking*. For these previous defined concepts is the contacting leg when only one leg is in contact with the ground and the *swing leg* is the leg in the air. For control

purposes, globally the gait is modelized by two phases:

1. The swing phase
2. The impact phase

The two phases are critically in the design of the virtual constraints, in stating a zero dynamics law[1], the same that will be the basis for an asymptotically stable design of controllers for achieving a stable walking gait.

For this reason, there are certain assumptions that the robot may hold and gait hypotheses that will take place.

- RH1) The robot has N links with mass the same that are connected by revolute joints and all the structure lacks of closed kinematic chains
- RH2) The robot posses a 2 degrees of freedom in each body, which means that the movement of the robot is designed to be at the sagittal plane.
- RH3) The robot has to symmetric legs, both connected to a reference body, the hip, which makes the model symmetrical with reference to the hip.
- RH4) Is actuated in each joint
- RH5) It has point feet, which means it lacks of actuation at the point of contact between the current stance leg and the ground.

RH1) and RH2) imply the robot has (N+2)- degrees of freedom (DOF)(N joint angles plus the Cartesian coordinates of the hip).

Conditions on the controller will be imposed and shown to that the robot's consequent motion satisfies the following gait hypotheses as well:

- GH1) The gait is composed by alternating double support and single support phases.
- GH2) during the single support phase, the stance leg acts as a pivot joint. Moreover, the vertical component of the ground reaction force is positive and the ratio of the horizontal component to the vertical component does not exceed the coefficient of static friction.
- GH3) the double support phase is instantaneous, so the contact can be modeled by a rigid contact model
- GH4) at the impact the swing leg neither slips nor rebounds. This could be implemented in the model by certain algebraic constraints.
- GH5) in steady state, successive phases of single support are symmetric with respect to the two legs
- GH6) in each step, the swing leg starts from behind the stand leg and is placed strictly in front of the stance leg at impact.
- GH7) Walking is from left to right and takes place on a level surface.

[1]The zero dynamics law is a control

5.3 The Swing phase model

This phase has N degrees of freedom, which are the 5 angular coordinates describing the configuration of the robot. Since the robot structure is assumed to be symmetrical, the swing phase model can be used irrespectively for which leg is the current stance leg.

The Dynamic model of this phase is the same model is obtained by the method of Lagrange[2] and the model is written in the form:

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = Bu \tag{5.1}$$

Where the matrix D is the inertia tensor; C is the Corolis matrix; G is the gravity vector; and B is a linear map from joint torque space to configuration space.

According to the precedent model derivation, RH4) and RH5) torques $u_i, i = 1, 2, 3, 4.$ are applied to every connection of two links. As assumed before, the contact point of the leg with the ground is unactuated. The swing phase model is written in state space form by:

$$\dot{x} = \dot{q} \tag{5.2}$$

$$\dot{x} = D^{-}(q)\left[-C(q,\dot{q}) - G(q) + B(q)u\right] \tag{5.3}$$

$$\dot{x} =: f(x) + g(x)u \tag{5.4}$$

Where $x := (q^{'}, \dot{q}^{'})^{'}$.

The state space of the model is taken as $\left[TQ := \left\{x := \left(q^{'}, \dot{q}^{'}\right) | q \in Q, \dot{q} \in \Re^N\right\}\right]$, Where Q is a simply-connected, open subset of $[0, 2\pi)^N$ corresponding to the physical reasonable configurations of the robot, for example avoiding the knees are backwards.

5.4 The Impact model

Am impact occurs when the swing leg touches the walking surface. Therefore, the impact model delivers the change of the robots coordinates that occur after each phase of double support and in addition to this the impact model accounts for the labeling of the robot's coordinates that occurs after each phase of double support. For developing the impact model the full (N+2)-DOF model of the robot is required. The first steep for generating the impact model is adding Cartesian coordinates $\left(p_H^h, p_H^v\right)$ to the hip. With this extended coordinates the new model can be obtained by the method of Lagrange.

$$D_e(q_e)\ddot{q_e} + C_e(q_e,\dot{q_e})\dot{q_e} + G_e(q_e) = B_e u + \delta F_{ext} \tag{5.5}$$

Where $q_e := \left(q_1, q_2, q_3, q_4, q_N, p_H^h, p_H^v\right)$, which are the relative angles between the torso and femurs, the two relative angles between the torso and femurs, the two relative angles at the knees, the angle of the torso with respect to the vertical, and the Cartesian position of the hips, $\left(p_H^h, p_H^v\right)$ and δF_{ext} represents the vector of external forces acting on the robot at the contact point.

[2]The Lagrange method consists of first computing the kinetic energy and potential energy of each link and then summing terms to compute the total kinetic energy, K, and the total potential energy, V and the Lagrangian is defined as L=K-V [23]

Since is assumed that the impact is instantaneous; the impulsive forces due to the impact may result in an instantaneous change in velocities, no change in the positions and the contact of the swing leg end with the ground results in no rebound and no slipping of the swing leg, and the stance leg lifting from the ground without interaction. Moreover, the double support phase is instantaneous. Under these assumptions 5.5 can be used to determine an expression of $\dot{q}^+$ (the vector of angular velocities just after the impact, in terms of the configuration of the robot at impact) and $\dot{q}^-$, the vector of angular velocities just before impact [6]. The post-impact velocity is then used to re-initialize the model for the next step. Since the model 5.4 makes the choice of the stance left, a change of coordinates is necessary since the former swing leg must now become the stance leg, and vice versa. An expression for $x^+ = (q^+, q^+)$ in terms of $x^- = (q^-, q^-)$ can be written as:

$$x^+ = \Delta\left(x^-\right) \tag{5.6}$$

5.5 Plant model: a hybrid nonlinear under actuated control system

With the addition of an appropriately chosen switching set swing phase can be combined with the impact model and expressed as a nonlinear system with impulse effects.

$$\begin{array}{ll} \dot{x} = f(x) + g(x)\,u & x^- \notin \boldsymbol{S} \\ x^+ = \Delta(x^-) & x^- \in \boldsymbol{S} \end{array} \tag{5.7}$$

where

$$S := \left\{(q, \dot{q}) \in TQ, p_2^v(q) = 0, p_2^h(q) > 0\right\}$$

is the set of point where the swing leg touches the ground. In simple words, a solution of the model is specified by the single support model until an impact occurs. An impact occurs when the state reaches the set S(the swing leg touches the ground), which represents the walking surface, resulting into a very rapid change in the velocity components of the state vector.

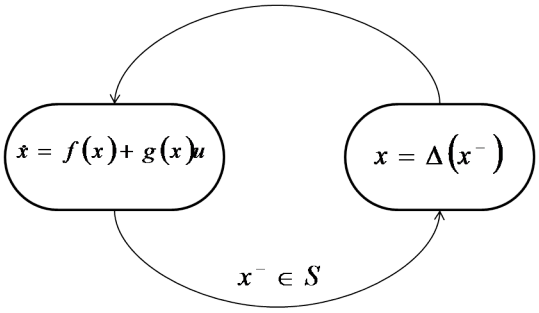

Figure 5.3: A graphical representation of the hybrid model for walking

The impact event is compressed into an instantaneous moment in time, resulting into a discontinuity in the velocities. The result of the model is a new

initial condition from which the single support model evolves until the next impact. For avoiding that the state of the robot does not take two values at the instant of the impact, the impact event is described in terms of the values of the state just prior to the impact at time t^-, and just after impact at time t^+. These values are represented by x^- and x^+, respectively[6]. Figure 5.3 shows a graphical representation of the hybrid model for walking corresponding to Rabbit.

5.6 The role of Gravity in Walking

Gravity helps Rabbit to rotate about the support leg end and advance forward in step despite, the lack of actuation at the end legs. Considering the angular momentum of the robot about the stance leg, which is assumed to be a pivot (under the assumption it does not slip and remains in contact with the walking surface), like σ the angular momentum balance theorem says that the time derivate of the angular momentum about a fixed point equals the sum of moments of the external forces about that point[14]. Since the motor torques act internally to the robot, their contribution to the moment balance is zero, leaving only gravity. This fact is expressed by:

$$\dot{\sigma} = M.g.x_c \tag{5.8}$$

where x_c is the difference between the x-coordinate of the stance leg and the x-coordinate of the center of mass of the robot, M is the total mass of the biped, and g is the gravity constant. In this regard, Rabbit functions just like a passive bipedal walker. Thus the main role of the actuators at the hips and the knees, is acting on the posture of the robot, thereby change the position of the center of mass and, thus, the moment arm through which gravity acts on the robot. In addition, the posture of the robot has a large effect on the energy lost at impact. The main challenge for an implementation of a control law is bringing all this together in a manner that ensures the creation of a desired stable periodic motion[6].

5.7 Virtual constraints

Virtual constraints are a concept that allows imposing holonomic constraints on a dynamic system through feedback control.

An simple example of this is fig5.4 shows a planar piston in an open cylinder. This sytem has a 1 DOF (the system can be modelized in terms of the angle of the crank θ_1 and its derivates). Figure 5.4(b) represents the planar piston without the constraints imposed by the walls of the cylinder. The system now has three degrees of freedom involving three coupled equations in the angles $\theta_1, \theta_2, \theta_3$ and its derivates. Only one degree of motion freedom when two constrains are imposed in the system:

- (a) the center of the position lies always on a vertical line passing through the rotational point of the crank
- (b) the angle of the piston head is horizontal throughout the stroke.

This is equivalent to imposing:

$$0 = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) \tag{5.9}$$

$$\pi = \theta_1 + \theta_2 + \theta_3 \tag{5.10}$$

These two constraints can be imposed through the physical means of the cylinder walls shown in Fig 5.4(a) or through the use of additional links like in Fig 5.4(c). If the system is appropriately actuated, the constraints can be imposed through feedback control. For this let's assume that the joints θ_2 and θ_3 are actuated and define two outputs in such a way that zeroing the outputs is equivalent to satisfying the constraints; for example:

$$y_1 = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) \tag{5.11}$$

$$y_2 = \theta_1 + \theta_2 + \theta_3 - \pi \tag{5.12}$$

The constraints will then be imposed by any feedback controller that drives y_1 and y_2 tp zero; for the design of the feedback controller, one could use a computed torque, or other methods. Outputs 5.12 are expressed as implicit functions of the actuated joint angles. As long as $L_1 < L_2$ constraints 5.10 can also be rewritten as explicit functions of the crank angle θ_1,

$$\theta_2 = \pi - \theta_1 - \arccos\left(\frac{L_1}{L_2}\cos(\theta_1)\right) \tag{5.13}$$

$$\theta_3 = \arccos\left(\frac{L_1}{L_2}\cos(\theta_1)\right) \tag{5.14}$$

Leading to the alternate output functions

$$y_1 = \theta_2 - \left(\pi - \theta_1 - \arccos\left(\frac{L_1}{L_2}\cos(\theta_1)\right)\right) \tag{5.15}$$

$$y_2 = \theta_3 - \arccos\left(\frac{L_1}{L_2}\cos(\theta_1)\right) \tag{5.16}$$

When constraints are imposed on a system via feedback control they are called *virtual constraints*. The planar three DOF piston of the precedent example can be virtually constrained to achieve the same kinematic behavior as the

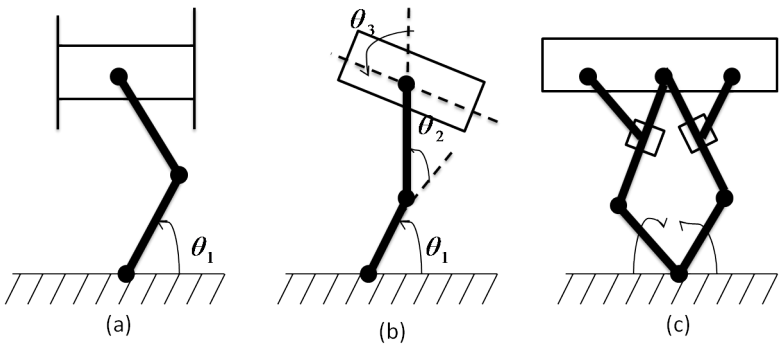

Figure 5.4: Virtual Constraints in a simpler context

one DOF piston in the figure 5.4(a); the resulting dynamic models are different because the constraint forces are applied at different points of the three-DOF position [6].

The virtual constraints can be imposed through the implicit constraints given 5.10 or the explicit constraints in 5.14. For controlling Rabbit explicit and implicit forms of the constraints have been used and the advantage of imposing constraints on the mechanism virtually (via feedback control) rather than physically is evident: the robot can be electronically reconfigured to achieve different tasks, such as walking at different speeds, going up stairs, and running.

For biped locomotion one important aspect, is the impacts of the swing leg with the ground and for designing virtual constraints, some care should be taken to account for the impacts. This is caused by the fact that at the end of the step, the impact map comes into play when the swing leg contacts the ground, providing a new initial condition for the next step and there is no reason for the new initial condition to satisfy the virtual constraints. For this reason the theory requires the introduction of the hybrid zero dynamics notion, for the controller.

5.8 Swing Phase Control of Rabbit through virtual constraints

For Rabbit several and complex different constraint choices were implemented, for regulating the angles of the torso, the height of the hips, and the position of the end of the swing leg (both horizontal and vertical components).

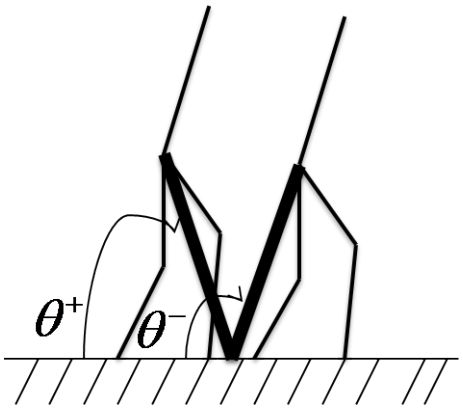

Figure 5.5: Virtual support leg

Since along a step the horizontal position of the hips monotonically increase and the desired motion of the robot is described in terms of the evolution of relative joint angles, the four relative angles can be virtually constrained as explicit function of the angle of the virtual support leg or the support tibia[6]. In figure 5.5, the angle of the virtual support leg is the line connecting the stance leg end to the hips. Parameterization of the relative joint angles (two relative angles of the torso with the femurs and the two relative angles of the knees) by this monotonic parameter makes the robot resemble as a single support inverted

pendulum which allows building a simple control scheme.

$$y = h_0(q) - h_d \circ \theta(q) \tag{5.17}$$

Where h_0 specifies (N-1) independent quantities that are to be controlled, $\theta(q)$ is an scalar function, that specifies the desired evolution of the configuration variables that is independent of h_0 and is monotonically increasing along a step. The function $h_d(\theta(q))$ specifies the virtual constraints as a function of $\theta(q)$, which are imposed under the condition $y = 0$ [26]. It is possible to interpret the quantity $\theta(q)$ as playing the role of time and the function h_d as taking the place of a desired trajectory, so in this way the evolution of the posture of the robot is synchronized to an internal variable [6]

5.9 The Hybrid Zero Dynamics

Lets the system of the planar piston described above, along with the set of outputs. When the dynamics of the system, compatible with the outputs being identically to zero, that is the constraints 5.14 being perfectly respected, is called the *zero dynamics*. In the case of the piston the number DOF is reduced to one, because of the well know principle that an N DOF mechanical system plus M(independent and holonomic) constraints leads to an N-M DOF mechanical system [6]. When the constraints are applied virtually instead of physically, the zero dynamics describe the exact behavior of the closed-loop system whenever the system is initialized so that the constraints are exactly satisfied, and the applied feedback controller maintains the outputs exactly zeroed; otherwise, the zero dynamics describe the asymptotic behavior of the closed-loop system as long as it is initialized sufficiently well that the feedback controller manages to drive the outputs asymptotically to zero[26].

The same ideas could be applied to Rabbit, but with the important difference is that it has a swing phase and an impact phase, so the idea of zero dynamics must be adjusted to deal with impacts, leading to the notion of *Hybrid Zero Dynamics* (HZD). The swing phase zero dynamics are the dynamics of the swing phase model restricted to Z, which is the surface of all points in the state space of the swing phase model of the robot corresponding to the outputs being identically to zero. Since the swing phase model has five DOF and there are four constraints, the swing phase will have one DOF. The swing phase zero dynamics, with help of the momentum balance theorem, can be written as a pair of first-order equations:

$$\dot{\theta} = \frac{1}{I(\theta)}\sigma_Z \tag{5.18}$$

$$\sigma_Z = Mgx_c(\theta) \tag{5.19}$$

where σ_Z is the angular momentum of the robot about the pivot point of the stance leg, restricted to Z, and $I(\theta)$ is the inertia.

At the contact instant of the swing phase, the impact model must be applied, resulting in a new initial condition of the five DOF model. The initial condition resulting from the impact model must lie in Z for finding a solution of the zero dynamics that can be continued and the evolution the high DOF robot is exactly

and completely described by the one DOF model. In this case, where impacts in Z are mapped back into new initial conditions in Z, the constraints are said to be invariant under the impact map[6]. The HDZ consist of the zero dynamics of the swing phase in combination with the impact map, which leads a one-DOF hybrid system. Defining $z = (\theta, \sigma_Z)$, the HDZ can be defined as:

$$\dot{z} = f_{zero}(z) \quad z^- \notin S \cap Z \tag{5.20}$$

$$z^+ = \begin{matrix} \theta^+ & z^- \in S \cap Z \\ \delta_{zero}\sigma_z^- \end{matrix} \tag{5.21}$$

Where f_{zero} is by 5.19 and δ_{zero} is a constant that is computed from restricting Δ to Z. When finally a solution of the HDZ is obtained, is important to prove its existence and closed loop stability properties, which is a very difficult task since the system is very non-linear. For this reason the HDZ provides useful tools for performing a complete stability analysis via the Poincare map method[6]. The main results will provide simple analytical expressions that can be used in feedback design and they are robust to a certain amount of error. The main results are presented here:

a) There exist a periodic solution HDZ if, and only if, $\delta_{zero}^2 \neq 1$ and

$$\frac{\delta_{zero}^2}{1-\delta_{zero}^2} V_{zero}\left(\theta^-\right) + V_{zero}^{max} < 0 \tag{5.22}$$

Where

$$V_{zero}(\theta) = -\int_{\theta^+}^{\theta} I(\xi) M g x_c(\xi)\, d\xi$$

$$V_{zero}^{max} = \underbrace{max}_{\theta^+ \leq \theta \leq \theta^-} V_{zero}(\theta)$$

b) there exists an exponentially stable periodic solution if and only if 5.22 holds and

$$0 < \delta_{zero}^2 < 1 \tag{5.23}$$

For a further reference, see [14], [26] and [6]

5.9.1 Controller design

The Bezier polynomials are used to parametrice the linear output function 5.17, yielding:

$$y = h(q) = h_0(q) - h_d(\theta(q), a) \tag{5.24}$$

Where $h_0(q)$ Specifies (N-1) independent quantities that are to be controlled, $h_d \circ \theta(q)$ specifies the desired evolution of these quantities as function of $\theta(q)$ and a is a vector of real coefficients. It's important to remember that the controller doesn't perform a tracking of the trajectory, since the trajectory doesn't depends of time. Moreover, $h_0(q)$ depends of $\theta(q)$. The Bezier polynomials make it very easy to satisfy the invariance condition, so that the HZD are guaranteed to exist[6]. The cost function of the Bezier polynomials is posed as:

$$J(a) = \frac{1}{sl(T)} \int_0^T \|u^*(t,a)\|^2 dt \tag{5.25}$$

where T is the time to complete the a step, $sl\,(T)$ corresponds to step length and $u^*\,(t, a)$ is the vector of constraint torques from zeroing the output 5.24. A programming package designed and implementing in the control simulink model is used to minimize $J\,(a)$ with respect to a for ensuring the existence of an asymptotically stable orbit, inequities 5.22 and 5.23, a desired walking rate; adequate contact conditions; maximum deflection of stance leg and swing leg knees and actuator limitations [6] Thus, when function of cost finds a solution, the result in an asymptotically stable, closed-loop system that meets the kinematic and the dynamic imposed constraints. This method has been used by the designer of Rabbit's control system for achieve asymptotically stable walking for a wide range of speeds [14].

5.10 Controller selection and implementation

The controlled variables $h_0\,(q)$ are the relative knee angles and the relative angles between the torso and femurs, which are the actuated variables.

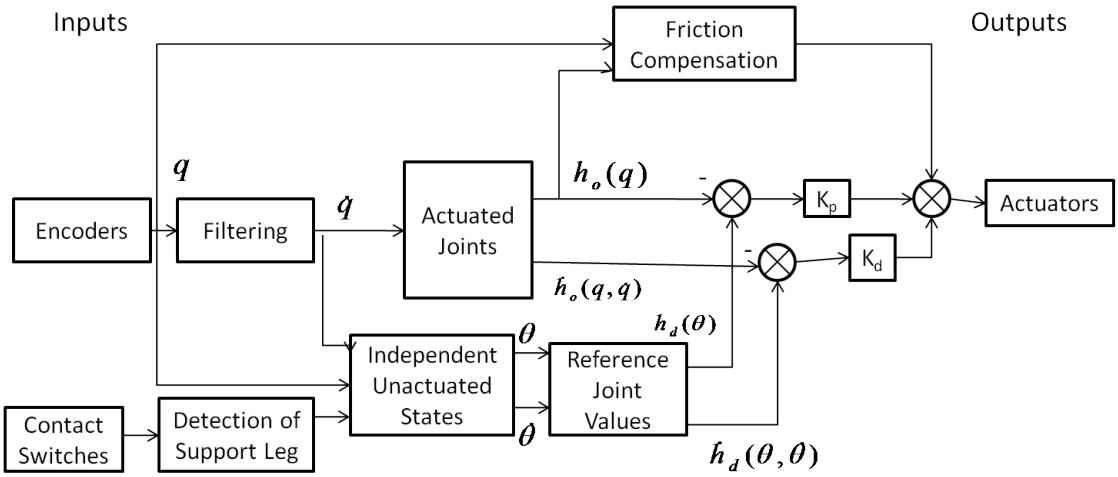

Figure 5.6: Block diagram of the control system

Figure 5.6 shows the diagram of the control system implemented on Rabbit. In the control simulink block, the outputs are zeroed by independent joint-level PD controls. Furthermore, the set of gains of the PD controller result in stable operation for the both possible contact conditions for walking, double support and single support.

In addition to this the swing leg touchdown is determined when the penetration of each foot is higher than a settled threshold by the user. Thus, when touchdown is detected the robot coordinates are permuted so that the same controller could be used independently of whether the inside or outside leg is the current stance leg. The filtering is performed by an observer included in the simulink block. It is important to remember that in systems were all the controlled variables are know, an observer can be used as a filter. In the case of the encoders and the actuator present in 5.6, the computational model will provide the positions and speeds of the configuration variables.

Chapter 6

Simulation and Results

6.1 Dynamic Behaviour under equilibrium

Under equilibrium conditions ($q_2 = 0, q_3 = 0, q_4 = 0, q_5 = 0, q_6 = 0, q_7 = 0$), the following experiment has been done, the same that lets the robot fall from a certain height $q_1 = 0.6m$ with the 2 feet closed, certain expected behaviours are observed:

- The position of the translational articulation q_1, after the MBS was dropped from a certain height, is stabilizated in a certain position is stabilized in a time t.
- The normal force has a maximun peak in the instant t of the collision, of the feet with the ground, after that it stabilizes reaching a transitory value, with little disturbances. Furthermore, the tangential force is zero

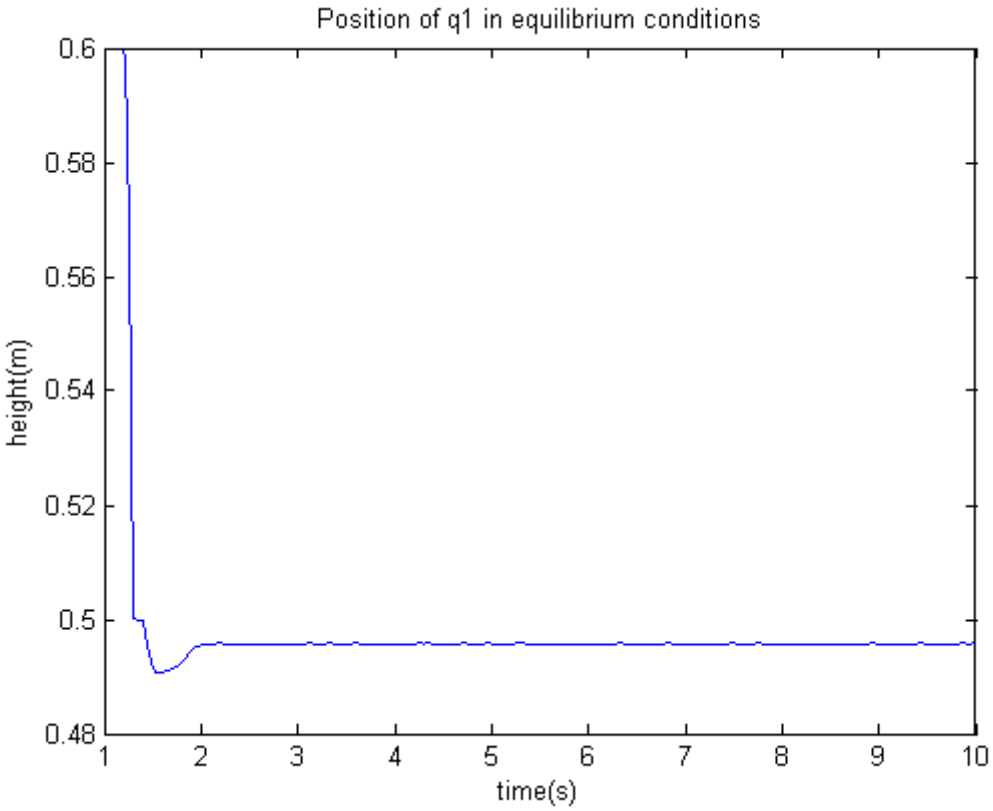

Figure 6.1: The following figure shows the behaviour of q1, when the robot falls from a certain altitude

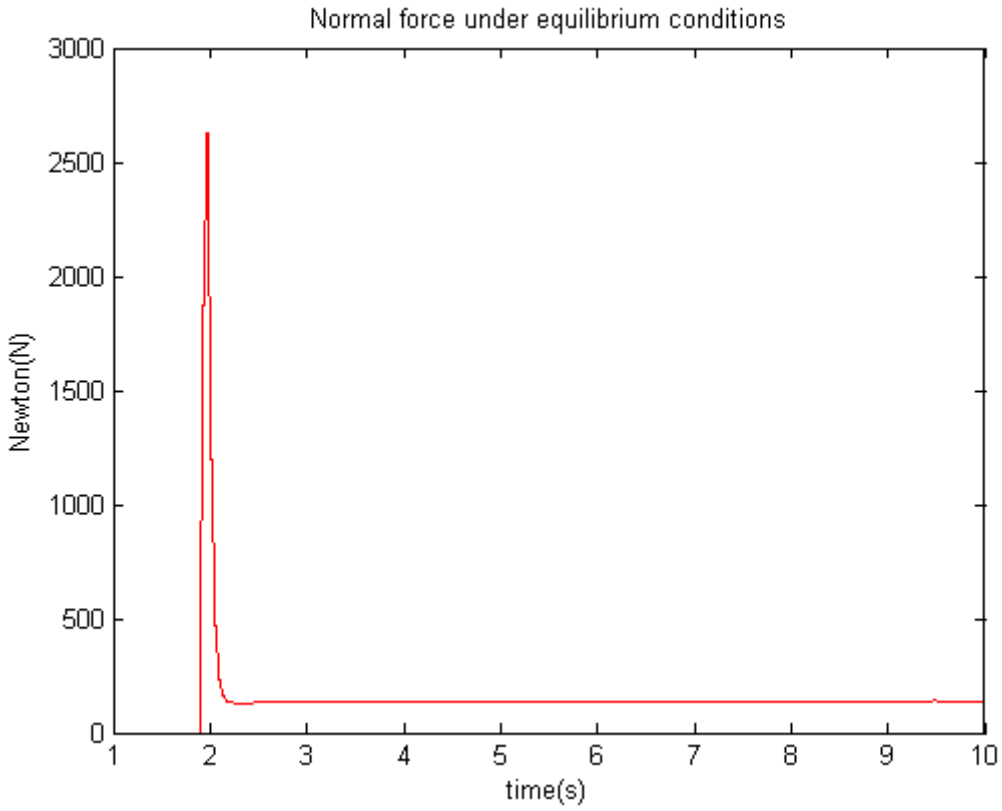

Figure 6.2: Resultant Normal Force, after the robot is dropped for a certain height

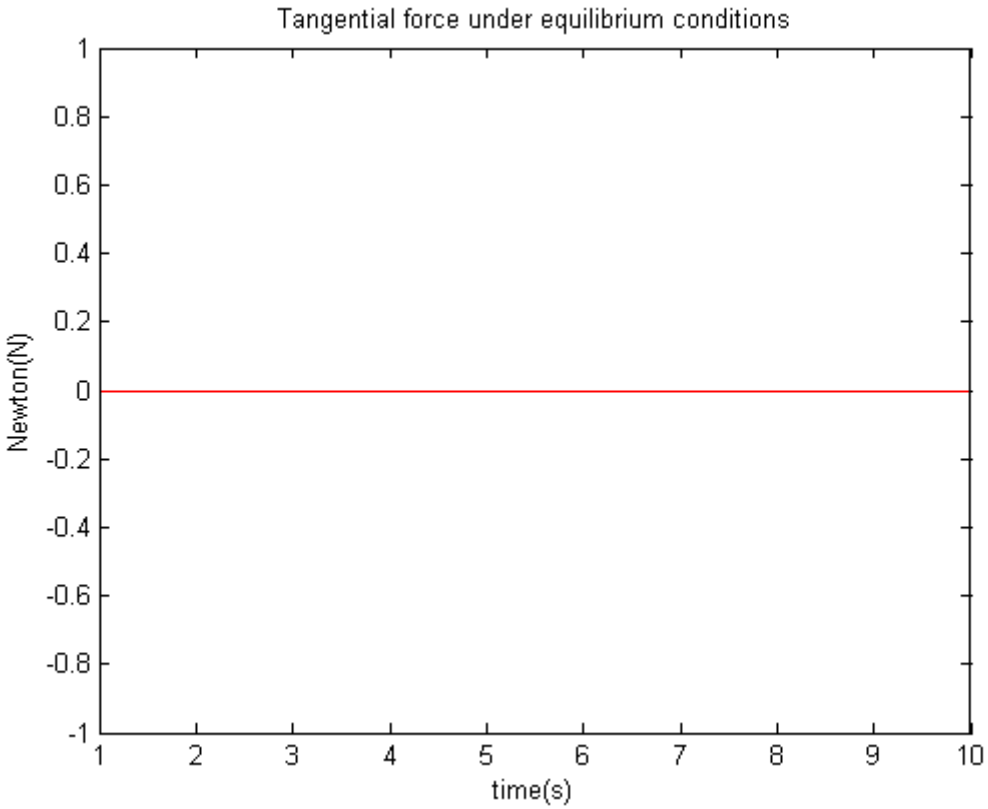

Figure 6.3: Tangential Force

Chapter 7

Conclusions

7.1 Conclusion and future work

7.2 Conclusion

This dissertation presents the multibody modelization of the prototype Rabbit. The work builds on previous models of Rabbit, the same that were built over modelization hypotheses like the number of articulations for the characterization of a planar biped robot, enabling to have a stable and energetically efficient biped gait. Moreover, the contact forces acting over Rabbit and its model were important for the implementation of a multibody model.

All this work, including the generation of trajectories and the control theory implemented in the real prototype, which was used with permission of the Grenoble Laboratory of Automatics in this dissertation, is mainly presented in the work of Eric Westervel [26] and Laurence Roussel [21].

This model is an attempt to give a multibody model of Rabbit and hopes providing a foundation for understanding how ground-feet contact forces, the posture of the Robot due to the calculated torques and the dynamical parameters affect over an stable gait. As well, this work hopes to be the basis for further development of control design, despite they are already implemented and developed on the real prototype, it would be interesting for students robotic research. Development of the theoretical framework involved a sequence of key steps over which several important facts where concluded.

The first step, described in Chapter 1 and 2, was performing a bibliographical research of the prototype Rabbit, for being able to have an idea of its structure and its parameters, as well as the control law and the external forces acting over it. Moreover, the bibliographical research as well focused on general information about biped locomotion. The main fact that arouses from this step is that Rabbit is a planar biped, since its motions are restricted to the sagittal plane by a boom which is attached to his hip. This allows us to conclude that our model will be a 2D model, which has displacements only in the y and z directions of a reference frame for the system.

The second step, described in Chapter 3, proposes a multibody model of Rabbit. For this was assumed to be a tree-like structure, of rigid bodies attached by rotational joints. The rigid bodies in this structure respectively represent the

trunk, the right and left femurs and the right and left tibia of each leg. Therefore, the trunk, where Rabbit's center of mass is located, has three degrees of freedom, two translations (in the y and z direction of the reference inertial frame) and one rotation around the axe x of the reference inertial frame. In addition of this, each articulation has one degree of freedom (rotation around the x axe of reference inertial frame). Thus, Rabbit is 7 degree of freedom planar mechanism and since it has just 4 actuators, is an underactuated system.

The third step, also described in Chapter 3, was the contribution of external forces over Rabbit, which are the feet-ground contact forces actingt over the end of the tibias. Since Rabbit has no feet, the end of the bodies of the tibia is considered as feet. These external forces are the normal force acting on the normal sense and the friction force acting in a tangential sense over the feet. The main effect of the external forces is avoiding slipping of the stance leg foot with the ground, when the robot is in the swing phase of the gait.

The fourth step, described in Chapter 4, proposes a model a model for the feet-ground contact forces. The feet-ground contact forces are the normal force acting over the feet on a normal sense and the friction force acting over the feet on the tangential sense are modelized by non-linear models. The normal force non-linear model overcomes many problems of the linear model, depending on the speed of the instant of contact and is continuous during contact transitions. For implementing the tangential force a model of dynamic friction was implemented which under low contact velocities is able to generate the switching between the slip and stick states of the stick-slip motion phenomenon presented at the moment of the contact of a foot with the ground.

The fifth step is to connect the model to the controller of Rabbit and for testing it. The controller is a time-invariant controller which does not tracks a trajectory, allowing the robot response to a disturbance to converge back to the period orbit. This is done by parametrizing the walking motion with respect to a scalar-valued function. The feedback controller maintains appropriate limb positions and velocities for keeping the periodic orbits of motion, without re-synchronizing with an external clock.

7.3 Future Work

This dissertation has been a very small contribution to the huge field of biped locomotion research. Despite, a correct modelization of a biped mechanism is very important for the implementation of a prototype and an adequate control law, designing stable trajectories for keeping gait stability and continuity is the main challenge of biped locomotion. Some research topics that follow as more direct extensions to the framework begun in this dissertation are now listed.

Designing Trajectories and Control algorithms Since the modelization of Rabbit is fully understand, the design and implementation of a control that assurances energy efficiency and stability is one of the main challenges and future work for this dissertation. Even though in the real prototype the control algorithm is efficiently robust against perturbations, implementing as an student work this long and difficult theory and fully understanding its background theory, seems the next step for this work, for fully understand the control of biped locomotion and non-linear control.

Addition of feet In this dissertation Rabbit modelization was the main objective. Since Rabbit is a biped which does not have feet, the dissertation main subject was the modelization of underactuated bipeds. As a future objective, the introduction of feet into the model will result into a full actuated model, which can improve robustness over an underactuated model.

Even though, this adds additional phases to the walking cycle and the additional torque available during the stance phase which results in full actuation and a more complex model and control theory, since it has more degrees of freedom. Despite of the increasing complexity of the dynamic model and the control of a biped with feet, any practicable biped will need feet to perform the statically stable maneuvers necessary for walking on surfaces with low coefficients of friction, for climbing and for negotiating obstacles.

Walking in three-dimensions Since the sagittal plane dynamics are almost decoupled from those in the frontal plane, it is conjectured that the controllers implemented for Rabbit (whic is a planar biped) can be implemented with controllers for the stabilization of motions in the frontal plane to produce stable, dynamic three-dimensional walking [14].

Running This is an extension of the walking model of Rabbit. The primary difference between running and walking is the presence of a flight phase, when no part of the biped is in contact with the ground. The presence of the flight phase adds constraints to the dynamic model and controlling the robot to land in a desired configuration, makes more difficult the design of a feedback law. [26]. Despite, in the trajectory generator of Rabbit which was used in this dissertation to test the implemented model, the adequate trajectories are calculated for running and the same model can be used, simulations were not successfully performed due the lack of time.

7.4 Final thoughts

A Multibody model of Rabbit imposes several advantages over the traditional methods for obtaining the dynamical equations of a mechanical system, not just the facility the fast generation equation in matlab syntax, but as well for the possibility of analizing the interaction of external forces with each body of the system, the effects of external forces over the whole system, the static and dynamical behavior of each body of the system and the whole system as well.

These obvious advantages are really useful for detecting modelizing assumption errors, like wrong positioned articulations in the system, incorrect inertias or position vectors of each body, wrong modelization of external forces acting over each body and the whole system, etc, by analyzing the static and dynamical behavior of each body in the system.

The idea of considering a complex dynamical system like a biped robot into a multibody system can be as well applied to more complex system, like the human being. With this idea is possible modelizing the dynamical interactions of extremities prostheses with the attachment articulations of invalid patients, as well modelizing and implemented an exo-esquelon that enables invalid patients recover mobility and freedom. The mechanical modelization and

dynamic behavior of prostheses, like contact forces between an extremity and it, the rotations of artificial articulations in the prostheses. Moreover, adequate dynamical parameters like inertias, masses, lengths of the prostheses can be optimally determined for reaching a desired dynamical behavior, which is the behavior and function that the patient will require of a customized prostheses for an individual case.

Finally, the modelization of a system like prostheses, enables the whole understanding of the dynamical interactions between them and the possibility of developing a control law that assurances a desired dynamical behavior, which is stable and energetic efficiently. Like in the case of more independent bipeds, a control systems that deals with external perturbations and assurances a energetically efficient gait which are the main objectives in the field biped robotics research.

Appendix A

Implementation on Matlab and Simulink

After obtaining the symbolic fields that represent the kinematic and dynamical equations in matlab syntax, is possible to build a matlab function that charges in the workspace parameters such as absolute vectors for calculating contact forces and articulate coordinates, inertia matrices, masses, lengths, etc.

A.1 Implementing the Rabbit's dynamical model in a Simulink block

First of all the purpose of this section is shortly describe how the simulink blocks and its functions where constituted. For implementing Rabbit's dynamical model into a simulink block, we used a level 1 matlab S-Function and then charging this function into a simulink block. For this let us define a Simulink block that will contain our function.

It consists of a set of inputs, a set of states, and a set of outputs, where the outputs are a function of the simulation time, the inputs, and the states. In our case, the set of inputs will be respective joint torques $(u_1(q), u_2(q), u_3(q), u_4(q))$and the contact forces $(\mathbf{F}_{ext} = [F_{n1}, F_{n2}, F_{t1}, F_{t2}])$. The set of states will be the joint's accelerations$(\ddot{q} = \ddot{z}, \ddot{x}, \ddot{q_1}, \ddot{q_{31}}, \ddot{q_{41}}, \ddot{q_{32}}, \ddot{q_{42}})$ and the outputs will be joint's positions $(q = z, x, q_1, q_{31}, q_{41}, q_{32}, q_{42}, \dot{q} = \dot{z})$, the speeds $(\dot{x}, \dot{q_1}, \dot{q_{31}}, \dot{q_{41}}, \dot{q_{32}}, \dot{q_{42}})$ and the absolute speed and the vertical positions of each feet $(v_{z1}, v_{x1}, p_{z1}, v_{z2}, v_{x2}, p_{z2})$.

The S-function is organized as a set of S-function callback methods that perform tasks required at each simulation stage. During simulation of a model, at each simulation stage, the Simulink engine calls the appropriate methods for each S-Function block in the model. The tasks performed by S-function callback methods in our function include:

- Initialization, Prior to the first simulation loop, the engine initializes the S-function, including:
  - Initializing the SimStruct, a simulation structure that contains information about the S-function. During this task the number and dimensions of input and output ports are set, which in our function

they are 8 and 20 respectively. In addition to this, the vector of the initial conditions for time integration of the states $(q_O, \dot{q_0})$ and parameters (like the mass, inertia, length and position vectors of the each body of the system) should be

  - setting the block sample times. In this case we want that the function performs its task for a continuous sample time, so we set the vector ts like this: $ts = [0, 0]$.

- Calculation of next sample hit - Since in our function needs the next sample hit for storing data for animation, this stage calculates the time of the next sample hit; that is, it calculates the next step size.

- Integration - this applies to models with continuous states and/or no sampled zero crossings. Since our S-function has 14 continuous states, the engine calls the output and derivative portions of the S-function at minor time steps. During this flag the function is called mbs_exe_dirdyn for solving equation A.1and finding $\ddot{q}$ prior time integration and allowing time integration for finding the position and speed of the joints $q, \dot{q}$ for our S-function.

Calculation of outputs in the major time step - after this call is complete; all the block output ports are valid for the current time step. During this stage, our function will calculate the absolute vertical positions and speeds of each foot for the output vector. For this purpose the function mbs_sensor_Rabbit will be called during the execution of this stage. For the output vector of the function sys the joint speeds and displacements will be also included.

Terminate- Performs any necessary end-of-simulation tasks. For our function it closes the file that saves the data for the animation.

Figure A.1 shows the algorithm implemented our S-Function called model.m. It tries to show the steps on which the function computes the joint positions, speeds and the absolute positions and speed of each foot.

A.2 Implementation of contact forces

For implementing the equation 4.3,another S-function was used. This S-function is able to calculate the dynamic friction coefficient denoted by equation 4.27, by simply doing the time integration of the deflection of the bristles. As explained before, this block need as entries the absolute feet speeds v_z, v_x and the penetration p_z . fig A.2 shows a block that implement the normal and tangential force for one feet.

The model is charged on a enabling block, that is activated at the time interval just after the contact. This means that when a penetration exist, block will be activated and just for the period of time the penetration is present, after that it will be turn off. This will assure a proper switching between the two feet, while one is over the ground the other is in contact with it, so its respective block calculates its contact forces and the other is turned off.

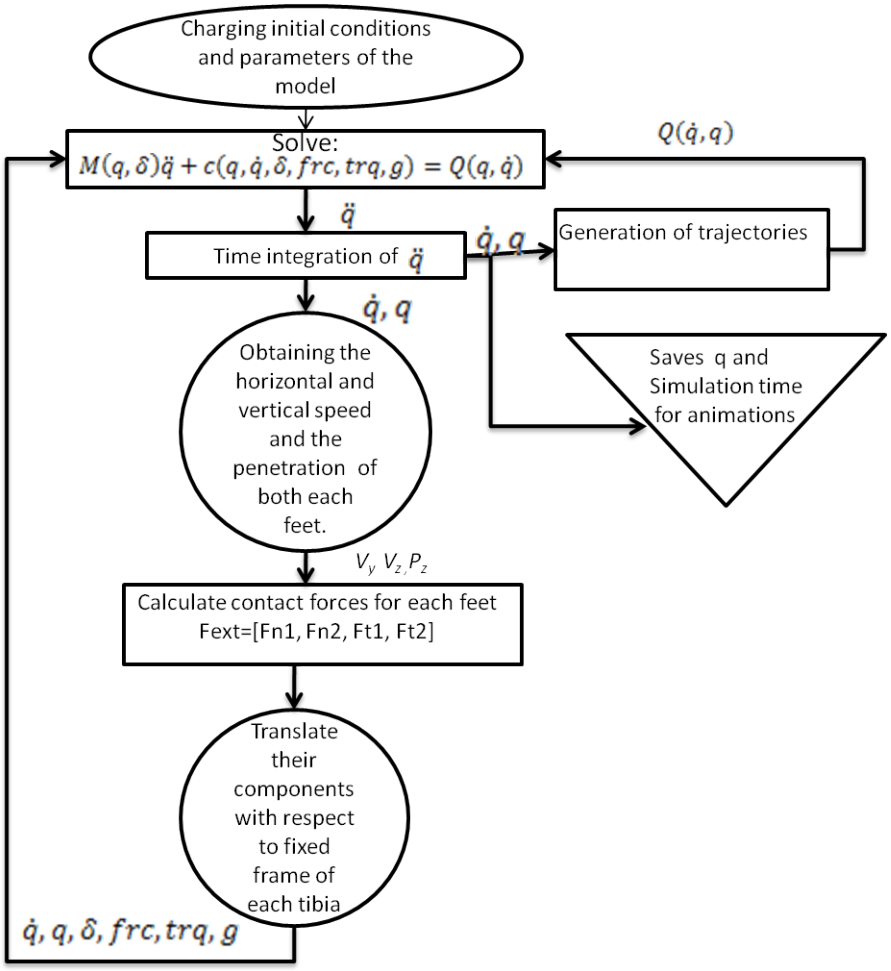

Figure A.1: Flowchart of the process

A.3 Animation of the Model

After simulating the model for a time t, the S-function of the main dynamic model saves the vector of time (the simulation time) and the articulated coordinates $(q_1, q_2, q_3, q_4, q_5, q_6, q_3,)$ in a file a *.res* extension.

Robotran, offers to the user the possibilty of 3D animations of the model. In the animation interface of Robotran's menu, is possible to directly charge the file of generated results for the animation of the 3D animation of the model and visualizate the dynamic and kinematic behaviour of the model.

In figure A.3 shows the 3D Animation model for Rabbit and its user interface.

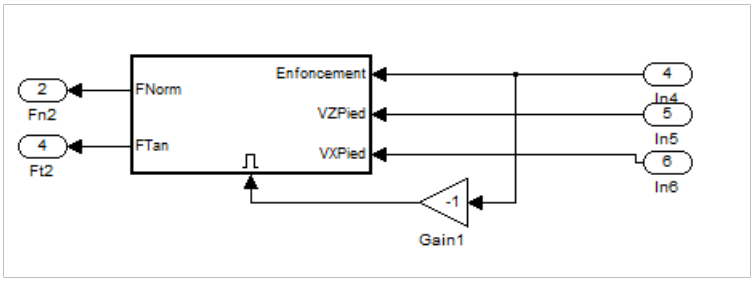

Figure A.2: External force's Simulink block

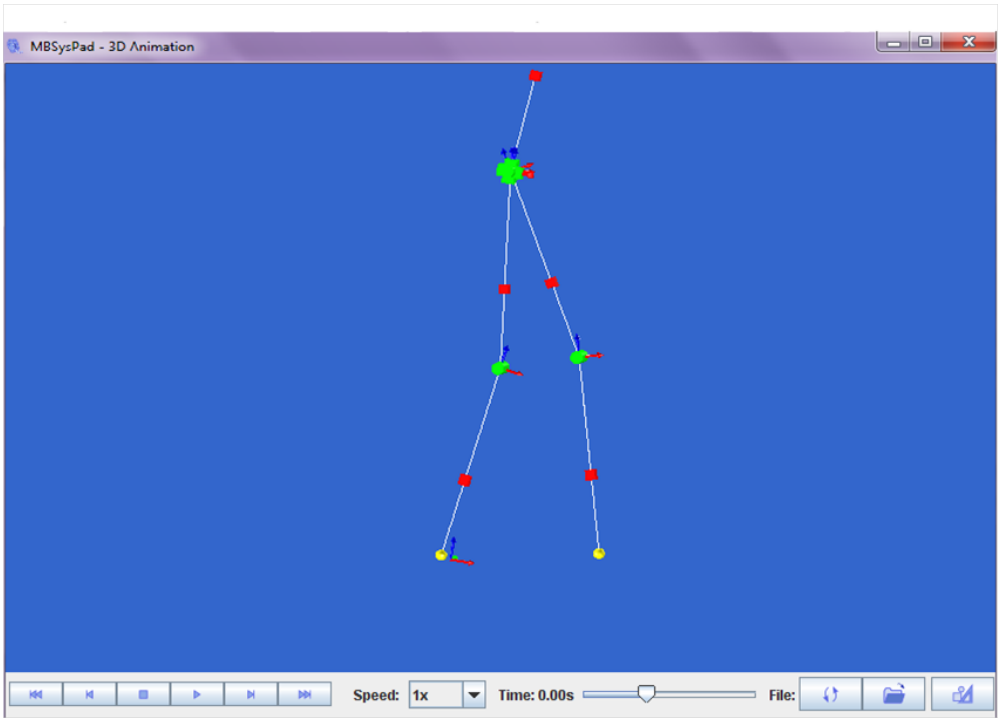

Figure A.3: Rabbit's 3D Animation

Appendix A

Implementation of the model in Robotran

For building an appropriate virtual multibody model of Rabbit, it was important to generate symbolic files that implement the kinematic and dynamical equations. For this purpose a multibody symbolic generator named Robotran was used.

Robotran is an environment developed at UCL-CEREM,according to Fig. 2. Starting from a graphical description of the MBS (with the MBsysPAD graphical editor), the MBS equations can be generated - in a few milliseconds - by the symbolic translator MBsysTran in MATLAB or SIMULINK syntax; these symbolic equations are then automatically interfaced with the MBsys-Lab program (MATLAB /SIMULINK -based), taking advantage of all the MathWorks programming facilities and specific toolboxes[24]. Moreover,3D animation of the virtual system can then be performed via the MBsysPad editor. Because of its obvious advantages, Robotran was one of the main tools for the implementation of this project.

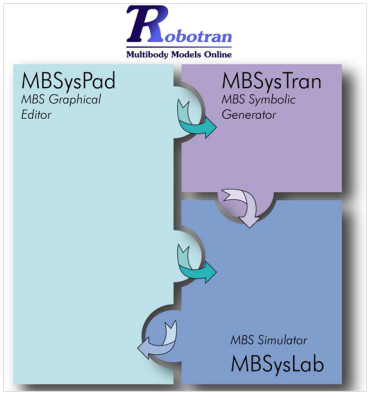

Figure A.1: The Robotran Program

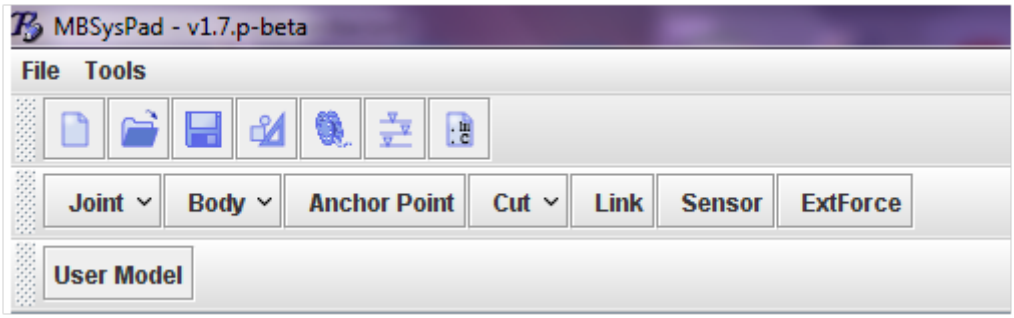

Figure A.2: MBSysPad editor user graphical interface menu

A.1 MBsysPad

The MSsysPad is the schematic editor of Robotran, which offers the advantage of a user graphical interface for building a Multybody Model closest possible to the original model. Bodies and joints (topology, type, shapes) are graphically sketched by the user. Moreover, it allows the user setting the masses, vector components, the inertia and anchor points[1] of each body. Closed-loop systems must be converted into tree-like structures by suitable loop cuts. The joints that fix the bodies are modelized as rotational or translational joints, with respect to its rotation or translational axe. For translational joints $T1, T2, T3$ and for rotational joints $R1, R2, R3$.

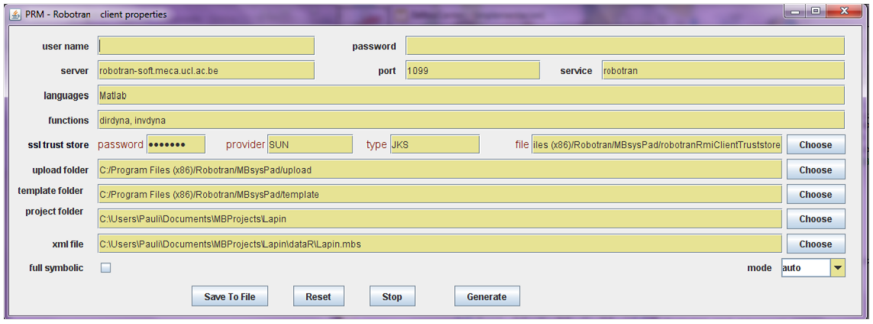

Figure A.3: Conecting window to the UCL server

Figure A.2 shows the main menu of MBsysPad with its utilities. After completely finishing sketching all the bodies of the system, the respective joints of each body and its parameters, it's possible to generate the symbolic files containing the kinematic equations by contacting the server of the UCL, see figure A.3. For systems with no more than 5 joints and bodies, it's possible to introduce as username:demo and password: demo, otherwise permission should be asked to the department of mechanical engineering of the UCL.

[1] An anchor point, is a point of a body where an articulation is fixed to it.

A.2 Dynamics of Tree-like Multibody Systems and Symbolic Files

In this section we will explain the structure the Kinematic quantities and dynamical equations implemented in the symbolic fields that were previously obtained. Their vector components and their body frames. For this work formalisms for tree-like MBS will be used, for obtaining its symbolic fields. Nevertheless, constrained systems (containing loops of bodies) can be modeled in Robotran by first becoming a temporary tree-like structures by either cutting a body into two parts or by cutting a joint, as seen in previous sections.

A.2.1 Direct Dynamics Symbolic Files

For multibody systems direct dynamics is the computation of the generalized accelerations $\dot{q}$ (joint accelerations) for a given configuration (q,$\dot{q}$) of the system to which forces and torques are applied. [24]. Direct dynamical equations will be used for predicting the motion of a system $(q(t), \dot{q}(t))$, starting from an initial configuration $(q\,(t=0)\,, \dot{q}\,(t=0))$ and by time integrating the accelerations $\ddot{q}$. Furthermore, various multibody formalisms can be used to compute joint accelerations $\ddot{q}$, like the standard Newton/Euler laws formulated recursively, the virtual principle and the Lagrange equations. The equation that describes the direct dynamics can be generated in this form:

$$M\,(q,\delta)\,\ddot{(q)} + c\,(q,\dot{q},\delta,frc,trq,g) = Q\,(q,\dot{q}) \tag{A.1}$$

Where:

- M[m*n] is the symmetric generalized mass matrix of the system
- c[n*1] is the non linear dynamical vector which contains the gyroscopic, centrifugal and gravity terms as well as the contribution of components of external resultant forces *frc* and torques *trq*.
- q[n*1] is the relative coordinates.
- $\delta\,[10n*1]$ gathers together the dynamical parameters of the system (body mass, centers of mass (the three components of vector l^i)) and the inertia matrices.
- Q[n*1] represents the generalized joint forces (torques).

The accelerations $\ddot{q}$ can be obtained by solving the linear system A.1. The calculation of **??**eq:vida) is computed by calling the mbs_exe_dirdyn matlab function.

A.2.2 External Forces and torques symbolic files

External forces and torques represent any force or torque acting on the bodies, in addition to the joint forces/torques and gravity. In Robotran, they are gathered together for body i in a unique resultant force $\mathbf{frc}^i$ and unique resultant torque vector $\mathbf{trq}^i$ with respect to the center of mass CM^i[24]. External forces and

torques on body i must be introduced by the user in terms of their components in the inertial frame $\left[\hat{I}\right]$:

$$\mathbf{frc}^i = \left[\hat{I}\right] \mathbf{frc}^i \tag{A.2}$$

$$\mathbf{trq}^i = \left[\hat{I}\right] \mathbf{trq}^i \tag{A.3}$$

In our model the external forces will be applied just to the final points of the two tibias, which will constitute the feet, see Figure A.4. Moreover, the resultant force vector $\mathbf{frc}^i$ for each tibia will be constituted by the contact forces (normal and tangential force) and the resultant torque vector $\mathbf{trq}^i$ will be zero.

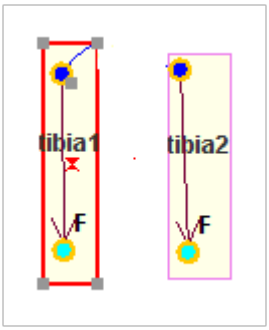

Figure A.4: Aplication points of the external forces acting on Rabbit

For implementing them on Robotran, external forces and torques are first graphically mentioned by the user in the MBsysPad editor (via **F** type sensor). In Robotran they must be implemented by the user in the function $user_Extforces$(Matlab and Simulink sintax). Since in this dissertation the models for the external forces are complex, avoiding difficulties during their implementation, this function won't be used. Instead, they will be implemented by Matlab functions and Simulink blocks. Nevertheless, the resultant force vector will be in terms of their components in the inertial frame $\left[\hat{I}\right]$, so it must be expressed with respect to the center of mass of each tibia. For accomplishing this Robotran automatically generates the function $mbs_extforces$ which contains the rotational matrices.

A.2.3 Sensor Kinematics symbolic fields

Robotran offers the user the possibility of computing the symbolic expression of the forward kinematics of any sub-chain in a multibody system. By this tool, is possible to compute the position, the orientation, the Jacobian matrix, the linear/angular velocities, the linear/angular accelerations of a given body (and of a particular point S of this body denoted $sensor$)[24].

In this dissertation the absolute position and speed of the feet will be calculated for the equation of the external contact forces acting over them. For this purpose we will consider the two sub-chains from the base body of Rabbit's model to bodies tibia1 and tibia2.

As defined in section 3, sub-chains are only defined for a tree-like MBS by covering the system from the base to the leaves bodies. Since closed structures

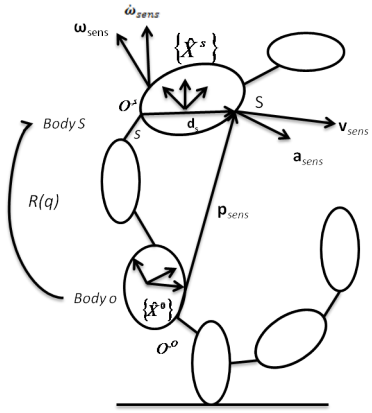

Figure A.5: Sub-chain kinematics

are first cut in order to restore a tree, sub-chain kinematics can be used in any situation. Considering Fig.A.5, the sub-chain o...s in which o represents the original body and s the body carrying the sensor S, located on body s by a constant position vector $\mathbf{d}_s = \left[\hat{X}^s\right]^t d_s$ with respect to the joint s connection point $\left(O'^s\right)$. For this sub-chain, the forward kinematics aims at computing:

- $P_{sens} = \left[\hat{X}^0\right]^t p_{sens}$: the position vector of sensor S with respect to point O'^0
- $v_{sens} = \left[\hat{X}^0\right]^t v_{sens}$: the relative velocity of point S with respect to frame $\left\{\hat{X}^0\right\}$

Most of the time, the desired computation is related to *absolute* quantities (position, velocities,...)with respect of the inertial frame $\left\{\hat{I}\right\}$. In that case, the original body o will simply be the base (body 0). In Robotran, sensors are first graphically introuduced (**S** sensor point + components of the $\mathbf{d}_s$ vector) by the user in the MBsysPad editor and the computation of their kinematics with respet to the inertial base is automatically generated in symbolic form in the function $mbs_projectname_sensor$.

Bibliography

[1]

[2] B. Espiau A. Goswami and A. Keramane. Limit cycles and theri stability in a passive bipedal gait. In *Proc. of the IEEE International Conference on Robotics and Automation*, Minneapolis, 1996.

[3] B. Armstrong-Helouvry. Stick-slip arising from stribeck friction. *Proc. IEEE Int. Conf. Robot. Autom.*, 2:1377–1382, 1990.

[4] Asimo's homepage. Honda Corporation, March 2010. [online] http://world.honda.com/ASIMO/.

[5] IEEE H.Olson Student Member IEEE K.J. Astrom-Fellow-IEEE C. Canudas de Wit, Associate and P. Lischinsky. A new model for control of systems with friction. *IEEE Transactions on Automatic Control, vol. 40, No. 3*, 0:0, 1995.

[6] Yannick Aoustin Franck Plestan E.R. Westervelt Carlos Canudas-de-Wit and J.W. Grizzle Christine Chevallereau, Gabriel Abba. Rabbit: A testbed for advanced control theory. *IEEE Control Systems Magazine*, 0:0, 2003.

[7] J. W. Grizzle Eric Westervelt, Gabriel Buche. Experimental validation of a framework for the design of controllers that induce stable walking in planar bipeds. *The International Journal of Robotics Research*, 23:0, 2004.

[8] M. Gienger F. Pfeiffer. The concept of jogging johnnie. In *International Conference on Robotics and Automation*, pages 3129–3135, May 2002.

[9] D. Haessing and B. Friedland. On the modeling and simulation of friction. In *1990 American Control Conference, San Diego, CA*, pages 1256–1261, 1990.

[10] K.H. Hunt and F.R.E Crossley. Coefficient of restitution interpreted as a damping in vibroimpact. *ASME, Journal of Applied Mechanics*, pages 440–445, June 1975.

[11] H. Dankowicz J. Adolfsson and A. Nordmark. 3d passive walkers: finding periodic gaits in the presence of discontinuities. *Nonlinear Dynamics*, 0(2):205–229, 2001.

[12] J-Courtney-Pratt and E. Eisner. The effect of tangential force on the contact of metallic bodies. In *Royal Society*, pages 529–550, 1957.

[13] Paul Fisette. Jean-Claude Samin. *Symbolic Modeling of Multiboyd System.* Kluwer Academic Publishers, 2009.

[14] Christine Chevallereau JunHo Choi Benjamin Morris Jessy W. Grizzle, Eric R. Westervelt. *Feedback Control of Dynamical Bipedal Robot Locomotion.* Boca Raton, CRC Press, 2007.

[15] G. Abba J.W. Grizzle and F. Plestan. Asymptotically stable walking for biped robots: Analysis via systems with impulse effects. In *IEEE Transactions on Automatic Control*, pages 46:51–64, January 2001.

[16] R.A. Liston and Mosher R.S. A versatile walking truck. In *In Proceedings of the Transportation Engineering Conference*, Institution of Civil Engineers, London,, 1968.

[17] Duane W. Marhefka and David E. Orin. Simulation of contact using a nonlinear damping model. In *International Conference on Robotics and Automation*, pages 1662–1668, April 1996.

[18] T. McGeer. Passive dynamic walking. *International Journal of Robotics Research*, 2:62–68, 1990.

[19] J.E Pratt. *Exploiting inherent robustness and natural dynamics in the control of bipedal walking robots.* PhD thesis, MIT, 2000.

[20] M.H. Raibert. *Legged Robots that Balance.* MIT Press, Cambridge, MA, 1986.

[21] Laurence Roussel. *Gï¿½nï¿½ration de trajectoires de marche optimales pour un robot bipï¿½de.* PhD thesis, Institut National Polytechnique de Grenoble, 1998.

[22] Laurent Sass. *Symbolic Modeling of Electromechanical Multibody Systems.* PhD thesis, Universitï¿½ Catholique de Louvain, 2004.

[23] M.W. Spong and M. Vidyasagar. *Robot Dynamics and Control.* Wiley, New York, 1991.

[24] Universite Catholique de Louvain UCL. *Modeling Multibody Systems with ROBOTRAN*, 2009.

[25] Biped humanoid robot group. Takanishi Laboratory, March 2009. [online] http://www.takanishi.mech.waseda.ac.jp/top/research/wabian/index.htm.

[26] Eric R. Westervelt. *Toward a Coherent Framework for the Control of Planar Biped Locomotion.* PhD thesis, University of Michigan, 2003.