



Universidad
Carlos III de Madrid

DEPARTMENT OF SYSTEMS ENGINEERING AND AUTOMATION

MASTER THESIS

WHOLE BODY TRAJECTORY GENERATION FOR A HUMANOID ROBOT

Autor: Domingo Francisco Esteban Cabala

Tutor: Prof. Santiago Martínez de la Casa Díaz

Director: Prof. Carlos Balaguer Bernaldo de Quirós

A Thesis Submitted for the Degree of
M.Sc. in Robotics and Automation

LEGANÉS, MADRID, SPAIN

2014

UNIVERSIDAD CARLOS III DE MADRID
MÁSTER OFICIAL EN ROBÓTICA Y AUTOMATIZACIÓN

El tribunal aprueba la Tesis de Máster titulada "**WHOLE BODY TRAJECTORY GENERATION FOR A HUMANOID ROBOT**" realizada por **Domingo Francisco Esteban Cabala**.

Fecha: Julio 2014

Tribunal: _____
Dr./Dra.

_____ Dr./Dra.

_____ Dr./Dra.

“Ser más para servir mejor ...”

S.I.D.L.

Contents

Index of Tables	xi
Index of Figures	xiii
Acknowledgments	xvii
Resumen	xix
Abstract	xxi
1 Introduction	1
1.1 Humanoid Robotics	2
1.1.1 Characteristics of the humanoid robots	3
1.2 Objectives	4
1.2.1 General objective	4
1.2.2 Specific objectives	4
1.3 Organization of the chapters	4
2 TEO Humanoid Robot	7
2.1 TEO: Task Environment Operator	7
2.2 Mechatronic systems of TEO humanoid	8
2.2.1 Sensorial system	8
2.2.2 Exteroceptive perception	9

2.2.3	Proprioceptive perception	9
2.2.4	Integration centres	10
2.2.5	Action System	10
2.2.6	Communication channels	10
3	Humanoid robots representation	13
3.1	Limbs of a humanoid robot: manipulators	13
3.2	Generalized coordinates space	13
3.3	Operational Space	15
4	Humanoid robots kinematics	17
4.1	Kinematics	17
4.1.1	Direct Kinematics	17
4.1.2	Inverse Kinematics	18
4.1.3	Kinematic Model of TEO	18
4.1.3.1	Kinematic model of legs	18
4.1.3.2	Kinematic model of torso	20
4.1.3.3	Kinematic model of arms	21
4.2	Differential Kinematics	23
4.2.1	The Jacobians	24
4.2.1.1	Geometric Jacobian	24
4.2.1.2	Analytical Jacobian	26
4.2.1.3	Kinematic Singularities	27
4.2.1.4	Kinematic Redundancy	28
4.3	Inverse Differential Kinematics	29
4.3.1	First-order algorithms	30
4.3.2	Second-order algorithm	31
5	Whole-body movement of humanoid robots	33
5.1	Trajectory planning in the operational space	33
5.2	Humanoid robots manipulation	34
5.2.1	Single manipulation	34

5.2.2	Bi-manual manipulation	36
5.3	Humanoid robots locomotion	39
5.3.1	Gait analysis	39
5.3.1.1	Phases of a step	40
5.3.1.2	Tasks of a step	41
5.3.2	Biped Locomotion Stability Criteria	42
5.3.2.1	Floor Projection of the Center of Gravity (FCOG)	43
5.3.2.2	Zero Moment Point (ZMP)	43
5.3.2.3	Fictitious Zero Moment Point (FZMP) and Foot-Rotation Indicator (FRI)	45
5.3.2.4	Center of Pressure (CoP)	47
5.3.2.5	Contact Wrench Sum	48
5.3.3	Humanoid Robot Gaits	51
5.3.3.1	Passive and active walking	51
5.3.3.1.1	Passive walking	51
5.3.3.1.2	Active walking	51
5.3.4	Gait generation models	53
5.3.4.1	Static gait	53
5.3.4.2	Dynamic gait	53
5.3.4.3	Mass Distributed Models	54
5.3.4.3.1	Pattern generator taking into account the contact wrench sum (CWS)	55
5.3.4.3.2	Two Masses Inverted Pendulum Mode (TMIPM)	57
5.3.4.3.3	Multiple Masses Inverted Pendulum Mode (MMIPM)	59
5.3.4.4	Mass Concentrated Models	60
5.3.4.4.1	3D-Linear Inverted Pendulum Model	61
5.3.4.4.2	Cart-table model	62
6	Whole-body movement in TEO robot	69
6.1	Manipulation task	69
6.1.1	Single manipulation	69

6.2	Locomotion task	69
6.2.1	Global Motion	69
6.2.2	Local Motion	71
6.2.3	Gait pattern generation	72
6.2.3.1	Feet pattern	75
6.2.3.1.1	Feet trajectories	78
6.2.3.2	FZMP pattern	79
6.2.3.2.1	FZMP trajectories	80
6.2.3.3	CoG pattern	80
6.2.3.3.1	CoG trajectories	84
6.2.3.4	Arms pattern	84
6.2.3.4.1	FZMP trajectories	86
6.2.4	Kinematics Transformation	86
7	Conclusions and Future Work	87
7.1	Conclusions	87
7.2	Future Work	87
Appendices		89
A	TEOTraGen	91
A.1	TEOWholeGen	92
A.2	TEOStepGen	98
B	TEO ROS packages	105
B.1	teo_description	105
B.2	teo_remote	109
B.3	teo_messages	109
B.4	teo_control	110
Bibliography		111

Index of Tables

4.1	Dimensions for the legs of TEO	19
4.2	Rotation angles for the leg joints of TEO	19
4.3	Denavit-Hartenberg parameters for a floating leg of TEO	20
4.4	Denavit-Hartenberg parameters for a standing leg of TEO	20
4.5	Dimensions for the torso of TEO	21
4.6	Rotation angles for the torso joints of TEO	21
4.7	Denavit-Hartenberg parameters for the torso of TEO	21
4.8	Dimensions for the arms of TEO	22
4.9	Rotation angles for the arm joints of TEO	22
4.10	Denavit-Hartenberg parameters for an arm of TEO	22

Index of Figures

2.1	TEO communication architecture	11
2.2	TEO humanoid robot	12
3.1	TEO operational points	16
4.1	Kinematic Model for a Floating Leg of TEO	23
4.2	Kinematic Model for a Standing Leg of TEO	24
4.3	Kinematic Model for the torso of TEO	25
4.4	Kinematic Model for an Arm of TEO	26
4.5	Mapping between the generalized coordinates space and the operational space	29
4.6	Second-order inverse kinematics algorithm with Jacobian inverse	32
5.1	Tasks and phases of a step	41
5.2	Phases of a step	42
5.3	Forces and Moments acting on a rigid foot with a flat sole and fully supported with the floor [4]	44
5.4	Fictitious ZMP (FZMP) [25]	46
5.5	Center of Pressure (COP) [4]	48
5.6	Contact Wrench Sum Model	49
5.7	A chain of four rigid, inanimate links will walk by themselves down a shallow incline	52

5.8	Static and dynamic gait	54
5.9	Mechanical model of a robot with two masses	58
5.10	Mechanical model of a robot with multiple masses	60
5.11	A pendulum under constraint	61
5.12	Cart-table model	63
6.1	Left hand operational point of TEO	70
6.2	Left hand operational point of TEO	70
6.3	Locomotion Task	71
6.4	Local axis gait algorithm	73
6.5	Walking pattern generation process	74
6.6	Swing foot movement	76
6.7	FZMP trajectory - XY plane	80
6.8	Waist trajectory - XY plane	84
A.1	TEOTraGen main window	91
A.2	TEOWholeGen settings window	92
A.3	Trajectory generation window	93
A.4	TEOWholeGen: Number of points option	93
A.5	TEOWholeGen: Final positions and interpolation panel	93
A.6	TEOWholeGen: Final orientations and interpolation panel	94
A.7	TEOWholeGen: Final time panel	94
A.8	TEOWholeGen: Trajectory generation panel	94
A.9	TEOWholeGen: Visualization of the operational space trajectory in the Plots window	95
A.10	TEOWholeGen: Visualization of the joint space trajectory in the Plots window	96
A.11	TEOWholeGen: Visualization of TEO tasks with the Robotics Tool- box for MATLAB®	97
A.12	TEOWholeGen: Visualization of TEO tasks in RViz	97
A.13	TEOStepGen main window	98
A.14	TEOStepGen footprints generation main window	99

A.15 TEOStepGen footprints generation panels	99
A.16 TEOStepGen: Visualization of the operational space trajectories .	100
A.17 TEOStepGen: Visualization of the operational space trajectories .	100
A.18 TEOStepGen: Visualization of locomotion task of TEO in RViz .	101
A.19 TEOStepGen settings window	102
A.20 TEOStepGen: Fast steps generation window	102
A.21 TEOStepGen: Fast steps generation window	103
B.1 teo_description ROS package: URDF structure of TEO	106
B.2 teo_description ROS package: Simple TEO URDF model visualized in RViz	107
B.3 teo_description ROS package: Complete TEO URDF model visualized in RViz	108

Acknowledgments

Thanks to all ...

Resumen

Esta tesis desarrolla ...

Abstract

This master thesis attempts to develop whole-body movement using inverse kinematics control, reason why a state of the art of kinematics and a description of locomotion terms, stability criteria and gait generation models are realized.

To facilitate the generation of whole-body trajectories, it has been developed a graphic user interface (GUI) that allows customize movement in the operational space and transform it to the joints space. The application also considers the locomotion (walk) of the humanoid allowing custom gait.

...

Introduction

In recent years, there has been a strong impetus to research on humanoid robots, developing robots with high workability, interaction and mobility. This impetus is motivated by the many advantages of humanoid robots against other kind of robots. Among these, the most important is that humanoid robots can operate directly on the same human environment without any modification. While the humanoid robots with wheels also can manipulate objects in this environment, only bipedal humanoid robots get a full adaptation to the environment due to its capacity of moving on complex terrains.

Although bipedal humanoid robots have a complex design, a slow speed, a complex control and a locomotion that difficult its integral development, the research that they have had in recent years is very relevant and promising.

Within the movement of humanoid robots, motion planning by trajectory generation in operational (task or cartesian) space is a very used method by the possibility to define the movement in the real environment where the robot will perform the task.

Humanoid robotics also offers a unique research tool for understanding the human brain and body. Using humanoids, researchers can embody their theories

and take them to task at a variety of levels. Many researchs on biomechanics, cognition and human behavioral studies found in humanoids some useful platforms for experimentation.

Nevertheless, there are many challenges to overcome. Humanoid Robotics has many unresolved challenges and the research is focused in the stability and mobility of the robot under different environmental conditions, complex control systems to coordinate the whole body motion, as well as the development of fast intelligent sensors and light energy-saving actuators. The research in humanoid robots also has many challenges with the autonomy, manipulation, locomotion and human-robot interaction.

On the other hand, the humanoid robot TEO, developed by the Carlos III of Madrid University, is an excellent robotic platform for research in this field. Its design is similar to humans, so that the movements generated can be inspired in humans allowing the develop of human-like tasks.

1.1 Humanoid Robotics

Humanoid Robotics is a branch of the robotics field that deals with the design, construction, operation, and application of humanoid robots. A *humanoid robot* or *humanoid*¹ is an electro-mechanical machine with anthropomorphic form guided by a computer program or electronic circuitry in order to emulate some subset of the physical, cognitive and social dimensions of the human body and experience.

The goal of the humanoid robotics is to create a robotic system to work with and for humans. Humanoids are designed to act safely alongside humans, extending our capabilities in a wide variety of tasks and environments.

¹The difference between a humanoid and an *android* is only skin-deep. An android has an internal mechanics of a humanoid robot but looks exactly like humans on the outside.

As the robotics, the humanoid robotics field is an interdisciplinary science that integrates the knowledge of many disciplines. However, a fully-fledged humanoid robot will incorporate work from each of the areas below [14]:

- **Perception:** It includes many sensing modalities as computer vision, taste, smell, sonar, IR, haptic feedback, tactile sensors, and range of motion sensors. It also includes implementation of unconscious physiological mechanisms such as the vestibulo-ocular reflex, which allows humans to track visual areas of interest while moving. Lastly, this area includes the attentional, sensor fusion and perceptual categorization mechanisms which roboticists implement to filter stimulation and coordinate sensing.
- **Human-robot interaction:** It includes the study of human factors related to the tasking and control of humanoid robots. Additionally, it considers the communications with humans, safety insurance, and gestures and facial expressions recognition. The area it also considers the ways in which humanoids can be profitably and safely integrated into everyday life.
- **Learning and adaptive behaviour:** It considers the supervised and unsupervised learning of new tasks by sequencing existing behaviours.
- **Legged locomotion:** It includes walking up stairs and steeping inclines and over rough, uneven terrain. It considers full-body balancing act that must occur faster than real-time.
- **Arm control and dexterous manipulation:** It focuses on the hardware and control of dexterous tasks. It wants to solve the problem of how to move from brittle, hard-coded dexterity toward adaptive control where graceful degradation can be realized.

1.1.1 Characteristics of the humanoid robots

The main differences between humanoids and other kinds of robots (manipulators, mobile robots, multi-legged robot) are:

- Bipedal human-like locomotion: That allows it to execute any type of trajectory.
- Anthropomorphic appearance: A humanoid robot has two legs, two arms, a torso and a head. This means that is a hyper degree of freedom system and that it has a complex kinematics and dynamics.
- Arms and legs coordination

1.2 Objectives

1.2.1 General objective

The objective of this master thesis is the develop of whole-body motion for a humanoid robot using inverse kinematic control.

1.2.2 Specific objectives

In order to achieve the general objective of Section 1.2.1, the following goals have to be achieved:

- Describe the state of the art of kinematics.
- Describe the models and criteria commonly used for the human locomotion task.
- Develop a method that allows...
- Develop a software that allows to define custom movement in the operational space and transform it to the joints space.
- Implement the above to the TEO humanoid robot.

1.3 Organization of the chapters

The Chapters of this document are organized as follows. Chapter 1 has given an introduction to the generalities of humanoid robotics. Additionally, the mo-

tivation and the objectives of this thesis have also been presented. Chapter 2 describes briefly TEO humanoid robot, humanoid platform used in this thesis. Chapter 3 describes the forms how TEO is represented. Chapter 4 introduces the state of the art in kinematics focusing on differential kinematics and how it is implemented in TEO. Chapter 5 tries to unify notions and terms used in humanoid robots manipulation and locomotion, it describes most common stability criteria and gait generation models. Chapter 6 details the gait generation method developed and the algorithms used in the different phases. Chapter 7 points out the conclusions of this master thesis and proposes some possible future work. Finally the appendices briefly presents *TEOTraGen* application and the ROS packages developed for TEO.

Chapter 2

TEO Humanoid Robot

2.1 TEO: Task Environment Operator

TEO (Task Environment Operator) is a biped humanoid robot developed by Carlos III of Madrid University. It is a robust and open humanoid platform for research on biped walking, balancing control, sensor fusion, human-robot interaction (HRI) for collaborative tasks, and other related issues.

The whole platform is conceived under the premise of high efficiency in terms of mechanical robustness and energy consumption [17]. The mechatronics of this platform is inspired by the human natural and adaptive locomotion, and its design is oriented to achieve human physical capacities and performances.

Unlike its predecessors, RH-0 and RH-1 platforms, who had a cantilever type hip, TEO has been designed with a planar type hip in which the three joints axes are matched in a single point, a more complex design but more robust for practical purposes. The new design of the structure, within which the electronics and mechanical components are housed, seeks greater protection to the most vulnerable elements of the humanoid.

The previous models had 21 degrees of freedom (without consider the head)

while TEO considers three additional degrees: one on each shoulder in the craniocaudal axis that allows the arms to perform human-like movements and another degree of freedom in the trunk to control the roll of the body forward and backward and facilitate stability of the equilibrium.

The decision subsystem consists of two CPU that control the mobility skills of the legs and arms respectively. The driver of each motor joints are lighter and smaller than its predecessors.

The information subsystem incorporates sensors to measure proprioceptive sensors for internal data as encoders, joint-force or torque sensors and inclinometers, and the acquisition of information from the environment through cameras, microphones, and other exteroceptive Sensors.

Finally, action subsystem is constituted by the engines presented in each joint.

2.2 Mechatronic systems of TEO humanoid

The selection, integration and operation of the mechatronic systems of TEO are inspired from the human physiology [3]. Next, a brief description of these systems are presented.

2.2.1 Sensorial system

The sensorial system is composed by two main perception types depending of the origin of the sensed stimulus. The exteroceptive perception is composed by the information about environmental circumstances or external stimuli. This perception is the result of the combination of the information coming from the vestibular and visual sensorial systems. Nevertheless, the proprioceptive system collects the information of the body state itself.

2.2.2 Exteroceptive perception

- **Inertial System:** The inertial system used in TEO robot is the Inertial Measurement Unit (IMU) MTiTM from the company XsensTM. This unit comprise a 3D accelerometer, a 3D gyroscope (gyro) and a 3D magnetometer (compass) as sensor devices.
- **Force/Torque sensor system:** The mechanical structure of TEO robot integrates two F/T sensors drom the company JR3, Inc. in the wrists and another two in the ankles.
- **Visual system:** The vision system selected for TEO robot is a Microsoft KinectTM. This low cost device, combined with an external image processing, provides TEO robot with the capacity to detect moving objects, optical flows, self-motion, etc. and apply the resulting information to improve postural control. This device is composed by two different vision sensor systems: an infrared laser projector combined with a monochrome CMOS sensor, which captures video data in 3D under any ambient light conditions, and a 2D colour 640 x 480 pixels VGA camera.

2.2.3 Proprioceptive perception

- **Joints:** The movement of each mechanical part of TEO is sensed by means of absolute and incremental (relative) encoders. The incremental encoders are RENCO RCML15 optical encoders and they are located at the joint inputs (motor/actuator) measuring motor position and velocity. These devices have 1024 line count resolution with A/B output format in phase quadrature. The absolute encoders are AVAGO 7500 AEAS single-turn absolute optical encoders are located at the joints outputs. These absolute encoders have 11 digital tracks plus 2 sine/cosine tracks to generate a precise 16 bit Gray code and the information is provided by means of a serial output at 16MHz.

2.2.4 Integration centres

TEO robot architecture divides posture control processing between the Central Process Unit (CPU) and one intelligent servo-drive to control each joint. Both systems are interconnected by means of a CANBus.

- **CPU's:** The mainboard integrated in TEO robot has an IntelTM Core 2 Duo E6400 (2 x 2.13GHz) processor. It is the 'brain' of the robot in which posture reactions and movement control is calculated.
- **ISCM8005 Intelligent Drives:** This kind of drivers has a built-in Digital Signal Processor (DSP) that enables control with the minimum CPU involvement. In this case, the driver can maintain the desired joint angle against external disturbances.

2.2.5 Action System

- **Motor and transmission lower limbs:** The bigger demand of torque of the leg joints forces the use of an elevated transmission rate design. For this reason, the lower extremities joints use a transmission composed by a Harmonic DriveTM and a pulley/belt train. The articular joints were designed considering an estimated TEO body weight about 65Kg and a step velocity of 0.75m/s.
- **Motor and transmission upper limbs:** The upper limbs use only Harmonic DriveTM with fixed reduction rate to increase the output torque.

2.2.6 Communication channels

The perception information in TEO robot has been divided depending of the importance of the perceptual information for postural control. High velocity CAN-Bus networks are used to transmit the main proprioceptive information from all joints. The exteroceptive perceptions (visual and inertial) are transmitted using USB 2.0 interfaces, meanwhile F/T information is transmitted by RS-485 interface. The communication between the two mainboards (the brain) of the robot is

implemented by means of a high-speed Ethernet connection. This communication architecture is shown in Figure 2.1.

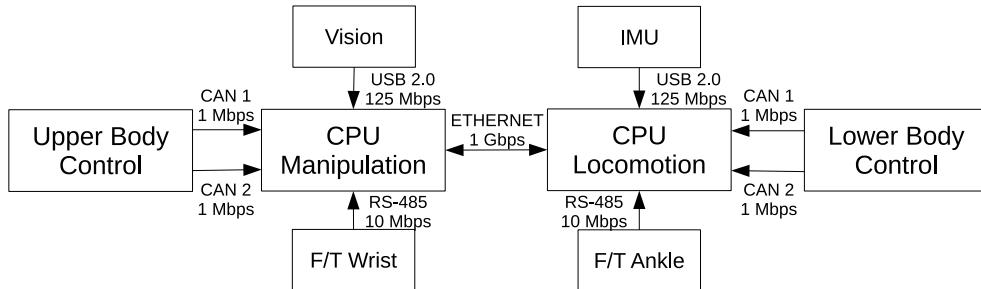


Figure 2.1: TEO communication architecture

- **CANBus:** The transmission data rate selected for TEO robot communication is 1Mbps (top end). With this, the transmission time of the complete frame (128 bits) is 128 microseconds. Each lower limb has six devices attached to one CANBus communication channel and the upper limbs have seven CAN devices attached to two buses. Taken this into account, the time to communicate 8 bytes to one whole lower limb in the worst case would be about 768 microseconds. This time doesn't consider possible communication delays neither the responses from the CAN devices to the CPU controller.
- **Ethernet:** This kind of communication channel is used to inter-communicate the CPU controller of upper body with the controller of the lower body. The connexion has a transmission rate of 1Gbps.
- **USB serial bus:** Secondary devices with information such as visual perception or inertial measurements communicate with postural controllers by means of USB (Universal Serial Bus) with a maximum transmission data rate of 125Mbps.

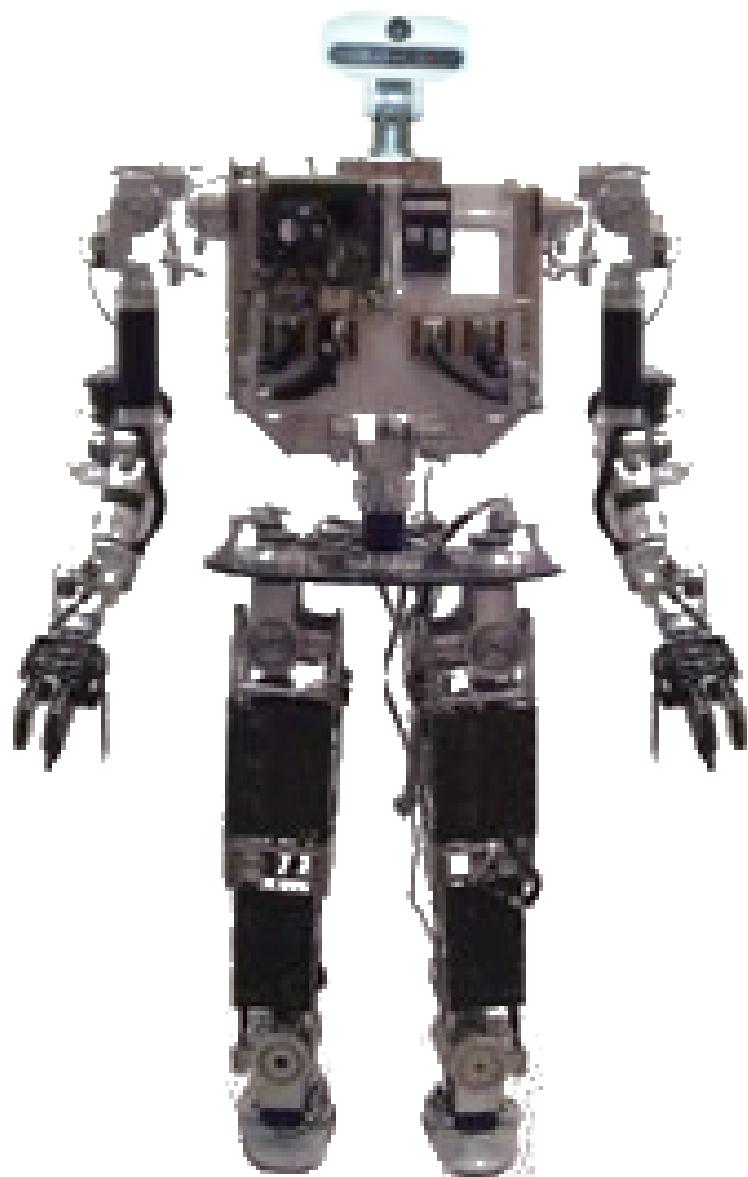


Figure 2.2: TEO humanoid robot

Chapter 3

Humanoid robots representation

Planning and control of the movement of humanoids robots need to define the current and the desired configuration of the different parts of the robot. This configuration can be represented using the Generalized coordinates space or the operational space. In this chapter, the configuration of TEO that are used in this thesis is defined in both spaces.

3.1 Limbs of a humanoid robot: manipulators

The arms and legs of humanoids are robot manipulators. A manipulator consists of a series of rigid bodies (*links*) connected by means of kinematic pairs or *joints* [23]. Actually, it exists different joint types but the most common are *revolute* and *prismatic*. The whole structure forms a *kinematic chain*. From a topological viewpoint, a kinematic chain can be *open* when there is only a sequence of links connecting the two ends of the chain or *closed* when the sequence of links forms a loop.

3.2 Generalized coordinates space

A robotic manipulators with n joints is fully represented with the *joint angle vector* $q = [q_1 \dots q_n]^T \in \mathbb{R}$, where q_i represents the i -th joint. The space of all

such joint vectors is called the *Joint Space* [23] and will be represented by Q . The dimension of Q represents the *number of degrees of freedom (DOF)* of the robot: $DOF = \dim\{Q\} = n$. However, unlike robotic arms that possess a fixed base, humanoid robots are mobile robots therefore the position and orientation of a certain point in the robot's structure (usually referred as *base*, *root* or *free-floating*) with respect to the fixed inertial frame or world frame needs to be specified. This base is represented by $x_b = [p_b \ \varphi_b]^T \in \mathbb{R}^{m_b}$ (with $m_b = 3 + r$), where $p_b \in \mathbb{R}^3$ is the position in Cartesian, cylindrical or spherical coordinates, and $\varphi_b \in \mathbb{R}^r$ is the orientation which can be given using any type of Euler Angles ($r = 3$), Quaternions/Euler Parameters ($r = 4$), or Cosine Directors ($r = 9$). The full representation of the robot is, then, given by the generalized joint vector

$$q_g = \begin{bmatrix} x_b \\ q \end{bmatrix} \in \mathbb{R}^{m_b+n} \quad (3.2.1)$$

The space containing every q_g constitutes the *Generalized Coordinates Space*, and will be represented by Q_g .

In this thesis, the base point of TEO is located in its waist. The position of this point is represented by Cartesian coordinates, $p_b = [p_{bx} \ p_{by} \ p_{bz}]^T$, and its orientation by Roll, Pitch and Yaw angles, $\varphi_b = [\varphi_{bx} \ \varphi_{by} \ \varphi_{bz}]^T$. Consequently, $x_b \in \mathbb{R}^6$ and $q_b \in \mathbb{R}^{6+n}$.

As mentioned in Section 2.1, TEO has 26 degrees of freedom, then $n = 26$. Considering this, the joint angle vector component of the TEO's configuration, q , in the generalized coordinates space $q_g \in \mathbb{R}^{32}$, is:

$$q = \begin{bmatrix} q_{rl} \\ q_{ll} \\ q_t \\ q_{ra} \\ q_{la} \end{bmatrix} \quad (3.2.2)$$

where $q_{rl} \in \mathbb{R}^6$ is the joint vector of right leg, $q_{ll} \in \mathbb{R}^6$ is the joint vector of left leg, $q_t \in \mathbb{R}^2$ is the joint vector of torso, $q_{ra} \in \mathbb{R}^6$ is the joint vector of right arm, and $q_{la} \in \mathbb{R}^6$ is the joint vector of left arm. The components of these vectors are detailed in Section 4.1.3.

3.3 Operational Space

The position and orientation of the end-effector, in the case of a robotic manipulator, is represented by the *operational* (or *task*) vector $x_e = [p_e \varphi_e]^T \in \mathbb{R}^m$, where p_e and φ_e are defined as above. The space of all such operational vectors is called the *Operational Space* [23], *Task Space* or *Cartesian Space*[2].

For humanoid robots, the “end-effectors” concept is generalized to cope other points. The term *Operational Point*, $x = [p \varphi]^T \in \mathbb{R}^m$, is any part of the body whose position and/or orientation are intended to be controlled.

The abstraction of TEO used in this work considers six operational points: right hand(x_{rh}), left hand(x_{lh}), chest(x_{ch}), waist(x_w), right foot(x_{rf}) and left foot(x_{lf}).¹ This points are shown in Figure 3.1.

¹More specifically, right and left hands can be considered as right and left wrist respectively. On the other hand, right and left can be considered as right and left ankle.

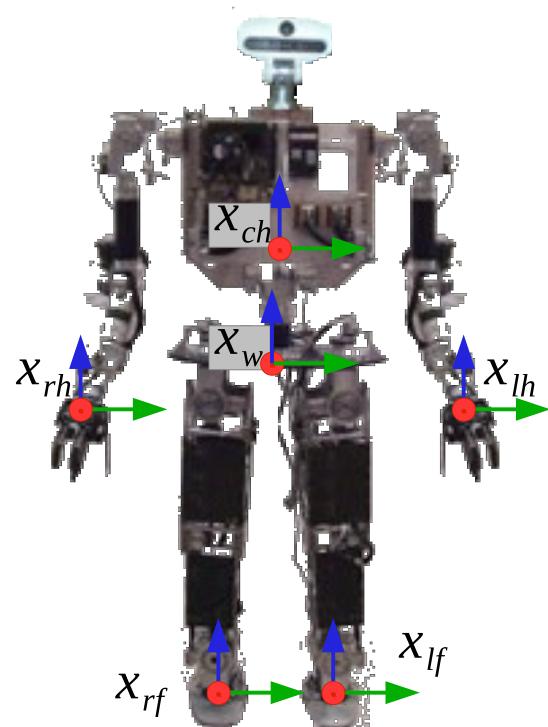


Figure 3.1: TEO operational points

Chapter 4

Humanoid robots kinematics

The control of movement of a humanoid can be faced by two different approaches, with the kinematic model or with the dynamic model of the robot. The kinematic approach, is based on the relation between the position and velocity of the joints and the operational points. On the other hand, the dynamic approach extends the analysis to the accelerations, forces and torques that generate the motion.

This chapter presents the state of the art of the kinematic-based control.

4.1 Kinematics

Kinematics is the branch of mechanics that studies the motion of a body or a system of bodies without consideration given to its mass or the forces acting on it [2].

4.1.1 Direct Kinematics

The mechanical structure of a manipulator is characterized by its DOFs which uniquely determine its *posture*. The aim of *direct kinematics* or forward kinematics is to compute the pose (position and orientation) of the end-effector x as a

function of the joints variables q , stating that $\forall q \in Q, \exists x \in \mathbb{R}^m$ such that:

$$x = f(q) \quad (4.1.1)$$

4.1.2 Inverse Kinematics

The aim of inverse kinematics is to determine the set of joint angles q that achieves a given end-effector pose x :

$$q = g(x) \quad (4.1.2)$$

when $g(x)$ is defined. Classical methods for inverse kinematics include algebraic approaches, geometric approaches and decoupling.

4.1.3 Kinematic Model of TEO

The kinematic model proposed in this section is based on Denavit-Hartenberg (DH) convention [20] and it is used for the trajectory generation of TEO robot.

4.1.3.1 Kinematic model of legs

Due to the human-like inspiration of TEO, the kinematic of the legs seeks to correspond with human legs. For this reason, the kinematic model of the legs considers five links whose dimensions are described in Table 4.1.

Additionally, it considers six joints for reply the degrees of freedom of a human leg: three joints for the hip, one for the knee and one for the ankle. The maximum and minimum rotation of each joint is described in Table 4.2

The kinematic model of the leg considers two different Denavit-Hartenberg conventions. The first one imitate a floating leg, in other words it assumes that the leg is a robot manipulator with the base in the center of the hip and the foot at the end, as can be seen in Table 4.3 and in Figure 4.1. The second convention

Name	Symbol	Length (m)
<i>Waist</i>	Leg_Link1	0.146
<i>Femur</i>	Leg_Link2	0.330
<i>Tibia</i>	Leg_Link3	0.300
<i>Patella</i>	Leg_Link4	0.033
<i>Astragalus</i>	Leg_Link5	0.124

Table 4.1: Dimensions for the legs of TEO

Name	Joint type	number	Min. angle (rad)	Max. angle(rad)
<i>Hip</i>	Yaw	1	-0.524	0.419
	Roll	2	-2.094	0.784
	Pitch	3	-0.698	0.524
<i>Knee</i>	Pitch	4	0.000	0.571
<i>Ankle</i>	Pitch	5	-1.484	1.484
	Roll	6	-0.349	0.349

Table 4.2: Rotation angles for the leg joints of TEO

Name	Joint type	number	a_i	α_i	d_i	θ_i
<i>Hip</i>	Yaw	1	0	$\pi/2$	0	$\theta_1 + \pi/2$
	Roll	2	0	$-\pi/2$	0	$\theta_2 - \pi/2$
	Pitch	3	Leg_Link2	0	0	θ_3
<i>Knee</i>	Pitch	4	Leg_Link3	0	Leg_Link4	θ_4
<i>Ankle</i>	Pitch	5	0	$\pi/2$	0	θ_5
	Roll	6	Leg_Link5	0	0	θ_6

Table 4.3: Denavit-Hartenberg parameters for a floating leg of TEO

Name	Joint type	number	a_i	α_i	d_i	θ_i
<i>Ankle</i>	Roll	1	0	$\pi/2$	0	θ_1
	Pitch	2	Leg_Link3	0	Leg_Link4	θ_2
<i>Knee</i>	Pitch	3	Leg_Link2	0	0	θ_3
<i>Hip</i>	Pitch	4	0	$-\pi/2$	0	θ_4
	Roll	5	0	$-\pi/2$	0	$\theta_5 + \pi/2$
	Yaw	6	Leg_Link1	0	0	θ_6

Table 4.4: Denavit-Hartenberg parameters for a standing leg of TEO

assumes that the leg is a robot manipulator with the base in the foot and the center of the waist at the end, as can be seen in Table 4.4 and in Figure 4.2.

4.1.3.2 Kinematic model of torso

The kinematic model of the torso considers four possible links [20] which for the case of TEO only two of them are considered and whose dimensions are described in Table 4.5. The torso of TEO considers two joints for reply the degrees of freedom of a human waist. The maximum and minimum rotation of each joint is described in Table 4.6.

The kinematic model of the torso it also assumes that the torso is a robot manipulator with the base in the center of the hip and the IMU position at the end, as

Name	Symbol	Length (m)
Waist	Waist_Link1	0.000
	Waist_Link2	0.287
Torso	Torso_Link1	0.000
	Torso_Link2	0.060

Table 4.5: Dimensions for the torso of TEO

Name	Joint type	number	Min. angle (rad)	Max. angle(rad)
Torso	Yaw	1	-0.784	0.784
	Roll	2	-0.349	1.571

Table 4.6: Rotation angles for the torso joints of TEO

can be seen in Table 4.7 and in Figure 4.3.

4.1.3.3 Kinematic model of arms

The kinematic model of the arms considers six links whose dimensions are described in Table 4.8. These links are connected for six joints for reply the degrees of freedom of a human arm. The maximum and minimum rotation of each joint is described in Table 4.9.

The kinematic model of the arms assumes that an arm is a robot manipulator with the base in the IMU position and the hand at the end, as can be seen in Table 4.10 and in Figure 4.4.

Name	Joint type	number	a_i	α_i	d_i	θ_i
Torso	Yaw	1	0	$-\pi/2$	0	θ_1
	Pitch	2	-Torso_Link1	0	0	θ_2

Table 4.7: Denavit-Hartenberg parameters for the torso of TEO

Name	Symbol	Length (m)
<i>Chest</i>	Chest_Link1	0.000
	Chest_Link2	0.245
<i>Clavicle</i>	Arm_Link1	0.340
<i>Humerus</i>	Arm_Link2	0.337
<i>Ulna</i>	Arm_Link3	0.210

Table 4.8: Dimensions for the arms of TEO

Name	Joint type	number	Min. angle (rad)	Max. angle(rad)
<i>Shoulder</i>	Pitch	1	-3.141	3.141
	Roll	2	-0.785	2.094
	Yaw	3	-1.047	1.047
<i>Elbow</i>	Pitch	4	-1.745	1.745
<i>Wrist</i>	Yaw	5	-1.833	1.833
	Pitch	6	-0.698	0.960

Table 4.9: Rotation angles for the arm joints of TEO

Name	Joint type	number	a_i	α_i	d_i	θ_i
<i>Shoulder</i>	Pitch	1	0	$\pi/2$	0	$\theta_1 + \pi/2$
	Roll	2	0	$\pi/2$	0	$\theta_2 - \pi/2$
	Yaw	3	0	$-\pi/2$	-Arm_Link2	$\theta_3 + \pi/2$
<i>Elbow</i>	Pitch	4	0	$\pi/2$	0	θ_4
<i>Wrist</i>	Yaw	5	0	$-\pi/2$	-Arm_Link3	θ_5
	Pitch	6	0	$\pi/2$	0	θ_6

Table 4.10: Denavit-Hartenberg parameters for an arm of TEO

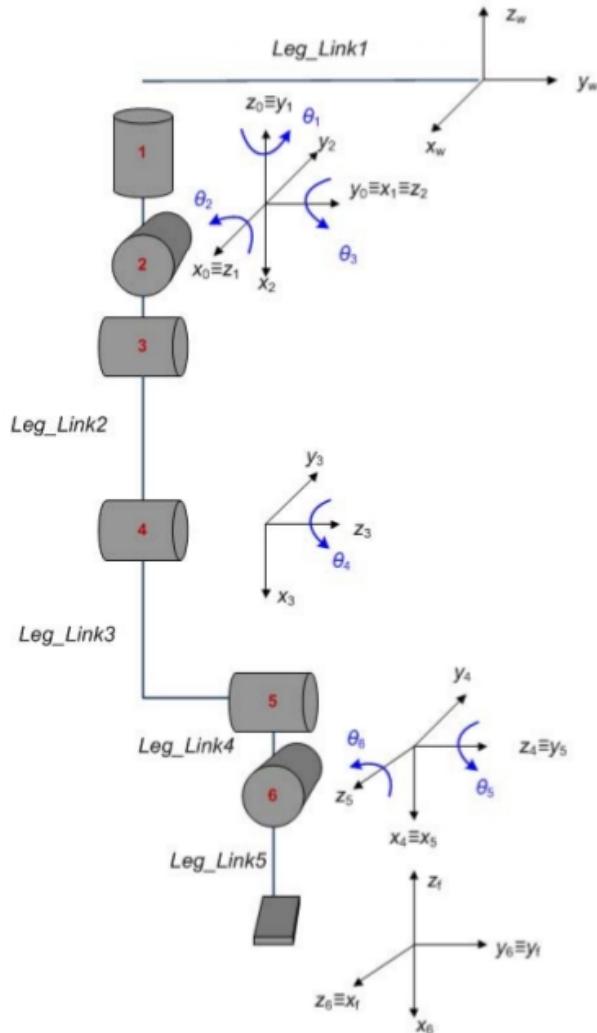


Figure 4.1: Kinematic Model for a Floating Leg of TEO

4.2 Differential Kinematics

The *differential kinematics* gives the relationship between the joint velocities and the corresponding end-effector linear and angular velocity. This mapping is described by the *Geometric Jacobian*, which depends on the manipulator configuration. Alternatively, if the end-effector pose is expressed with reference to a

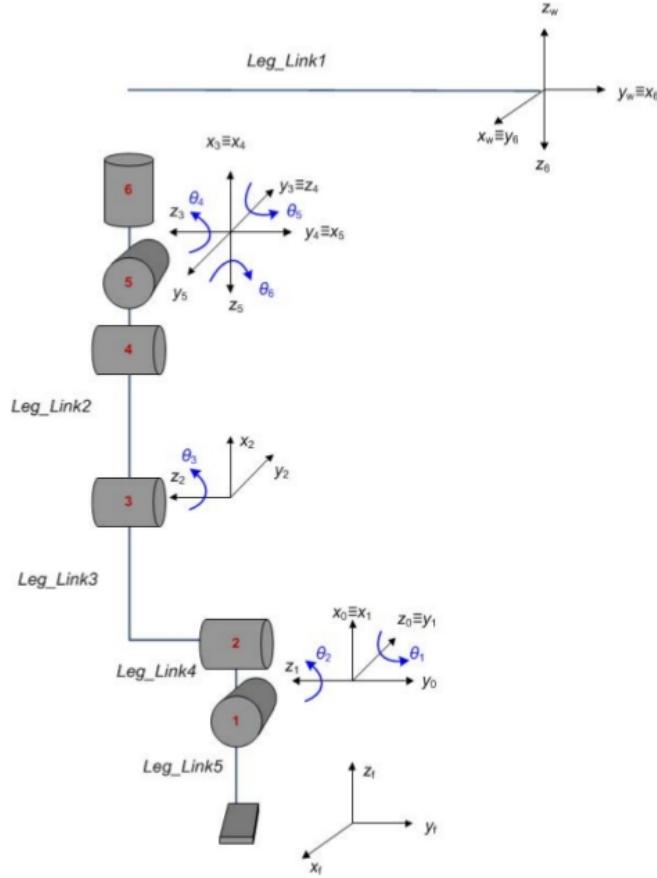


Figure 4.2: Kinematic Model for a Standing Leg of TEO

minimal representation in the operational space, as in section 3.3, the mapping is performed through the *Analytical Jacobian* matrix.

4.2.1 The Jacobians

4.2.1.1 Geometric Jacobian

The velocity of the operational point can be expressed in terms of the linear ($\dot{p} \in \mathbb{R}^3$) and the angular ($\omega \in \mathbb{R}^3$) velocities, as $v = [\dot{p} \ \omega]^T \in \mathbb{R}^6$. The *geometric*

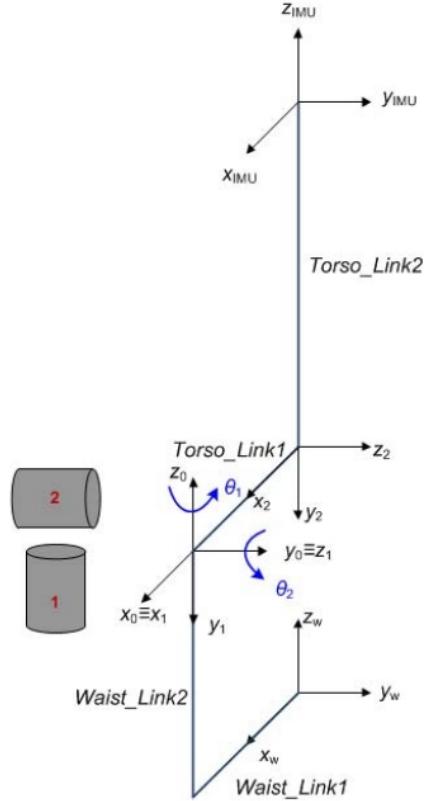


Figure 4.3: Kinematic Model for the torso of TEO

Jacobian J_0 , also called *basic Jacobian*¹, allows this mapping through

$$v = J_0(q_g)\dot{q}_g \quad (4.2.1)$$

which represents the *differential kinematics equation*. The $(6 \times n)$ geometric Jacobian matrix J_0 , is usually expressed as the combination of two $(3 \times n)$ matrices: J_P and J_O , as in (4.2.2). J_P allows a mapping to the end-effector linear velocity \dot{p} , as in (4.2.3), and J_O a mapping to the end-effector angular velocity ω , as in

¹Some authors use the symbol J to represent the geometric Jacobian, but in this thesis the symbol J_0 is preferred

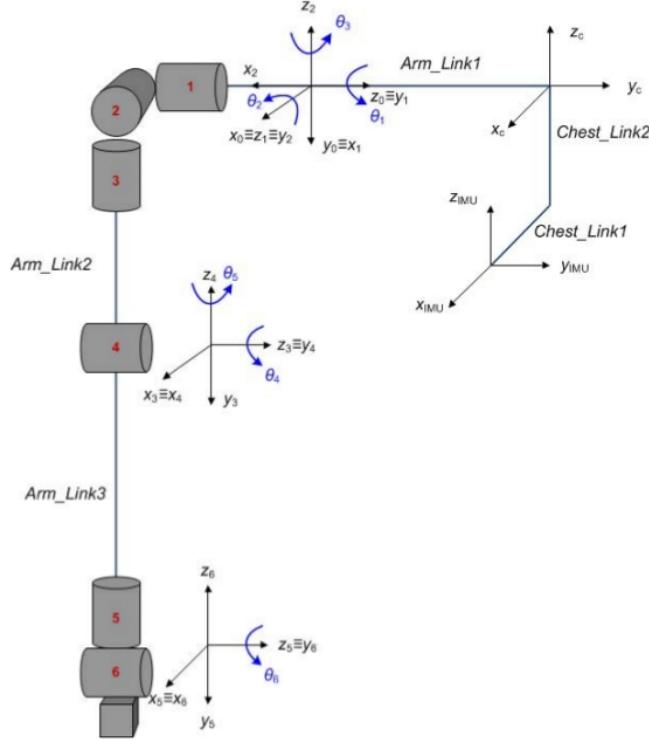


Figure 4.4: Kinematic Model for an Arm of TEO

(4.2.4).

$$J_0 = \begin{bmatrix} J_P \\ J_O \end{bmatrix} \quad (4.2.2)$$

$$\dot{p} = J_P(q_g)\dot{q}_g \quad (4.2.3)$$

$$\omega = J_O(q_g)\dot{q}_g \quad (4.2.4)$$

4.2.1.2 Analytical Jacobian

Due to the operational points of Section 3.3 are specified in terms of parameters in the operational space, it is necessary a relationship between the joint velocities

to the derivative of the operational points. The *analytical Jacobian* J , also called *task Jacobian*,² allows this mapping through

$$\dot{x} = J(q_g)\dot{q}_g \quad (4.2.5)$$

where J is the analytical Jacobian:

$$J(q_g) = \frac{\partial k(q_g)}{\partial q_g} \quad (4.2.6)$$

Just as the geometric jacobian, the analytical jacobian matrix J , is usually expressed as the combination of two $(3 \times n)$ matrices: J_P and J_ϕ , as in (4.2.7). J_P allows a mapping to the derivative of the position \dot{p} , as in (4.2.8), and J_O a mapping to the derivative of the orientation $\dot{\varphi}$, as in (4.2.9).

$$J = \begin{bmatrix} J_P \\ J_\phi \end{bmatrix} \quad (4.2.7)$$

$$\dot{p} = J_P(q_g)\dot{q}_g \quad (4.2.8)$$

$$\dot{\varphi} = J_\phi(q_g)\dot{q}_g \quad (4.2.9)$$

4.2.1.3 Kinematic Singularities

Both, geometric and analytical jacobians, are a function of the configuration q_g ; those configurations at which J_0 or J are rank-deficient are termed *kinematic singularities*.

Singularities can be classified into [23]:

- *Boundary* singularities that occur when the manipulator is either outstretched or retracted.

²Some authors use the symbol J to represent the analytical Jacobian, but in this thesis the symbol J is preferred

- *Internal* singularities that occur inside the reachable workspace and are normally caused by the alignment of two or more axes of motion, or else by the attainment of particular end-effector configurations.

4.2.1.4 Kinematic Redundancy

A *kinematic redundancy* occurs when the number of DOFs n of the structure is greater than the minimum number m required to execute a task. The kinematic redundancy is related to the task to be performed, being redundant for certain tasks and non-redundant for other tasks. For example, considering an operational point x_i with m_i DOFs of a task i , the system is *kinematically redundant* with respect to the task if $m_i < n$. The difference $n - m_i$ is the number of *redundant DOFs* or *degree of redundancy* with respect to task i .

The differential kinematics equation in (4.2.5) can be characterized in terms of the range and null spaces of the Jacobian [23] [21]:

- The *range space* or *column space* of J is the subspace $R(J) = \{\dot{x} \in \mathbb{R}^m | J(q_g)\dot{q}_g = \dot{x}\}$ and represents the operational point velocities \dot{x} that can be generated by the joint velocities \dot{q}_g at the configuration q_g . The dimension of the range space is called the rank: $\rho(J) = \dim\{C(J)\}$.
- The *null space* of J is defined as $N(J) = \{\dot{x} \in \mathbb{R}^{m_b+n} | J(q_g)\dot{q}_g = 0\}$ and represents the joint velocities \dot{q}_g that do not produce any velocity in the operational point, \dot{x} , at the configuration q_g . The dimension of the null space is called the nullity: $\nu(J) = \dim\{N(J)\}$.

For kinematically redundant robots, the existence of a subspace $N(J)$ allows the determination of techniques to handle the redundant DOFs and, thus, the solution to the differential kinematics Equation (4.2.5) is given by:

$$\dot{q}_g = J^\# \dot{x} + P \dot{q}_0 \quad (4.2.10)$$

where $J^\#$ is a generalized inverse of J , $P = (I - J^\# J)$ is the projector onto $N(J)$ and $\dot{q}_0 \in \mathbb{R}^{m_b+n}$ is an arbitrary vector. In fact, $P \dot{q}_0 \in N(J)$ and \dot{q}_0 only generate

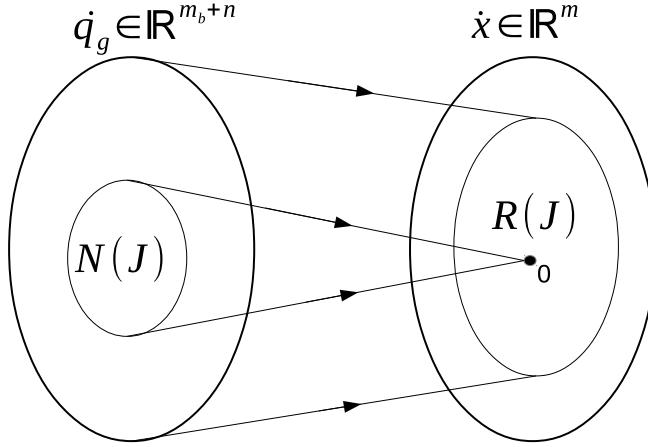


Figure 4.5: Mapping between the generalized coordinates space and the operational space

internal motion that has no effect in the operational point, \dot{x} , from the definition of the null space, but gives freedom to satisfy additional restrictions like more dexterous postures for a certain task.

4.3 Inverse Differential Kinematics

Considering that a motion trajectory is assigned to the operational points in terms of \dot{x} and a initial pose. The feasible generalized coordinates space trajectory $(q_g(t), \dot{q}_g(t))$ that reproduces the given trajectory can be obtained via simple inversion of the Jacobian matrix as in (4.3.1) if J is square.

$$\dot{q}_g = J^{-1}(q_g)\dot{x} \quad (4.3.1)$$

In the case of kinematic redundancy, the generalized right pseudo-inverse, $J^\#(q_g) = J^T(q_g)(J(q_g)J^\#(q_g))^{-1}$, is used:

$$\dot{q}_g = J^\#(q_g)\dot{x} + (I_n - J^\#(q_g)J(q_g))\dot{q}_0 \quad (4.3.2)$$

The positions in the generalized coordinates space can be calculated with numerical techniques in discrete time. Considering the Euler integration method,

given an integration interval Δt , if $q_g(t)$ and $\dot{q}_g(t)$ at time t_k are known, the positions in the generalized coordinates space at time $t_{k+1} = t_k + \Delta t$ can be computed as:

$$q_g(t_{k+1}) = q_g(t_k) + \dot{q}_g(t_k)\Delta t \quad (4.3.3)$$

4.3.1 First-order algorithms

reconstruction of joint variables q_g is entrusted to a numerical integration which involves drift phenomena of the solution; as a consequence, the end-effector pose corresponding to the computed joint variables differs from the desired one [23]. For this reason, it is important to define the *operational space error* between the desired (x_d) and the actual “end-effector” (x) position and orientation is:

$$e = x_d - x \quad (4.3.4)$$

Considering the time derivative of (4.3.4) and the differential kinematics (4.2.5)

$$\dot{e} = \dot{x}_d - \dot{x} = \dot{x}_d - J(q)\dot{q}_g \quad (4.3.5)$$

For this equation to lead to an inverse kinematics algorithm, it is worth relating the computed \dot{q}_g to the error e so that (4.3.5) gives a differential equation describing error evolution over time. Considering this error, the algorithmic version of (4.3.1) is:

$$\dot{q}_g = J^{-1}(q_g)(\dot{x}_d + Ke) \quad (4.3.6)$$

where, K is a positive definite (usually diagonal) matrix.

In the case of kinematic redundancy, the algorithmic version of 4.3.2 is:

$$\dot{q}_g = J^\#(q_g)(\dot{x}_d + Ke) + (I_n - J^\#(q_g)J(q_g))\dot{q}_0 \quad (4.3.7)$$

4.3.2 Second-order algorithm

A humanoid is inherently a second-order mechanical system. For this reason and for control purposes, it is convenient to specify trajectories of the joint and operational points motions at acceleration level.

The time differential of the differential kinematics equation (4.2.5) leads to:

$$\ddot{x} = J(q_g)\ddot{q}_g + \dot{J}(q_g, \dot{q}_g)\dot{q}_g \quad (4.3.8)$$

which gives the relationship between the joint space accelerations and the operational space accelerations.

If J is square and non-singular, the second-order differential kinematics (4.3.8) can be inverted in terms of the joint accelerations:

$$\ddot{q} = J^{-1}(q_g)((\ddot{x} - \dot{J}(q_g, \dot{q}_g)\dot{q}) \quad (4.3.9)$$

Considering the derivative of the error defined in (4.3.5) and the value of \ddot{x} in (4.3.8), the acceleration error is:

$$\ddot{e} = \ddot{x}_d - \ddot{x} = \ddot{x}_d - J(q_g)\ddot{q}_g - \dot{J}(q_g, \dot{q}_g)\dot{q}_g \quad (4.3.10)$$

Finally, the acceleration vector in the generalized coordinate space is:

$$\ddot{q}_g = J^{-1}(q)(\ddot{x}_d + K_D \dot{e} + K_P e - \dot{J}(q_g, \dot{q}_g)\dot{q}_g) \quad (4.3.11)$$

where K_D and K_P are positive definite matrices.

The second-order inverse kinematics algorithm is illustrated in the block scheme of Figure 4.6

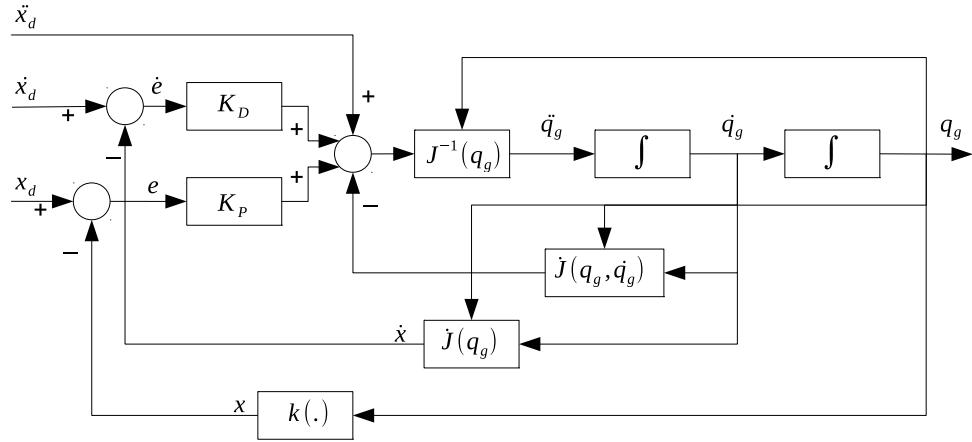


Figure 4.6: Second-order inverse kinematics algorithm with Jacobian inverse

In the case of kinematic redundancy, the generalized Jacobian pseudo-inverse $J^\#$ has to be used in (4.3.12):

$$\ddot{q}_g = J^\#(q_g)(\ddot{x}_d + K_D \dot{e} + K_P e - \dot{J}(q_g, \dot{q}_g) \dot{q}_g) + (I_n - J^\# J) \ddot{q}_0 \quad (4.3.12)$$

where the vector \ddot{q}_0 represents an arbitrary acceleration in the generalized coordinates space.

Chapter **5**

Whole-body movement of humanoid robots

5.1 Trajectory planning in the operational space

Before carry on, it is important to distinguish the notions of path and trajectory. *Path* denotes the locus of points in the joint space, in the generalized coordinates space or in the operational space, which the robot has to follow in the execution of the assigned motion; a path is then a pure geometric description of motion. On the other hand, a *trajectory* is a path on which a timing law is specified, for instance in terms of velocities and/or accelerations at each point [23].

The goal of trajectory planning in the operational space is to generate a time sequence of variables that describes end-effector position and orientation over time in respect of the imposed constraints. An important characteristic of this trajectory is that is smooth, i.e. position and orientation vary smoothly with time.

Trajectory planning in the operational space naturally allows the presence of path constraints to be accounted, as presence of obstacles or just a region where the operational point must to be kept. In this thesis, the whole-body movement

of TEO are generated through trajectories of the six operational points of Section 3.3 in order to use this feature.

5.2 Humanoid robots manipulation

5.2.1 Single manipulation

Grasping and manipulating is a key feature in humanoid robotics because allows humanoids to interact with its environment. An isolated hand movement in humanoid robots can be considered as the movement of a manipulator with base in the chest and a hand as end-effector. Considering the analytical Jacobian of (4.2.5) the mapping between velocity in the joint space and velocity in the operational space of one hand is:

$$\dot{x}_{hand} = J(q_{arm})\dot{q}_{arm} \quad (5.2.1)$$

where x_{hand} is the pose of the hand in the operational space and q_{arm} is the kinematic chain of its respective arm.

This relation assumes that the legs and torso are fixed and do not move during the execution of the task. On the other hand, if these parts contributes to the final movement of the hand, this equation is not useful. For example, the whole-body analytical Jacobian that allows the mapping between generalized coordinates space and operational space of the right hand is:

$$J_{rh} = \begin{bmatrix} J_{rh}^{(\partial x_b)} & J_{rh}^{(\partial RL)} & J_{rh}^{(\partial LL)} & J_{rh}^{(\partial T)} & J_{rh}^{(\partial RA)} & 0 \end{bmatrix} \quad (5.2.2)$$

where $J_{rh}^{(\partial x_b)}$ is the analytical Jacobian of the with respect to the base motion, $J_{rh}^{(\partial RL)}$ is the analytical Jacobian of the with respect to the right leg joints, $J_{rh}^{(\partial LL)}$ is the analytical Jacobian of the right hand with respect to the left leg joints, $J_{rh}^{(\partial T)}$ is the analytical Jacobian of the right hand with respect to the torso joints, $J_{rh}^{(\partial RA)}$ is the analytical Jacobian of the right hand with respect to the right arm joints. Finally, 0 is a $(m_b \times n_{LA})$ null matrix, where m_b is the number of parameters of

the operational point and n_{LA} is the number of joints of the left arm.

For the left hand case, the whole-body analytical Jacobian that allows the mapping between generalized coordinates space and operational space is:

$$J_{lh} = \begin{bmatrix} J_{lh}^{(\partial x_b)} & J_{lh}^{(\partial RL)} & J_{lh}^{(\partial LL)} & J_{lh}^{(\partial T)} & 0 & J_{lh}^{(\partial LA)} \end{bmatrix} \quad (5.2.3)$$

where $J_{lh}^{(\partial x_b)}$ is the analytical Jacobian of the left hand with respect to the base motion, $J_{lh}^{(\partial RL)}$ is the analytical Jacobian of the left hand with respect to the right leg joints, $J_{lh}^{(\partial LL)}$ is the analytical Jacobian of the left hand with respect to the left leg joints, $J_{lh}^{(\partial T)}$ is the analytical Jacobian of the left hand with respect to the torso joints, $J_{lh}^{(\partial LA)}$ is the analytical Jacobian of the left hand with respect to the left arm joints. Finally, 0 is a $(m_b \times n_{LA})$ null matrix, where m_b is the number of parameters of the operational point and n_{LA} is the number of joints of the right arm.

Considering this whole-body analytical Jacobians, the inverse differential kinematics of the right hand using the second-order algorithm of section 4.3.2 is:

$$\ddot{q} = J_{rh}^\#(q)(\ddot{x}_d + K_D \dot{e} + K_P e - \dot{J}_{rh}(q, \dot{q})\dot{q}) + (I_n - J_{rh}^\# J_{rh})\ddot{q}_0 \quad (5.2.4)$$

where the vector \ddot{q}_0 represents an arbitrary acceleration in the generalized coordinates space, $J_{rh}^\#(q)$ is the generalized Jacobian pseudo-inverse of the right hand and I_n is a identity matrix of size $n = DOFs$ of the humanoid.

For the case of the left hand, there are the same relation than (5.2.4) but using the J_{lh} matrix.

In order to reduce computational complexity, it is possible to add two virtual joints representing the movement of the COG in the horizontal and sagittal plane. This model has been defined as the Virtual COG Joints approach¹ in

¹In [19] the name Virtual CoM Joints is used.

[19].

These two virtual prismatic joints resume the kinematics of the robot legs and torso. If the position of this point is placed in the COG of the robot, the solution of the inverse kinematics becomes a reference for the mobile legged part. With this, for manipulation tasks, the analytical Jacobian of the right hand is:

$$J_{rh} = \begin{bmatrix} J_{rh}^{(\partial C)} & J_{rh}^{(\partial RA)} & 0 \end{bmatrix} \quad (5.2.5)$$

where $J_{rh}^{(\partial C)}$ is the analytical Jacobian of the right hand with respect to the Virtual COG Joints. $J_{rh}^{(\partial RA)}$ and 0 are the same than (5.2.2)

For the left hand case, the whole-body analytical Jacobian that allows the mapping between generalized coordinates space and operational space is:

$$J_{lh} = \begin{bmatrix} J_{lh}^{(\partial C)} & 0 & J_{lh}^{(\partial LA)} \end{bmatrix} \quad (5.2.6)$$

where $J_{lh}^{(\partial C)}$ is the analytical Jacobian of the left hand with respect to the Virtual COG Joints. $J_{lh}^{(\partial LA)}$ and 0 are the same than (5.2.3)

5.2.2 Bi-manual manipulation

One of the biggest advantages of humanoids robots in manipulation tasks is the ability to use both hands to perform any task. However, the movement of the right and left hands, either differently or in a coordinated manner, requires some additional considerations.

Considering the operational points x_{rh} and x_{lh} that have to execute a coordinated movement, the tasks to be implemented with right and left hands are the following ones [19]:

- Task 1 : x_{rh} and x_{lh} should coincide in position and orientation.

- Task 2 : x_{rh} and x_{lh} must follow a desired trajectory.

The first task can be written as:

$$\dot{x}_{rh} = \dot{x}_{lh} \implies J_{rh}\dot{q} = J_{lh}\dot{q} \quad (5.2.7)$$

where J_{rh} and J_{lh} are the analytical Jacobian matrices of the right and left hands with respect to the whole-body joints, respectively, and q are the configuration of the humanoid in the generalized coordinates space.

Considering Eqs. (5.2.5) and (5.2.6), it is possible to rewrite (5.2.7) as:

$$\begin{aligned} \dot{x}_{rh} &= \begin{bmatrix} J_{rh}^{(\partial C)} & J_{rh}^{(\partial RA)} & 0 \end{bmatrix} \dot{q} \\ \dot{x}_{lh} &= \begin{bmatrix} J_{lh}^{(\partial C)} & 0 & J_{lh}^{(\partial LA)} \end{bmatrix} \dot{q} \end{aligned} \quad (5.2.8)$$

Then, the objectives can be written as follows:

- Task 1 : $\dot{e}_1 = 0 = (\begin{bmatrix} J_{rh}^{(\partial C)} & J_{rh}^{(\partial RA)} & 0 \end{bmatrix} - \begin{bmatrix} J_{lh}^{(\partial C)} & 0 & J_{lh}^{(\partial LA)} \end{bmatrix})\dot{q}$
- Task 2 : $\dot{e}_2 = \dot{x}_{rh} = \begin{bmatrix} J_{rh}^{(\partial C)} & J_{rh}^{(\partial RA)} & 0 \end{bmatrix} \dot{q}$

Writing the tasks in matrix form and considering that $J_{rh}^{(\partial C)} = J_{lh}^{(\partial C)}$ because both arms share the same Virtual COG Joints, the next equation is obtained:

$$\dot{e} = \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \end{bmatrix} = \begin{bmatrix} 0 & J_{rh}^{(\partial RA)} & -J_{lh}^{(\partial LA)} \\ J_{rh}^{(\partial C)} & J_{rh}^{(\partial RA)} & 0 \end{bmatrix} \dot{q} \quad (5.2.9)$$

Denoting by J_h the resulting analytical Jacobian and by $J_h^\#$ its right pseudo-inverse matrix, the inverse differential kinematics of the right hand using the second-order algorithm of section 4.3.2 is:

$$\ddot{q} = J_h^\#(q)(\ddot{x}_d + K_D\dot{e} + K_P e - \dot{J}_h(q, \dot{q})\dot{q}) + (I_n - J_h^\#J_h)\ddot{q}_0 \quad (5.2.10)$$

where \ddot{q}_0 is an homogeneous solution, used to satisfy the additional constraint of the distance from mechanical joint limits.

The task vector e is defined as:

$$e = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} e_{P,1} \\ e_{O,1} \\ e_{P,2} \\ e_{O,2} \end{bmatrix} \quad (5.2.11)$$

where $e_{P,i}$ and $e_{O,i}$ denote the position and orientation tasks, and are defined as²:

$$\begin{aligned} e_{P,i} &= p_{d,i} - p_i \\ e_{O,i} &= o_{d,i} - o_i \end{aligned} \quad (5.2.12)$$

For $i = 1$, $p_{d,1} = 0$ and $o_{d,1} = [0 \ 0 \ 0]$, then:

$$\begin{aligned} e_{P,i} &= p_{lh} - p_{rh} \\ e_{O,i} &= o_{lh} - o_{rh} \end{aligned} \quad (5.2.13)$$

For $i = 2$, the desired position $p_{d,2}$ and desired orientation $o_{d,2}$ are determined for the desired trajectory that both arms have to execute.

²The relation $e_{O,i} = o_{d,i} - o_i$ is for the case of an orientation expressed in RPY angles. Orientations expressed in other representation have to consider their respective differences. E.g. in quaternions, the relation is $e_{O,i} = \eta_i \varepsilon_d - S(\varepsilon_d) \varepsilon_i$.

5.3 Humanoid robots locomotion

5.3.1 Gait analysis

Due to the fact that some basic notions and terms in the domain of humanoid robotics are sometimes interpreted in different ways, it is important to define the notions and terms that are considered in this thesis.

Walk is the movement by putting forward (or backward) each foot in turn, not having both feet off the ground at once. And the manner of walking or running is called *gait* [24].

A *step* is the action of move one leg in contact with the ground from the rear position to the front position. The *gait cycle* is the period of time between any two identical events in the walking cycle. The initial contact is currently called as the starting and finishing event. The *gait stride* is the distance between two initial contacts of one foot. The stance and swing are the events of the gait cycle. *Stance* is the event when the foot is in contact with the ground (around 60 percent of the gait cycle). *Swing* is the event when the foot is in the air (around 40 percent of the gait cycle).

Each step consists at least two phases: a *single support* phase, when only one foot is in contact with the ground and a *double support* phase when both feet are in contact with the ground. The double support happens at the beginning and at the stance event. When the human is running the double support phase disappears. Thus, the locomotion mechanism changes its structure during a single walking cycle from an open to a closed kinematic chain. Generally, when a human walks normally, the initial and final double support take up about 15 percent of the gait cycle, and the single support time is equal to the swing event of the other leg.

A *periodic gait* occurs when the gait is realized by repeating the same step in an

identical way, involving then a periodicity of the motion of legs joints. A *repeatability* condition occurs when the state at the end of a step is equal to the state at its beginning. If a step can be divided in two equal time periods, and if the left leg in one period behaves as the right leg in the other then we speak of a *symmetric gait*. A *half-step* corresponds to the half-period and the motion realized in it.

An *ideal gait* represents a regular gait for which the repeatability and symmetry conditions can be mathematically checked. A *regular gait* is a periodic gait in which the leg in the single support phase is in contact with the ground by the whole foot area or the area of its front part (the toes link with the two-link foot), and in the case of the double-support phase the requirement applies to at least one foot.

5.3.1.1 Phases of a step

As detailed above, a step consists of single support and double support phases nevertheless a step can be divided in more specific phases [22]:

- Initial contact (IC)
- Loading response (LR)
- Midstance (MSt)
- Terminal stance(TSt)
- Preswing (PSw)
- Initial swing (ISw)
- Midswing (MSw)
- Terminal swing (TSw)

This involves that double support phase has five sub-phases and the single support phase has three sub-phases. The preswing sub-phase adapts the body for

going ahead, so it is included in the single support phase.

5.3.1.2 Tasks of a step

Considering the phases of section ??, a step involves three main tasks constitute a step, during the double support and single support phases:

- The *weight acceptance* includes the tasks of *initial contact* and *loading response* who must be compensated in order to maintain the kinetic stability while walking.
- The *single foot support* is composed by mid-stance, terminal stance and preswing.
- The *body or foot advancement* is constituted by preswing, initial swing, midswing and terminal swing.

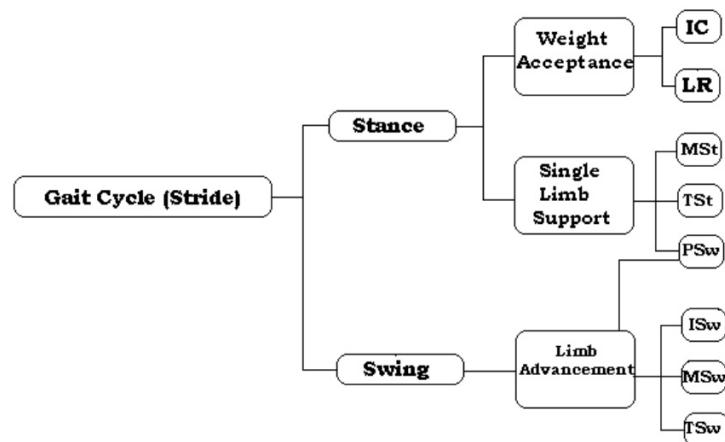


Figure 5.1: Tasks and phases of a step

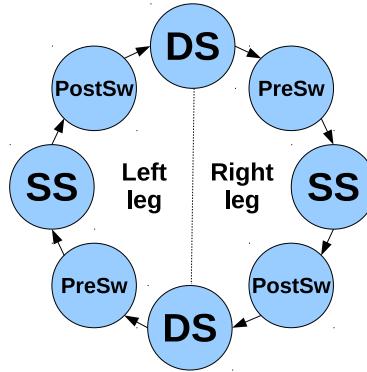


Figure 5.2: Phases of a step

5.3.2 Biped Locomotion Stability Criteria

In general terms, a gait is *stable* if the only contact between the biped and the floor is realized with the soles of the foot or feet, i.e. no other extremity of the biped is in contact with the floor [4].

Nevertheless, the above condition is very general and vague to generate stable steps in a humanoid robot. Therefore, it is important to define some metrics to consider in the gait generation algorithm. Before carrying on with the analysis of these criteria, an additional concept should be defined.

The *support polygon* is the surface determined by the contact of the foot and the ground. In a regular gait, the support polygon always has finite size. In the single-support phase the support polygon coincides with the area of the foot in contact with the ground, whereas in the double-support phase, the support polygon is a convex area determined by the areas of the feet and the ground and common tangents, so that the encompassed area is maximized [24].

5.3.2.1 Floor Projection of the Center of Gravity (FCOG)

A motionless biped only experiences gravitational forces that can be replaced by a virtual force R_N acting at the equivalent *center of gravity* of the biped (COG)³. The vector p_{COG} from the base-frame-origin to the COG can be described with:

$$p_{COG} = \frac{\sum_{i=1}^n m_i p_i}{\sum_{i=1}^n m_i} \quad (5.3.1)$$

where n is the number of links and p_i is the distance of the individual COGs. The *floor projection of the COG* (p_{FCOG}) can be taken from vector p_{COG} , by taking the x and y component:

$$\sum_{i=1}^n ((p_{FCOG} - p_i) \times m_i g) = 0 \quad (5.3.2)$$

If p_{FCOG} remains in the support polygon (SP), the motionless biped will not tip over or fall. However, when the motions become faster the dynamic forces will dominate the static forces and this criterion is not sufficient any more. Hence, other criteria have to be applied.

5.3.2.2 Zero Moment Point (ZMP)

The basic characteristics of all biped locomotion systems are: (i) the possibility of rotation of the overall system about one of the foot edges caused by strong disturbances, which is equivalent to the appearance of an unpowered (passive) and uncontrollable DOF, (ii) gait repeatability (symmetry), which is related to regular gait only, and (iii) regular interchangeability of single and double support phases [25].

The overall indicator of the mechanism behaviour is the point where the influence of all forces acting on the mechanism can be replaced by one single force.

³Centre of mass is the weighted average of location with respect to mass, whereas the centre of gravity is the weighted average of location with respect to mass times local g . In this thesis we assume that g is constant over the whole of the body, so they might have same value.

This point was termed the *Zero-Moment Point (ZMP)*.

In order to give a formal definition of the ZMP, the situation of a single-support is assumed. Considering a rigid foot with a flat sole which is fully contacting and supported by the ground, as in Figure 5.3. Replacing the influence of the biped by the force F_A and the moment M_A acting on a point A on the floor⁴. The gravitational acceleration g is acting in the negative z direction. To keep the whole biped in balance: in point P the reaction force $F_P = (F_{PX}, F_{PY}, F_{PZ})$ and the moment $M_P = (M_{PX}, M_{PY}, M_{PZ})$ are acting.

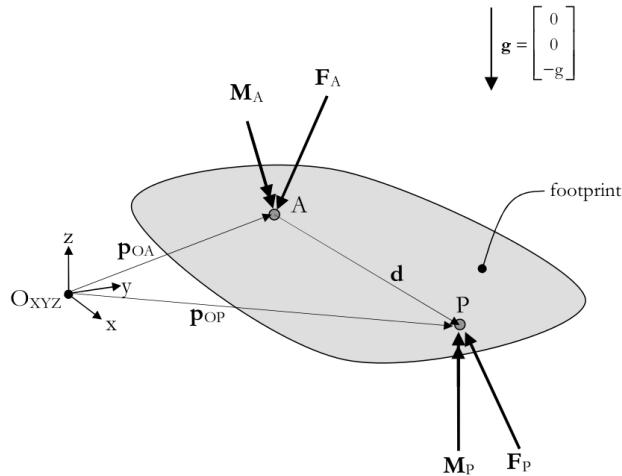


Figure 5.3: Forces and Moments acting on a rigid foot with a flat sole and fully supported with the floor [4]

The horizontal reaction force (F_{PX}, F_{PY}) is the friction force that is compensating for the horizontal components of force F_A . The vertical component of the reaction moment M_{PZ} is balancing the vertical component of moment M_A and the moment induced by the force F_A . Assuming there is no slip, the static friction is represented with (F_{PX}, F_{PY}) and M_{PZ} .

Compensating for the horizontal components of M_A , being (M_{AX}, M_{AY}) , the

⁴Point A can be considered as the ankle of the support foot

point P is shifted in such a way that F_{PZ} is fully compensating for them. This implies that the horizontal components of M_P are reduced to zero. Hence:

$$M_{PX} = M_{PY} = 0 \quad (5.3.3)$$

When the support polygon is not large enough to include the point P , the force F_P will act on the edge of the foot and the uncompensated part of M_A and F_A will produce a rotation about this edge resulting in the fall of the biped. The conservation of forces and moments around the origin allows to establish the static equilibrium equations:

$$F_P + F_A = 0 \quad (5.3.4)$$

$$p_{OP} \times F_P + M_A + M_{PZ} + p_{OA} \times F_A = 0 \quad (5.3.5)$$

where p_{OP} is the vector from the base-frame-origin O_{XYZ} to P and p_{OA} to point A .

If the base-frame-origin is placed on the XY-plane, (5.3.5) provides:

$$(p_{OP} \times F_P)_{XY} + (M_A)_{XY} + (p_{OA} \times F_A)_{XY} = 0 \quad (5.3.6)$$

The point P at which the reaction force F_P is acting represents the ZMP. This means that, in order to achieve a dynamically stable gait, the ZMP should be within the support polygon at every time instance.

5.3.2.3 Fictitious Zero Moment Point (FZMP) and Foot-Rotation Indicator (FRI)

In (5.3.6), the point P or ZMP was determined from the condition $M_{PX} = M_{PY} = 0$.

In the situation when the mechanism dynamics changes so that the ZMP approaches the support polygon edge (in either single-support or double-support

phases), the corresponding point will remain the ZMP only if no additional moments are acting at this point.

But if an additional moment appeared, the locomotion mechanism would start to rotate about the foot edge and the mechanism would collapse. In such a situation, the acting point of ground reaction force would be on the foot edge, but this point would not be ZMP any more, since both conditions (5.3.6) would not be fulfilled simultaneously. So the ZMP, outside the SP can be stated as a *fictitious ZMP (FZMP)*. This situation is represented in Figure 5.4.

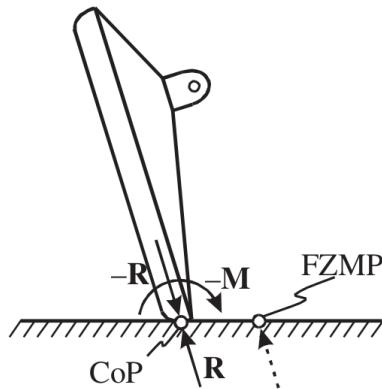


Figure 5.4: Fictitious ZMP (FZMP) [25]

The ZMP can only exist within the SP, if the ZMP reaches its edge, the biped is marginally stable, because a given perturbation that leads to a movement of the point P outside the SP (and becoming FZMP) will result in turning over. So the stable region is the SP, without its edges.

The same idea than the FZMP is presented explained in [5]. The FZMP is called *Foot-Rotation Indicator (FRI)* and it is defined as the point on the foot/ground contact surface, inside or outside the convex hull of the foot-support area, at which the resultant moment of the force/torque impressed on the foot is normal to the surface . With “impressed force/torque”, the author means the force and

torque at the ankle joint, other external forces, plus the weight of the foot, and not the ground-reaction forces.

The FRI point, and therefore the FZMP, presents some properties that can be used in gait planning. At first, the FRI indicates the occurrence of foot rotation and its direction. Secondly, once the FRI is outside the support area, it indicates the magnitude of the unbalanced moment on the foot. Finally, the FRI point indicates the stability margin of the robot. The stability margin of a robot against foot rotation may be quantified by the minimum distance of the support polygon boundary from the current location of the FRI point within the footprint.

Although FZMP and FRI indicators express the same idea, in this thesis the term FZMP is preferred.

5.3.2.4 Center of Pressure (CoP)

The *center of pressure* (CoP) is the point on the support polygon of the biped where the total sum of the contact forces F_R acts, causing a force but no moment. When standing, the part(s) of the body exerted by contact forces is(are) the foot(feet).

There are two types of forces that can be exerted on a foot: tangential and normal forces. These forces are called friction forces, F_F , and (normal) pressure forces, F_N , respectively. If the assumption that the foot cannot slide over the surface of the floor is made, the friction forces cancel each other out, what remains is a pressure field depicted in figure 5.5. The resultant in (normal direction) of this pressure field with n contact points, namely

$$F_{RN} = \sum_{i=1}^n F_{N_i} \quad (5.3.7)$$

acts on the CoP. So the position of the CoP with respect to the base-frame-origin,

denoted with pCoP, can be calculated with the equation:

$$p_{CoP} = \frac{\sum_{i=1}^n p_{FN_i} F_{N_i}}{F_{RN}} \quad (5.3.8)$$

The graphical representation of (5.3.8) is depicted in Figure 5.5. If the CoP is outside the support polygon, the biped tends to tip over, i.e. the biped will fall.

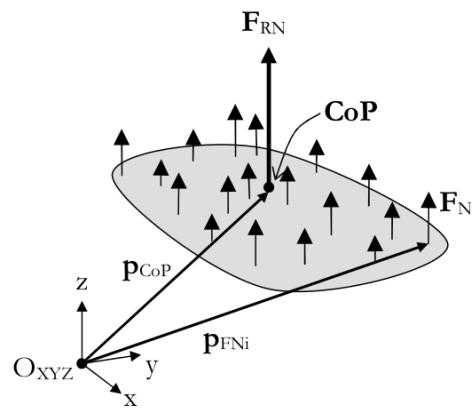


Figure 5.5: Center of Pressure (COP) [4]

5.3.2.5 Contact Wrench Sum

The ZMP is a rigorous stability criterion only when a robot walks on a flat plane without a contact between a hand and the environment under the sufficient friction assumption.

The *contact wrench sum* (CWS) is a strong stability criterion that can judge if the contact between a robot and the environment should be kept rigorously under the sufficient friction assumption when the robot walks on an arbitrary terrain with a possible contact between a hand and the environment [7]. The contact should be strongly stable if the contact wrench sum is an internal element of the *contact wrench cone* (CWC). Considering this, the CWS and CWC are equivalent to the ZMP and the supporting polygon respectively when the robot walks on a flat plane without a hand contact.

Considering a humanoid robot whose hands may be in contact with the environment as is illustrated in Figure 5.6 where Σ_R is the reference frame, Σ_B a frame fixed to the waist, and Σ_{L_i} a frame fixed to the COG of the i-th link of the robot ($i = 1, \dots, N$). Let \mathbf{p}_{L_i} and \mathbf{p}_B be the vector with respect to Σ_R of Σ_{L_i} and Σ_B respectively, \mathbf{p}_k ($k = 1, \dots, K$) the vertices of the support polygon of the hands and feet, and \mathbf{p}_G the position of the COG of the robot.

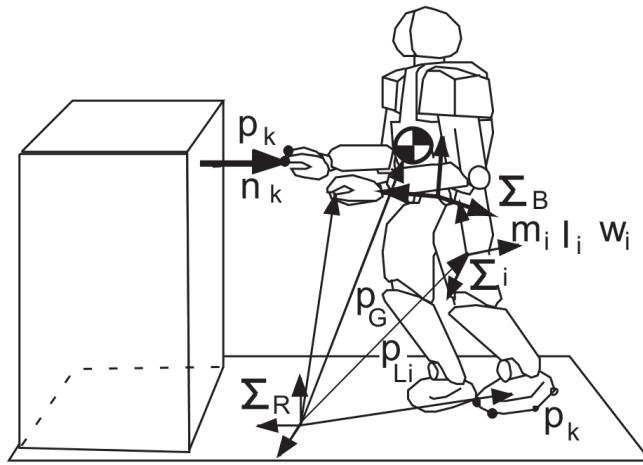


Figure 5.6: Contact Wrench Sum Model

The sum of the gravity and the inertia force applied to the robot, \mathbf{f}_G , and the sum of the moments about the COG of the robot, $\boldsymbol{\tau}_G$, can be given by:

$$\mathbf{f}_G = M(\mathbf{g} - \ddot{\mathbf{p}}_G) \quad (5.3.9)$$

$$\boldsymbol{\tau}_G = \mathbf{p}_G \times M(\mathbf{g} - \ddot{\mathbf{p}}_G) - \dot{\mathcal{L}} \quad (5.3.10)$$

where M is the total mass of the robot, \mathbf{g} the gravity vector, and \mathcal{L} the angular momentum of the robot with respect to the COG defined by:

$$\mathcal{L} = \sum_{i=1}^N \{m_i(\mathbf{p}_{L_i} - \mathbf{p}_G) \times \dot{\mathbf{p}}_{L_i} + \mathbf{I}_i \boldsymbol{\omega}_i\} \quad (5.3.11)$$

The contact force \mathbf{f}_C with respect to \sum_R which can be applied from the environment to the robot and its corresponding moment, $\boldsymbol{\tau}_C$, can be given by:

$$\mathbf{f}_C = \sum_{k=1}^K \sum_{l=1}^L \epsilon_k^l (\mathbf{n}_k + \mu_k \mathbf{t}_k^l) \quad (5.3.12)$$

$$\boldsymbol{\tau}_C = \sum_{k=1}^K \sum_{l=1}^L \epsilon_k^l \mathbf{p}_k (\mathbf{n}_k + \mu_k \mathbf{t}_k^l) \quad (5.3.13)$$

where the friction cone at \mathbf{p}_k is approximated by polyhedral cone, \mathbf{t}_k^l is a unit tangent vector to make $\mathbf{n}_k + \mu_k \mathbf{t}_k^l$ be the l -th edge of the polyhedral cone, μ_k the friction coefficient at \mathbf{p}_k and ϵ_k^l a nonnegative scalar that gives the magnitude of the force of the l -th edge of the approximated friction cone at the k -th contact point.

The set of $(\mathbf{f}_C, \boldsymbol{\tau}_C)$ forms the *polyhedral convex cone of the contact wrench*. The contact between the robot and the environment is *strongly stable* when it is guaranteed that the contact is stable to $(\mathbf{f}_C, \boldsymbol{\tau}_C)$. The contact is *weakly stable* when it is possible that the contact is stable to $(\mathbf{f}_C, \boldsymbol{\tau}_C)$. The contact is *strongly unstable* when it is not weakly stable.

If sufficient friction exists at the contact, an arbitrary friction force can be generated at every contact point independent to the normal force at the point:

$$\mathbf{f}_C = \sum_{k=1}^K (\epsilon_k^0 \mathbf{n}_k + \sum_{l=1}^4 \epsilon_k^l \mathbf{t}_k^l) \quad (5.3.14)$$

$$\boldsymbol{\tau}_C = \sum_{k=1}^K \mathbf{p}_k \times (\epsilon_k^0 \mathbf{n}_k + \sum_{l=1}^4 \epsilon_k^l \mathbf{t}_k^l) \quad (5.3.15)$$

where $\mathbf{t}_k^l (l = 1, \dots, 4)$ are the unit tangent vectors at \mathbf{p}_k whose nonnegative linear combination spans the tangent plane.

Considering this reasoning, if $(-\mathbf{f}_G, -\boldsymbol{\tau}_G)$ is an internal element of the polyhedral convex cone of the contact wrench given by equations (5.3.14) and (5.3.15),

then the contact is strongly stable to $(-\mathbf{f}_G, -\boldsymbol{\tau}_G)$. [7]

5.3.3 Humanoid Robot Gaits

5.3.3.1 Passive and active walking

5.3.3.1.1 Passive walking

Passive walking models or *cyclic walking* models use the potential field of gravity to produce walking. Normally, it is implemented on under-actuated and semi under-actuated legged robots.

A known passive walking model is McGeer's theory of passive dynamic bipedal locomotion, where the gait is simply a "natural repetitive motion of a dynamical system". The advantage of this system is that it consumes minimal energy and requires no computer control for normal walking on a flat surface. The disadvantage of the plain skeleton is that it is good for nothing else.

Giving only a downhill slope as a source of energy, a human-like pair of legs will settle into a natural gait generated by passive interaction of gravity and inertia, without any kind of muscular input (Figure 5.7).

Unlike straight-legged walking, knee-jointed legs has two advantages in passive walking: it offers a simple solution to the problem of foot clearance during the recovery phase and, in some cases, it is more stable [16]

5.3.3.1.2 Active walking

Active walking models is characterized by full actuated robots, i.e. all joints are actuated by a motor and a servo-driver. With active walking the swing leg does not fall on landing motion, that way the impact force is reduced considerably, so

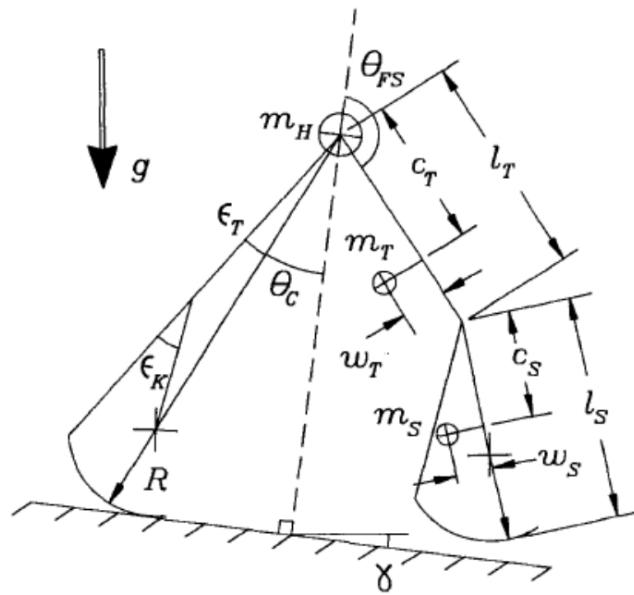


Figure 5.7: A chain of four rigid, inanimate links will walk by themselves down a shallow incline

stable motion is obtained.

For generating active walking motion of biped robots, suitable patterns should be developed taking into account the dynamics and stability during the step. In order to generate the motion patterns, The models that generate motion patterns with this approach can be classified as mass distributed or mass concentrated models.

With the active walking motion approach, the reference patterns are the spatial and the orientation trajectories with respect to the local axis for the COG and the swing foot. With these data, the inverse kinematics is used in order to obtain the joint trajectories.

5.3.4 Gait generation models

5.3.4.1 Static gait

A *static gait* is a gait where the FCoG and the ZMP always remain within the support foot during the entire motion or gait, usually close to the centroid of the support polygon. 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. Reason why the static gait is also known as slow walking.

In a statically stable gait the motion of the COG on the frontal plane is increased considerably compared to the normal walking motion and static torques (due to gravity) have a stronger effect during the act of walking compared to dynamic torques (due to the inertia). [22] In this case, the joint evolution is easily computed between the range limits to create a step with low order polynomials. In some cases, it is not necessary to compute the inverse kinematics, so it is possible to work with the joint space of the robot. The static gait is commonly used on humanoid robots of reduced scale which move many times with pre-planned patterns without whole body control, because the inertial and structural effects do not cause considerable disturbances during motion. In this case, the ZMP is almost the same as the COG projection on the walking surface.

5.3.4.2 Dynamic gait

If the ZMP resides within the support polygon during the motion or a gait of a humanoid while the FCoM leaves the SP, then the motion is a *dynamic gait* or dynamically stable gait. This kind of gait can be stable for faster movements, but the gait has to meet the requirements of the definition of a walk.

In a dynamic gait, smooth patterns must be generated in order to reduce the inertial effects, which are higher with respect to the static gait. Paradoxically, the inertial effects help to maintain biped stability during the step, because the

COG acceleration locates the ZMP closer to the middle of the SP. In this case, the COG acceleration must be controlled so that the ZMP does not overcome the SP. Some techniques reduce the jerk with high-order polynomials, but the disadvantage of high computation time doesn't allow real time applications of gait generation, so the gait is pre-planned and the on-line biped control cancel the external disturbance. Other techniques optimize the jerk, which reduces the computation time and it is possible to develop real-time walking patterns. The on-line control reduces external disturbances, such as gravity, terrain and structural imperfections. [22]

Running implies that at some time instance, both feet are off the floor, so there is no SP. For this reason, running can not be considered either as a dynamic gait or static gait.

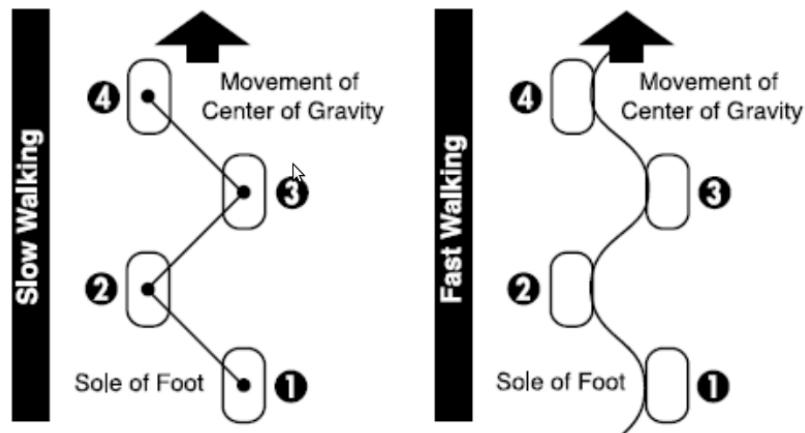


Figure 5.8: *Static and dynamic gait*

5.3.4.3 Mass Distributed Models

In broad terms, a *mass distributed model* is a gait generation model that takes into account the whole or partial body inertia and masses of the humanoid robot.

5.3.4.3.1 Pattern generator taking into account the contact wrench sum (CWS)

This walking generator method takes into account the contact wrench sum (CWS) described in section 5.3.2.5 as stability criteria. The whole body dynamics are taken into account with a multi-contact wrench.

The momentum of the robot \mathcal{P} and the angular momentum about the center of the gravity \mathcal{L} can be given by:

$$\begin{bmatrix} \mathcal{P} \\ \mathcal{L} \end{bmatrix} = \begin{bmatrix} M\mathbf{E} & -M\hat{\mathbf{r}}_{B \rightarrow G} & \mathbf{M}_{\dot{\theta}} \\ 0 & \tilde{\mathbf{I}} & \mathbf{H}_{\dot{\theta}} \end{bmatrix} \begin{bmatrix} \mathbf{v}_B \\ \boldsymbol{\omega}_B \\ \dot{\boldsymbol{\theta}} \end{bmatrix} \quad (5.3.16)$$

where \mathbf{p}_B , \mathbf{v}_B , $\boldsymbol{\omega}_B$ are the position, velocity and the angular velocity of the coordinates on the waist link of a humanoid robot, M is the total mass of the robot, \mathbf{E} the 3×3 unit matrix, $\mathbf{r}_{B \rightarrow G}$ the position vector from the origin of the waist coordinates to the COG of the robot, $\tilde{\mathbf{I}}$: 3×3 the inertia matrix about the COG, $\mathbf{M}_{\dot{\theta}}$: $3 \times n$, $\mathbf{H}_{\dot{\theta}}$: $3 \times n$ the inertia matrices which relate the joint velocities into the momentum and the angular momentum of the robot respectively. $\hat{\cdot}$ is the operator converting a 3×1 vector into a 3×3 skew-symmetric matrix whose multiplication from the left makes a vector product.

Since the reference of $\mathbf{v}_{F_i}^{ref}$, $\boldsymbol{\omega}_{F_i}^{ref}$ and $\mathbf{v}_{H_i}^{ref}$, $\boldsymbol{\omega}_{H_i}^{ref}$ are given, the joint velocity of the legs and the arms can be given by:

$$\dot{\boldsymbol{\theta}}_{leg_i} = \mathbf{J}_{leg_i}^{-1} \begin{bmatrix} \mathbf{v}_{F_i} \\ \boldsymbol{\omega}_{F_i} \end{bmatrix} - \mathbf{J}_{leg_i}^{-1} \begin{bmatrix} \mathbf{E} & -\hat{\mathbf{r}}_{B \rightarrow F_i} \\ 0 & \mathbf{E} \end{bmatrix} \begin{bmatrix} \mathbf{v}_B \\ \boldsymbol{\omega}_B \end{bmatrix} \quad (5.3.17)$$

$$\dot{\boldsymbol{\theta}}_{arm_i} = \mathbf{J}_{arm_i}^{-1} \begin{bmatrix} \mathbf{v}_{H_i} \\ \boldsymbol{\omega}_{H_i} \end{bmatrix} - \mathbf{J}_{arm_i}^{-1} \begin{bmatrix} \mathbf{E} & -\hat{\mathbf{r}}_{B \rightarrow H_i} \\ 0 & \mathbf{E} \end{bmatrix} \begin{bmatrix} \mathbf{v}_B \\ \boldsymbol{\omega}_B \end{bmatrix} \quad (5.3.18)$$

The reference trajectories of COG x_G , y_G are planned to follow the equations of

the moments about the x and y axes:

$$M(\ddot{z}_G + g)(y_G - y_C) - M\ddot{y}_G(z_G - z_C) + \dot{\mathcal{L}}_x = \sum_{k=1}^K \epsilon_k(y_k n_{kz} - z_k n_{ky}) \equiv \tau'_{C_x} \quad (5.3.19)$$

$$-M(\ddot{z}_G + g)(x_G - x_C) - M\ddot{x}_G(z_G - z_C) + \dot{\mathcal{L}}_y = -\sum_{k=1}^K \epsilon_k(x_k n_{kz} - z_k n_{kx}) \equiv \tau'_{C_y} \quad (5.3.20)$$

Considering the solution of the equations (5.3.19) and (5.3.20) be (x_G^{ref}, y_G^{ref}) , the reference of the momentum $(\mathcal{P}_x^{ref}, \mathcal{P}_y^{ref}, \mathcal{P}_z^{ref})$ can be given by:

$$\mathcal{P}_x^{ref} = M\dot{x}_G^{ref} \quad (5.3.21)$$

$$\mathcal{P}_y^{ref} = M\dot{z}_G^{ref} \quad (5.3.22)$$

$$\mathcal{P}_z^{ref} = M\dot{y}_G^{ref} \quad (5.3.23)$$

The reference of velocity and the angular velocity of the coordinates on the waist link, can be found by the resolved momentum control as:

$$\boldsymbol{\xi}_B = \mathbf{A}^{-1} \mathbf{y} \quad (5.3.24)$$

where

$$\mathbf{y} \equiv \begin{bmatrix} \mathcal{P}^{ref} \\ \mathcal{L}^{ref} \end{bmatrix} - \sum_{i=1}^2 \begin{bmatrix} \mathbf{M}_{F_i}^* \\ \mathbf{H}_{F_i}^* \end{bmatrix} \boldsymbol{\xi}_{F_i}^{ref} - \sum_{i=1}^2 \begin{bmatrix} \mathbf{M}_{H_i}^* \\ \mathbf{H}_{H_i}^* \end{bmatrix} \boldsymbol{\xi}_{H_i}^{ref} \quad (5.3.25)$$

$$\mathbf{A} \equiv \begin{bmatrix} \mathbf{M}_B^* \\ \mathbf{H}_B^* \end{bmatrix} \quad (5.3.26)$$

Finally, the motion pattern of a humanoid robot can be generated by the following algorithm [8]:

1. Give $(\mathbf{v}_{F_i}^{ref}, \boldsymbol{\omega}_{F_i}^{ref})$, $(\mathbf{v}_{H_i}^{ref}, \boldsymbol{\omega}_{H_i}^{ref})$ and $(\mathbf{v}_B^{ref}, \boldsymbol{\omega}_B^{ref})$.

2. Find $(\dot{\theta}_{leg_1}^{ref}, \dot{\theta}_{arm_2}^{ref}), (\dot{\theta}_{leg_1}^{ref}, \dot{\theta}_{arm_2}^{ref})$ by Eq. (5.3.17) and (5.3.18) respectively.
3. Find \mathcal{L}^{ref} by Eq. (5.3.16) and $\dot{\mathcal{L}}^{ref}$ by differentiation.
4. Give \ddot{z}_G^{ref} .
5. Give λ_k^{ref} .
6. Find $\ddot{x}_G^{ref}, \ddot{y}_G^{ref}$ by solving Eqs: (5.3.19) and (5.3.20).
7. Find \mathcal{P}^{ref} by Eqs. (5.3.21), (5.3.22)and (5.3.23).
8. Find ξ_B^{ref} by Eq. (5.3.24).
9. If v_B^{ref} found in Step 8 is close enough to v_B^{ref} given in Step 1, go to Step 10, otherwise let v_B^{ref} be that found in Step 8 and return to Step 1.

5.3.4.3.2 Two Masses Inverted Pendulum Mode (TMIPM)

The *Two Masses Inverted Pendulum Mode (TMIPM)*, is an approach that simplifies a robot model into a system with two masses. One mass characterizes the torso and the second one the swinging leg. The swing leg motion is almost arbitrary and the torso motion is calculated analytically.

Considering the system made of two masses as depicted in 5.9. The mass m_6^* represents the swinging leg and it is assumed to reside at the foot of the swinging leg. The mass m_3^* represents the remaining masses of the robot and it is assumed to reside at the torso. With respect this assumption, a torque equation around the ZMP is carried out:

$$\begin{aligned} \begin{bmatrix} 0 \\ M_y \\ 0 \end{bmatrix} + \begin{bmatrix} x_3 - p_x \\ y_3 \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ -m_3^*g \\ 0 \end{bmatrix} + \begin{bmatrix} x_6 - p_x \\ y_6 \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ -m_6^*g \\ 0 \end{bmatrix} \\ = \begin{bmatrix} x_3 - p_x \\ y_3 \\ 0 \end{bmatrix} \times \begin{bmatrix} m_3^*\ddot{x}_3 \\ m_3^*\ddot{y}_3 \\ 0 \end{bmatrix} + \begin{bmatrix} x_6 - p_x \\ y_6 \\ 0 \end{bmatrix} \times \begin{bmatrix} m_6^*\ddot{x}_6 \\ m_6^*\ddot{y}_6 \\ 0 \end{bmatrix} \end{aligned} \quad (5.3.27)$$

The evaluation of (5.3.27) around the z -axis (under the consideration of $y_3(t) = y_H$, $\dot{y}_3(t) = 0$) yields:

$$\ddot{x}_3(t) - \frac{g}{y_H}(x_3(t) - p_x(t)) = \frac{m_6^*}{m_3^* y_H}((x_6(t) - p_x(t))(\ddot{y}_6(t) + g) - \ddot{x}_6(t)y_6(t)) \quad (5.3.28)$$

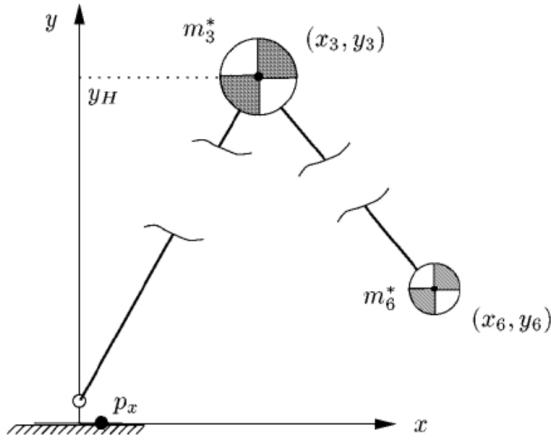


Figure 5.9: Mechanical model of a robot with two masses

Since the foot motion of the swinging leg $(x_6(t), y_6(t))$ and the ZMP motion $p_x(t)$ are prescribed, equation (5.3.28) represents an ordinary, linear, inhomogeneous differential equation for the torso motion $x_3(t)$. Theoretically, the highest gait stability is achieved for $p_x(t) = 0$ and the ZMP resides in any time instance exactly in the middle of the foot and the distance to the stability margin has a maximum value. Unfortunately, this quite restrictive constraint requires a hard discontinuity of the ZMP during the stride exchange which tends to excite oscillations. Nevertheless, assuming the ideal situation $p_x(t) = 0 \forall t$ in Equation (5.3.28):

$$\ddot{x}_3(t) - \frac{g}{y_H}x_3(t) = f(t) \quad (5.3.29)$$

with

$$f(t) = \frac{m_6^*}{m_3^* y_H}(gx_6(t) + x_6(t)\ddot{y}_6(t) - \ddot{x}_6(t)y_6(t)) \quad (5.3.30)$$

The motion of the torso now depends on the trajectory of the swinging leg. The functions $x_6(t)$ and $y_6(t)$ describe the motion of the foot of the swinging leg and are real analytic. Accordingly, the inhomogeneous excitation $f(t)$ on the right-hand side of Equation (5.3.29) itself is real analytic and can be calculated with standard mathematics.

5.3.4.3.3 Multiple Masses Inverted Pendulum Mode (MMIPM)

The *Multiple Masses Inverted Pendulum Mode (MMIPM)* is based on a model with several masses. One mass models the torso and an arbitrary number of masses is used to model the swinging leg. The foot motion of the swinging leg can be selected in accordance to the particular situation. The remaining trajectories of the robot are then calculated iteratively.

Considering the mechanical model in Figure ?? where tree masses are used for the modeling of the swinging leg. The inhomogeneous differential equation for the horizontal motion of the torso is:

$$\ddot{x}_3(t) - \frac{g}{y_H}x_3(t) = f(t) \quad (5.3.31)$$

with

$$f(t) = \sum_{i=4}^k \frac{m_i^*}{m_3^*} (gx_i(t) + x_i(t)\dot{y}_i(t) - \ddot{x}_i(t)y_i(t)) \quad (5.3.32)$$

Since the individual motions of the $k-3$ masses m_i^* depend on each other through the kinematic linkage, a direct solution is not feasible. On the other hand, this iterative algorithm is used:

1. Calculation of the joint trajectories $q_1(t)$ to $q_6(t)$ using inverse kinematics.
2. Calculation of the trajectories of the centres of gravity $(x_4(t), y_4(t)), (x_5(t), y_5(t))$ and $(x_6(t), y_6(t))$ for the masses of the swinging leg by direct kinematics.
3. Determination of the inhomogeneous excitation $f(t)$ using equation (5.3.32) and generation of a set of interpolation points.

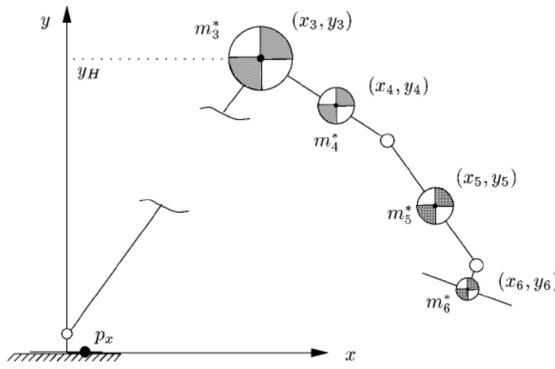


Figure 5.10: Mechanical model of a robot with multiple masses

4. Interpolation with a polynomial $f^x(t)$ function that consider all powers up to the order N:

$$f^*(t) = \sum_{i=1}^N a_i t^i + \dots, \quad N \in \mathbb{N}^+ \quad (5.3.33)$$

5. Solution of the differential equation (5.3.31) with the inhomogeneous excitation $f^*(t)$.
6. Check the terminal condition:

$$\int_{t_B}^{t_E} (x_3^{new}(t) - x_3^{old}(t))^2 dt < e, \quad e \in \mathbb{R}^+ \quad (5.3.34)$$

where the time instances t_B and t_E mark the beginning and the end of a stride. The trajectory $x_3^{new}(t)$ is the result of the current iteration and $x_3^{old}(t)$ is the trajectory of the last iteration.

If both trajectories differ only marginally stop the iteration, otherwise go to step 1 with the new torso trajectory $x_3^{new}(t)$.

5.3.4.4 Mass Concentrated Models

The *mass concentrated* models simplify the whole body dynamics to the center of gravity motion, by concentrating the whole-body mass in that point.

5.3.4.4.1 3D-Linear Inverted Pendulum Model

The *Three-Dimensional Linear Inverted Pendulum Mode* (3D-LIPM) is a simple linear dynamics obtained by the assumption that a constraint control is applied to an inverted pendulum such that the mass should move along an arbitrary defined plane [10] [11]. Considering the Cartesian coordinates as shown in Figure 5.11 and the x-axis as the ordinal walking direction. The constraint plane is represented with given normal vector $(k_x, k_y, -1)$ and z intersection z_c as:

$$z = k_x x + k_y y + z_c \quad (5.3.35)$$

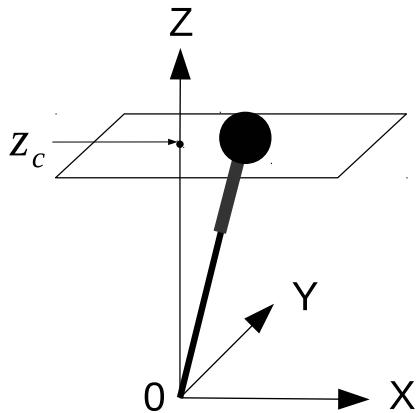


Figure 5.11: A pendulum under constraint

If the constraint plane is horizontal , the dynamics under the constraint control is given by:

$$\ddot{x} = \frac{g}{z_c} x - \frac{1}{mz_c} \tau_y \quad (5.3.36)$$

$$\ddot{y} = \frac{g}{z_c} y - \frac{1}{mz_c} \tau_x \quad (5.3.37)$$

where m is the mass of the pendulum, g is the gravity acceleration and τ_x, τ_y are the torques around x-axis and y-axis respectively.

On the other hand, if the sloped constraint where $k_x, k_y \neq 0$, the dynamics is the same by considering the additional constraint:

$$\tau_x x + \tau_y y = 0 \quad (5.3.38)$$

In the linear equations (5.3.36) and (5.3.37), z_c is the only parameter which governs those dynamics, i.e., the z intersection of the constraint plane and the inclination of the plane never affects the horizontal motion.

For the 3D-LIPM with the horizontal constraint ($k_x = k_y = 0$) the location of the Zero Moment Point (ZMP) on the floor (p_x, p_y) are:

$$\begin{aligned} p_x &= -\frac{\tau_y}{mg}, \\ p_y &= -\frac{\tau_x}{mg}, \end{aligned} \quad (5.3.39)$$

By substituting equations (5.3.39) to the 3D-LIPM (equations (5.3.36) and (5.3.37)):

$$\ddot{x} = \frac{g}{z_c} (x - p_x) \quad (5.3.40)$$

$$\ddot{y} = \frac{g}{z_c} (y - p_y) \quad (5.3.41)$$

5.3.4.4.2 Cart-table model

The *Cart-table model* [9] is a biped walking pattern generator that uses a preview control of the ZMP.

The method starts rewriting the equations (5.3.40) and (5.3.41) to control the ZMP, also called *ZMP equations*:

$$p_x = x - \frac{z_c}{g} \ddot{x} \quad (5.3.42)$$

$$p_y = y - \frac{z_c}{g} \ddot{y} \quad (5.3.43)$$

Kajita suggests a model that directly corresponds to equation (5.3.42). This model depicts a running cart of mass m on a pedestal table whose mass is negligible (It is necessary two sets of a cart on a table for the motion of x and y).

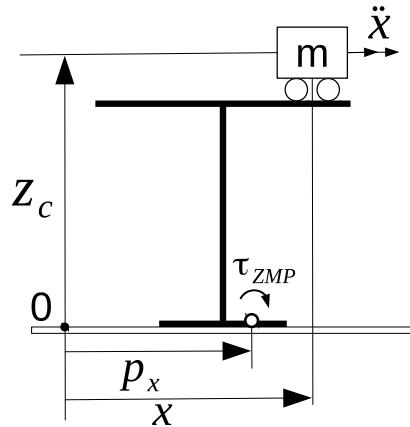


Figure 5.12: Cart-table model

As shown in the figure, the foot of the table is too small to let the cart stay on the edge. However, if the cart accelerates with a proper rate, the table can keep upright for a while. At this moment, the ZMP exists inside of the table foot. Since the moment around the ZMP must be zero, the following equations remains:

$$\tau_{ZMP} = mg(x - p_x) - m\ddot{x}z_c = 0 \quad (5.3.44)$$

Representing a humanoid robot as the cart-table model and relating the trajectory of the center of gravity (CoG) of the robot with the cart motion, the resulted ZMP can be calculated by equation (5.3.42).

Defining u_x as the time derivative of the horizontal acceleration of CoG and regarding it as the input of equation 5.3.42, the following dynamical systems are

obtained:

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_x \\ p_x &= \begin{bmatrix} 1 & 0 & -z_c/g \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} \end{aligned} \quad (5.3.45)$$

To define a preview servo controller, the system of equation 5.3.45 is discretized with sampling time of T :

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k), \\ p(k) &= Cx(k), \end{aligned} \quad (5.3.46)$$

where

$$\begin{aligned} x(k) &\equiv \begin{bmatrix} x(kT) & \dot{x}(kT) & \ddot{x}(kT) \end{bmatrix}^T, \\ u(k) &\equiv u_x(kT), \\ p(k) &\equiv p_x(kT), \\ A &\equiv \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, \\ B &\equiv \begin{bmatrix} T^3/6 \\ T^2/2 \\ T \end{bmatrix}, \\ C &= \begin{bmatrix} 1 & 0 & -z_c/g \end{bmatrix} \end{aligned}$$

and considering the performance index for a given reference of ZMP $p^{ref}(k)$ as:

$$J = \sum_{i=k}^{\infty} \{Q_e e(i)^2 + \Delta x^T(i) Q_x \Delta x(i) + R \Delta u^2(i)\} \quad (5.3.47)$$

where $e(i) \equiv p(i) - p^{ref}(i)$ is servo error, $Q_e, R > 0$ and Q_x is a 3×3 symmetric non-negative definite matrix, $\Delta x(k) \equiv x(k) - x(k-1)$ is the incremental state vector and $\Delta u(k) \equiv u(k) - u(k-1)$ is the incremental input.

When the ZMP reference can be previewed for N_L step future at every sampling time, the optimal controller which minimizes the performance index (5.3.47) is given by:

$$u(k) = -G_i \sum_{i=0}^k e(k) - G_x x(k) - \sum_{j=1}^{N_L} G_p(j) p^{ref}(k+j) \quad (5.3.48)$$

where G_i, G_x and $G_p(j)$ are the gains calculated from the weights Q_e, Q_x, R and the system parameter of equation (5.3.46).

The preview control is made of three terms, the integral action on the tracking error, the state feedback and the preview action using the future reference.

The ZMP reference is designed to stay in the center of support foot during single support phase, and to move from an old support foot to a new support foot during double support phase. To obtain a smooth ZMP trajectory in double support, Kajita uses a cubic spline.

Kajita considers the center of the pelvis link since it approximates the motion of the CoG.

If the ZMP error becomes too big relative to the stability margin determined by the foot geometry, the robot can fall. To fix the ZMP error, the preview control can be used. That is, first the CoG trajectory is calculated from the table-cart

model and the expected ZMP error is obtained from the multibody model.

These information are stored to the buffer memory and loaded to use after delay time of $T * N_L$. By this way, the future ZMP error can be used for the preview control to calculate a proper compensation.

The idea behind the ZMP preview control scheme proposed is therefore to minimize the jerk while maintaining a position $p(k)$ of the CoP as close as possible to the prescribed reference positions $p^{ref}(k)$. Considering this reference in the middle of the minimal $p^{min}(k)$ and maximal values $p^{max}(k)$, it corresponds to solving over a finite time interval $[kT, (k + N)T]$ the quadratic program [26]:

$$\min_{\ddot{x}(k), \dots, \ddot{x}(k+N)} \sum_{i=k}^{k+N-1} \frac{1}{2} Q(p(i+1) - p^{ref}(i+1))^2 + \frac{1}{2} R \ddot{x}^2(i) \quad (5.3.49)$$

where the ratio R/Q allows to balance the minimization of the jerks $\ddot{x}(i)$ with the tracking of the reference positions $p^{ref}(i)$.

The recursive relation (5.3.46) can be iterated N times in order to relate at once N values of the jerk $\ddot{x}(k)$ of the COG with N values of the position $p(k)$:

$$Z_{k+1} = P_x \hat{x}_k + P_u \ddot{X}_k \quad (5.3.50)$$

where

$$\begin{aligned}
Z_{k+1} &\equiv \begin{bmatrix} p(kT + 1) \\ \vdots \\ p(kT + N) \end{bmatrix}, \\
P_x &\equiv \begin{bmatrix} 1 & T & T^2/2 - z_c/g \\ \vdots & \vdots & \vdots \\ 1 & NT & N^2T^2/2 - z_c/g \end{bmatrix}, \\
\hat{x}_k &\equiv \begin{bmatrix} x(kT) \\ \dot{x}(kT) \\ \ddot{x}(kT) \end{bmatrix}, \\
P_u &\equiv \begin{bmatrix} T^3/6 - Tz_c/g & 0 & 0 \\ \vdots & \ddots & 0 \\ (1 + 3N + 3N^2)T^3/6 - Tz_c/g & \dots & T^3/6 - Tz_c/g \end{bmatrix}, \\
\ddot{X}_k &\equiv \begin{bmatrix} \ddot{x}(kT) \\ \vdots \\ \ddot{x}(kT + N - 1) \end{bmatrix}
\end{aligned}$$

with which the quadratic program (5.3.49) can be rewritten as:

$$\min_{\ddot{X}_k} \frac{1}{2} Q(Z_{k+1} - Z_{k+1}^{ref})^2 + \frac{1}{2} R \ddot{X}_k^2 \quad (5.3.51)$$

This quadratic program can be solved analytically, leading to:

$$\ddot{X}_k = -(P_u^T P_u + \frac{R}{Q} I_{N \times N})^{-1} P_u^T (P_x \hat{x}_k - Z_k^{ref}) \quad (5.3.52)$$

where $I_{N \times N}$ is an identity matrix. This way, the control signal ($u_x(kT)$) applied to the dynamics (5.3.46) is:

$$\ddot{x}_k = e^T \ddot{X}_k \quad (5.3.53)$$

with $e = [1, 0, \dots, 0]^T$.

Verifying the stability of this whole control scheme amounts to verifying that the norms of the 3 eigenvalues are smaller than 1 in the matrix:

$$A - Be^T(P_u^T P_u + \frac{R}{Q} I_{N \times N})^{-1} P_u^T P_x \quad (5.3.54)$$

Whole-body movement in TEO robot

6.1 Manipulation task

6.1.1 Single manipulation

In this section the implementation of some single manipulation tasks are shown. Considering the left hand operational point x_{lh} , some trajectories in the operational space are generated. This point in the real robot is shown in Figure 6.2.

At first, a trajectory that follows the outline of a square in the Y-Z plane and considers 7 points in the operational space that x_{lh} must follow. These points are shown in Figure ??.

These seven points have the next states:

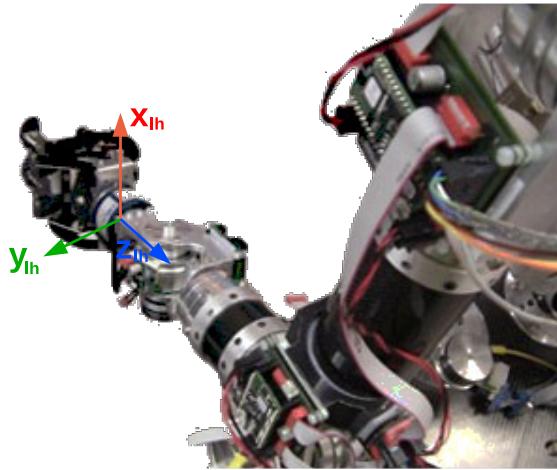


Figure 6.1: Left hand operational point of TEO

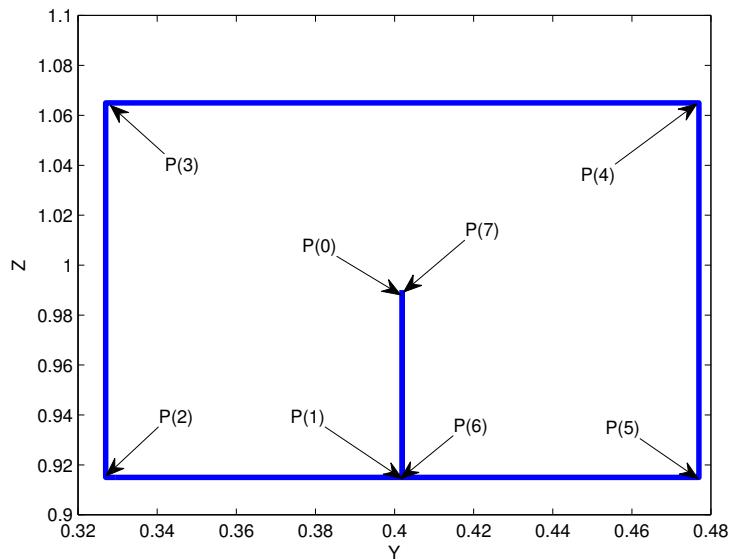


Figure 6.2: Trajectory 1 for the left hand x_{lh}

6.2 Locomotion task

The complete locomotion task for TEO robot is divided into several processes that are showed in Figure 6.3. Of all these processes, this thesis develops the gait

pattern generation for the operational points of 3.3 and the inverse differential kinematics that uses this points to generate the trajectories in the joints space.

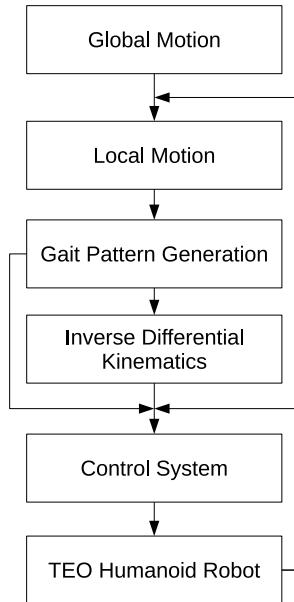


Figure 6.3: Locomotion Task

6.2.1 Global Motion

The objective of this layer is to find the global path of the humanoid robot to reach its global goal. This path can be defined by a user, if the humanoid is teleoperated, or obtained by any well-known path planning algorithms used in robotics if the humanoid is autonomous. For this purpose, the humanoid is considered one solid with its width, depth and height as boundaries. This solid should move to the desired position avoiding collision with the walking environment. This approach has been used by many researchers, such as [12] and [6].

6.2.2 Local Motion

Once the global trajectory for the humanoid is defined, it is necessary to translate it into local motion that allows the displacement of the robot. As detailed in section 5.3.1, humanoid displacement is produced by the feet movement in the swing phase. For this reason, the objective of this layer is to generate the sequence of footsteps that follows the global trajectory.

Considering an autonomous humanoid, the gait planning process can be developed by two approaches: a discrete footstep planning and a continuous footstep planning, as in [18], [13], [15], among others.

In this thesis, the goal swing pose is generated with the local axis gait algorithm proposed by Arburu [1]. Therefore, the goal footprint configuration (position p^{n+1} and yaw orientation φ_z^{n+1}) can be calculated by:

$$p^{n+1} = p^n + R(\varphi_z^{n+1})^T L^{n+1} \quad (6.2.1)$$

where,

$p^{n+1} = \begin{pmatrix} p_x^{n+1} \\ p_y^{n+1} \\ p_z^{n+1} \end{pmatrix}$ is the desired footprint position, $p^n = \begin{pmatrix} p_x^n \\ p_y^n \\ p_z^n \end{pmatrix}$ is the footprint of the support foot, $R(\theta_z^{n+1})^T = \begin{pmatrix} \cos(\theta_z^{n+1}) & -\sin(\theta_z^{n+1}) & 0 \\ \sin(\theta_z^{n+1}) & \cos(\theta_z^{n+1}) & 0 \\ 0 & 0 & 1 \end{pmatrix}$ is the elemental rotation matrix about Z axis and $L^{n+1} = \begin{pmatrix} L_x^{n+1} \\ L_y^{n+1} \\ L_z^{n+1} \end{pmatrix}$ is the desired footprint translation.

As seen in equation 6.2.1, goal footprint configuration depends on support foot position (p^n) because this foot is the local axis of gait input parameters (L^{n+1} and φ_z^{n+1}). Considering the locomotion task is for a walking in flat terrain, then $L_z^{n+1} = 0$ for all the footprints. This relation is shown in Figure 6.4.

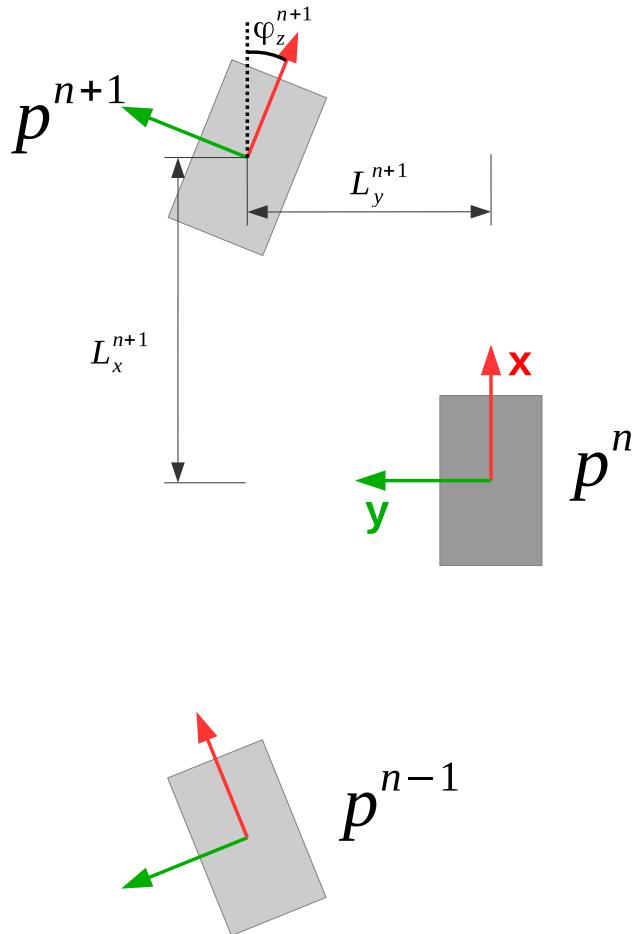


Figure 6.4: Local axis gait algorithm

6.2.3 Gait pattern generation

Once the final pose of the swing foot is determined, the gait pattern generator produces the trajectories of the operational points during the gait cycle. This process is shown in Figure 6.5. The gait pattern generator considers the poses of the operational points with respect to the support foot frame, reason why it is necessary a transformation of the inputs from the world frame to the support foot frame.

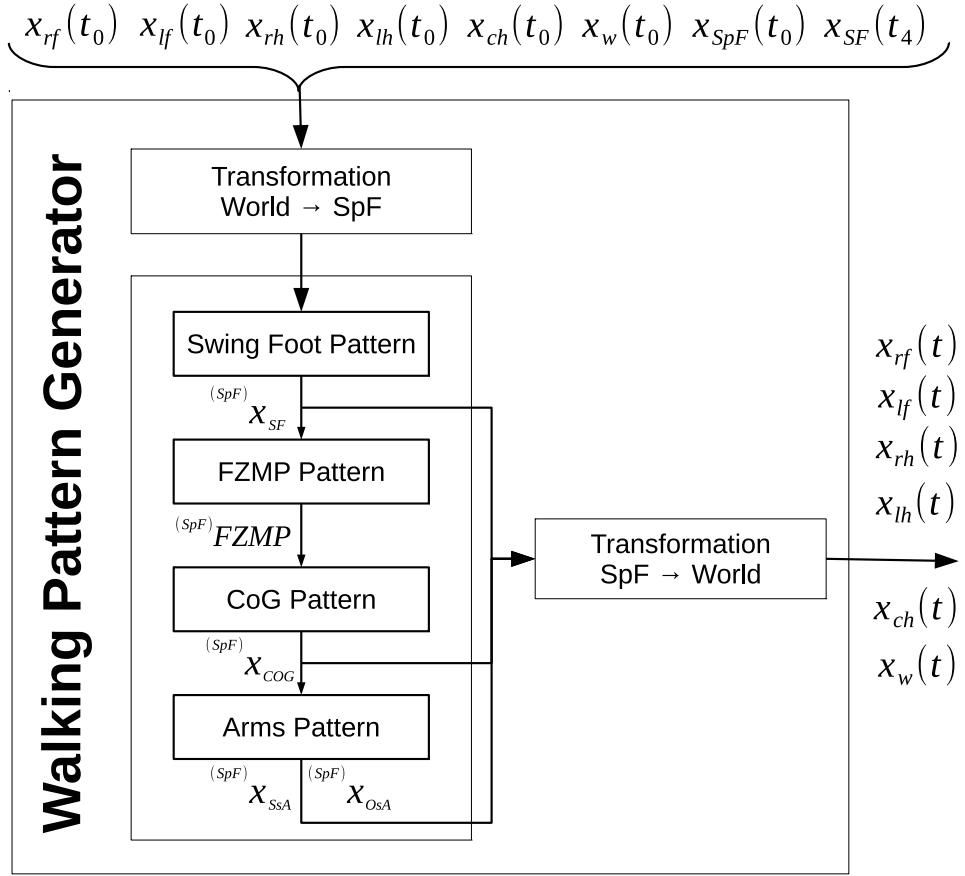


Figure 6.5: Walking pattern generation process

The proposed pattern generator divides the gait cycle in 4 periods, considering the phases of section 5.3.1.1:

1. **Period 1:** It corresponds to the period of time between double support (DS) and pre-swing (PreSw) phases: $t_0 \leq t \leq t_1$
2. **Period 2:** It corresponds to the period of time between pre-swing (PreSw) and the half of single support (SS) phases: $t_1 \leq t \leq t_2$
3. **Period 3:** It corresponds to the period of time between the half of single support (SS) and post-swing (PostSw) phases: $t_2 \leq t \leq t_3$

4. **Period 4:** It corresponds to the period of time between post-swing (PostSw) and double support (DS) phases: $t_3 \leq t \leq t_4$

Additionally, the assumptions of the gait pattern generator proposed in this thesis are detailed below:

- The gait is realized on a flat plane.
- The gait cycle starts and finishes in the DS phase.
- The heel of the swing foot does not lift from the floor in the PreSw phase and the entire sole of the swing foot lands in the PostSw phase, so swing foot will always stay parallel to the walking surface.
- Final linear velocity, final angular velocity, linear acceleration and final angular acceleration of the swing foot are zero in period 2 and period 3.
- The initial projection of the CoG in Phase 1 and the final projection of the CoG in Phase 4 have to be in the center of the support polygon (SP).
- The torso during walking is straight so the waist horizontal position is equal to the CoG horizontal position (the waist and CoG are rigidly fixed).
- The pitch and roll parameters of the waist orientation will stay at zero so always stay parallel to the walking surface.
- The initial yaw orientation of the waist in Phase 1 and the final yaw orientation of the waist in Phase 4 are equal to the mean yaw orientation of both feet.
- Final linear velocity, final linear acceleration, final linear acceleration and final angular acceleration of the waist are zero at the end of Phase 4.

6.2.3.1 Feet pattern

To produce a step, the swing foot has to be moved from the rear position to the front position. This movement is shown in 6.6 and the desired conditions at the end of each period are presented below:

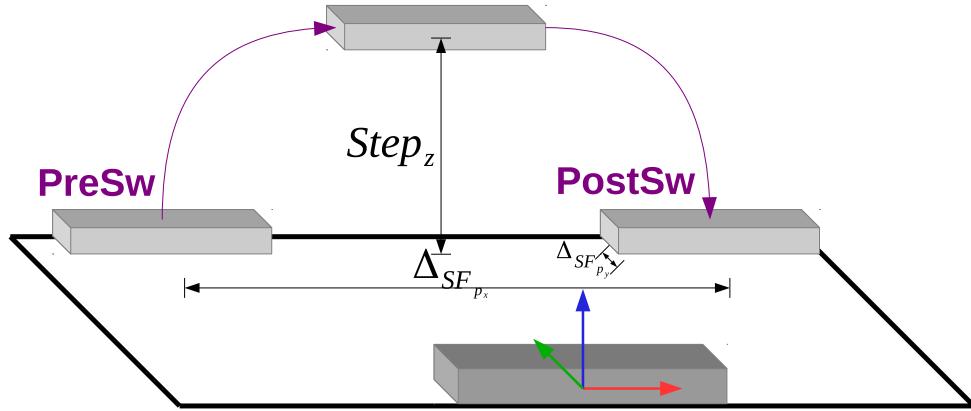


Figure 6.6: Swing foot movement

- **Period 1**

As mentioned in the assumptions in Section 6.2.3, the heel of the swing foot does not lift from the floor in the PreSw phase then the pose of the foot is the same than its pose in the double support phase. Therefore, at the end of this period the swing foot has the final state:

$$\begin{aligned}
 {}^{(SpF)}x_{SF}(t_1) &= {}^{(SpF)}x_{SF}(t_0), \\
 {}^{(SpF)}\dot{x}_{SF}(t_1) &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad {}^{(SpF)}\ddot{x}_{SF}(t_1) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{6.2.2}
 \end{aligned}$$

- **Period 2**

At the end of this period, the swing foot has the final state:

$$x_{SF}(t_2) = \begin{bmatrix} (SpF)x_{SF_{px}}(t_1) + \nu_{p_x} \cdot \Delta_{SF_{px}} \\ (SpF)x_{SF_{py}}(t_1) + \nu_{p_y} \cdot \Delta_{SF_{py}} \\ Step_z \\ (SpF)x_{SF_{\varphi_x}}(t_1) \\ (SpF)x_{SF_{\varphi_y}}(t_1) \\ (SpF)x_{SF_{\varphi_z}}(t_1) + \nu_{\varphi_z} \cdot \Delta_{SF_{\varphi_z}} \end{bmatrix},$$

$$(SpF)\dot{x}_{SF}(t_2) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (SpF)\ddot{x}_{SF}(t_2) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (6.2.3)$$

where, $\Delta_{SF_{px}}$ and $\Delta_{SF_{py}}$ are the total translation of swing foot from initial swing footstep and final swing footstep in X and Y axes respectively. $Step_z$ is the maximum height that swing foot is lifted. ν_{p_x} and ν_{p_y} are the percentage of total $\Delta_{SF_{px}}$ and $\Delta_{SF_{py}}$ translation executed during this period respectively.

- **Period 3**

At the end of this period, the swing foot has the final state:

$$x_{SF}(t_3) = \begin{bmatrix} {}^{(SpF)}x_{SF_{px}}(t_2) + (1 - \nu_{px}) \cdot \Delta_{SF_{px}} \\ {}^{(SpF)}x_{SF_{py}}(t_2) + (1 - \nu_{py}) \cdot \Delta_{SF_{py}} \\ 0 \\ {}^{(SpF)}x_{SF_{\varphi_x}}(t_2) \\ {}^{(SpF)}x_{SF_{\varphi_y}}(t_2) \\ {}^{(SpF)}x_{SF_{\varphi_z}}(t_2) + (1 - \nu_{\varphi_z}) \cdot \Delta_{SF_{\varphi_z}} \end{bmatrix},$$

$${}^{(SpF)}\dot{x}_{SF}(t_3) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad {}^{(SpF)}\ddot{x}_{SF}(t_3) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (6.2.4)$$

- **Period 4**

As mentioned in the assumptions in Section 6.2.3, the entire sole of the swing foot lands in the PostSw phase, then the pose of the foot is the same than its pose in period 3. Therefore, at the end of this period the swing foot has the final state:

$${}^{(SpF)}x_{SF}(t_4) = {}^{(SpF)}x_{SF}(t_3),$$

$${}^{(SpF)}\dot{x}_{SF}(t_4) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \ddot{x}_{SF}(t_4) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (6.2.5)$$

6.2.3.1.1 Feet trajectories

- **Polynomial interpolation** Due to the trajectory of the feet has defined boundary conditions for position, velocity and acceleration, at least a quin-

tic (fifth-order) polynomial has to be used. Using a quintic polynomial smooth trajectories are obtained for the feet poses and its linear/angular velocities and accelerations, as is shown in Figures .

6.2.3.2 FZMP pattern

Due to the objective of generate stable motion, it is necessary to determine the FZMP pattern during the 4 proposed periods. The desired poses of the FZMP are shown in 6.7 and their values are presented below:

- **Period 1**

As mentioned in the assumptions in Section 6.2.3, the cycle starts in DS phase and the FZMP is in the centroid of the SP, later the FZMP has to be translated to the center of the support foot. Therefore, at the end of this period the FZMP has the final state:

$$\begin{aligned} {}^{(SpF)}FZMP_{p_x}(t_1) &= 0 \\ {}^{(SpF)}FZMP_{p_y}(t_1) &= 0 \\ {}^{(SpF)}FZMP_{p_z}(t_1) &= 0 \end{aligned} \quad (6.2.6)$$

- **Period 2**

As the FZMP trajectory in this period is executed in single support, this indicator position does not vary so the FZMP has the final state:

$$\begin{aligned} {}^{(SpF)}FZMP_{p_x}(t_2) &= {}^{(SpF)}FZMP_{p_x}(t_1) \\ {}^{(SpF)}FZMP_{p_y}(t_2) &= {}^{(SpF)}FZMP_{p_y}(t_1) \\ {}^{(SpF)}FZMP_{p_z}(t_2) &= {}^{(SpF)}FZMP_{p_z}(t_1) \end{aligned} \quad (6.2.7)$$

- **Period 3**

Considering the same arguments than Period 2, the FZMP has the final state:

$$\begin{aligned} {}^{(SpF)}FZMP_{p_x}(t_3) &= {}^{(SpF)}FZMP_{p_x}(t_2) \\ {}^{(SpF)}FZMP_{p_y}(t_3) &= {}^{(SpF)}FZMP_{p_y}(t_2) \\ {}^{(SpF)}FZMP_{p_z}(t_3) &= {}^{(SpF)}FZMP_{p_z}(t_2) \end{aligned} \quad (6.2.8)$$

- **Period 4**

As mentioned in the assumptions in Section 6.2.3, the cycle finishes in DS phase and the FZMP has to be in the centroid of the SP. Therefore, at the end of this period the final state of the FZMP is:

$$\begin{aligned} {}^{(SpF)}FZMP_{px}(t_4) &= {}^{(SpF)}x_{SF_{px}}(t_4)/2 \\ {}^{(SpF)}FZMP_{py}(t_4) &= {}^{(SpF)}x_{SF_{py}}(t_4)/2 \\ {}^{(SpF)}FZMP_{pz}(t_4) &= {}^{(SpF)}FZMP_{pz}(t_3) \end{aligned} \quad (6.2.9)$$

6.2.3.2.1 FZMP trajectories

- **Polynomial interpolation** Although the trajectory of the FZMP only has conditions for position and a linear function is enough, it is necessary a smooth change in the FZMP translation from the centroid of the SP to the support foot. For this reason, any polynomial interpolator produces a trajectory with this feature. In this case, a quintic polynomial is preferred.

Figure 6.7: FZMP trajectory - XY plane

6.2.3.3 CoG pattern

As mentioned in the assumptions in Section 1.1.3, the waist and CoG are rigidly fixed, for this reason the trajectory of the CoG is the same than the waist but with a translation in the axis Z. Considering this, the trajectory of the waist ${}^{(SpF)}x_w(t)$ is generated and projected in the CoG plane.

- **Period 1**

As mentioned in the assumptions in Section 6.2.3, the waist and CoG are rigidly fixed. Therefore, at the end of this period the waist has the final state:

$$\begin{aligned}
 {}^{(SpF)}x_w(t_1) &= \begin{bmatrix} \lambda \cdot {}^{(SpF)}x_{SF_{px}}(t_0)/2 \\ \beta \cdot {}^{(SpF)}x_{SF_{py}}(t_0)/2 \\ {}^{(SpF)}x_{w_{pz}}(t_0) \\ {}^{(SpF)}x_{w_{\varphi_x}}(t_0) \\ {}^{(SpF)}x_{w_{\varphi_y}}(t_0) \\ \Omega_1 \cdot ({}^{(SpF)}x_{SF_{\varphi_z}}(t_4)/2 - {}^{(SpF)}x_{w_{\varphi_z}}(t_0)) \end{bmatrix} \\
 {}^{(SpF)}\dot{x}_w(t_1) &= \begin{bmatrix} V_{w_{px}}(t_1) \\ V_{w_{py}}(t_1) \\ 0 \\ 0 \\ 0 \\ V_{w_{\varphi_z}}(t_1) \end{bmatrix}, \quad {}^{(SpF)}\ddot{x}_w(t_1) = \begin{bmatrix} A_{w_{px}}(t_1) \\ A_{w_{py}}(t_1) \\ 0 \\ 0 \\ 0 \\ A_{w_{\varphi_z}}(t_1) \end{bmatrix} \tag{6.2.10}
 \end{aligned}$$

where λ and β are percentages of the distance from the initial position to the projection of centroid of the support foot in X and Y axis. Ω_1 is a percentage of the total yaw variation from initial yaw angle and the desired at the end of the gait cycle. $V_{w_{px}}(t_1)$ and $V_{w_{py}}(t_1)$ are the X and Y components of the waist linear velocity at the end of period 1 respectively. $V_{w_{\varphi_z}}(t_1)$ is the yaw angular velocity of the waist at the end of period 1. $A_{w_{px}}(t_1)$ and $A_{w_{py}}(t_1)$ are the X and Y components of the waist linear acceleration at the end of period 1. $A_{w_{\varphi_z}}(t_1)$ is the yaw angular acceleration of the waist at the end of period 1.

- **Period 2**

At the end of this period, the waist has the final state:

$$\begin{aligned}
 {}^{(SpF)}x_w(t_2) &= \begin{bmatrix} 0 \\ \alpha \cdot {}^{(SpF)}x_{SF_{py}}(t_0)/2 \\ {}^{(SpF)}x_{w_{pz}}(t_1) \\ {}^{(SpF)}x_{w_{\varphi x}}(t_1) \\ {}^{(SpF)}x_{w_{\varphi y}}(t_1) \\ \Omega_2 \cdot ({}^{(SpF)}x_{SF_{\varphi z}}(t_4)/2 - {}^{(SpF)}x_{w_{\varphi z}}(t_1)) \end{bmatrix} \\
 {}^{(SpF)}\dot{x}_w(t_2) &= \begin{bmatrix} V_{w_{px}}(t_2) \\ V_{w_{py}}(t_2) \\ 0 \\ 0 \\ 0 \\ V_{w_{\varphi z}}(t_2) \end{bmatrix}, \quad {}^{(SpF)}\ddot{x}_w(t_2) = \begin{bmatrix} A_{w_{px}}(t_2) \\ A_{w_{py}}(t_2) \\ 0 \\ 0 \\ 0 \\ A_{w_{\varphi z}}(t_2) \end{bmatrix} \tag{6.2.11}
 \end{aligned}$$

where α is a percentages of the distance from the initial position of the waist to the projection of centroid of the support foot in Y axis. Ω_2 is a percentage of the total yaw variation from initial yaw angle in phase 2 and the desired at the end of the gait cycle. And $V_{w_{px}}(t_2)$, $V_{w_{py}}(t_2)$, $V_{w_{\varphi z}}(t_2)$, $A_{w_{px}}(t_2)$, $A_{w_{py}}(t_2)$ and $A_{w_{\varphi z}}(t_2)$ has the same interpretation than their counterparts in period 1.

• Period 3

At the end of this period, the waist has the final state:

$$\begin{aligned}
 {}^{(SpF)}x_w(t_3) &= \begin{bmatrix} \lambda \cdot {}^{(SpF)}x_{SF_{px}}(t_4)/2 \\ \beta \cdot {}^{(SpF)}x_{SF_{py}}(t_4)/2 \\ {}^{(SpF)}x_{w_{pz}}(t_2) \\ {}^{(SpF)}x_{w_{\varphi_x}}(t_2) \\ {}^{(SpF)}x_{w_{\varphi_y}}(t_2) \\ \Omega_3 \cdot ({}^{(SpF)}x_{SF_{\varphi_z}}(t_4)/2 - {}^{(SpF)}x_{w_{\varphi_z}}(t_2)) \end{bmatrix} \\
 {}^{(SpF)}\dot{x}_w(t_3) &= \begin{bmatrix} V_{w_{px}}(t_3) \\ V_{w_{py}}(t_3) \\ 0 \\ 0 \\ 0 \\ V_{w_{\varphi_z}}(t_3) \end{bmatrix}, \quad {}^{(SpF)}\ddot{x}_w(t_3) = \begin{bmatrix} A_{w_{px}}(t_3) \\ A_{w_{py}}(t_3) \\ 0 \\ 0 \\ 0 \\ A_{w_{\varphi_z}}(t_3) \end{bmatrix} \tag{6.2.12}
 \end{aligned}$$

- **Period 4**

As mentioned in the assumptions in Section 6.2.3, Therefore, at the end of this period the waist has the final state:

$$\begin{aligned}
 {}^{(SpF)}x_w(t_4) &= \begin{bmatrix} {}^{(SpF)}x_{SF_{px}}(t_4)/2 \\ {}^{(SpF)}x_{SF_{py}}(t_4)/2 \\ {}^{(SpF)}x_{w_{pz}}(t_3) \\ {}^{(SpF)}x_{w_{\varphi_x}}(t_3) \\ {}^{(SpF)}x_{w_{\varphi_y}}(t_3) \\ {}^{(SpF)}x_{SF_{\varphi_z}}(t_4)/2 \end{bmatrix} \\
 {}^{(SpF)}\dot{x}_w(t_4) &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad {}^{(SpF)}\ddot{x}_w(t_4) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{6.2.13}
 \end{aligned}$$

Figure 6.8: Waist trajectory - XY plane

6.2.3.3.1 CoG trajectories

- **Polynomial interpolation** Due to the trajectory of the feet has defined boundary conditions for position, velocity and acceleration, at least a quintic (fifth-order) polynomial has to be used. Using a quintic polynomial smooth trajectories are obtained for the feet poses and its linear/angular velocities and accelerations, as is shown in Figures .

Polynomial of degree 5 because it allows to specify the initial pose, initial linear velocity, initial angular velocity, final pose, final linear velocity and final angular velocity.

The final horizontal position of the CoM and ZMP coincide

VIAPOINTS

6.2.3.4 Arms pattern

- **DS - PreSw period**

As mentioned in the assumptions in Section 6.2.3, the swing foot does not move in this phase, and therefore the pose of the foot is the same than its pose in the double support phase.

$$\begin{aligned} FZMP_{p_x}(t_1) &= FZMP_{p_x}(t_0), & FZ\dot{M}P_{p_x}(t_1) &= 0 \\ FZMP_{p_y}(t_1) &= FZMP_{p_y}(t_0), & FZ\dot{M}P_{p_y}(t_1) &= 0 \\ FZMP_{p_z}(t_1) &= FZMP_{p_z}(t_0), & FZ\dot{M}P_{p_z}(t_1) &= 0 \\ FZMP_{\varphi_x}(t_1) &= FZMP_{\varphi_x}(t_0), & FZ\dot{M}P_{\varphi_x}(t_1) &= 0 \\ FZMP_{\varphi_y}(t_1) &= FZMP_{\varphi_y}(t_0), & FZ\dot{M}P_{\varphi_y}(t_1) &= 0 \\ FZMP_{\varphi_z}(t_1) &= FZMP_{\varphi_z}(t_0), & FZ\dot{M}P_{\varphi_z}(t_1) &= 0 \end{aligned}$$

- **PreSw - SS period**

In this period, the swing foot is translated:

$$\begin{aligned}
 FZMP_{p_x}(t_2) &= \nu_x \cdot L_{p_x}, & FZ\dot{M}P_{p_x}(t_2) &= 0 \\
 FZMP_{p_y}(t_2) &= \nu_y \cdot L_{p_y}, & FZ\dot{M}P_{p_y}(t_2) &= 0 \\
 FZMP_{p_z}(t_2) &= L_{p_z}, & FZ\dot{M}P_{p_z}(t_2) &= 0 \\
 FZMP_{\varphi_x}(t_2) &= FZMP_{\varphi_x}(t_1), & FZ\dot{M}P_{\varphi_x}(t_2) &= 0 \\
 FZMP_{\varphi_y}(t_2) &= FZMP_{\varphi_x}(t_1), & FZ\dot{M}P_{\varphi_y}(t_2) &= 0 \\
 FZMP_{\varphi_z}(t_2) &= \kappa_{yaw} \cdot L_{\varphi_z}, & FZ\dot{M}P_{\varphi_z}(t_2) &= 0
 \end{aligned}$$

where, L_{p_x} , L_{p_y} and L_{p_z} are the total translation of swing foot from initial swing footprint and final swing footprint. L_{φ_z} is the total yaw rotation from initial swing footprint and final swing footprint. ν_x and ν_y are the percentage of total L_{p_x} and L_{p_y} translation executed during this period respectively. κ_{yaw} is the percentage of total L_{φ_z} rotation executed during this period.

- **SS - PostSw period**

In this period, the swing foot is landed and translated:

$$\begin{aligned}
 FZMP_{p_x}(t_3) &= (1 - \nu_x) \cdot L_{p_x}, & FZ\dot{M}P_{p_x}(t_3) &= 0 \\
 FZMP_{p_y}(t_3) &= (1 - \nu_y) \cdot L_{p_y}, & FZ\dot{M}P_{p_y}(t_3) &= 0 \\
 FZMP_{p_z}(t_3) &= 0, & FZ\dot{M}P_{p_z}(t_3) &= 0 \\
 FZMP_{\varphi_x}(t_3) &= FZMP_{\varphi_x}(t_2), & FZ\dot{M}P_{\varphi_x}(t_3) &= 0 \\
 FZMP_{\varphi_y}(t_3) &= FZMP_{\varphi_x}(t_2), & FZ\dot{M}P_{\varphi_y}(t_3) &= 0 \\
 FZMP_{\varphi_z}(t_3) &= (1 - \kappa_{yaw}) \cdot L_{\varphi_z}, & FZ\dot{M}P_{\varphi_z}(t_3) &= 0
 \end{aligned}$$

- **PostSw - DS period**

As mentioned in the assumptions in Section 6.2.3, the swing foot does not move in this phase, and therefore the pose of the foot is the same than its

pose in the previous period.

$$\begin{aligned}
 FZMP_{p_x}(t_4) &= FZMP_{p_x}(t_3), & FZ\dot{M}P_{p_x}(t_4) &= 0 \\
 FZMP_{p_y}(t_4) &= FZMP_{p_y}(t_3), & FZ\dot{M}P_{p_y}(t_4) &= 0 \\
 FZMP_{p_z}(t_4) &= FZMP_{p_z}(t_3), & FZ\dot{M}P_{p_z}(t_4) &= 0 \\
 FZMP_{\varphi_x}(t_4) &= FZMP_{\varphi_x}(t_3), & FZ\dot{M}P_{\varphi_x}(t_4) &= 0 \\
 FZMP_{\varphi_y}(t_4) &= FZMP_{\varphi_y}(t_3), & FZ\dot{M}P_{\varphi_y}(t_4) &= 0 \\
 FZMP_{\varphi_z}(t_4) &= FZMP_{\varphi_z}(t_3), & FZ\dot{M}P_{\varphi_z}(t_4) &= 0
 \end{aligned}$$

6.2.3.4.1 FZMP trajectories

- **Polynomial interpolation** Polynomial of degree 5 because it allows to specify the initial pose, initial linear velocity, initial angular velocity, final pose, final linear velocity and final angular velocity.

6.2.4 Kinematics Transformation

Conclusions and Future Work

7.1 Conclusions

7.2 Future Work

- a joint acceleration-based controller coupled with inverse dynamics, had slightly better performance in light of modelling errors.



Appendices

Appendix

A

TEOTraGen

TEOTraGen is a graphical user interface (GUI) written in MATLAB®, that allows the user to generate trajectories in the operational space for TEO humanoid robot. The main window of TEOTraGen is shown in Figure A.1. This window allows the access to the two primary applications: TEOWholeGen and TEOStepGen.



Figure A.1: TEOTraGen main window

A.1 TEOWholeGen

TEOWholeGen is the application of TEOTraGen that allows the user to generate trajectories in the operational space for the six operational points of TEO: x_{rf} , x_{lf} , x_w , x_t , x_{rh} and x_{lh} and map them in the joint space. The first window that appears when the user access to the application is shown in Figure A.2. This window allows the user to select the parameters that the trajectory generator will use.

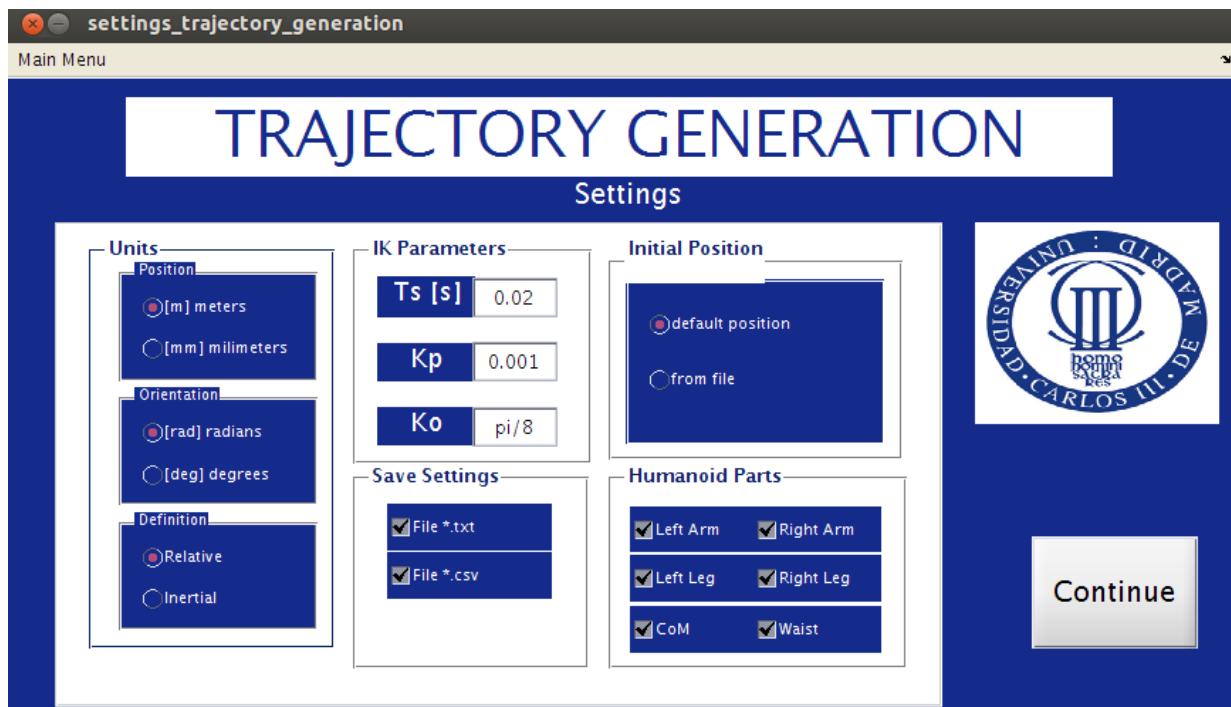


Figure A.2: TEOWholeGen settings window

The main window of TEOWholeGen is shown in Figure A.3. This window contains the main buttons, text boxes and panels where the user can customize the desired trajectories.

The application allows the user to select the number of points that the trajectory

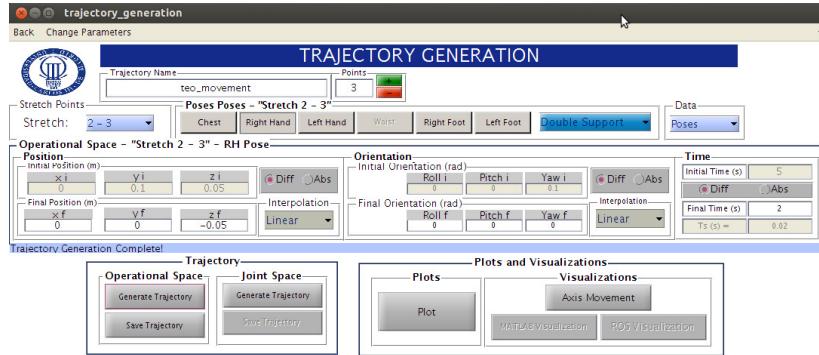


Figure A.3: Trajectory generation window

generator will consider by changing the value of the points text box. This text box is shown in Figure A.4.

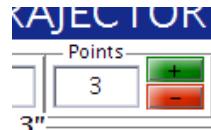


Figure A.4: TEOWholeGen: Number of points option

The final poses, final velocities, final accelerations, interpolation function and final time of each operational point are specified in their respective panels as in Figures A.5, A.6 and A.7.

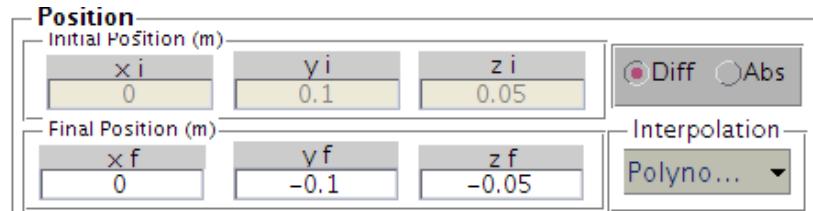


Figure A.5: TEOWholeGen: Final positions and interpolation panel

Once the previous values are specified, the trajectories in the operating space and their equivalent joint trajectories are generated in the panel of Figure A.8.

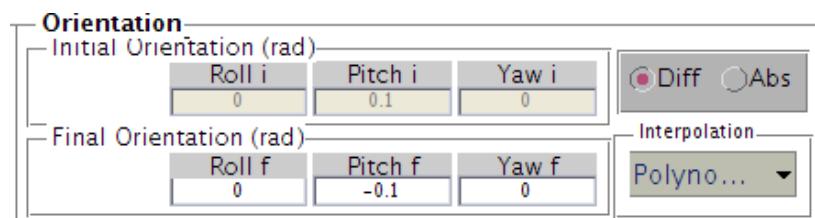


Figure A.6: TEOWholeGen: Final orientations and interpolation panel

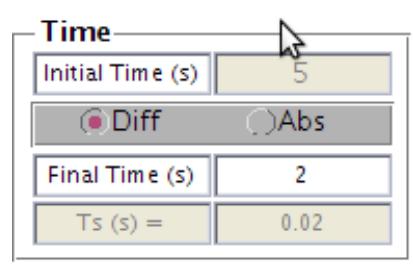


Figure A.7: TEOWholeGen: Final time panel

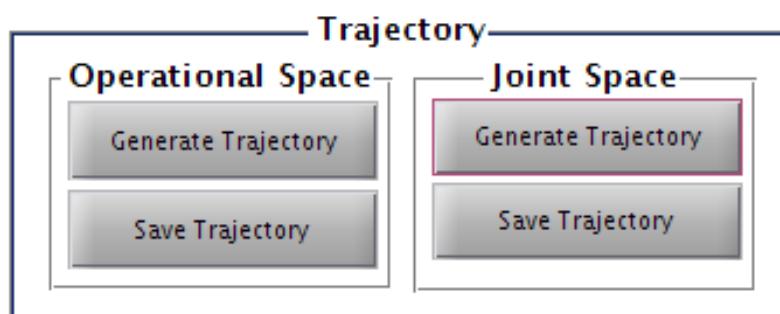


Figure A.8: TEOWholeGen: Trajectory generation panel

The resulting trajectories can be visualized in the operational space as Figure A.9 or in the joint space as Figure A.10.

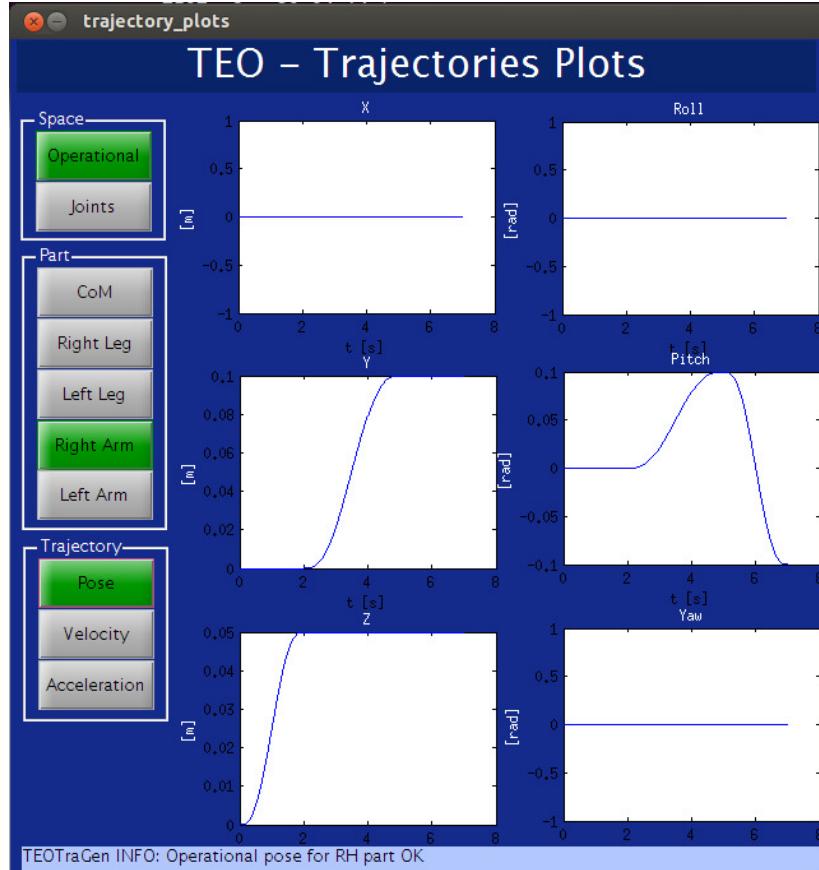


Figure A.9: TEOWholeGen: Visualization of the operational space trajectory in the Plots window

Additionally, the application allows the user to visualize the execution of the tasks with a TEO model modelled with the Robotics Toolbox for MATLAB®[2] as in Figure A.11.

The application it also allows the user to send the trajectories to ROS and use them with the appropriate nodes, e.g. using the RViz node to visualize them as in Figure A.12.

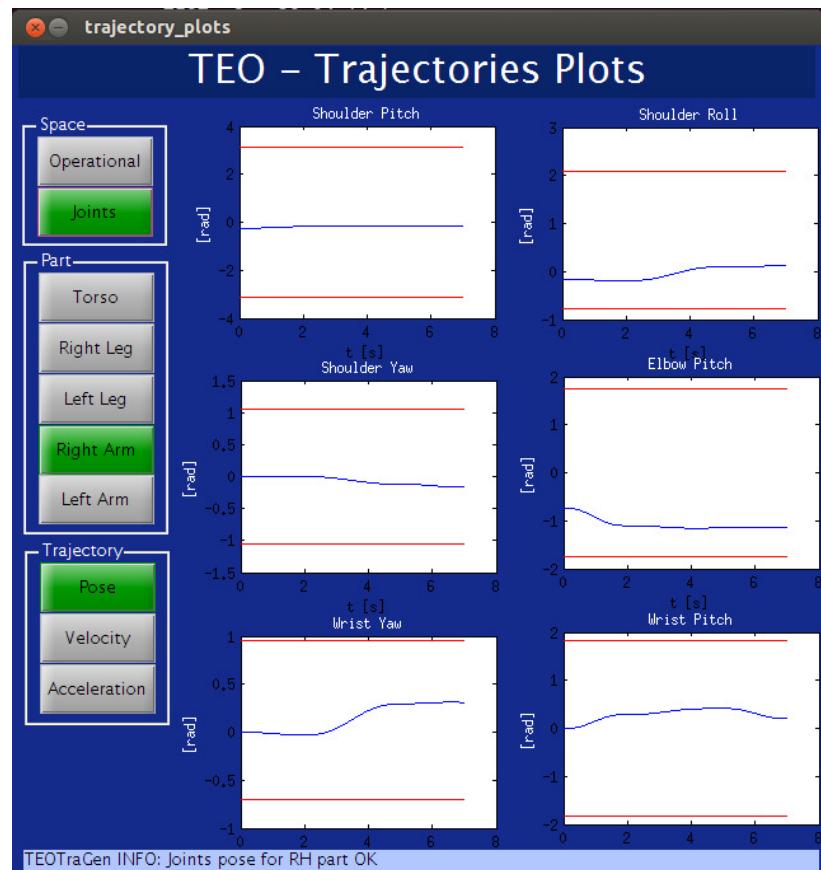


Figure A.10: TEOWholeGen: Visualization of the joint space trajectory in the Plots window

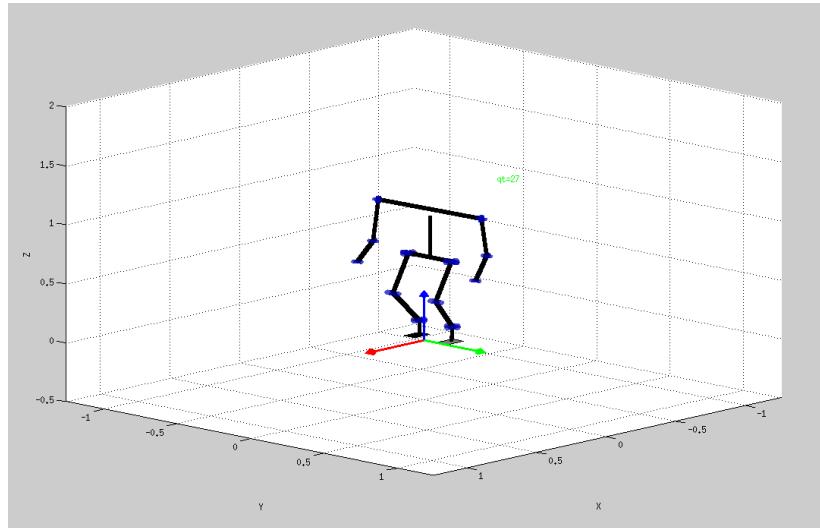


Figure A.11: TEOWholeGen: Visualization of TEO tasks with the Robotics Toolbox for MATLAB®

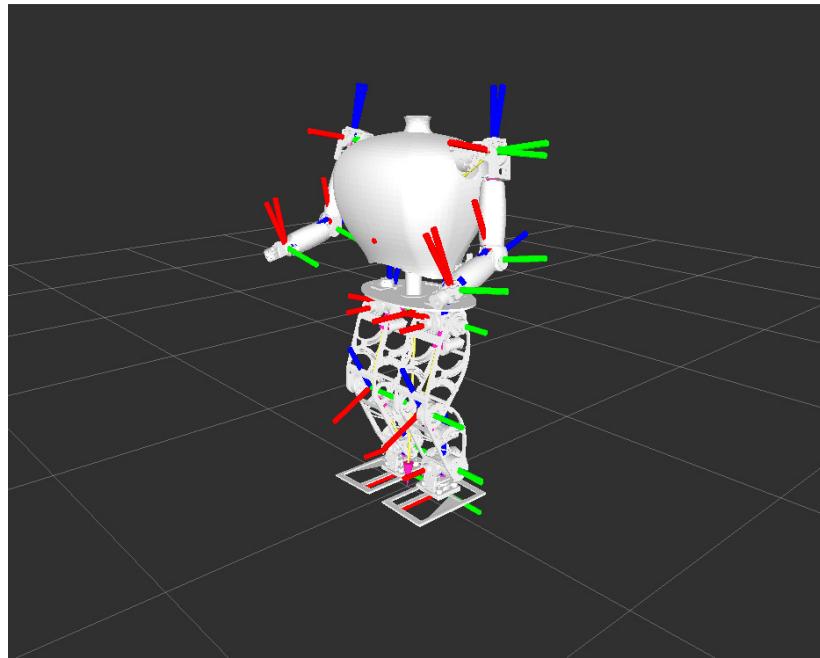


Figure A.12: TEOWholeGen: Visualization of TEO tasks in RViz

A.2 TEOStepGen

TEOStepGen is an application that allows the user to generate gaits for TEO humanoid. The main window of TEOStepGen is shown in Figure A.13 and provides access to the two gait generation options.

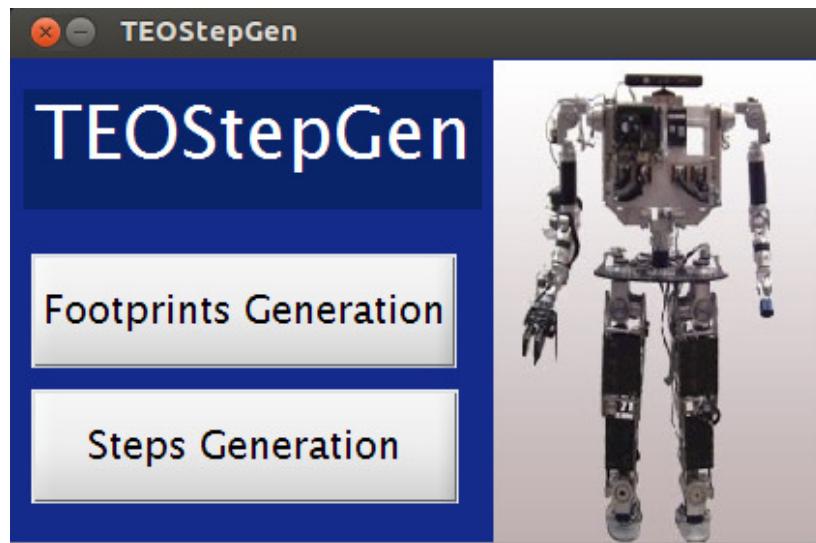


Figure A.13: *TEOStepGen main window*

The first option to generate a gait with TEOStepGen is through the footprints generation option that is shown in Figure A.14. This option allows the user to generate footprints with the local axis gait algorithm [1]. Then the user can generate the trajectories for the six operational points of TEO using the gait pattern generator proposed in 6.2.3.

To generate the footprints and the operational space trajectories, the user has to specify some parameters of the gait in the different panels. These panels are enumerated in Figure A.15. Considering this figure, the footprints are generated with the parameters introduced in Panel 1. The interpolation functions for the feet, FZMP, arms, and double support phase of the COG are introduced in Panel 2. The COG parameters and the interpolation function of the COG for the swing

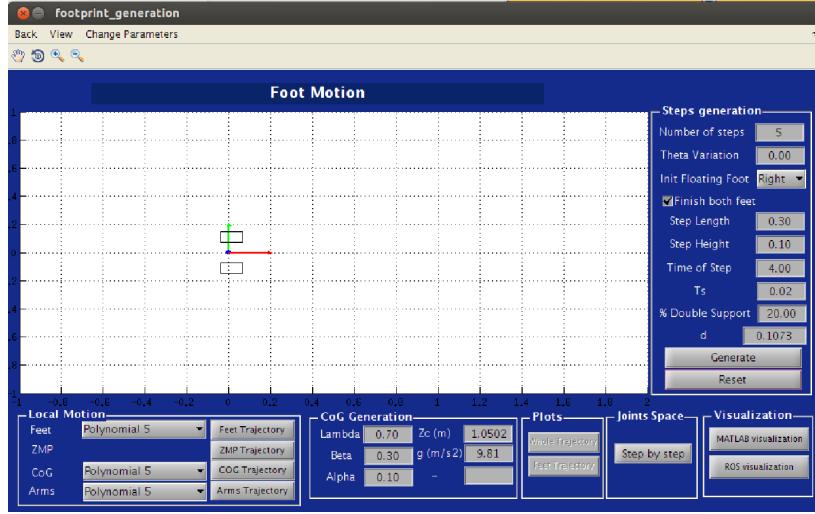


Figure A.14: TEOStepGen footprints generation main window

phase are introduced in Panel 3.

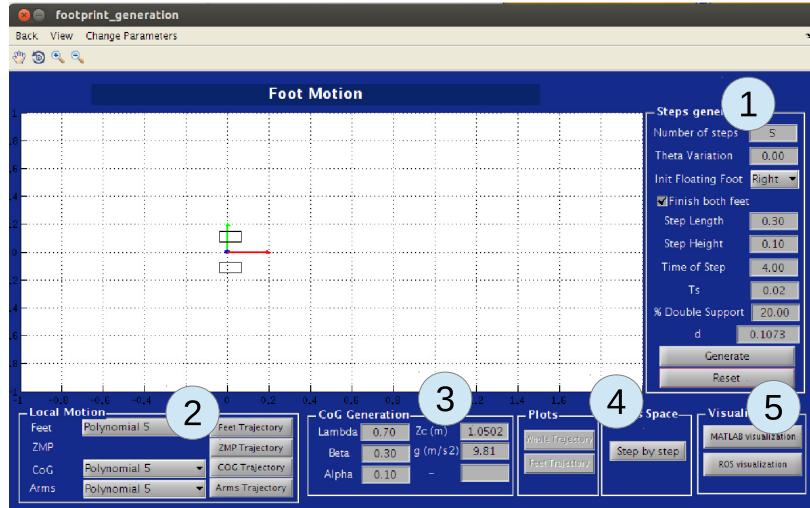


Figure A.15: TEOStepGen footprints generation panels

The Panel 4 allows the user to use the pattern generator of section 6.2.3 step by step. The trajectories generated are shown in Figures A.16 and A.17.

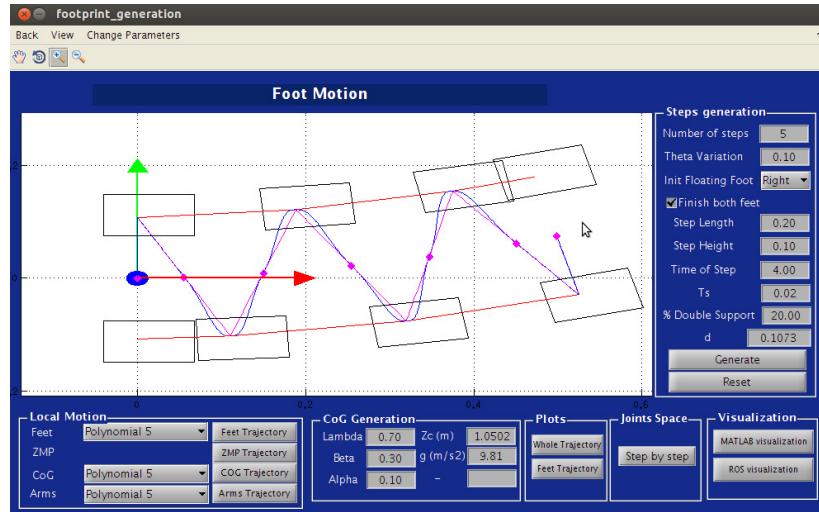


Figure A.16: TEOSTepGen: Visualization of the operational space trajectories

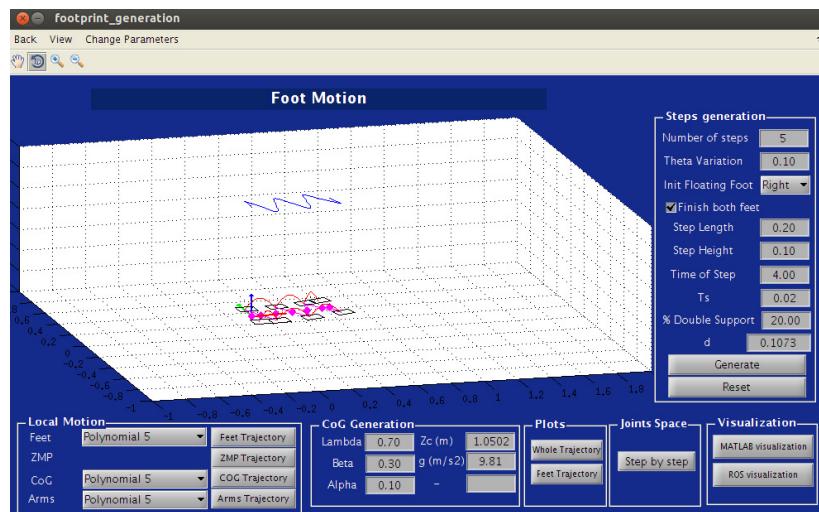


Figure A.17: TEOSTepGen: Visualization of the operational space trajectories

Finally, the Panel 5 allows the user to visualize the execution of the locomotion task with the TEO model modelled with the Robotics Toolbox for MATLAB®[2]. It also has a button to send the trajectories to ROS and use them with the appropriate nodes, e.g. using the RViz node to visualize them as in Figure A.18.

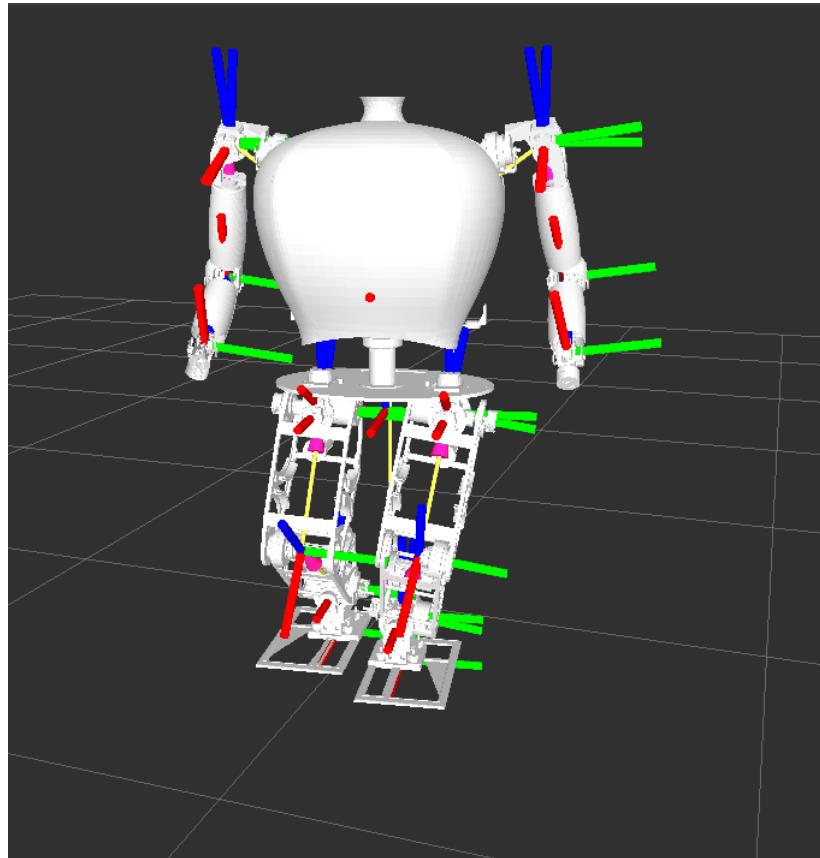


Figure A.18: TEOStepGen: Visualization of locomotion task of TEO in RViz

The second option to generate a gait with TEOStepGen is through the fast steps generation option. Inside this window, the user can specify a step through the CoM, swing foot and arms variation in the operational space. The first window that appears when the user access to the application is shown in Figure A.19. This window allows the user to select the parameters that the step generator will use.



Figure A.19: TEOSTepGen settings window

Once the configuration of a the step generator is defined, the user can define the gait by specifying the variations of the CoM, swing foot and arms in the operational space in the window of Figure A.20.

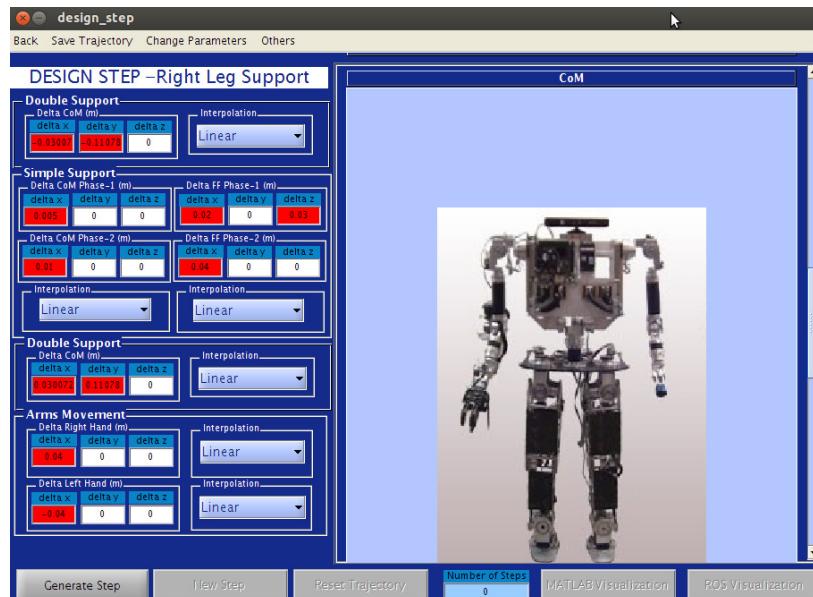


Figure A.20: TEOSTepGen: Fast steps generation window

Once the parameters are defined the ‘Generate step’ button generates the trajectory in the operational space and intermediately map it in the joint space. The trajectories of all the 26 joints of TEO can be visualized in the right panel as in Figure A.21.

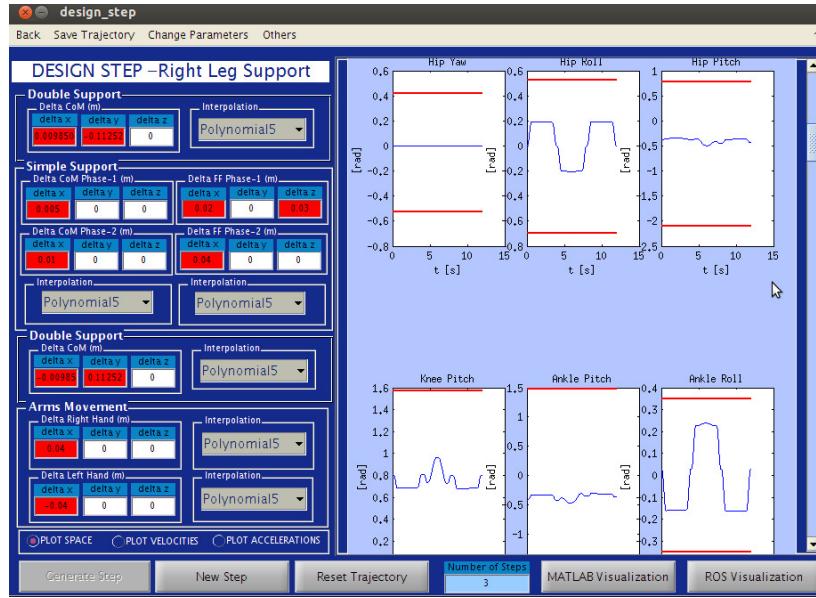


Figure A.21: TEOStepGen: Fast steps generation window

The resulting gait can also be visualized in the TEO model modelled with the Robotics Toolbox for MATLAB® and send it to the ROS environment as in the Foot print option.



TEO ROS packages

In order to test the algorithms and models proposed in this thesis and have an environment that allows the development of future research with TEO humanoid, some packages in ROS have been developed. Next, some of this packages are briefly detailed.

B.1 `teo_description`

The package `teo_description` contains the Unified Robot Description Format (URDF) files with the description (mechanical, kinematic, visual, etc.) of TEO humanoid. In accordance with two ROS Enhancement Proposals (REPs): REP-105 (Coordinate Frames for Mobile Platforms) and REP-120 (Coordinate Frames for Humanoid Robots), the root link is `base_link` that is directly connected to the waist¹.

The defined frames for the operational points x_{rh} and x_{lh} are 'l_gripper' and 'r_gripper' respectively. For the operational points x_{rf} and x_{lf} the frames are 'l_sole' and 'r_sole' respectively. The URDF graphically is shown in Figure B.1.

The URDF files are simplified with Xacro (XML Macros) language that allows the URDF files to be shorter and more readable, because the macros expand to larger

¹This robot point is the same than the base or free-floating frame in section 3.2

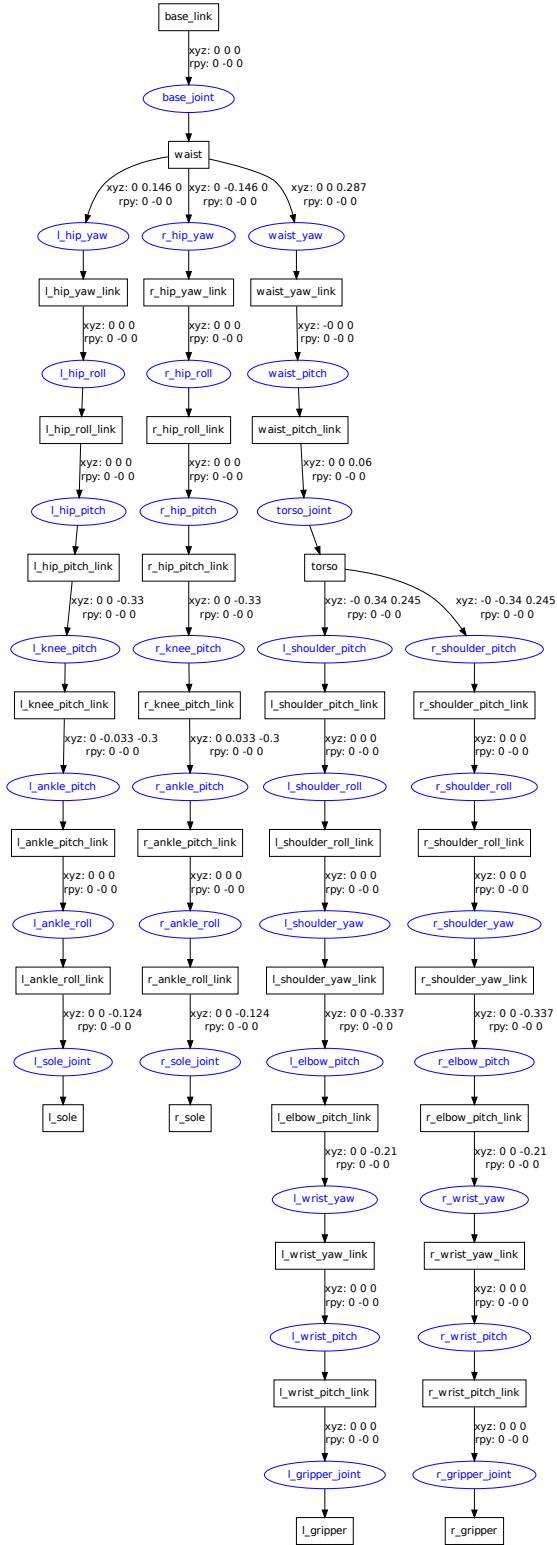


Figure B.1: *teo_description* ROS package: URDF structure of TEO

XML expressions.

There are two versions of TEO URDF: a simple version and a complete one. The main difference is that the first simplifies the visualization of TEO to simple geometric bodies as in Figure B.2. The second version, contains a more detailed visualization using STereoLithography (.STL) and Collada (.DAE) files as in Figure B.3.

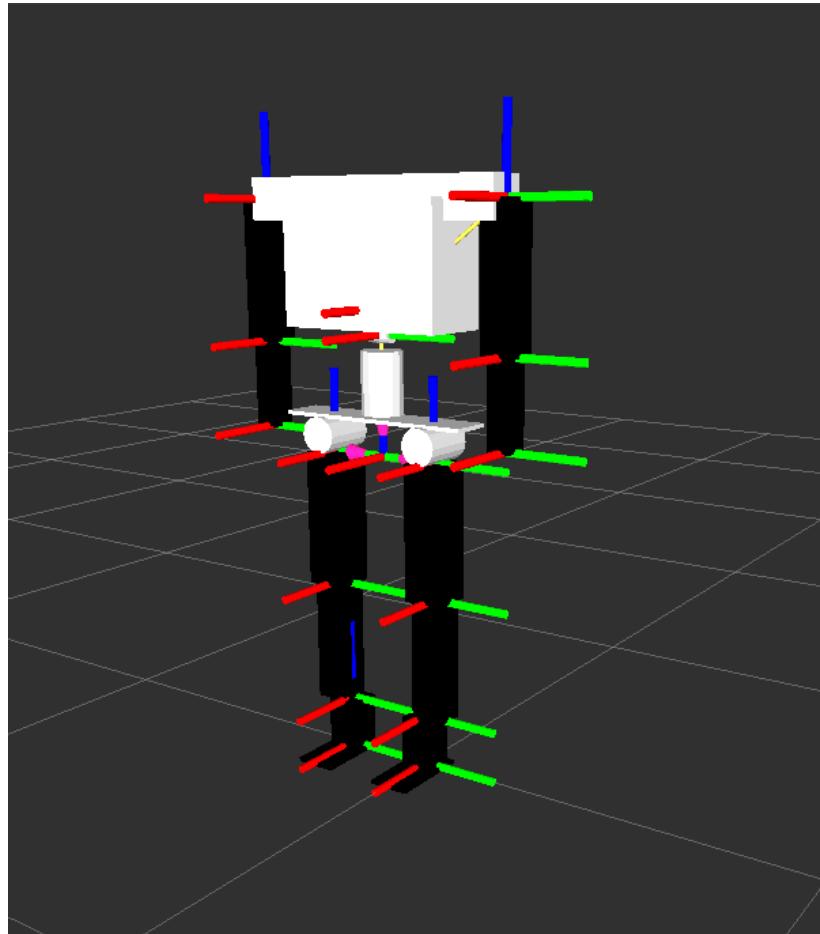


Figure B.2: `teo_description` ROS package: Simple TEO URDF model visualized in RViz

This package is organized into subdirectories as follows:

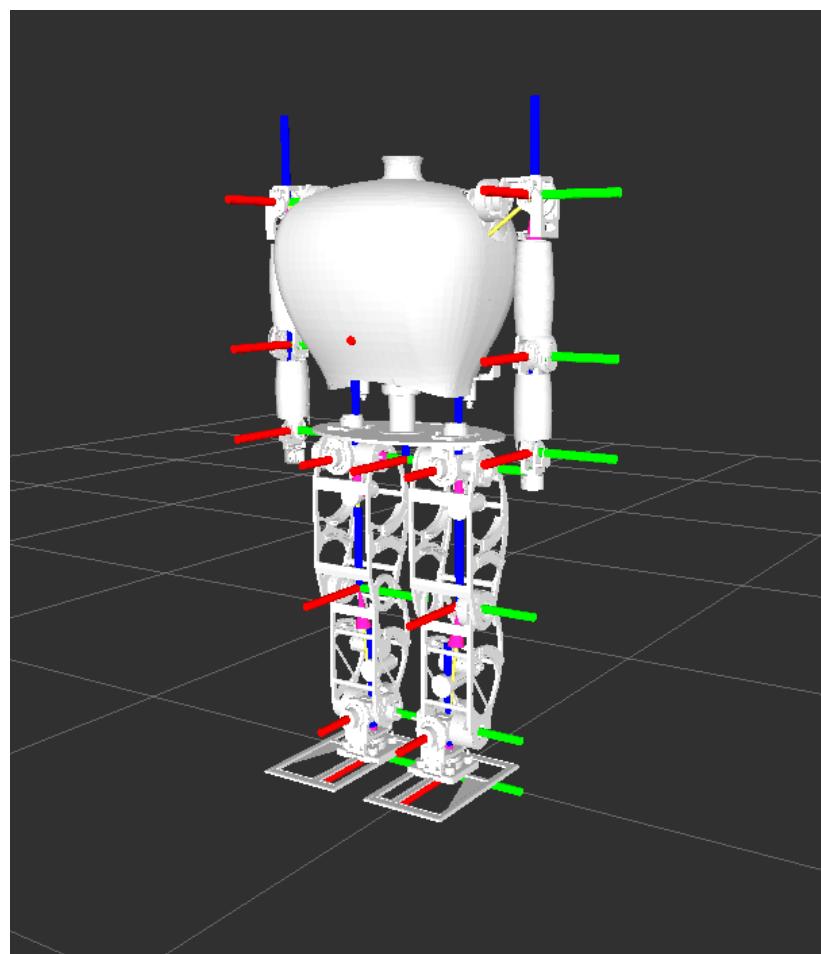


Figure B.3: *teo_description* ROS package: Complete TEO URDF model visualized in RViz

- **urdf**: contains (xacro representations of) urdf descriptions of various parts of TEO (leg, arm, torso, etc.).
- **robot** contains (xacro representations of) urdf descriptions of the full robot, that refer to the macros in urdf directory.
- **gazebo**: contains (xacro representations of) urdf descriptions of simulated TEO components.
- **meshes**: contains mesh files (.stl,.dae) for visualization and collision properties.

B.2 teo_remote

teo_remote contains the node that publishes 'base_link' transformation with respect to 'odom' frame and the 'base_footprint' frame that represents the robot position on the floor. This last frame is the barycenter of the feet projections on the floor with a yaw angle that corresponds to the 'base_link' yaw angle.

B.3 teo_messages

The package teo_messages contains the custom messages used in the implementation of TEO in ROS. The current messages in the package are: SupportFoot.msg and TEOConfiguration.msg.

The compact message definition of SupportFoot.msg is:

- int8 support_foot

And the compact message definition of TEOConfiguration.msg is:

- teo_msgs/SupportFoot support_foot
- sensor_msgs/JointState joints

B.4 teo_control

The teo_control package contains a configuration file and some launch files for the TEO controllers that interface with Gazebo. The package depends of ros_control package that brings the controller plugins used in the implementation of TEO in ROS.

The package is organized into subdirectories as follows:

- **config**: It contains a .yaml config file with the PID gains and controller settings of the 26 joints of TEO.
- **launch**: It contains the launch files to start the ros_control controllers.

Bibliography

- [1] M. Arbulu and C. Balaguer. Real-time gait planning for rh-1 humanoid robot, using local axis gait algorithm. In *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, pages 563–568, Nov 2007.
- [2] Peter Corke. *Robotics, vision and control: fundamental algorithms in MATLAB*, volume 73 of *Springer tracts in advanced robotics*. Springer, 2011.
- [3] Santiago Martínez de la Casa Díaz. *Human Inspired Humanoid Robot Control Architecture*. PhD thesis, Universidad Carlos III de Madrid, July 2012.
- [4] Maarten H. P. Dekker. Zero-moment point method for stable biped walking. Master’s thesis, Technische Universiteit Eindhoven, July 2009.
- [5] A. Goswami. Foot rotation indicator (fri) point: a new gait planning tool to evaluate postural stability of biped robots. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 1, pages 47–52 vol.1, 1999.
- [6] José Manuel Pardos Gotor. *Algoritmos de Geometría Diferencial para la Locomoción y Navegación Bípedas de Robots Humanoides Aplicación al robot RH0*. PhD thesis, Universidad Carlos III de Madrid, September 2005.
- [7] H. Hirukawa, S. Hattori, K. Harada, S. Kajita, K. Kaneko, F. Kanehiro, K. Fujiwara, and M. Morisawa. A universal stability criterion of the foot

- contact of legged robots - adios zmp. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 1976–1983, May 2006.
- [8] H. Hirukawa, S. Hattori, S. Kajita, K. Harada, K. Kaneko, F. Kanehiro, M. Morisawa, and S. Nakaoka. A pattern generator of humanoid robots walking on a rough terrain. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 2181–2187, April 2007.
- [9] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 2, pages 1620–1626 vol.2, Sept 2003.
- [10] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Yokoi, and H. Hirukawa. A realtime pattern generator for biped walking. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 1, pages 31–37 vol.1, 2002.
- [11] S. Kajita, O. Matsumoto, and M. Saigo. Real-time 3d walking pattern generation for a biped robot with telescopic legs. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 3, pages 2299–2306 vol.3, 2001.
- [12] James Jr. Kuffner, Satoshi Kagami, Koichi Nishikawi, Masayuki Inaba, and Hirochika Inoue. Dynamically-stable motion planning for humanoid robots. *Autonomous robots*, 12:105–118, 1 2002.
- [13] James Jr. Kuffner, Satoshi Kagami, Koichi Nishikawi, Masayuki Inaba, and Hirochika Inoue. Online footstep planning for humanoid robots. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 1, pages 932–937 vol.1, Sept 2003.

- [14] Idaho National Laboratory. Humanoid Robotics. What is a humanoid robot? https://inlportal.inl.gov/portal/server.pt/community/what_is_a_humanoid_robot_/539. Accessed: 2014-06-22.
- [15] Pin-Yong Ling, K.Y.-W. Chao, Huang Han-Pang, and Jiu-Lou Yan. Footprint searching and trajectory design of a humanoid robot. In *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on*, pages 259–264, Dec 2012.
- [16] T. McGeer. Passive walking with knees. In *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pages 1640–1645 vol.3, May 1990.
- [17] C.A. Monje, S. Martinez, A. Jardon, P. Pierro, C. Balaguer, and D. Munoz. Full-size humanoid robot teo: Design attending mechanical robustness and energy consumption. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages 325–330, Oct 2011.
- [18] Nicolas Perrin. *Footstep Planning for Humanoid Robots: Discrete and Continuous Approaches*. PhD thesis, LAAS-CNRS, October 2011.
- [19] P. Pierro, C.A. Monje, and C. Balaguer. The virtual com joints approach for whole-body rh-1 motion. In *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, pages 285–290, Sept 2009.
- [20] Paolo Pierro. *Stabilizer architecture for humanoid robots collaborating with humans*. PhD thesis, Universidad Carlos III de Madrid, May 2012.
- [21] Oscar Efrain Ramos Ponce. Generation of complex dynamic motion for a humanoid robot. Master’s thesis, LAAS-CNRS, 2011.
- [22] Mario Ricardo Arbulu Saavedra. *Stable locomotion of humanoid robots based on mass concentrated model*. PhD thesis, Universidad Carlos III de Madrid, October 2008.

-
- [23] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing. Springer, 2009.
 - [24] M. Vukobratovic, B. Borovac, and V. Potkonjak. Towards a unified understanding of basic notions and terms in humanoid robotics. *Robotica*, 25:87–101, 1 2007.
 - [25] Miomir Vukobratovic and Branislav Borovac. Zero-moment point - thirty five years of its life. *International Journal of Humanoid Robotics*, 1(1):157–173, 1 2004.
 - [26] P. B Wieber. Trajectory free linear model predictive control for stable walking in the presence of strong perturbations. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 137–142, Dec 2006.