



Università degli Studi dell'Aquila

**Dipartimento di Ingegneria e Scienze
dell'Informazione e Matematica**



**Corso di Laurea Magistrale in
Ingegneria Informatica e Automatica**

**Modeling and nonlinear closed-loop gait control
of humanoid robots**

Relatore

Prof. Costanzo Manes

Studente

Federico Cecati

Matricola

244872

A.A. 2016/2017

Ai miei Genitori

che mi hanno insegnato l'armonia e il coraggio

A mio Padre che mi ha insegnato la tenacia e l'originalità

A mia Madre che mi ha insegnato l'empatia e la serenità

Alla Musica

che mi accompagna da sempre

che mi ascolta e parla per me

Ringrazio sentitamente il Professor Manes che mi ha guidato nel lavoro di tesi
sostenendomi sia dal punto di vista ingegneristico nei calcoli e nelle simulazioni
che dal punto di vista psicologico nei momenti in cui credevo di non farcela

Amo coloro che non cercano dietro alle stelle una ragione per tramontare ed
offrirsi in sacrificio, ma coloro che si sacrificano alla terra, perchè questa
appartenga un giorno al Superuomo

Frederich Nietzsche

Passare troppo tempo a studiare è pigrizia

Francis Bacon

La nostra immaginazione è tesa al massimo: non, come nelle storie
fantascientifiche, per immaginare cose che in realtà non esistono, ma proprio
per comprendere ciò che davvero esiste

Richard Feynman

Le persone sono come le biciclette: riescono a mantenere l'equilibrio solo se
continuano a muoversi

Albert Einstein

List of Figures

1.0.1 Examples of robots	11
1.0.2 Tulip	12
1.1.1 Sagittal, frontal, transversal planes	13
1.2.1 Typical shapes of the Support Polygon (SP), in grey: (a) Double Support (b) Double Support (Pre-Swing Phase). (c) Single Support	15
1.2.2 Phasing of a periodic gait	16
1.3.1 Humanoid 5DOF model	17
2.1.1 Kinematic review: position, velocity, acceleration	20
2.8.1 On the left Leonard Euler, on the right Joseph Louis Lagrange .	34
2.8.2 Euler-Lagrange equation in the non-forced case	35
2.8.3 Humanoid robots	41
3.1.1 Robot planes	48
3.1.2 Humanoid GRF and CMP(equivalent to A)	50
3.1.3 Main points review	51
3.2.1 From left: Ankle strategy, Hip strategy, Stepping	52
3.3.1 The Linear Inverted Pendulum Model	54

3.3.2 Capture region	59
4.1.1 Jacobian Transpose control scheme	68
4.2.1 Physical interpretation of penalty function described with equation (4.2.14)	73
4.3.1 Industrial robot Kinematic	76
4.3.2 Kinematic chain of an industrial robot	77
5.1.1 Relative coordinates on the left (a) and absolute coordinates on the right (b)	81
5.2.1 The length of femur L_3 and the length of the tibia L_4	89
5.2.2 Graphic of the variables defined in (5.2.8)	90
6.0.1 Walking Animation	106
6.0.2 Output functions plots	107
6.0.3 Joint velocities and accelerations	108

Contents

1	Introduction	9
1.1	Biped Locomotion fundamentals	11
1.2	Gait analysis	13
1.3	Overview of the thesis	16
2	Modeling	19
2.1	Kinematics	20
2.2	Inverse Kinematics	22
2.3	Differential Kinematics	23
2.4	Calculus of Variation and Dynamics	25
2.5	Basic Calculus of Variation problem	26
2.6	First-order necessary conditions for weak extrema	27
2.6.1	Euler-Lagrange equation	28
2.7	Principle of least action and conservation laws	32
2.8	Lagrange mechanics	34
2.9	Hybrid model	41
3	Gait Analysis	45

3.1	Definitions and Concepts	46
3.2	Push recovery	50
3.3	The Linear Inverted Pendulum Model	54
3.3.1	Without flywheel	55
3.3.2	With flywheel	56
3.3.3	Capture Point Dynamics	58
3.3.4	Model Predictive Control	60
4	Motion Control Techniques	65
4.1	Transpose Jacobian PD	66
4.2	Walking primitives and inverse kinematic based techniques	69
4.2.1	Walking primitives	69
4.2.2	Inverse differential kinematic solution	70
4.3	Humanoid robot control challenges	74
5	Gait Control	79
5.1	Relative coordinates and absolute coordinates	80
5.2	Feedback Linearization	85
5.2.1	The output function choice	89
5.2.2	Controller Synthesis	93
5.3	Impact model and gait stability	95
5.3.1	Impact model	97
5.3.2	Poincaré map	101
6	Simulation Results and Conclusions	105

Chapter 1

Introduction

Legged robots are very diverse, often subdivided in groups indicated by the number of legs. If a legged robot has two legs it is often called a biped. Bipedes are supposed to mimic human like walking. They usually consist of rigid bodies interconnected with joints that are actively or passively actuated.

The ground-breaking works in the field of bipeds were accomplished around 1970 by two famous researchers, Kato and Vukobratovic. In Japan, the first anthropomorphic robot, WABOT 1, was demonstrated in 1973 by Kato at Waseda University. Using a very simple control scheme, it was able to realize a few slow steps, being statically stable. This achievement was the starting point of a prolific generation of bipeds in Japan. Parallel to this research, Vukobratovic and his team were very involved in the problems generated by functional

rehabilitation. In Belgrade Yugoslavia, his team designed the first active exoskeletons, but the most well-known outcome remains their analysis of locomotion stability, which exhibited around 1972 the concept of the Zero Moment Point, widely used since then. The Zero-Moment Point criterion takes the dynamical effects during walking into consideration; therefore it is an extension to the static stability criterion that was used by Kato. The exact interpretation and consequences of this criterion are explained in detail in chapter 3.

Dutch Robotics and TULip Research groups at three universities of technology in the Netherlands, being University of Twente, Delft University of Technology and Eindhoven University of Technology, have agreed to join efforts in creating humanoid robots, cooperating in the Dutch Robotics initiative. The Dutch Robotics project is part of a long term vision, shared by three Dutch universities and the Dutch industry for the development of a new generation of robots. Together they designed TULip, shown in Figure 1.0.2, a biped robot, especially designed for mimicking human like walking. TULip is about 1.2 meters tall and consists of fourteen degrees of freedom (DoFs). Two DoFs in its arms and twelve DoFs in its legs out of which two are unactuated. The joints all are rotational and are actuated with DC motors. Planetary gears are used for the transfer of the driving power and some joints are actuated by series elastic actuation: an electric motor drives a joint through an elastic/compliant element. By measuring the elongation of this element, the torque that the total actuator system delivers to the joint can be controlled.

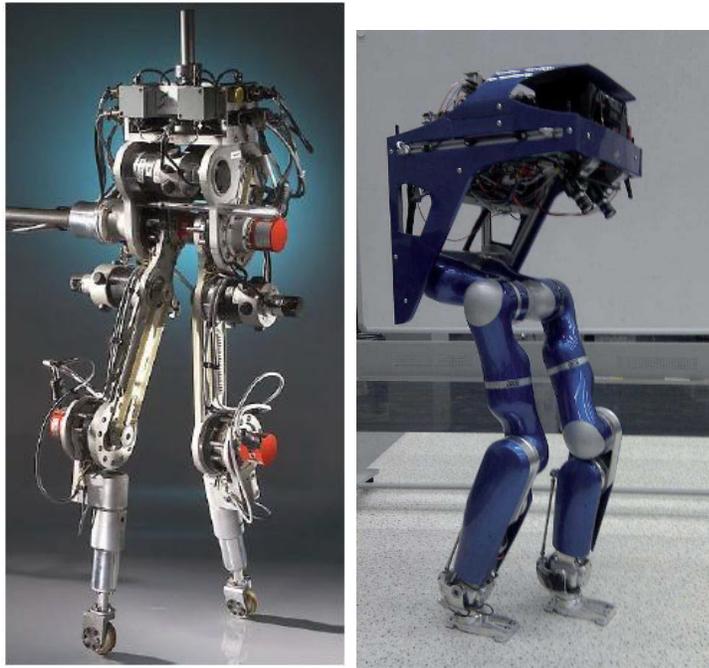


Figure 1.0.1: Examples of robots

1.1 Biped Locomotion fundamentals

Three Dimensional Motion Positioning a biped in three dimensional space can be done with a base-frame-origin and three planes that are all perpendicular to each other. To visualize this, the anatomical position is depicted in figure 1.1.1. The anatomical position is the standing position with the face turned straight forward and the arms hanging along the sides of the body with the palms turned straight forward and the legs stretched with the feet close together. Occasionally, in robot anatomies this anatomical position is with the palms pointing inwards to the body. Motions can be described relative to three perpendicular planes through the body. These planes are also depicted in figure 1.1.1, as well as the base-frame-origin, that is placed with the x-axis pointing forward, the y-axis pointing to the left-hand side, and the z-axis pointing

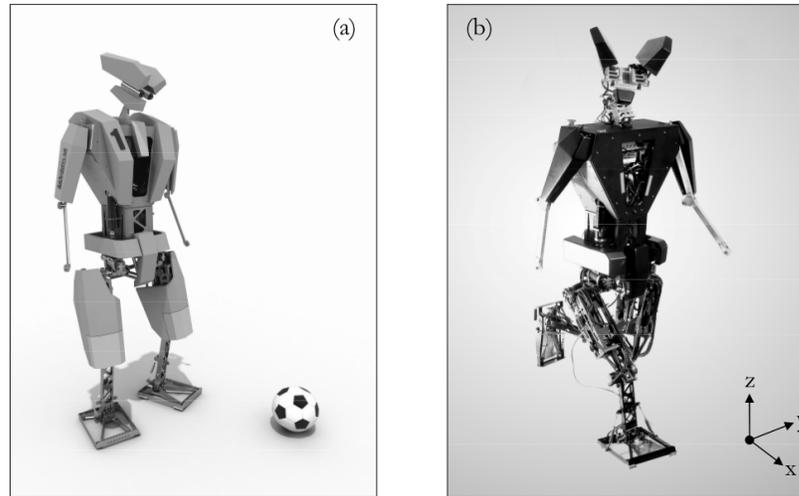


Figure 1.0.2: Tulip

upwards. The origin of this base-frame-origin is at the floor. The definitions of these planes are:

- **Frontal plane** The plane parallel to the yz -plane is called the Frontal plane.
- **Sagittal plane** The plane parallel to the xz -plane is called the Sagittal plane.
- **Transverse plane** The plane parallel to the xy -plane is called the Transverse plane.

Moreover the plane parallel to the Sagittal plane and containing the Center of Mass (CoM) is called the Median plane.

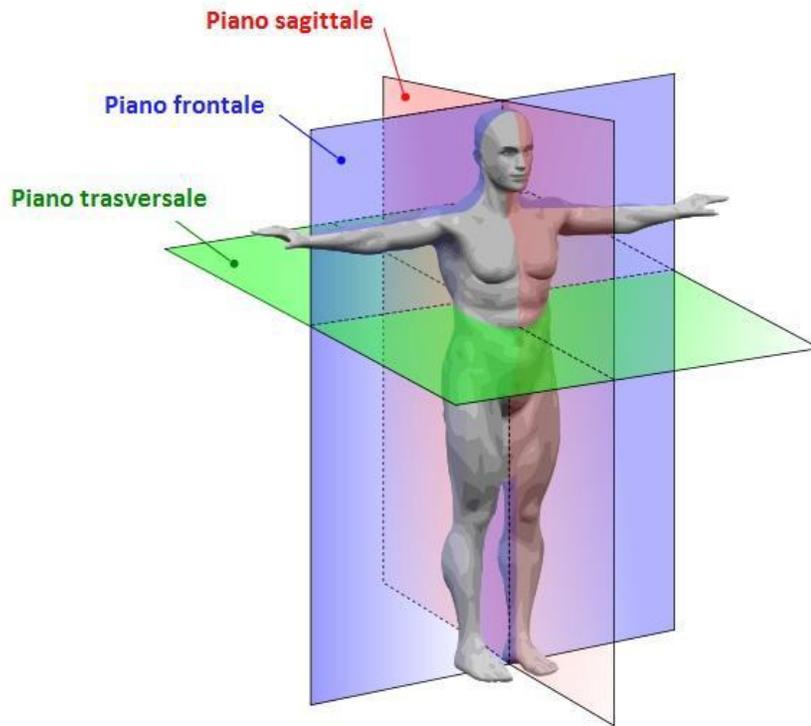


Figure 1.1.1: Sagittal, frontal, transversal planes

1.2 Gait analysis

Describing human gait requires some specific terms, which are defined in this section. Some key terms with respect to biped locomotion from [3] are:

- **Walk:** walk is defined as: “Movement by putting forward each foot in turn, not having both feet off the ground at once.” Walking backwards and running are not taken into consideration in this report.
- **Gait:** every human has a specific unique walk, hence gait means: “Manner of walking or running”. Moreover, every walk is realized with a certain gait.
- **Periodic gait:** if the gait is realized by repeating each step in an identical

way, it is a periodic gait.

- **Double Support (DS):** this term is used for situations where the biped has two isolated contact surfaces with the floor. This situation occurs when the biped is supported by both feet, but it is not necessarily that both feet are fully supported with the floor, (see figure 1.2.1 (b)).
- **Single Support (SS):** this term is used for situations where the biped has only one contact surface with the floor. This situation occurs when the biped is supported with only one foot as in figure 1.2.1 (c).
- **Support Polygon (SP):** the Support Polygon is formed by the convex hull about the floor support points. This term is widely accepted for any support area and is shown in figure 1.2.1 (a) and (b). The convex hull is the boundary of the minimal convex set containing a given non-empty finite set of points in the plane.
- **Swing leg:** the leg that is performing a step, i.e. moving forward through the air, is denoted with the term swing leg. The foot that is attached to this leg is called the swing foot.
- **Stance leg:** while the swing leg is moving through the air, the stance leg is fully supported with the floor by the stance foot and supports all the weight of the biped.
- **Gait Phases** When the biped is in periodic gait, the gait can be divided into four phases:
 - **Double Support Phase (DSP):** this is the phase where both feet are fully supported with the floor, depicted in Figure 1.2.1(a)

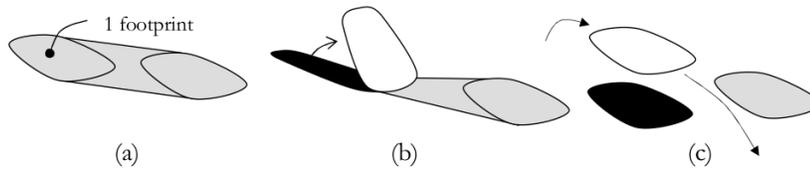


Figure 1.2.1: Typical shapes of the Support Polygon (SP), in grey: (a) Double Support (b) Double Support (Pre-Swing Phase). (c) Single Support

- **Pre-Swing Phase:** in this phase the heel of the rear foot is lifting from the floor but the biped is still in double support due to the fact that the toes of this foot are still on the floor as depicted in Figure 1.2.1(b)
- **Single Support Phase (SSP):** the phase where only one foot is fully supported with the floor and the other foot swings forward, depicted in Figure 1.2.1(c)
- **Post-Swing Phase:** in this phase the toe of the front foot is declining towards the floor. The biped is in double support because the heel of this foot is contacting the floor.

These four phases for each leg form a walking gait; this is depicted in Figure 1.2.2

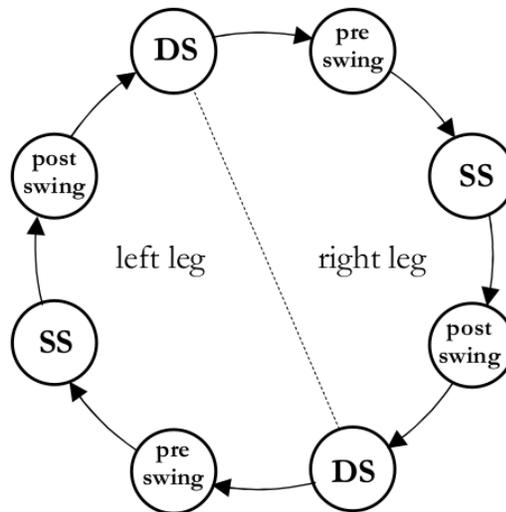


Figure 1.2.2: Phasing of a periodic gait

1.3 Overview of the thesis

In this thesis the model used is a simple bidimensional 5 link robot shown in figure 1.3.1. The model deduced is based on Lagrange mechanics and hybrid system theory, and is described in chapter 2. In chapter 3 basic definitions and concepts of human locomotion are analyzed and used for the problem of Push Recovery, solved with a Model Predictive Control. Push Recovery is a different problem respect to gait control, but it's important to analyze because many concepts of it are important to better understand closed-loop gait control. Then fundamental motion control techniques are illustrated in chapter 4 going from a Lyapunov-based approach to a precomputed trajectory one. These approaches are demonstrated not to be the most performing ones; at the end of the chapter the most relevant problems of humanoid gait control are presented and it emerges the definitive control strategy able to overtake them: the feedback linearization. It is analyzed in chapter 5 with detailed explana-

tion. However this strategy is useful just for the single-step motion control: to extend the controller to multiple-step gait it's necessary to consider the impact model to compute the reset map associated to the hybrid automaton. Overall to study the stability of the walking orbits Poincarè-analysis must be exploited. The last chapter is about simulation result, and shows the good performance of feedback linearization controller.

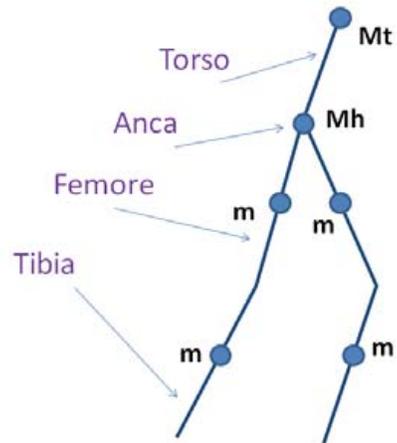


Figure 1.3.1: Humanoid 5DOF model

Chapter 2

Modeling

This chapter is a path from the basics of robotics to the description of the complete hybrid model of the humanoid robot. The differential equation which describes the robot behavior comes from the dynamic model deduced from the solution of Euler-Lagrange equation. Kinematics and differential kinematics are necessary to express positions and velocities of the arms of the robot as a function of joints position and velocity, to include them in Euler-Lagrange equation.

2.1 Kinematics

In robotics is always useful to find the relationship between the position of a joint and the robot configuration. For this reason it's necessary to define the kinematic function $k(\cdot)$. Given the robot configuration expressed with five generalized coordinates $q = (q_1 \ q_2 \ q_3 \ q_4 \ q_5)^T$ where $q_i \ i = 1 \dots 5$ are the joint variable there exists one kinematic function for each joint of the form

$$x_i = k_i(q) \quad (2.1.1)$$

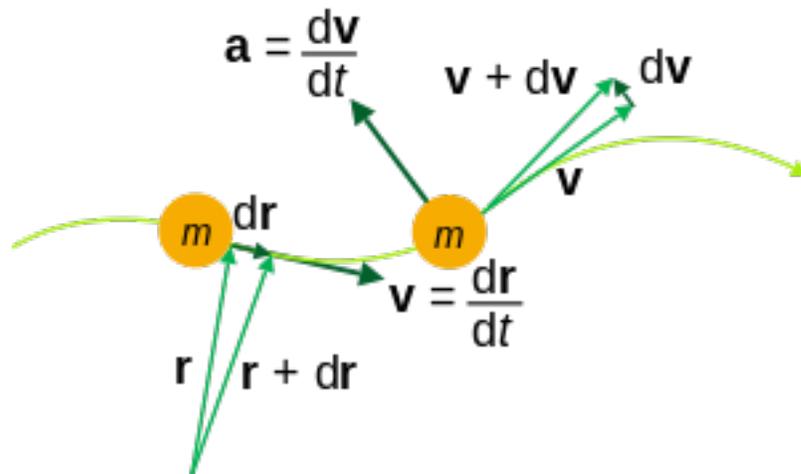


Figure 2.1.1: Kinematic review: position, velocity, acceleration

where x_i is the Cartesian position of the $i - th$ joint.

Given $L_1 \ L_2 \ L_t$ respectively the length of the calf, the thigh and the torso, the kinematic functions of the humanoid robot are:

$$\begin{cases}
k_1(q) = p_f + L_1 \begin{pmatrix} \cos q_1 \\ \sin q_1 \end{pmatrix} \\
k_2(q) = p_f + L_1 \begin{pmatrix} \cos q_1 \\ \sin q_1 \end{pmatrix} + L_2 \begin{pmatrix} \cos(q_1 + q_2) \\ \sin(q_1 + q_2) \end{pmatrix} \\
k_t(q) = p_f + L_1 \begin{pmatrix} \cos q_1 \\ \sin q_1 \end{pmatrix} + L_2 \begin{pmatrix} \cos(q_1 + q_2) \\ \sin(q_1 + q_2) \end{pmatrix} + L_t \begin{pmatrix} \cos(q_1 + q_2 + q_3) \\ \sin(q_1 + q_2 + q_3) \end{pmatrix} \\
k_4(q) = p_f + L_1 \begin{pmatrix} \cos q_1 \\ \sin q_1 \end{pmatrix} + L_2 \begin{pmatrix} \cos(q_1 + q_2) \\ \sin(q_1 + q_2) \end{pmatrix} + L_4 \begin{pmatrix} \cos(q_1 + q_2 + q_4) \\ \sin(q_1 + q_2 + q_4) \end{pmatrix} \\
k_5(q) = p_f + L_1 \begin{pmatrix} \cos q_1 \\ \sin q_1 \end{pmatrix} + L_2 \begin{pmatrix} \cos(q_1 + q_2) \\ \sin(q_1 + q_2) \end{pmatrix} + L_4 \begin{pmatrix} \cos(q_1 + q_2 + q_4) \\ \sin(q_1 + q_2 + q_4) \end{pmatrix} + L_5 \begin{pmatrix} \cos(q_1 + q_2 + q_4 + q_5) \\ \sin(q_1 + q_2 + q_4 + q_5) \end{pmatrix}
\end{cases} \quad (2.1.2)$$

Kinematic function aren't just related to joints. Indeed they can be defined for each point of the robot whom position depends on the joint configuration. Usually for industrial manipulators kinematic functions are defined only for joints because these are important point for the study of the robot. In the case of humanoid robot there are also other points to be declared that are fundamental for the control algorithm project; one of these point is the Center of Mass that is defined as the unique point where if a force is applied it doesn't generate momentum. In this model it has been considered, as an approximation, each robot arms as a segment, so that the Center of Mass of each arm is located on the medium point of the segment. Therefore it's possible to define a kinematic function for the Center of Mass of each arm;

$$\begin{cases}
k_{CoM1}(q) = p_f + \frac{L_1}{2} \begin{pmatrix} \cos q_1 \\ \sin q_1 \end{pmatrix} \\
k_{CoM2}(q) = p_f + L_1 \begin{pmatrix} \cos q_1 \\ \sin q_1 \end{pmatrix} + \frac{L_2}{2} \begin{pmatrix} \cos(q_1 + q_2) \\ \sin(q_1 + q_2) \end{pmatrix} \\
k_{CoMt}(q) = p_f + L_1 \begin{pmatrix} \cos q_1 \\ \sin q_1 \end{pmatrix} + L_2 \begin{pmatrix} \cos(q_1 + q_2) \\ \sin(q_1 + q_2) \end{pmatrix} + \frac{L_t}{2} \begin{pmatrix} \cos(q_1 + q_2 + q_3) \\ \sin(q_1 + q_2 + q_3) \end{pmatrix} \\
k_{CoM4}(q) = p_f + L_1 \begin{pmatrix} \cos q_1 \\ \sin q_1 \end{pmatrix} + L_2 \begin{pmatrix} \cos(q_1 + q_2) \\ \sin(q_1 + q_2) \end{pmatrix} + \frac{L_4}{2} \begin{pmatrix} \cos(q_1 + q_2 + q_4) \\ \sin(q_1 + q_2 + q_4) \end{pmatrix} \\
k_{CoM5}(q) = p_f + L_1 \begin{pmatrix} \cos q_1 \\ \sin q_1 \end{pmatrix} + L_2 \begin{pmatrix} \cos(q_1 + q_2) \\ \sin(q_1 + q_2) \end{pmatrix} + L_4 \begin{pmatrix} \cos(q_1 + q_2 + q_4) \\ \sin(q_1 + q_2 + q_4) \end{pmatrix} + \frac{L_5}{2} \begin{pmatrix} \cos(q_1 + q_2 + q_4 + q_5) \\ \sin(q_1 + q_2 + q_4 + q_5) \end{pmatrix}
\end{cases} \quad (2.1.3)$$

Now, starting from these five formulas, it's possible to write the expression of the Center of Mass of the whole humanoid robot in a compact form:

$$k_{CoM}(q) = \frac{m_1 k_{CoM1}(q) + m_2 k_{CoM2}(q) + m_t k_{CoMt}(q) + m_4 k_{CoM4}(q) + m_5 k_{CoM5}(q)}{m_1 + m_2 + m_t + m_4 + m_5} \quad (2.1.4)$$

The important feature of the Center of Mass kinematic function is that it's not referred to a concrete and tangible point of the mechanical structure, as in the other kinematic functions defined above, but to an abstract and non tangible point, that is not constrained to the mechanical structure of the manipulator, and whose position depends on the robot configuration, and can be both internal and external to the mechanical structure.

2.2 Inverse Kinematics

Equation (2.1.1) defines the relationship between joints position and Cartesian position of a given point of the robot. This function is also called the direct kinematic function in order to highlight the difference with the inverse kinematic function that defines the relation between Cartesian position and joints position. The solution of inverse kinematics is very useful for the determination of the trajectory on the joint space that realizes a given trajectory on the operative space.

However inverse kinematic presents some problems that direct kinematic doesn't:

- equations are nonlinear and it's often required a numeric algorithm, because the analytic solution could not exist
- there is the possibility of multiple solutions
- there is the possibility of infinite solutions in the case of the redundancy

- it may exist non feasible solutions

The redundancy is a property of the structure of the robot: a robot is redundant if its degrees of freedom are more than the order of the space on which the duty is defined.

There exist many numerical algorithms for the solution of inverse kinematics that have not been dealt in this thesis. In chapter 4 it has been discussed the algorithm for the solution of inverse differential kinematics in subsection 4.2.2, that are useful for pre-computed trajectory tracking.

2.3 Differential Kinematics

Once kinematic function has been defined, it's useful to find a relationship between \dot{q} and \dot{x} . From the kinematic equation (2.1.1) by differentiation it results:

$$\dot{x} = \frac{dx_i}{dt} = \frac{\partial k_i(q)}{\partial q} \frac{dq}{dt} = J_i(q) \dot{q} \quad (2.3.1)$$

that is the differential kinematic relation. The matrix J_i is called the Jacobian matrix referred to the i -th joint. The Jacobian matrix of the Center of Mass, obtained from the derivative $\frac{\partial k_{CoM_i}(q)}{\partial q}$, can be defined as $J_{CoM,i}$ and expresses the relationship between the joints velocity and the Cartesian velocity of the Center of Mass of the i -th joint. Thus as in the case of kinematic function, Jacobian can be defined for each point, and each time it exists a kinematic function as-

sociated to a certain point, it exists the relative Jacobian matrix as well. It's important to notice that the existence of the Jacobian matrix highlights that the instantaneous relationship between joints velocity and Cartesian velocity is linear. That is true only for an arbitrary small neighborhood of the current joint position q for the fact that $J(q)$ depends on it.

Jacobian matrices dealt above are related to the linear velocity of a certain point: it can be defined also an angular Jacobian matrix \tilde{J}_i that expresses the relationship between the angular velocity ω_i of the i -th arm and the joints velocity \dot{q} . In the three-dimensional case the computation of those matrices is quite laborious, while in the bi-dimensional case it's very quickly and it results that they are row vectors independent of q : in fact in the planar case of humanoid robot the angular velocity of a certain arm is scalar and it's obtained by the sum of the angular velocity of each joints that come before the given one, considering the stance foot as the basement. So the expression of the angular velocities is:

$$\left\{ \begin{array}{l} \omega_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \end{pmatrix} \dot{q} \quad \tilde{J}_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \end{pmatrix} \\ \omega_2 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \end{pmatrix} \dot{q} \quad \tilde{J}_2 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \end{pmatrix} \\ \omega_t = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \end{pmatrix} \dot{q} \quad \tilde{J}_t = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \end{pmatrix} \\ \omega_4 = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \end{pmatrix} \dot{q} \quad \tilde{J}_4 = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \end{pmatrix} \\ \omega_5 = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 \end{pmatrix} \dot{q} \quad \tilde{J}_5 = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 \end{pmatrix} \end{array} \right. \quad (2.3.2)$$

Jacobian defined in (2.3.1) can be denominated as linear Jacobian to better distinguish it from the angular Jacobian defined in equation (2.3.2).

As well as linear and angular velocities, expression of linear and angular acceleration can be useful in many cases, and can be obtained by a second dif-

ferentiation of (2.3.1) and (2.3.2) using the product derivative rule.

$$\ddot{x}_i = J_i \ddot{q} + \dot{J}_i \dot{q} \quad (2.3.3)$$

$$\alpha_i = J_i \ddot{q} \quad (2.3.4)$$

2.4 Calculus of Variation and Dynamics

After studying kinematics, it's necessary to introduce a new important chapter for the description of robots. The dynamic model provides to find the relationship between torques and joint accelerations. Classical mechanics has two fundamental laws, that are the force and the torque relations:

$$\begin{cases} F = ma \\ \tau = I\alpha \end{cases} \quad (2.4.1)$$

where F is the force, m the mass, a the linear acceleration, τ the torque, I the inertia and α the angular acceleration. The greatest contribution to this part of mechanics is mainly attributed to Isaac Newton. In robotics it's more convenient to use a latter approach, developed by Euler and Lagrange. This approach has its roots on calculus of variation and Euler Lagrange equation: from a particular choice of the Lagrange function, Euler Lagrange equation coincides with Newton laws. This method has great advantages in robot modeling, because it's not necessary to consider reaction forces and constraints, it's just necessary to compute the Lagrange function.

Calculus of variations is a field of mathematical analysis that deals with maximizing or minimizing functionals, which are mappings from a set of functions to the real numbers. Functionals are often expressed as definite integrals involving functions and their derivatives. The interest is in extremal functions that make the functional attain a maximum or minimum value – or stationary functions – those where the rate of change of the functional is zero. For this aim it's necessary to define the first variation: given a functional $J(y)$ the first variation $\delta J(y)$ is defined as

$$\delta J|_y(\eta) = \lim_{\alpha \rightarrow 0} \frac{J(y + \alpha\eta) - J(y)}{\alpha} \quad (2.4.2)$$

When the first variation is 0 the functional attain a stationary function: this condition is known as first-order necessary condition for optimality.

It's now given an overview of Calculus of Variation and it's derived the Euler Lagrange equation, with the demonstration provided by Daniel Liberzon [7].

2.5 Basic Calculus of Variation problem

The basic Calculus of Variation problem is to find among all C^1 curves $y : [a, b] \rightarrow \mathbb{R}$ satisfying given boundary conditions

$$y(a) = y_0 \quad y(b) = y_1$$

the local minima of a cost functional of the form

$$J(y) := \int_a^b L(x, y(x), \dot{y}(x)) dx$$

Since y takes values in \mathbb{R} , it represents a single planar curve connecting the two fixed points (a, y_0) and (b, y_1) . This is the single-degree-of-freedom case. In the multiple-degrees-of-freedom case, one has $y : [a, b] \rightarrow \mathbb{R}^n$ and accordingly $L : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$. This generalization is useful for treating spatial curves ($n = 3$) or for describing the motion of many particles and, in the case of humanoid robots, to joint space trajectories ($n = 5$); the latter setting was originally proposed by Lagrange in his 1788 monograph *Mécanique Analytique*. The assumption that $y \in C^1$ is made to ensure that J is well defined. The function L is called the Lagrangian, or the running cost. It is clear that a maximization problem can always be converted into a minimization problem by flipping the sign of L . In the analysis that follows, it will be important to remember that even though y and \dot{y} are the position and velocity along the curve, L must be considered as a function of three independent variables. To emphasize this fact, the Lagrangian is expressed with three distinct variables $L = L(x, y, z)$. When deriving optimality conditions, it's necessary to impose some differentiability assumptions on L .

2.6 First-order necessary conditions for weak extrema

In this section it will be derived the most fundamental result in calculus of variations: the Euler-Lagrange equation. Unless stated otherwise, it will be working with the Basic Calculus of Variations. Thus the function space V is

$\mathcal{C}^1([a, b], \mathbb{R})$, the subset A consists of functions $y \in V$ satisfying the boundary conditions

$$y(a) = y_0 \quad y(b) = y_1 \quad (2.6.1)$$

and the functional J to be minimized takes the form

$$J(y) := \int_a^b L(x, y(x), \dot{y}(x)) dx \quad (2.6.2)$$

The Euler-Lagrange equation provides a more explicit characterization of the first-order necessary condition for optimality for this situation. In deriving the Euler-Lagrange equation, it has been followed the basic variational approach, considering nearby curves of the form

$$y + \alpha\eta \quad (2.6.3)$$

where the perturbation $\eta : [a, b] \rightarrow \mathbb{R}$ is another \mathcal{C}^1 curve and α varies in an interval around 0 in \mathbb{R} . For α close to 0, these perturbed curves are close to y in the sense of the 1-norm.

2.6.1 Euler-Lagrange equation

It's now convenient to introduce the notational convention that denotes by $L_x, L_y, L_z, L_{xx}, L_{xy}$ the partial derivatives of the Lagrangian $L = L(x, y, z)$. It's assumed that all derivatives appearing in that calculations exist and are continuous. Let $y = y(x)$ be a given test curve in A . For a perturbation η in (2.6.3), to be admissible the new curve (2.6.3) must again satisfy the boundary conditions

(2.6.1). Clearly, this is true if and only if

$$\eta(a) = \eta(b) = 0 \quad (2.6.4)$$

In other words, all the perturbations must vanish at the endpoints. Now, the first-order necessary condition says that if y is a local extremum of J , then for every η satisfying (2.6.4) it must be $\delta J|_y(\eta) = 0$. The first variation $\delta J|_y$ is such that

$$J(y + \alpha\eta) = J(y) + \delta J|_y(\eta)\alpha + o(\alpha) \quad (2.6.5)$$

The left-hand side of (2.6.5) is

$$J(y + \alpha\eta) = \int_a^b L(x, y(x) + \alpha\eta(x), \dot{y}(x) + \alpha\dot{\eta}(x)) dx \quad (2.6.6)$$

It can be written its first-order Taylor expansion with respect to α by expanding the expression inside the integral with the help of the chain rule:

$$J(y + \alpha\eta) = \int_a^b (L(x, y(x), \dot{y}(x)) + L_y(x, y(x), \dot{y}(x))\alpha\eta(x) + L_z(x, y(x), \dot{y}(x))\alpha\dot{\eta}(x) + o(\alpha)) dx \quad (2.6.7)$$

Matching this with the right-hand side of (2.6.5), we deduce that the first variation is

$$\delta J|_y(\eta)\alpha = \int_a^b (L_y(x, y(x), \dot{y}(x))\eta(x) + L_z(x, y(x), \dot{y}(x))\dot{\eta}(x)) dx \quad (2.6.8)$$

Note that, proceeding slightly differently, it's possible to arrive at the same result by remembering that

$$\delta J|_y(\eta) = \lim_{\alpha \rightarrow 0} \frac{J(y + \alpha\eta) - J(y)}{\alpha} = \frac{d}{d\alpha} \Big|_{\alpha=0} J(y + \alpha\eta) \quad (2.6.9)$$

and using differentiation under the integral sign on the right-hand side of

(2.6.6).

The first variation depends not just on η but also on $\dot{\eta}$: this is not surprising since L has \dot{y} as one of its arguments. However, it's possible to eliminate the dependence on $\dot{\eta}$ by applying integration by parts to the second term on the right-hand side of (2.6.8):

$$\delta J|_y(\eta) = \int_a^b \left(L_y(x, y(x), \dot{y}(x)) \eta(x) - \frac{d}{dx} L_z(x, y(x), \dot{y}(x)) \eta(x) \right) dx + L_z(x, y(x), \dot{y}(x)) \eta(x) \Big|_a^b \quad (2.6.10)$$

where the last term is 0 when η satisfies the boundary conditions (2.6.4). Thus if y is an extremum, the relation

$$\int_a^b \left(L_y(x, y(x), y'(x)) - \frac{d}{dx} L_z(x, y(x), y'(x)) \right) \eta(x) dx \quad (2.6.11)$$

is true for all \mathcal{C}^1 curves η vanishing at the endpoints $x = a$ and $x = b$. The condition (2.6.11) does not yet give a practically useful test for optimality, because it must be checked for all admissible perturbations η . However, it is logical to suspect that the only way (2.6.11) can hold is if the term inside the parentheses which does not depend on η equals 0 for all x .

The next lemma shows that this is indeed the case.

Lemma 2.5.1 *If a continuous function $\zeta : [a, b] \rightarrow \mathbb{R}$ is such that*

$$\int_a^b \zeta(x) \eta(x) dx = 0$$

for all \mathcal{C}^1 functions $\eta : [a, b] \rightarrow \mathbb{R}$ with $\eta(a) = \eta(b) = 0$, then $\zeta \equiv 0$.

Proof. Suppose that $\zeta(\bar{x}) \neq 0$ for some $\bar{x} \in [a, b]$. By continuity, ζ is then nonzero and maintains the same sign on some subinterval $[c, d]$ containing \bar{x} . Just for concreteness, ζ is supposed positive on $[c, d]$. Construct a function

$\eta \in C^1([a, b], \mathbb{R})$ that is positive on (c, d) and 0 everywhere else for example $\eta(x) = (x - c)^2(x - d)^2$ for $x \in [c, d]$ and $\eta(x) = 0$ otherwise. This gives $\int_a^b \zeta(x) \eta(x) dx > 0$, that is a contradiction. \square

It follows from (2.6.11) and Lemma 2.6.1 that for $y(\cdot)$ to be an extremum, a necessary condition is

$$L_y(x, y(x), y'(x)) = \frac{d}{dx} L_z(x, y(x), \dot{y}(x)) \quad \forall x \in [a, b] \quad (2.6.12)$$

This is the celebrated Euler-Lagrange equation providing the first-order necessary condition for optimality. It is often written in the shorter form

$$L_y = \frac{d}{dx} L_{\dot{y}} \quad (2.6.13)$$

We must keep in mind, however, that the correct interpretation of the Euler-Lagrange equation is 2.6.12: y and \dot{y} are treated as independent variables when computing the partial derivatives L_y and $L_{\dot{y}}$, then one plugs in for these variables the position $y(x)$ and velocity $\dot{y}(x)$ of the curve, and finally the differentiation with respect to x is performed using the chain rule. Written out in detail, the right-hand side of (2.6.12) is

$$\frac{d}{dx} L_z(x, y(x), \dot{y}(x)) = L_{zx}(x, y(x), \dot{y}(x)) + L_{zy}(x, y(x), \dot{y}(x)) \dot{y}(x) + L_{zz}(x, y(x), \dot{y}(x)) \ddot{y}(x) \quad (2.6.14)$$

2.7 Principle of least action and conservation laws

Newton's second law of motion in the three-dimensional space can be written as the vector equation

$$\frac{d}{dt}(m\dot{q}) = -U_q \quad (2.7.1)$$

where $q = (x, y, z)^T$ is the vector of coordinates, $\dot{q} = \frac{dq}{dt}$ is the velocity vector, and $U = U(q)$ is the potential; consequently, $m\dot{q}$ is the momentum and $-U_q$ is the force: since the force is conservative, it corresponds to the negative gradient of some potential function. Planar motion is obtained as a special case by dropping the z -coordinate. It turns out that there is a direct relationship between (2.7.1) and the Euler-Lagrange equation in (2.6.13). This is difficult to see right now because the notation in (2.7.1) is very different from the one it has been used. First, the independent variable x in this case represents the time, so has to be replaced by t . Second, the dependent variable y is now the joint position q , and as a consequence \dot{y} is the joint velocity \dot{q} . Equation (2.7.1) becomes

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) = \frac{\partial L}{\partial q} \quad (2.7.2)$$

By defining the Lagrangian function as the difference between the kinetic energy $T = \frac{1}{2}m(\dot{q} \cdot \dot{q})$ and the potential energy $U(q)$

$$L = T(\dot{q}) - U(q) \quad (2.7.3)$$

Since $T(\dot{q})$ depends only on \dot{q} and $U(q)$ depends only on q , equation (2.7.2) becomes

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}} \right) = - \frac{\partial U}{\partial q} \quad (2.7.4)$$

that is

$$\frac{d}{dt} (m\dot{q}) = - \frac{\partial U}{\partial q} \quad (2.7.5)$$

$$m\ddot{q} = - \frac{\partial U}{\partial q} \quad (2.7.6)$$

The left side term is the product of the mass for the acceleration that is the force. The right side term is the gradient of the potential of the gravitational field, that for the theory of conservative vector fields is equal to the gravitational force. Thus (2.7.6) becomes

$$m\ddot{q} = g$$

that represent second Newton's law in absence of nonconservative forces.

From this result it comes that Newton's equations of motion can be considered as a path optimization problem on which the functional to minimize is the integral of the Lagrangian function which is called the action integral.

$$\int_{t_0}^{t_1} (T - U) dt \quad (2.7.7)$$

So trajectories of mechanical systems are extremals, in particular minima, of the action functional defined in (2.7.7). The case analyzed is the simplest three dimensional linear motion case, with $q = (x, y, z)^T$ Cartesian coordinates, but q can be defined arbitrarily with angles position or Cartesian coordinates or both; in that case the vector q is called generalized coordinates. In the case of humanoid robots it's convenient to define q as the joint angles.

The result achieved on this section is a very important result and is known as Hamilton's principle of least action. It finds its application in many fields of engineering and physics including quantum mechanics, electrodynamics and robotics.

2.8 Lagrange mechanics

By using Euler-Lagrange equation derived above, kinematics and differential kinematics it's possible to get the dynamic model of the robot, which provides to find the relationship between torques and joints accelerations. Equation (2.6.13) in the case of robotics and in absence of nonconservative forces takes the form exhibited in figure 2.8.2



Figure 2.8.1: On the left Leonard Euler, on the right Joseph Louis Lagrange

$$\frac{\partial L}{\partial q} - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} = 0$$

Figure 2.8.2: Euler-Lagrange equation in the non-forced case

The equation in presence of nonconservative forces becomes

$$\frac{d}{dt} \frac{\partial L(q, \dot{q})}{\partial \dot{q}} - \frac{\partial L(q, \dot{q})}{\partial q} = \tau_{nc}^T \quad (2.8.1)$$

where τ_{nc} are the nonconservative generalized forces.

The Lagrangian function $L(q, \dot{q})$ is defined as

$$L(q, \dot{q}) = T(q, \dot{q}) - U(q) \quad (2.8.2)$$

where $T(q, \dot{q})$ is the kinetic energy and $U(q)$ is the potential energy.

Substituting the Lagrangian expression into Euler-Lagrange equation:

$$\frac{d}{dt} \frac{\partial T(q, \dot{q})}{\partial \dot{q}} - \frac{\partial T(q, \dot{q})}{\partial q} + \frac{\partial U(q)}{\partial q} = \tau_{nc}^T \quad (2.8.3)$$

The kinetic energy is always a quadratic form of the velocity, so it can be expressed as a quadratic function of joints velocity \dot{q} through the symmetric definite positive matrix $B(q)$; in the one dimensional case, a point particle with mass \bar{m} moving at velocity \bar{v} has a kinetic energy equal to $\frac{1}{2}\bar{m}\bar{v}^2$; in the case of the robot, it can be expressed through the form

$$T(q, \dot{q}) = \frac{1}{2} \dot{q}^T B(q) \dot{q} \quad (2.8.4)$$

To compute the $B(q)$ matrix it's necessary to exploit differential kinematic to express arms velocities as a function of joint velocities. The total kinetic energy is the sum of the kinetic energy of each arm, which includes a translational term and a rotational term. Thus in the bi-dimensional case

$$T(q, \dot{q}) = \sum_{i=1}^5 \left[\frac{1}{2} m_i \dot{q}^T J_{CoM,i}^T(q) J_{CoM,i}(q) \dot{q} + \frac{1}{2} I_{c,i} \dot{q}^T \tilde{J}_i^T \tilde{J}_i \dot{q} \right] \quad (2.8.5)$$

where m_i is the mass of the i -th arm, and $I_{CoM,i}$ is its inertial mass evaluated respect to the Center of Mass of the i -th arm.

So by equation (2.8.4) it comes that the matrix $B(q)$ expression is

$$B(q) = \sum_{i=1}^5 \left[m_i J_{CoM,i}^T(q) J_{CoM,i}(q) + I_{CoM,i} \tilde{J}_i^T \tilde{J}_i \right] \quad (2.8.6)$$

The partial derivatives of kinetic energy are

$$\frac{\partial T(q, \dot{q})}{\partial \dot{q}} = \dot{q}^T B(q) \quad (2.8.7)$$

$$\frac{\partial T(q, \dot{q})}{\partial q} = \frac{1}{2} \left(\frac{\partial}{\partial q} \dot{q}^T B(q) \dot{q} \right) \quad (2.8.8)$$

The partial derivative of potential energy is defined as $g(q)$ and represents the gravity generalized force over the joints

$$g(q) = \left(\frac{\partial U(q, \dot{q})}{\partial q} \right)^T \quad (2.8.9)$$

So the equation (2.8.3) becomes

$$\frac{d}{dt} \dot{q}^T B(q) - \frac{1}{2} \left(\frac{\partial}{\partial \dot{q}} \dot{q}^T B(q) \dot{q} \right) + g(q) = \tau_{nc}^T \quad (2.8.10)$$

Analyzing each component of the vector

$$b_i^T(q) \ddot{q} + \dot{q}^T \frac{db_i}{dq} \dot{q} - \frac{1}{2} \dot{q}^T \frac{\partial B(q)}{\partial q_i} \dot{q} + g_i(q) = \tau_{nc,i}, \quad i = 1, \dots, n \quad (2.8.11)$$

By defining the matrix $\tilde{S}_i(q)$

$$\tilde{S}_i(q) = \frac{db_i}{dq} - \frac{1}{2} \frac{\partial B(q)}{\partial q_i} \quad (2.8.12)$$

equation (2.8.11) becomes

$$b_i^T(q) \ddot{q} + \dot{q}^T \tilde{S}_i(q) \dot{q} + g_i(q) = \tau_{nc,i} \quad (2.8.13)$$

that in vectorial form is

$$B(q) \ddot{q} + m(q, \dot{q}) + g(q) = \tau_{nc} \quad (2.8.14)$$

where

$$m(q, \dot{q}) = \begin{pmatrix} \dot{q}^T \tilde{S}_1(q) \dot{q} \\ \vdots \\ \dot{q}^T \tilde{S}_n(q) \dot{q} \end{pmatrix} \quad (2.8.15)$$

Equation (2.8.14) is called dynamic equation and is very important in robot modeling and control. The $B(q)$ matrix expresses the instant linear relation between the joint accelerations and the generalized forces τ_{nc} . The term $m(q, \dot{q})$ includes both centrifugal and Coriolis forces; $g(q)$ is the gravitational equivalent force. An other notation can be used by defining the $C(q, \dot{q})$ matrix as:

$$C(q, \dot{q}) = \begin{pmatrix} \dot{q}^T \tilde{S}_1(q) \\ \vdots \\ \dot{q}^T \tilde{S}_n(q) \end{pmatrix} \quad (2.8.16)$$

so

$$m(q, \dot{q}) = C(q, \dot{q})\dot{q} \quad (2.8.17)$$

With this notation the dynamic equation becomes:

$$B(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau_{nc} \quad (2.8.18)$$

Nonconservative generalized forces τ_{nc} include actuator forces τ_A , friction forces τ_F and external forces τ_{EXT} .

Actuator forces are the torques applied by electric engines to the joints of the manipulator. In robotic problems those torques are computed by micro controllers that implement a control law, and can be considered as an input u for the system. In many cases the choice of the generalized coordinates is not coherent with the action of these torques, so it's necessary to introduce a matrix B_u applied on the input u , composed of zeros and ones that allow to solve this correspondence problem between torques and angles on which these torques act. The matrix B_u is very useful also in the case in which the robot has some non-actuated joints. In such cases the row of the matrix B_u which corresponds to the non-actuated joint is null. In the case of humanoid robot, the joint q_1 is non-actuated, because it represents the angle between the tibia and the floor, and there is no engine in that point able to torque directly the angle q_1 ; so the

first row of B_u is null. For the choice of coordinate used in this thesis there aren't any problem of correspondence between torques and angles on which torques act, so the matrix B_u is defined as:

$$B_u = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.8.19)$$

Friction forces are dissipative torques acting on joints; the most important characteristic of this kind of forces is that they are proportional to joint velocities, and they are always negative in terms of energy. The easiest way to define it is to use a negative definite matrix D applied to velocity, id est:

$$\tau_F = -D\dot{q} \quad D > 0 \quad (2.8.20)$$

The last important nonconservative forces to model are external forces applied by external agents over the robot. An important relation of dynamics affirms that the isomorphism which transforms an external force applied to a certain point of the robot into the relative torques generated on the manipulator joints by this force is the transpose of the Jacobian matrix referred to the given point. With this relation the term of external force can be expressed in this way:

$$\tau_{EXT} = J(q)^T F_{EXT} \quad (2.8.21)$$

With these considerations the dynamic model of the robot can be written in an extended form:

$$B(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = B_u u - D\dot{q} + J(q)^T F_{EXT} \quad (2.8.22)$$

Once the dynamic model has been derived, it can be set in form of a non-linear differential vectorial equation i.e. $\dot{x} = f(x) + g(x)u$: defining the state variable

$$x = \begin{pmatrix} q \\ \omega \end{pmatrix} \quad (2.8.23)$$

where q is the joints position and ω is the joints velocity and remembering that $\omega = \dot{q}$, the nonlinear model is

$$\begin{pmatrix} \dot{q} \\ \dot{\omega} \end{pmatrix} = \begin{pmatrix} \omega \\ B^{-1}(-C(q, \dot{q})\dot{q} - g(q) - D\dot{q} + J(q)^T F_{EXT}) \end{pmatrix} + \begin{pmatrix} 0 \\ B^{-1}B_u \end{pmatrix} u \quad (2.8.24)$$

where

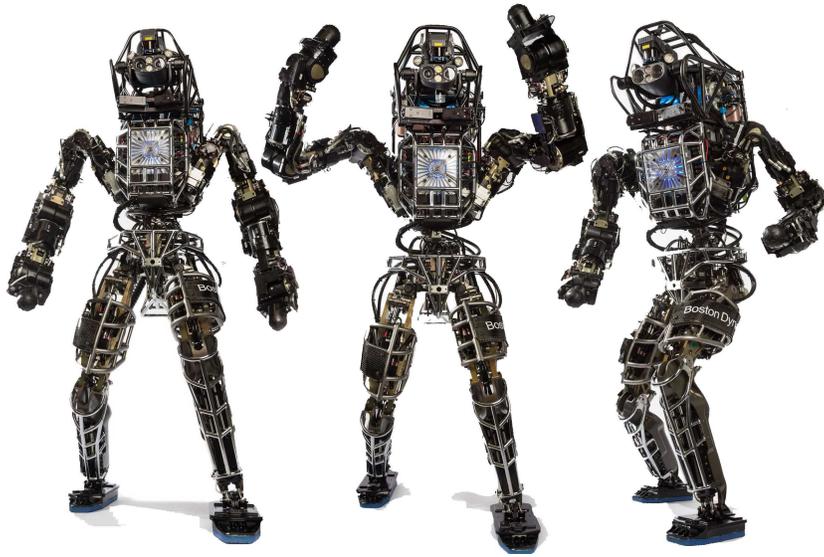


Figure 2.8.3: Humanoid robots

$$\begin{cases} f(x) = \begin{pmatrix} \omega \\ B^{-1}(-C(q, \dot{q})\dot{q} - g(q) - D\dot{q} + J(q)^T F_{EXT}) \end{pmatrix} \\ g(x) = \begin{pmatrix} 0 \\ B^{-1}B_u \end{pmatrix} \end{cases} \quad (2.8.25)$$

2.9 Hybrid model

One of the most important feature of humanoid robot is that it hasn't a fixed

support, it can walk changing his support leg.

It's important to distinguish two different configurations: the single support configuration and the double support configuration. In the first one the humanoid has one leg fixed on the ground, called stance leg, and the other one moving over the ground, called swing leg. In the present case the robot model is bi-dimensional, so it's not important to distinguish left leg and right leg, but stance leg and swing leg. The position of the stance foot is constant during each step. The swing foot position is always constrained to be over the ground namely with positive ordinate. To tackle the topic of bipedal gait it's very useful to use an hybrid model, i.e. to use a set of differential equations that describe the robot in all his configuration. To simplify the calculus, in the control proposed in chapter 5 the double support phase has not been considered, because it has been supposed the the impact lifts the stance leg instantly.

A hybrid automaton is a dynamical system that describes the evolution in time of the values of a set of discrete and continuous state variables.

Definition 1. A hybrid automaton H is a collection $H = (Q, X, f, g, U, Init, Dom, E, G, R)$, where

- $Q = \{q_1, q_2, \dots\}$ is a set of discrete states;
- $X = \mathbb{R}^n$ is a set of continuous states;
- $U = \mathbb{R}^p$ is the input space;
- $f(\cdot), g(\cdot) : Q \times X \rightarrow \mathbb{R}^n$ are vector fields;
- $Init \subseteq Q \times X$ is a set of initial states;
- $Dom(\cdot) : Q \rightarrow 2^X$ is a domain;
- $E \subseteq Q \times Q$ is a set of edges;

- $G(\cdot) : E \rightarrow 2^X$ is a guard condition;
- $R(\cdot, \cdot) : E \times X \rightarrow 2^X$ is a reset map.

Recall that 2^X denotes the power set (set of all subsets) of X .

In the case of humanoid robot the hybrid automaton H is defined as:

- $Q = \{q_1\}$: the model has a single discrete state characterized by the single support situation;
- $X = \mathbb{R}^{10}$: the order of the continuous state is 10, 5 for the joint positions and 5 for joint velocities;
- $f(\cdot), g(\cdot) : Q \times X \rightarrow \mathbb{R}^n$ are the vector fields defined in (2.8.25) which describe the model behavior for the single discrete state q_1 ;
- $U = \mathbb{R}^4$: the input space order is four as the number of actuated joints;
- $Init \subseteq Q \times X$ is a set of initial states;
- $Dom(q_1)$ is composed by all the states on which the swing foot position has positive ordinate;
- $E \subseteq Q \times Q$ is defined by the only self transition because the automaton has a single state;
- $G(\cdot) : E \rightarrow 2^X$ the self transition happens at the moment on which the swing foot touches the ground. The hyper surface that represents this condition will be discussed in chapter 5 section 5.3;
- $R(\cdot, \cdot) : E \times X \rightarrow 2^X$: the computation of the reset map is not simple and is dealt on chapter 5 subsection 5.3.1 with detailed explanation.

Chapter 3

Gait Analysis

After studying kinematics, differential kinematics and dynamics of the humanoid robot, it can be introduced the problem of the gait.

Many researchers faced this problem finding several solutions, each one with its implications. Some of these techniques are based on pre-computed trajectories that create a periodic orbit of the legs motion. This trajectories are based on walking primitives, simple standard movements periodically applied through the solution of inverse kinematics. This method is described in section 4.2; it is a good approach for its simplicity but it presents the problem of non-robustness of the gait; in fact the controller does not provide to stabilization in the case of a perturbation on joint velocities. It can be considered an open loop controller, because it doesn't present a feedback on robot parameters, it

just applies a pre-computed trajectory. Other techniques are based on Push Recovery, that is the control of recovering the stable position of the robot subject to a perturbation, like a push on the back. To better understand Push Recovery an overview on locomotion theory is necessary.

3.1 Definitions and Concepts

In humanoid robots it's important to define some points that are very important for gait analysis.

The first one is the Center of Mass (CoM). This point is used in several mechanical applications from robotics to aircraft control; its definition is fundamental in problems that include rigid bodies, because they might have very complex shapes and might be very difficult to study. The Center of Mass of a rigid body is defined as the unique point where if a force is applied it doesn't generate momentum. Each rigid body has a Center of Mass: convex bodies always have the CoM inside the body itself. This is not always true for non-convex bodies. In the case of humanoid robot the position of the CoM depends on the configuration of the joint q , and in some cases it could be outside the body. Moreover it's often used the notation FCoM to express the floor projection of the CoM, so in a bidimensional case it can be defined as

$$FCoM = \begin{pmatrix} 1 & 0 \end{pmatrix} CoM \quad (3.1.1)$$

The second important point is called Center of Pressure (CoP). This is the point in which is applied the resulting Ground Reaction Force (GRF) at the contact surface. In the case of flat surface, this point corresponds to an other point called Zero Moment Point (ZMP) that is the point on the ground surface in which the horizontal component of the moment of Ground Reaction Force is zero. In the case of humanoid robot the Center of Pressure is always located within the convex hull of the feet support area. Its position depends of the weight distribution between the feet. In this thesis feet are modeled as points with null support area: thus in the single support phase the Center of Pressure always coincides with the ground contact point of the stance foot, while in double support phase it's a point on the ground between the two feet.

Moreover it's important to introduce an other point called Centroidal Moment Pivot (CMP) that is defined as the point where a line parallel to the Ground Reaction Force passing through the Center of Mass intersects with the external contact surface. In Figure (3.1.2) CMP corresponds to A. The distance between Center of Pressure and Centroidal Moment Pivot is very important because is related to the arm length of the torque generated by the gravity force. If the Center of Pressure coincides with Centroidal Moment Pivot the lever arm is null and the gravity force doesn't generate any torque on the humanoid body; if not, the gravity force generates a torque that makes the robot tip forward or backward. A good control strategy must ensure that the distance between the Center of Pressure and the Centroidal Moment Pivot is kept small.

Other important definition are:

- **Capture State:** State in which the kinetic energy of the biped is zero and can remain zero with suitable joint torques. Note that the Center of Mass must lie above the Center of Pressure in a Capture State. The vertical

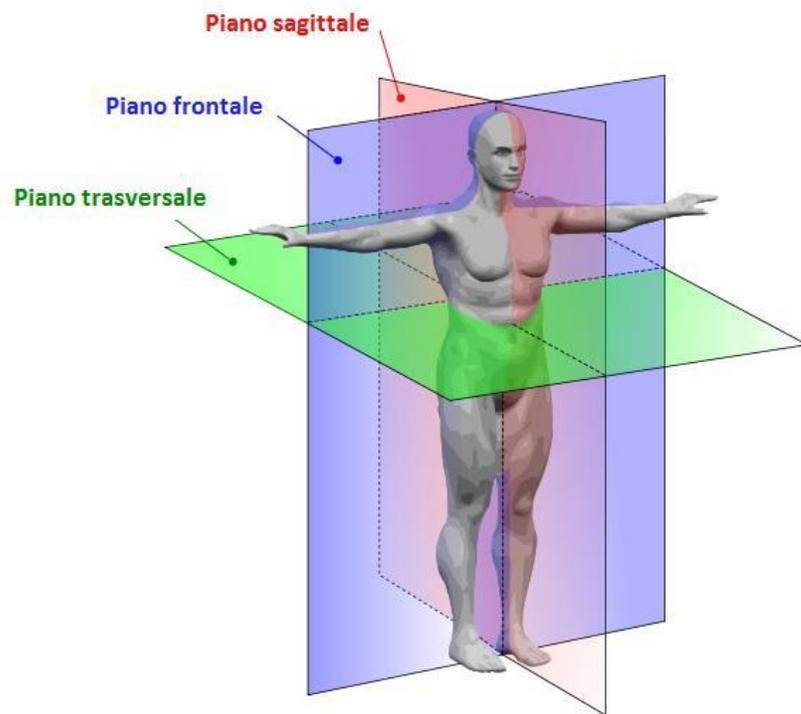


Figure 3.1.1: Robot planes

upright “home position” is an example of a Capture State.

- **Safe Feasible Trajectory:** Trajectory through state space that is consistent with the robot’s dynamics, is achievable by the robot’s actuators, and does not contain any states in which the robot has fallen.
- **Capture Point:** For a biped in state x , a Capture Point P is a point on the ground such that if the biped covers P (makes its Base of Support include P), either with its stance foot or by stepping to P in a single step and then maintains its Center of Pressure to lie on P , then there exists a Safe Feasible Trajectory leading to a Capture State. The location of a Capture Point is dependent on the trajectory through state-space before and after swinging the leg and thus is not a unique point. Therefore, there exists a Capture Region such that if the Center of Pressure is placed inside this region, then the biped can stop for some state space trajectory.
- **Capture Region:** The set of all Capture Points.

There are several classifications for a gait: in Statically Stable Gait the FCoM and the ZMP always remain between the two legs during the entire motion or gait. This implies that if the movement is stopped, the biped will remain in a stable position. These kind of stable gaits are only for really low walking velocities, which impose also low angular velocities in the joints. On the other

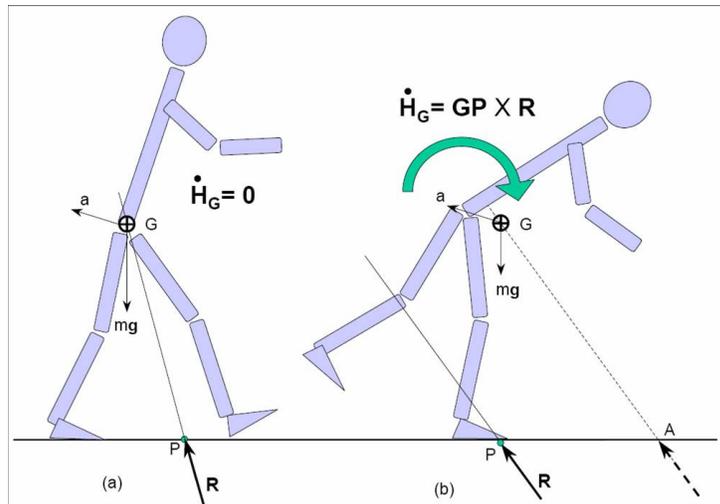


Figure 3.1.2: Humanoid GRF and CMP(equivalent to A)

hand with Dynamically Stable Gait the ZMP resides between the feet during the motion or a gait of a humanoid while the FCoM does not. This kind of gait can be stable for faster movements, but the gait has to meet the requirements of the definition of a walk.

3.2 Push recovery

An other approach derives from an other parallel topic very studied in the field of humanoid robots, that is Push Recovery which will be further examined in this chapter. Push Recovery is the control of recovering the stable position of the robot subject to a perturbation, like a push on the back. The humanoid is compared to a simplified model that is the Linear Inverted Pendulum Model,

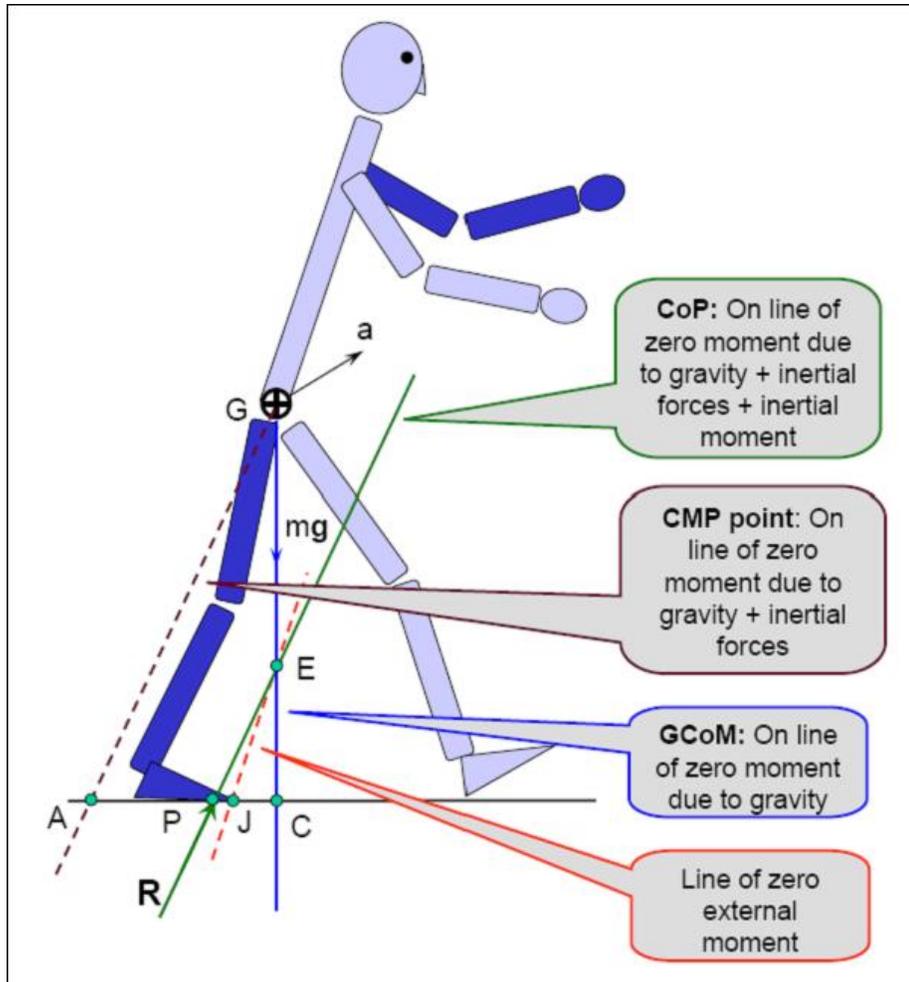


Figure 3.1.3: Main points review

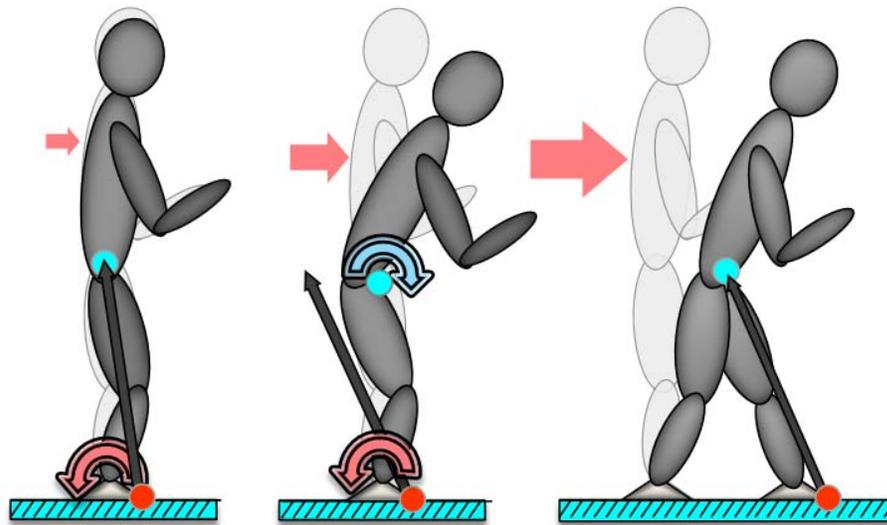


Figure 3.2.1: From left: Ankle strategy, Hip strategy, Stepping

that is used to approximate and study the robot behavior. An other more complex model is the Linear Inverted Pendulum with Flywheel Model, that consider also the application of a recovery torque by the torso to compensate the push.

With this approach there are different recovery strategies depending on the strength of the push.

- With small entity push the strategy used is called *ankle strategy*: this strategy is based on the forward shift of the Center of Pressure, that allows the robot to apply a backward force that compensates the push. This approach is possible thanks to the support area of the feet: the bigger this area, in particular the length of the foot, the bigger the maximum recovery force that the robot is able to apply. Moreover, the closer to the front edge the Center of Pressure, the higher the recovery force. The maximum force is obtained with the Center of Pressure posed on the front edge of the stance foot. In the case of the model used in this thesis, the ankle

strategy is not possible because the foot is modeled as a point, with null support area;

- With bigger entity pushes the strategy used is called *hip strategy*: this strategy is often combined with ankle strategy, and it's based on the application of a torque by the torso to control the position of Centroidal Moment Pivot. The simplified model used to study and compute the required torque is the Linear Inverted Pendulum with Flywheel Model;
- The last strategy is used with great entity push, when ankle strategy and hip strategy are not sufficient to compensate it. In this case *stepping* is necessary. With stepping a new chapter has to be opened, for the requirement of a motion control technique. After several attempts, it has been stated that feedback linearization is the best approach to deal with motion control of humanoid robot. The problems and the implications of motion control in humanoid robots are treated in chapter 4. In these cases many researchers used an approach based on Model Predictive Control, that will be further discussed in this chapter
- When a step is not sufficient to compensate the push, more than one is necessary. This case brings again to gait control problem; therefore a hybrid model is necessary and after each step the model gets a reset, and the optimization algorithm is executed again. At each step, the body velocity will be lower, until it will be such lower to be compensated with ankle and hip strategies

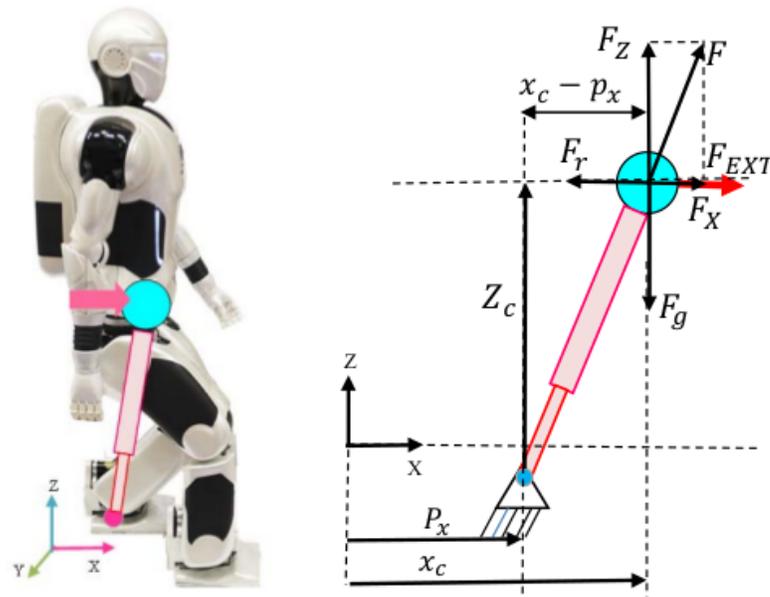


Figure 3.3.1: The Linear Inverted Pendulum Model

3.3 The Linear Inverted Pendulum Model

An important simplified model very used in literature to deal with gait issues is the Linear Inverted Pendulum Model (LIPM). With this model the whole humanoid is represented by its Center of Mass posed on the top of a pendulum. The simplest case of Linear Inverted Pendulum Model is described in the next subsection.

3.3.1 Without flywheel

The dynamics for bipedal robots are strongly nonlinear that makes the gait planning difficult. For this reason there exists simplified model used to study the problem of gait and push recovery. In this case it's assumed that there is no angular momentum and no change of angular momentum in the system, so there isn't any force that generates a change of angular momentum about the Center of Mass. If it's assumed that the Center of Mass is at constant height the dynamics are identical to the well known Linear Inverted Pendulum Model:

$$(m\ddot{x}_c)z_c = mg(x_c - p_x) \quad (3.3.1)$$

where m is the mass of whole robot, x_c is the x component of the position of the Center of Mass, p_x is the x component of the position of the Zero Moment Point, z_c is the height of the Center of Mass and g is the gravity acceleration. By posing $\omega = \sqrt{\frac{g}{z_c}}$ the equation can be written as

$$\ddot{x}_c = \omega^2(x_c - p_x) \quad (3.3.2)$$

that is a second order differential equation. The variable ω has a physical interpretation as well: it is the natural frequency of the oscillation of the the Linear Inverted Pendulum.

The interpretation of this formula is that Ground Reaction Force acting on the basement generates a torque on the pendulum that is proportional to the moment arm of this force, that is $(x_c - p_x)$.

This model can be extended with the introduction of an external horizontal force F_{EXT} acting on the Center of Mass. The model becomes

$$\ddot{x}_c = \omega^2(x_c - p_x) + \frac{F_{EXT}}{m} \quad (3.3.3)$$

F_{EXT} is used to model the recovery force applied with the shift of the Center of Pressure from the center of the foot to the edge. With this shift the Ground Reaction Force, that is centered on the Center of Pressure, reaches a position on which the moment arm is such that it's generated a moment that is opposite to the moment due to the gravity force.

This model assumes that there is no angular momentum and no change of angular momentum in the system. In the next subsection the Linear Inverted Pendulum Model with Flywheel is analyzed.

3.3.2 With flywheel

The dynamic of the torso can play an important role in push recovery and gait control problem. This joint can be used to apply a torque about the CoM. The Centroidal Moment Pivot is equal to the Center of Pressure in the case of zero moment about the Center of Mass. For the Linear Inverted Pendulum Model this is always the case, because there isn't any rigid body able to generate a moment. However for a non zero moment about the Center of Mass the Centroidal Moment Pivot can move ahead or behind the Center of Pressure, generating a torque that makes the robot tip forward or backward, respectively. This effect is achieved thanks to the conservation of angular momentum. The result is a greater horizontal recovery force on the Center of Mass. This effect can be studied by approximating the torso as a flywheel that can be torqued directly;

$$\ddot{x}_c = \omega^2 (x_c - p_x) + \frac{\dot{H}}{mz} + \frac{F_{EXT}}{m} \quad (3.3.4)$$

where \dot{H} is rate of upper-body angular momentum that can be handled by torque of torso.

The relation between ZMP and CMP can be written as

$$CMP_x = p_x + \frac{\dot{H}}{F_z} \quad (3.3.5)$$

where F_z is the vertical component of Ground Reaction Force. With this relation the (3.3.4) becomes

$$\ddot{x}_c = \omega^2 (x_c - CMP_x) + \frac{F_{EXT}}{m} \quad (3.3.6)$$

From (3.3.5) it emerges that in the case of no moment, thus $\dot{H} = 0$, the Centroidal Moment Pivot and the Zero Moment Point coincide. This is the case of the basic Linear Inverted Pendulum Model analyzed in the previous section. With a moment $\dot{H} \neq 0$, the Centroidal Moment Pivot differs from the Zero Moment Point. The distance between Centroidal Moment Pivot and the Zero Moment Point is considered an index of stability of the humanoid. In gait application the controller should be projected in order to drive this distance to zero, to avoid that the robot tips forward or backward. In Push Recovery application the Centroidal Moment Pivot is shifted by the power of the push, and the torso torque has to be actuated to compensate the push and drive the Centroidal Moment Pivot towards the Zero Moment Point.

3.3.3 Capture Point Dynamics

From the research of Pratt [1], it emerges that in Push Recovery application it's important to define the Capture Point dynamics: using the concept of Capture Points and the Capture Region it's possible to determine *when* and *where* to take a step to recover from a push:

- *When to take a step:* if a Capture Point is situated within the convex hull of the foot support area, the robot is able to recover from the push without having to take a step, see Figure 3.3.2, top. Otherwise, it must take a step, see Figure 3.3.2, middle.
- *Where to take a step:* In order to stop in one step the robot must step such that its foot support area regains an intersection with the Capture Region.
- *Failure:* The humanoid will fail to recover from a push in one step if the Capture Region in its entirety lies outside the kinematic workspace of the swing foot. In this case the robot must take at least two steps in order to stop, if it can stop at all. This is shown in Figure 3.3.2, bottom.

Its dynamics is related to the unstable part of the LIPM dynamics and can be defined as follow:

$$\ddot{\zeta}_x = x_c + \frac{\dot{x}_c}{\omega} \quad (3.3.7)$$

where ζ is the Capture Point. From (3.3.7), the CoM dynamics is given by:

$$\dot{x}_c = \omega (\zeta_x - x_c) \quad (3.3.8)$$

By a substitution of this relation on equation (3.3.6) it can be obtained the Capture Point dynamics:

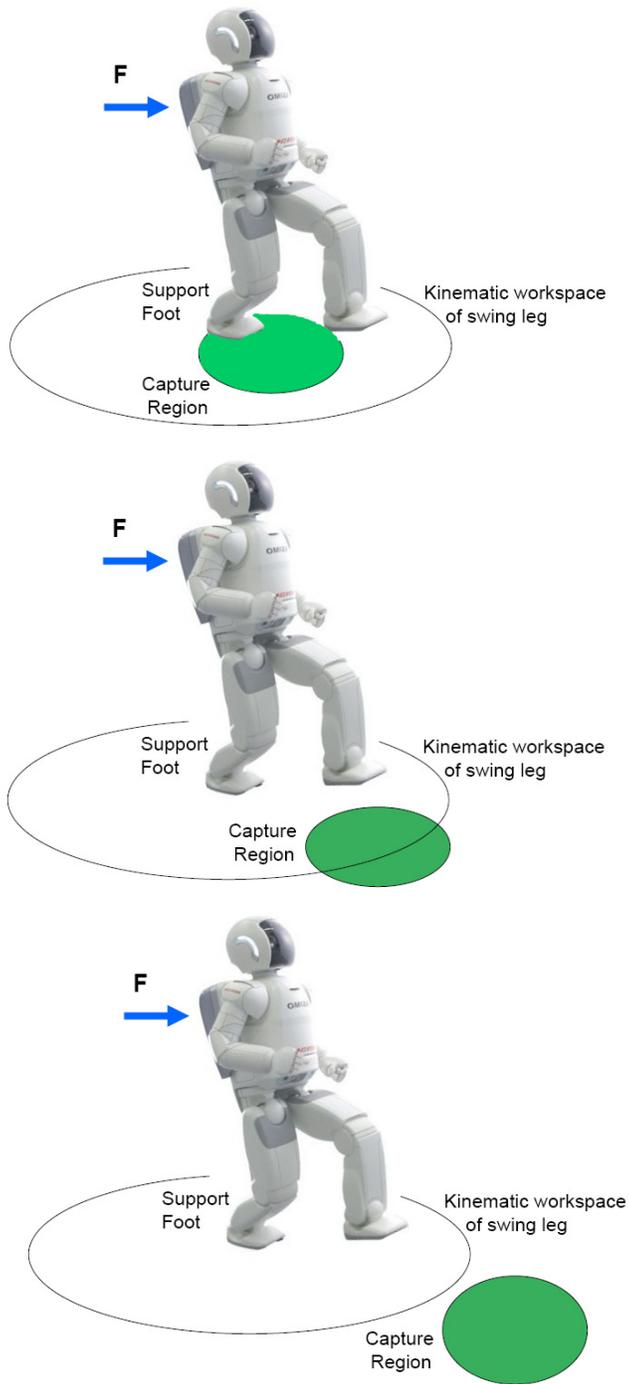


Figure 3.3.2: Capture region

$$\dot{\xi}_x = \omega (\xi_x - CMP_x) + \frac{F_{EXT}}{m\omega} \quad (3.3.9)$$

As obvious in (3.3.9), the CMP can repellent the capture point. In order to balance recovery of a humanoid robot Capture Point must be controlled. When CP is located within support polygon it can be controlled by ZMP and when it is located out of support polygon it can be controlled by CMP or stepping. Using the concept of Capture Points it's possible to determine when and where to take a step to recover from a push. Next sections deals how to use the potential of Capture Point in Push recovery controller based on the MPC scheme.

3.3.4 Model Predictive Control

From the previous analysis a linear dynamic model can be derived. To apply the Model Predictive Control it's necessary to discretize the system; from the research of Shafiee-Ashtiani on the article [2], the discrete model is derived:

$$\begin{cases} x_{t+1} = (1 - \omega T) x_t + \omega T \xi_t \\ \xi_{t+1} = (1 + \omega T) \xi_t - \omega T \left(p_{x,t} + \frac{\dot{H}_t}{mg} \right) + \frac{F_{EXT}}{m\omega} \\ p_{x,t+1} = p_{x,t} + \dot{p}_{x,t} T \\ \dot{H}_{t+1} = \dot{H}_t + \ddot{H}_t T \end{cases} \quad (3.3.10)$$

By defining the state X_t as

$$X_t = \begin{pmatrix} x \\ \zeta \\ p_x \\ \dot{H} \\ F_{EXT} \end{pmatrix} \quad (3.3.11)$$

and the input

$$U_t = \begin{pmatrix} \dot{p}_x \\ \dot{H} \end{pmatrix} \quad (3.3.12)$$

The linear model can be obtained by activating the last state variable, F_{EXT} by defining the parameter μ which is equal to 1 when a push is exerted and 0 in the other time steps. The obtained model is

$$X_{t+1} = A_t X_t + B U_t \quad (3.3.13)$$

$$A_t = \begin{pmatrix} (1 - \omega T) & \omega T & 0 & 0 & 0 \\ 0 & (1 + \omega T) & -\omega T & \frac{-\omega T}{mg} & \frac{1}{m\omega} \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \mu \end{pmatrix} \quad (3.3.14)$$

$$B = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ T & 0 \\ 0 & T \\ 0 & 0 \end{pmatrix} \quad (3.3.15)$$

Given a sequence of control inputs \hat{U} , the linear model in (3.3.13) can be converted into a sequence of states \hat{X} , for the whole prediction horizon

$$\hat{X}_{t+1} = \hat{A}\hat{X} + \hat{B}\hat{U} \quad (3.3.16)$$

$$\hat{A} = \begin{pmatrix} A \\ A^2 \\ \vdots \\ A^{N-1} \\ A^N \end{pmatrix} \quad (3.3.17)$$

$$\hat{B} = \begin{pmatrix} B & O & \cdots & O & O \\ AB & B & O & \cdots & O \\ \vdots & \vdots & \ddots & & \vdots \\ A^{N-2}B & A^{N-3}B & & B & O \\ A^{N-1}B & A^{N-2}B & \cdots & AB & B \end{pmatrix} \quad (3.3.18)$$

$$\hat{X} = \begin{pmatrix} \hat{X}_{t+1} \\ \hat{X}_{t+2} \\ \vdots \\ \vdots \\ \hat{X}_{t+N} \end{pmatrix} \quad (3.3.19)$$

$$\hat{U} = \begin{pmatrix} \hat{U}_t \\ \hat{U}_{t+1} \\ \vdots \\ \vdots \\ \hat{U}_{t+N-1} \end{pmatrix} \quad (3.3.20)$$

The cost functional is defined as

$$J = k_1 \left(\zeta_x - \zeta_{ref,x} \right)^2 + k_2 (\dot{p}_x)^2 + k_3 (\dot{H})^2 + k_4 (\ddot{H})^2 \quad (3.3.21)$$

that minimize the inputs \dot{p}_x , \ddot{H} , the Capture Point error $\zeta_x - \zeta_{ref,x}$ and the Zero Moment Point speed \dot{p}_x using the weight coefficients k_1, k_2, k_3 and k_4 .

Chapter 4

Motion Control Techniques

Besides the theory of locomotion which deals with the equilibrium of the robot with a theoretical approach, it's important to face the problem of motion control. Motion control is the control of the trajectory of the robot exploiting its kinematic and dynamic model. In the case of an industrial robot, that is represented with an open kinematic chain, there exist many control methods, for example the one described below in section 4.1. Other techniques more specific for humanoid robots are described in section 4.2.

4.1 Transpose Jacobian PD

A very used approach to control the motion of the robot derives from the attempt to drive the position error to zero through a Lyapunov approach. Given

$$\tilde{x} = x_d - x \quad (4.1.1)$$

with \tilde{x} the position error, x_d the desired position and x the actual position, a Lyapunov function can be chosen as

$$V(\dot{q}, \tilde{x}) = \frac{1}{2} \dot{q}^T B(q) \dot{q} + \frac{1}{2} \tilde{x}^T K_p \tilde{x} \quad (4.1.2)$$

where K_p is a positive defined matrix of gains. By differentiation of (4.1.2) it's obtained

$$\dot{V} = \dot{q}^T B(q) \ddot{q} + \frac{1}{2} \dot{q}^T \dot{B}(q) \dot{q} + \frac{1}{2} \tilde{x}^T K_p \dot{\tilde{x}} \quad (4.1.3)$$

in the case x_d is a constant reference $\dot{\tilde{x}}_d = 0$ and

$$\dot{\tilde{x}} = -J_{foot}(q) \dot{q} \quad (4.1.4)$$

thus

$$\dot{V} = \dot{q}^T B(q) \ddot{q} + \frac{1}{2} \dot{q}^T \dot{B}(q) \dot{q} - \dot{q}^T J_{foot}^T(q) K_p \tilde{x} \quad (4.1.5)$$

From a known physics property that asserts that matrix $\dot{B} - 2C$ is skew-symmetric, equation (4.1.5) can be written as

$$\dot{V} = -\dot{q}^T D \dot{q} + \dot{q}^T \left(u - g(q) - J_{foot}^T(q) K_p \tilde{x} \right) \quad (4.1.6)$$

where D is the friction matrix defined in chapter 2.

Now the input u can be chosen in order to make the derivative of the Lyapunov function definite negative, and prove the stability of the system. The input is posed

$$u = g(q) + J_{foot}^T(q)K_P\tilde{x} - J_{foot}^T(q)K_DJ_{foot}(q)\dot{q} \quad (4.1.7)$$

composed of three terms:

- the term $g(q)$ is the gravity compensation: with this term the robot is not affected by gravity field and it can be considered as posed in a 0-gravity environment. This term is fundamental to allow an high performance control, but it requires a good knowledge of the model of the robot. The computation of the term $g(q)$ is given by the partial derivative $g(q) = \left(\frac{\partial \mathcal{U}(q,\dot{q})}{\partial q}\right)^T$ dealt in the modeling chapter with a detailed explanation. This term is introduced to simplify the $g(q)$ on the expression of the derivative \dot{V}
- the term $J_{foot}^T(q)K_P\tilde{x}$ is introduced to simplify the expression of \dot{V} as well, but it has a precise control interpretation: $K_P\tilde{x}$ is a term proportional to the position error that gives an acceleration. The matrix J_{foot}^T from a well known property of mechanics is the linear application which transform a force applied on the foot in the torques of the joint necessary to realize that force.
- the term $-J_{foot}^T(q)K_DJ_{foot}(q)\dot{q}$ pre-multiplied by \dot{q}^T gives a definite negative quadratic term $-\dot{q}^T J_{foot}^T(q)K_DJ_{foot}(q)\dot{q}$ that gives an additional stabilizing term to Lyapunov function derivative. This term comes out from mathematical reasoning, but it has a physical interpretation as well: starting from right going left, the term $J_{foot}(q)\dot{q}$ coincides with $\dot{\tilde{x}}$ by differen-

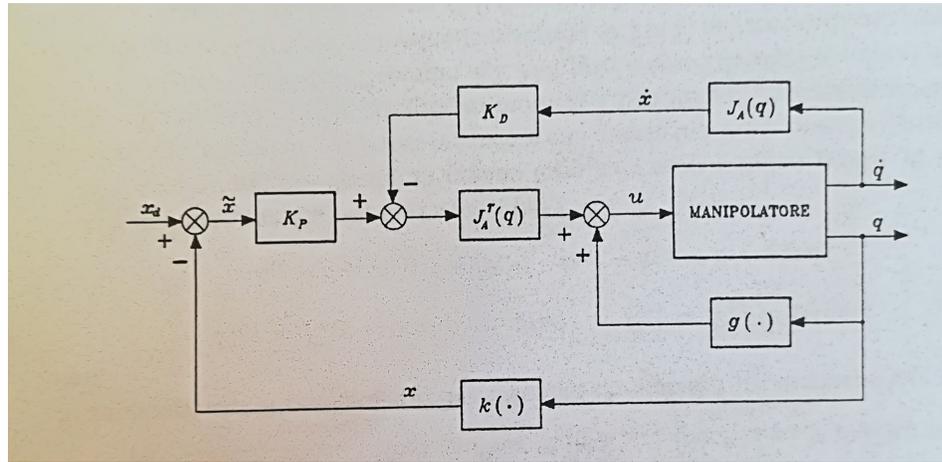


Figure 4.1.1: Jacobian Transpose control scheme

tial kinematic fundamental equation (2.3.1). K_D is a gain matrix on the velocity error $\dot{\tilde{x}}$ and J_{foot}^T is also in this case the linear application which transform a force applied on the foot in the torques of the joint necessary to realize that force.

Overall the controller is composed of a gravity compensation term, a proportional term and a derivative term. For this reason it can be considered a PD controller with gravity compensation.

4.2 Walking primitives and inverse kinematic based techniques

An other technique is described in the article “Zero-Moment point method for stable biped walking” by M.H.P. Dekker [3], and is based on two phases: the first is the offline trajectory generation, and the second is the inverse kinematic solution to actuate the computed trajectory. The two distinct phases are analyzed in the next subsections

4.2.1 Walking primitives

Walking primitives are fraction of the walking gait of biped robot. An example of walking primitive may be to perform a step of length sl_d . All these walking primitives are computed offline and then stored in a database; then a step sequence planner can select and concatenate walking primitives during runtime in order to obtain a walking pattern.

The most used trajectory for the swing foot is the parabolic one: at first it's necessary to choose, as control parameters, the desired step length sl_d and the maximum height reached by the foot $z_{2_{\max}}$. The choice of these parameters must be very accurate: the step length must be chosen considering legs length and maximum legs angle allowed. Maximum foot height must be chosen not too small in order to ensure a robust walk, and not too high in order to reduce energy consumption. Once parameters are defined, the typical function chosen is a parabola that reaches the maximum height at the medium point of the step.

By a well know theorem of algebra, given three points, there exists a unique parabola that intersect all the three points. So from the condition on the desired trajectory $z_{2d}(x)$

$$\begin{cases} z_{2d}\left(-\frac{sl_d}{2}\right) = 0 \\ z_{2d}(0) = z_{2\max} \\ z_{2d}\left(\frac{sl_d}{2}\right) = 0 \end{cases} \quad (4.2.1)$$

the expression of the parabolic trajectory is

$$z_{2d}(x) = \frac{-4z_{2\max}}{sl_d^2}x^2 + z_{2\max} \quad (4.2.2)$$

defined in the interval $x \in \left[-\frac{sl_d}{2}, \frac{sl_d}{2}\right]$. This equation describes the desired trajectory in the Cartesian space. Through the solution of inverse kinematic it can be transformed into a trajectory in the joint space, and tracked with an inverse kinematic-based algorithm, discussed in the next subsection.

Equation (4.2.2) appears again in a different form in Chapter 5 to describe the desired trajectory of the swing foot as a function of the swing foot horizontal displacement from the hip.

4.2.2 Inverse differential kinematic solution

The realization of the trajectory by the robot can be actuated by the solution of inverse kinematics. In the case of humanoid robot, there is a redundancy on the degrees of freedom: in fact the robot has 5 degrees of freedom and the duty is on the 2D-plane. This redundancy can be solved using an optimization

4.2. WALKING PRIMITIVES AND INVERSE KINEMATIC BASED TECHNIQUES 71

based approach: in fact the solution, in the case of redundancy is not unique and through optimization it can be find a solution that minimizes a given functional. It's considered, as functional,

$$g(\dot{q}) = \frac{1}{2} \dot{q}^T W \dot{q} \quad (4.2.3)$$

where W is a weight matrix for joint velocity. This optimization problem can be solved using Lagrange multipliers

$$g(\dot{q}, \lambda) = \frac{1}{2} \dot{q}^T W \dot{q} + \lambda^T (v - J\dot{q}) \quad (4.2.4)$$

The condition for optimality is

$$\left(\frac{\partial g}{\partial \dot{q}} \right)^T = 0 \quad \left(\frac{\partial g}{\partial \lambda} \right)^T = 0 \quad (4.2.5)$$

From the first condition it can be deduced

$$\dot{q} = W^{-1} J^T \lambda \quad (4.2.6)$$

because W is chosen invertible. The solution is a minima because the second derivative $\frac{\partial^2 g}{\partial \dot{q}^2} = W$ positive definite. By the second condition it's obtained

$$v = J\dot{q} \quad (4.2.7)$$

that coincides with the constraint. By combining the two condition it's obtained the relation

$$\lambda = (JW^{-1}J^T)^{-1} v \quad (4.2.8)$$

and by (4.2.6)

$$\dot{q} = W^{-1}J^T (JW^{-1}J^T)^{-1} v \quad (4.2.9)$$

the matrix $W^{-1}J^T (JW^{-1}J^T)^{-1}$ is a pseudo-inverse matrix of J with weight W . The inverse matrix can be computed only in the case of square matrices. Given a square matrix T , the inverse T^{-1} has the property

$$T \cdot T^{-1} = I \quad (4.2.10)$$

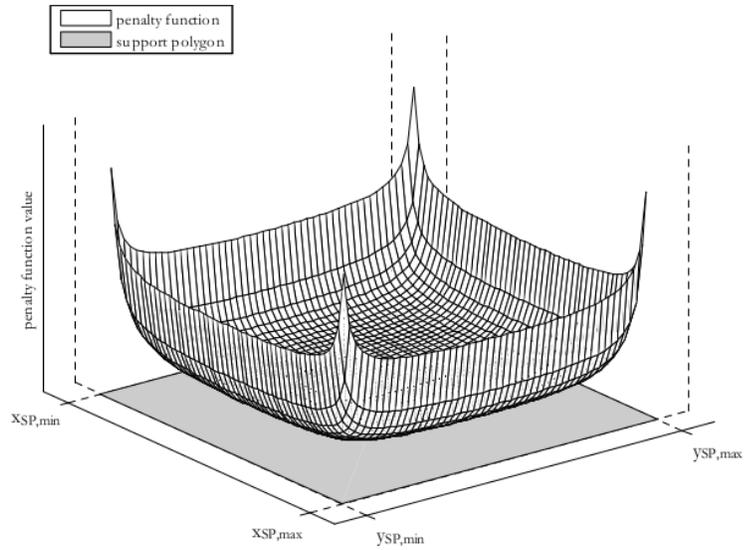
When the matrix is rectangular, the inverse is not feasible, but it can be defined a matrix that conserve the property (4.2.10). This matrix is the one defined in (4.2.9): in fact it's very easy to verify that $J \cdot W^{-1}J^T (JW^{-1}J^T)^{-1} = I$. The pseudoinverse of a rectangular matrix is not unique: each matrix of the form $W^{-1}J^T (JW^{-1}J^T)^{-1}$ with arbitrary W nonsingular matrix is a pseudoinverse of the matrix J . A particular pseudoinverse is the one with $W = I$ that has the form $J^T (JJ^T)^{-1}$.

The redundancy of the robot can be exploited by the optimization algorithm by choosing a more specific functional $g(\dot{q}, \lambda)$. With the choice (4.2.3) the optimization algorithm simply minimizes the joint velocities. A more specific and powerful optimization can be obtained by choosing a functional of the form

$$g(\dot{q}) = \frac{1}{2} (\dot{q} - \dot{q}_0)^T W (\dot{q} - \dot{q}_0) \quad (4.2.11)$$

In this case the norm to be minimized is the weighted norm of $(\dot{q} - \dot{q}_0)$, so the solution will be close to \dot{q}_0 . For the choice of \dot{q}_0 it's suitable to define it as the derivative of an objective function $G(q, \dot{q})$ such as

$$\dot{q}_0 = k_0 \left(\frac{\partial G(q, \dot{q})}{\partial q} + \frac{\partial G(q, \dot{q})}{\partial \dot{q}} \right)^T \quad (4.2.12)$$



4

Figure 4.2.1: Physical interpretation of penalty function described with equation (4.2.14)

There are many possible choice of the objective function; a typical choice, dependent only on q and not \dot{q} in the case of redundant industrial robot is an index of manipulability defined as

$$G(q) = \sqrt{\det(J(q)J^T(q))}$$

with this function the redundancy is exploited by assuming joint configuration far away from singularities.

In the case of humanoid robot, there are more suitable choices of the functional shown in the article of Dekker [3] that are

$$G = x_{ZMP}^2 + y_{ZMP}^2 \quad (4.2.13)$$

and

$$G = \frac{(x_{SP,max} - x_{SP,min})^2}{(x_{SP,max} - x_{ZMP})(x_{ZMP} - x_{SP,min})} + \frac{(y_{SP,max} - y_{SP,min})^2}{(y_{SP,max} - y_{ZMP})(y_{ZMP} - y_{SP,min})} \quad (4.2.14)$$

In particular, as shown in figure 4.2.1, the second objective function $G(q, \dot{q})$ declared in equation (4.2.14) increases when the ZMP goes towards the edge of the support area. Thus it keeps the ZMP inside the support area, and prevent that the robot tips forward or backward.

4.3 Humanoid robot control challenges

The case of a 5 link humanoid the stance foot is constrained to the floor and the swing foot can be driven to the desired position with different approaches. The field of humanoid robot is very difficult because it introduces many problems that are not present in industrial robots:

- **there is an non actuated joint:** the angle between the stance foot and the floor has been modeled as a joint q_1 and is non actuated. Industrial robot are constrained to the floor and for this reason the position of the tool center point can be controlled with high level of freedom. Humanoid robots aren't constrained to the floor and so the position of the swing foot can not be controlled without facing the problem of balancing
- **there is the problem of balancing** that can not be ignored. Classical control techniques assume that each joint of the robot are actuated and it

can be defined a manifold of reachability of the tool center point. In the case of an humanoid robot the concept of reachability manifold is much more complex: it can be defined a manifold of persistent reachability \mathcal{M}_p and transient reachability \mathcal{M}_t : the manifold \mathcal{M}_p includes all the states $x(\bar{t}) \in \mathcal{X}$ where \mathcal{X} is the set of all possible configuration that the humanoid may assume, in which the robot is able to remain for an arbitrary big amount of time, under a certain input. So in can be expressed as

$$\mathcal{M}_p = \{x(\bar{t}) \in \mathcal{X}, \forall T, \exists u(t) t \in [\bar{t}, \bar{t} + T] \text{ s.t. } x(t) = x(\bar{t}) \text{ for } t \in [\bar{t}, \bar{t} + T]\} \quad (4.3.1)$$

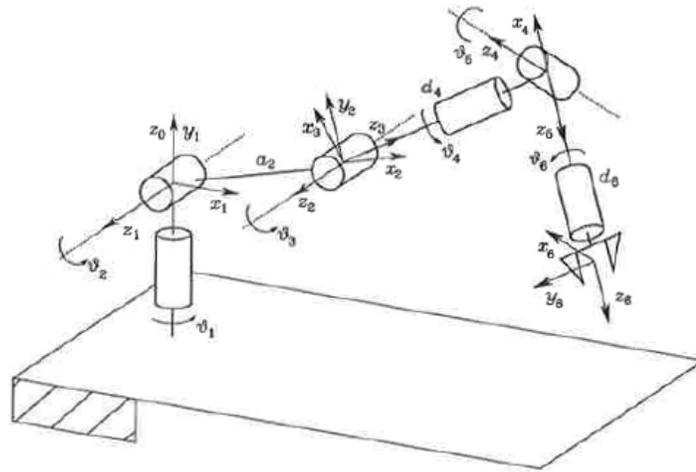
Transient reachability manifold \mathcal{M}_t can be reached but not hold, so

$$\mathcal{M}_t = \{x(\bar{t}) \in \mathcal{X}, \forall u(t) t \in [\bar{t}, \bar{t} + T], \exists \epsilon > 0 \text{ s.t. } x(\bar{t} + \epsilon) \neq x(\bar{t})\} \quad (4.3.2)$$

All the control techniques that exploit the gravity compensation, like the Transpose Jacobian PD one can not be used if the robot reaches the manifold \mathcal{M}_t , because in this manifold the torque able to compensate gravitational field doesn't exist

- **there is the requirement to deal with more than one kinematic function and Jacobian matrices:** in industrial robot the interest of the motion control is the tool center point that perform the duty for which the robot has been programmed; so the kinematic function used is the one that expresses the relation between the position of the tool center point and the joint configuration. By differentiation it's obtained the Jacobian of the tool center point used for the control. In the case of humanoid robot the duty is much more complex and there isn't a tool center point to be controlled. In the first analysis the tool center point can be considered

the swing foot, but this is not a correct approach because the robot must be considered as a complex system of legs and torso to reach the aim of stability



Anthropomorphic arm with spherical wrist

Figure 4.3.1: Industrial robot Kinematic

- **there is not an ordinary kinematic chain:** industrial robot can be considered an ordinary open kinematic chain, because it can be established an order of the joints, starting from the one constrained to the floor going on to the last joint with the tool. The case of humanoid robot is more complex on the kinematic point of view, because there is a bifurcation at the level of the hips: in fact the hip is a point on which the two legs and the torso are conjunct. For this reason it's not possible to establish an order of the link, unlike the case of industrial robot
- **there are constrain for the angle of the joint:** it's well known that human being are not able to rotate the knee over an angle of 180 degree. This can

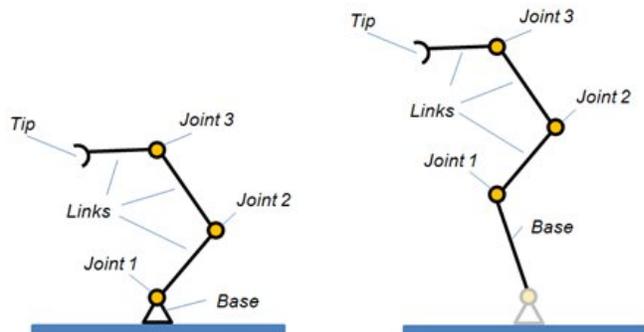


Figure 4.3.2: Kinematic chain of an industrial robot

be translated to a bound on the generalized coordinates q . Control of robots under bounded joint angles is not a simple problem and requires advanced control techniques

- **there is the problem of the impact with the floor:** after the impact with the floor the joint velocity \dot{q} of the robot meet a discontinuity. This is well studied by Grizzle on his article [4] and can be computed through a given procedure. This discontinuity can be modeled with a hybrid transition
- **it includes single and double support:** humanoid robot includes two models: the single support model and the double support model that have different behavior and evolve with different laws. In this thesis the double support model is dealt only to better give a theoretical background of humanoid robot, but it's ignored on the project of the controller. The basic hypothesis is that at the moment of the impact between the swing leg and the floor, the stance leg instantly rise up from the floor becoming the new swing leg, and the old swing leg keep constrained to the floor and becomes the new stance leg. With this hypothesis the double support dwell time is infinitesimal, and for this reason the double support phase is neglected in the controller project.

As explained above, there are several limitations and problems on humanoid robot control, that makes it a very challenging topic. Feedback linearization is proved to be the best approach for closed loop gait control. This is the control used in this thesis, starting from the researches of Grizzle[4, 5].

Chapter 5

Gait Control

This chapter faces the problem of closed-loop gait control. In the previous chapters the problem of humanoid gait and push recovery have been analyzed under several points of view, starting from the modeling through Lagrange formalism, going on with theory of human locomotion and Linear Inverted Pendulum Model. All the problems that make the gait control such a challenging topic in comparison with industrial robots control have been highlighted and many approaches based on pre-computed trajectories and Push Recovery have been illustrated, exploring the works of several researchers. However approaches based on pre-computed trajectories are not robust and work well just under almost ideal conditions, and Push Recovery approaches encounter the problem of motion control in the case of stepping, that is nontrivial with

the first joint non actuated. For this reason the best approach is to define a set of output functions that, if driven to zero, ensure the desired behavior of the humanoid robot. This approach is obtained through feedback linearization technique that is discussed in this chapter.

5.1 Relative coordinates and absolute coordinates

In chapter 2 robot model has been derived. The choice of the joint coordinates has been done in order to obtain a model in the simplest possible way. The coordinate chosen are represented in figure 5.1.1 (a). This coordinates are called relative coordinates, because they are referred to other joints: for example q_2 is the angle formed by the front femur and the direction of the front tibia, as shown in figure 5.1.1 (a). Thus each angle is defined depending on other joints configuration, and for this reason these choice of coordinates is denominated relative coordinates. The advantages of this coordinates choice are multiple, but the most important one is that the joint torques act directly on these angles. In other words, if a torque is applied on the q_2 joint, only q_2 will vary and all other angles will not. Absolute coordinates haven't got this property. Considering figure 5.1.1 (b), if a torque is applied on joint q_{42} all the five joint angles will vary. That is because absolute coordinates aren't defined depending on other joints configuration, but they are defined depending on a fixed reference, that in the case of figure 5.1.1 (b) is the vertical direction. For this reason, this choice of coordinates is called absolute, and each angle does not depend on

other ones. For control purpose it has been used absolute coordinates because they simplify the computation and the definition of output functions.

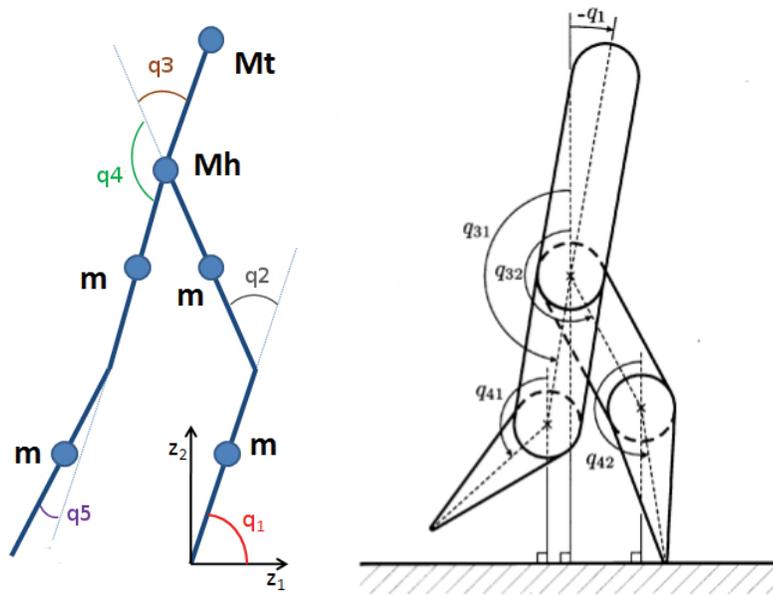


Figure 5.1.1: Relative coordinates on the left (a) and absolute coordinates on the right (b)

Relative coordinates and absolute coordinates are not uncorrelated. There exists an invertible application that provides to pass from one coordinate choice to the other one. This application has the form

$$\bar{q} = \Theta q + h \quad (5.1.1)$$

To compute this application it's necessary to express each angle of a coordinates system as a function of the angles of the other coordinates system. In this thesis the transformation matrix Θ transforms the relative coordinates denoted by q_i into absolute coordinates denoted by \bar{q}_i . The expressions are

$$\begin{cases} \bar{q}_{31} = q_1 + q_2 + \frac{\pi}{2} \\ \bar{q}_{41} = q_1 + \frac{\pi}{2} \\ \bar{q}_{32} = q_1 + q_2 + q_4 - \frac{\pi}{2} \\ \bar{q}_{42} = q_1 + q_2 + q_4 + q_5 - \frac{\pi}{2} \\ \bar{q}_1 = q_1 + q_2 + q_3 - \frac{\pi}{2} \end{cases} \quad (5.1.2)$$

This relation can be written in a form suitable with (5.1.1) as

$$\begin{pmatrix} \bar{q}_{31} \\ \bar{q}_{41} \\ \bar{q}_{32} \\ \bar{q}_{42} \\ \bar{q}_1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \end{pmatrix} + \begin{pmatrix} \frac{\pi}{2} \\ \frac{\pi}{2} \\ -\frac{\pi}{2} \\ -\frac{\pi}{2} \\ -\frac{\pi}{2} \end{pmatrix} \quad (5.1.3)$$

$$\Theta = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix} \quad (5.1.4)$$

$$h = \begin{pmatrix} \frac{\pi}{2} \\ \frac{\pi}{2} \\ -\frac{\pi}{2} \\ -\frac{\pi}{2} \\ -\frac{\pi}{2} \end{pmatrix} \quad (5.1.5)$$

The inverse formula can be easily computed starting from (5.1.1):

$$q = \Theta^{-1} (\bar{q} - h) \quad (5.1.6)$$

$$q = \Theta^{-1} \bar{q} - \Theta^{-1} h \quad (5.1.7)$$

$$\bar{\Theta} = \Theta^{-1} \quad (5.1.8)$$

$$\bar{h} = -\Theta^{-1} h \quad (5.1.9)$$

$$q = \bar{\Theta} \bar{q} + \bar{h} \quad (5.1.10)$$

Equation 5.1.10 is the inverse formula of (5.1.1). The matrix Θ is invertible since its determinant is equal to -1 .

The derivatives \dot{q} and \ddot{q} can be easily derived

$$\dot{\bar{q}} = \Theta \dot{q} \quad (5.1.11)$$

$$\ddot{\bar{q}} = \Theta \ddot{q} \quad (5.1.12)$$

$$\dot{q} = \bar{\Theta} \dot{\bar{q}} \quad (5.1.13)$$

$$\ddot{q} = \bar{\Theta} \ddot{\bar{q}} \quad (5.1.14)$$

Summarizing, the coordinate transformation from absolute to relative are

$$\begin{cases} q = \bar{\Theta}\bar{q} + \bar{h} \\ \dot{q} = \bar{\Theta}\dot{\bar{q}} \\ \ddot{q} = \bar{\Theta}\ddot{\bar{q}} \end{cases} \quad (5.1.15)$$

and from relative to absolute

$$\begin{cases} \bar{q} = \Theta q + h \\ \dot{\bar{q}} = \Theta \dot{q} \\ \ddot{\bar{q}} = \Theta \ddot{q} \end{cases} \quad (5.1.16)$$

The coordinates transformation computed above has to be applied on the dynamic model (2.8.22) derived in the modeling chapter. By some simple reasoning it's possible to compute the model in absolute coordinates starting from the relative ones. The first step is to multiply both member for $\bar{\Theta}$ on the left; that is done to give to the computation a more elegant form. In this case it's considered zero external force, thus $F_{EXT} = 0$.

$$B(q)\ddot{q} + m(q, \dot{q}) + g(q) = B_u u - D\dot{q} \quad (5.1.17)$$

$$\bar{\Theta}(B(q)\ddot{q} + m(q, \dot{q}) + g(q)) = \bar{\Theta}(B_u u - D\dot{q}) \quad (5.1.18)$$

This model is expressed in relative coordinates, and the purpose is to express the same model with absolute coordinates: so the transformation needed is the one from absolute to relative summarized in (5.1.15). Applying this transformation the model in absolute coordinates is derived

$$\bar{\Theta}B(\bar{\Theta}\bar{q} + \bar{h})(\bar{\Theta}\ddot{q}) + \bar{\Theta}m(\bar{\Theta}\bar{q} + \bar{h}, \bar{\Theta}\dot{q}) + \bar{\Theta}g(\bar{\Theta}\bar{q} + \bar{h}) = \bar{\Theta}B_u u - \bar{\Theta}D\bar{\Theta}\dot{q}$$

New matrices and vectors of the model come directly from the above expression: it can be defined

$$\left\{ \begin{array}{l} \bar{B}(\bar{q}) = \bar{\Theta}B(\bar{\Theta}\bar{q} + \bar{h})\bar{\Theta} \\ \bar{m}(\bar{q}, \dot{\bar{q}}) = \bar{\Theta}m(\bar{\Theta}\bar{q} + \bar{h}, \bar{\Theta}\dot{q}) \\ \bar{g}(\bar{q}) = \bar{\Theta}g(\bar{\Theta}\bar{q} + \bar{h}) \\ \bar{B}_u = \bar{\Theta}B_u \\ \bar{D} = \bar{\Theta}D\bar{\Theta} \end{array} \right. \quad (5.1.19)$$

and the model in absolute coordinates becomes

$$\bar{B}(\bar{q})\ddot{\bar{q}} + \bar{m}(\bar{q}, \dot{\bar{q}}) + \bar{g}(\bar{q}) = \bar{B}_u u - \bar{D}\dot{\bar{q}} \quad (5.1.20)$$

Model (5.1.20) is equivalent to model (5.1.17), and is used for the control because it presents simpler computation for the output functions.

5.2 Feedback Linearization

Feedback Linearization is a nonlinear control technique based on the definition of a set of outputs to drive to zero through a coordinates transformation. Given

a nonlinear system

$$\begin{cases} \dot{x} = f(x) + g(x)u \\ y = h(x) \end{cases} \quad (5.2.1)$$

feedback linearization uses differential geometry tools, in particular the Lie derivative denoted by \mathcal{L} , to compute the derivative of the output. It evaluates the change of a vector field along the flow of another one.

In the case of zero input system

$$\dot{y} = \frac{dh(x)}{dx} \frac{dx}{dt} = \frac{dh(x)}{dx} f(x) = \mathcal{L}_f h(x) \quad (5.2.2)$$

In the case of a system with input in the form of (5.2.1) the derivative is

$$\dot{y} = \frac{dh(x)}{dx} \frac{dx}{dt} = \frac{dh(x)}{dx} f(x) + \frac{dh(x)}{dx} g(x)u = \mathcal{L}_f h(x) + \mathcal{L}_g h(x)u \quad (5.2.3)$$

The notation of Lie derivative is useful to express in compact form the higher order derivatives. In general the n – th derivative is:

$$y^{(n)} = \mathcal{L}_f^n h(x) + \mathcal{L}_g \mathcal{L}_f^{n-1} h(x)u \quad (5.2.4)$$

An important concept to be introduced is the relative degree. In linear SISO systems it is the difference between the number of poles and the number of zeros of the transfer function. It indicates which is the lowest order of the output derivative that depends on the input u . In nonlinear system the interpretation is analogous, but for the definition it's necessary to use Lie derivative formalism. A system with r relative degree for x in the neighborhood of \bar{x} is a system such that

$$\begin{cases} \mathcal{L}_g \mathcal{L}_f^k h(x) = 0 \forall x \in C(\bar{x}) \quad \forall k < r-1 \\ \mathcal{L}_g \mathcal{L}_f^{k-1} h(\bar{x}) \neq 0 \end{cases} \quad (5.2.5)$$

where $C(\bar{x})$ is in the neighborhood of \bar{x} .

In feedback linearization control application the relative degree must be $n - 1$. In the case it's less than $n - 1$ in the new system will appear also input derivative, and it can create some problems because it will be necessary an estimator.

The case of a MIMO system is:

$$\begin{cases} \dot{x} = f(x) + g(x)u \\ y_1 = h_1(x) \\ \vdots \\ y_q = h_q(x) \end{cases} \quad (5.2.6)$$

each output h_i has a relative degree, depending on how it is defined. A diffeomorphism $\Phi(x)$ has to be defined in order pass from the current coordinate system to the system in which the state variables are the output functions and their derivatives. It must include all outputs and the maximum order of the derivative of each output must be the relative degree of that output. In other words, each output function must be derived until the input u appears in its expression: in this way the $i - th$ output can be controlled through the input that influences its $k - th$ derivative, where k is the relative degree of the $i - th$ output. On the one hand if that output is derived one more time, the expression of the $(k + 1) - th$ derivative will contain the derivative of the input \dot{u} , that makes the control very difficult; on the other hand if that output is derived one less time, the input u will not appear on the expression of the $(k - 1) - th$ derivative, and there will not be any possibility to drive that output to zero.

Therefore the coordinates transformation must be an invertible diffeomorphism of the form

$$\Phi(x) = \begin{pmatrix} h_1 \\ \mathcal{L}_f h_1 \\ \vdots \\ \mathcal{L}_f^{n-1} h_1 + \mathcal{L}_g \mathcal{L}_f^{n-2} h_1 u \\ h_2 \\ \mathcal{L}_f h_2 \\ \vdots \\ \mathcal{L}_f^{n-1} h_2 + \mathcal{L}_g \mathcal{L}_f^{n-2} h_2 u \\ \vdots \\ \vdots \\ \vdots \\ h_q \\ \mathcal{L}_f h_q \\ \vdots \\ \mathcal{L}_f^{n-1} h_q + \mathcal{L}_g \mathcal{L}_f^{n-2} h_q u \end{pmatrix} \quad (5.2.7)$$

The choice of the output functions h_i is not unique. A necessary condition for the realizability of the controller is that the diffeomorphism that describes the change of coordinates is invertible. In fact if it's not, it will not be possible to come back from the new coordinates defined through the outputs to the old ones.

The output functions have to be chosen in order to achieve the desired behavior of the system: in the case of gait control of humanoid robot, there are many possible choice. The one that is going to be analyzed has been proved to

be very efficient.

5.2.1 The output function choice

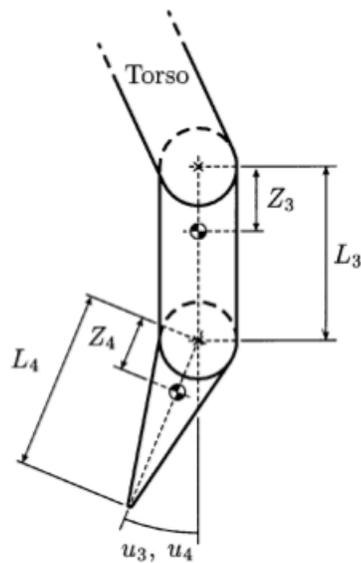


Figure 5.2.1: The length of femur L_3 and the length of the tibia L_4

The humanoid gait problem can be expressed through some simple and intuitive conditions:

- the robot must keep an erect and stable position during his walk
- the robot must not tip forward or backward
- the body must move forward during the walk, with moderate vertical

oscillations

- the foot must track a proper trajectory with a given step length

This intuitive conditions have to be transformed into output functions to be driven to zero. For the definition of output functions it's appropriate to define some variables that make the computation easier, shown in figure 5.2.2. For uniformity of notation with the article by Grizzle [5] it has been introduced the variable L_3 and L_4 to express the length of the femur and the tibia respectively as shown in figure 5.2.1.

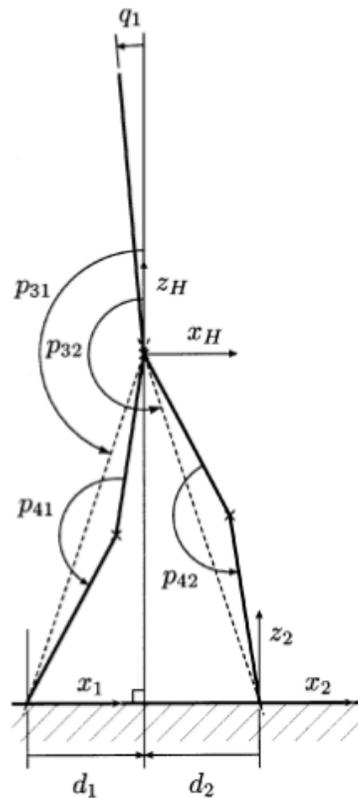


Figure 5.2.2: Graphic of the variables defined in (5.2.8)

$$\left\{ \begin{array}{l} x_1 = 0 \\ z_1 = 0 \\ x_H = L_3 \cdot \sin(\bar{q}_{31}) + L_4 \cdot \sin(\bar{q}_{41}) \\ z_H = -L_3 \cdot \cos(\bar{q}_{31}) - L_4 \cdot \cos(\bar{q}_{41}) \\ x_2 = x_H - L_3 \cdot \sin(\bar{q}_{32}) - L_4 \cdot \sin(\bar{q}_{42}) \\ z_2 = z_H + L_3 \cdot \cos(\bar{q}_{32}) + L_4 \cdot \cos(\bar{q}_{42}) \\ d_1 = x_H - x_1 = L_3 \cdot \sin(\bar{q}_{31}) + L_4 \cdot \sin(\bar{q}_{41}) \\ d_2 = x_H - x_2 = L_3 \cdot \sin(\bar{q}_{32}) + L_4 \cdot \sin(\bar{q}_{42}) \end{array} \right. \quad (5.2.8)$$

The variable $\begin{pmatrix} x_1 \\ z_1 \end{pmatrix}$ is the Cartesian position of the stance foot, posed on 0 for the first step; $\begin{pmatrix} x_H \\ z_H \end{pmatrix}$ is the Cartesian position of the hip, $\begin{pmatrix} x_2 \\ z_2 \end{pmatrix}$ the Cartesian position of the swing foot. The variables d_1 and d_2 are the horizontal distances between the hip and the two feet as shown in figure 5.2.2. Given this notation, four output functions can be defined, in order to achieve the desired behavior of the robot. Each output is multiplied for a gain k_i useful for the controller tuning.

The first output h_1 is defined as

$$h_1 = k_1 \cdot (\bar{q}_1 - \bar{q}_{1d}) \quad (5.2.9)$$

where \bar{q}_{1d} is the desired position of the torso. This function is important for the stability of the upper part of the humanoid and, if driven to zero, keeps the torso in the desired position. For a gait application the torso must have an almost vertical position, slightly tilted forward.

The second output h_2 is defined as

$$h_2 = k_2 \cdot (d_1 + d_2) \quad (5.2.10)$$

If $h_2 = 0$ then $d_1 = -d_2$: in this case the the hip projection on the floor coincides with the medium point between the two feet. This variable is useful for the stability of the lower part of the robot. By driving h_2 to zero the robot keeps a stable and erect position, and it doesn't tip forward or backward.

The third output h_3 is defined as

$$h_3 = k_3 \cdot (z_H - z_{Hd}(d_1)) \quad (5.2.11)$$

This variable ensures the desired behavior of the hip given by the reference trajectory $z_{Hd}(d_1)$ as a function of d_1 . The best trajectory to be used is a parabolic trajectory with upper and lower bounds defined by $z_{H \max}$ and $z_{H \min}$ respectively. The expression chosen for $z_{Hd}(d_1)$ is

$$z_{Hd}(d_1) = \frac{4(z_{H \min} - z_{H \max})}{sld^2} d_1^2 + z_{H \max} \quad (5.2.12)$$

where sld is the step length desired. This expression of $z_{Hd}(d_1)$ represents a parabola defined in the interval $d_1 \in \left[-\frac{sld}{2}, \frac{sld}{2}\right]$ with the properties:

$$\begin{cases} z_{Hd}\left(-\frac{sld}{2}\right) = z_{H \min} \\ z_{Hd}(0) = z_{H \max} \\ z_{Hd}\left(\frac{sld}{2}\right) = z_{H \min} \end{cases} \quad (5.2.13)$$

The fourth output h_4 is defined as

$$h_4 = k_4 \cdot (z_2 - z_{2d}(d_1)) \quad (5.2.14)$$

This variable, as $z_{Hd}(d_1)$, ensures the desired behavior of the swing foot given by the reference trajectory $z_{2d}(d_1)$ as a function of d_1 . The best trajectory to be used is a parabolic trajectory with upper bound defined by $z_{2\max}$. The expression chosen for $z_{2d}(d_1)$ is

$$z_{2d}(d_1) = \frac{-4z_{2\max}}{sl^2}d_1^2 + z_{2\max} \quad (5.2.15)$$

This expression of $z_{2d}(d_1)$ represents a parabola defined in the interval $d_1 \in \left[-\frac{sl}{2}, \frac{sl}{2}\right]$ with the properties:

$$\begin{cases} z_{2d}\left(-\frac{sl}{2}\right) = 0 \\ z_{2d}(0) = z_{2\max} \\ z_{2d}\left(\frac{sl}{2}\right) = 0 \end{cases} \quad (5.2.16)$$

Overall the vector y has the form

$$y = h(x) = \begin{pmatrix} h_1(x) \\ h_2(x) \\ h_3(x) \\ h_4(x) \end{pmatrix} = \begin{pmatrix} k_1 \cdot (\bar{q}_1 - \bar{q}_{1d}) \\ k_2 \cdot (d_1 + d_2) \\ k_3 \cdot (z_H - z_{Hd}(d_1)) \\ k_4 \cdot (z_2 - z_{2d}(d_1)) \end{pmatrix} \quad (5.2.17)$$

5.2.2 Controller Synthesis

The control objective is to drive the output defined in equation (5.2.17) to zero. Since the output (5.2.17) only depends on generalized position q and the dy-

dynamic model is second order as typical in mechanic systems, the relative degree of each output is two. Using standard Lie notation direct calculus yields:

$$\ddot{y} = \mathcal{L}_f^2 h(x) + \mathcal{L}_g \mathcal{L}_f h(x) \cdot u \quad (5.2.18)$$

To ensure the controllability of the system in the new coordinates the matrix $\mathcal{L}_g \mathcal{L}_f h(x)$ must be invertible in the region of interest. The demonstration of that is dealt in the article of Grizzle [5]. The method of inverse dynamics can be used to define the new control input $v = \ddot{y}$ equal to the second derivative, thus $v = \mathcal{L}_f^2 h(x) + \mathcal{L}_g \mathcal{L}_f h(x) \cdot u$, resulting in four double integrators $\ddot{y}_i = v_i, i = 1, 2, 3, 4$. The approach used in this thesis is to apply a linear controller for each double integrator of the form

$$v_i = -k_{i1}y_i - k_{i2}\dot{y}_i. \quad (5.2.19)$$

The choice of the gain is done by considering that the convergence to zero must be achieved in a time interval smaller than the time interval of a single step. For this purpose Grizzle uses a discontinuous nonlinear high gain controller that stabilizes the system in finite time. The choice of the parameters is done in order to obtain convergence time smaller than the step time. In this thesis it has been used a linear controller that gives an asymptotic convergence, but the gains are chosen high in order to have a quick convergence. From the controller form in equation (5.2.19), given $v = \ddot{y}$, it can be deduced the characteristic polynomial and the transfer function:

$$\ddot{y} + k_{i2}\dot{y}_i + k_{i1}y_i = 0 \quad (5.2.20)$$

$$s^2 + k_{i2}s + k_{i1} = 0 \quad (5.2.21)$$

The second order transfer function (5.2.21) has two poles λ_1 and λ_2 , and can be written as

$$(s - \lambda_1)(s - \lambda_2) = 0 \quad (5.2.22)$$

$$s^2 - (\lambda_1 + \lambda_2)s + \lambda_1\lambda_2 = 0 \quad (5.2.23)$$

By comparing equation (5.2.21) and (5.2.23) it comes that

$$\begin{cases} k_{i1} = \lambda_1\lambda_2 \\ k_{i2} = -(\lambda_1 + \lambda_2) \end{cases} \quad (5.2.24)$$

So the pole assignment can be easily done through the parameters k_{i1} and k_{i2} that represents the proportional and the derivative coefficients. In the simulation done in this thesis the poles λ_1 and λ_2 are posed respectively on -100 and -120 . They are high because the convergence must be achieved in a time smaller than the step time. With a linear controller like that the convergence is asymptotic, so it's not achieved in finite time, but in approximation after a certain time interval the error becomes so small to be considered negligible.

5.3 Impact model and gait stability

After developing the controller for a single step, the final objective is to extend that to a multiple steps situation, to obtain a walking pattern. Therefore the

impact model must be derived in order to compute the reset map of the hybrid transition associated with the impact between the swing foot and the ground. The impact effect on the humanoid robot can be approximated with a discontinuity on the joint velocity. At the moment of the discontinuity the control algorithm, that is function of the joints velocity, gives a discontinuity on the inputs as well. The stability of the controller for a single step doesn't ensure the stability of the gait: intuitively it can be considered a situation in which the initial state of the robot belongs to the convergence region of the controller, thus it's able to drive the output $h(x)$ to zero and the robot can make its first step. At the moment of the impact the system resets, so the controller finds a system with a new initial state to stabilize. The new initial state may or may not belong to the convergence region: if not then the controller is no longer able to stabilize the system and the robot tips forward on its second step; if so then the robot is able to make its second step, but there exists the possibility that step after step the new initial state goes towards the border of the convergence region and after a certain number n of steps it goes beyond that. When this happens the controller is no longer able to stabilize the system, so after n steps the robot tips forward. Therefore the study of the stability of the walk is not a simple topic, but it exists a mathematical tool used to analyze the gait orbit stability starting from the initial state of the robot. This tool is called Poincaré map and is analyzed further in this section.

5.3.1 Impact model

When the swing foot touches the floor, the controller objective is successfully reached. However for a gait application it's necessary to make multiple steps: for this purpose a hybrid model is used. The impact between the swing leg and the ground is modeled as a contact between two rigid bodies. The basic hypotheses are:

- the contact of the swing leg with the ground results in no rebound and no slipping of the swing leg
- at the moment of impact, the stance leg lifts from the ground without interaction
- the impact is instantaneous
- the external forces during the impact can be represented by impulses
- the impulsive forces may result in an instantaneous change in the velocities, but there is no instantaneous change in the positions
- the torques supplied by the actuators are not impulsive

From this hypotheses the angular momentum is conserved about the impact point, thus

$$\bar{B}_i (\dot{q}^+ - \dot{q}^-) = F_{ext} \quad (5.3.1)$$

where \dot{q}^- is the joints velocity before the impact, \dot{q}^+ is the joints velocity after the impact and \bar{B}_i is the \bar{B} matrix computed on the impact point. F_{EXT} is a generalized force, so in the case of humanoid robot is the vector of the joint torques. It is the result of the contact impulse forces and can be expressed with the integral of the impulse between the impact initial and final times.

$$F_{ext} = \int_{t^-}^{t^+} \delta F_{ext}(\tau) d\tau \quad (5.3.2)$$

where δF_{ext} is the contact impulse. Since the joint configuration does not change after the impact the relation $\dot{q}^+ = \dot{q}^-$ is valid.

In order to be able to solve for all of the unknowns, the above equations must be augmented with additional equations that proscribe what happens at the two contact ends.

The swing foot position given by the coordinates $p_{swf} = \begin{pmatrix} x_2 \\ z_2 \end{pmatrix}$ is expressed in equation (5.2.8) as

$$p_{swf} = \begin{pmatrix} x_2 \\ z_2 \end{pmatrix} = \begin{pmatrix} L_3 \cdot \sin(\bar{q}_{31}) + L_4 \cdot \sin(\bar{q}_{41}) - L_3 \cdot \sin(\bar{q}_{32}) - L_4 \cdot \sin(\bar{q}_{42}) \\ -L_3 \cdot \cos(\bar{q}_{31}) - L_4 \cdot \cos(\bar{q}_{41}) + L_3 \cdot \cos(\bar{q}_{32}) + L_4 \cdot \cos(\bar{q}_{42}) \end{pmatrix} \quad (5.3.3)$$

Generalized forces can be expressed with the formula (2.8.21) dealt in the modeling chapter (2) through the transpose of the Jacobian of the swing foot J_{swf}

$$J_{swf} = \frac{\partial p_{swf}}{\partial \bar{q}} = \begin{pmatrix} L_3 \cdot \cos(\bar{q}_{31}) & -L_3 \cdot \cos(\bar{q}_{32}) & L_4 \cdot \cos(\bar{q}_{41}) & -L_4 \cdot \cos(\bar{q}_{42}) & 0 \\ L_3 \cdot \sin(\bar{q}_{31}) & -L_3 \cdot \sin(\bar{q}_{32}) & L_4 \cdot \sin(\bar{q}_{41}) & -L_4 \cdot \sin(\bar{q}_{42}) & 0 \end{pmatrix} \quad (5.3.4)$$

as

$$F_{ext} = J_{swf}^T \begin{pmatrix} F_T \\ F_N \end{pmatrix} \quad (5.3.5)$$

where F_T and F_N are the tangent and normal forces, respectively, applied at the end of the swing leg.

An additional set of two equations is obtained from the condition that the swing leg does not rebound nor slip at impact, thus the Cartesian velocity of the swing foot after the impact is null

$$J_{swf}\dot{q}^+ = 0 \quad (5.3.6)$$

Equations (5.3.1) and (5.3.6) can be combined into a linear system of the form

$$\begin{pmatrix} \bar{B}_i & -J_{swf}^T \\ J_{swf} & O_{2 \times 2} \end{pmatrix} \begin{pmatrix} \dot{q}^+ \\ \bar{F}_{ext} \end{pmatrix} = \begin{pmatrix} \dot{q}^- \\ O_{2 \times 1} \end{pmatrix} \quad (5.3.7)$$

where

$$\bar{F}_{ext} = \begin{pmatrix} F_T \\ F_N \end{pmatrix} \quad (5.3.8)$$

This is the equation with 7 unknowns that when solved gives the velocity after the impact and the external generalized force applied by the ground on the swing foot. Through the solution of this system, it can be computed the reset map of the hybrid transition associated to the contact of the swing foot with the ground. This reset map depends on the joint position \bar{q} and joint velocity $\dot{\bar{q}}$ thus it must be recomputed at each impact.

Moreover the reset map computed above is not sufficient to develop the hybrid controller because it doesn't consider the exchange of the feet after the impact: in fact in that moment, to ensure the cyclicity of the gait, left foot has to become right foot and vice versa. So the relation $\dot{q}^+ = \dot{q}^-$ is valid under a theoretical point of view, but doesn't consider the feet exchange: in the physic system no feet exchange really happens but for the project of the hybrid controller this trick has been used, thanks to the symmetry of legs, to have a single-state hybrid system instead of a double-state one, in order to consider-

ably simplify the model and the computation. This can be obtained through the left product for a matrix that changes the order of the coordinates:

$$\dot{q}^+ = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \dot{q}^- \quad (5.3.9)$$

Overall, the reset map computation is divided in two phases: the first one is dedicated to the calculus of the new joint velocity \dot{q}^+ , done through the solution of the impact model linear system described in equation (5.3.7). The second one is dedicated to the calculus of the new joint position, simply done through the application of a switching matrix as in equation (5.3.9).

To obtain a compact notation, the reset map of the whole state is denoted by $\Delta(\cdot)$, thus

$$x(t^+) = \Delta(x(t^-)) \quad (5.3.10)$$

Overall, recalling the notation in (2.8.25) the hybrid system can be written as

$$\begin{cases} \dot{x} = f(x) + g(x)u & \text{when } x \notin \mathcal{S} \\ x(t^+) = \Delta(x(t^-)) & \text{when } x \in \mathcal{S} \end{cases} \quad (5.3.11)$$

where \mathcal{S} is the hyper surface in which the swing foot has negative ordinate, defined as

$$\mathcal{S} = \{x \in \mathbb{R}^{10} \mid z_2 < 0\} \quad (5.3.12)$$

5.3.2 Poincaré map

Poincaré map is a very powerful tool used for the study of orbit convergence; it comes from astrophysics, in which it's used to study the stability of the orbits of planets. In the case of humanoid robot it's used to study the stability of the gait that, by the fact that it's a pseudo-periodic motion, can be seen as an orbit in the state space of the system. If the gait orbit converges, after some steps the walking pattern will become periodic; if it diverges, step after step the robot will go towards a situation of instability and imbalance, until it will tip forward. As explained at the beginning of this section, the study of the stability of the walk is not a simple topic.

At first, it's necessary to find a transversal hyper surface that intersect all the trajectories of the dynamics. In the case of hybrid system the best choice of the hyper surface is the one that defines the guard condition, in this case \mathcal{S} , defined in (5.3.12). The sequence of intersections with \mathcal{S} for a given realization of the hybrid system, that is a sequence of states at the moment immediately following the impact $\{x_1^+, x_2^+, x_3^+, \dots\}$, defines a map $P : \mathcal{S} \rightarrow \mathcal{S}$ that is denominated Poincaré map and is such that:

$$\begin{cases} x_2^+ = P(x_1^+) \\ x_3^+ = P(x_2^+) \\ \vdots \end{cases} \quad (5.3.13)$$

Poincaré map $P(\cdot)$ is very difficult to compute in a closed form and in this thesis is dealt only under a theoretical point of view. Thanks to the definition of $P(\cdot)$, the sequence of states that intersect the hyper surface \mathcal{S} may be expressed as a discrete time system of the form

$$x_{k+1} = P(x_k) \quad (5.3.14)$$

A fixed point x_f of the map $P(\cdot)$ is a point for which the relation

$$x_f = P(x_f) \quad (5.3.15)$$

is true.

Stability of systems in form of (5.3.14) has been widely studied in control theory, and can be done with Lyapunov method or by the evaluation of Jacobian matrix eigenvalues. The definition of fixed point (5.3.15) in Poincaré map is equivalent to the definition of equilibrium point in dynamic discrete time system, so all the theories valid for that kind of systems can be used to study the equilibrium of fixed points. Therefore if the Jacobian matrix of $P(\cdot)$ computed on the fixed point x_f

$$J_P(x_f) = \left. \frac{\partial P(x)}{\partial x} \right|_{x=x_f}$$

has all his eigenvalues λ on the unitary hyper ball, i.e. $|\lambda_i| < 1 \forall i$, the fixed point x_f is stable. If there is an eigenvalue λ_q with $|\lambda_q| > 1$ the fixed point x_f is unstable.

Once fixed point x_f has been found, it follows that the equilibrium orbit is the one which starts from that point and evolves according to the dynamical model of the system. Roughly speaking, with Poincaré theory the study of the convergence of the orbits to a periodic one is reduced to the study of the convergence of the sequence of intersections with the hyper surface S to a fixed point.

Poincaré map is a very powerful mathematics tool and a very wide topic that can be used for many purposes, including robustness analysis under external perturbations, robust controller synthesis and steady state parameters computation. More detailed explanations are illustrated in the researches of

Grizzle [4, 5, 6].

Chapter 6

Simulation Results and Conclusions

The simulation is done by MATLAB[®] and gives nice results. The walking animation is very fluid and very robust under parameters variations and initial joints velocity variations. It's shown in figure [6.0.1](#).

The four outputs are driven to zero in short time as shown in figure [6.0.2](#). During the impacts the discontinuity of the space variable causes a sudden deviation of the output between the zero reference, but the controller provides to drive them back to zero.

Joint velocities and accelerations are shown in figure [6.0.3](#). Controller gains

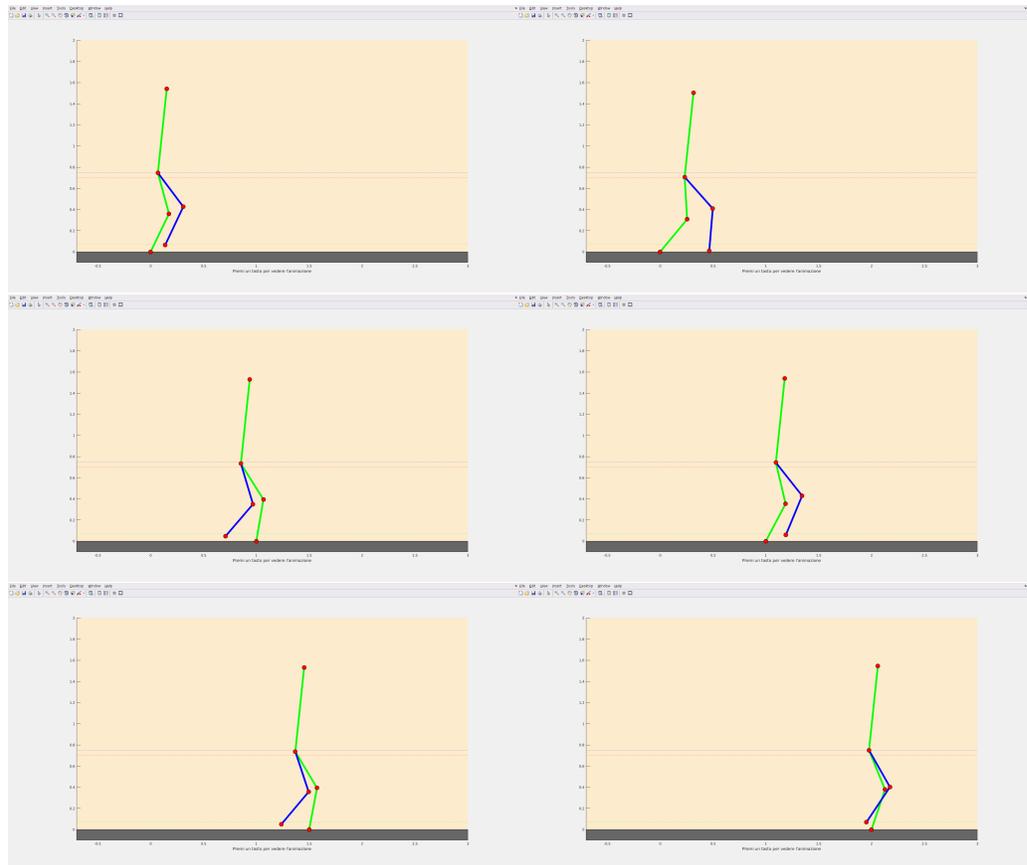


Figure 6.0.1: Walking Animation

are high because for a stable walk it's needed the convergence of output to zero in a time interval smaller than the step time. If this objective is not achieved, the walking orbit can become unstable.

Accelerations are very big at the beginning of the simulation because the initial state of the robot is such that the output $h(x)$ is far from the zero value. Thus the controller provides to apply big torques to drive it to zero. Once the outputs are near to zero, the torque required to stabilize the walking orbit is quite small. During the impacts it can be observed a modest spike on torque functions: this is caused by the spike on the output shown in figure 6.0.2 dis-

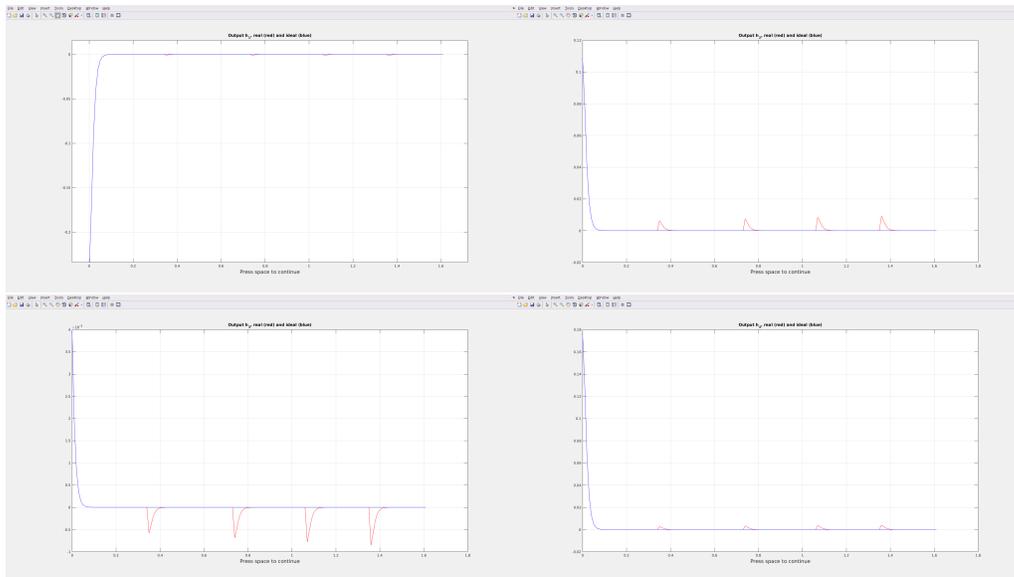


Figure 6.0.2: Output functions plots

cussed previously that are quickly compensated by the controller.

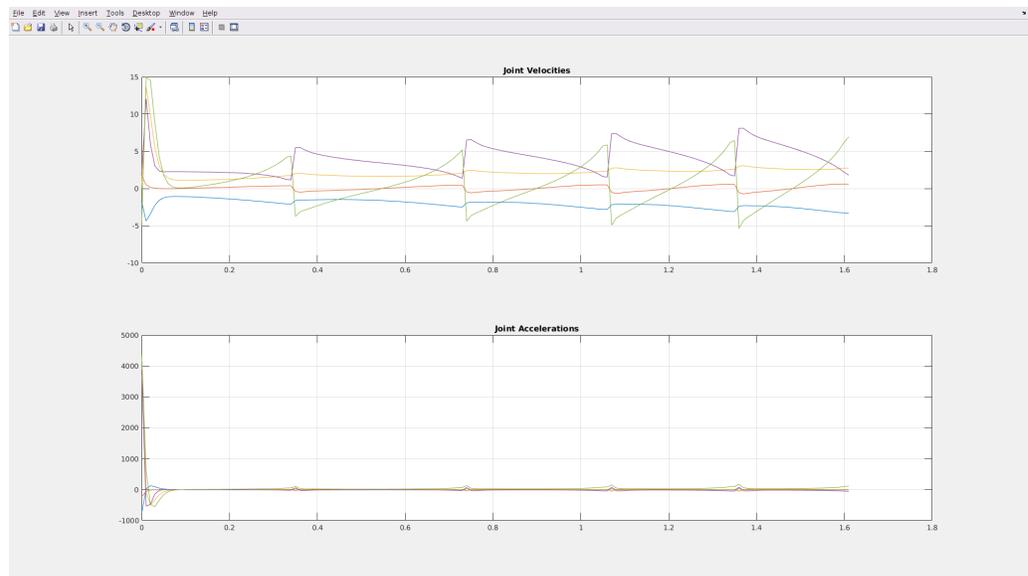


Figure 6.0.3: Joint velocities and accelerations

Bibliography

- [1] Jerry Pratt, John Carff, Sergey Drakunov, Ambarish Goswami, "Capture point: A step toward humanoid push recovery" - *2006 6th IEEE-RAS International Conference on Humanoid Robots. IEEE, 2006*
- [2] Milad Shafiee-Ashtiani, Aghil Yousefi-Koma, Masoud Shariat-Panahi, Majid Khadiv, "Push Recovery of a Humanoid Robot Based on Model Predictive Control and Capture Point", *International Conference on Robotics and Mechatronics October 26-28, 2016, Tehran, Iran*
- [3] M.H.P. Dekker, "ZERO-MOMENT POINT METHOD FOR STABLE BIPED WALKING" - *Eindhoven, July 2009*
- [4] Jesse W. Grizzle, Gabriel Abba, and Franck Plestan, "Asymptotically Stable Walking for Biped Robots: Analysis via Systems with Impulse Effects" - *IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. 46, NO. 1, JANUARY 2001*
- [5] Franck Plestan, Jessy W. Grizzle, Eric R. Westervelt, Gabriel Abba, "Stable Walking of a 7-DOF Biped Robot" - *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, VOL. 19, NO. 4, AUGUST 2003*

- [6] B. Morris, J.W. Grizzle "A Restricted Poincaré Map for Determining Exponentially Stable Periodic Orbits in Systems with Impulse Effects: Application to Bipedal Robots" - *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005 Seville, Spain, December 12-15, 2005*
- [7] Daniel Liberzon, "Calculus of variation and optimal control theory - A concise introduction" - *Lecture notes*
- [8] Enzeng Dong, Dandan Wang, Chao Chen, Jigang Tong, "Realization of biped robot gait planning based on NAO robot development platform" - *Proceedings of 2016 IEEE International Conference on Mechatronics and Automation August 7 - 10, Harbin, China*
- [9] Benjamin Stephens, "Push Recovery Control for Force-Controlled Humanoid Robots" - *PhD thesis, Robotics Institute Carnegie Mellon University Pittsburgh, Pennsylvania USA*
- [10] Riccardo De Luca, Costanzo Manes, "Modellistica e controllo di robot bipedi tramite stabilizzazione di dinamiche zero-ibride" - *Tesi di Laurea Specialistica in Ingegneria Informatica e Automatica, Università degli Studi dell'Aquila*
- [11] Asif Šabanović, Kouhei Ohnishi, "Motion Control System" - © 2011 John Wiley & Sons (Asia) Pte Ltd. Published 2011 by John Wiley & Sons (Asia) Pte Ltd. ISBN: 978-0-470-82573-0
- [12] Bruno Siciliano, Oussama Khatib (Eds.), "Handbook of Robotics" - *Springer-Verlag Berlin Heidelberg 2008 - ISBN: 978-3-540-23957-4*
- [13] Costanzo Manes, "Appunti del corso di Robotica Industriale" - *Università degli Studi dell'Aquila*

- [14] Alfredo Germani, "Appunti del corso di Complementi di Automatica" - *Università degli Studi dell'Aquila*
- [15] Maria D. Di Benedetto, "Appunti del corso di Sistemi Ibridi" - *Università degli Studi dell'Aquila*
- [16] Xin Xin, Masahiro Kaneda, "Swing-Up Control for a 3-DOF Gymnastic Robot With Passive First Joint: Design and Analysis" - *IEEE TRANSACTIONS ON ROBOTICS, VOL. 23, NO. 6, DECEMBER 2007*
- [17] Marko B. Popovic, Ambarish Goswami, Hugh Herr, "Ground Reference Points in Legged Locomotion: Definitions, Biological Trajectories and Control Implications" - *The International Journal of Robotics Research Vol. 24, No. 10, October 2005, pp. xxx-xxx, DOI: 10.1177/0278364905058363 ©2005 Sage Publications*
- [18] John Lygeros, "Lecture Notes on Hybrid Systems" - *Automatic Control Laboratory ETH Zurich CH-8092, Zurich, Switzerland*

Agli Amici

Agli Amici,
che dopo una lunga giornata di studio mi ricaricano le batterie.

A Flavia, sorella acquisita da una vita, alle guide spericolate in macchina a 15 anni, le imitazioni dei professori, i discorsi altolocati sulla mafia, la corruzione, zio Benito e MEB, le speranze periodicamente infrante sui dimagrimenti insieme a Feliciani, i discorsi filosofici, la porchetta di Acili....ARICCIA, le feste nelle case private a Giulianova da imbucati senza conoscere nessuno.

Alla Signorina[®], le camminate in montagna, la signora alta, le infinite vicende di vita quotidiana che ci accadono ogni giorno, Nadine di Twin Peaks, i Cosmopolitan, a Valerio e Giorgia e i viaggi in macchina verso i concerti, le JAM SESSIONS, le avventure a Perugia e Roma, a Anna e le feste con strane ultracinquantenni imbucate, le sue puntuali 2 ore di ritardo e a sua sorella, sorella anche mia, Giulia, e i tempi in cui lei faceva il primo superiore e io il quinto e la

introducevo al mondo liceale, poi ripetutisi in modo analogo quando lei faceva il primo anno di università e io il quinto. E a Giuseppe e il suo chiodo fisso, sempre lo stesso, e ai suoi discorsi dove a me cadono ogni volta le braccia mentre Rosa e Claudia ridono per non piangere.

A Corrado e le avventure sulle montagne, sulla neve, nelle profondità dei mari, la mezza maratona che forse un giorno farò, le pazzie con le moto, le trazioni con Ludovico al castello, e le abbuffate ai ristoranti con Alessia.

A Claudia S e le interminabili attese alla stanza della musica, all'accordo dodecafonico iniziale della Patetica, alle lezioni di balli tradizionali polacchi di Hanzelewicz, agli strani incontri della notte prima del mio diploma e le serate passate a suonare e cantare le canzoni di de Andrè con Agnese; alle serate da San Basilio fino a San Lorenzo con Agnese terminate alle otto di mattina quando tra il sonno e il sole fortissimo in faccia non riuscivo a tenere gli occhi aperti per guidare. A Simone, le uscite notturne per Bevagna e il nostro colossale secondo concerto di Rachmaninov.

A Pasquarella, Sokolov, i viaggi improvvisati a Roma, le antiche rovine romane e le signore schizofreniche che ci fermano per strada per parlarci insistentemente delle loro allucinazioni su Ponzio Pilato e San Pietro affermando che sono le uniche al mondo a conoscere le risposte dei grandi enigmi dell'esistenza umana.

A Laura e le strane serate rimasti chiusi fuori casa a svegliare il vicinato col sassofono, i tempi del solfeggio quando sognavo di fare il jazzista e improvvisavo di nascosto, e le serate assurde in centro con Pamela tirate avanti fino al pomeriggio del giorno dopo stando casa mia col vinile dei Pink Floyd

e l'incenso dagli odori orientali. Agli altri amici musicisti, il mio alter ego musicale Federico S, la grande settimana di Beethoven con lui al pianoforte e io al basso elettrico e la nostra spettacolare improvvisazione a 4 mani concordata 2 minuti prima di salire sul palco; alle gemelle Sara e Martina e le cene con la pizza fatta in casa e i giri in macchina, e a Simona e la sua ancia per oboe che custodisco in una teca di vetro. E a Claudia P e al pazzo giorno in cui ci siamo svegliati la mattina in Germania e dal nulla abbiamo deciso di prendere la macchina e andare ad Amsterdam senza nemmeno prenotare l'albergo, al nostro compagno di stanza gallese uguale a Van Gogh, ai parcheggi privati con le sbarre dove siamo rimasti chiusi dentro, e alle estati a San Benedetto.

A Pigna e Elisa la naturalizzata americana, le suonate con i Madness, i succhi di frutta alla pera dentro i piatti, la tastiera che mi cadde durante il concerto della giornata dell'arte, il concerto che venni a fare zoppicando e coi pantaloni strappati perchè ero appena caduto dalla moto, e le serate in centro con Scalzini e Luca. A Valeria e i suoi singolari regali di compleanno.

A Ludovica, amica da una vita, compagna di confessioni e consulenze, e alla vecchia storia dello zaino nascosto con tutti i libri dentro.

A Pierluigi e il Manager e la storica vacanza a Villa Rosa, iniziata alla grande con una piacevolissima gita all'ospedale di Sant'Omero e terminata con un feragosto assurdo in spiaggia fino all'alba.

Ai compagni dell'Università, Emanuele che da piccolo voleva fare l'automatico ma poi non ce l'ha fatta, a Enrica e "Cecà, ci stanno novità???", a Gioele e Irene e Venosaur che un giorno troveremo, a Carlotta e Anthony, le serate all'Andalucia e le pizze di Collarmele, e a tutti gli altri e gli scherzi telefonici di ogni genere.

Ai compagni di corso dell'Università Carlotta, Alberto, Marco M, Marco I, Aldo, Fabrizio, Tommaso, che per cinque lunghi anni MI HANNO PRESTATO LE PENNE E I FOGLI!

Ai ragazzi dell'IECON Mattia, Eleonora, Mario, Marino, Filippo, Aleandro, Andrea, Edy, Alessio, Han, e Francesca che si è aggiunta dopo, e alle 2000 fatture e tessere da stampare, le fiorentine da mezzo kilo, i miei sbrocchi contro i pakistani che facevano inceppare la stampante, le serate contemplative a Santa Maria del Fiore.

Ai ragazzi del teatro, le entrate inaspettate del fantasma, ai passati remoti improbabili, le imprecazioni contro le hostess e i piloti, e le allergie al MCDonald.

E infine a Cristina, i ragazzi della DigiPower a fianco dei quali ho lavorato quest'estate per la tesi, i Lithium Quartet e le atmosfere psichedeliche, e il club di Sant'Agnese che ormai va avanti da almeno 10 anni;

alle persone italiane o di nazionalità diversa che ho incontrato all'università, in conservatorio, nei master di musica, al mare, nella movida aquilana o nei miei viaggi, che anche stando a contatto con me solo per qualche settimana, qualche giorno e in alcuni casi anche qualche ora, sono rimaste impresse dentro di me; a tutte le persone appartenenti al mio passato che hanno stampato la loro impronta dentro di me e a tutte le persone che hanno sempre creduto in me.

Federico