

**POLITECNICO DI TORINO**

**MASTER's Degree in MECHATRONIC  
ENGINEERING**



**Politecnico  
di Torino**

**MASTER's Degree Thesis**

**Modelling of a biped kinematics, analysis  
and simulation of the human walk for a  
future lower-limb exoskeleton**

**Supervisors**

**Prof. Maurizio MORISIO**

**Prof. Giuseppe MENGA**

**Candidate**

**Gabriele CONGEDO**

**OCTOBER 2021**

# Abstract

Humanoid biped robots are typically complex in design, having numerous degrees-of-freedom (DOF) due to the ambitious goal of mimicking the human gait.

This thesis describes a walking control algorithm for the stable walking of a lower-limb exoskeletons of a fifteen DOF (six for each leg and three for the support point). First of all, the gait locomotion, the human foot and the zero moment point (ZMP) get studied from an anatomical and kinesiological point of view.

Input variables are imported from the big dataset of the University of Dallas. These are used for the computation of ZMP and, according to the Linear Inverted Pendulum Model (LIPM), the COG pattern is generated. Consequently, an estimation algorithm is used to fit the calculated trajectories and the joint angles going from a dataset to a kinematic model.

Considering the study of kinematic model, this work will be used for the interaction with the patient and it will take place by controlling the motors of the joints through electromyographic signals measured by the muscles of the legs and the torso.

Moreover, it will be possible to create an exoskeleton adding an online analysis of the EMG signals through deep-learning, where the patient can close the control loop through his/her movements.

# Summary

After a brief *introduction* there are two macro part. A *theoretical part* that performs an analysis of the gait kinematics aimed at the evaluation of main features of the walk and a *simulation part* that shows, after all, the human walk of the exoskeleton on a software.

## Theoretical Part

- The *First Chapter* is an introduction to the history of the walking robots where is explained the periods of innovation of the wearable device for the rehabilitation.
- The *Second Chapter* which provide the study of gait biomechanics and kinematics, by introducing the analysis of the human foot, providing the fundamentals for the design of a kinematic model and finally showing an overview of the stability criteria that talks about ZMP and LIPM.
- The *Third Chapter* shows the instrumentation and the best option for the evaluation.
- The *Fourth Chapter* provides the creation, step by step, of each element of the exoskeleton through SketchUp and OnShape.

## Simulation Part

- The *Fifth Chapter* describes the experimental datasets used in the various tests.
- The *Sixth Chapter* offers a report on analysis of the experimental information, and explains the processes of generating ZMP, foot and CoM trajectories from data processing, according to the LIPM model. Then is illustrated the model that simulated the human walk with the trajectiories generated above.
- The *Seventh Chapter* shows another simulation of a different model, explained in Chapter 3, with flat-foot on CoppeliaSim.

The research has been realised cooperatively with my colleague Michela Rosati. In particular, the study of *Chapter 1*, *Chapter 2*, *Chapter 3* was done together; the study of *Chapter 5* and *Chapter 6* was done by me; the study of *Chapter 4* and *Chapter 7* was done by my colleague.

The report concludes with a chapter in which the results of the experiments are reviewed and are explained the future works.

# Acknowledgements

I would like to extend my sincere thanks to my primary supervisor, *Prof. Maurizio Morisio*, who helping me finalize the project and I wish to show my appreciation to *Prof. Giuseppe Menga* for their assistance at every stage of the research project.

My heartfelt thanks to *Michela Rosati* for his contribution. We worked together to finalize this research.

My gratitude goes also to my *friends* for their unwavering support and belief in me.

I am deeply grateful to *my family* for their love and encouragement and last, but certainly not least, to my girlfriend, *Daniela*, who's love and support has seen me through my degree's many trials and tribulations.

*"Dream on until your dreams come true"*  
*Aerosmith*



# Table of Contents

<b>Abstract</b>	I
<b>List of Tables</b>	IX
<b>List of Figures</b>	X
<b>Acronyms</b>	XIV
<b>Introduction</b>	2
0.1 Objectives . . . . .	2
<b>I Theoretical Part</b>	<b>5</b>
<b>1 History of walking robots</b>	<b>6</b>
1.1 Walking Robots . . . . .	6
1.2 Development of Exoskeletons . . . . .	7
1.2.1 Periods of innovation . . . . .	8
<b>2 Kinesiology of the Human walk</b>	<b>14</b>
2.1 Analysis of the Human Foot . . . . .	14
2.2 Kinematics . . . . .	16
2.2.1 Anthropomorphic human dimensions, volume and weight distribution . . . . .	16
2.2.2 Reference systems conventions . . . . .	17
2.3 Gait Cycle . . . . .	20
2.3.1 Cycle Divisions . . . . .	20
2.3.2 Stride and Step . . . . .	23
2.4 Defining the Measure of Balance . . . . .	24
2.4.1 Zero Moment Point . . . . .	24
2.5 Model for biped robots . . . . .	29
2.5.1 Linear Inverted Pendulum Model (LIPM) . . . . .	29

2.5.2	3D Linear Inverted Pendulum Model (3D-LIPM)	32
<b>3</b>	<b>Instrumentation</b>	<b>37</b>
3.1	Non-wearable sensors	37
3.1.1	Force transducers and force plates	37
3.1.2	Treadmills	38
3.2	Wearable sensors	39
3.2.1	Optoelectronic stereophotogrammetry	39
3.2.2	Accelerometer	40
3.2.3	Gyroscope	40
3.2.4	Magnetometer	41
3.2.5	Inertial Measurement Units	42
3.2.6	Electrogoniometers	43
3.2.7	Optical fiber sensors	43
3.2.8	Textile-based sensors	43
3.2.9	EMG signals	45
<b>4</b>	<b>Biped creation</b>	<b>47</b>
4.1	Biped design	47
4.1.1	Biped parts	48
<b>II</b>	<b>Simulation Part</b>	<b>53</b>
<b>5</b>	<b>Incline Experiment</b>	<b>54</b>
<b>6</b>	<b>Simulation on Matlab</b>	<b>60</b>
6.1	Cartesian variables generation	63
6.1.1	ZMP and feet trajectories generation	64
6.1.2	CoG computation	67
6.2	From variables to the model on the kinematic cycle	70
6.2.1	Direct Kinematics	71
6.3	Differential inverse kinematics	73
6.4	Results	77
<b>7</b>	<b>Simulation on CoppeliaSim</b>	<b>82</b>
7.1	CoppeliaSim	82
7.1.1	Design simulation	82
7.1.2	Walking analysis	85
7.1.3	Angle and position graphs	86
7.1.4	Code	92

<b>Conclusion Future Works</b>	98
<b>Bibliography</b>	99

# List of Tables

2.1	Floor Contact Periods . . . . .	22
6.1	Chosen center offsets for the subject AB01 and trial s1i0 . . . . .	63
6.2	Time-averaged extension of the body segments . . . . .	64

# List of Figures

1.1	ASIMO developed by Honda . . . . .	7
1.2	MAHRU developed by Samsung Electronics . . . . .	7
1.3	QRIO developed by Sony . . . . .	8
1.4	WABIAN developed by Waseda Univerisity . . . . .	8
1.5	by Robert Seymour . . . . .	9
1.6	by Ira C.C. Rinehart . . . . .	9
1.7	Hardiman by General Electric . . . . .	10
1.8	by Tsinghua University . . . . .	11
1.9	SpringWalker . . . . .	11
1.10	by Kanagawa Institute of Technology . . . . .	12
1.11	HAL-1 of University of Tsukuba . . . . .	12
1.12	HAL- 5 . . . . .	13
2.1	Arch-type skeletal structure of human foot . . . . .	15
2.2	Transient pressure distribution on sole during a single stride (standard male) . . . . .	16
2.3	Body segment lengths expressed as a fraction of body height . . . . .	17
2.4	Location of mass centers of body segment . . . . .	17
2.5	Human motion planes ©NASA . . . . .	18
2.6	Division of the gait cycle . . . . .	21
2.7	A step versus a stride . . . . .	23
2.8	Support foot and influence of by the force, moment, ground reaction	25
2.9	The use of center of gravity as a measure of balance is only acceptable when the motions are slow and the dynamic forces are negligible. . . . .	26
2.10	Foot of the supporting in the single-support phase . . . . .	27
2.11	Rotation of the supporting foot about its edge. . . . .	28
2.12	Projection of the ZMP for a constant step walk on the Sagittal, Frontal and ground planes . . . . .	29
2.13	Linear Inverted Pendulum Model . . . . .	31
2.14	3D pendulum . . . . .	33
2.15	3D Linear Inverted Pendulum Model . . . . .	35

3.1	Scheme of a piezoelectrical force plate . . . . .	38
3.2	Example of a modern system for gait analysis with the use of a treadmill . . . . .	39
3.3	Stereophotogrammetry cameras system . . . . .	40
3.4	Scheme of a capacitive accelerometer . . . . .	41
3.5	Scheme of a piezoresistive accelerometer . . . . .	41
3.6	MEMS gyroscope scheme . . . . .	42
3.7	Inertial Measurement Units system . . . . .	44
3.8	A potentiometric electrogoniometer . . . . .	44
3.9	Optoelectronic system positioned on a patient . . . . .	44
3.10	Scheme of an optical fiber sensor . . . . .	45
3.11	Scheme of wearable wire sensor . . . . .	45
3.12	Scheme of an instrumentation amplifier . . . . .	45
3.13	Example of a possible placement of electrodes on patient bodies . .	46
4.1	Biped design . . . . .	48
4.2	Trunk design . . . . .	48
4.3	Right hip design . . . . .	49
4.4	Right up leg and knee joint design . . . . .	49
4.5	Right low leg and ankle joint(Y) design . . . . .	50
4.6	Right foot and ankle joint(X) design . . . . .	50
4.7	Prototype created on <i>Onshape</i> . . . . .	51
5.1	Continuous struct Data Hierarchy . . . . .	58
5.2	Gaitcycle struct Data Hierarchy . . . . .	59
6.1	State machine representation . . . . .	62
6.2	Comparison between computed ZMP footprints and meta trajectories	66
6.3	Comparison between computed ZMP and the CoP along x . . . . .	66
6.4	Comparison between computed ZMP and the CoP along y . . . . .	67
6.5	Comparison of the reference and tracked ZMP with the generated CoG along x,y . . . . .	71
6.6	Comparison between CoG and ZMP on the xy ground plane . . . . .	72
6.7	Biped walker animation . . . . .	79
6.8	CoM trajectory along x-axis,y-axis,z-axis . . . . .	79
6.9	Feet trajectories along x-axis,y-axis,z-axis . . . . .	80
6.10	Trunk angles around x-axis,y-axis,z-axis . . . . .	80
6.11	Left and right knee joints angles . . . . .	81
7.1	Dummy target . . . . .	83
7.2	Dummy tip . . . . .	83
7.3	Hierarchy of the Biped . . . . .	84

7.4	Phase 1 . . . . .	85
7.5	Phase 2 . . . . .	85
7.6	Phase 3 . . . . .	85
7.7	Matlab Feet Trajectory . . . . .	86
7.8	Trunk X and Y position . . . . .	87
7.9	Trunk Trajectory Matlab . . . . .	87
7.10	Left and right hip joint angles (Yaw) . . . . .	88
7.11	Left and right hip joint angles (Roll) . . . . .	88
7.12	Left and right hip joint angles (Pitch) . . . . .	89
7.13	Left and right knee joint angles (Pitch) . . . . .	89
7.14	Left and right ankle joint angles (Pitch) . . . . .	90
7.15	Left and right ankle joint angles (Roll) . . . . .	90
7.16	Feet Z position . . . . .	91



# Acronyms

**ZMP**

Zero Moment Point

**COG**

Center of Gravity

**CoM**

Center of Mass

**LIPM**

Linear Inverted Pendulum Model

**W.R.T.**

With Reference to

**MEMS**

Microelectromechanical systems

**MIMU**

Magneto-Inertial Measurement Units

**OFS**

Optical fiber sensors

**WBC**

Whole Body Conditions



# Introduction

## 0.1 Objectives

The objectives of this research develop:

1. **A study of the kinematics of humanoid robots.**
2. **A study of gait locomotion.**
3. **A study of gait control algorithm.**
4. **Stable walking patterns for humanoid robots.**

### 1. *A study of the kinematics of humanoid robots*

To introduce humanoid robot mobility, many aspects should be taken into account. Some of them do not include the field of gravity or forces of inertia. These aspects are related to the number of joints and joint motion ranges for reaching any local or global goal. Furthermore, the motion planning is constrained to physical velocity limits of the actuators, so the suitable kinematics model must be developed. This model computes the joint pattern references (position and velocity) of the control system.

### 2. *A study of gait locomotion*

The word "gait" has various meanings, for example (as selected from a dictionary): "A particular way or manner of moving on foot".

It is clear that its meaning is suitable for studying humanoid robot gait locomotion. It is been confirmed that the human gait develops a stable walking pattern. The passive walkers make a walking motion under the field of gravity and the swing foot like the human one. The main characteristic of the swing foot motion is that it falls down from the effect of gravity.

Otherwise, gait stability is controlled by center of gravity (COG) motion. Studies of biomechanics demonstrates that, when the human walks, the human center of gravity follows the laws of motion of the inverted pendulum. That way stable gait is obtained, because the COG is concentrated near the support foot during the swing

motion, and the COG quickly changes its position during the change of support foot.

### 3. *A study of gait control algorithm*

The gait control algorithms proposed by a few researchers, after about fifteen years of research, such as by HONDA Mo. Co. Ltd., are the reference for the next humanoid robot prototypes. This kind of research introduces an algorithm based on human gait control. This research proposed this control loop: the Zero Moment Point (ZMP) control.

A method of generating a highly stable, smooth walking pattern is presented. By combining the proposed offline walking pattern with real-time modification, the biped robot can walk smoothly and adapt to unknown environments.

### 4. *Stable walking patterns for humanoid robots*

Many walking pattern generation methods are studied in order to obtain stable humanoid robot walking motion. When it is not possible to be applied in real time some of them deal with the distributed mass model due to the high complexity of the multibody dynamics [1], [2]. Other methods deal with the mass concentrated models, which could be applied in real time because the whole body dynamics are simplified to one mass motion in space under the field of gravity. The mass concentrated models for generating dynamics and stable walking pattern of humanoid robots are selected. They are: "Three dimensional Inverted Pendulum Model, (3D-LIPM)" [3]. These models are successfully tested on the Rh-1 humanoid robot platform. Otherwise, the new walking pattern generation method in order to generate an "acyclic gait" is proposed and successfully tested in the HRP-2 humanoid robot platform, currently the most advanced humanoid robot platform.



**Part I**

**Theoretical Part**

# Chapter 1

## History of walking robots

### 1.1 Walking Robots

In recent years, several efforts of the robotics community have focused on developing bio-inspired robots, particularly in humanoid biped robots. Many studies on biped walking robots have been performed since 1970. During that period, biped walking robots have transformed into biped humanoid robots through the technological development.

Furthermore, the biped humanoid robot has become a one of representative research topics in the intelligent robot research society. Many researchers anticipate that the humanoid robot industry will be the industry leader of the 21st century and we eventually enter an era of one robot in every home. The strong focus on biped humanoid robots stems from a long-standing desire for human-like robot that is desirable for coexistence in a human-robot society [4].

At present, biped humanoid robot research groups developed their own robot platform and dynamic walking control algorithms. This has resulted in promising developments such as the ASIMO, Figure [1.1], (the acronym of "Advanced Step In Mobility") humanoid robot, developed by HONDA with 32 DOF and 52 kg weight or the 62 kg MAHRU, Figure [1.2], series of Samsung Electronics with 32 DOF. Further examples of contributions in the humanoid robots field are the QRIO, Figure [1.3], which has an adaptable motion controller that allows the displacements on uneven surfaces and external forces and the WABIAN, Figure [1.4], series of Waseda University with 35 DOF, which has played a fundamental role in the evolution of humanoid robots. Some of the most studied architectures found in the literature include the H7 from the University of Tokyo with a total of 30 DOF, including a one DOF toe joint. Another example is the JOHNNIE robot from the Technical University of Munich, where each leg incorporates six driven joints, three DOF in the hip, one DOF for the knee, and another two DOF for

the ankle joint. Nowadays, most of the humanoid robots mentioned above consist of two 6-DOF legs, namely 3-DOF hip, 1-DOF knee and 2-DOF ankle. Although the efforts have mainly focused on achieving human gait, this feature has not been successful accomplished with a limited number of DOF. Thus, it becomes necessary to incorporate redundant DOF in order to achieve an approximate human gait motion.

The addition of an active toe joint to the kinematic model of each leg has drawn enormous interest within the robotics scholars, because compared with conventional humanoid robots that are equipped with flat feet, this architecture allows a robot to walk in a more natural way. There are several references in the literature related to toe joints in biped robots [5].



**Figure 1.1:** ASIMO developed by Honda



**Figure 1.2:** MAHRU developed by Samsung Electronics

## 1.2 Development of Exoskeletons

The term ‘exoskeleton’ was used in biology referring to the chitinous or calcified external skeleton used by numerous animal taxa for structural support and defense against predators. Now, the exoskeletons are generally regarded as a technology that extends, complements, substitutes or enhances human function and capability or empowers the human limb where it is worn. Different from other robots, the operator of an exoskeletons is human who need to make decisions and perform tasks with exoskeletons. Through combining human intelligence and machine power exoskeletons enhance the abilities of both human power and machine intelligence. Since the concept of exoskeleton was produced in the 19th century, the development of exoskeletons have undergone five phases, i.e. sprout period, exploration period, dormancy period, accumulation period and climax period. Exoskeletons apply and



**Figure 1.3:** QRIO developed by Sony



**Figure 1.4:** WABIAN developed by Waseda University

merge manifold techniques involving mechanical and electronic engineering, automation technology, biological, medical, and material science. Recently, exoskeletons are applied in military, civilian and rehabilitation. For military purposes, the exoskeletons are designed to augment the travel and loading abilities of soldiers; for civil applications, the exoskeletons are used to increase an operator's load-handling capabilities and reduce the occurrence of musculoskeletal disorders, or for rescue; for rehabilitation, exoskeletons are aiming at improving the lost functions and the quality of life of patients with severe or degenerative disabilities, motor cognitive limitation.

## 1.2.1 Periods of innovation

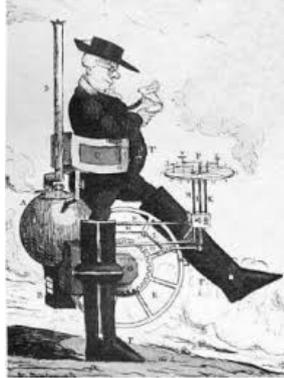
### Sprout period

The sprout period lasted more than one century from 1830 to 1960. During this period, a British inventor Robert Seymour proposed the concept to help people walk by a wearable device which was propelled by steam in 1830 [1.5].

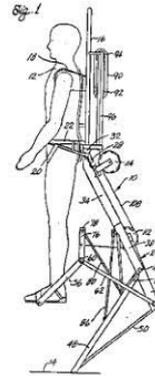
An American inventor Ira C.C. Rinehart conceptually designed a walking machine which enabled an individual to step seven feet and four inches at an ordinary stride in 1889 [1.6].

From 1889 to 1890, Nicholas Yagn, of St.Petersburg, Russia, designed a walking, jumping, and running assisted device using a giant leaf spring. In 1890, another inventor Yagn designed an exoskeleton with long leaf springs in parallel to the legs

to help people run faster and jump higher. In stance phase, the weight of body can be transferred to the ground directly by the spring to reduce the forces on the standing leg. Most exoskeletons were conceptual design in sprout period due to the limitations of the technology at that time.



**Figure 1.5:** by Robert Seymour



**Figure 1.6:** by Ira C.C. Rinehart

### Exploration period

The exoskeleton HARDIMAN developed by the US Department and General Electric in 1965 marked the exoskeleton development entering the exploration period. HARDIMAN aimed at augmentation that the individual who worn it could lift 1,500 lbs (682kg)[1.7]. In fact, only one arm of HARDIMAN was developed and achieved to lift 750lbs (341kg) until the 1970s. The failure of HARDIMAN was mainly caused by which the energy supplies were too huge to be portable, and the speed of data processing and function control was slow.

In the late 1960s and 1970s, an active anthropomorphic exoskeletons with pneumatic power and partly kinematical program for paraplegics was developed at the Mihailo Pupin Institute under the Prof. Vukobratovic's guidance. At the same time, the theory of legged locomotion systems was first put forward by Prof. Vukobratovic, which established foundation for present modern high-performance exoskeletons.

The researchers of University of Wisconsin started to develop a full lower limb exoskeleton in 1968. This exoskeleton was designed to help those paraplegics with complete upper limb capabilities to walk again. The wearer can implement the sit-to-stand, stand-to-sit translation and walk at 50% of normal speed. The hip and ankle joint both had three rotational degrees of freedom (DOF) and the knee joint had one rotational DOF. The joints at hip and knee for flexion/extension

were actuated by hydraulic power, and the other joints were passive. Although this exoskeleton was developed for paraplegics, there was not any report about the relevant tests.



**Figure 1.7:** Hardiman by General Electric

### **Dormancy period**

The development of exoskeleton entered the dormancy period in the 1980s. In the middle 1980s, the exoskeleton concept “Pitman” was put forward by Jeffrey Moore at the Los Alamos National Laboratory (Los Alamos, NM) to apply in military to augment the soldiers’ capabilities. However, this exoskeleton program was not funded by the U.S. Defense Advanced Research Projects Agency (DARPA).

In 1988, Prof. Jichuan Zhang started to research the electric walking machine for high leg paraplegia patients at Tsinghua University [1.8]. Using bar linkage mechanism, the ipsilateral hip joint and knee joint of the exoskeleton were actuated by only one motor. This structure decreased the weight of the exoskeleton and became more compact and portable.

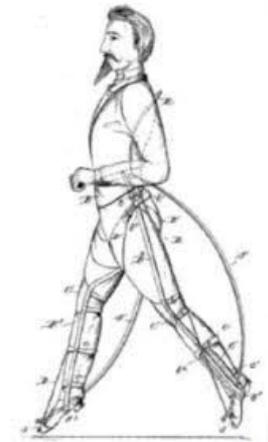
In 1990, G. John Dick and Eric A. Edwards developed SpringWalker according to the mechanism that a device in series with the human leg can reduce the metabolic cost of running by lowering impact losses and by providing energy return [1.9]. However, SpringWalker can only enhance jumping height. For running, it even increased metabolic cost by 20% compared to locomotion without it.

### **Accumulation period**

From 1990 to 2000, the research of exoskeleton went into the accumulation period. In 1992, Prof. Yoshiyuki Sankai of University of Tsukuba started to develop a



**Figure 1.8:** by Tsinghua University



**Figure 1.9:** SpringWalker

wearable type robot ‘Robot Suit HAL’ (Hybrid Assistive Limb), which was intended to physically support a wearer’s daily activities and heavy work. The first prototype named HAL-1 adopted DC motors and ball screws to augment the wearer’s joint torque [1.11].

In 1994, researchers of Kanagawa Institute of Technology developed a wearable power assisting suit for nurses to enhance their muscle strength to lift patients and avoid back injuries [1.10]. The movement of the joints at arms, waist and legs of the suit were sensed by strain sensors to detect the muscle force and actuated by pneumatic rotary actuators with concentric round boxes sliding each other. Compared to the over-ground exoskeletons, Hocoma AG developed an immobile exoskeleton Lokomat consisting of an over-ground exoskeleton, an advanced body weight support system (BWS) and a treadmill in 2000 at Switzerland. The Lokomat with repetitive walking on one hand help to improve circulation, strengthen bones and muscles and gain a natural walking pattern, on the other hand decrease the physical effort and constraint of the therapists.

### Climax period

Exoskeletons attracted much more attention of researchers from different countries including US, Japan, Israel, France, Switzerland, South Korea, China, etc. and the development of exoskeletons went into the climax period since 2000.

One representative of exoskeletons applied in military, Berkeley Lower Extremity Exoskeleton (BLEEX) was developed to increase soldier’s load capacity, lessen the



**Figure 1.10:** by Kanagawa Institute of Technology



**Figure 1.11:** HAL-1 of University of Tsukuba

risk of leg and back injury, decrease the metabolic consumption and reduce the perceived level of difficulty. BLEEX adopted the hybrid hydraulic-electric portable power supply in order to achieve carrying its own power source. The hip and ankle joint of BLEEX had three DOFs, respectively, among which hip flexion/extension, abduction/adduction and ankle flexion/extension were actuated by linear hydraulic actuators. Its knee joint had one DOF actuated for flexion/extension. The control system of BLEEX mainly collected sensory information from exoskeletons to determine the kinematic and dynamic parameters. It was reported that the soldier who wore BLEEX can walk at 0.9 m/s with load up to 75 kg and 1.3m/s without load.

The representative civil application of exoskeletons was the Robot Suit Hybrid Assistive Limb (HAL)-5 developed by Professor Yoshiyuki Sankai at University of Tsukuba for both power augmentation and walk assistance. The hip and ankle joint of HAL-5 were actuated by a DC motor with harmonic drive for flexion/extension, respectively, and the ankle joint for flexion/extension DOF was passive with springs to return a normal angle. HAL-5 adopted joint torque augmentation at the hip, knee and ankle joint, which is different from BLEEX transferring the load to ground. HAL-5 had two types of control systems: “Cybernic Voluntary Control System” and “Cybernic Autonomous Control System”. Cybernic Voluntary Control System understood the wearer’s voluntary intention according to the surface electromyographic (sEMG) signals through placing the sEMG electrodes below the hip and above the knee. Then the power units of HAL-5 generated power assist torque by amplifying the wearer’s joint torque estimated from sEMG signals. Cybernic Autonomous Control System was developed to provide effective physical supports for the handicaps by the potentiometers, ground reaction force sensors, a gyroscope and accelerometer on the backpack to estimate the posture since the signals of handicaps could cause a broken walking pattern.

ReWalk from Argo Medical Technologies has been commercialized for fundamentally changing the health and life experiences of individuals with spinal cord injuries (SCI). It consisted of a wearable brace support suit with DC motors at hip and knee joint, respectively, rechargeable batteries, a computer-based controller contained in a backpack, a wireless mode selector, and an array of sensors that measure upper-body tilt angle, joint angles, and ground contact. ReWalk utilized a closed-loop algorithm software control and triggered and maintained the walking pattern by detecting the wearer's upper-body movements. Additionally, ReWalk can also help the wearer climb stairs, transform from sitting to standing and vice versa. The crutches were necessary to keep balance [6].



**Figure 1.12:** HAL- 5

## Chapter 2

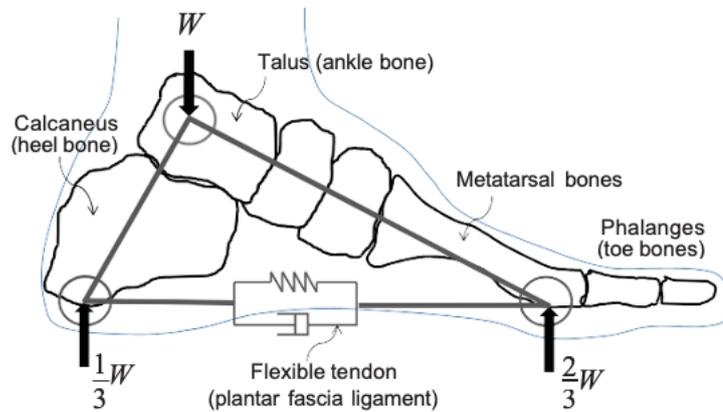
# Kinesiology of the Human walk

### 2.1 Analysis of the Human Foot

In human walking motion, the two feet play a crucial role in absorbing impact from the ground, maintaining stability on uneven terrains, and helping turning motions. In the same way, the critical design consideration with the foot of a bipedal robot is to enhance walking stability and performance. The study of the robotic foot is purposed to implement a human-like walking motion for biped robots by applying the characteristics of the human foot to the mechanism design. There has been a recent trend in foot design of adopting simple toe and heel joints to follow human ankle motion in landing on and taking off from the ground. First of all, it was confirmed that the flexible foot with toe joints enables increased walking speed and step length and much reduces energy consumption compared with flat feet. Along with the mechanical design of the flexible foot, a well-planned walking pattern is required in order to achieve human-like motion. It has also been found that the foot trajectory pattern with heel-contact and toe-off motion produces a smoother hip trajectory and increases adaptability to rough terrains.

These works suggest that adopting toes and heels in the design of the foot mechanism is of great benefit in enhancing the performance and stability of bipedal robots. However, most of these approaches are not matured yet and are lacking in analytical considerations for determining design parameters. Hence, in terms of an anatomical and kinesiological analysis of the human foot, this section will investigate how to determine some critical foot parameters from the point of view of walking stability. In biomechanics, the functional efficiency of the human foot mechanism to support weight and absorb ground impact has been well investigated. As shown in Figure [2.1] the human foot has an arch-type skeletal structure which connects heel, toes,

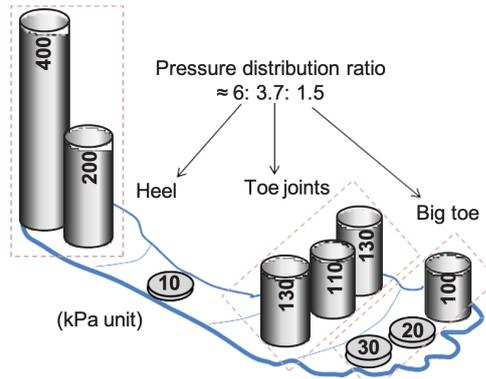
and ankle, where the large bone at the heel called calcaneus supports about one third of the load and the metatarsal bones connected to the toes absorb the other impact force from the ground . The flexible tendon on the sole called the plantar fascia ligament is in charge of the structural damping with the movement of the bones by varying the distance between heel and toe. It is also known that the division of toes helps to maintain stability while walking during the stance phase by effectively distributing loads. In the mechanical design of a robot foot, more than



**Figure 2.1:** Arch-type skeletal structure of human foot

the morphological imitation it is important to consider how to apply the benefits that the human foot brings from a mechanical point of view. During the walking motion, the sole of a human foot experiences pressure change due to the weight and ground reaction.

The sole of a human foot mainly consists of three parts: heel part, toe joints (the part where the metatarsal bones end) and the five toes. Figure [2.2] displays the pressure transition on the sole during a single stride by a standard male as the ZMP is moving forward from the heel to the toe. As shown, the largest pressure is exerted at the heel when the foot strikes the ground. The frontal big toe accepts much more pressure than others when the foot takes off from the ground, which is a reasonable result considering that the centre of mass of the human body is located between the two feet [7].



**Figure 2.2:** Transient pressure distribution on sole during a single stride (standard male)

## 2.2 Kinematics

### 2.2.1 Anthropomorphic human dimensions, volume and weight distribution

To build a proper kinematic model of an human-exoskeleton system, human dimensions must be considered as reference. Kinetics analysis require also data regarding mass distributions, mass centers, moment of inertia, and the like. It is also necessary to identify the exact location of the body joints centers of rotation. Anthropometry is the major branch of anthropology that studies the physical measurements of the human body to determine differences in individuals and groups. It relates the body characteristics described above with some of their determinants such as race, sex, age and body type, thanks to a wide variety of measurements.

Dempster [8] and coworkers have summarized estimates of segments lengths and joint center locations relative to anatomical landmarks. Drillis and Continini [9] reported an average set of segment lengths expressed as a percentage of the body total height, used as relative unit of measure 2.3. By reasoning in this manner, they also computed the position of the center of mass of each body segment, and expressed it as percentage of the total length of the segment 2.4. These segment proportion could be useful as a good approximation in absence of more accurate data, better if measured directly from the individual [10] .

The moments of inertia of a body segment are defined about an axis of rotation which, in most studies, is passing through its center of gravity. Occasionally are defined as passing through an estimated joint center of rotation.

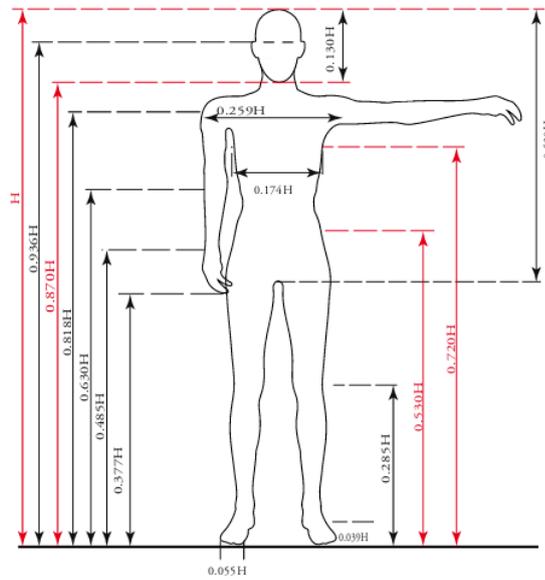


Figure 2.3: Body segment lengths expressed as a fraction of body height

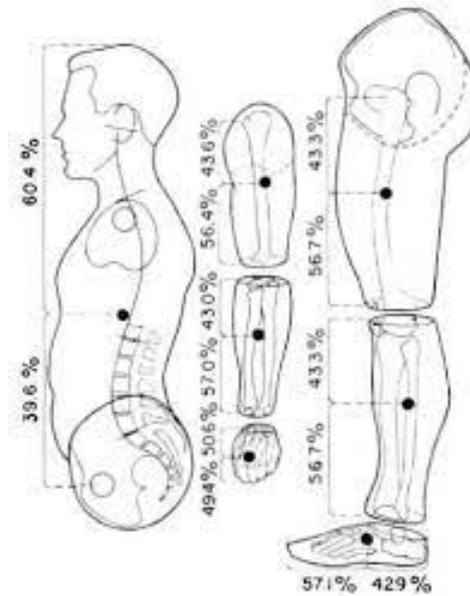


Figure 2.4: Location of mass centers of body segment

## 2.2.2 Reference systems conventions

Interests in the actual patterns of movement of humans and animals goes back to prehistoric times and was depicted in cave drawings, statues and paintings. Such

replications were subjective impressions of the artist. It was not until a century ago that the first motion picture cameras recorded locomotion patterns of both humans and animals. Marey, the French physiologist, used a photographic "gun" in 1885 to record displacements in human gait and chronophotographic equipment to get a stick diagram of a runner. About the same time, Muybridge in the United States triggered 24 cameras sequentially to record the patterns of a running man. Progress has been rapid during this century, and we now can record analyze everything from the gait of a child with cerebral palsy to the performance of an elite athlete [10].

The term used for these descriptions of human movement is *kinematics*. Kine-

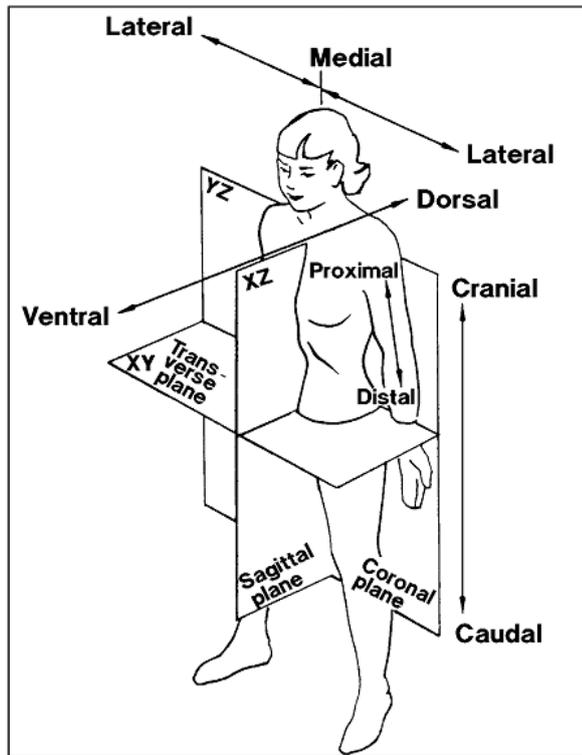


Figure 2.5: Human motion planes ©NASA

tics is not concerned with the forces, either internal or external, that cause the movement, but rather with the details of the movement itself. A complete and accurate quantitative description of the simplest movement requires a huge volume of data and a large number of calculations, resulting in a enormous number of graphic plots. For example, describing the movement of the lower limb in the sagittal plane during one stride can require up to 50 variables. These include linear and angular displacements, velocities and accelerations. It should be understood that any given analysis may use only a small fraction of the available kinematic variables. An assessment of a running broad jump, for example, may require only the velocity

and height of the body's center of gravity. On the other hand, a mechanical power analysis of an amputee's gait may require almost all the kinematic variables that are available.

In order to keep track of all the kinematic variables, it is important to establish a convention system. Thus if we wish to analyze movement relative to the ground or the direction of gravity, we must establish a spatial reference system [2.5]. Such conventions are mandatory when imaging devices are used to record the movement. However, when instruments are attached to the body, the data become relative, and we lose information about gravity and the direction of movement.

Trajectories are usually represented w.r.t. an absolute global spatial reference system, with fixed origin and axis directions. The one utilized throughout this text is commonly used to describe gait. It is based on the *body planes*, hypothetical geometric planes used to divide body into sections, used in anatomical terminology. The *frontal* or *coronal* plane is the vertical plane dividing the body in anterior and posterior part, the *transversal* or *medial* plane is horizontal, and splits the body in upper and lower part, the *sagittal* plane is vertical, and separates the body in left and right part.

The axis of our reference system are X in the direction of progression at the intersection of the sagittal and transverse planes, Y in the sideways direction at the intersection of the transverse and frontal plane and Z in the vertical direction at the intersection of the transverse and frontal plane. X is positive in the anterior direction, Y in the left direction and Z in the superior direction. The angles are measured starting from the direction of the first axis that expresses the plane (e.g. for the XY the angle starts from  $0^\circ$ ), increasing counterclockwise. The origin of this absolute reference system is located on the body *center of gravity* or of *mass* (CoG or CoM), positioned at this point at the start of the walk, and then held fixed in space.

In some cases it has been useful to represent the kinematic variables not w.r.t. to the center of mass, but w.r.t. the ground. The new reference system will have as its center the projection of the CoG on the soil, and the XY plane will no longer correspond to the transverse (or medial) plane but to the ground surface.

Joint angular displacements and velocities are expressed w.r.t. a local frame, centered in the center of the articulation. It has as its axes the axis of joint rotation, an axis in directed along one of the body segments connected by the joint and the last one positioned according to the right hand rule. This convention is very close to the one used for describing the manipulators in robotics.

In the anatomical literature is established a definite convention to describe the angular motion of the joints, w.r.t. the body planes described above.

- *Flexion* and *extension*: for joint rotations in the sagittal plane. With flexion the articulation angle is decreasing while during extension increases.

- *Abduction* and *adduction*: for joint rotations in the sagittal plane. During abduction the body segment is moved away from the medial line, and in the adduction the opposite occurs.
- *External* and *internal* rotation: rotations of a limb around its axis on the horizontal plane. The former brings the limb closer to the medial line, in the latter further.

## 2.3 Gait Cycle

### 2.3.1 Cycle Divisions

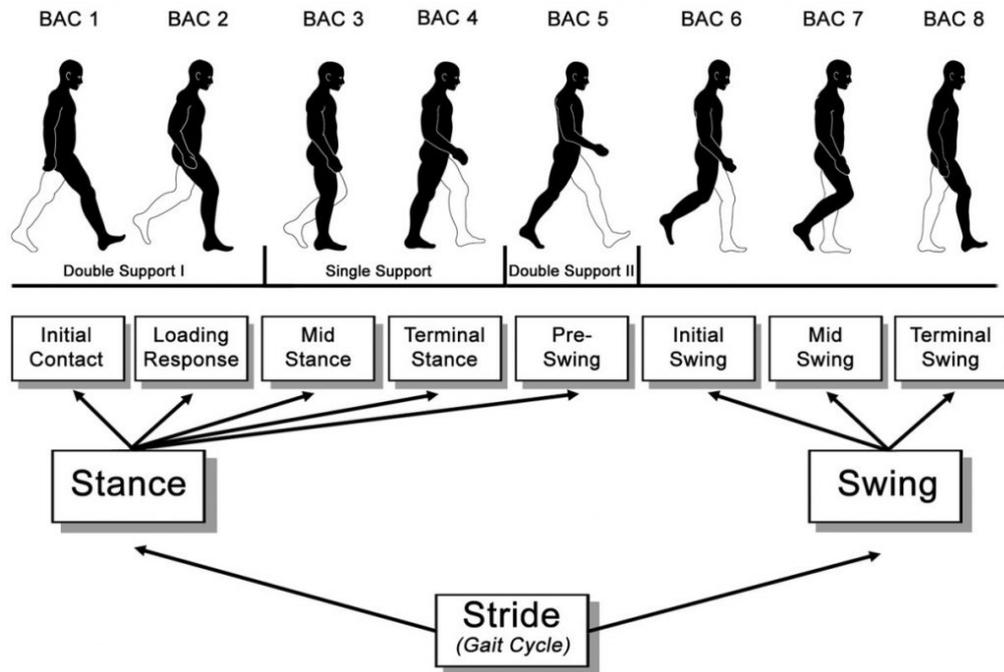
As the body moves forward, one limb serves as a mobile source of support while the other limb advances itself to a new support site. Then the limbs reverse their roles. For the transfer of body weight from one limb to the other, both feet are in contact with the ground. This series of events is repeated by each limb with reciprocal timing until the person's destination is reached.

A single sequence of these functions by one limb is called a *gait cycle* (GC). With one action flowing smoothly into the next, there is no specific starting or ending point. Hence, any event could be selected as the onset of the gait cycle. Because the moment of floor contact is the most readily defined event, this action generally has been selected as the start of the gait cycle. Normal persons initiate floor contact with their heel. As not all patients have this capability, the generic term *initial contact* (IC) will be used to designate the onset of the gait cycle.

Each gait cycle is divided into two periods, stance and swing. These often are called gait phases.

*Stance* is the term used to designate the entire period during which the foot is on the ground. Stance begins with initial contact 2.6. The *swing* applies to the time the foot is in the air for limb advancement. Swing begins as the foot is lifted from the floor (toe-off). Stance is subdivided into three intervals according to the sequence of floor contact by the two feet. Both the start and end of stance involve a period of bilateral foot contact with the floor (*double stance*), while the middle portion of stance has one foot contact.

*Initial double stance* begins the gait cycle. It is the time both feet are on the floor after initial contact. An alternate term is double limb support. This designation is to be avoided, however, as it implies an equal sharing of body weight by the two feet, which is not true during most of the double stance interval. *Single limb support* begins when the opposite foot is lifted for swing. In keeping with the terminology for the double contact periods, this should be (and often is) called *single stance*. To emphasize the functional significance of floor contact by one foot, the term *support* is preferred. During the single limb support interval the body's entire weight is



**Figure 2.6:** Division of the gait cycle

resting on that one extremity. The duration of single stance is the best index of the limb's support capability. *Terminal double stance* is the third subdivision. It begins with floor contact by the other foot (contralateral initial contact) and continues until the original stance limb is lifted for swing (ipsilateral toe-off). The term terminal double limb support has been avoided, as weight bearing is very asymmetrical.

**Timing.** The gross normal distribution of the floor contact periods is 60% for stance and 40% for swing 2.1. Timing for the phases of stance is 10% for each double stance interval and 40% for single limb support. Note that single limb support of one limb equals swing of the other, as they are occurring at the same time.

The precise duration of these gait cycle intervals varies with the person's walking velocity. The duration of both gait periods shows an inverse relationship to walking speed. That is, both total stance and swing times are shortened as gait velocity increases. The change in stance and swing times becomes progressively greater as speed slows. Among the subdivisions of stance a different relationship exists. Walking faster proportionally lengthens single stance and shortens the two double stance intervals. The reverse is true as the person's walking speed slows. This

<b>Stance</b>	60%
Initial Double Stance	10%
Single Limb Support	40%
Terminal Double Stance	10%
<b>Swing</b>	40%

**Table 2.1:** Floor Contact Periods

pattern of change also is curvilinear. Having an interval when both feet are in contact with the ground for the limbs to exchange their support roles is a basic characteristic of walking. When double stance is omitted, the person has entered the running mode of locomotion [11].

For Perry [11], the gait cycle can be split in more refined sub-phases depending on the support phases and the task which is being executed, in Figure [2.6].

- *Phase 1.* Initial Contact (IC)
- *Phase 2.* Loading Responde (LR)
- *Phase 3.* MidStance (MSt)
- *Phase 4.* Terminal Stance (TSt)
- *Phase 5.* PreSwing (PSw)
- *Phase 6.* Initial Swing (ISw)
- *Phase 7.* MidSwing (MSw)
- *Phase 8.* Terminal Swing (TSw)

*Phase 1* includes the moment when the foot just touches the floor. The joint postures present at this time determine the limb's loading response pattern.

*Phase 2* is the initial double stance period. The phase begins with initial floor contact and continues until the other foot is lifted for swing.

*Phase 3* is the first half of the single limb support interval. It begins as the other foot is lifted and continues until body weight is aligned over the forefoot.

*Phase 4* completes single limb support. It begins with heel rise and continues until the other foot strikes the ground. Throughout this phase body weight moves ahead of the forefoot.

*Phase 5* is the final phase of stance and is the second double stance interval in the gait cycle. It begins with initial contact of the opposite limb and ends with ipsilateral toe-off.

*Phase 6* is approximately one-third of the swing period. It begins with lift of the foot from the floor and ends when the swinging foot is opposite the stance foot.

*Phase 7* begins as the swinging limb is opposite the stance limb. The phase ends when the swinging limb is forward and the tibia is vertical (i.e., hip and knee flexion postures are equal).

*Phase 8* begins with a vertical tibia and ends when the foot strikes the floor. Limb advancement is completed as the leg (shank) moves ahead of the thigh.

### 2.3.2 Stride and Step

The gait cycle also has been identified by the descriptive term *stride*. Occasionally the word *step* is used, but this is inappropriate, in Figure [2.7].

Stride is the equivalent of a gait cycle. It is based on the actions of one limb.

The duration of a stride is the interval between two sequential initial floor contacts by the same limb (i.e., right IC and the next right IC).

Step refers to the timing between the two limbs. There are two steps in each stride (or gait cycle). At the midpoint of one stride the other foot contacts the ground to begin its next stance period.

The interval between an initial contact by each foot is a step (i.e., left and then right). The same offset in timing will be repeated in reciprocal fashion throughout the walk.

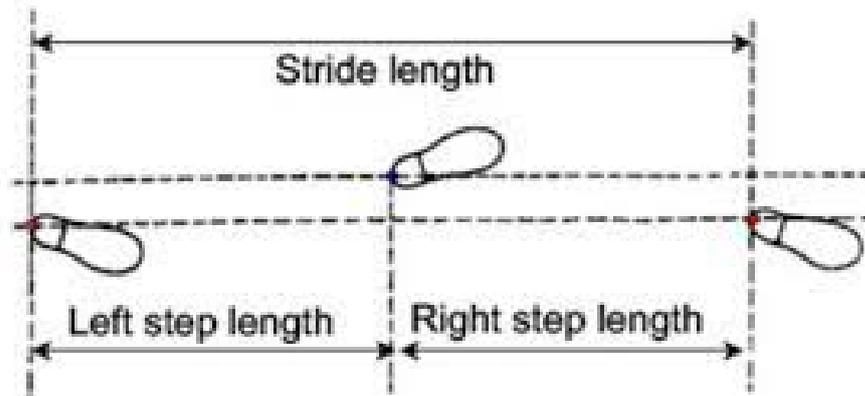


Figure 2.7: A step versus a stride

## 2.4 Defining the Measure of Balance

### 2.4.1 Zero Moment Point

Zero moment point (ZMP), in Figure [2.8], is a concept related with dynamics and control of legged locomotion, e.g., for humanoid robots. It specifies the point with respect to which dynamic reaction force at the contact of the foot with the ground does not produce any moment in the horizontal direction, i.e. the point where the total of horizontal inertia and gravity forces equals to 0 (zero). The concept assumes the contact area is planar and has sufficiently high friction to keep the feet from sliding.

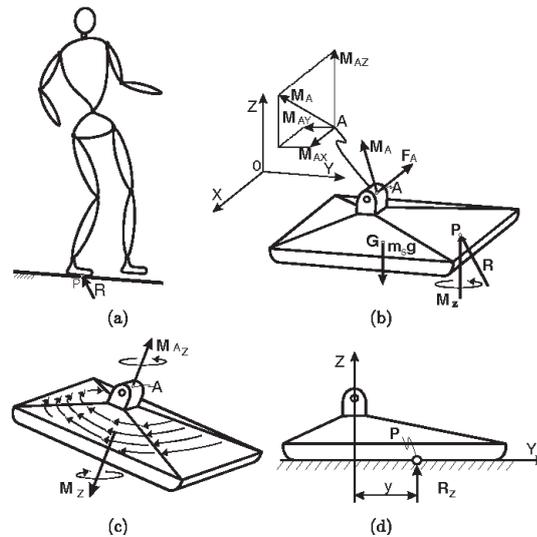
This concept was introduced in January 1968 by Miomir Vukobratović at The Third All-Union Congress of Theoretical and Applied Mechanics in Moscow.

The zero moment point is a very important concept in the motion planning for biped robots. Since they have only two points of contact with the floor and they are supposed to walk, “run” or “jump” (in the motion context), their motion has to be planned concerning the dynamical stability of their whole body. This is not an easy task, especially because the upper body of the robot (torso) has larger mass and inertia than the legs which are supposed to support and move the robot. This can be compared to the problem of balancing an inverted pendulum.

The trajectory of a walking robot is planned using the angular momentum equation to ensure that the generated joint trajectories guarantee the dynamical postural stability of the robot, which usually is quantified by the distance of the zero moment point in the boundaries of a predefined stability region. The position of the zero moment point is affected by the referred mass and inertia of the robot’s torso, since its motion generally requires large angle torques to maintain a satisfactory dynamical postural stability [12].

Hence, ZMP is a measure of balance, not a control methodology, but many different control systems have been built exclusively on this measure as a feedback mechanism and are therefore together here as ZMP control strategies.

One of the most basic measures of balance is the vertical projection of the center of mass (COM) also known as the center of gravity (COG). If the system moves slowly enough that the dynamic forces are negligible, then the system will be balanced if the COG lies within the base of support, or more technically, the convex hull of contact points. The problem with a COG measure is that it does not account for the dynamic forces of faster motions and it has a limited ability to deal with external disturbances. As a result, only a few systems have been based on this measure. A more suitable measure that takes dynamics into account is called the center of pressure (COP). The COP is basically a weighted sum of vertical forces applied to the foot to find the location of the net applied force. Another way of describing the COP is the location where a single force vector could be applied



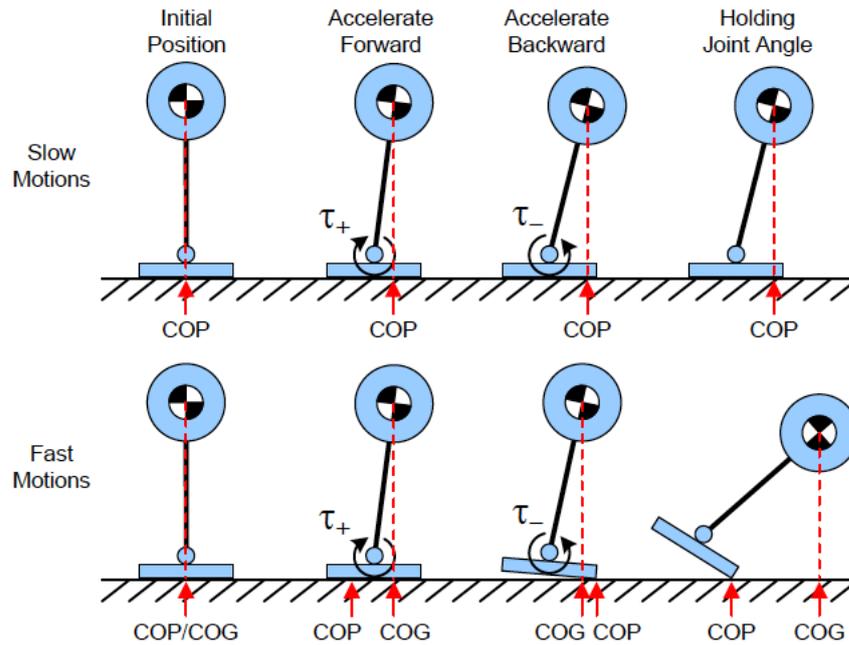
**Figure 2.8:** Support foot and influence of by the force, moment, ground reaction

without creating a moment about the foot, hence the zero moment point [13].

Figure [2.9] compares the center of pressure with the center of gravity. For slow motions, the COP and COG coincide. The COP and COG remain within the base of support and thus the biped remains balanced. For fast motions, however, as the COM accelerates forward, the COP moves behind the COG. Then as the COM decelerates, the COP moves in front of the COM until it hits the edge of the foot and cannot move any further forward. The COM is still within the base of support, but the COP has moved to the boundary of support, indicating that foot rotation is about to begin and a fall is imminent.

It should be noted that there is some debate in the literature about the equivalence of ZMP and COP, however, the differences are semantics. On a flat walking surface, it has been shown that the ZMP is mathematically equivalent to the COP [13], but according to Vukobratović, COP and ZMP only coincide in a dynamically balanced gait. When the gait is not dynamically balanced, the ZMP does not exist.

Walk is understood as moving “by putting forward each foot in turn, not having both feet off the ground at once.” From this definition, it transpires that walk is characterized by the displacement of legs such that both feet are not separated from the ground at the same time, which ensures that the body in the space (usually) moves forward. In view of the fact that the body is supported by the legs, ensuring that “the body in the space moves forward” is possible only if avoiding overturning is constantly taken care of, i.e. preserving the dynamic balance of the mechanism. Having in mind that all of the humanoid robot joints are powered and directly controllable except for the contact of the foot and the ground, this contact is essential for the walk realization.



**Figure 2.9:** The use of center of gravity as a measure of balance is only acceptable when the motions are slow and the dynamic forces are negligible.

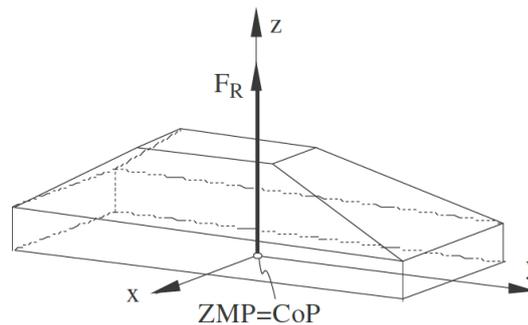
The mechanism's position with respect to the environment depends on both the relative positions of the links and the relative position of the foot with respect to the ground. In order for the humanoid to perform the reference motion, it is necessary to realize the predefined motions at the joints, and at the same time preserve the relative position of the foot with respect to the ground. Therefore, to prevent the humanoid from falling, it is necessary to ensure the appropriate dynamics of the mechanism above the foot to preserve the regular contact of the supporting foot with the ground. In other words, since the sole-ground contact is unilateral, a necessary condition for avoiding overturning is that the motion of the humanoid as a whole is such that, while the regular sole-ground contact is preserved, the overall ground reaction can be replaced by one force only. This can be mathematically expressed in the following.

If we introduce a Cartesian frame with the origin at the point where the resultant ground reaction (pressure) force is acting, with two axes ( $x$  and  $y$ ) being tangential to the ground and the third (the  $z$ -axis) being normal, then a mathematical expression for the fulfillment of dynamic balance is:  $\sum M_x = 0$  and  $\sum M_y = 0$ . The moments include gravity, inertial forces and other external forces acting on the humanoid body (like wind, different strike, etc.). It should be noted that it is not necessary for the third component of the moment (about the  $z$ -axis) to be

zero, provided it is compensated by the friction between the foot and ground. In such a case,  $\sum Mz \neq 0$  does not cause motion. The point inside the support area (excluding its edges) for which it holds that  $\sum Mx = 0$  and  $\sum My = 0$  is termed the Zero-Moment Point (ZMP).

To make the above clearer, let us return to the source interpretation of the ZMP. Let us consider the single-support phase of a dynamically balanced gait of the mechanism having a one-link foot. The foot of the supporting leg is in contact with the support surface as presented in [2.10].

Further, let us consider how to preserve dynamic balance of the mechanism and

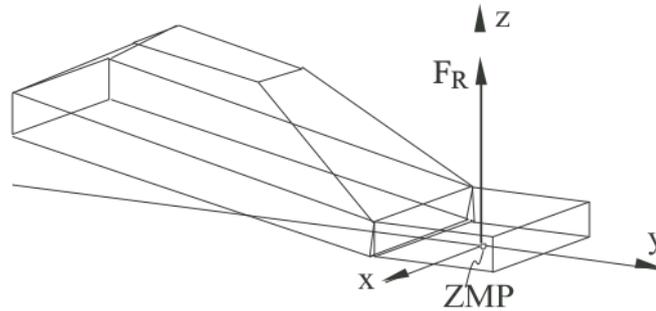


**Figure 2.10:** Foot of the supporting in the single-support phase

prevent it from falling. The answer is quite simple: by using an indicator that will warn of a critical situation approaching and it being necessary to undertake appropriate action to compensate. This indicator is the position of the ZMP inside the support area, and it corresponds to the position of the ground reaction force. The ZMP position inside the support area can easily be determined with the aid of force sensors on the sole, Figure [2.11]. All the time the ZMP is within the support area, there will be no rotation about the foot edge and the robot will preserve its dynamic balance. A warning means that the ZMP is coming closer to the foot edge [14]. Hence, the notion of the ZMP was introduced in order to control inertial forces. In the stable single support phase, the ZMP is equal to the COP on the sole. The advantage of the ZMP is that it is a point where the center of gravity is projected onto the ground in the static state and a point where the total inertial force composed of the gravitational force and inertial force of mass goes through the ground in the dynamic state. If the ZMP strictly exists within the supporting polygon made by the feet, the robot never falls down.

Most research groups have used the ZMP as a walking stability criterion of dynamic biped walking. To this end, the robot is controlled such that the ZMP is maintained within the supporting polygon.

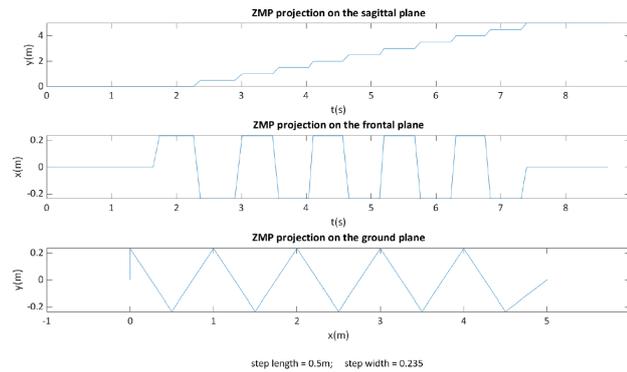
In general, the walking control strategies using the ZMP can be divided into two



**Figure 2.11:** Rotation of the supporting foot about its edge.

approaches. First, the robot can be modeled by considering many point masses, the locations of the point masses and the mass moments of inertia of the linkages. The walking pattern is then calculated by solving ZMP dynamics derived from the robot model with a desired ZMP trajectory. During walking, sensory feedback is used to control the robot. Second, the robot is modeled by a simple mathematical model such as an inverted pendulum system, and then the walking pattern is designed based on the limited information of a simple model and experimental hand tuning. During walking, many kinds of online controllers are activated to compensate the walking motion through the use of various sensory feedback data including the ZMP. The first approach can derive a precise walking pattern that satisfies the desired ZMP trajectory, but it is hard to generate the walking pattern in real-time due to the large calculation burden. Further, if the mathematical model is different from real robot, the performance is diminished. On the contrary, the second approach can easily generate the walking pattern online. However, many kinds of online controllers are needed to compensate the walking pattern in real-time, because the prescribed walking pattern cannot satisfy the desired ZMP trajectory. In addition, this methods depends strongly on the sensory feedback, and hence the walking ability is limited to the sensor's performance and requires considerable experimental hand tuning [4].

The ZMP projection on the sagittal, frontal and ground planes during a walk with constant step width will appear like shown in 2.12. Each one is a broken line. The constant traits in the sagittal plane projection represent gait phases in which the ZMP is stationary on the sole of the support foot (thus they are referred to the single stance phase), while in the sloping ones it moves forward from the rear foot towards the front one (thus they are referred to the single stance phases). The same happens in the frontal projection, where is possible differentiate between constant single support traits and sloped double support traits. The combination of the two will produce a graph representing the projection of the ZMP on the ground.



**Figure 2.12:** Projection of the ZMP for a constant step walk on the Sagittal, Frontal and ground planes

## 2.5 Model for biped robots

### 2.5.1 Linear Inverted Pendulum Model (LIPM)

Linear inverted pendulum model (LIPM) 2.13 is an effective and widely used simplified model for biped robots. However, LIPM includes only the single support phase (SSP) and ignores the double support phase (DSP). In this situation, the acceleration of the center of mass (CoM) is discontinuous at the moment of leg exchange, leading to a negative impact on walking stability. If the DSP is added to the walking cycle, the acceleration of the CoM will be smoother and the walking stability of the biped will be improved.

Compared with other types of robots, humanoid robots have good adaptability to the environment, stronger obstacle avoidance ability, and a smaller moving blind area, which has attracted the attention and in-depth research of scholars. At present, biped robots are still quite far away from the real sense of anthropomorphism, and there are many problems to be solved in this field. For example, due to the inherent instability of biped walking, walking stability analysis is still an important issue for biped robots. In addition, the biped robot is a high-order and strong coupling nonlinear system, which makes the trajectory planning and control difficult. The realization of stable walking is the primary task in the research of humanoid robots. There are many methods for gait planning of biped robots. These methods could be divided into two classes. The first uses the accurate information of dynamical parameters to generate walking patterns. Joint angle trajectories or trajectories of some key parts, e.g. hip and/or feet, are usually fitted by spline or polynomial functions, then the coefficients of spline or polynomial functions are determined by parameter optimization technique. However, these gait-planning methods need a lot of computation and cannot meet the requirement of trajectory planning in real

time. The more degrees of freedom and the higher the order of the polynomials, the more computation time is needed for solving the optimization problem.

The other class is based on a simplified model to generate walking patterns. Inverted pendulum is widely used because of its simplicity. A biped robot is usually regarded as a concentrated mass and massless leg. The trajectory of the center of mass (CoM) is planned with a simplified model, and then the angles of other joints are solved by inverse kinematics. One of the widely used methods is the linear inverted pendulum model (LIPM). The advantage of LIPM is that the trajectory of the CoM has an analytical solution. Moreover, its forward and lateral motions are decoupled. Another model is the inverted pendulum model (IPM) with constant leg length. In this model, the CoM moves along an arc. Although the dynamic equation of IPM is simple, there is no analytical solution due to its nonlinearity. The disadvantages of LIPM and IPM are that they can only generate the trajectory of the single support phase (SSP), but cannot generate the trajectory of the double support phase (DSP).

From an application perspective, when the biped robot is walking outdoors, due to the unstructured ground environment, the robot is required to have the ability of real-time gait generation according to the current environment. However, the more accurate the model is, the more computation is needed. Hence real-time gait planning may become very difficult. Therefore, the simplified model is a feasible and very useful method for real-time gait planning.

On the other end of spectrum, there is little attention on the DSP. Many gait-planning methods consider only the SSP and ignore the DSP, or the DSP is assumed to be instantaneous. In this situation, the center of pressure (CoP) or zero-moment point (ZMP) needs to transfer from the trailing foot to the leading foot instantaneously when the support leg is switched. This requires an impulsive force between the rear foot and the ground. The emerge of impulsive force could lead to some adverse factors:

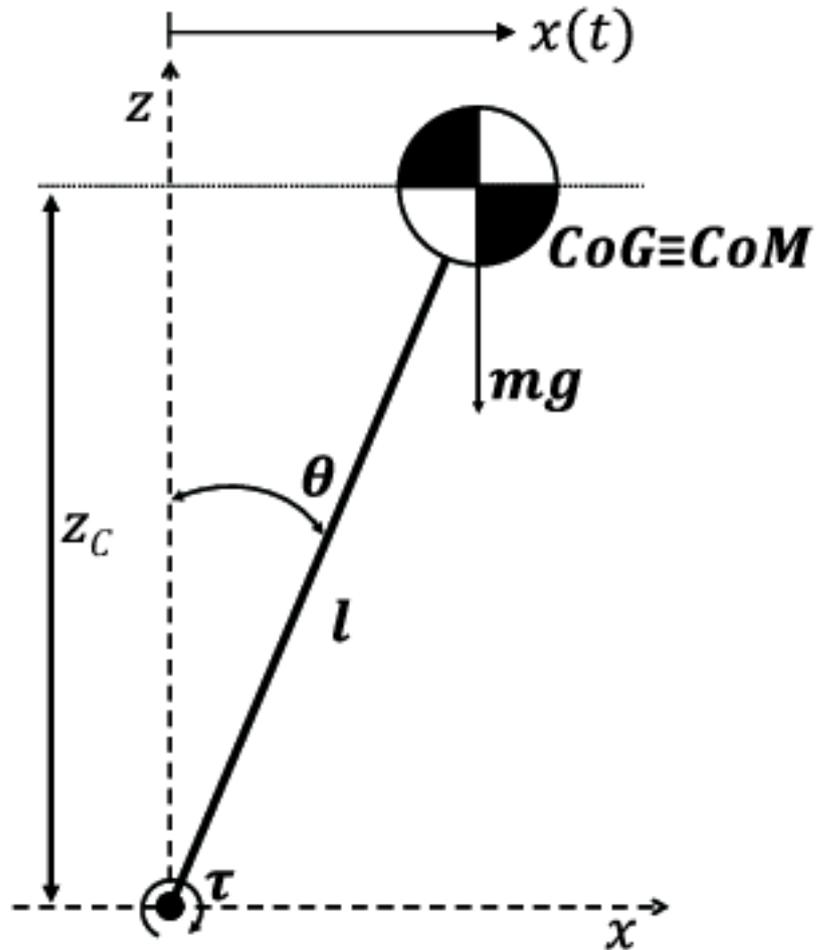
- it has a negative effect on the walking stability analysis;
- generating impulse force needs a sufficiently large joint torque that the joint driving motors may not provide;
- it may damage the hardware of the robot.

The introduction of the DSP can reduce the impact between the foot and the ground, make a smooth ZMP transition from the trailing foot to the leading foot, and improve walking stability. In addition, the support polygon area of the DSP is larger than that of the SSP, so the ground can provide greater external torque to the robot during the DSP. Therefore, the robot has stronger state-adjustment ability during the DSP. During the SSP, because the robot's foot is small, the ground cannot provide a large enough external torque to avoid the robot falling

down; as a result, the robot needs more adjustment of the internal state. Kajita and Tani reported that adding the DSP to the LIPM reduced the loss of the CoM's velocity when the support leg exchanges [15].

To overcome shortcomings of models without the DSP, some scholars introduced the DSP in gait planning. Kajita et al. [16] planned the CoM's trajectory of the DSP as a fourth order polynomial function. The coefficients of the polynomial are determined by the boundary condition and the specified duration of the DSP.

Motoi et al. [17] designed the CoM's trajectory in the DSP as a fifth order



**Figure 2.13:** Linear Inverted Pendulum Model

polynomial function. The disadvantage of their method is that walking stability was not taken into consideration during the DSP. With the increase of the order of the polynomial, unexpected oscillation of CoM may occur; as a result, unexpected

oscillation of the ZMP may occur during the DSP. In addition, the displacement of the CoM during the DSP is not intuitive and cannot be perfectly integrated with LIPM.

Shibuya et al. [18] proposed the linear pendulum model (LPM) to plan the trajectory of the CoM in the DSP, and determine the appropriate suspension point, which can ensure that the acceleration of the CoM is continuous at the moment of the switch between the SSP and the DSP. However, they only plan the cyclic gait of the robot on the horizontal ground, and do not give the gait-planning method when the robot faces a more complex environment. Shibuya et al. [19] extended the results of [18] to generate DSP trajectories in two situations. One is to land the swing leg earlier than planned, and the other is trajectory planning to stop walking in the DSP. However, they still did not put forward the method in more situations. In this work, LIPM and LPM are used to plan the trajectories of the SSP and the DSP, respectively. The dynamic equations of LIPM and LPM are linear, so they have analytic solutions. Trajectory planning only needs a small amount of computation. Through dynamic analysis of two pendulum models and their ZMP, the stability of gait can be guaranteed. Moreover, LPM is well-compatible with LIPM.

### 2.5.2 3D Linear Inverted Pendulum Model (3D-LIPM)

The 3D Linear Inverted Pendulum Model (3D-LIPM) consists in a point mass and a massless staff. Let be  $p=(x,y,z)$  the position of the mass,  $r$  the length of the staff and  $\theta_p$  and  $\theta_r$  the angles of the staff w.r.t. the axis  $x$  and  $y$  respectively. The pendulum equations are:

$$x = r \sin(\theta_p) \tag{2.1}$$

$$y = -r \sin(\theta_r) \tag{2.2}$$

$$z = \sqrt{(1 - \sin(\theta_r)^2 - \sin(\theta_p)^2)} \tag{2.3}$$

While the equation is:

$$\begin{bmatrix} \tau_r \\ \tau_p \\ f \end{bmatrix} = m \begin{bmatrix} 0 & -r \cos(\theta_r) & -\frac{r \cos(\theta_r) \sin(\theta_r)}{\sqrt{(1 - \sin(\theta_r)^2 - \sin(\theta_p)^2)}} \\ r \cos(\theta_p) & 0 & -\frac{r \cos(\theta_p) \sin(\theta_p)}{\sqrt{(1 - \sin(\theta_r)^2 - \sin(\theta_p)^2)}} \\ \sin(\theta_p) & -\sin(\theta_r) & -\sqrt{(1 - \sin(\theta_r)^2 + \sin(\theta_p)^2)} \end{bmatrix} +$$

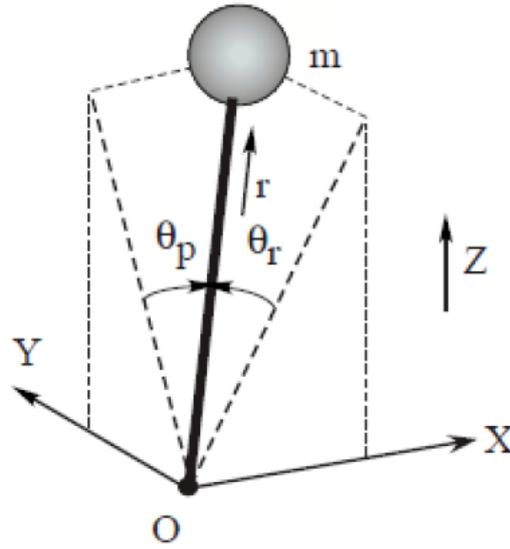


Figure 2.14: 3D pendulum

$$+ mg \begin{bmatrix} -\frac{r \cos(\theta_r) \sin(\theta_r)}{\sqrt{(1 - \sin(\theta_r)^2 - \sin(\theta_p)^2)}} \\ -\frac{r \cos(\theta_p) \sin(\theta_p)}{\sqrt{(1 - \sin(\theta_r)^2 - \sin(\theta_p)^2)}} \\ -\sqrt{(1 - \sin(\theta_r)^2 + \sin(\theta_p)^2)} \end{bmatrix}$$

The dynamic along the x-axis is:

$$m(z\ddot{x} - x\ddot{z}) = \frac{\sqrt{(1 - \sin(\theta_r)^2 - \sin(\theta_p)^2)}}{\cos(\theta_p)} \tau_p + mgx \quad (2.4)$$

And the equation for the dynamics along the y-axis is:

$$m(z\ddot{y} - y\ddot{z}) = \frac{\sqrt{(1 - \sin(\theta_r)^2 - \sin(\theta_p)^2)}}{\cos(\theta_r)} \tau_r - mgy \quad (2.5)$$

The pendulum motion can be constrained in the  $xy$  plane , considering that the

oscillations around the axis  $z$  are small compared to the others. The constrain plane is represented by the respective normal vector  $(k_x, k_y, -1)$  and by its  $z$  intersection  $z_c$  as [20]:

$$z = k_x x + k_y y + z_c \quad (2.6)$$

Replacing (2.6) and its second derivate in (2.4) and (2.5) we get:

$$\ddot{x} = \frac{g}{z_c} x + \frac{k_y}{z_c} (x\ddot{y} - \ddot{x}y) + \frac{1}{mz_c} \frac{\sqrt{(1 - \sin(\theta_r)^2 - \sin(\theta_p)^2)}}{\cos(\theta_p)} \tau_p \quad (2.7)$$

$$\ddot{y} = \frac{g}{z_c} y + \frac{k_x}{z_c} (x\ddot{y} - \ddot{x}y) - \frac{1}{mz_c} \frac{\sqrt{(1 - \sin(\theta_r)^2 - \sin(\theta_p)^2)}}{\cos(\theta_r)} \tau_r \quad (2.8)$$

If the above equation allow the pendulum motion in any plane and slope, constraining that to a flat plane ( $k_x = k_y = 0$ ) gives:

$$\ddot{x} = \frac{g}{z_c} x + \frac{1}{mz_c} \frac{\sqrt{(1 - \sin(\theta_r)^2 - \sin(\theta_p)^2)}}{\cos(\theta_p)} \tau_p \quad (2.9)$$

$$\ddot{y} = \frac{g}{z_c} y - \frac{1}{mz_c} \frac{\sqrt{(1 - \sin(\theta_r)^2 - \sin(\theta_p)^2)}}{\cos(\theta_r)} \tau_r \quad (2.10)$$

Considering small oscillations around the axes  $x$  and  $y$ , the formulas can be simplified further:

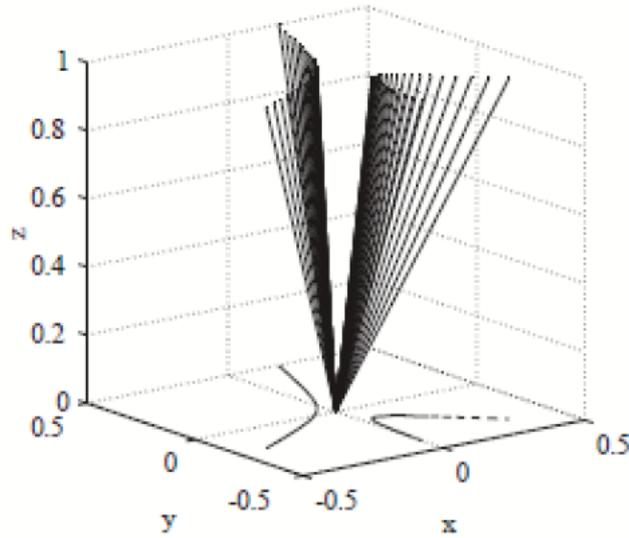
$$\ddot{x} = \frac{g}{z_c} x + \frac{1}{mz_c} \tau_p \quad (2.11)$$

$$\ddot{y} = \frac{g}{z_c}y - \frac{1}{mz_c}\tau_r \quad (2.12)$$

These are linear equations, and the pendulum dynamic is governed only by the parameter  $z_c$ . Even in the case of a sloped constrain where ( $k_x = k_y \neq 0$ ) we can obtain the same dynamics by applying the additional constrain:

$$\tau_x x + \tau_y y = 0 \quad (2.13)$$

We can conclude that the plane inclination never affect the horizontal motion. Solving the (2.11) and (2.12) with zero input torques (i.e.  $\tau_r = \tau_p = 0$ ) the trajectories of the pendulum ball motion is obtained in the field of gravity, such as the two examples in 2.15.



**Figure 2.15:** 3D Linear Inverted Pendulum Model

For the 3D-LIPM constrained horizontally ( $k_x = k_y = 0$ ), we can easily calculate the ZMP position on the floor ( $p_x, p_y$ ):

$$p_x = -\frac{\tau_y}{mg} \quad (2.14)$$

$$p_y = -\frac{\tau_x}{mg} \quad (2.15)$$

By substituting (2.14) and (2.15) in (2.11) and (2.12)

$$\ddot{x} = \frac{g}{z_c}(x + p_x) \quad (2.16)$$

$$\ddot{y} = \frac{g}{z_c}(y - p_y) \quad (2.17)$$

# Chapter 3

## Instrumentation

From the study of [21], in gait analysis the biomechanics of human motion, can be evaluated with different methods and instrumentation. These methods can be classified according to various criteria. For example they can be categorized into non-wearable or wearable sensors:

- *Non-wearable sensors*, are those that cannot be placed on a specific part of the body. However, they interact with the patient through their contact. Some examples in this category are force plates and instrumented treadmills.
- *Wearable sensors*, of several types, with different measurement accuracy and purposes. We can find markers, accelerometers, gyroscopes, magnetometers, electrodes for EMG and many others among them.

### 3.1 Non-wearable sensors

#### 3.1.1 Force transducers and force plates

To measure the force exerted by the human body on an external body or load, we need a suitable force-measuring device. Such a device, called force transducer, generates an electrical signal proportional to the applied force. There are many kinds available: strain gauge, piezoelectric, piezoresistive, capacitive, and others. All these work on the principle that the applied force causes a specific strain within the transducer [10].

- For the *gauge type*, a calibrated metal plate or beam within the transducer undergoes to a tiny change (strain) in one of its dimensions.
- *Piezoelectrical*, require slight of the deformations of the atomic structure within a block of special crystalline material, such as quartz. Deformation

of its crystalline structure changes the electrical characteristics such that the electrical charge across appropriate surfaces of the block is altered and can be translated via suitable electronics to a signal proportional to the applied force.

- The *piezoresistive* types exhibit a change in resistance which, like the strain gauge, upset the balance of a bridge circuit.

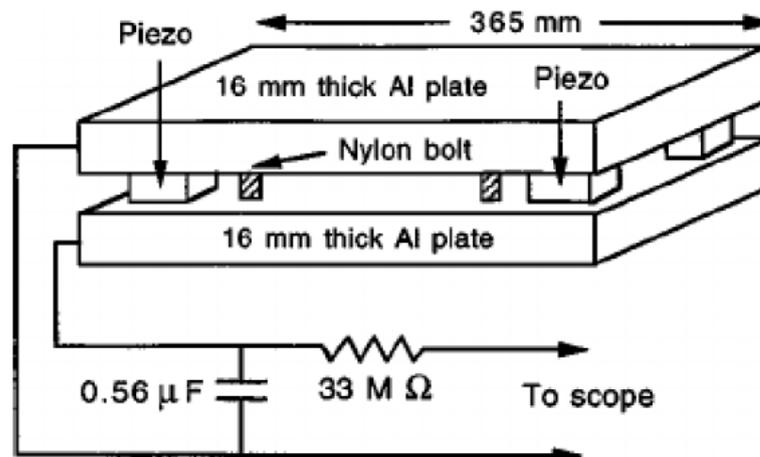


Figure 3.1: Scheme of a piezoelectrical force plate

### 3.1.2 Treadmills

Recently is becoming more popular to use treadmills for gait studies. They can be combined with cameras, markers systems, and force platforms can be inserted directly under the rollers. The use of treadmills in gait analysis has made possible to record straight line walking that can go beyond the distance covered in a laboratory limited by the cameras. It's also much easier to set a walking speed and it's also possible to conduct experiments with different inclinations w.r.t. the ground. However, depending on the future use of the recorded data, one of the disadvantages may be that the space-temporal recordings of the markers remain confined in the treadmill dimension's instead of advancing meter by meter like a real walk, so this data may need further adjustments [22].



**Figure 3.2:** Example of a modern system for gait analysis with the use of a treadmill

## 3.2 Wearable sensors

### 3.2.1 Optoelectronic stereophotogrammetry

It's a technique that involves cameras to capture the trajectory of spherical retro-reflective markers attached to the desired locations of the body. With stereophotogrammetry we can evaluate, with a good precision, movement and orientation of each body segment. It enables realistic reconstructions and representations of the musculoskeletal system during a certain motion task. For these reasons it is considered one of the best instrumentation for gait analysis [23].

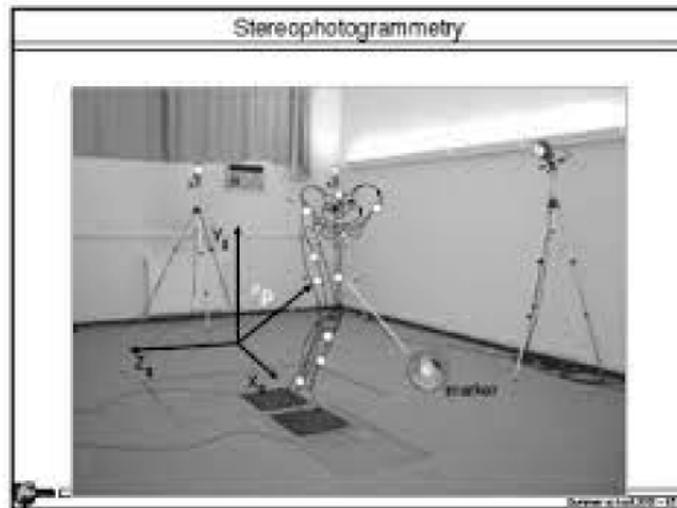
By the way it also suffers a bit from trajectory gaps, it takes long time for preparation and the space for analysis is restricted to the area in which the cameras are operating. In addition it is expensive.

### 3.2.2 Accelerometer

It's a measurement device whose output consist of the proper acceleration, the acceleration of the body on which is attached w.r.t. its instantaneous coordinate frame. Accelerometers can be single or multi axis and detect magnitude and direction of the acceleration, seen as a vector quantity.

In most accelerometers, the physical principle exploited to measure the acceleration is based on the inertia of a mass subjected to an acceleration. An elastic element suspends a mass, and this mass, in case of acceleration, moves from its rest position. Equating Hooke's law to Newtons law we have  $kx = ma$  and see that the displacement of the elastic element is proportional to the mass acceleration.

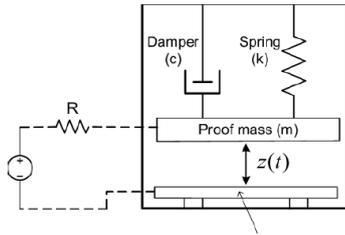
A displacement-sensitive sensor transforms the it into an electrical signal [23]. There are many types of accelerometers, such as capacitive, strain gauge, piezoresistive and piezoelectric. For gait analysis the most commonly used are capacitive and piezoresistive (3.4 and 3.5 respectively).



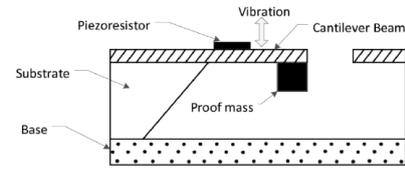
**Figure 3.3:** Stereophotogrammetry cameras system

### 3.2.3 Gyroscope

They measure the angular velocity around their sensing axis. Typically they are mechanical and consist of a rotating device which maintains fixed its rotating axis exploiting the conservation of angular momentum law. A 3D gyroscope can be described as a wheel mounted in three gimbals, which are the pivoted supports that enable the rotation around three different axes. The fundamental equation



**Figure 3.4:** Scheme of a capacitive accelerometer



**Figure 3.5:** Scheme of a piezoresistive accelerometer

describing a rotating rigid system is the following one.

$$M = \frac{dL}{dt} = \frac{d(I\omega)}{dt} \quad (3.1)$$

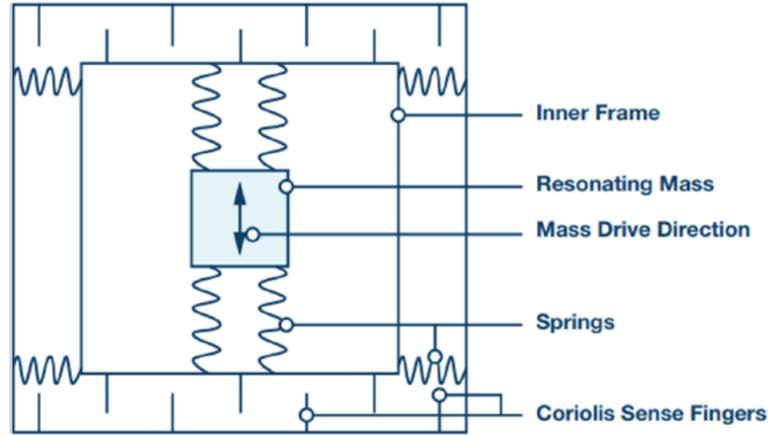
Where  $M$  is the torque,  $L$  the momentum,  $I$  the inertia and  $\omega$  the angular velocity. The derived motion is the precession, and the reaction force induces the gyroscope to rotate around a fixed axis, called spin axis, which does not change its direction even if the support varies its orientation. Thanks to the development of MEMS, miniaturized gyroscopes can become widespread. They consist of a vibrating element that, if subjected to a rotation, is also affected by a vibration in the orthogonal direction to the original one, according to the Coriolis effect [23]:

$$F = -2m(\omega \times v) \quad (3.2)$$

$F$  is the Coriolis force,  $\omega$  the angular velocity and  $v$  the linear velocity of the mass  $m$ .

### 3.2.4 Magnetometer

A magnetometer is a measuring device that detects a magnetic field. A scalar magnetometer measures the magnitude of the magnetic field directly, while the vectorial ones measure the direction and the strength of the magnetic field detecting the component along a particular axis. Using a three axial magnetometer, thus knowing the components of the magnetic field in three different and independent directions, allows to determine the vector in 3D space [23].



**Figure 3.6:** MEMS gyroscope scheme

$$h = (h_{earth} + h_{external})n \quad (3.3)$$

(3.3) represent a single axis magnetometer model, where  $n$  is the sensing axes. The most common of the magnetometers is the compass, which points in the direction of the Earth's magnetic north.

### 3.2.5 Inertial Measurement Units

In many fields, such as navigation, robotics and motion analysis, we need to know as much precise as possible the angular position in the space of objects. So for an accurate estimation of the orientation of a rigid body, w.r.t. an inertial frame, we can use an Inertial Measurement Units (IMU). They are composed by two sensors, a gyroscope which measures the angular rate, and an accelerometer which measures the linear and gravity acceleration. With these two sensors an IMU can estimate its attitude.

But since the accelerometer is not sensitive to the rotation around the gravity axis, an additional reference vector is needed to estimate the heading direction. Recent studies have discovered that combining an accelerometer with a magnetometer makes possible to find out both attitude and heading directions. This system is called Magneto-Inertial Measurement Units or MIMU [23]).

### 3.2.6 Electrogoniometers

An electrogoniometer is an electronic device that uses angle sensors, such as potentiometers, strain gauges and, more recently, accelerometers, appropriately positioned across a joint to measure its angle. It gives good results when used for body movements where we have limited speed and amplitude [26]. The most common electrogoniometers employ one of the following three sensor schemes:

- In *Potentiometric Electrogoniometer* an electrical resistance can be used to determine the angle between the joints. These types of electrogoniometers are somewhat bulky and restrict patient movement.
- For *Flexible Electrogoniometer* the strain gauge mechanism is housed inside a spring, which changes its electrical resistance proportionally to the variation of the angle between the plastic end blocks longitudinal axes.
- *Optoelectronic Systems* are video systems that use one or more video cameras to track bright markers placed at various locations on the patient's body. The system keeps track of the vertical and horizontal coordinates of each marker, and a software processes this information to determine the angle on the body segments of interest.

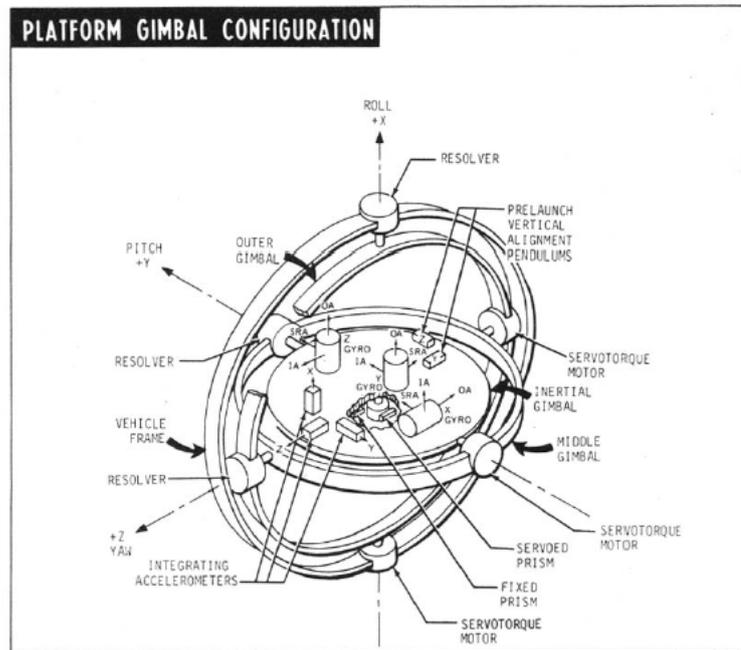
### 3.2.7 Optical fiber sensors

These sensors are made of flexible plastic optical fibers (OFS) through which optical signals are transmitted. The basic components of an OFS-based system are a light source, a flexible optical fiber and a photodetector. The light source at one of the extremities generates the optical signal, which travels through the flexible optical fiber and is received by the photodetector at the other extremity of the fiber. By measuring the attenuation of the optical signal, it is possible to determine the bending angle of the fiber.

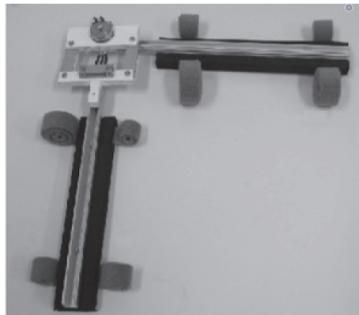
Due to this simple sensing principle and structure, OFS can be easily integrated into a monitoring system for measuring human joint angles. The main benefits of OFS are high resolution, flexibility, light-weight and immunity to electromagnetic interference [24].

### 3.2.8 Textile-based sensors

Textile-based sensors are very suitable for developing a wearable joint monitoring system. The working principles of all these sensors are similar. In all cases, changes of resistance are measured, and these changes are directly related to the



**Figure 3.7:** Inertial Measurement Units system

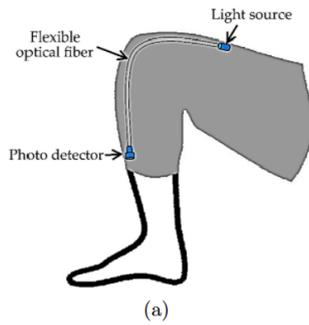


**Figure 3.8:** A potentiometric electrogoniometer

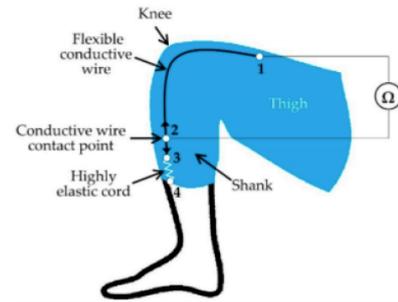


**Figure 3.9:** Optoelectronic system positioned on a patient

corresponding joint angles. To develop a long-term and regular wearable monitoring device, textile-based sensors can be a good choice because of their flexibility and simple sensing principle. Furthermore, they can be easily integrated into stretchable skin-tight fabrics around the joints. The measurement parameter is the resistivity change of the conductive wire w.r.t. the joints movement [24].



**Figure 3.10:** Scheme of an optical fiber sensor



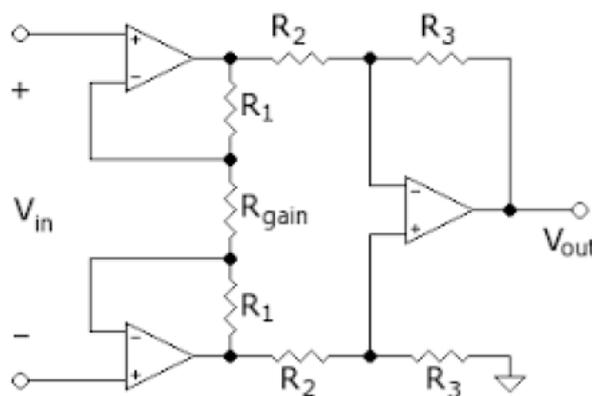
**Figure 3.11:** Scheme of a wearable wire sensor

### 3.2.9 EMG signals

The electrical signal associated with the contraction of a muscle is called an electromyogram, or EMG. The study of EMGs, is called electromyography. An EMG signal increases in amplitude as the intensity of the voluntary muscle activity it quantifies increases.

Electrodes are used for their recording and can be divided into two main groups, surface and indwelling electrodes. For both groups, the basic function is linked to the correct positioning on the patient (position and surface of contact) and the appropriate adjustment of the amplifier with which they operate (Instrumentation Amplifier). EMG signals, depending on the application and specifications of the acquisition system, once recorded are processed with, for example, filters and rectifiers.

EMG signals are a fundamental tool in the analysis of muscle behavior associated with a particular task: for this reason they are widely used in gait analysis [10].



**Figure 3.12:** Scheme of an instrumentation amplifier

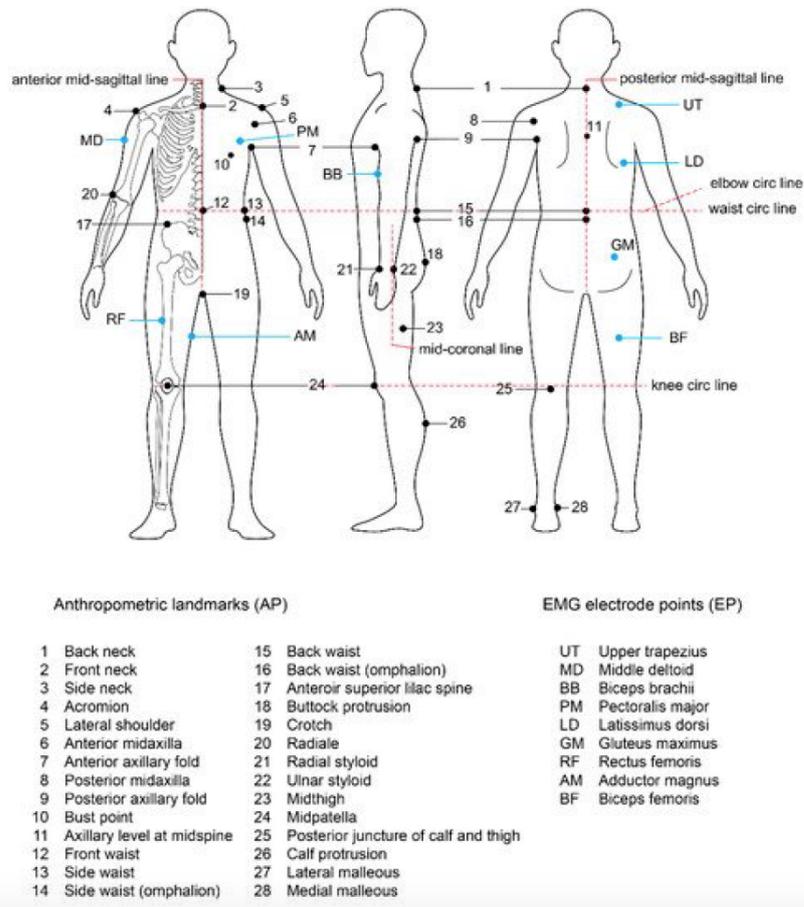


Figure 3.13: Example of a possible placement of electrodes on patient bodies

# Chapter 4

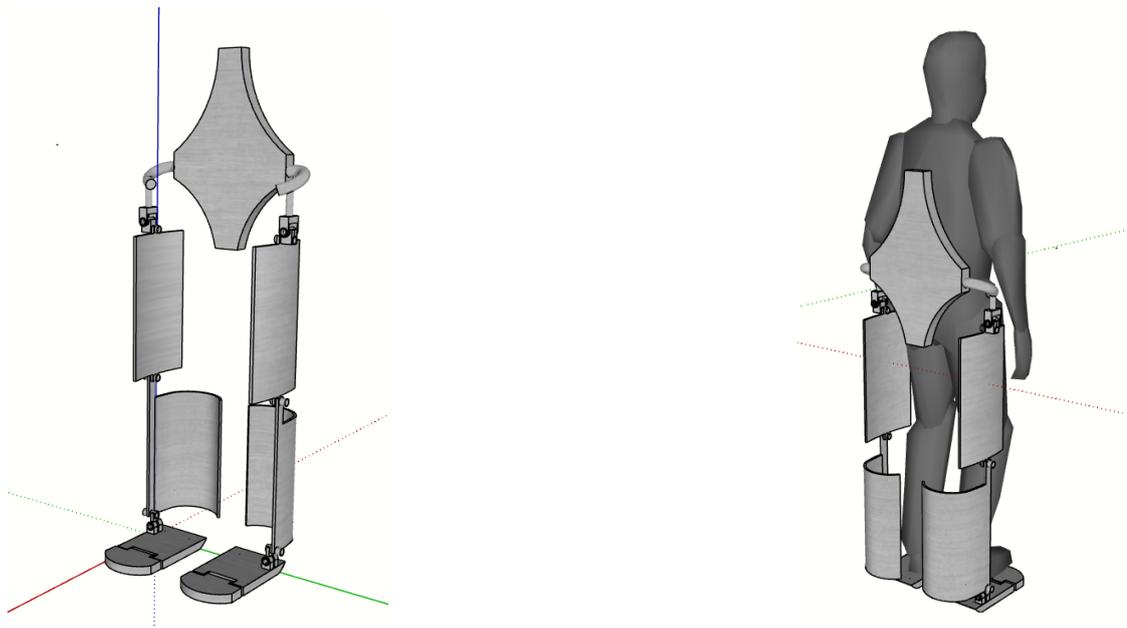
## Biped creation

In this chapter is explained how the biped has been created.

In the first place the design was created using a 3D design software (*SketchUp*) and then the biped was made on another CAD software system (*Onshape*) in order to export and use the project on Coppeliasim.

### 4.1 Biped design

The bipedal design was done using *SketchUp 2020*. SketchUp is a computer graphics application for 3D modeling, originally created by Last (founded in 2000 by Brad Schell and Joe Esch) and oriented towards architectural design, urban planning, civil engineering, video game development and related professions [25]. With this software each component is created to safely support a patient undergoing rehabilitation. The height of the structure is  $1579\text{mm}$ , its width is  $473\text{mm}$  and the thickness is not always the same, it varies between  $10\text{mm}$  and  $43\text{mm}$ . The figures [4.1] show the entire structure of the biped.



**Figure 4.1:** Biped design

### 4.1.1 Biped parts

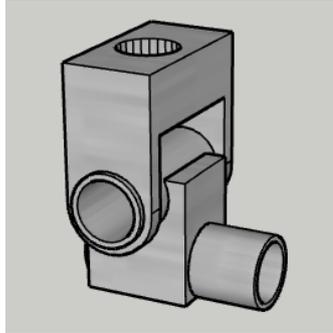
In this section all the parts that make up the biped are shown.

In Figure [4.2] there is the trunk design together with the hip joint responsible for the rotation around the Z-axis (Yaw):



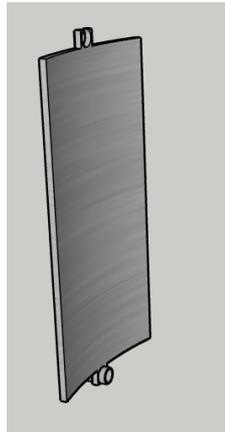
**Figure 4.2:** Trunk design

In Figure [4.3] is shown the right hip design composed by both the joint responsible for the rotation around the X-axis (Roll) and the rotation around the Y-axis (Pitch):



**Figure 4.3:** Right hip design

In Figure [4.4] is shown the right up leg together with the knee joint responsible for the rotation around the Y-axis (Pitch):



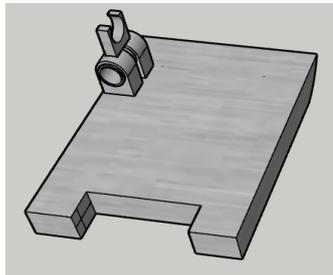
**Figure 4.4:** Right up leg and knee joint design

In Figure [4.5] is shown the right low leg together with the ankle joint responsible for the rotation around the Y-axis (Pitch):



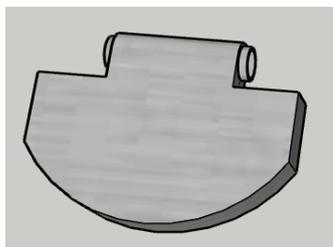
**Figure 4.5:** Right low leg and ankle joint(Y) design

In Figure [4.6] is shown the right foot together with the ankle joint responsible for the rotation around the X-axis (Roll):



**Figure 4.6:** Right foot and ankle joint(X) design

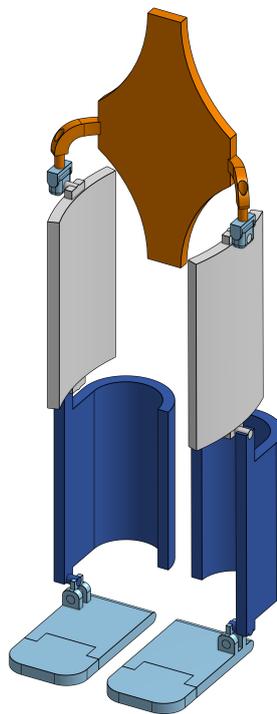
In Figure [4.1.1] is shown the toe together with its joint responsible for the rotation around the Y-axis (Pitch):



In order to implement everything on *matlab* a prototype with this design has been created on *Onshape*.

Onshape is a computer-aided design (CAD) software system, delivered over the Internet via a Software as a Service (SAAS) model. It makes extensive use of cloud computing, with compute-intensive processing and rendering performed on Internet-based servers, and users are able to interact with the system via a web browser or the iOS and Android apps[26].

In the following figure is shown the prototype.



**Figure 4.7:** Prototype created on *Onshape*



**Part II**  
**Simulation Part**

## Chapter 5

# Incline Experiment

*InclineExperiment.m* contains leg joint kinematics, kinetics, and EMG activity from an experimental protocol approved by the Institutional Review Board at the University of Texas at Dallas. Ten able-bodied subjects walked at steady speeds and inclines on a Bertec instrumented treadmill for one minute per trial. Each subject walked at every combination of the speeds 0.8 m/s, 1.0 m/s, and 1.2 m/s and inclines from -10 degrees to +10 degrees at 2.5 degree increments, for a total of 27 trials. During each trial, a 10-camera Vicon motion capture system recorded leg kinematics, while force plates in the Bertec treadmill recorded ground reaction forces, and a Delsys Trigno EMG system recorded muscle activation of the rectus femoris, biceps femoris, tibialis anterior, and gastrocnemius. This data can be used to test different hypotheses and models of human locomotion at varying speeds and inclines.

The file is of \*.mat format and can be opened using MATLAB.

This experiment's data is saved in two MATLAB structs, Continuous and Gaitcycle.

**Continuous:** This struct has one minute data recorded at 100 Hz (6000 measurements). The EMG and force plate data, which record at 2000 and 1000 Hz respectively, were downsampled so that all data in this struct is shown at 100 Hz. The units for all quantities are included in the struct Data Hierarchies below.

$$y = \text{Continuous}.(subject).(trial).(datatype).(leg).(variable) \quad (5.1)$$

[y will generally be of dimensions  $6000 \times 1$  or  $6000 \times 3$ , depending on if the variable is scalar-valued or has x,y,z components]

- (subject) =
  - ABXX: Individual results are shown for all 10 anonymous able-bodied subjects.
- (trial) =
  - subjectdetails: This field gives the subject’s gender, age, height, weight, and left/right leg lengths.
  - sXXi/dYY: Results are given for each combination of walking speed and incline a subject performed. In the naming convention shown, XX = speed of the trial in m/s, and YY = incline of walking in degrees. Decimal points are replaced with an “x.” An “i” before YY indicates an inclined trial, while a “d” indicates a decline. For example, s1x2i7x5 would mean the trial was recorded at 1.2 m/s at positive incline of 7.5 degrees. This information is also provided explicitly in (trial).description.
- (datatype) =
  - description: A cell array containing the speed and incline of this trial.
  - time: An array with the time since the beginning of the experiment for every frame.
  - kinematics.markers: An array with world-frame positions of all motion-capture markers, located on the Anterior Superior Iliac Spine (asi), Posterior Superior Iliac Spine (psi), the thigh, knee, tibia, ankle, heel and toe of both legs.
  - kinematics.jointangles: An array with the joint angles for the pelvis, hip, knee, ankle, and foot as calculated by Vicon Plug-in Gait (Vicon, Oxford, UK).

Joint definitions, provided by Vicon:

  - \* pelvis: (absolute) The angle between the pelvis and the laboratory coordinate system.
  - \* hip: (relative) The angles between the pelvis and the thigh (+x is flexion, -x is extension).
  - \* knee: (relative) The angles between the thigh and the shank (+x is flexion, -x extension).
  - \* ankle: (relative) The angles between the shank and the foot (+x dorsiflexion, -x plantarflexion).
  - \* foot: Absolute. The angles between the foot and the global coordinate system.

- (emgdata.emg):
  - Delsys EMG sensors (Model:Trigno wireless system, Delsys, Natick, MA) were attached to the rectus femoris (RF), biceps femoris (BF), tibialis anterior (TA), and gastrocnemius (GC). The EMG signals have been rectified and low-pass filtered (fc=40) with a zero-phase digital filter (MathWorks, Natick, MA).
  - emgdata.accel: Each Delsys EMG also contains a 3-axis accelerometer that reports an acceleration vector in the local frame.
  - kinetics.jointpower: An array of the power generated by each joint, determined by Plug-In Gait (Vicon).
  - kinetics.jointforce: An array of the force applied at each joint, determined by Plug-In Gait (Vicon).
  - kinetics.jointmoment: An array of the moment generated by each joint, determined by Plug-In Gait (Vicon).
  - kinetics.forceplate.force: A 3D force vector from force plates in the split belt instrumented treadmill (Bertec, Columbus, OH). These signals have been low-pass filtered (fc=40) with a zero-phase digital filter (MathWorks).
  - kinetics.forceplate.moment: A 3D moment vector from force plates in the split belt instrumented treadmill (Bertec).
  - kinetics.forceplate.cop: The center of pressure location (world-frame) from force plates in the split belt instrumented treadmill (Bertec).
- (variable)=
  - (marker): In the markers (datatype), results are given per marker: asi, psi, thigh, knee, tibia, ankle, heel or toe.
  - (joint): In several (datatype), results are given per joint: hip, knee, or ankle and for the jointangles (datatype), also pelvis and foot.
  - (muscle): For the emgdata (datatype), results are given per muscle: RF, BF, TA, or GC.
  - (forceplate): For the forceplate (datatype), results are given per the force, moment, or center of pressure (cop) measurement from Bertec forceplates (Bertec).

**Gaitcycle:** This struct contains the same underlying data as *Continuous*, but has been broken down into individual gait cycles which begin and end at heel strike. It contains most of the same fields as *Continuous*, plus the fields listed below:

$$y = \text{Gaitcycle}(\text{subject}).(\text{trial}).(\text{datatype}).(\text{leg}).(\text{variable}) \quad (5.2)$$

[y will generally be of dimensions 150xM where M is the number of strides recorded for that subject and trial. The 150 points are a linear interpolation of the continuous signal over a given gait cycle.]

- (datatype)=
  - stepsout: Contains a vector of strides that we have identified to be outliers, as defined by having kinematics 3 standard deviations from the mean.
  - cycles.time: An array with the same dimensions as the other Gaitcycle data that indicates the time since the beginning of the corresponding stride.
  - cycles.frame: A vector that indicates what frame each heel strike occurred on.
- (variable)=
  - (joint): For the jointpower (datatype), results are given per joint: hip, knee, or ankle.
  - (muscle): For the emgdata (datatype), results are given per muscle: RF, BF, TA, or GC.
  - (direction): For most other (datatype), results are split into three component directions: x, y, or z.
  - (joint)<sub>mean</sub>, (muscle)<sub>mean</sub>, or (direction)<sub>mean</sub>: The mean of all M strides' (datatype) for a given subject, trial, and leg. This will be a 150 x 1 vector. Note this mean includes strides that were indicated to be outliers.
  - (joint)<sub>std</sub>, (muscle)<sub>std</sub>, or (direction)<sub>std</sub>: The standard deviation of all M strides' (datatype) for a given subject, trial, and leg. This will be a 150 x 1 vector. Note this standard deviation includes strides that were indicated to be outliers.

**Known issues:**

Kinematic data will be missing when less than three cameras have a concurrent view of a marker. These points are represented by a NaN in this dataset. We recommend the use of 'nanmean' and 'nanstd' on these data.

Jointpower, jointforce and jointmoment have not been filtered. We recommend a low-pass filter before use, but the cut-off frequency may depend on the application.

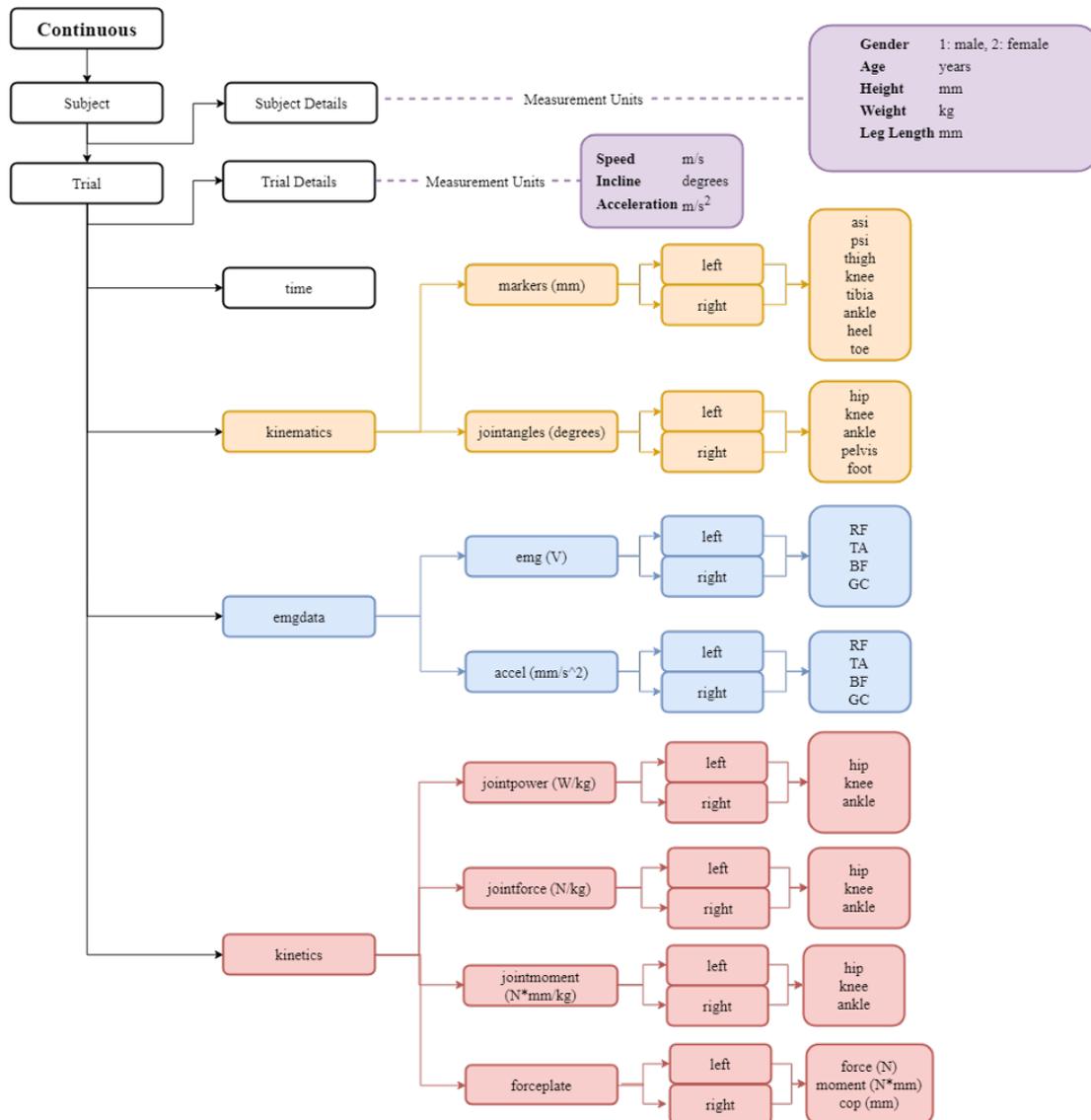


Figure 5.1: Continuous struct Data Hierarchy

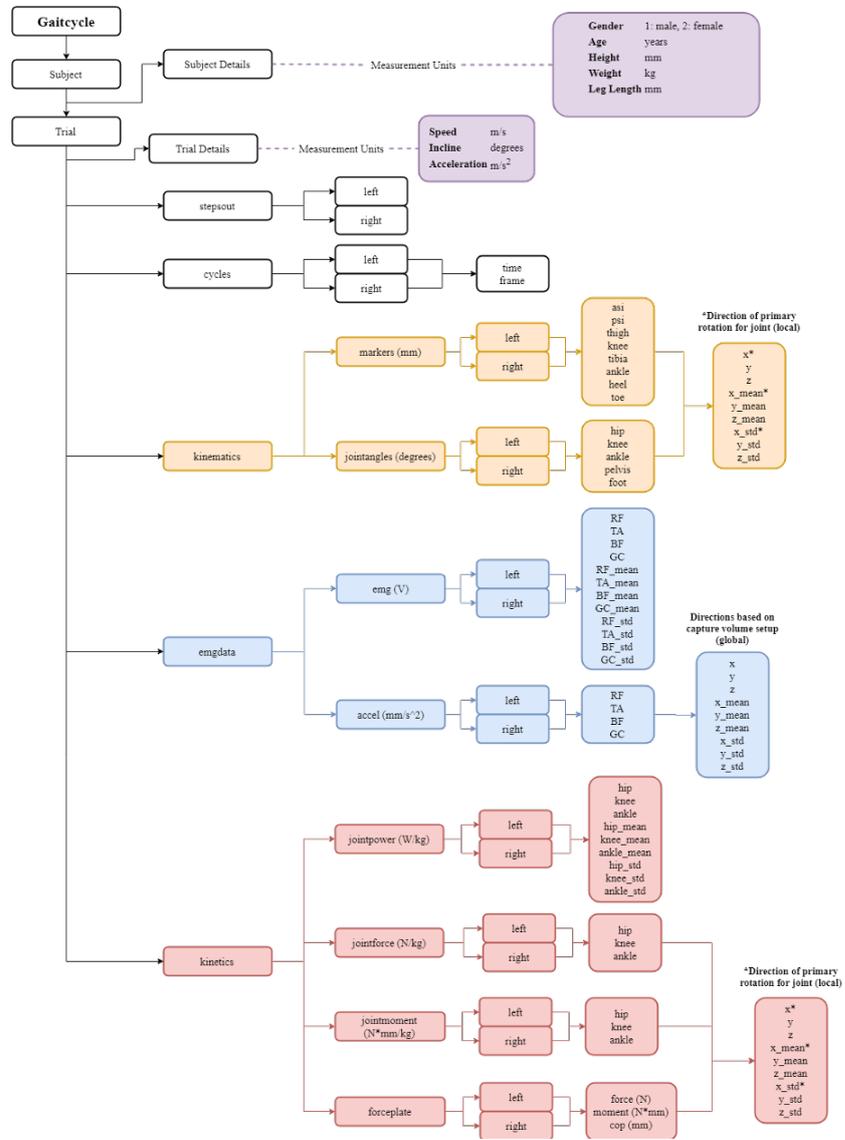


Figure 5.2: Gaitcycle struct Data Hierarchy

# Chapter 6

## Simulation on Matlab

In collaboration with the study made by [21], the first step to do, at this point, is to generate the necessary input variables for the kinematic cycle. These variables are CoM, meta trajectories (the projection of the ankle on the floor), feet angles and trunk angles w.r.t. the ground and their respective speeds.

The first stage of the reference variables generation process consists in obtaining of the ZMP trajectory from the gait and the feet and steps characteristics. Then there are four events determined from the feet markers contained in the experimental dataset, that determined the motion of the meta of the feet. From the ZMP trajectory, according to the LIPM model theory, were computed the CoM position and speed.

Finally, the trajectories and speeds variables were packed together to be used in the kinematics cycle.

The data used for this simulation are imported from the subject AB01 (1 m/s of treadmill's speed, treadmill with 0 inclination).

It is very important to establish the support point (heel and toe) and the support leg (left or right) at each moment of the trial.

There are two states variables: the *stance leg* (right or left) and the *stance point* (heel or toe). The alternation between these determines four states: left heel support, left tip support, right heel support, right tip support. The events that must occur in order to change state are:

- *The change of the support point:* when  $z_{\text{stancefootH}} \leq z_{\text{stancefootT}}$  the toe marker is at the same level or below the heel marker, the foot must be flat or the heel must be lifted w.r.t. the terrain. The support point is changing from the heel to the toe of the stance foot.
- *The change of the support leg:* if there is a local minimum of  $z_{\text{swingingfootH}}$  while standing on the toe of the opposite foot, it corresponds to a heel-strike of the

swinging foot. The support leg changes, as well the support point, and a step was completed.

The first state is determined by the different time plots of  $z_{\text{footpoint}}$ , and depends on which marker is lowest, while the others depend on events.

Sometimes the subject walks without placing the heel on the ground, especially at high walking speed. However  $z_{\text{swingingfootH}}$  has a local minima also in this cases, so a short period of heel support will always be accounted before switching to the tip support state, even if only of one sample. In addition the condition for event A makes possible to change support point even if the heel and the toe do not lie together on the ground for a moment.

It has been verified that the double support period corresponds to the time spent in heel support, while the single support period corresponds to the time passed in toe stance, in the case of a slow walk. Thus it is possible not only to determine the stance leg and the support point, but also if the subject is in double or single stance.

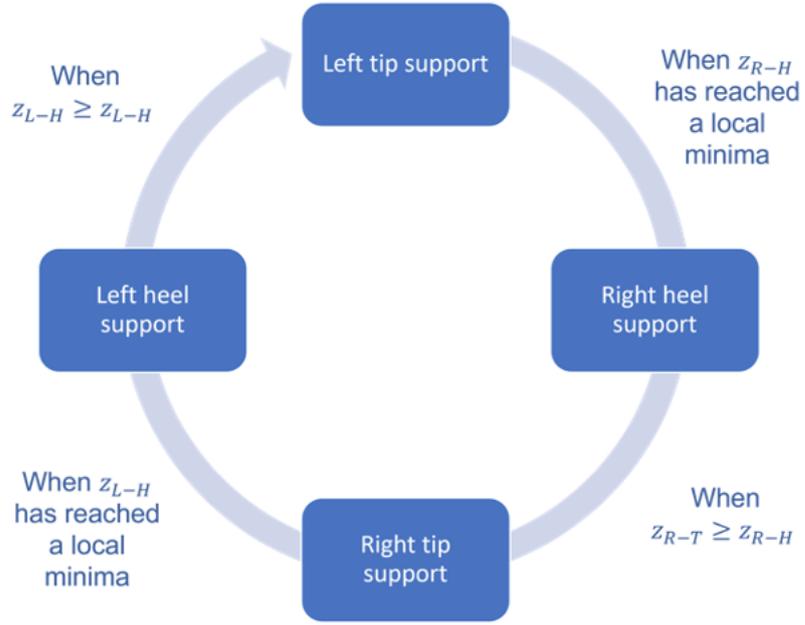
At the end of this process, is generated an array with the following information:

- Stance leg.
- Support point.
- Step number. Together with the support point it gives the support state.
- The initial instant of the time period spent in a certain state.
- The final instant of the time period spent in a certain state.
- An array providing the current state of stance, single or double for the whole period spent in a certain state.
- The time array of the period, starting from zero and ending at the final time less the initial time.

Using this array it is possible to segment the data according to the status of the media.

### *Evaluation of kinematic variables and body segment lengths*

In the gait analysis it is fundamental the evaluation of the antropometric data of the subject, and if it is conducted on a kinematic level, the information on the correct lengths of the body segments and the position of the joint centers are essential. Due to the lack of measurements on the lower limb segments, their length were



**Figure 6.1:** State machine representation

calculated from the markers, following the procedure given by the Vicon Plug-in Gait Reference guide [27]. The Newington-Cage [28] model is used to define the positions of the hip joint centers in the pelvis segment. The interAsis distance is computed as the value between the left ASI (LASI) and the right ASI (RASI) markers. The distances from the Asis to the Trocanter are calculated independently for each leg, using the following formula:

$$AsisTrocanDist = 0.1288LegLength - 48.56 \quad (6.1)$$

The offsets vectors for the two hip joint centers (the left LHJC and the right RHJC) are calculated as follow:

$$X = C\cos(\theta)\sin(\beta) - (AsisTrocanDist + mm)\cos(\beta) \quad (6.2)$$

$$Y = -(C\sin(\theta) - aa) \quad (6.3)$$

$$Z = -C\cos(\theta)\cos(\beta) - (AsisTrocanDist + mm)\sin(\beta) \quad (6.4)$$

where:

$$C = 0.115\text{MeanLegLength}-15.3$$

$aa$  is the interAsis distance

$\theta$  is taken as 0.5 rad

$\beta$  is taken as 0.314 rad

For the right joint center, the Y is negated, since Y is in the lateral direction for the pelvis embedded coordinate system.

### *Outputs*

The extensions of the tibial segments, of the femur segments, of the plant of the feet and of the pelvic segment are calculated at each instant. The time-averaged length of each body segment was determined, and the standard deviation of each time sample.

To determine whether the values of the joint offsets have been chosen correctly, the standard deviations must not exceed a certain threshold: this would mean that the lengths vary excessively over time. In any case, it is impossible to obtain the time invariant length of the body segments, because of the same measurement methodology. The motion capture system may not measure the position of the markers correctly, and markers may shift during walking. There will always be a few values that deviate significantly from the average, even in the best situation. As output to this process we have the data obtained from the state machine and the average values of the body segments.

Joint	Joint Offset
Left KJC	40 mm
Right KJC	40 mm
Left AJC	36 mm
Left AJC	35 mm

**Table 6.1:** Chosen center offsets for the subject AB01 and trial sli0

## 6.1 Cartesian variables generation

The output from the event based data were used to compute the reference variables needed for the kinematic analysis. Our reasoning starts from the paper of SangJoo Known et al. [7] in which it is explained how to adapt the generation of the trajectories of the ZMP and feet developed by Choi et al. [29] to produce human-like trajectories instead. Then the trajectory of the ZMP is used to reconstruct the

Body Segment	Time-Averaged Length
Pelvis	187 mm
Right Femur	409 mm
Left Femur	400 mm
Right Tibia	425 mm
Left Tibia	448 mm
Right feet plant	250 mm
Left feet plant	245 mm
Right feet dorso	187 mm
Left feet dorso	174 mm
Right AJC-meta distance	39 mm
Left AJC-meta distance	32 mm

**Table 6.2:** Time-averaged extension of the body segments

COG pattern, according to the LIPM modeling. The remaining reference variables needed for the kinematic analysis, i.e. trunk angles w.r.t. the ground and knee joint angles, are taken directly from the experimental data.

### 6.1.1 ZMP and feet trajectories generation

In order to describe the trajectories of the ZMP and the meta of the feet, a series of variables obtained from the outputs of the state machine and the markers' data are required. First of all, the incomplete steps were deleted from the recordings, to start and end the trial with a flat-foot event. Then, starting from the first flat-foot event of the trial, the following parameters are found iteratively:

- $stepN$ : the step number, assigned from supportinfos, deleting the first incomplete step.
- $t_{to}$ : time between flat-foot and toe-off events of the support foot.
- $t_{hs}$ : time between toe-off and the heel-strike events of the support foot.
- $t_{ff}$ : time between heel-strike and flat foot events of the support foot.
- $\Delta x_{step}$ : the longitudinal distance between the meta of the feet, calculated at the flat-foot event.
- $\Delta y_{step}$ : the lateral distance between the meta of the feet, calculated at the flat-foot event.
- $l_f$ : the longitudinal distance between the markers of the meta and of the toe of the supporting foot during the flat-foot event.

- $l_b$ : the longitudinal distance between the markers of the meta and of the heel of the supporting foot during the flat-foot event.

The first six parameters vary with each step, while the last two  $l_f$  and  $l_b$  are constant, because they depend only on the length of the foot.

*ZMP generation from the support states*

In flat-footed walking, the ZMP remains fixed in the middle of the sole of the supporting foot until it abruptly moves to the opposite foot during the double stance phase. However, in human walking the ZMP always travels from the heel to the toe of the supporting foot, and changes foot more gradually. The ZMP longitudinal trajectory between two flat-footed events occurring at time  $t_i$  and time  $t_f$  respectively is given by:

$$\text{ZMP}_x(t) = \begin{cases} \text{ZMP}_x(t_i) + (l_f/t_{ff})t & \text{for } t_i < t \leq t_i + t_{ff} \\ \text{ZMP}_x(t_i) + (l_b/t_{to})t - l_b + \Delta(x_{\text{step}}) & \text{for } t_i + t_{ff} < t \leq t_f \end{cases}$$

Instead in the lateral direction is given by:

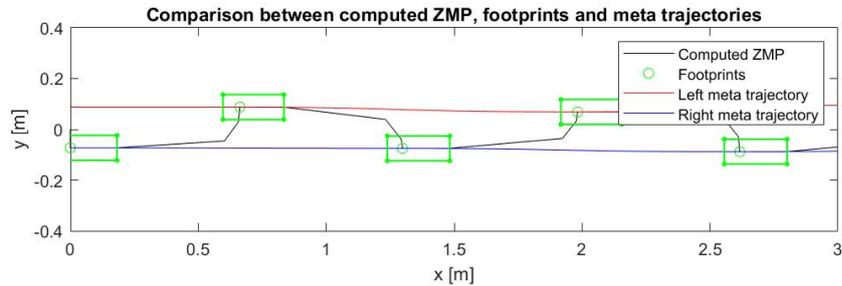
$$\text{ZMP}_y(t) = \begin{cases} \Delta(y_{\text{step}}) & \text{for } t_i < t \leq t_i + t_{ff} \\ (K_f - K_y)(t/t_{to}) - K_f & \text{for } t_i + t_{ff} < t \leq t_f \end{cases}$$

The parameter  $K_y$  is defined at each steps and is calculated as:

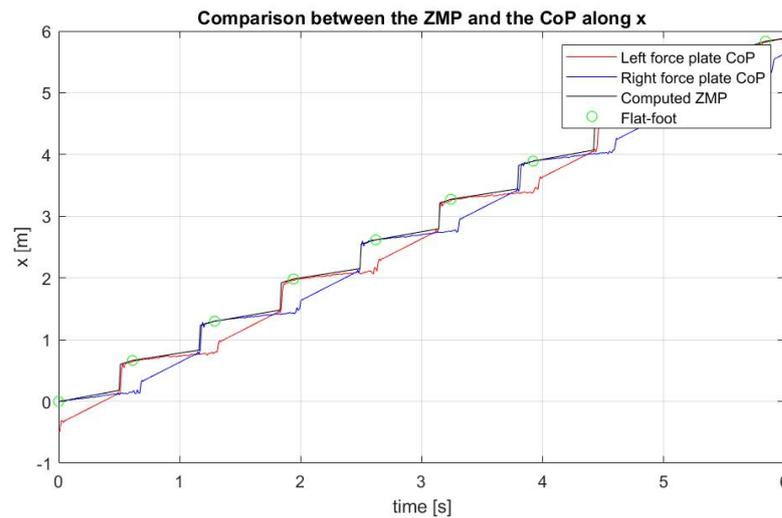
$$K_y = \frac{\Delta(y_{\text{step}}\omega_n \tanh(\omega_n)(t_{to}))}{(+1 + \Delta(y_{\text{step}}\omega_n \tanh(\omega_n)(t_{to})))+} \quad (6.5)$$

where  $\omega_n = g/z_{\text{CoG}}$  is the natural frequency of the inverted pendulum that from the ZMP (its base) to the CoM. This formula was derived from the CoG trajectory constrain along the lateral direction defined by Choi et al. in [29]. The parameter  $K_f$  is equal to  $K_f$  calculated for the next step. It can be noticed that the time difference between  $t_i$  and  $t_f$  (i.e. the time between two flat-foot events) is equal to the sum of  $t_{ff}$  and  $t_d$ .

The comparison shows that the calculated ZMP has a similar trend to that of the CoP in the longitudinal direction  $x$  and they almost overlap completely. In the lateral direction  $y$ , however, the situation is different: the lines parallel to the time axis of the ZMP and those of the CoP corresponding to the single support phase do not coincide. This occurs for two reasons: during the double stance phase, force



**Figure 6.2:** Comparison between computed ZMP footprints and meta trajectories

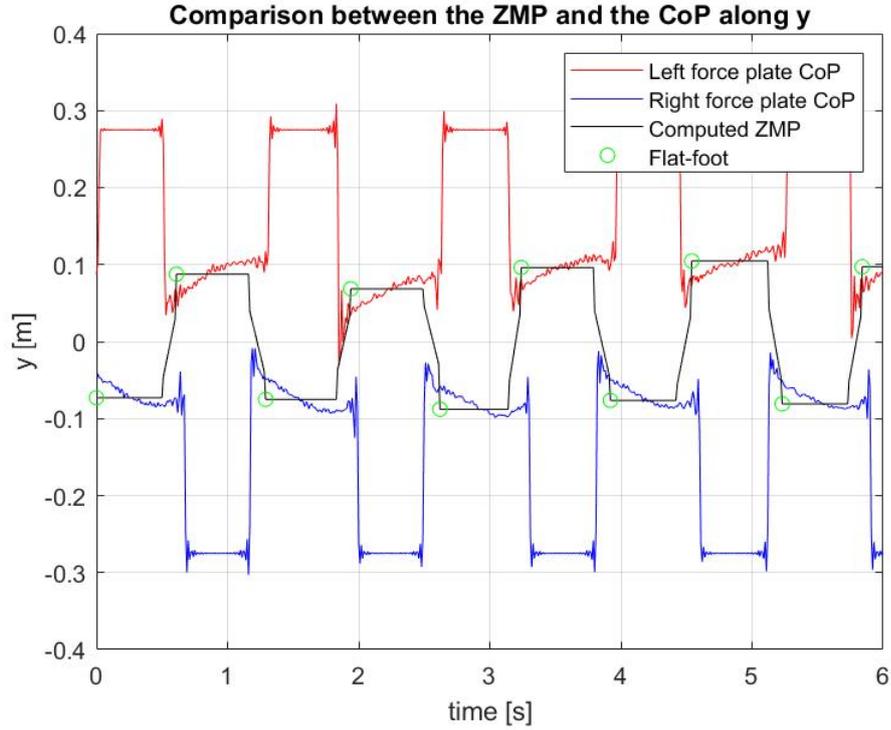


**Figure 6.3:** Comparison between computed ZMP and the CoP along x

platforms are unable to measure the displacement of the CoP from one foot to the other, and during the single stance phases, the CoP moves from the inner part of the sole to the outer part due to the pronation of the stance foot.

The double stance phases correspond to the intervals between a heel-strike event and a flat foot event of the same foot.

The generated trajectory of the ZMP can be seen as a continuous piecewise linear functions. Its break-points have been recorded to be used in the CoG generation



**Figure 6.4:** Comparison between computed ZMP and the CoP along  $y$

process to compute the angular coefficients of the ZMP linear traits.

### *Outputs*

The outputs of the processes defined previously in the section are the ZMP trajectory, the meta trajectories and the angles of the feet w.r.t. the ground.

To these variables a first trait was added to simulate the transition from standing with both feet touching the ground to the first half-step. This ensures that the biped starts from a stable position.

The recorded ZMP break-points are collected to be used for reconstructing the continuous piece-wise linear function which matches the ZMP trajectory.

## 6.1.2 CoG computation

The CoG motion can be derived from the ZMP as stated by Kajita et al. [30]. He proposed to solve numerically an infinite-horizon LQR problem to stabilize the ZMP, optimizing the quadratic cost of the ZMP tracking against the CoG acceleration. The weakness of this method is the high computation time. For this reason, the method of Russ Tedrake et al. [31], which proposes an iterative method

for finding solutions to the LQR problem, was preferred for calculating the CoG trajectory and verifying the proper ZMP tracking at the same time. The CoM and ZMP dynamics of a legged rigid body systems can be written in the state space form as:

$$\dot{x} = Ax + Bu = \begin{bmatrix} 0_{2 \times 2} & I_{2 \times 2} \\ 0_{2 \times 2} & 0_{2 \times 2} \end{bmatrix} x + \begin{bmatrix} 0_{2 \times 2} \\ I_{2 \times 2} \end{bmatrix} u \quad (6.6)$$

$$\dot{y} = Cx + Du(x, u) = \begin{bmatrix} I_{2 \times 2} & 0_{2 \times 2} \end{bmatrix} x + \frac{-z_{\text{com}}}{\ddot{z}_{\text{com}}} I_{2 \times 2} u \quad (6.7)$$

where  $x = [x_{\text{CoG}}, x_{\dot{\text{CoG}}}, \dot{x}_{\text{CoG}}, \dot{x}_{\dot{\text{CoG}}}]^T$ ,  $u = [\ddot{x}_{\text{CoG}}, \ddot{x}_{\dot{\text{CoG}}}]^T$ ,  $y = [x_{\text{ZMP}}, x_{\dot{\text{ZMP}}}]^T$ ,  $g$  the constant gravitational acceleration and  $z_{\text{CoM}}$  the center of gravity vertical position. Assuming a constant CoG height  $z_{\text{CoG}}$  the terms  $D(x, u)$  becomes  $D(u)$ . The desired ZMP trajectories,  $y_d(t)$  can be described by a continuous piecewise polynomial of degree  $k$  with  $n$  breaks at  $t_j$  (with  $t_0=0$  and  $t_n=t_f$ ).

Given the desired trajectory of the ZMP  $y_d(t)$  the optimal ZMP tracking controller can be obtained solving a continuous-time LQR problem.  $Q$  and  $R$  explicitly trade off ZMP tracking performance against the cost of accelerating the CoG. The final LQR problem with a cost on state in coordinates relative to the final conditions is:

$$\begin{aligned} & \underset{u(t)}{\text{minimize}} \int_0^{\text{inf}} (||y(t) - y_d(t)||^2_Q + ||u(t)||^2_R) dt \\ & \text{subject to} \quad Q = Q^T > 0 \\ & \quad \quad \quad R = R^T > 0 \\ & \quad \quad \quad y_d(t) = y_d(t_f), \forall t \geq t_f \\ & \quad \quad \quad \dot{x}(t) = Ax(t) + Bu(t) \\ & \quad \quad \quad y(t) = Cx(t) + Du(t) \end{aligned}$$

where:

$$g(\bar{x}(t), u(t)) = \bar{x}^T Q_1 \bar{x} + \bar{x}^T q_2(t) + q_3(t) + u(t)^T R_1 u(t) + u(t)^T r_2(t) + 2\bar{x}^T(t) N u(t) \quad (6.8)$$

$$Q_1 = C^T Q C \quad q_2(t) = -2C^T Q \bar{y}_d(t) \quad q_3 = \|\bar{y}_d(t)\|_Q^2 \quad (6.9)$$

$$R_1 = R + D^T Q D \quad r_2(t) = -2D Q \bar{y}_d(t) \quad N = C^T Q D \quad (6.10)$$

The optimal cost-to-go for this problem has the general form:

$$J(\bar{x}(t), t) = \bar{x}^T(t) S_1(t) \bar{x}(t) + \bar{x}^T s_2(t) + s_3(t) \quad (6.11)$$

The optimal controller is defined as:

$$u^*(t) = -R_1^{-1} (N_B \bar{x}(t) + r_s(t)) \quad (6.12)$$

where  $N_B = N^T + B^T S_1$  and  $r_s = (1/2)(r_2(t) + B^T s_2)$  and  $S_1, s_2$  and  $s_3$  terms are computed via the Riccati differential equation. After some considerations the optimal feedback controller can be expressed as:

$$u^*(t) = K_1 \bar{x}(t) + k_2(t) \quad (6.13)$$

where the feedback matrix  $K_1$  is a constant and:

$$k_2(t) = -R_1^{-1} \left( \frac{1}{2} B^T s_2(t) - D Q \bar{y}_d(t) \right) \quad (6.14)$$

The solution to this systems, as in the above has the general form:

$$z(t) = e^{A_z(t-t_j)} a_j + \sum_{i=0}^k b_{j,i} (t - t_j)^i \quad (6.15)$$

The following algorithm solves for the coefficients of 6.15 forward in time.

**Data:**  $x(0)$ ,  $A_z$ ,  $B_z$ , degree  $k$  piecewise polynomial  $\bar{y}_d(t)$  with  $n$  breaks

**Result:**  $a_j$ ,  $b_{j,i}$ ,  $\forall j \in (1, \dots, n)$ ,  $\forall i \in (0, \dots, k)$

**for**  $j = n, \dots, 1$  **do**

$\beta_{j,k} = -A_z^{-1} B_z c_{j,k}$ ;

**for**  $i = k - 1, \dots, 0$  **do**

$b_{j,i} = A_z^{-1} ((i + 1)b_{j,i+1} - B_z c_{j,i})$ ;

**end**

$a_j = \begin{bmatrix} x - b_{j,1} \\ \alpha_j \end{bmatrix}$ ;

$x = \begin{bmatrix} I \\ 0 \end{bmatrix} e^{A_z(t_{j+1}-t_j)} a_j + \sum_{i=0}^{k-1} b_{j,i} (t_{j,i} - t_i)^i$ ;

**end**

$b_{j,1}[1 : 2] = b_{j,1}[1 : 2] + y(t_f)$ ;

### Outputs

The process outputs are the optimal input  $u$ , the output  $y$  and the state variables  $x$  of the state space system of 6.6. Using the ZMP generated from the support states and the LQR gains in 6.16 in the plots in 6.5 and 6.6 are obtained.

$$Q = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix} \quad R = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix} \quad (6.16)$$

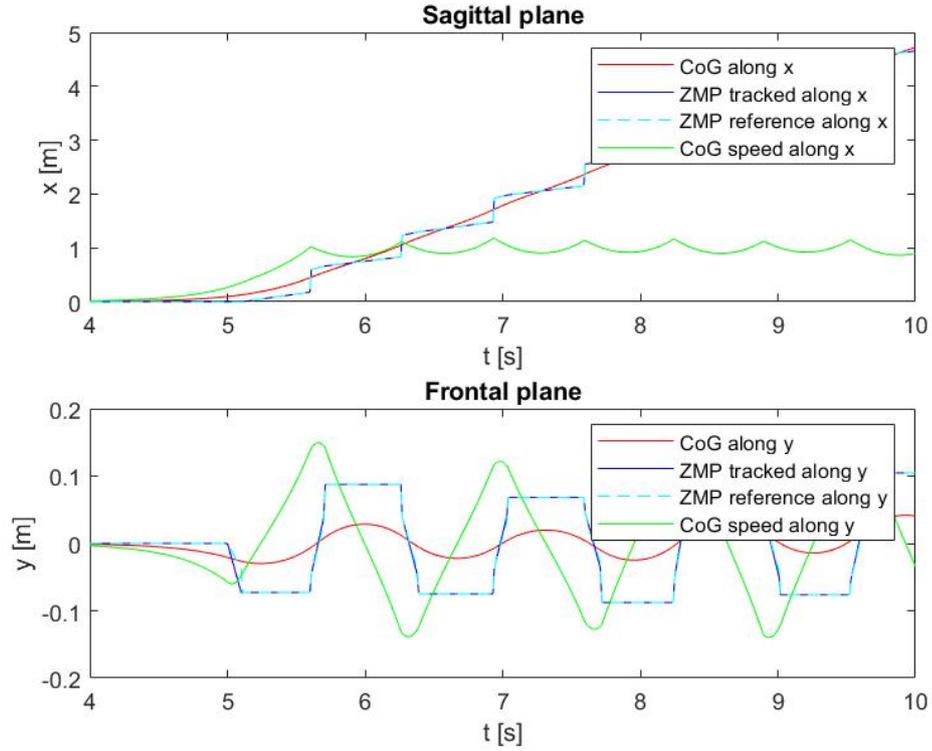
The reference ZMP was perfectly tracked thanks to the use of a rather large time horizon of about sixty seconds. The longer the reference ZMP, the more accurate the tracking.

The motion of the CoG is stable, and never moves outside the line traced by the ZMP, except during the first half step. Along the longitudinal direction its trajectory seems a straight line, while in the lateral a sine wave which peaks approach the ZMP during the single support phase.

The velocity of the CoG along  $x$  starts from zero, increases, and then begins to oscillate around a constant value, equal to the trial's walking speed, reported in the experimental data. Instead the CoG's speed along  $y$  is represented by a triangular wave with rounded edges, and reach its peak during the double support phase, when the position of the CoG pass from a side the other.

## 6.2 From variables to the model on the kinematic cycle

The kinematic with redundancies guarantees the WBC of the model, and is structured take into account together the Cartesian variables related to the balanced of



**Figure 6.5:** Comparison of the reference and tracked ZMP with the generated CoG along x,y

the model while walking, and the angular variables of the joints to guarantee the synchronous movement of the biped with the experimental test data.

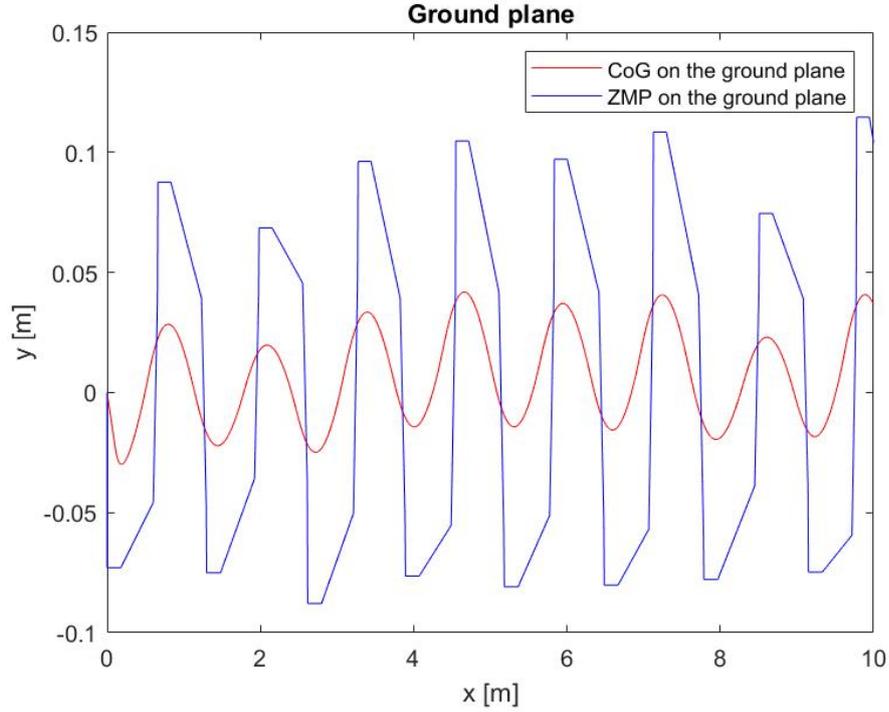
### 6.2.1 Direct Kinematics

The direct kinematics is a function  $K(\theta)$  that relates the linear and angular displacements of the model in the Cartesian space to its configuration, i.e. the set of its joint angles  $\theta$ .

$$x = K(\theta) \quad (6.17)$$

Direct kinematics creates a link between joint space variables  $\theta$  and Cartesian space variables  $x$ .

Consider two reference systems  $RF_0$ ,  $RF_1$ , and an arbitrary point P in space. Let  $p_0$  be the vector of coordinates of P w.r.t. the reference frame  $RF_0$ . Instead let  $p_1$  be the vector representing the position of P w.r.t.  $RF_1$ . Let  $d_{(0 \rightarrow 1)}$  be the distance



**Figure 6.6:** Comparison between CoG and ZMP on the xy ground plane

of the origin of  $RF_1$  from the origin of  $RF_0$ , and  $R_{(0 \rightarrow 1)}$  be the rotation matrix of  $RF_1$  with respect to  $RF_0$ . The position  $p_0$  of P in  $RF_0$  is given by:

$$p_0 = d_{(0 \rightarrow 1)} + R_{(0 \rightarrow 1)}p_1 \quad (6.18)$$

Hence, 6.18 represents the coordinate transformation (translation + rotation) of a bound vector between two frames.

After the computation of the homogeneous transformation matrix  $T_{(0 \rightarrow 1)}$ , that gives us all the information needed to describe the direct kinematics: it is a suitable instrument for describing the orientation and position of a body in space w.r.t a reference frame, in terms of its ZYX angles  $\phi$  (calculated from R) and its translation vector p, giving all the Cartesian space variables x:

$$x = \begin{bmatrix} \theta \\ p \end{bmatrix} = \begin{bmatrix} \psi \\ \theta \\ \phi \\ p_x \\ p_y \\ p_z \end{bmatrix} \quad (6.19)$$

Then is computed the homogeneous transformations of the reference frames of the meta of the supporting foot ( $F_1$ ), of the model's CoG (CoG) and of the meta of the swinging foot ( $F_2$ ). These are obtained by the product of a series of homogeneous transformations starting from the base frame (i.e. the Cartesian reference system) to the corresponding point:

$$\begin{aligned} T_{0 \rightarrow F_1} &= T_{0 \rightarrow 1} T_{1 \rightarrow 2} T_{2 \rightarrow 3} T_{3 \rightarrow F_1} \\ T_{0 \rightarrow CoG} &= T_{0 \rightarrow 3} T_{3 \rightarrow 4} T_{4 \rightarrow 5} T_{5 \rightarrow 6} T_{6 \rightarrow 7} T_{7 \rightarrow 8} T_{8 \rightarrow 9} T_{9 \rightarrow CoG} \\ T_{0 \rightarrow F_2} &= T_{9 \rightarrow 10} T_{10 \rightarrow 11} T_{11 \rightarrow 12} T_{12 \rightarrow 13} T_{13 \rightarrow 14} T_{14 \rightarrow 15} T_{15 \rightarrow F_2} \end{aligned} \quad (6.20)$$

Given a model such as those defined in the previous section, we consider fixed the support point of the foot until it is changed. In each joint and point of interest there's a local frame, like described in the previous section. The value 0 indicates the reference frame located on the support point and attached to the terrain, while the other numbers indicates the other reference frames located on the joints. Each of these, except the last of each formula in 6.20 ( $T_{(3 \rightarrow F_1)}$ ,  $T_{(9 \rightarrow CoG)}$ ) which are constant transformation,  $T_{(15 \rightarrow F_2)}$ , depends on the  $i$ -th joint angle. Thus the position of  $F_1$  depends only on the angles of the support foot w.r.t. the ground. Instead  $F_2$  and CoG position depends also on the other joint variables.

### 6.3 Differential inverse kinematics

The differential direct kinematics gives us the relationship between the joint velocities and the linear and angular velocities of the various points forming the model. It is expressed by the following formula:

$$\dot{x} = J_A(\theta)\dot{\theta} \quad (6.21)$$

where  $J_A$  is the analytical Jacobian and:

$$J_A(\theta) = \frac{\partial K(\theta)}{\partial \theta} \quad (6.22)$$

The matrix  $J_A(\theta)$  can be obtained from the partial derivatives of the ZYX angles  $\phi$  and of the vector  $p$ , coming from both the homogeneous transformation representing the direct kinematics, w.r.t. the joint angles  $\theta$ . Given a set of three linear and three angular velocities ( $\dot{p}$  and  $\dot{\phi}$ ) to be computed, the analytical Jacobian is a 6x6 matrix such that:

$$\dot{x}_{ref} = \begin{bmatrix} \dot{\phi} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} J_\phi(\theta) \\ J_p(\theta) \end{bmatrix} = J_A(\theta)\dot{\theta} \quad (6.23)$$

This formula represents the linear relationship between the velocities in Cartesian space and those at the joints. In our case, we want to track precisely the motion of

the CoG, of  $F_1$  and of  $F_2$ , the orientation and position of their reference frames in space, and how their time evolution. Their Cartesian linear and rotational velocities are grouped within the vector  $\dot{x}_{ref}$ , the vector of the reference speeds. The differential direct kinematic that gives the set of reference variables  $x_{ref}$  can be written as follows:

$$\dot{x}_{ref} = \begin{bmatrix} \dot{\phi}_{CoG} \\ \dot{p}_{CoG} \\ \dot{\phi}_{F_1} \\ \dot{\phi}_{F_2} \\ \dot{p}_{F_2} \end{bmatrix} = J_{A(15 \times 15)}(\theta) \begin{bmatrix} \dot{\theta}_{1-F_1} \\ \dot{\theta}_{2-F_1} \\ \dot{\theta}_{3-F_1} \\ \dot{\theta}_{1-Ankle_1} \\ \dot{\theta}_{2-Ankle_1} \\ \dot{\theta}_{Knee_1} \\ \dot{\theta}_{1-Hip_1} \\ \dot{\theta}_{2-Hip_1} \\ \dot{\theta}_{3-Hip_1} \\ \dot{\theta}_{3-Hip_2} \\ \dot{\theta}_{2-Hip_2} \\ \dot{\theta}_{1-Hip_2} \\ \dot{\theta}_{Knee_2} \\ \dot{\theta}_{2-Ankle_2} \\ \dot{\theta}_{1-Ankle_2} \end{bmatrix} \quad (6.24)$$

where:

$$J_{A(15 \times 15)}(\theta) = \begin{bmatrix} J_{\phi CoG}(\theta) \\ J_{p CoG}(\theta) \\ J_{\phi F_1}(\theta) \\ J_{\phi F_2}(\theta) \\ J_{p F_2}(\theta) \end{bmatrix} \quad (6.25)$$

To obtain the redundancy of constrains, the number of degree of freedom has to be lower than the reference variables. Constrain can be added in both joint space and Cartesian space. The knee angles were added on the left side of Eq. 6.24 as

additional reference variable, obtaining:

$$\begin{bmatrix} \dot{\phi}_{CoG} \\ \dot{p}_{CoG} \\ \dot{\phi}_{F_1} \\ \dot{\phi}_{F_2} \\ \dot{p}_{F_2} \\ \dot{\theta}_{Knee_1} \\ \dot{\theta}_{Knee_1} \end{bmatrix} = J_{A(17 \times 15)}(\theta) \begin{bmatrix} \dot{\theta}_{1-F_1} \\ \dot{\theta}_{2-F_1} \\ \dot{\theta}_{3-F_1} \\ \dot{\theta}_{1-Ankle_1} \\ \dot{\theta}_{2-Ankle_1} \\ \dot{\theta}_{Knee_1} \\ \dot{\theta}_{1-Hip_1} \\ \dot{\theta}_{2-Hip_1} \\ \dot{\theta}_{3-Hip_1} \\ \dot{\theta}_{3-Hip_2} \\ \dot{\theta}_{2-Hip_2} \\ \dot{\theta}_{1-Hip_2} \\ \dot{\theta}_{Knee_2} \\ \dot{\theta}_{2-Ankle_2} \\ \dot{\theta}_{1-Ankle_2} \end{bmatrix} \quad (6.26)$$

where:

$$J_{A(17 \times 15)}(\theta) = \begin{bmatrix} J_{\phi CoG}(\theta) \\ J_{p CoG}(\theta) \\ J_{\phi F_1}(\theta) \\ J_{\phi F_2}(\theta) \\ J_{p F_2}(\theta) \\ J_{AKnee_1} \\ J_{AKnee_2} \end{bmatrix} \quad (6.27)$$

The analytical Jacobian  $J_A$  passed from being square to being rectangular, with more rows than columns. When the Jacobian is square, it can be inverted to obtain the inverse differential kinematics. Instead if it is rectangular and the number of rows exceeds the number of columns, the inversion is not possible, and instead is used the left pseudo-inverse. When the number of constrains exceed the number of DOF, the inverse kinematics has no solution. Thus an approximate solution can be obtained using the weighted least square method. Multiplying both sides of 6.21 for a diagonal matrix  $W$  of weights we have:

$$W \dot{x}_{ref} = W J_A \dot{\theta} \quad (6.28)$$

The matrix  $W$  is a  $17 \times 17$  square matrix, with different weights for each of the reference variables. The weighted least-square method gives the pseudo-inverse of  $J_A$ :

$$\dot{\theta} = J_A^\dagger W \dot{x}_{ref} = (J_A^T W J_A)^{-1} J_A^T W \dot{x}_{ref} \quad (6.29)$$

The least square method gives the solution which minimizes  $\|W J_A(\theta)\dot{\theta} - \dot{x}_{ref}\|$  and minimizes  $\|W x_{ref}\|$ . The higher the value of a weight, the more the respective Cartesian speed variable given by the result of the least squares method  $\dot{\theta}$  will be similar to the respective reference variable, to the detriment of the variables with a lower weight. Solving the inverse kinematics with the computed pseudo-inverse we obtain the joint speeds:

$$\dot{\theta} = J_A^\dagger \dot{x}_{ref} \quad (6.30)$$

The joint angles can be obtained integrating the inverse differential kinematic solution  $\dot{\theta}$ .

#### *Fitting the data*

A series of kinematics cycles is used for fitting the experimental data to the biped model. The aim is to ensure that there is simultaneous tracking of the generated variables in the Cartesian space and of the joint angles experimental data [32]. The selected angles are those at the knees joints, but these can be replaced by different ones or others can be added to increase the constraints. As the number of reference variables is greater than the number of DOF, it is not possible for all of them to be tracked accurately. Thus, the weighted least squares method was used as described in the inverse kinematic section: it enables the choice of which variables should be tracked more precisely by giving them a higher weight.

The home configuration is thus determined, which will be used to calculate the first Jacobian  $J_A(\theta(s_0))$ . It can be noted that the upright position does not correspond to any of the various models defined above, as the feet touch the ground with the entire sole. If both feet of the model touch the ground, a closed kinematic chain is formed, whereas our models correspond to open kinematic chains. The first model selected was the one corresponding to the first support other than both feet lying on the terrain, and the home configuration was established using it.

A global reference system was positioned at the point where the CoG was projected onto the floor, with the longitudinal axis in the direction of the walk, the lateral axis in the direction of the right foot and the vertical axis in the direction of the CoG. The distances between the origins of the model's base frame and the global frame just defined are calculated.

The values of the elements of the diagonal matrices  $W$ ,  $G$  and  $G_i$  are then chosen. The first is the already mentioned matrix of weights used in the least squares method, the others will play a role in correcting the reference variables from their tracking error.

$W$ ,  $G$  and  $G_i$  are to be chosen for best results in fitting inverse kinematics with constrain redundancy.

The solution  $\dot{\theta}$  does not guarantee a perfect tracking of all reference speeds, but a better tracking of the variables with a high weight in  $W$ , at the expense of those with a low weight in  $W$ .

By integrating  $\dot{\theta}(s)$ , the next configuration  $\theta(s)$  is obtained. With the an apposite MATLAB command we obtain the matrices of the homogeneous transformation of the CoG,  $F_1$  and  $F_2$  reference frames w.r.t. the base frame. The homogeneous transformations can be used to determine the vector  $x(s)$  containing the Cartesian variables of interest.

It should be noted that  $\theta$  is not the same for all the four models, i.e. it differs according to the kinematic chain. A matrix containing the model configurations at each sample was defined for each model: depending on the support point we have  $\theta_{LH}$  (left heel support),  $\theta_{LT}$  (left tip support),  $\theta_{RH}$  (right heel support),  $\theta_{RT}$  (right tip support). This matrix is an  $n \times m$ , where  $n$  is the number of configurations variables and  $m$  is the number of samples. The first three values of a columns depend on the rotations on the support point. The value of  $\theta$  does not change when selecting a new model with the base frame located on a support point on the same side of the body ( $\theta_{LH} = \theta_{LT}$  and  $\theta_{RH} = \theta_{RT}$ ). Instead when the stance leg is swapped, the configuration vector changes along with the model. The three values of the configuration depending on the orientation of the new stance foot are calculated from the ZYX angles of its meta w.r.t. the base frame of the previous model, while the remainder are the same but with their order is reversed.

To make the Cartesian quantities converge in time to the values of the reference trajectories, a proportional integral loop is closed on the Cartesian positions. The reference speeds are corrected with the reaction:

$$\dot{x}_{ref}(s+1) = \dot{x}_{ref}(s) + G(x_{ref}(s) - x(s)) + G_i \int_0^t (x_{ref}(s) - x(s)) dt \quad (6.31)$$

where  $G$  and  $G_i$  are diagonal matrices of gains and  $(x_{ref} - x)$  is the tracking error. High gain values impose a higher error consideration, and therefore more severe correction. The presence of the integrating action is justified by the fact that the integration in time of the velocities leads to drifts in time.

When a model with a different support point than the previous one is selected, the distance to the global reference frame is recalculated, adding the distance between the new and old support point. The global frame distance is added to the variables in the local reference frame (the base frame), obtaining a representation in the global one.

When changing support leg, the angular velocities of the joints at the new support point are not available. These are obtained by deriving the values of the angles in time obtained previously for the new support foot.

## 6.4 Results

The simulation inputs are the reference speeds  $\dot{x}_{ref}$ , and the respective angles and position contained in  $x_{ref}$ . The reference velocities are the angular and linear

velocities of the CoM and of the feet, and the rotational speed of the knees. The torso angular speeds about the y-axis and the z-axis, and the knee angular speeds were derived directly from the experimental data. To complete the set of reference variables, the angular speeds of the feet around the axes x and z and those of the trunk around z are missing. These have been set to zero because no lateral oscillation of the feet due to a prone-supination action is wanted, and the rotations around z are small and cannot be considered while walking in a straight line. The values chosen for the gains (for the data coming from the trial s1i0, subject AB01) are as following:

$$\begin{aligned}
 W &= \text{diag}(10^2, 10^2, 10^2, 10^3, 10^3, 10^3, 10^3, 10^3, 10^3, 10^2, 10^4, 10^4, 10^3, 10^2, 10^9, 70, 70); \\
 G &= \text{diag}(2, 3.5, 3, 3, 1, 3, 0.5, 0.8, 1, 1, 1, 1, 1, 1, 5.3, 0.6, 0.6); \\
 G_i &= \text{diag}(0.1, 0.15, 0.4, 0.1, 0.1, 0.3, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.61, 0.01, 0.01);
 \end{aligned}$$

These have been chosen to ensure good tracking (the perfect is impossible) of all reference variables.

It was not possible track precisely the input angles are the knees and at the same time of the trajectories of the CoM and of the feet. Other gains were also tried, in an attempt to obtain a pattern of the knee angles similar to those in  $x_{ref}$  : unfortunately it was not possible to obtain a good result, and the other reference variables deviated significantly from those in the data. This could be caused by a discrepancy in the origin of some Cartesian variables w.r.t. others. As we already mentioned, some are obtained from the trial measurement while others are generated according to the LIMP model. When the variables from the dataset are valued by giving them high weights, this is to the detriment of the variables generated, and vice versa. Finally, with the resulting data, a real-time animation was developed to show the evolution of walking over time.

In general, the plots of reference trajectories and angles have two lines: one named "Reference", which refers to the simulation inputs, and one "Effective", which represents the actual movements made by the biped (i.e. the outputs of the fitting process). On the other hand, as regards the speed graphs, a third line, "Corrected reference", is added to the above-mentioned ones, representing the input speed corrected by the feedback in 6.31.

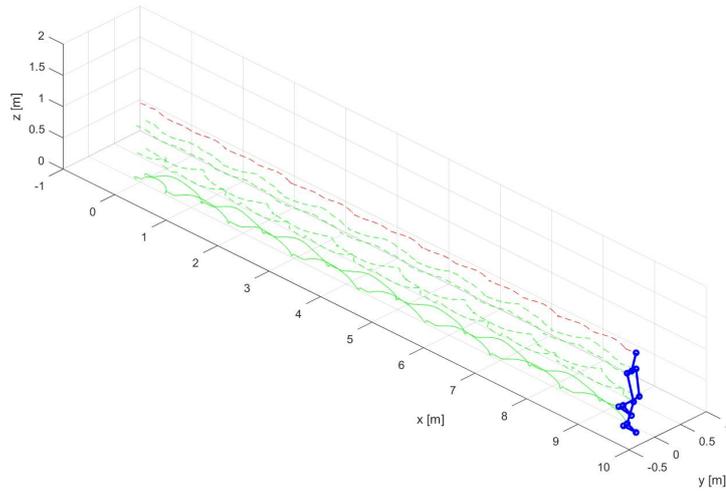


Figure 6.7: Biped walker animation

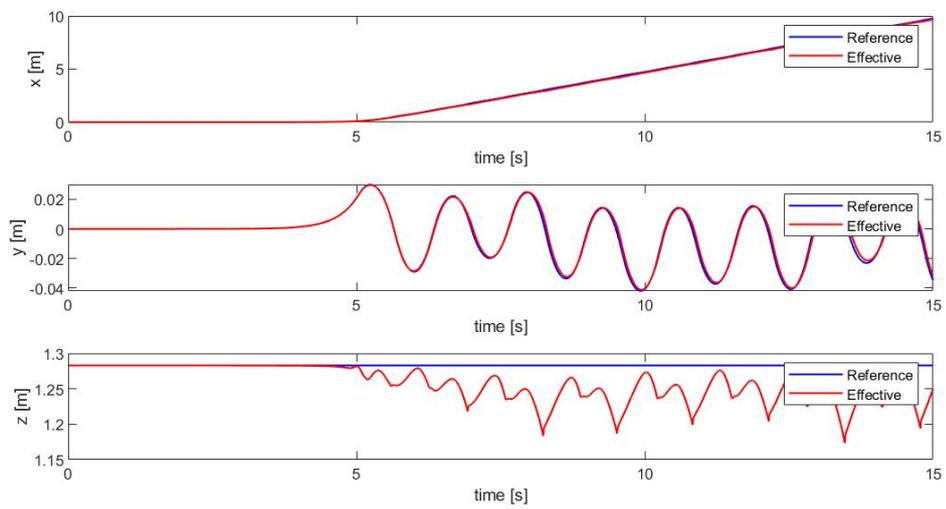
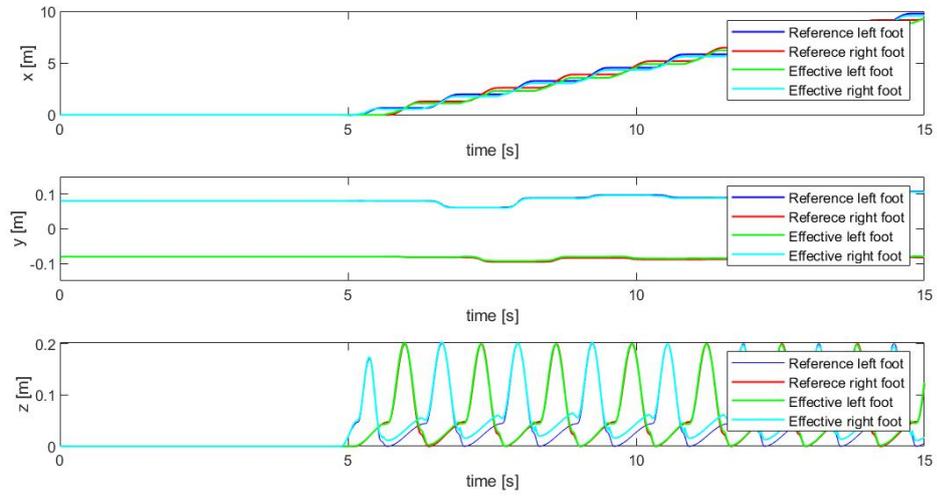
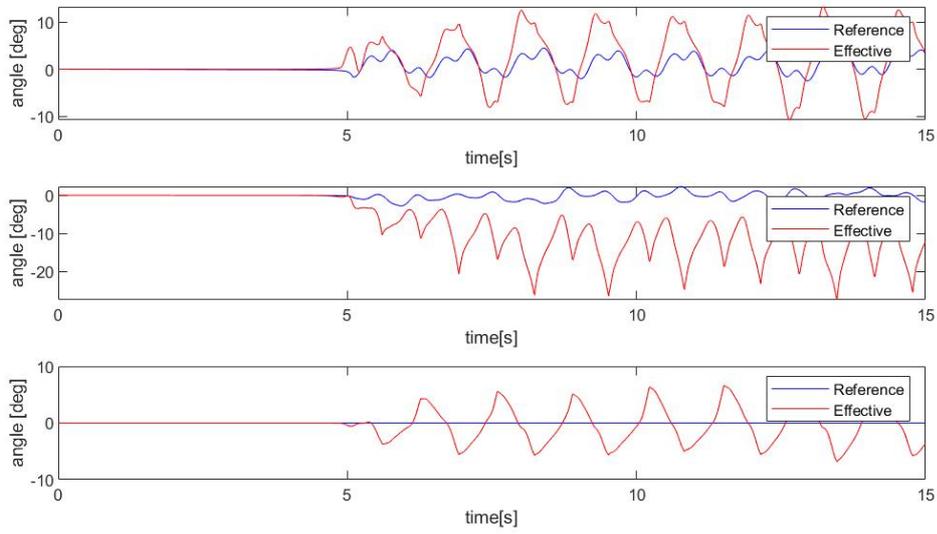


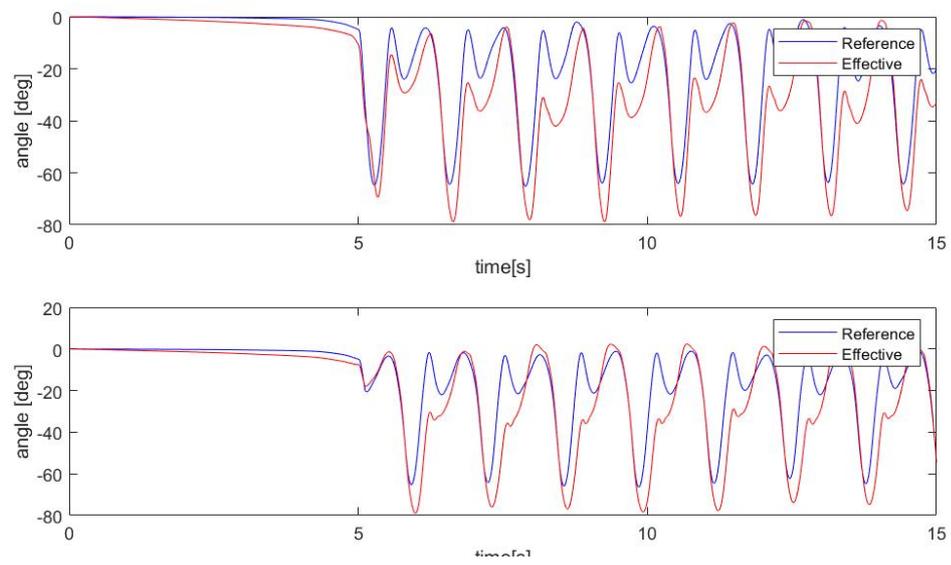
Figure 6.8: CoM trajectory along x-axis,y-axis,z-axis



**Figure 6.9:** Feet trajectories along x-axis,y-axis,z-axis



**Figure 6.10:** Trunk angles around x-axis,y-axis,z-axis



**Figure 6.11:** Left and right knee joints angles

# Chapter 7

## Simulation on CoppeliaSim

### 7.1 CoppeliaSim

The robot simulator CoppeliaSim, with integrated development environment, is based on a distributed control architecture: each object/model can be individually controlled via an embedded script, a plugin, a ROS or BlueZero node, a remote API client, or a custom solution. This makes CoppeliaSim very versatile and ideal for multi-robot applications. Controllers can be written in C/C++, Python, Java, Lua, Matlab or Octave [33]. CoppeliaSim is used for fast algorithm development, factory automation simulations, fast prototyping and verification, robotics related education, remote monitoring, safety double-checking, as digital twin, and much more [34].

#### 7.1.1 Design simulation

In order to create a good simulation using our prototype first we import as a Mesh the *Biped.stl* file, then we create the pure shapes from the mesh adding the joints in the correct position. The hierarchy, as can be seen from the figure 7.3 in the next page, has the following structure:

1. The *Base* is the pure shape associated to the *Trunk*
2. The *leftHipJointZ* is the joint associated to the rotation around Z-axes (same for the right side)
3. The *leftHipJointX* is the joint associated to the rotation around X-axes (same for the right side)
4. The *leftHipJointY* is the joint associated to the rotation around Y-axes (same for the right side)

5. The *leftUpperLeg* is the pure shape associated to the *lUpLeg* (same for the right side)
6. The *leftKneeJointY* is the joint associated to the rotation around Y-axes (same for the right side)
7. The *leftLowerLeg* is the pure shape associated to the *lLowLeg* (same for the right side)
8. The *leftAnkleJointX* is the joint associated to the rotation around X-axes (same for the right side)
9. The *leftAnkleJointY* is the joint associated to the rotation around Y-axes (same for the right side)
10. The *Spheres* between some joints connect the latter to each other (same for the right side)
11. The *leftFoot* is the pure shape associated to the *lFoot* (same for the right side)
12. The *leftFootPath*: a path is a pseudo object, representing a succession of points with orientation in space [35] (same for the right side)
13. The *leftFootTip* connected to the *leftFootTarget* they are both dummies. A dummy object is the simplest object available: it is a point with orientation, and it can be seen as a reference frame. Dummies are multipurpose or helper objects: they are used alone to identify specific points or reference frames in the scene, they are also used in pairs to specify loop closures or tip-target relationships for dynamics or kinematics calculations. [36] Linked dummy: dummy linked to this one. Linked dummies (easily recognizable by a colored segment linking them, in the scene hierarchy) have special properties and behavior. Link type: the link type will specify the behavior of the linked dummies during simulation. [37] See the figures below (same for the right side)

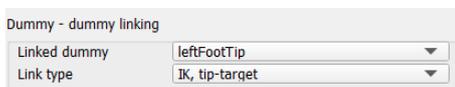


Figure 7.1: Dummy target

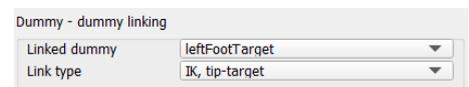


Figure 7.2: Dummy tip

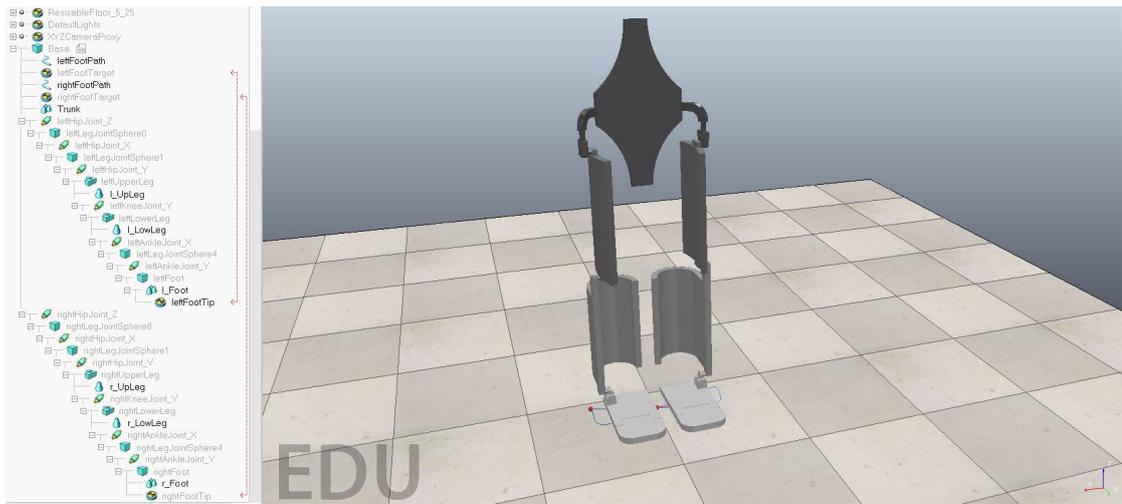
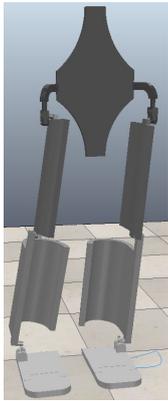


Figure 7.3: Hierarchy of the Biped

## 7.1.2 Walking analysis

In this simulation there are only three phases:

- *Phase 1.* In this phase the support leg is the left one (Single stance), Figure 7.4
- *Phase 2.* In this phase both the foot are in contact with the ground (Double stance), Figure 7.5
- *Phase 3.* In this phase the support leg is the right one (Single stance), Figure 7.6



**Figure 7.4:**  
Phase 1



**Figure 7.5:**  
Phase 2



**Figure 7.6:**  
Phase 3

### 7.1.3 Angle and position graphs

In this section are shown the trend of the biped and the angles of the joints over time. As first step in order to move the structure is to assign the trajectory to the feet using the function 'Path' in CoppeliaSim, the design of this path is equal to the one proposed by Matlab [38] shown in the following figure.

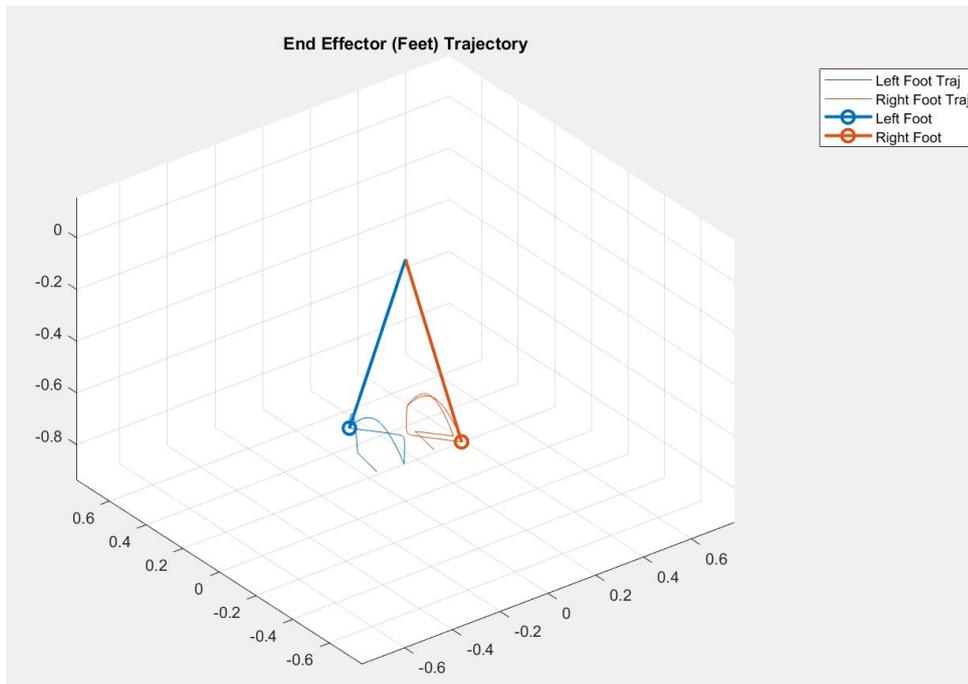


Figure 7.7: Matlab Feet Trajectory

The first graph shown is in the figure below. It is the representation of the position of the trunk along X and Y.

The Y position is just the forward movement of the biped.

The X position is sinusoidal, the center of the trunk is oscillating. The single stance is on the peaks instead the double stance is when it goes from one peak to another. This last trend is similar to the one proposed by Matlab [38] shown in the figure ??.

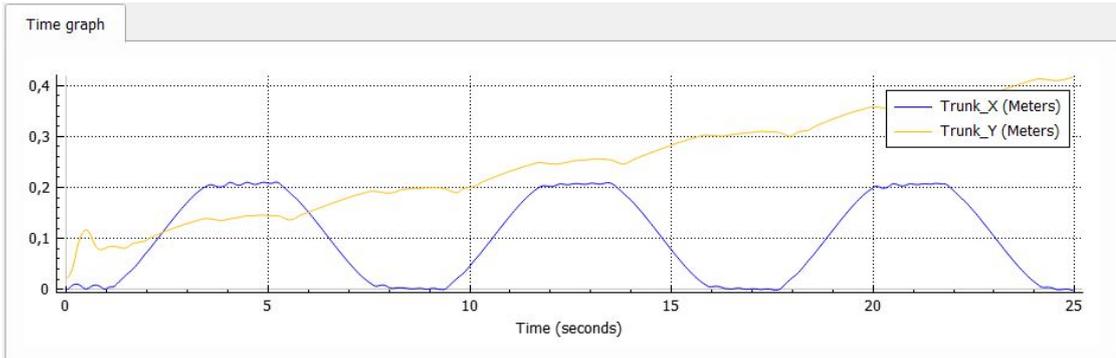


Figure 7.8: Trunk X and Y position

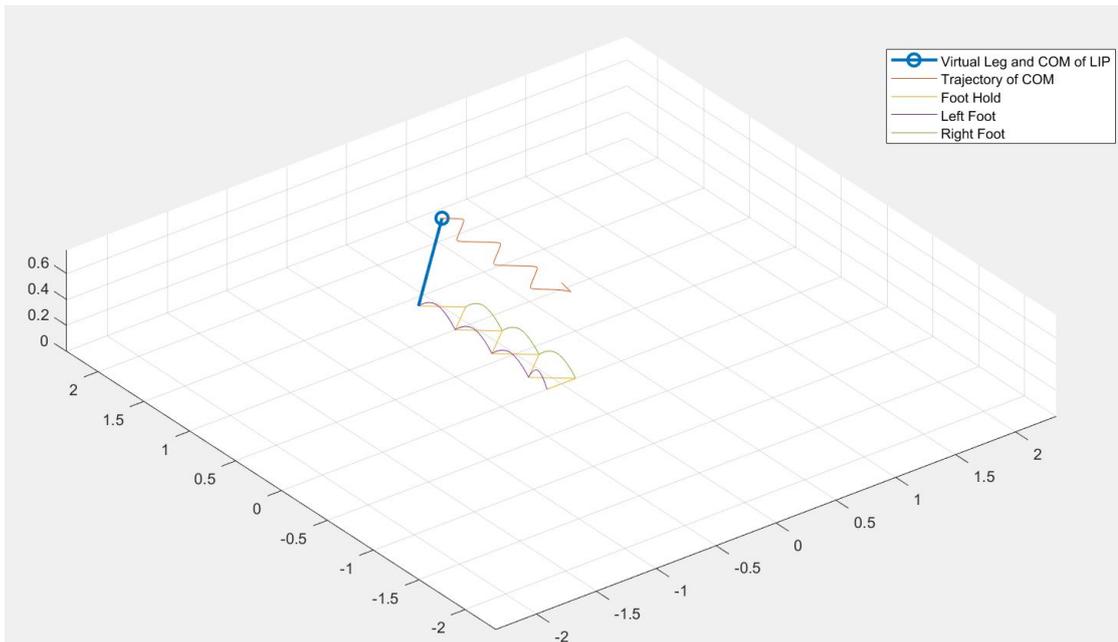


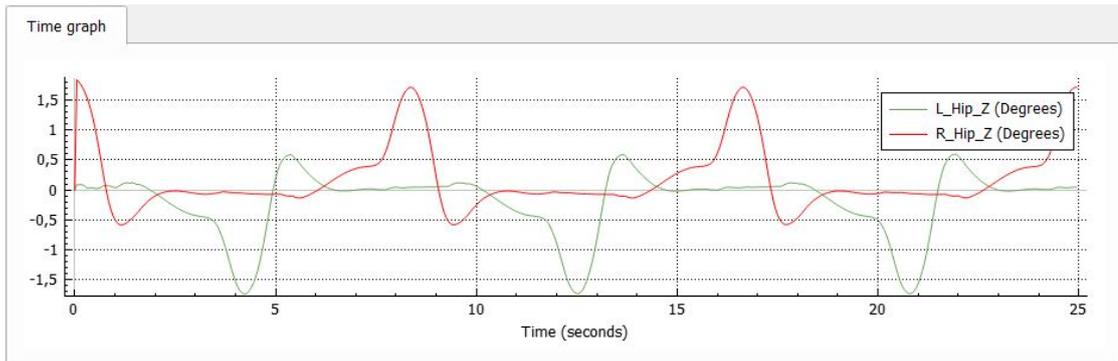
Figure 7.9: Trunk Trajectory Matlab

The following three figures represent the joints hip angles. In the biped there are three hip joints in each leg: Yaw, Roll and Pitch.

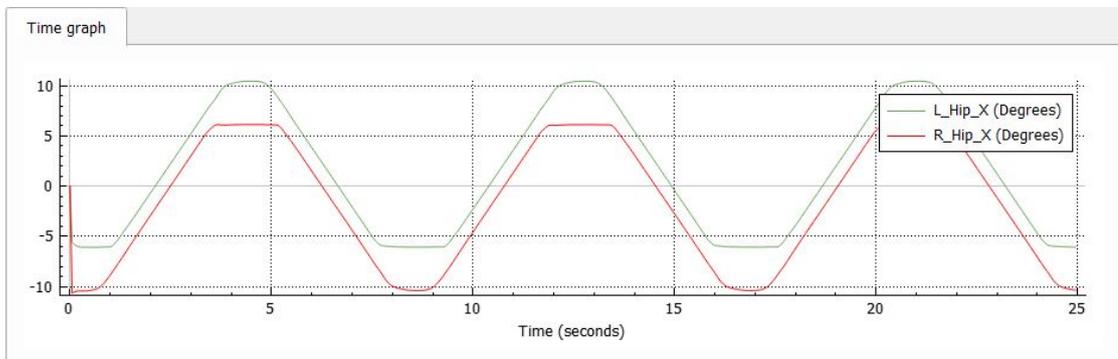
In the first figure there are the yaw angles, it can be seen that those angles are very small.

In the second figure the trend is similar to the trunk one.

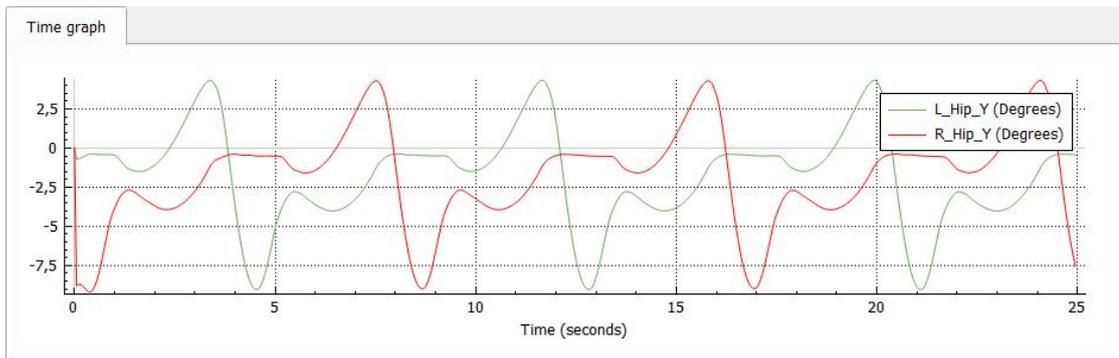
At the starting point of the simulation the support leg is the left one, this can be seen in the third figure. If the left leg is on the ground the right one is bent and the pitch joint create an angle.



**Figure 7.10:** Left and right hip joint angles (Yaw)

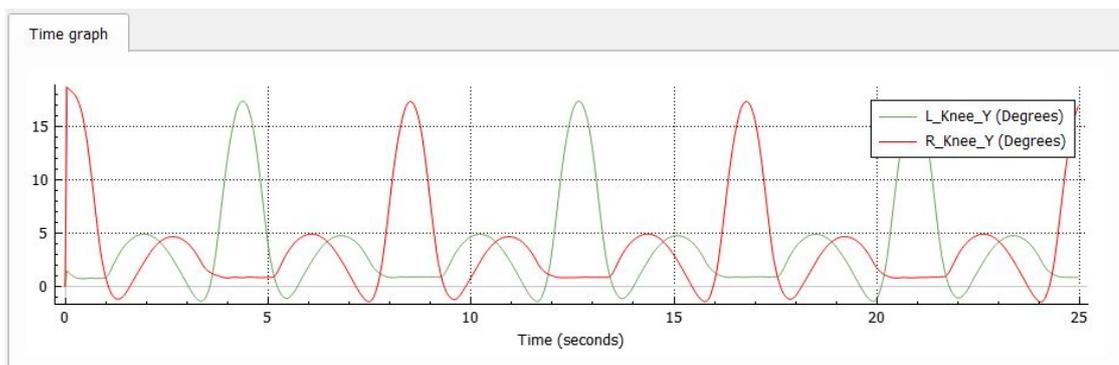


**Figure 7.11:** Left and right hip joint angles (Roll)



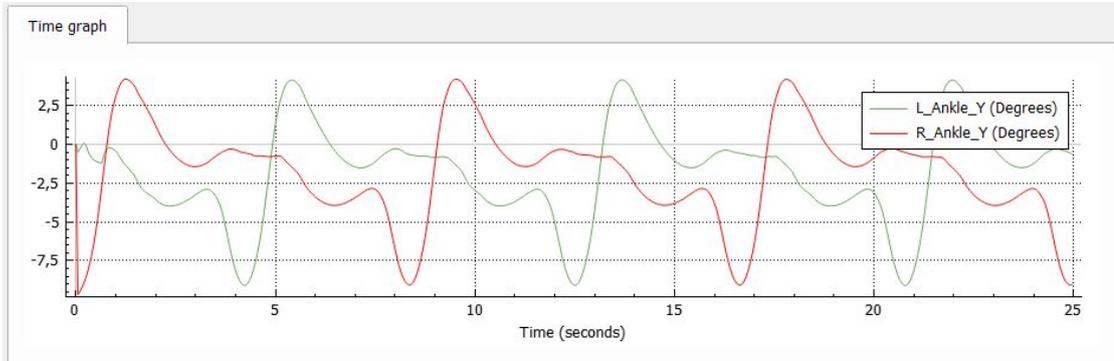
**Figure 7.12:** Left and right hip joint angles (Pitch)

In the following figure there is the trend of knee angles, also in this graph is possible to see that at the starting point of the simulation the support leg is the left one, instead the right knee creates an angle.

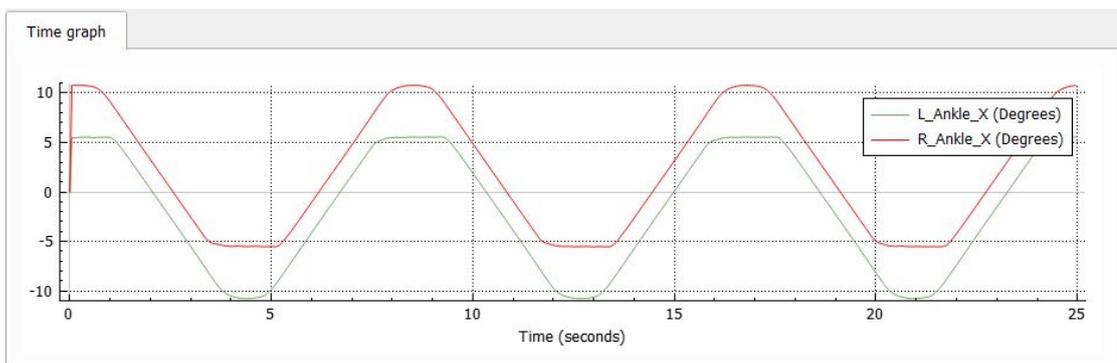


**Figure 7.13:** Left and right knee joint angles (Pitch)

This two figures are the representation of ankle angles. The same considerations for the previous graphs can be made.



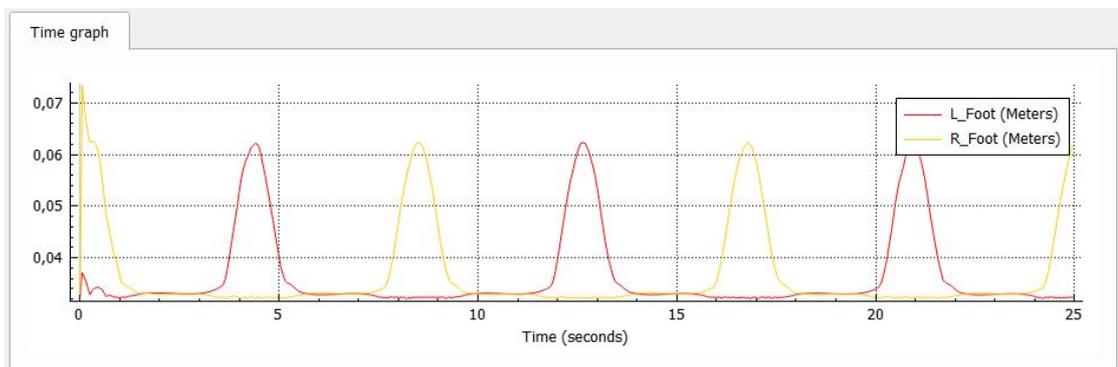
**Figure 7.14:** Left and right ankle joint angles (Pitch)



**Figure 7.15:** Left and right ankle joint angles (Roll)

This last figure confirm the consideration made before because represent all the walk phases, in particular:

- From 0 seconds to 2 seconds there is the single stance with the left foot on the ground.
- From 2 seconds to 4 seconds there is the double stance phase.
- From 4 seconds to 6 seconds there is the single stance with the right foot on the ground.
- And so on...



**Figure 7.16:** Feet Z position

### 7.1.4 Code

In order to move the biped a code attached to the *Base* must be included. The code is wrote in *Lua* (lightweight, high-level, multi-paradigm programming language designed primarily for embedded use in applications [39]) and it is wrote taking as example the *Asti* mobile robot proposed in CoppeliaSim. All the functions are explained as comments (—) in the code:

```

1 function sysCall_init()
2     sim.getObjectHandle("Base")
3
4     —sim.getObjectHandle(): it retrieves an object handle based on
5     its name
6
7     lFoot=sim.getObjectHandle("leftFootTarget")
8     rFoot=sim.getObjectHandle("rightFootTarget")
9     lPath=sim.getObjectHandle("leftFootPath")
10    rPath=sim.getObjectHandle("rightFootPath")
11    lPathLength=sim.getPathLength(lPath)
12
13    —sim.getPathLength(): it returns the lengths of a path.
14    —Each path point will have a corresponding length value
15    —(taken as the distance from the path's first point, along the
16    path)
17    —Lua synopsis -> number length=sim.getPathLength(number
18    objectHandle)
19    —Lua parameters -> objectHandle: handle of the path object
20    —Lua return values -> length: length of the path given in meters
21
22    rPathLength=sim.getPathLength(rPath)
23    dist=0
24    correction=0.0305
25
26    minVal={0,      — Step size
27            0      — Walking speed
28            }
29
30    rangeVal={2,   — Step size
31             0.8  — Walking speed
32             }
33
34    relativeStepSize=1
35    nominalVelocity=0.1
36 end
37
38 function sysCall_actuation()
39
40

```

```

36  —Get the desired position and orientation of each foot from the
    paths:
37
38  t=sim.getSimulationTimeStep()*nominalVelocity
39
40  —sim.getSimulationTimeStep(): it retrieves the simulation time
    step
41  —(the simulation time (i.e. not real-time)
42  —that passes at each main script simulation pass).
43  —This value might not be constant for a given simulation.
44  —Lua synopsis -> number timeStep=getSimulationTimeStep()
45
46  dist=dist+t
47  lPos=sim.getPositionOnPath(lPath, dist/lPathLength)
48
49  —sim.getPositionOnPath(): it retrieves the absolute
    —interpolated position of a point along a path object
50  —Lua synopsis -> table_3 pos=sim.getPositionOnPath(numb
51  pathHandle,
52  —numb relativeDistance)
53  —Lua parameters -> pathHandle: handle of the path object
54  —relativeDistance: a value between 0 and 1, where 0
55  —is the beginning of the path,
56  —and 1 the end of the path.
57  —Lua return values -> position: table of 3 values (x, y and z)
58
59  lOr=sim.getOrientationOnPath(lPath, dist/lPathLength)
60
61  —Retrieves the absolute interpolated orientation of a point
62  —along a path object.
63  —Lua synopsis -> table_3 eulAng=sim.getOrientationOnPath(
64  —numPathHandle, numRelativeDistance)
65  —Lua parameters -> pathHandle: handle of the path object
66  —relativeDistance: a value between 0 and 1, where 0 is the
67  —beginning of the path, and 1 the end of the path.
68  —Lua return values -> eulAng: table of 3
69  —values (alpha, beta and gamma)
70
71  p=sim.getPathPosition(rPath)
72
73  —Retrieves the intrinsic position of a path object
74  —(a distance along the path).
75  —The position is given in meters
76  —Lua synopsis -> numb pos=sim.getPathPosition(numb objectHandle)
77  —Lua parameters -> objectHandle: handle of the path object
78  —Lua return values -> position: linear position on the path
79  —given in meters
80
81  rPos=sim.getPositionOnPath(rPath, (dist+correction)/rPathLength)

```

```

82   rOr=sim.getOrientationOnPath(rPath,(dist+correction)/rPathLength)
83
84   —Now we have the desired absolute position and orientation
85   —for each foot.
86   —Now transform the absolute
87   —position/orientation to position/orientation relative to biped
88   —Then modulate the movement forward with the desired "step size"
89   —Then transform back into absolute position/orientation:
90
91   trunkM=sim.getObjectMatrix(trunk,-1)
92
93
94
95
96   —Retrieves the transformation matrix of an object
97   —Lua synopsis -> table_12 mat=sim.getObjectMatrix(numb
objectHandle,
98   —numb relativeToObjectHandle)
99   —Lua parameters -> objectHandle: handle of the object.
100  —relativeToObjectHandle: indicates relative to which reference
frame
101  —we want the matrix.
102  —Specify -1 to retrieve the absolute transformation matrix
103  —Lua return values -> matrix: table of 12 numbers
104  --(the last row of the 4x4 matrix (0,0,0,1) is not returned)
105
106  trunkMInverse=simGetInvertedMatrix(trunkM)
107  m=sim.multiplyMatrices(trunkMInverse,sim.buildMatrix(lPos,lOr))
108
109  —Builds a transformation matrix based on a position vector and
Euler angles
110  —Lua synopsis -> table_12 matrix=sim.buildMatrix(table_3 pos,
table_3 eulAng)
111  —Lua parameters -> position: table to 3 numbers representing
112  —the position component
113  —eulerAngles: table to 3 numbers representing the angular
component
114  —Lua return values -> matrix: table containing the
transformation matrix
115  --(except for the last row).
116  —Note: table values in Lua are indexed from 1, not 0.
117
118  —Multiplies two transformation matrices
119  —Lua synopsis -> table_12 resultMatrix=sim.multiplyMatrices(
120  —table_12 matrixIn1,
121  —table_12 matrixIn2)
122  —matrixOut: the output matrix (the result of the multiplication:
123  —matrixIn1*matrixIn2).

```

```

124 | —A transformation matrix contains 12 values (the last row
      | (0,0,0,1)
125 | —is omitted):
126 | —The x-axis of the orientation component is (matrix[1],matrix
      | [5],matrix[9])
127 | —The y-axis of the orientation component is (matrix[2],matrix
      | [6],matrix[10])
128 | —The z-axis of the orientation component is (matrix[3],matrix
      | [7],matrix[11])
129 | —The position component is (matrix[4],matrix[8],matrix[12])
130 |
131 | m[8]=m[8]*relativeStepSize
132 | m=sim.multiplyMatrices(trunkM,m)
133 | lPos={m[4],m[8],m[12]}
134 | lOr=sim.getEulerAnglesFromMatrix(m)
135 |
136 | —Retrieves the Euler angles from a transformation matrix
137 | —Lua synopsis -> table_3 eulerAngles=sim.
      | getEulerAnglesFromMatrix(
138 | —table_12 matrix)
139 | —Lua parameters -> matrix: table to 12 numbers (the last row of
      | the
140 | —4x4 matrix (0,0,0,1) is not needed).
141 | —Table values in Lua are indexed from 1, not 0!
142 | —Lua return values -> eulerAngles: table to 3 numbers
143 | —representing the Euler angles, or nil in case of an error
144 |
145 | m=sim.multiplyMatrices(trunkMInverse,sim.buildMatrix(rPos,rOr))
146 | m[8]=m[8]*relativeStepSize
147 | m=sim.multiplyMatrices(trunkM,m)
148 | rPos={m[4],m[8],m[12]}
149 | rOr=sim.getEulerAnglesFromMatrix(m)
150 |
151 | —Finally apply the desired positions/orientations to each foot
152 | —We simply apply them to two dummy objects that are then handled
153 | —by the IK module to automatically calculate all leg joint
154 | —desired values
155 | —Since the leg joints operate in hybrid mode, the IK calculation
156 | —results are then automatically applied as the desired
157 | —values during dynamics calculation
158 |
159 | sim.setObjectPosition(lFoot,-1,lPos)
160 |
161 | —Sets the position (x, y and z-coordinates) of an object.
162 | —Lua synopsis -> sim.setObjectPosition(number objectHandle,
163 | —number relativeToObjectHandle,table_3 position)
164 | —Lua parameters -> objectHandle: handle of the object.
165 | —relativeToObjectHandle: indicates relative to which reference
166 | —frame the position is specified.

```

```
167 | —Specify -1 to set the absolute position
168 | —position: coordinates of the object (x, y and z)
169 |
170 | sim.setObjectOrientation(lFoot,-1,lOr)
171 |
172 | —Sets the orientation (Euler angles) of an object
173 | —Lua synopsis -> sim.setObjectOrientation(
174 | —number objectHandle,number relativeToObjectHandle,
175 | —table_3 eulerAngles)
176 | —Lua parameters -> objectHandle: handle of the object.
177 | —relativeToObjectHandle: indicates relative to which reference
178 | —frame the orientation is specified.
179 | —Specify -1 to set the absolute orientation.
180 | —eulerAngles: Euler angles (alpha, beta and gamma)
181 |
182 | sim.setObjectPosition(rFoot,-1,rPos)
183 | sim.setObjectOrientation(rFoot,-1,rOr)
184 | end
185 |
```



# Conclusion & Future Works

A robotic exoskeleton intended to be a powerful assistant for overground gait training has been developed. The device is a "powered suit" that is worn by the patient. The objective is creating a tool that helps clinicians in the rehabilitation process of stroke and spinal cord injury patients.

From the side of the simulation part, the generation of feet, ZMP and CoM trajectories gave good results consistent with the data used. But, the results of the fitting process were not perfect: a choice had to be made about whether to give value to the balancing action or to synchronism with physiological joint patterns.

The future improvements are:

- For further verification it would be important to develop a new dataset, where measurements are accompanied by the precise dimensions of the various body segments of the subjects.
- A complex deep-learning tool, able to measure the EMG signals while the patient is moving, translates them into torque signals for the joints of the robot, compute the difference between the effective torque produced by the patient's muscles and the required torque to accomplish the postural task.
- A control system for postural equilibrium works together with the neural network, to correct input torques at each joint.
- The development of a control system for a dynamic model uses the ZMP to assess balance while walking in real-time.

# Bibliography

- [1] A. Albert and W. Gerth. «Analytic path planning algorithms for bipedal robots without trunk». In: *Journal of Intelligent and Robotic Systems*, vol.36 (2003), pp. 109–127 (cit. on p. 3).
- [2] H. Hirukawa, S. Hattori, K. Harada S. Kajita, K.Kaneko, F. Kanehiro, M. Morisawa, and S. Nakaoka. «A pattern generator of humanoid robots walking on a rough terrain». In: *IEEE International Conference on Robotics and Automation* (Apr. 2007), pp. 2781–2187 (cit. on p. 3).
- [3] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. «A realtime pattern generator for biped walking». In: *IEEE International Conference on Robotics and Automation, Washington and DC* (May 2002), pp. 31–37 (cit. on p. 3).
- [4] Jung-Yup Kim, Ill-Woo Park, and Jun-Ho Oh. «Walking Control Algorithm of Biped Humanoid Robot on Uneven and Inclined Floor, HUBO Laboratory». In: (Jan. 2007) (cit. on pp. 6, 28).
- [5] C. Hernández-Santos, E. Rodríguez-Leal, R. Soto, and J.L. Gordillo. «Robotics| Kinematics and Dynamics of a New 16 DOF Humanoid Biped Robot with Active Toe Joint». In: (2012) (cit. on p. 7).
- [6] Xinyu Guan, Linhong Ji, and Rencheng Wang. «Development of Exoskeletons and Applications on Rehabilitation». In: *Division of Intelligent and Bio-mimetic Machinery, The State Key Laboratory of Tribology, Tsinghua University, China* (2016) (cit. on p. 13).
- [7] SangJoo Kwon and Jinhee Park. «Kinesiology-Based Robot Foot Design for Human-Like Walking». In: (2012) (cit. on pp. 15, 63).
- [8] Wilfred Dempster. *Space requirements of the seated operator geometrical, kinematic, and mechanical aspects of the body*. University of Michigan: Carpenter Litho, 1955 (cit. on p. 16).
- [9] R.Drillis, R.Contini, and M.Bluestein. «Body Segment Parameters; a Survey of Measurement Techniques». In: *J. Bone Joint Surg.* 48-A (1964) (cit. on p. 16).

- [10] David A. Winter. *Biomechanics and Motor Control of Human Movement: Fourth Edition*. John Wiley and Sons Inc., 2009 (cit. on pp. 16, 18, 37, 45).
- [11] MD Perry Jacquelin. *Gait Analysis, Normal and Pathological Function*. SLACK Incorporated, 1992 (cit. on p. 22).
- [12] *ZMP on Wikipedia*. URL: [https://en.wikipedia.org/wiki/Zero\\_moment\\_point](https://en.wikipedia.org/wiki/Zero_moment_point). (accessed: 17.09.2021) (cit. on p. 24).
- [13] M.B. Popovic, A. Goswami, and H. Herr. «Ground reference points in legged locomotion: Definitions, biological trajectories and control implications». In: *International Journal of Robotics Research*, vol.24 (2005), pp. 1013–1032 (cit. on p. 25).
- [14] M. Vukobratovic, B. Borovac, and V. Potkonjak. «ZMP: A review of some basic misunderstandings». In: *International Journal of Humanoid Robotics*, vol. 3 (2006), pp. 153–175 (cit. on p. 27).
- [15] Kajita S. and Tani K. *Dynamic Biped Walking Control on Rugged Terrain Using the Linear Inverted Pendulum Mode*. *Trans. Soc. Instr. Contr. Eng.*, 1995, pp. 1705–1714 (cit. on p. 31).
- [16] Kajita S., Hirukawa H., Harada H., and Yokoi K. *Introduction to Humanoid Robotics*. Springer: Berlin/Heidelberg, Germany, 2014 (cit. on p. 31).
- [17] Motoi N., Suzuki T., and Ohnishi K. *A Bipedal Locomotion Planning Based on Virtual Linear Inverted Pendulum Mode*. *IEEE Trans. Ind. Electron.*, 2009, pp. 54–61 (cit. on p. 31).
- [18] Shibuya M., Suzuki T., and Ohnishi K. «Trajectory Planning of Biped Robot Using Linear Pendulum Mode for Double Support Phase». In: *In Proceedings of the 32nd Annual Conference of IEEE Industrial Electronics, Paris, France*. 6–10 November 2006, pp. 4094–4099 (cit. on p. 32).
- [19] Shibuya M., Sato T., and Ohnishi K. «Trajectory generation of biped robots using Linear Pendulum Mode with virtual supporting point». In: *In Proceedings of the 10th IEEE International Workshop on Advanced Motion Control, Trento, Italy*. 26–28 March 2008, pp. 284–289 (cit. on p. 32).
- [20] Shuuji Kajita et al. «Biped walking pattern generation by using preview control of zero moment point». In: *In Proceedings-IEEE International Conference on Robotics and Automation 2*. 2003, pp. 1620–1626 (cit. on p. 34).
- [21] Menegazzi Luca and Floris Federico. «Kinematic analysis of the gait for the man-machine interface of a future lower-limb exoskeleton». In: Master’s degree thesis (2021) (cit. on pp. 37, 60).

- 
- [22] R Macaluso, K Embry, D Villarreal, and R Gregg. *Human Leg Kinematics, Kinetics, and EMG during Phase-Shifting Perturbations at Varying Inclines*. URL: <https://dx.doi.org/10.21227/12hp-e249>. (accessed: 20.09.2021) (cit. on p. 38).
- [23] Rachele Rossanigo. *Analysis of spatial parameters during gait with magneto-inertial sensors and infrared proximity sensors*. Politecnico di Torino, 2019 (cit. on pp. 39–42).
- [24] Abu Ilius Faisal et al. *Monitoring Methods of Human Body Joints*. Sensors (Switzerland), 2019 (cit. on pp. 43, 44).
- [25] *SketchUp on Wikipedia*. URL: <https://en.wikipedia.org/wiki/SketchUp>. (accessed: 17.09.2021) (cit. on p. 47).
- [26] *Onshape on Wikipedia*. URL: <https://en.wikipedia.org/wiki/Onshape>. (accessed: 17.09.2021) (cit. on p. 51).
- [27] *Vicon Motion Systems Ltd. Plug-in Gait Reference Guide. 2017*. URL: <http://www.%20crcnetbase.com/doi/10.1201/b10400-14>. (accessed: 1.10.2021) (cit. on p. 62).
- [28] Richard Baker et al. *Handbook of Human Motion*. Handbook of Human Motion. Ed. by Springer, 2018, Chap. The Conven. (Cit. on p. 62).
- [29] Youngjin Choi et al. «Posture/walking control for humanoid robot based on kinematic resolution of CoM Jacobian with embedded motion». In: *IEEE Transactions on Robotics* (23.6 2007), pp. 1285–1293 (cit. on pp. 63, 65).
- [30] S. Kajita et al. «Biped walking pattern generation by using preview control of zero-moment point». In: *Proceedings - IEEE International Conference on Robotics and Automation 2* (2003), pp. 1620–1626 (cit. on p. 67).
- [31] Russ Tedrake et al. «A closed-form solution for real-time ZMP gait generation and feedback stabilization». In: *IEEE-RAS International Conference on Humanoid Robots* (2015-December), pp. 936–940 (cit. on p. 67).
- [32] G. Menga and M. Ghirardi. «Modelling, Simulation and Control of Walk of Biped Robotic Devices». In: *Part I : Modelling and Simulation using Autolev* (2015) (cit. on p. 76).
- [33] *CoppeliaSim on Wikipedia*. URL: <https://en.wikipedia.org/wiki/CoppeliaSim>. (accessed: 17.09.2021) (cit. on p. 82).
- [34] *Site of the software for the simulation*. URL: <https://www.coppeliarobotics.com/>. (accessed: 05.09.2021) (cit. on p. 82).
- [35] *Path of the site for the simulation*. URL: <https://www.coppeliarobotics.com/helpFiles/en/paths.htm>. (accessed: 15.09.2021) (cit. on p. 83).

## BIBLIOGRAPHY

---

- [36] URL: <https://www.coppeliarobotics.com/helpFiles/en/dummies.htm>. (accessed: 15.09.2021) (cit. on p. 83).
- [37] URL: <https://www.coppeliarobotics.com/helpFiles/en/dummyPropertiesDialog.htm>. (accessed: 15.09.2021) (cit. on p. 83).
- [38] MathWorks. «Walking Robots – Linear Inverted Pendulum Model (LIPM)». In: (2019). (accessed: 15.09.2021) (cit. on pp. 86, 87).
- [39] *LUA on Wikipedia*. URL: <https://en.wikipedia.org/wiki/Wikipedia:Lua>. (accessed: 17.09.2021) (cit. on p. 92).