

Politecnico di Torino



**Department of CONTROL AND COMPUTER
ENGINEERING (DAUIN)**

Master's Degree course in Mechatronic Engineering

Master's Degree Thesis

**Modelling of a biped kinematics and analysis of the human
walk for a future lower-limbs exoskeleton design**

Candidate:

Lorenzo Russo

Supervisors:

Eng. Maurizio Morisio

Eng. Giuseppe Menga

Academic Year 2020/2021

Abstract

The objective of this thesis is to show the differences between a walking pattern of a humanoid robot and a future exoskeleton.

First of all, a deep analysis of autonomous bipeds has been conducted, underlining many differences between them and a proper human walking. The two main differences are: the hypothesis of flat foot during gait-cycle with respect to the floor; and the modelling of the foot as a whole body (not divided in foot and toes).

Starting from the second part of the thesis, there was the need of developing a humanoid model. In order to find an approximation of the human body, a biped model was created. It is a 14-dof model, with links' measures taken from standard human data and revolute joints representing each articulation. This process begins with the creation of a CAD model, in Solidworks, through which it was possible to generate another Simscape model to carry out simulations and results analysis.

The first experiment consists in recreating the inverse kinematics of a biped starting from dr. Ali's algorithm. Through this approach, each joint variable was calculated starting from the feet trajectories in the Cartesian space, the LIPM (Linear Inverted Pendulum Model) and the assumption of the flat foot.

On the other hand, in the second experiment, a two-bodies foot was modelled, with an appropriately tuned intermediate passive revolute joint. The joints' variables were taken from a complete dataset containing angle measurements of each articulation of a patient with no evident pathological condition walking on a treadmill. Thanks to the Simscape platform, it was possible to calculate the direct kinematics and analyse Cartesian variables such as ZMP and CoG with baso-signals reconstruction.

In conclusion, this thesis puts in confront the behaviour the system has when the joints' movement is imposed by inverse kinematic calculation with the one it has when movements are imposed by measurements on the patient. In future it will be possible to create an exoskeleton adding an online analysis of the EMG signals through deep-learning, in which the patient could close the control loop through his/her movements.

Index

Introduction.....	6
1 Biomechanics of walking.....	8
1.1 Walking: definitions and parameters.....	8
1.2 Gait cycle	12
1.3 Walking kinematics	18
1.3.1 Centre of mass kinematics	20
1.4 Angular articulations' measurements.....	21
1.5 Dynamical parameters	25
1.5.1 Zero-Moment Point (ZMP).....	25
1.5.2 Basometric signals.....	29
2 Biped modelling.....	30
2.1 CAD modelling.....	30
2.2 XLM file generation and analysis.....	36
2.3 Simulink's Simscape Multibody Model.....	38
2.4 Ground contact modelling	43
2.4.1 Frictional force	43
2.4.2 Simulink blocks for contact modelling with parameters tuning	44
2.5 Passive toe joint implementation	48
2.6 ZMP and Basometric signals extraction	50
2.6.1 ZMP algorithm and blocks.....	50
2.6.2 Basometric signals extraction	52
3 Simulations.....	53
3.1 Inverse kinematics approach	54
3.1.1 The Linear Inverted Pendulum Model	54
3.1.2 Closed-Form Inverse Kinematic Joint solution for Humanoid Robots	58
3.1.3 Simulation code setup.....	64
3.1.4 Simscape Multibody simulation	66
3.2 Simulations with human measurement.....	69
3.2.1 Measurement setup	70
3.2.2 Simulation with complete dataset	72
3.2.3 Simulation with complete dataset and joint constraints.....	77
4 Conclusion and future development.....	79
5 List of figures	97

6	Bibliography	100
---	--------------------	-----

Introduction

Since the popping off of autonomous technology, dreamers, researchers, comics creators have always been tempted by the idea of a giga-humanoid robot fighting for the defense of our planet. Old examples may be autonomous robots like the “Transformers” –ancient sentient machines coming from an extinct planet- or spatial fighters made of steel commanded by pilots such as “Gundam” and Pacific Rim’s “Jaegers”, and many others. In few words, these machines have always had a kind of attraction on each dreamer, especially from tender age. Here, in this thesis, we will follow my personal childish dream of creating a robot. Unfortunately, this requires many knowledges not coming from dreams, but from studies, experiments, failures and hours and hours of hard work between all the people who find their field on humanoid robots development.

The idea, following the comics association, is to give a smaller representation of both autonomous walkers (like the Autobots) and human-commanded walkers (like the Jaegers) –that can be called exoskeletons. In order to move on in the studies, it is necessary to firstly analyse human anatomy and how our body works while executing a simple task like walking –which is the first complex task we learn as children. At the moment, there is no humanoid robot that can show a human-like walk. This is due to the fact that, when walking, we put in act many physical principles such as the conversion from kinetic to potential energy and vice-versa, the large number of degrees-of-freedom our body has that are very difficult to mechanically implement, the proper and singular walking pattern we all have. Each bone, muscle, tendon, in each task we impose to our body, plays a specific role that can be difficult to even estimate. For this reason there is the need to simplify both the human body and the tasks. Many studies were carried out aimed to the modelling of the body and the walk. The ones we will consider in this thesis are essentially the best-known.

We said before that there are Autobots and there are Jaegers, so what is the difference between them? The actuation. They are built in the exact same way but controlled by two different entities: calculations or human will. The first one comes from algebraic equations, following laws imposed by the previous cited models, while the second represents a direct interaction between a senseless steel machine and a strong-willed human. In order to express this concept, we have created a humanoid robot model, to be controlled, in a first approach, by calculations, and in a second implementation, by signals recorded on a human patient. Given the pandemic period of loneliness we are living now, the robot model we have created and worked on during these past month, was called "Wilson" for both the cinematographic association with the blooded ball Tom Hanks had as friend in the famous film "Cast Away", and the name composition : "will" and "son".

Wilson was created starting from standard human parameters, taken from literature, by modelling a simple CAD assembly on the Solidworks platform that will be shown later on. Thanks to some software's tool, Wilson's assembly was translated in a Simulink's Simscape model, that will be the simulation mean to analyse results and to have a visual interface of the movement of the robot.

On this model, two different experiments will be done. The first one is essentially an inverse kinematics approach carved out for humanoid robots. Here through kinematics calculations, Wilson will receive a complete set of joint variables that will animate a robotic flat-foot walk with short steps. Hence, he will move more like a ninety years-old grandpa than like the powerful Optimus Prime, chief of the Autobots. Even though Wilson will not be so athletic, he will show interesting results of balance and evidence the main characteristics of the human walk, like the position of the center of mass with respect to the feet, the inclination of the trunk, the swing of the foot and the positioning of the whole body during the swing. In the second experiment, on the other hand, Wilson will receive the complete joint variables set from a database containing measurements on many different patients walking on a treadmill. Here, we will observe some other parameters, like the ZMP, and CoM position with respect to the ground. Thanks to this measurements, we will be able to give some constraint to fix the little flaws the patient showed when data were recorded, in order to understand the procedure to build a rehabilitation exoskeleton, that again, will be nothing like the Jaeger, but will be a good starting point for future development.

1 Biomechanics of walking

Since tender age, each child seeks for his own frontal progression, starting from a crawl, moving to a wobbling walk and arriving, at the end, to the final adult walking. In this process, we firstly learn –and develop moving on- a complex set of manoeuvres requiring strength, coordination and, after a while, instinct. Each day we make so many steps that, without a smart-watch or some technological support, would be impossible to count. This repetition makes our body develop the muscle memory needed to make each step very similar to the previous, and thanks to this special memory, we learn our proper walk, that is so peculiar and singular that could be used instead of fingerprints to recognise each individual^[6]. Obviously, the muscle memory needed to put in act any task, depends on our genetic heritage –in fact it is not an event that children walk like their parents-, proper body measures, muscle symmetry.

1.1 Walking: definitions and parameters

First of all, given the huge number of different progression profiles, it is necessary to normalize walking data in order to define the universal characteristics, shared among all human beings. The most common is the normalization performed with respect to the mass of the subject, usually followed by a division in three different cadence groups: slow, natural, fast^{[1][2]}. This normalization analyses the efficiency of a progression strategy, with equal distance covered, putting in account the product of the carried mass with the distance, since each muscular contraction, during walking is aimed to the frontal progression. Together with this parameter, the differences of potential energy between the starting point and the end of the path are taken into account^{[3][4]}.

In our body, the neural system, provides the correct information to each muscle aimed to the continuative conversion of the energy from the strike of the foot to the ground, to keep equilibrium and a proper trajectory of the whole body when walking^{[4][5]}. But before going in-depth of these factors, it is necessary to introduce definitions and conventions. The definitions of the reference axis, choices of angle orientations and nomenclature to define each part of the step- cycle are exposed as it follows and shown in fig 1.1-1.2.

- Progression plane: plane on which the centre of mass moves during the gait cycle.
- Frontal plane: vertical plane dividing the body in anterior and posterior part.
- Trasversal plane: horizontal plane dividing the body in upper and lower part.
- Sagittal plane: vertical plane dividing the body in left and right part.
- Gait cycle: interval between two consecutive contacts with the ground of the same foot. It is divided in two phases (stance and swing –see fig 1.2).
- Stride: complete walking cycle, measured at the beginning and the end of the gait cycle.
- Step: interval between the contact with the ground of one foot with respect to the other.
- Step width: half of the lateral distance between feet, or the distance between the heel and the average progression line. It spaces between 5 and 7 cm.

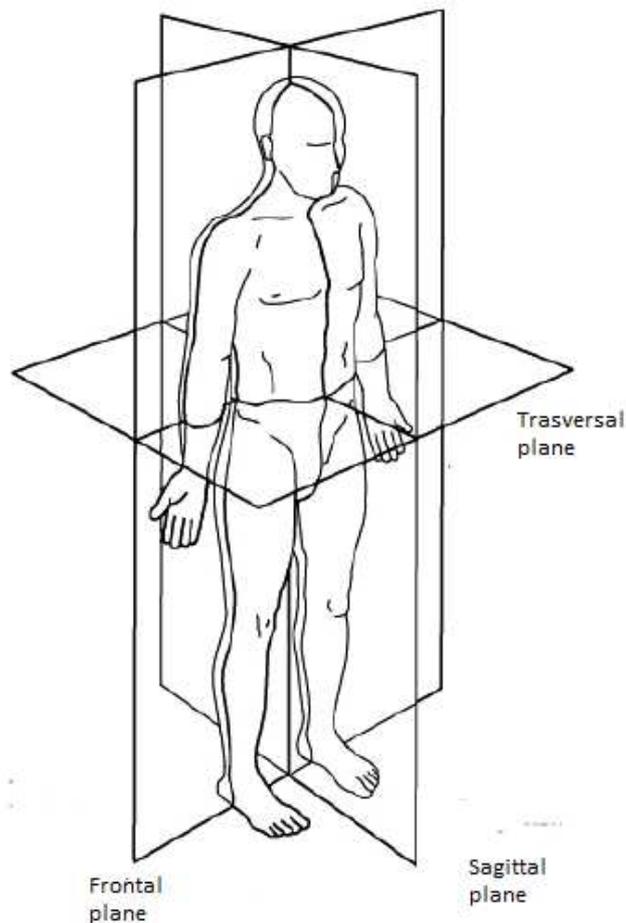


Fig 1.1 Frontal, trasversal and sagittal planes

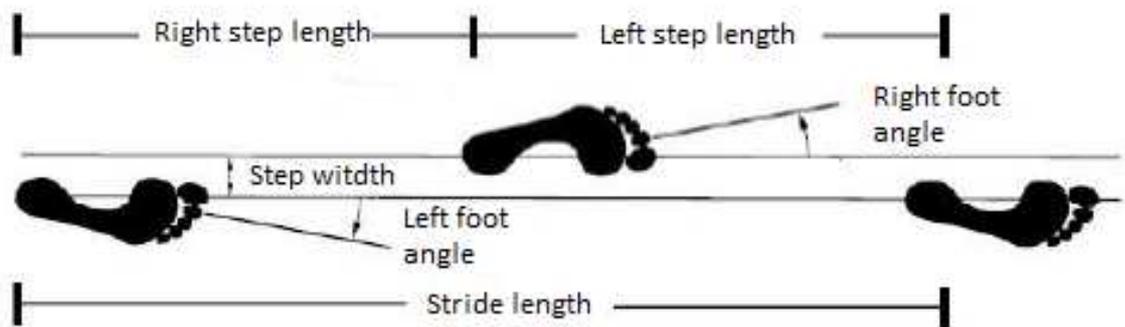


Fig 1.2 Step length and width and stride

The gait cycle is then divided in a periodic succession of motory tasks, which are presented in the following list and shown in fig 1.3.

Talking about the stance phase, we have:

- Heel-ground contact: moment in which the heel touches the ground. It determines the beginning of the gait cycle. The centre of gravity is at its lowest.
- Foot flat on the ground: interval through which the foot is flat in contact with the ground regardless of its dynamics.
- Intermediate stance: the foot on the ground is passed by the swinging one. The centre of mass is at its highest.
- Heel parting: the heel of the stance foot is raised from the ground, while the toes are still flat on the ground. In this moment the propulsion phase finds its beginning.
- Toes parting: after the heel, also the toes are raise from the ground. In this moment the stance finishes to leave place to the swing.

On the other hand, for the swing phase, we can define three different phases:

- Acceleration phase: initial part of the swing in which the foot accelerates because of the propulsion exercised on the ground and the plantar flexors of the ankle activate.
- Intermediate swing: the leg moves from a posterior to an anterior one, passing the stance foot.
- Deceleration: muscles slow down the leg, stabilizing the foot to prepare for the incoming heel strike.

We will have a deeper look on the gait cycle in the next paragraph.

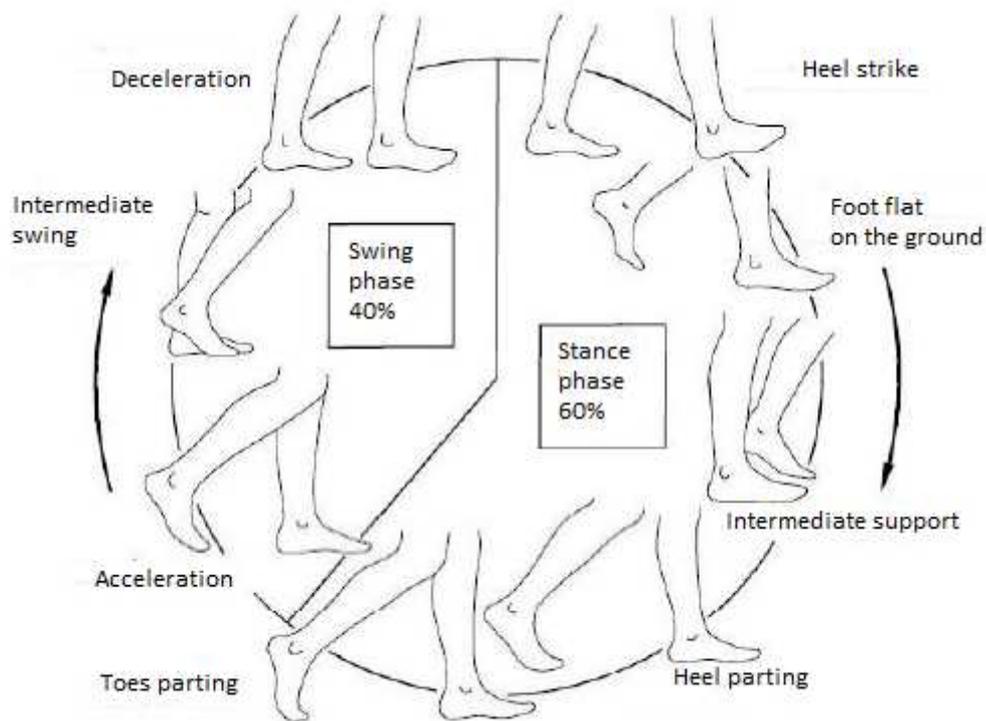


Fig 1.3 Gait cycle phases

1.2 Gait cycle

The gait cycle includes all the manoeuvres required for the forward progression of the body and, as said before, it characterizes each individual for all the small differences proper of each body. It is divided in two main phases: stance and swing. Each phase, necessities of a synchronous activation of different muscular groups, in order to keep the body balanced during walking. We can also see that, putting our attention on only one leg, there are eight main events that are shown in the following figure.

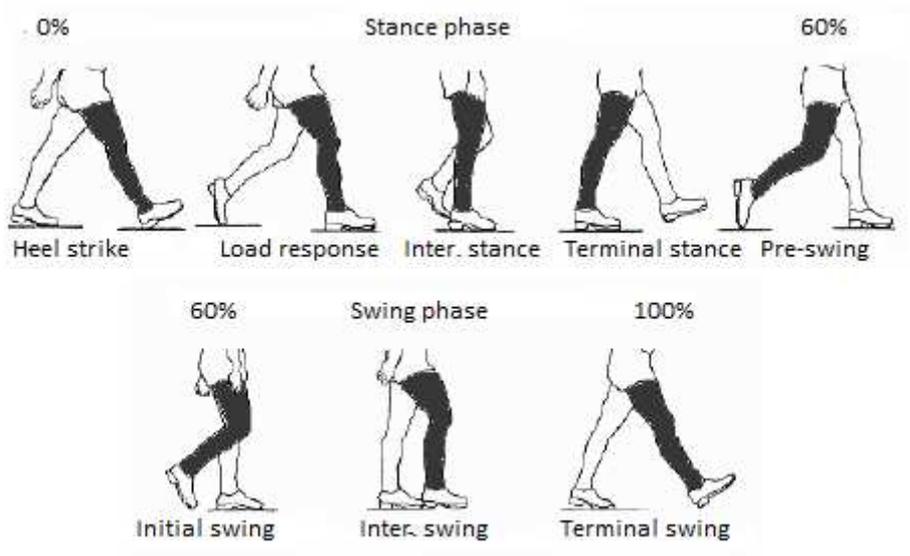


Fig 1.4 stance and swing phases

The first event of the gait cycle, as said before, is the heel strike with the ground. Here, the limb rolls around the heel-ground contact point, while the other leg is at the end of its stance. In this moment the hip is flexed and the knee fully extended. In order to prepare the foot for the supination on the ground, the tibiotarsal articulation is dorsally flexed (90°- neutral position). Here, we find an intensity peak of the three hamstrings that activate the swing to absorb the strike with the ground, slowing down firstly the thigh and lastly the leg. The quadriceps tighten to avoid the knee to flex, implying its full extension,

while the two hip extensors, curb the femur. Lastly, tibial muscles activate to position the foot for the imminent heel roll.

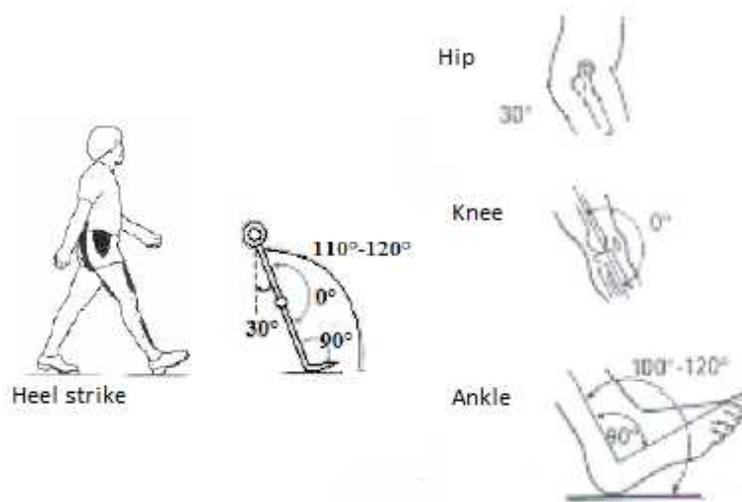


Fig 1.5 Initial contact on the ground

The second event is the response to the load. It begins with the contact with the ground and is perpetrated until the other foot is lifted for its swing. The whole body weight is transferred to the anterior leg, the knee is flexed because of the rolling around the heel to absorb the shock due to the strike on the ground. The tibiotarsal articulation flexes in order to limit the heel roll, through the flattening of the foot on the ground. The opposite limb is now preparing for the swing phase. In this moment, we find an intensity peak in the tibial muscles, due to the load acceptance, aimed to the curbing of the passive plantar flexion of the tibiotarsal. Another peak is found on the quads, to keep the knee stable during its flexion due to the heel rolling, accepting the load.

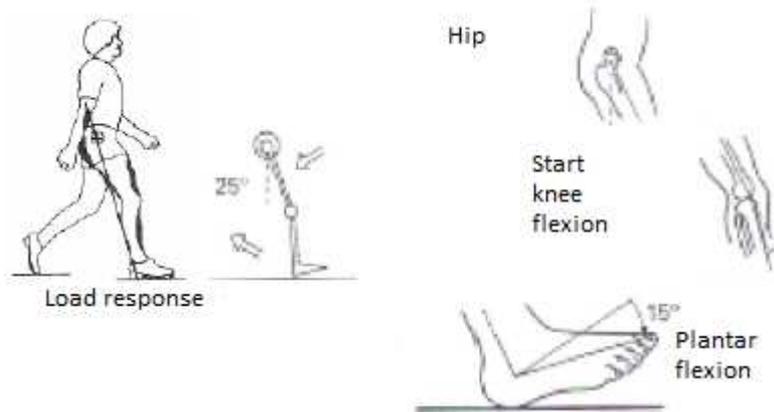


Fig 1.6 Load response

The third phase is the intermediate stance in which the body lays its weight only on one leg, while the other swings. It starts with the parting from the ground of the other foot and finishes when the swinging foot is aligned with the loaded leg. Through the dorsal flexion^[5] the leg passes the related foot, flat on the floor. Knee and hip extend. The swinging leg goes through the intermediate swing phase. Here there is a short quads activation, to give sustain to the knee, extending it. The tibiotarsal flexors ensure the smooth progression on the standing foot.

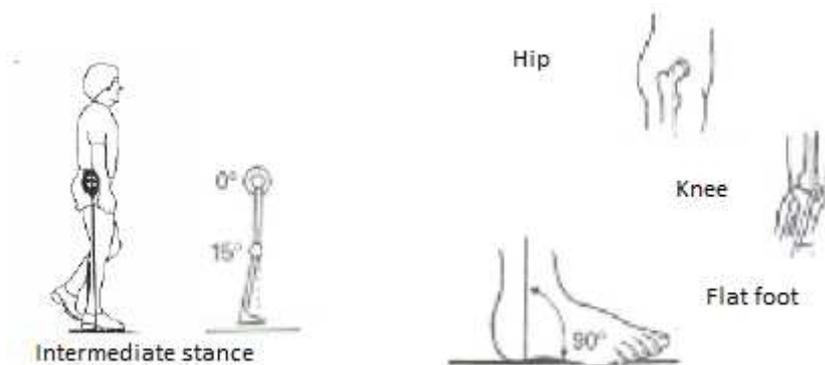


Fig 1.7 Intermediate stance

Fourth phase is the terminal stance. The first event is the parting of the heel from the ground, ending with the other foot touches the ground. The bodyweight is transferred in front of the standing foot, the heel is lifted from the ground and the standing leg goes forward for foot rolling effect. The knee keeps its extension into a small flexion and the tension on the hip pushes the leg forward. The other leg is in the terminal swinging phase.

Here, the soleus^[6] has an activity peak for a double aim: create a dorsiflexion moment opposite to the body vector, directed forward, to maintain the stability of the manoeuvre; and create a strong plantarflexion moment aimed to sustain the bodyweight while the heel is parting from the ground. Soleus contraction steadily interrupts when the foot touches the ground, leaving place to the gastrocnemius^[7] that, with its contraction, flexes the knee.

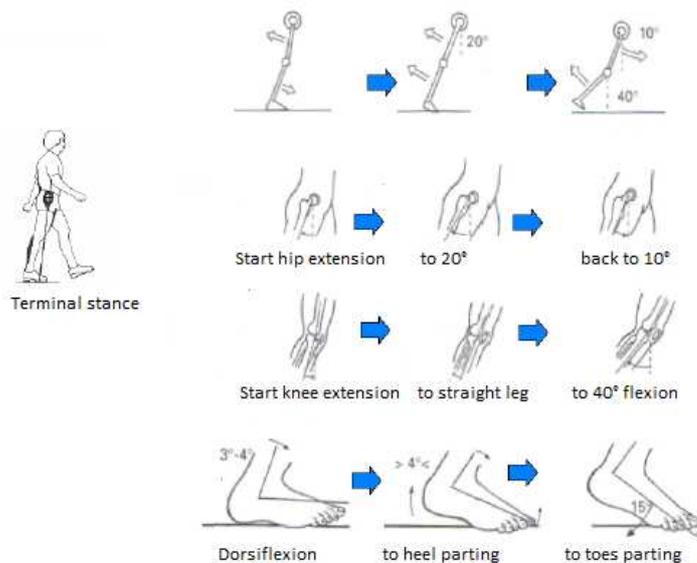


Fig 1.8 Terminal stance

Fifth phase is the pre-swing and it starts with the initial contact of the previously swinging foot, ending with the parting of the toes from the ground of the standing foot. The leg holding the bodyweight answers to the movement of the swinging foot by increasing the tibiotarsal's articulation flexion, lowering the hip extension, while the other leg is accepting the load imposed by the bodyweight. The adductor long muscle activates, resisting to the hip abduction, built by the switching of the bodyweight position between the two feet. In the final part of the manoeuvre, femoral rectus activates to decelerate the excessive knee flexion. With the double stance phase, force vector goes beyond the toes, making the foot lift from the ground. Potential energy is converted in kinetic and helps the tibia movement, flexing the knee and rotating the thigh. Residual triceps' tension increases knee flexion, countered by the dorsiflexion put in act by the tibial muscles, lifting the foot from its previous plantar flexion.

The initial swing, sixth phase, starts when the foot is lifted from the ground and finishes when the swinging leg is parallel to the standing leg. The foot is swinging and the leg is pulled forward by the conjunct flexing action of both hip and knee. Tibiotarsal articulation is partially dorsiflexed. The other leg is in the initial part of the intermediate stance phase. Here, three muscles activate- gracilis^[8], sartorius^[9] and iliacus^[9], causing the anterior rotation of the thigh. Together with the three muscles called above, femoral bicep activates to strengthen the flexing action on the knee.

Intermediate swing, second in the swing phase and seventh in the whole gait cycle, finds its place when the swinging foot passes by the standing straight leg until the tibia is totally vertical. Forward progression of the limb is due to the hip flexion. Knee extends responding to gravity force, while tibiotarsal articulation keeps flexing until it reaches its neutral position (90°). The other leg finds itself in the terminal intermediate stance phase. Here, it is found a significant reduction in the activity of the tibiotarsal's dorsal flexors. Only active muscle in the hip complex is the gracilis. In this phase all the other muscles are inactive because the strong propulsion generated in the initial swing gives the necessary inertia to the leg required for the making of the complete manoeuvre. Later on in the thesis we will analyse the difficulties a robot finds in making this movement, because it lacks muscles and is actuated by rotational motors at each joint, that do not provide any propulsion to the system. Hence, each limb is "pulled" by the joint, not "pushed".

Lastly, we have the final swing phase. It begins with the vertical tibia and ends with the significant heel strike with the ground. This event is said "significant" because it is the moment in which potential energy is converted in kinetic through the hit on the ground (other peculiarity that a robot, even the most complex one, cannot put in act). Going back to the focus topic, we find that the limb has its complete forward progression exactly in the moment in which the leg passes by its thigh, and it is accomplished by the knee extension. Hip keeps its initial flexion and tibiotarsal articulation stays dorsiflexed until it reaches the neutral position. Activated muscles are tibials, hamstrings and quads.

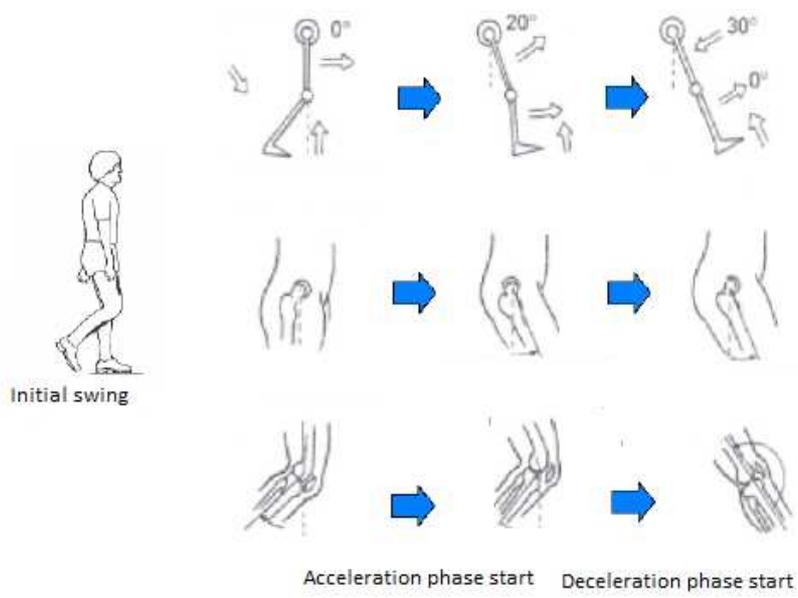


Fig 1.9 initial swing to following stance manoeuvre

1.3 Walking kinematics

In order to define the kinematics of the walking, it is necessary to analyse trajectories, movements, distances without taking into account any force, being them internal or external, causing the movement of the limbs. Since walking is fundamentally a periodic repetition of almost equal gait cycles, it will be sufficient to analyse a single double step in order to further reconstruct the whole walking. Human gait cycle, for its huge number of variables, is a particular complex manoeuvre to analyse in its totality. We will hence introduce a series of approximation to make its description and analysis more comprehensible. Results will be shown in form of temporal graphs, including vertical, horizontal positions, velocities and accelerations during stance phase. The body we will consider is a 7-links model. In this way, we only have to analyse 12-dof (3 for the hip, 1 for the knee and 2 for the ankle).

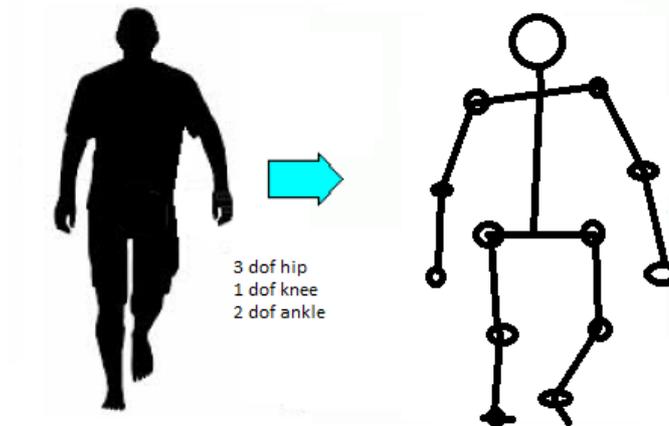


Fig 1.10 7-links human model

For example, we can observe vertical movements and vertical and horizontal velocities of two points in the foot: the heel and the meta. We see that the movements starts before the toes part from the floor, and the vertical velocity curve reaches its maximum peak in correspondence of this event. On the other hand, horizontal velocity of the heel, grows after the toes parting until it reaches its maximum during the swing. In addition, vertical trajectory decreases steadily that is stopped right before the heel strike with the ground, being immediately slowed in order to not create any shock in the hit with the floor. This

manoeuvre is justified also looking at the velocity trend, that is much lowered in correspondence of the heel strike. In the same way we can measure the same variables marking the meta to identify the different phases of the gait cycle.

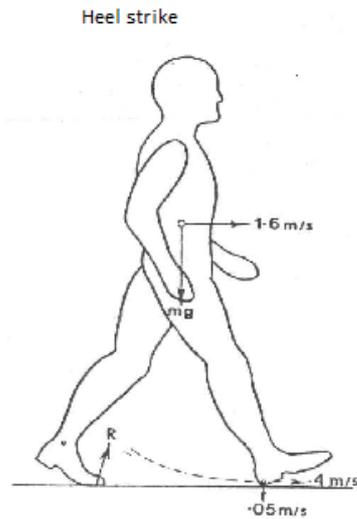
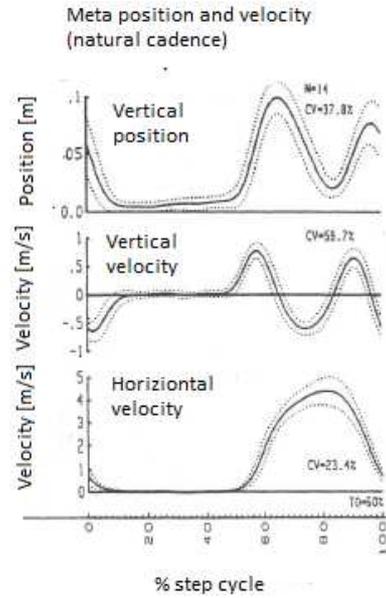
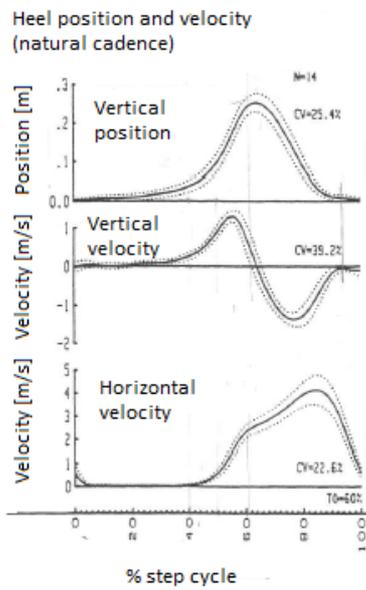


Fig 1.11 a) Heel position and velocity b) Meta position and velocity c) Representation of the body displacement at heel stike

1.3.1 Centre of mass kinematics

Once evaluated the total mass of the body, considering that the 2/3 of the total mass stays in the upper part, through a weighted sum on all the elements, it is possible to find the centre of mass of the body (that lays around the 60% of the total height). When the centre of mass is found, through measurements, we can evaluate its trajectories and its velocities in different directions, as shown in figure 1.11. The curves show a periodical trend, with a peak corresponding to the moment of single stance, while the minimum is found when the body finds itself in the middle of the double stance phase. These curves are in phase opposition with the velocity's ones. The reason of this opposition is that at its position's peak, the body has more potential energy than kinetic, while at its lowest it has more kinetic energy than potential. This conversion between energies, led researchers to the developing of the inverted pendulum model, that will be exploited later on in this thesis.

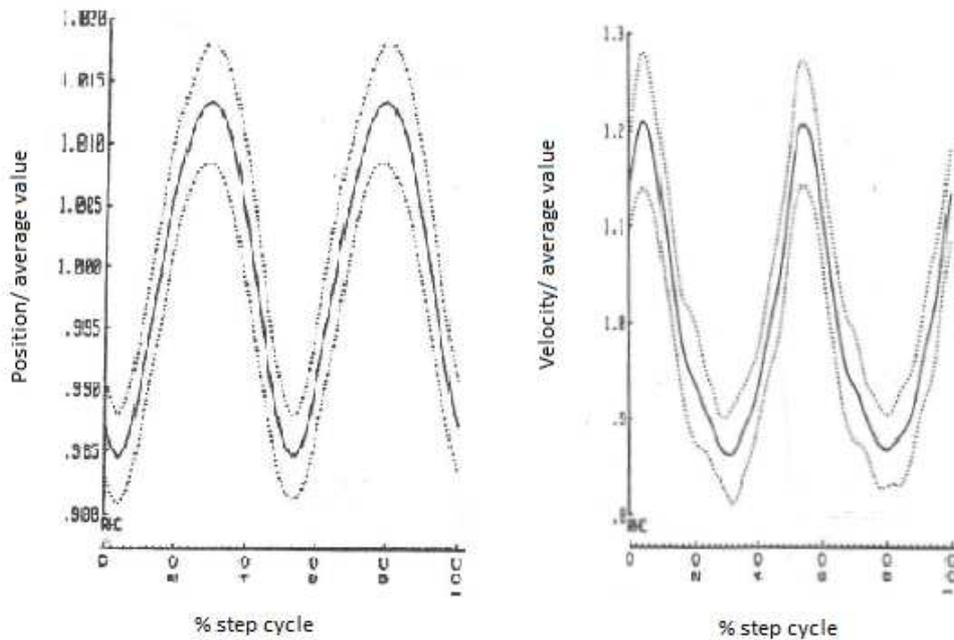


Fig 1.12 Position and velocity of the COM during the gait cycle

1.4 Angular articulations' measurements

All the angles are measured counter-clockwisely starting from the horizontal parallel to the ground. Addind the approximation of vertical trunk (the trunk always stays vertical during the whole cycle), it is possible to use the hip angle to evaluate the angular position of the thigh. In this paragraph, though, we will show a set of measures on different articulation, to find a reference in future chapters in which a complete dataset will be used.

Let us use a top-to-bottom approach. Hence, the first joint we will consider is the hip. Hip's maximum flexion is reached in the middle of the swing phase. The action of the hip's muscles is aimed to the mantaining of the correct forward trajectoy of the whole leg. Muscular activity, though, is found only in the initial part of the manoeuvre, because, as said before, walking is mainly carried out by gastrocnemius' propulsion. On the other hand, during the stance phase the principal movement is carried out by the conjunct action of both gluteus and knee flexors, before gravity takes their place, in the "free fall" phase found in the second part of the single stance phase.

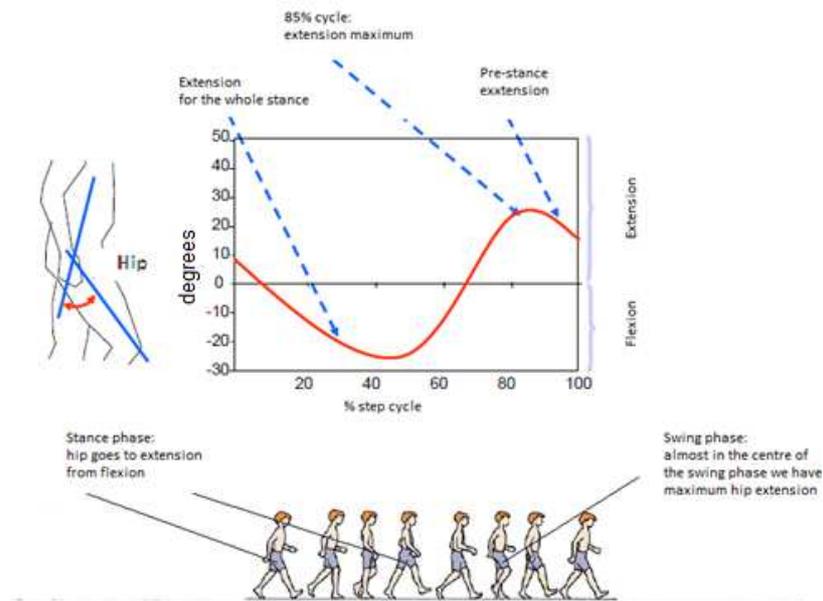


Fig 1.13 angular measures of the hip during gait cycle

Second joint to be considered is the knee. During the swing phase, the knee firstly is flexed, to be afterwards extended to prepare for the heel strike. During swing, as said many times now, muscular activity is almost null, because knee's movements are a direct consequence of hip's movement, At the end of the swing, knee's flexors stop the articulation's extension and quads contraction anticipates the heel strike. On the other hand, during stance phase, the knee flexes and extends until the heel is lifted from the ground. After this event, the knee is blocked. So, summing up, knee's flexors activate only in the initial and final part of the stance phase.

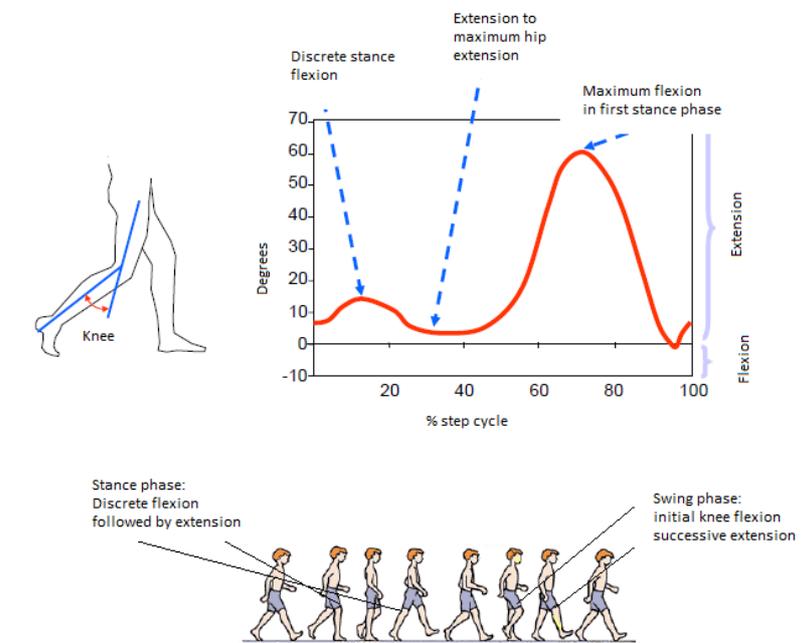


Fig 1.14 angular measures of the knee during gait cycle

Third articulation is the ankle. During swing phase, we have an initial dorsiflexion to prepare the limb for the heel strike with the ground. After that, during stance phase, ankle is in neutral position, made horizontal by the plantar flexion. To oppose the pushed forward body weight, the ankle activates a smooth dorsiflexion. At the end of the stance phase, a valuable plantar flexion is carried out in order to provide the necessary condition for a proper forward propulsion.

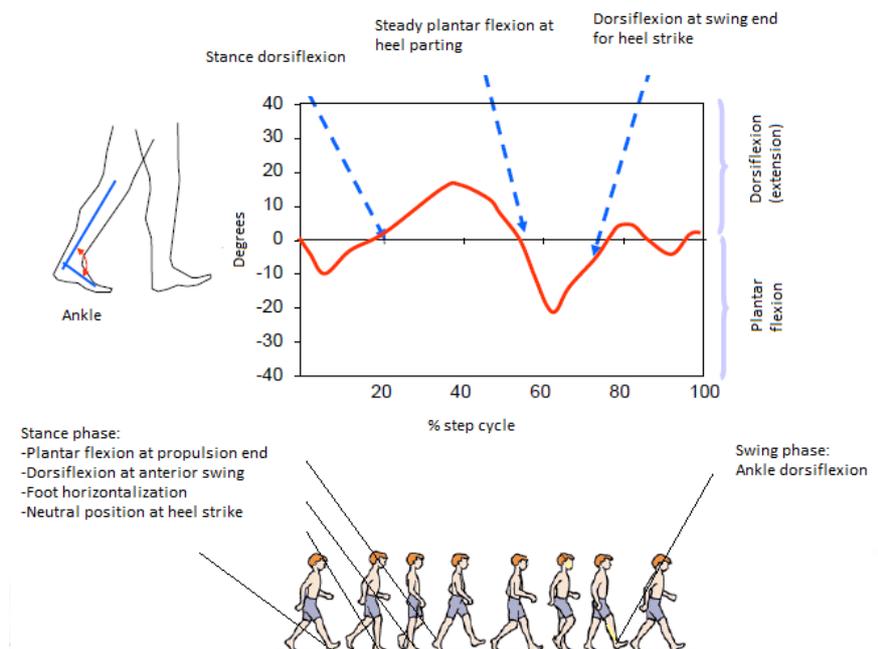


Fig 1.15 angular measures of the ankle during gait cycle

In order to normalize data, it is necessary to get measures on a large number of patients, walking with different cadences, taking into account their natural ones.

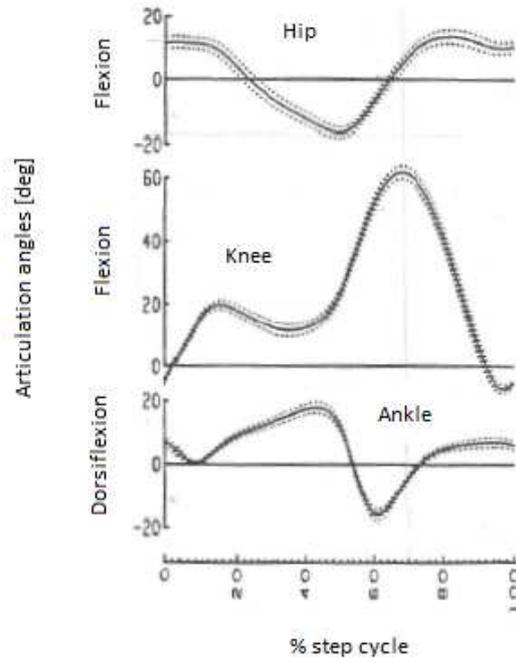


Fig 1.16 angular measures at natural cadence

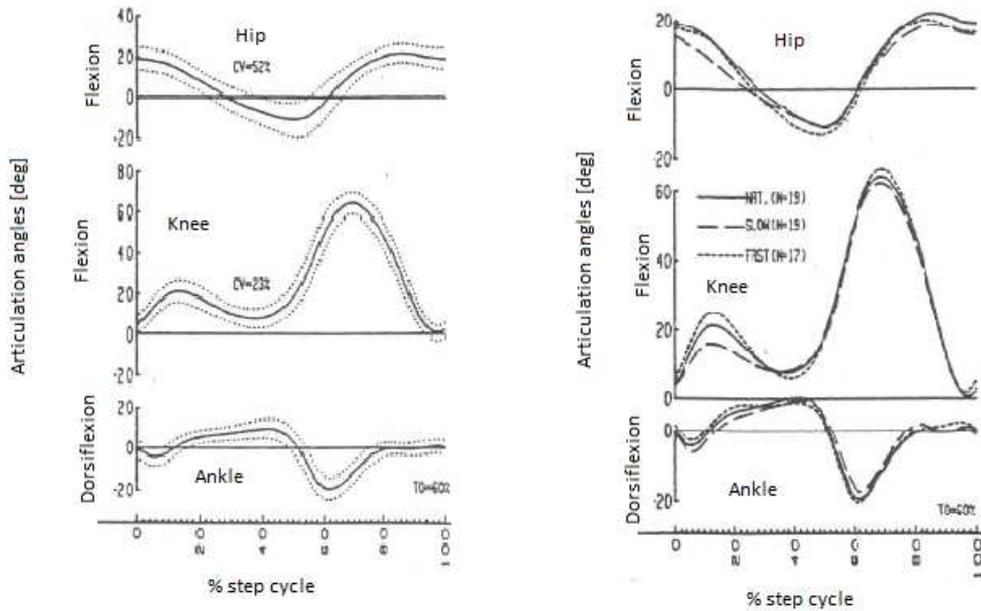


Fig 1.17 angular measures at natural cadence on a set of patients (left); angular measures at different cadences

1.5 Dynamical parameters

Even though the thesis is about a kinematic model for a walker to be in future implemented for an exoskeleton, there are two main parameters that are fundamental to be analysed to define whether a walking is pathological or standard. These parameters are the Zero-Moment Point (ZMP) and the basometric signals. These parameters' dynamics is a direct effect of the reaction contact forces with the ground in a certain point.

1.5.1 Zero-Moment Point (ZMP)

Before going to the definition of the ZMP, it is necessary to introduce two other entities: support polygon and Centre of Pressure (CoP)^{[7][8][9]}.

The support polygon is “a convex 2-D or 3-D polygon which is determined by the points of the body that are in contact with the ground”. CoP and ZMP have necessarily to lay in the support polygon's area to guarantee the stability equilibrium.

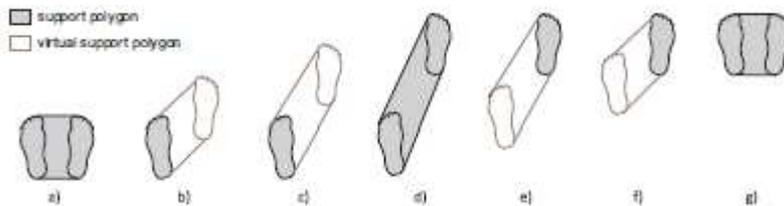


Fig 1.18 Support polygons during one gait cycle

CoP is defined by Vukubratovich et al as :“CoP represents the point on the support foot polygon at which the resultant of distributed foot ground reaction forces acts.” This point has to lay, as said before, within the support polygon and is found easily by applying Newton's third law. CoP is a single point in which we consider to be acting a single force, representing the whole set of contact forces with the ground. It progresses under the foot from the heel to the toe and can be calculated with the following equation.

$$CoP_x = - \frac{M_{Ry}}{F_{Rz}}; CoP_y = \frac{M_{Rx}}{F_{Rz}}; CoP_z = 0. \quad (2)$$

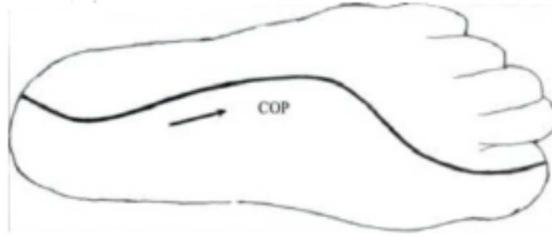


Fig 1.19 Progress of the COP under the foot during walking

We can now define the ZMP as Vukobratovich et al. did in 2001: “The pressure under supporting foot can be placed by the appropriate reaction force acting at a certain point of the mechanism’s foot. Since the sum of all moments of active forces with respect to this point is equal to zero, it is termed the ZMP.”. This definition was written analysing the walking during the single stance phase with the foot flat on the ground. Vukobratovich considers a robot’s foot, omitting the other parts of the body, replacing their influence with the force \vec{F}_{AK} and the torque \vec{M}_{AK} , and the foot’s centre of mass acts in its centre of gravity. The point P is the point in which the reaction forces of the ground acts on the foot, to provide the equilibrium. The resultant forces and torques are respectively \vec{R} and \vec{M} that can be divided in their components along the three axis x, y, z. Horizontal components of \vec{F}_{AK} are compensated by the friction forces acting on the ground, in no-sliding condition. In addition, the vertical component of \vec{M} represent the moment induced by the friction in opposition to \vec{M}_{AK} and \vec{F}_{AK} .

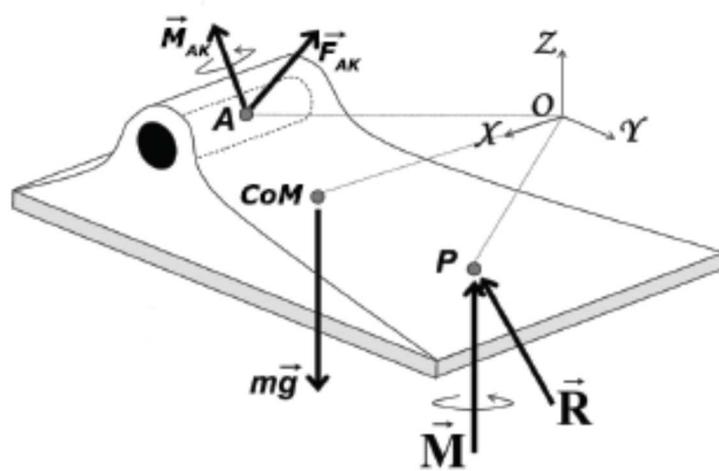


Figure 1.20. Support foot and the influence of the ankle by the force , moment , ground reaction by force and moment and gravity.

The necessary and sufficient condition for the stability is that no horizontal component of \vec{M} acts on P, because otherwise there will be an overturning of the body. In this case, P is called the “point of zero momentum” or ZMP. In addition, the ZMP can only lay inside the perimeter delimited by the support polygon in order to have stability.

Once having defined the ZMP we have to analyse the different strategies to locate it in the space. There are two different approaches [5]: the first one considers the whole body, its masses and positions; the second takes into account only the contact points and the normal forces generated in their surroundings.

The equations necessary to carry out the first approach are the followings:

$$p_{CoM} = \frac{\sum_{i=1}^n m_i p_i}{\sum_{i=1}^n m_i} \quad (3)$$

$$x_{ZMP} = \frac{m_{tot} g_z p_{CoMx} + z_{ZMP} \dot{P}_x - \dot{H}_y}{m_{tot} g_y + \dot{P}_z} \quad (4)$$

$$y_{ZMP} = \frac{m_{tot} g_z p_{CoMy} + z_{ZMP} \dot{P}_y - \dot{H}_x}{m_{tot} g_y + \dot{P}_z} \quad (5)$$

Where m_i indicates the mass of each individual link constructing the body, and p_i is the vector location of the centre of gravity for each link. n in the equation represents the total number of links. x_{ZMP} , y_{ZMP} , and z_{ZMP} represent the three components of the ZMP location. m_{tot} is the mass of the robot, and g_z is the gravity. H_x and H_y are the representations of the derivatives of the total angular momentum in the x- and y-directions, respectively. Furthermore, P_x , P_y , and P_z indicate the derivatives of the linear momentum.

Second approach needs to solve these other equations:

$$x_{ZMP} = \frac{\sum_{i=1}^n N_i x_i}{\sum_{i=1}^n N_i} \quad (6)$$

$$y_{ZMP} = \frac{\sum_{i=1}^n N_i y_i}{\sum_{i=1}^n N_i} \quad (7)$$

where N_i indicates the normal reaction force at one interaction point in the foot; n represents the total number of the interaction points for the model; x_i and y_i are the

locations of each interaction point in the x and y direction, respectively; and as mentioned previously, z_{ZMP} is equal to zero.

In the following figures it is possible to see the trends of the ZMP along the y-axis together with the CoG movement during walk.

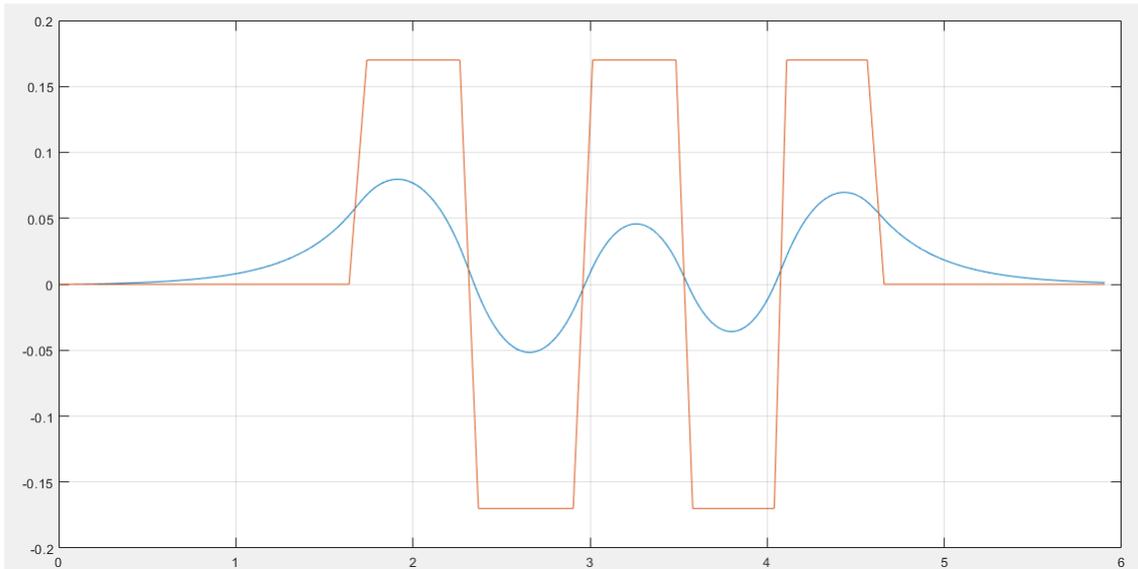


Figure 1.21 Ideal ZMP

1.5.2 Basometric signals

Baso signals are signals measured using switches under the foot in correspondence to the heel and the meta-toes articulation. These signals are a mean to analyse the quality of a walking pattern because they show all the events that must take place during walking, which are: heel strike, foot flat on the ground, heel parting from the floor and, lastly, swing phase. Each switch gives in return 1 if it is closed, 0 if it is open. For this reason, baso signals will have four steps representing the event and the time in which they take place. In the following figure it is possible to see the trend of these signals for each foot.

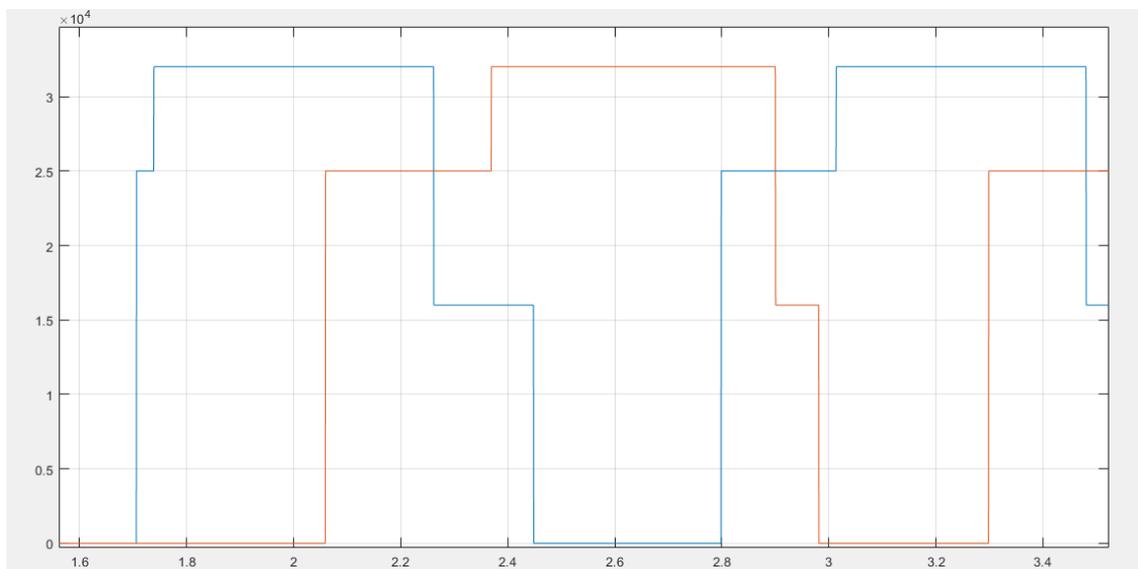


Figure 1.22 Ideal baso signals

2 Biped modelling

Up to now, we have been talking about the peculiarities of the human body: how it is composed, which muscles activate during walking, how the gait cycle is divided and on. In this chapter we will present the humanoid robot we have modelled in order to carry through different experiments.

2.1 CAD modelling

As said in the introduction to this thesis, our robot is called “Wilson”. Wilson represents only the lower part of the human body, and it is made of different parts whose measures are taken from standard human data. These parts are: “T” shaped trunk to represent the system trunk-hips-arms (HAT)^[17]; four joints able to implement the 3-dof hip; two thighs; two legs; two feet; two foot-tips. Talking about the degrees of freedom, it has 14. As the human body, the robot has hips, knees and ankles. In the following table it is possible to schematize the joints with the addition of the degrees of freedom they have.

Links	Joints
Trunk	Hips (3-dof each)
Thighs	Knees (1-dof each)
Legs	Ankles (2-dof each)
Feet	Tips (1-dof each)
Tips	

Table 2.1. Robot’s links and joints list

The main difference with other robots is that Wilson’s tips are not actuated by any mean; they are, hence, passive. This choice was taken in order to be able to create a system that could put in act not only stance and swing phases, but also all the phases that take place within the two mentioned before. In following paragraph we will have a closer look at the

techniques used in order to passivate the tip joint, and which are the consequence of this choice.

In the following pages it will be possible to see Wilson's parts and its assembly with measurements and tolerances.

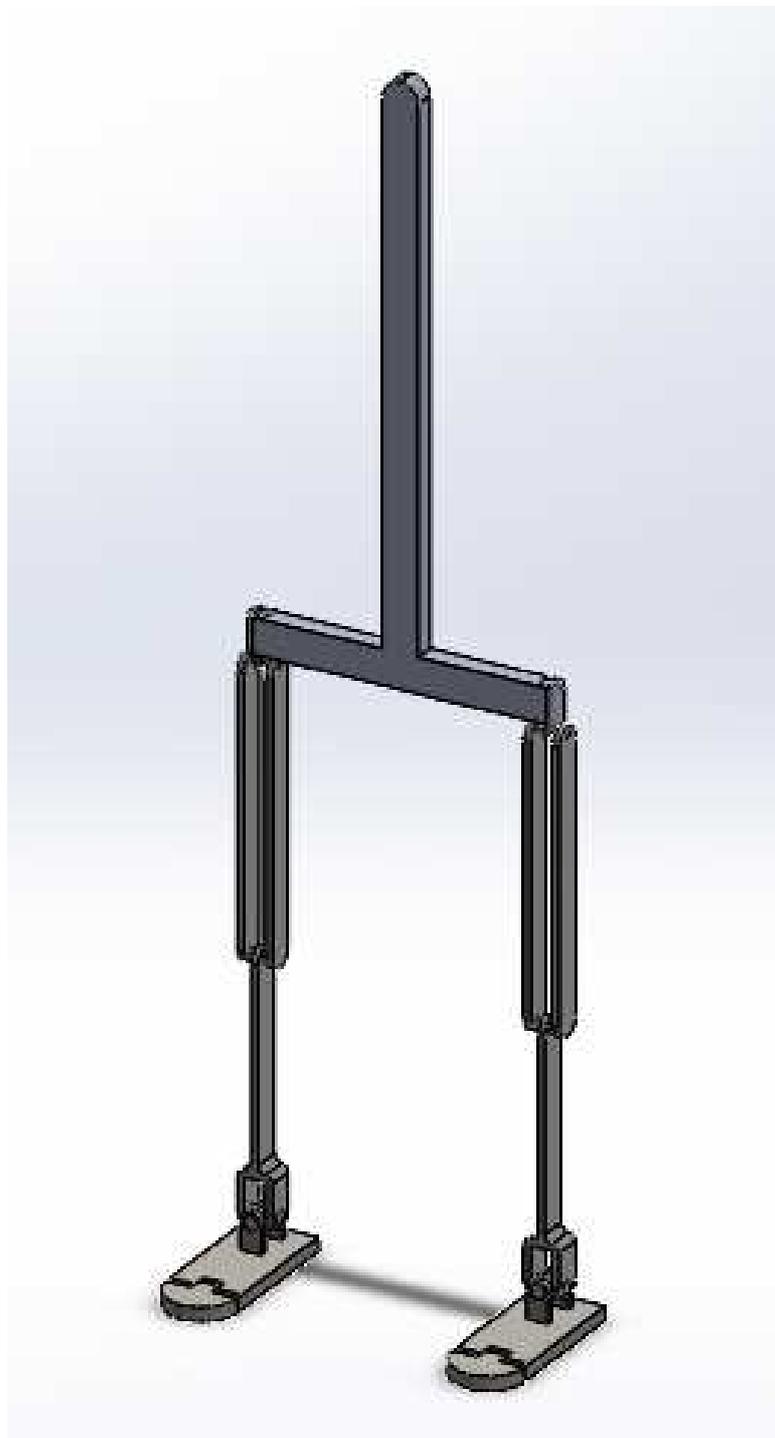


Figure 2.1 Robot assembly

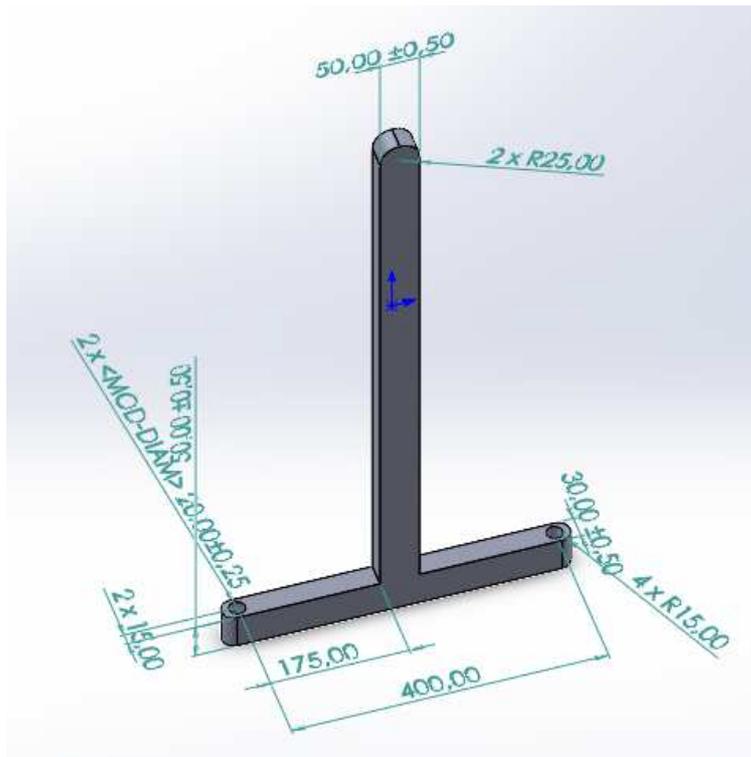


Figure 2.2 Robot's trunk

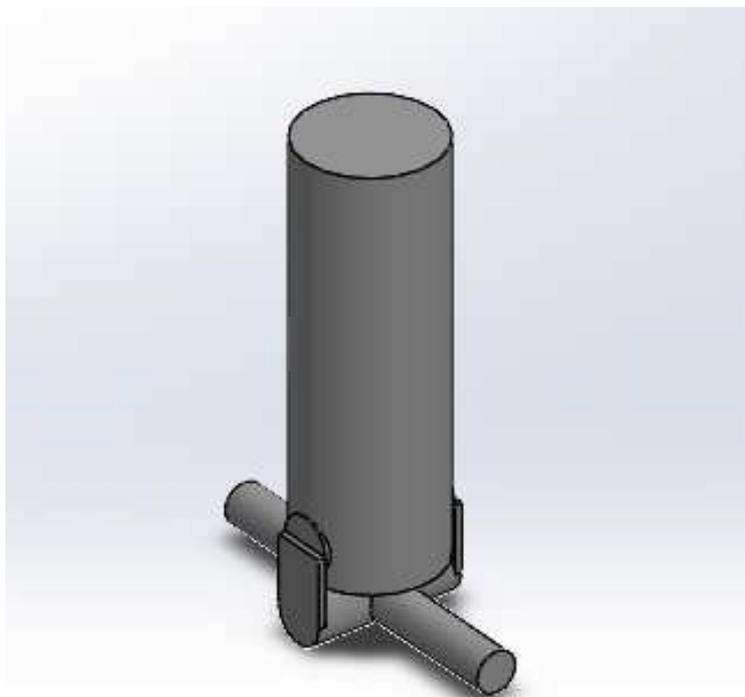


Figure 2.3 Robot's hip joint

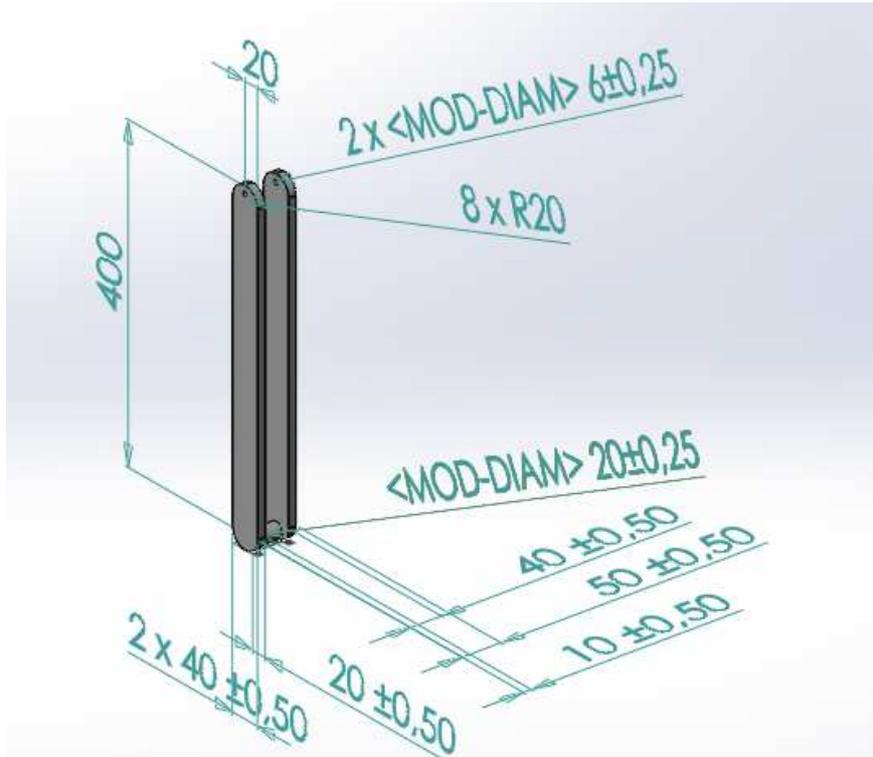


Figure 2.4 Robot's thigh



Figure 2.5 Robot's leg

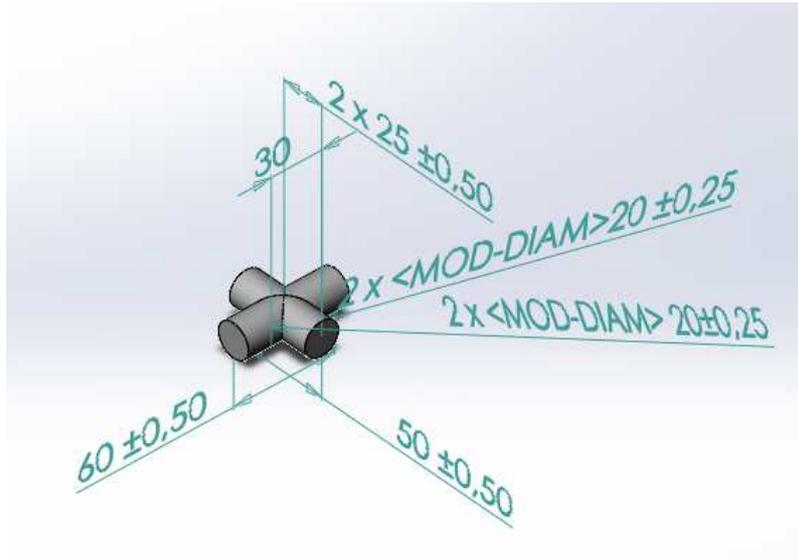


Figure 2.6 Robot's ankle

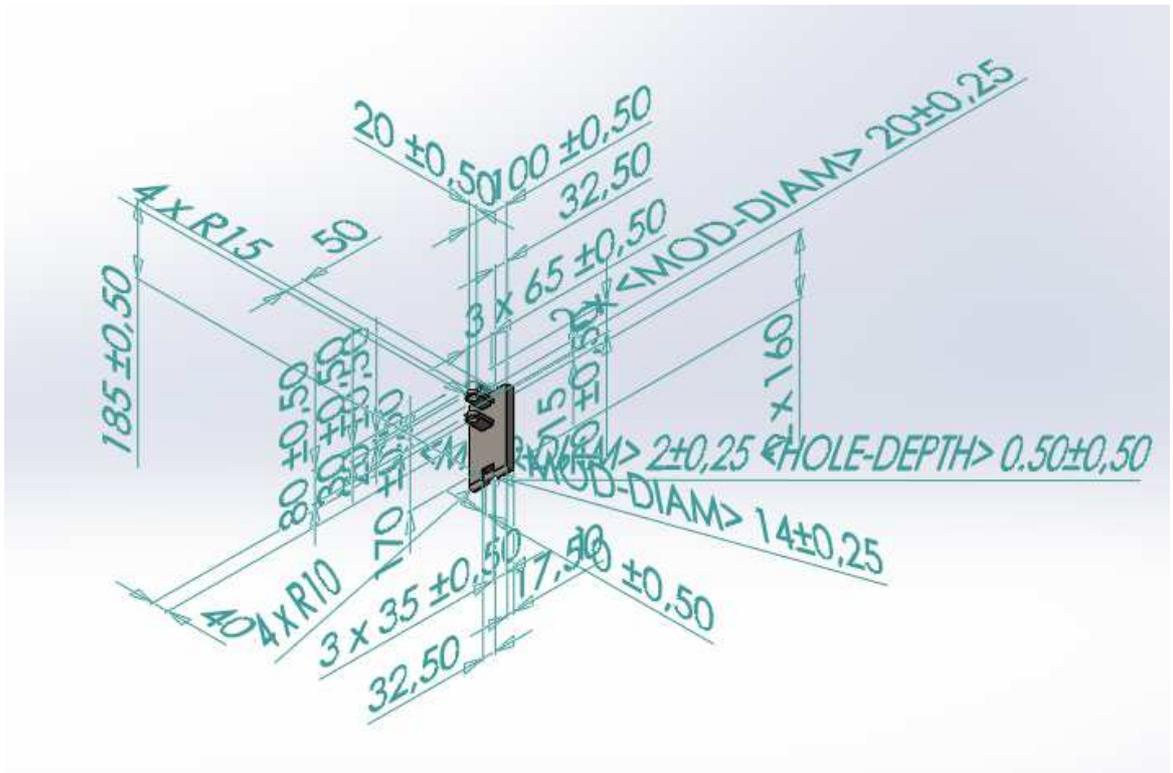


Figure 2.7 Robot's foot

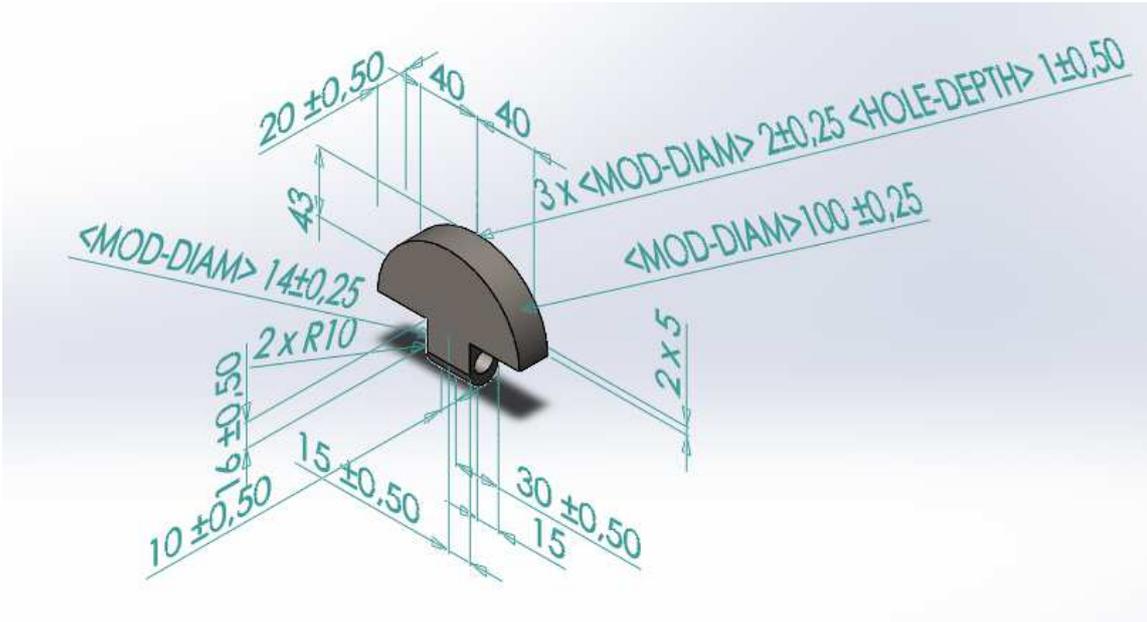


Figure 2.8 Robot's tip

Even though each part is modelled as an extended body, it will be treated as a simple “stick”, that is robotics’ language are called “links”. In the following table links’ measures are presented.

Link	Length (cm)
Trunk	90
Hip	20
Thigh	40
Leg	35
Foot (vertical)	5
Foot (horizontal)	17
Tip	5

Table 2.2 Robot's links' length

2.2 XLM file generation and analysis

In order to carry out simulations on the model, the CAD assembly is not sufficient. Hence, we need to convert it in an understandable language. All the information of Wilson are though translated directly from the CAD imported on the OnShape online platform into .xlm file, in order to be computed by the Simulink Simscape Multibody platform. In this file all the reference frames' transformations required to define the rigid body are listed and, in addition, they carry information about dynamical parameters as mass and inertial properties of the link. With information about the links, the file also carries the properties of the joints between the links. In the following pages it is possible to read the .xlm code defining our Wilson in the Matlab environment.

Before reading the code, it is useful to read a list of all the bodies and joints so be able to associate the numbers to the effective body and joint, that will be modified when acting on the Simulink Simscape Multibody project.

Number	Body	Joints
1	Left tip	Right tip
2	Left foot	Right ankle roll
3	Right leg	Left hip roll
4	Left leg	Left hip yaw
5	Right ankle	Right ankle pitch
6	Trunk	Right knee
7	Left hip yaw to roll	Right hip yaw
8	Right hip roll to pitch	Right hip roll
9	Right thigh	Left hip pitch
10	Left hip roll to pitch	Left ankle roll
11	Right hip yaw to roll	Left ankle pitch
12	Right tip	Left knee
13	Left thigh	Left tip
14	Left ankle	Right hip pitch
15	Right foot	/

Table 2.3 Robot's bodies and joint

2.3 Simulink's Simscape Multibody Model

After the generation of the .xlm code, followed by a check of all the bodies' measures, weights, centre of mass and joints' position and characteristics (they are all revolute, check was only carried out on the z-axis orientation), through the Matlab command "smimport" we have generated a first Simulink model. This model has been modified later on to introduce joints' properties such as stiffness and damping, actuation ports, sensing ports for position of all the joints and CoM and, lastly, algorithms for ZMP and basographic signals extraction. There will be a paragraph dedicated to the last two parameters. Now, we will give a quick look at the Simulink's Simscape Multibody Model, with its blocks and its parameters^[11].

First of all, it is necessary to define a world reference frame block, associated to other two configuration blocks which are the "mechanism configuration block" –in which gravity direction and magnitude is set, 9.81 m/s^2 - and the "solver configuration" block, required to set the solver to be used during simulation –default parameters were kept, considering the slow velocity the simulation will have to carry out. In the following figure we can see these three blocks and how they are linked together in order to be able to communicate later on with the system.

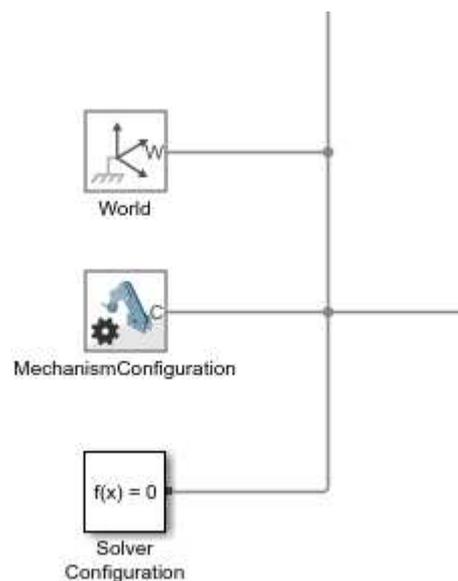


Figure 2.9 Mechanism configuration, solver configuration and world reference frame blocks

Secondly, after having set the three configuration blocks mentioned above, it is necessary to define a plane on which the robot will walk, and link it properly with the previous part of the project with a reference frame transformation block –pure translation. In addition, we will arrange in advance the future physical contact between the robot’s feet and the ground. We can see that the line between the body element “ground” is dashed because in the environment it expresses a physical connection between bodies.

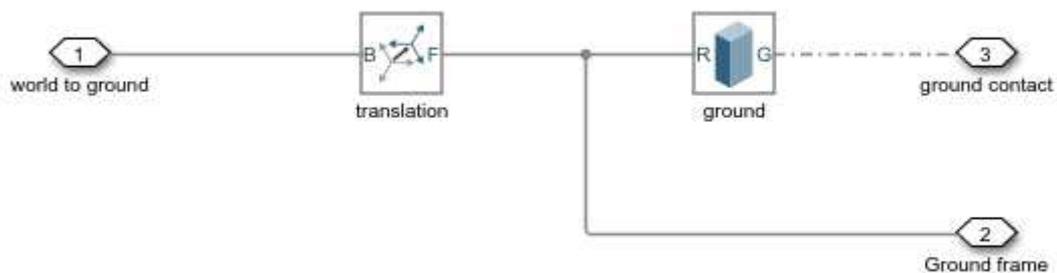


Figure 2.10 Ground Simulink block

Now, third action to make on the system is to link the world reference frame to our Wilson, through a simple transformation. Since, at the moment, the robot can freely move in the space, there is the need to constraint it, imposing different degrees of freedom depending on the different simulation we have to carry out. For this reason we decided to use a 6-dof joint for the first experiment –with Dr. Ali’s kinematic approach for biped robots- and a 3-dof joint for the second –in which we will simulate the robot’s movement giving as inputs joints’ angular position, recorded on different patients. In addition, it was added another reference frame to define the centre of gravity of the system. Since our robot does not have an upper body, we set its CoG at the hips’ centre, which is a sufficient approximation of a human’s body CoG, in order to study its position during walking, and underline the patient’s walking flaws.

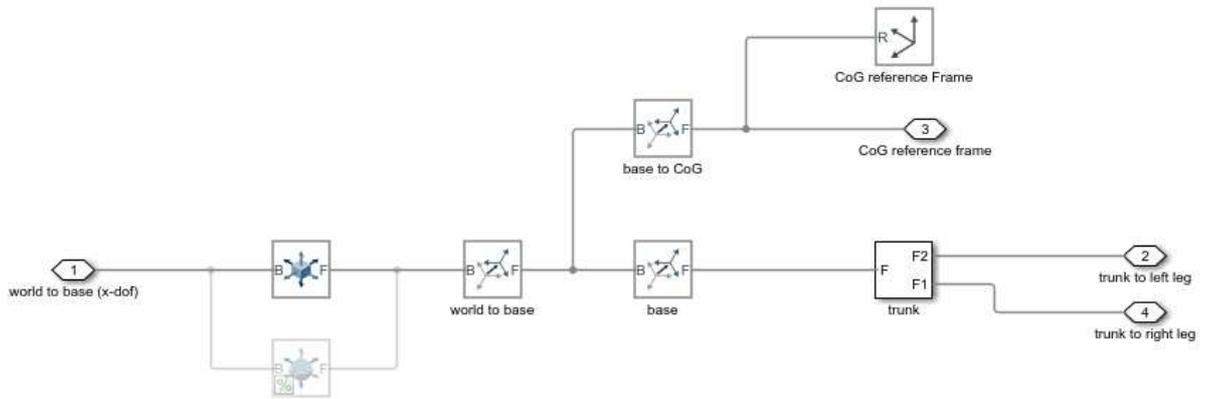


Figure 2.11 Trunk Simulink block

Now, as a fourth step, we have to link the legs to the trunk and to the ground. In addition it is necessary to arrange in advance an input port in order to be able to give the correct input to each joint. Lastly, output ports were displaced to have in return, from each leg, variables for ZMP calculation –further discussion in “ZMP calculation” paragraph, baso signals and sensing on each joint. In the following figure it is possible to see the whole right leg’s block, which is equally repeated for the left one.

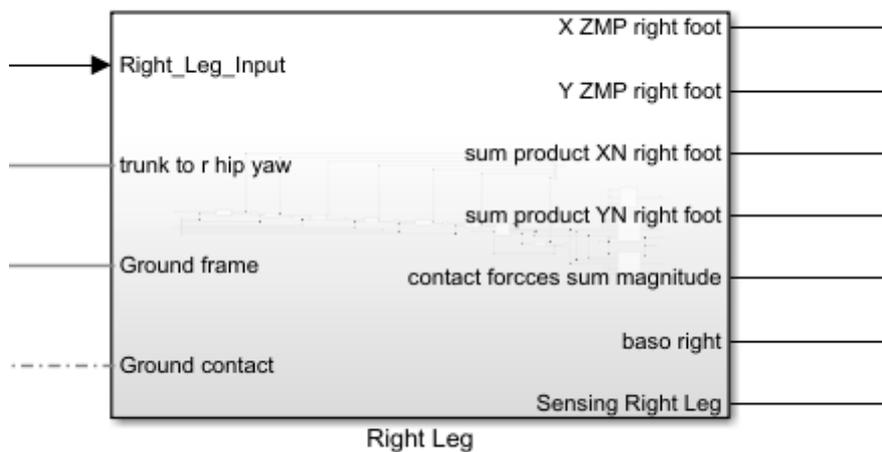


Figure 2.11 Right leg whole Simulink block

Inside the block presented above, there is a cascade of other blocks, representing, alternatively, a link and a joint, each associated to the correct body part. The following image shows a particular on the knee linking the right thigh to the right leg.

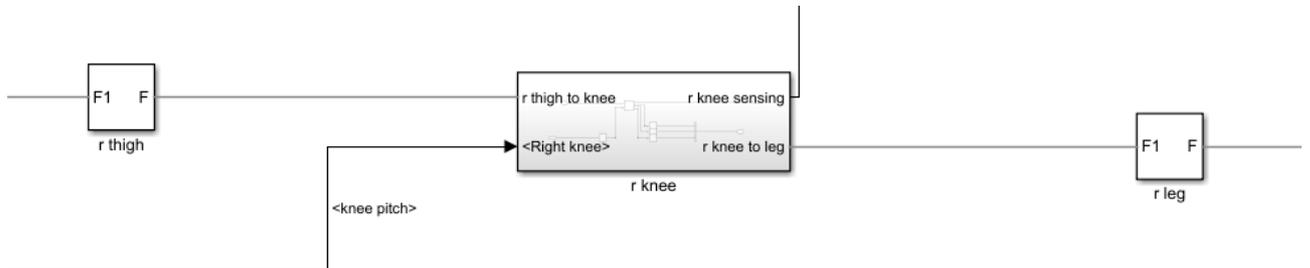


Figure 2.12 Particular on the leg's blocks cascade (right thigh linked to the right leg through the knee)

Each joint, as the knee, has four ports: two aimed to the connection between bodies – required for the definition of the parent and child-; one meant for input receiving and one for sensing providing. In order to make the input signals actuated by the joint, it is necessary to filter it. This action is accomplished through the Simulink-PS converter, that, filtering through a second-order function at the system's natural frequency (1 ms), converts the sampled data into a physical signal viable from the joint. Same happens in return, in the sensing extraction. Below we can see how each joint block is implemented.

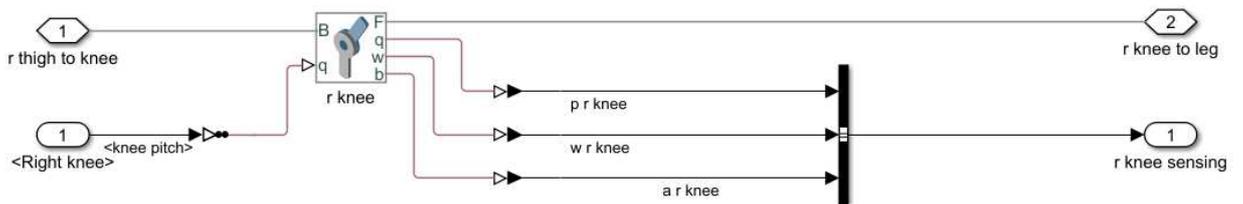


Figure 2.13 Knee joint block implementation

Lastly, each joint has to be tuned in order to get the desired dynamics. The main parameters to be introduced are damping, stiffness and angular limits. Damping was set to a very high value ($39000 \text{ N}\cdot\text{m}/(\text{deg}/\text{s})$) in order to avoid any kind of small oscillation of the joint and receive more position oriented data. Same was done for the stiffness ($1100000 \text{ N}\cdot\text{m}/\text{deg}$). Angular limits were taken from literature, in order to avoid hyperextension or flexion.

2.4 Ground contact modelling

Walking, as already said, is the most natural task our body can put in act but it does not depend only on our body itself. Walking's efficiency and precision depends mainly on the interaction between the body and the surrounding environment. Let us go through this topic with few simple examples. The first one is the surely the easiest-handling interaction we can find: shoes-asphalt. In this case, when walking, we just push our body forward, through the gastrocnemius' propulsion. Here, the foot is firmly hard-linked with the ground and the contact point acts as a rotation fulcrum. The second example represented by the condition of walking with wool socks on the parquet. It is possible to walk, but with more attention because if we even attempt a slow run, our feet will slip on the floor, with the risk of falling. The third and most extreme example is found in the interaction between our feet with a soaped-wet rubber floor. In this case it is almost impossible to walk as we are used to do, normally. In this paragraph, hence, we will analyse the physical laws that need to be satisfied in order to avoid slipping and, as a consequence, equilibrium when walking. After the theoretical part, we will explain how to implement these laws into the Simulink's Simscape Multibody model.

2.4.1 Frictional force

Frictional force is the force that lies tangentially in the contact frame and perpendicular to the normal force. This force express the difficulty two bodies find to show a relative slide and it depends directly to the force that pushes them together. Friction can be static –when bodies do not show relative motion- and kinetic –between moving bodies. The ratio between the friction force and the normal force applied in the contact frame is called “friction coefficient” expressed with the greek letter “ μ ”. Also this parameter can be static or dynamic, sharing the same circumstances of the force.

The following equation expresses the relation between frictional force and normal force, through the friction coefficient.

$$F_f = \mu \cdot F_N \quad (1)$$

The threshold between static and kinetic friction is represented by the critical velocity. When the critical velocity is equal the magnitude of the tangential velocity, the effective coefficient of friction is equal to the specified Coefficient of Static Friction. As the magnitude of the tangential velocity increases beyond this value, the effective coefficient of friction asymptotically approaches the specified Coefficient of Dynamic Friction. In the following graph it is possible to see the relation between friction coefficient and their relative velocity.

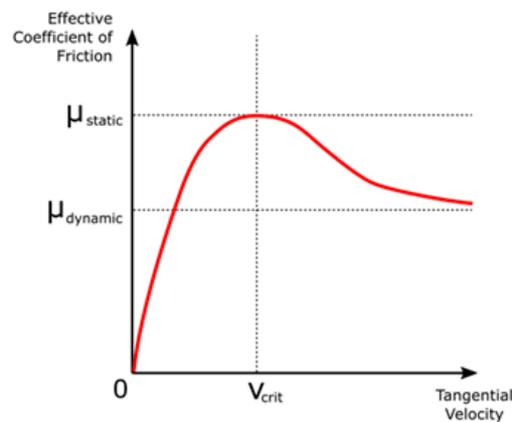


Figure 2.14 Friction coefficient in function of tangential velocity

2.4.2 Simulink blocks for contact modelling with parameters tuning

In order to create an application point or the frictional force between foot and ground, it was necessary to find the regions, under the human foot, of more intense pressure during the walking task. These regions coincide with the points in which switches for basographic signals measurement are placed (heel and metatarsal –inner and outer)^[13].

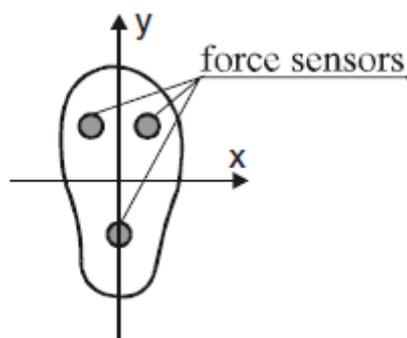


Figure 2.15 Sensor placed under the foot for baso extraction

The switches placement was carried out on the robot through Simscape's Multibody blocks with three consecutive steps:

1. Definition of the position of the contact point through three reference transformations;
2. Link of three small spheres to the reference frames define above under each foot through a "solid body" block;
3. Tuning of the contact parameters through the "spatial contact force" block.

The parameter that have been used, after a process of trial and error, are the following:

$\mu_s = 0.9$; Static friction coefficient: Around that of rubber-asphalt

$\mu_d = 0.8$; Kinetic friction coefficient: Lower than the static coefficient

$\mu_{vth} = 0.1$; Friction velocity threshold (m/s) –critical velocity

$contact_stiffness = 200/0.001$;

$contact_damping = contact_stiffness/10$;

The last two parameters represent are set to a very high value in order to have a minimal bouncing influence ,in return from the ground.

In the pictures below it is possible to analyse Simulink's blocks implementation and the visual result under the effective robot's foot. Other five spheres were added in order to increase the region in which the contact occurs in order to increase the stability of the robot.

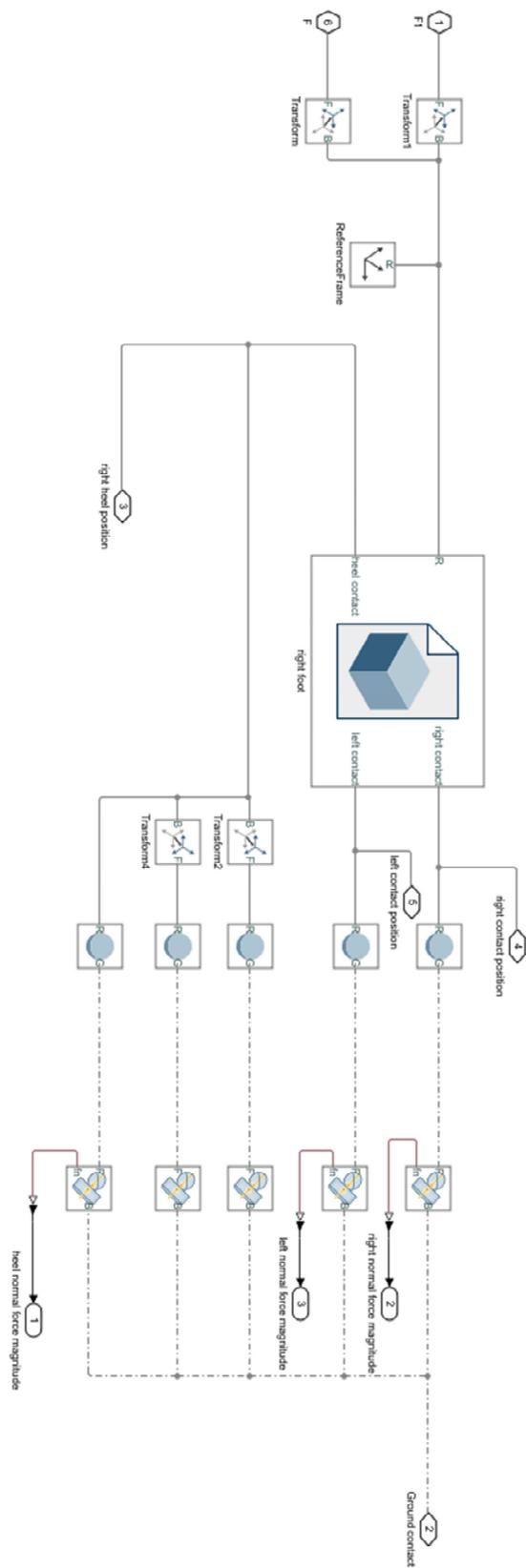


Figure 2.16 Simulink implementation of the contact with the ground

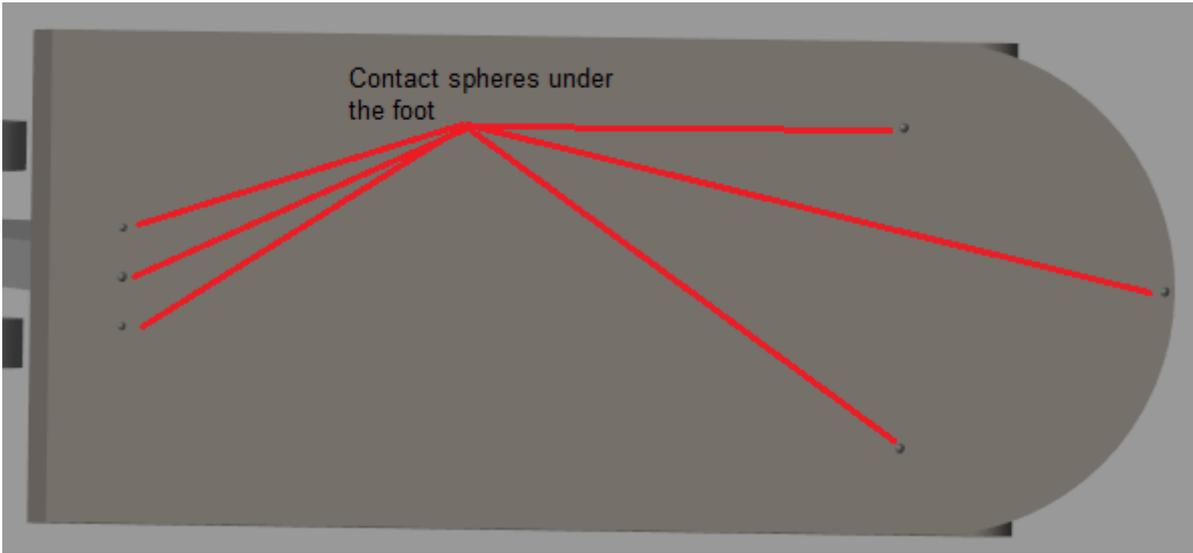


Figure 2.17 Contact spheres under the robot's foot

2.5 Passive toe joint implementation

We have already shown in the previous paragraphs that in each joint in our model has been implemented a port for the control input. This happened in all the joints of the project except one: the toe joint (called in the project as “tip joint”). This joint has been implemented as completely passive. Hence, the dynamics of the system “tip joints” are controlled by only a combination of a rotational spring and a rotational damper^[13].

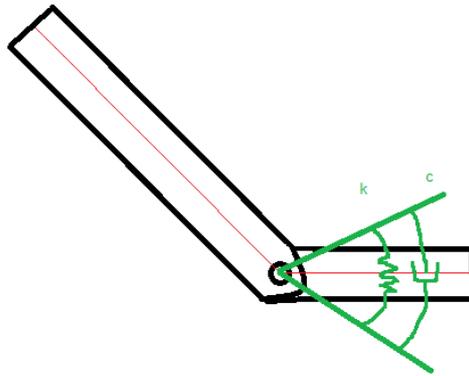


Figure 2.18 Passive toe joint representation

The angular value in which both spring and damper are in their equilibrium position, called “equilibrium value” is set, according to the literature, at 180° (flat foot). In addition, always according to the literature, an upper and lower angular limit has been set to the joint in order to avoid hyper-dorsiflexion or hyper-extension of the joint. Stiffness k and damping c were set through a process of trial and error (starting from the values of $k=0.10$ N/m and $c=0.05$ N*s/m) until the correct behaviour of the system was reached in terms of ZMP and baso signals. In the following table is possible to see the magnitude of all the parameters needed for the joint modelling.

Variables	Value
Upper limit	90°
Lower limit	0°
k	0.6 N/m
c	0.025 N*s/m

Table 2.4 Passive toe joint parameters

The Simulink implementation is similar to the one of the other joints, except for the missing of the input port in the joint because its only aim is to adapt its position to the surrounding environment.

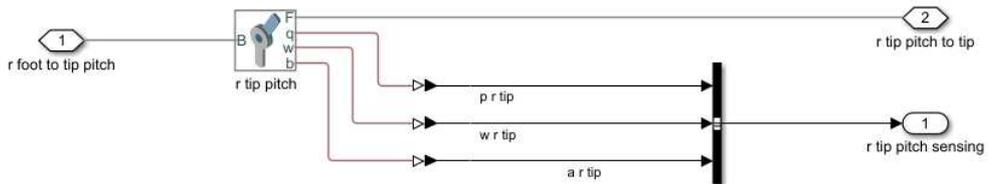


Figure 2.19 Passive toe joint Simulink implementation

2.6 ZMP and Basometric signals extraction

As said in paragraph 1.5, in order to analyse a walking pattern, angular measurements are not enough. Other two variables have to be analysed, that are the Zero Moment Point and the basometric signals. In this paragraph we will go through the implementation of the Simulinks' blocks required for their extraction, in order to be analysed later on.

2.6.1 ZMP algorithm and blocks

By taking into account the equation of the ZMP we can see that it carries all the information needed for both ZMP and baso signals^{[13][14]}.

$$x_{ZMP} = \frac{\sum_{i=1}^n N_i x_i}{\sum_{i=1}^n N_i} \quad (6)$$

$$y_{ZMP} = \frac{\sum_{i=1}^n N_i y_i}{\sum_{i=1}^n N_i} \quad (7)$$

In paragraph 2.2.4 we have presented how the foot is built, and through the “spatial contact force” block it is possible to find the magnitude of the normal force acting between the sphere and the ground, finding the N_i term. In addition, by saving the position of the spheres' reference frames in a dedicated variable, we can find both x_i and y_i terms. With the product of the normal force with the position of the sphere in which it is calculated, and summing all the terms, we obtain the numerator of the equations 6 and 7. By summing all the normal forces, we obtain the denominator. In Simulink, this algorithm is implemented in the blocks presented in the following figure.

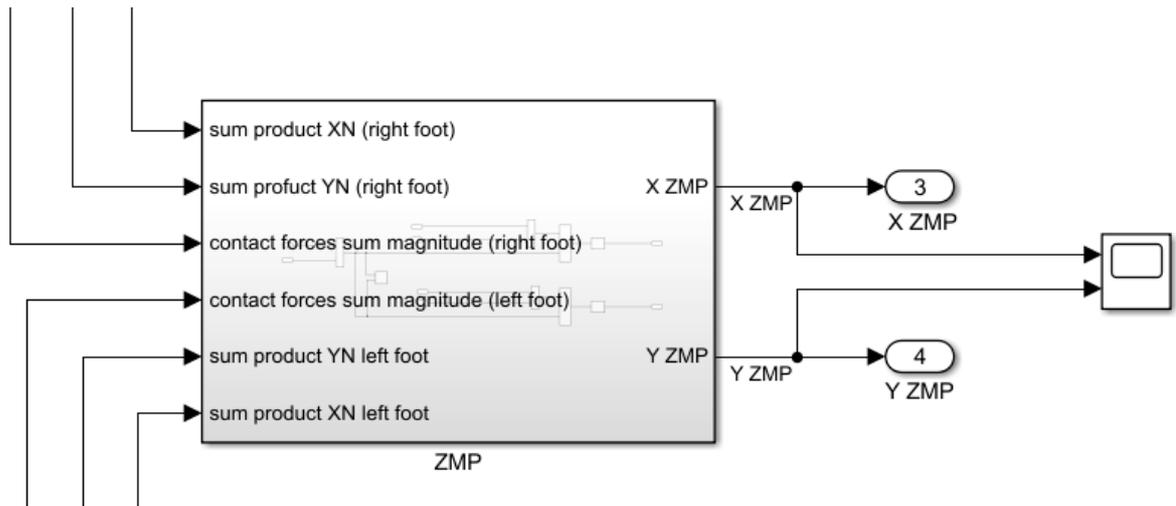


Figure 2.20 ZMP Simulinks block

2.6.2 Basometric signals extraction

For each sphere placed under the robot's foot, as said before, is possible to measure the normal force acting between the sphere and the ground. In order to extract the baso signal, the only action to carry through is comparing the normal force at each sphere to a constant signal -0 in our case- and, if the force is found to have a higher magnitude than the constant signal, it assumes the value of 1. By summing all the 0 and 1 found in the whole foot, we find the baso signal of the foot, at each time instant. In addition, a low-pass filter is added in order to avoid small bouncing and the floor, finding a continue and stable signal. In the following figure is shown the simulinks's implementation.

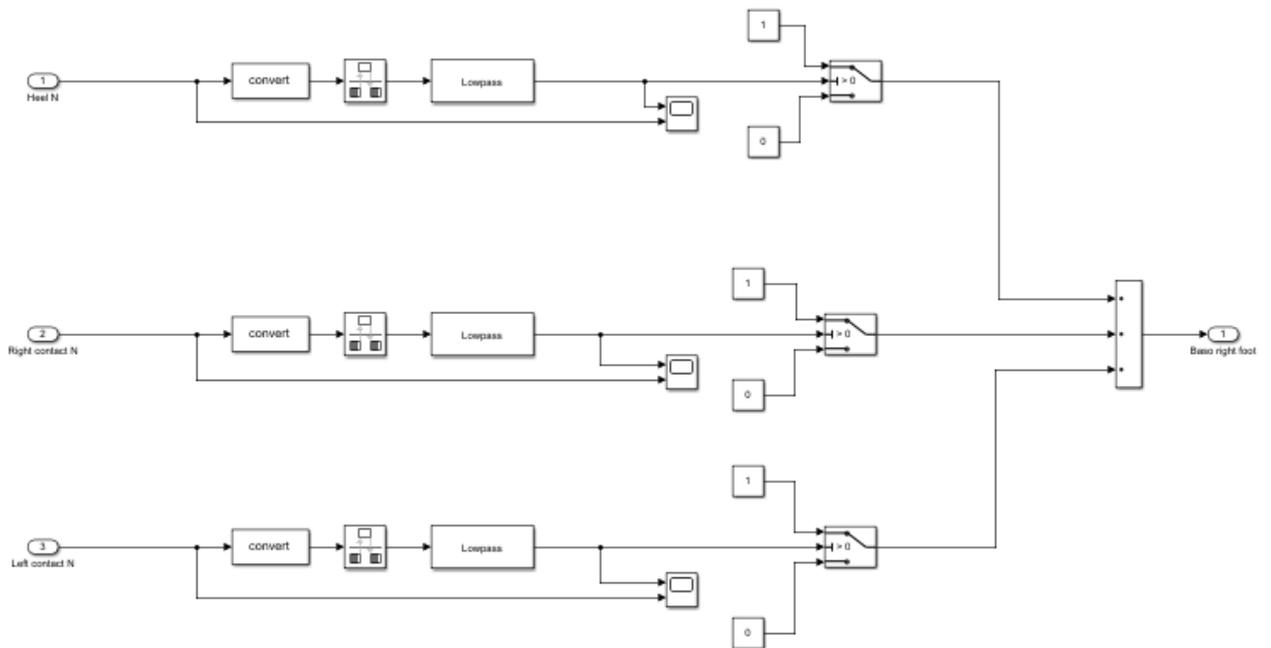


Figure 2.21 Baso signal algorithm simulinks' implementation

3 Simulations

After having created the CAD and Simulink model, we are going to simulate different walking patterns through the Simscape Multibody platform. A complete set of joint variables will be given to the robot in order to make it move its first steps and, later on, to find suitable settings aimed to an exoskeleton's development. In this last chapter we will carry out three simulations.

The first simulation consists in recreating the walking pattern through an inverse kinematic approach, developed by Dr. Ali et. All in 2015^[12], especially purposed for biped walkers. With this approach we will find a closed form solution for our biped, using, as first model the linear inverted pendulum model developed by Kajita^[10].

In the second simulation, a complete dataset of joint variables will be given to our robot. This dataset consists in angular measurements of each articulation of a patient walking on a treadmill. Through this experiment we will be able to recreate the patient's walking pattern in order to analyse all the little flaws in his walk. In addition, we will look at important variables such as ZMP, baso signals, CoG position to have a deeper look at the complex task which is walking.

The third experiment will use as control input the same dataset of the previous one, but, this time, some constraints will be added to the system in order to find a more stable walking pattern, simulating the exoskeleton's purpose of rehabilitation.

3.1 Inverse kinematics approach

As previously said, in this paragraph we will build a complete set of variables for our robot, starting from Kajita's linear inverted pendulum model, concluding with the closed form solution for biped robots developed by Dr. Ali et al. We will discuss the matlab implementation of both the models, with a discussion on the results. Lastly, we will see the behaviour of our robot when receiving the dataset as input.

3.1.1 The Linear Inverted Pendulum Model

Having analysed the gait cycle, especially the single stance phase with the free fall stopped with the heel strike of the opposite foot, is clear that we act, in that time interval, as an inverted pendulum driven by only the gravity force, with zero input. Kajita, hence, modelled our body as an inverted pendulum, constrained on the ground in one single point, with all the mass collapsed in the CoM^[10].

The position of the point mass can be defined through the position vector $\mathbf{p} = (x, y, z)$ and it is uniquely specified by a set of state variables $\mathbf{q} = (\theta_r, \theta_p, r)$.

$$x = r \cdot S_p, \quad (1)$$

$$y = -r \cdot S_r, \quad (2)$$

$$z = r \cdot D, \quad (3)$$

$$S_r \equiv \sin \vartheta_r, S_p \equiv \sin \vartheta_p, D \equiv \sqrt{1 - S_r^2 - S_p^2}.$$

Let (τ_r, τ_p, f) be the actuator force associate with the sate variables (θ_r, θ_p, r) . With these inputs, the equation of motion of the 3D inverted pendulum in Cartesian coordinates is given as follows.

$$m \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} = (J^T)^{-1} \begin{pmatrix} \tau_r \\ \tau_p \\ f \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -mg \end{pmatrix} \quad (4)$$

Where m is the mass of the pendulum and g is the gravity acceleration.

The structure of the Jacobian J is given as follows:

$$J = \frac{\partial \mathbf{p}}{\partial \mathbf{q}} = \begin{pmatrix} 0 & rC_p & S_p \\ -rC_r & 0 & -S_r \\ -\frac{rC_r S_r}{D} & -\frac{rC_p S_p}{D} & D \end{pmatrix}, \quad (5)$$

$$C_r \equiv \cos \vartheta_r, C_p \equiv \cos \vartheta_p.$$

To erase the inverse Jacobian that appears in (4), let us multiply the matrix J^T on the left term.

$$m \begin{pmatrix} 0 & -rC_r & -\frac{rC_r S_r}{D} \\ rC_p & 0 & -\frac{rC_p S_p}{D} \\ S_p & -S_r & D \end{pmatrix} \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} = \begin{pmatrix} \tau_r \\ \tau_p \\ f \end{pmatrix} - mg \begin{pmatrix} -rC_r S_r \\ -\frac{rC_p S_p}{D} \\ D \end{pmatrix} \quad (6)$$

Using the first row of (6) and multiplying it for D/C_r we find:

$$m(-rD\ddot{y} - RS_r\ddot{z}) = \frac{D}{C_r} \tau_r + rS_r mg. \quad (7)$$

With the substitution of (2) and (3) we get the equation describing the dynamics along the y-axis:

$$m(-z\ddot{y} + y\ddot{z}) = \frac{D}{C_r} \tau_r - mgy \quad (8)$$

The same procedure used for the first row can be carried out on the second one, leading to the equation describing the dynamics along the x-axis:

$$m(z\ddot{x} - x\ddot{z}) = \frac{D}{C_p} \tau_p - mgx \quad (9)$$

In order to select the class of motion of the pendulum more suitable for a human walking, it is necessary to add some constraints to the system. The first and more important one is limiting the motion to a plane with a given normal vector $(k_x, k_y, -1)$ and z intersection z_c .

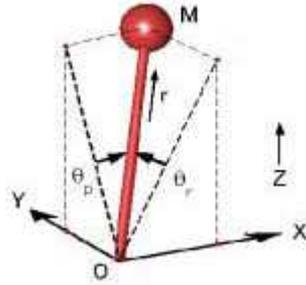
$$z = k_x x + k_y y + z_c \quad (10)$$

Considering that the robot walks on flat terrain, we can set the horizontal constraint plane ($k_x = 0, k_y = 0$) to obtain the following second derivative it is necessary to substitute the previous constraints into (8):

$$\ddot{y} = \frac{g}{z_c} y - \frac{1}{mz_c} u_r \quad (11)$$

$$\ddot{x} = \frac{g}{z_c} x - \frac{1}{mz_c} u_p \quad (12)$$

These last two equations are two independent linear equations and the only parameter that governs their dynamics is z_c .



Let us now analyse the nature of the trajectories and the trajectories themselves in the case of LIPM when driven by zero input torques. In this case only the gravity force acts on the point mass.

The equations, with given initial conditions, are the following:

$$\ddot{y} = \frac{g}{z_c} y \quad (13)$$

$$\ddot{x} = \frac{g}{z_c} x \quad (14)$$

The next step is represented by the division of the gravity force in the two directions x and y , through the Kepler's second law (areal velocity conservation). Hence, we obtain the following set of two equations. We call it a "set" because of the presence of the x term in the y equation and vice versa.

$$f_{G_x} = \frac{x}{r} \left(-\frac{k_G}{r^2} \right) = -\frac{x k_G}{(x^2 + y^2)^{\frac{3}{2}}} \quad (15)$$

$$f_{G_y} = \frac{y}{r} \left(-\frac{k_G}{r^2} \right) = -\frac{y k_G}{(x^2 + y^2)^{\frac{3}{2}}} \quad (16)$$

By integrating eq (13) and (14), respectively, we obtain the parameter called by Kajita “orbital energy”, which is presented in the following equations:

$$E_y = -\frac{g}{2z_c} y^2 + \frac{1}{2} \dot{y}^2 \quad (17)$$

$$E_x = -\frac{g}{2z_c} x^2 + \frac{1}{2} \dot{x}^2 \quad (18)$$

When the coordinate frame XY is rotated of an angle θ we obtain the new coordinate frame X'Y', which, given the discussion above, can be used to give a proper representation of the LIPM. We obtain the new orbital energy given by:

$$E'_x = -\frac{g}{2z_c} (cx + sy)^2 + \frac{1}{2} (c\dot{x} + s\dot{y})^2 \quad (19)$$

$$E'_y = -\frac{g}{2z_c} (-sx + cy)^2 + \frac{1}{2} (-s\dot{x} + c\dot{y})^2 \quad (20)$$

$$c \equiv \cos \vartheta, s \equiv \sin \vartheta$$

With simple calculations we can verify that the total energy of the system remains constant, regardless the chosen coordinate frame.

$$E'_x + E'_y = E_x + E_y = \text{const} \quad (21)$$

When the Y'-axis is the axis of symmetry, E'_x and E'_y become respectively the maximum and minimum. Given that, we can calculate the axis of symmetry by solving the following equation:

$$\frac{\partial E'_x}{\partial \vartheta} = A(s^2 - c^2) + Bsc = 0 \quad (22)$$

Where

$$A \equiv \left(\frac{g}{z_c} \right) xy - \dot{x}\dot{y} \quad (23)$$

$$B \equiv \left(\frac{g}{z_c} \right) (x^2 - y^2) - (\dot{x}^2 - \dot{y}^2) \quad (24)$$

We can now distinguish three different solution, switching the values of A and B.

$$B \neq 0, \quad \vartheta = \frac{1}{2} \tan^{-1} \left(2 \frac{A}{B} \right) \quad (25)$$

$$B = 0, A \neq 0, \quad \vartheta = \frac{\pi}{4} \quad (26)$$

$$B = 0, A = 0, \quad \vartheta = \text{atan2}(y, x) \quad (27)$$

If the Y-axis is already the axis of symmetry, θ is supposed to be zero.

Taking into account eq. (25) and (23), we need to satisfy the following condition:

$$\left(\frac{g}{z_c} \right) xy + \dot{x}\dot{y} = 0 \quad (28)$$

Substituting eq (17) and (18) in (28), we obtain the geometric equation of the 3D-LIPM

$$\frac{g}{2z_c E_x} x^2 + \frac{g}{2z_c E_y} y^2 + 1 = 0 \quad (29)$$

In addition we can say that, since $E_x > 0$ and $E_y < 0$, eq. (29) forms a hyperbolic curve, typical shape of the bodies in Kepler motion.

3.1.2 Closed-Form Inverse Kinematic Joint solution for Humanoid Robots

The most common control method adopted for humanoid robots are surely the iterative ones, in which Jacobian matrixes are used. This method, even though is the most usual and commonly approved, carries many drawbacks such as singularities, redundancies and lastly computational complexity. In this chapter we will use the approach presented by dr. Ali in 2015 “Closed-Form Inverse Kinematic Joint solution for Humanoid Robots”^[12]. Here, instead of the velocity based approach of the Jacobian method, dr. Ali uses a position based one. Before going through the presentation of the method, it is necessary to define which are the necessary conditions required for finding a closed-form solution^[x]. The conditions are essentially two and they do not require to be both satisfied. The first one is the parallelism condition of three adjacent joint axes of the robot, while the second one requires the intersection in one point of three adjacent joint axes. After this short presentation of the topic, we can now carry the calculations required for finding the solution.

The first step is the definition of the D-H parameters in a table, according to the reference frames defined in the following figure, for the right leg.

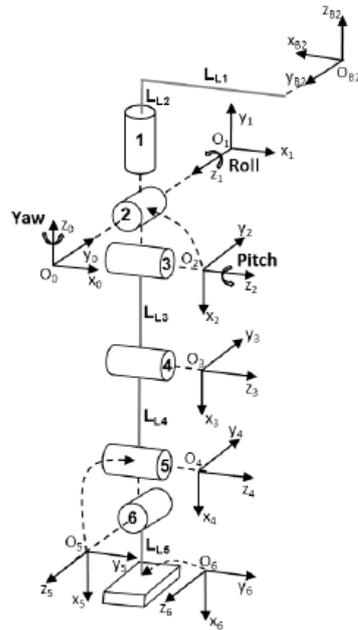


Figure 3.2 Link coordinate frames of the robot's right leg

Right leg link coordinate parameters

Joint i	a_i	α_i	d_i	θ_i
1	L1	0	-L2	0
2	0	$-\pi/2$	0	0
3	0	$-\pi/2$	0	0
4	L3	0	0	0
5	L4	0	0	0
6	0	$\pi/2$	0	0
7	L5	0	0	0

Table 3.1 D-H parameters for the robot's right leg

In the θ_i column, all the parameters are set to zero because they are the variables we want to later calculate. In addition, the L_i parameters are defined according to the length of the links.

Now, as second step is necessary to define a desired position of the end-effector (the foot in our case). We decided to define the foot trajectory by linking the waypoints in the LIPM, calculated according to our robot, with a cubic trajectory.

After having defined the end-effector trajectory, we can carry out the necessary ik calculations. First of all, taking into account Pieper^[x], we know that if three adjacent joint axes do not intersect at a point, the position vector \mathbf{p} is a function of the four joint angles $\theta_4, \theta_5, \theta_6, \theta_7$. So it is merely impossible to find a closed form solution for our robot. But with a closer look we can see that the first three joints' axes do intersect at a point, so, according to Pieper's theory, the position vector \mathbf{p} decouples the limb into positioning and orientation subsystems. In this case, we find three equations with known variables p_x, p_y, p_z , and unknown joint angles $\theta_1, \theta_2, \theta_3$.

Now, we have to invert the composite transformation matrix \mathbf{T} , finding the \mathbf{T}' , matrix.

$$T' = \begin{bmatrix} \mathbf{n} & \mathbf{s} & \mathbf{a} & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{n}' & \mathbf{s}' & \mathbf{a}' & \mathbf{p}' \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}_5^6\mathbf{A} {}_4^5\mathbf{A} {}_3^4\mathbf{A} {}_2^3\mathbf{A} {}_1^2\mathbf{A} {}_0^1\mathbf{A} = {}_0^6\mathbf{A} \quad (30)$$

Having defined the base coordinate frame B2, attached to the waist to the first coordinate frame, we can thus define the transformation between B2 and the reference frame 0:

$${}^{B2}_0\mathbf{A} = \begin{bmatrix} -1 & 0 & 0 & l_{L1} \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & l_{L2} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (31)$$

To obtain the solution for $\theta_4, \theta_5, \theta_6$ we equate the elements of \mathbf{p}' in both sides of eq (30), to obtain:

$$-C_6(C_{45}l_{L3} + C_5l_{L4}) = p'_x + l_{L5} \quad (32)$$

$$-S_6(C_{45}l_{L3} + C_5l_{L4}) = p'_y \quad (33)$$

$$-S_{45}l_{L3} + S_5l_{L4} = p'_z \quad (34)$$

$$S_i \equiv \sin \theta_i, C_i \equiv \cos \theta_i, S_{ij} \equiv \sin(\theta_i + \theta_j), C_{ij} \equiv \cos(\theta_i + \theta_j)$$

We can obtain θ_4 calculating C_4 and S_4 firstly, by squaring and adding the equations above

$$C_4 = \frac{(p'_x + l_{L5})^2 + p'^2_y + p'^2_z - l_{L3}^2 - l_{L4}^2}{2l_{L4}l_{L3}}$$

$$\theta_4 = \text{atan2}\left(\pm\sqrt{1 - C_4^2}, C_4\right) \quad (35)$$

We can now move on to the calculations to obtain θ_5 . Manipulating eq (32) and (33) we obtain:

$$C_5(C_4l_{L3} + l_{L4}) - S_4S_5l_{L3} = \pm\sqrt{(p'_x + l_{L5})^2 + p'^2_y} \quad (36)$$

Expanding eq (34), we obtain:

$$S_5(C_4l_{L3} + l_{L4}) + C_5(S_4l_{L3}) = -p'_z \quad (37)$$

Let us now define $C_4l_{L3} + l_{L4} = rC_\psi$ and $S_4l_{L3} = rS_\psi$ and substituting them into eq (36) and (37) correspondingly, we obtain:

$$rC_{5\psi}\sqrt{(p'_x + l_{L5})^2 + p'^2_y} \quad (38)$$

$$rS_{5\psi} = -p'_z \quad (39)$$

$$r = \sqrt{(p'_x + l_{L5})^2 + p'^2_y + p'^2_z}, \quad \psi = \text{atan2}(S_2l_{L3}, C_4l_{L3} + l_{L4})$$

We can now obtain the joint solution for θ_5 by dividing eq (39) by eq (38):

$$\theta_5 = \text{atan2}\left(-p'_z, \pm\sqrt{(p'_x + l_{L5})^2 + p'^2_y}\right) - \psi \quad (40)$$

We can finally get the solution for θ_6 by dividing eq (33) by eq (32).

$$\theta_6 = \text{atan2}(p'_y, -p'_x - l_{L5}) \quad (41)$$

$$\text{if } C_{45}l_{L3} + C_5l_{L4} < 0, \quad \text{we have } \theta_6 = \theta_6 + \pi$$

We can now move to the calculation required to obtain the first three joint variables. We will use the inverse transformation method ^[x] by moving the term 6_5A to the left-hand side of eq (30), obtaining another matrix that we will call G_2 . We will now analyse the left-hand side and the right-hand side of G_2 , respectively.

$$G_{2-leg}^{(LHS)} = {}^5_6A \begin{bmatrix} n' & s' & s' & p' \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} C_6 n'_x - S_6 n'_y & C_6 s'_x - S_6 s'_y & C_6 a'_x - S_6 a'_y & C_6 p'_x - S_6 p'_y + C_6 l_{L5} \\ S_6 n'_x + C_6 n'_y & S_6 s'_x + C_6 s'_y & S_6 a'_x + C_6 a'_y & S_6 p'_x + C_6 p'_y + S_6 l_{L5} \\ n'_z & s'_z & a'_z & p'_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$G_{2-leg}^{(RHS)} = {}^5_0A = {}^5_4A {}^4_3A {}^3_2A {}^2_1A {}^1_0A =$$

$$= \begin{bmatrix} C_1 C_2 C_{345} - S_1 S_{345} & S_1 C_2 C_{345} + C_1 S_{345} & S_1 C_{345} & -C_{45} l_{L3} - C_5 l_{L4} \\ -C_1 S_2 & -S_1 S_2 & C_2 & 0 \\ C_1 C_2 S_{345} + S_1 C_{345} & S_1 C_2 S_{345} - C_1 C_{345} & S_2 S_{345} & -S_{45} l_{L3} - S_5 l_{L4} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

We can compare now elements (2,3) of LHS and RHS of G_2 to obtain the joint solutions for θ_2 .

$$\theta_2 = \text{atan2} \left(\pm \sqrt{1 - (S_6 a'_x + C_6 a'_y)^2}, S_6 a'_x + C_6 a'_y \right) \quad (41)$$

Same happens for θ_1 . We take elements (2,1) and (2,2) of LHS and RHS, divide the resulting equations to obtain:

$$\theta_1 = \text{atan2}(-S_6 s'_x - C_6 s'_y, -S_6 n'_x - C_6 n'_y) \quad (42)$$

$$\text{if } S_2 < 0, \text{ then } \theta_1 = \theta_1 + \pi$$

Comparing elements (1,3) and (3,3) of LHS and RHS, we get two equations in S_{345} and C_{345} and dividing them we find the equation for θ_{345} .

$$\theta_{345} = \text{atan2}(a'_z, C_6 a'_x - S_6 a'_y) \quad (43)$$

$$\text{if } S_2 < 0, \text{ then } \theta_{345} = \theta_{345} + \pi$$

Finally, from eq (43) we can find the joint solution for θ_3

$$\theta_3 = \theta_{345} - \theta_4 - \theta_5$$

The same procedure applies to the left leg, replacing l_{L1} with $-l_{L1}$.

3.1.3 Simulation code setup

In order to animate our robot with the correct control variables, defined in the previous paragraph, it is necessary to write a code able to compute all the calculations required. We will give a fast explanation of the steps that were followed in writing the code^[11].

1. Definition of the 3D-LIPM: pendulum height, mass, step length, height.
2. Actuation of the 3D-LIPM: since we want to look at the LIP natural dynamics, there is no actuation but the gravity force.
3. Waypoints saving: in proper variables, feet (base of the LIP) and CoG positions are saved as waypoints.
4. Trajectory reconstruction: the waypoints are connected through a cubic polynomial trajectory (using the Robotics toolbox function “polytraj”).
5. Symmetrical trajectories definition: same trajectories are set for both feet, but with an alternative repetition (alternation between stance and swing foot).
6. Inverse kinematics: calculation of the joints control variable with the inverse kinematics approach by dr. Ali.
7. Initial manoeuvre: moving the CoG above one foot, lowering the body through knees’ bending, getting into the initial condition in which the CoG of the robot and the point mass of the 3D-LIPM coincide.
8. First half-step: while one foot is standing, the other one starts its swing, with given velocity, following the trajectory defined above, until it touches the ground with the same starting velocity. Note that the foot is set to always be parallel to the ground.
9. Switching foot: in the same instant the swinging foot touches the ground, it becomes the LIP’s base, and the previously standing foot starts its swing with the same exact properties of the other. Note that there is no double stance phase in order to simplify calculations.

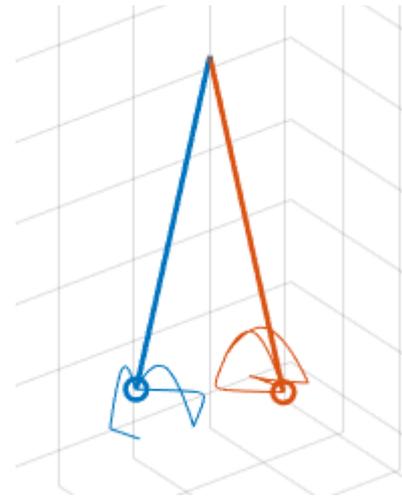


Figure 3.3 LIPM animation (left) and feet trajectories (right)

3.1.4 Simscape Multibody simulation

In this paragraph we will show the different gait-cycle's phases put in act by our robot and the resultant CoG position throughout the simulation itself.

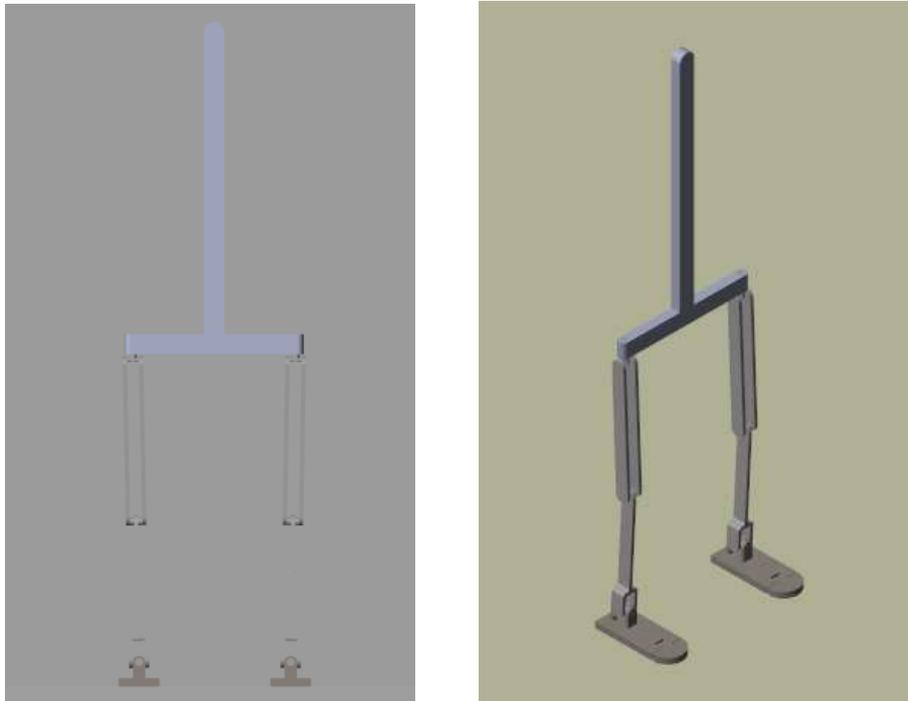


Figure 3.4 Initial stance isometric view (left) posterior view (right)

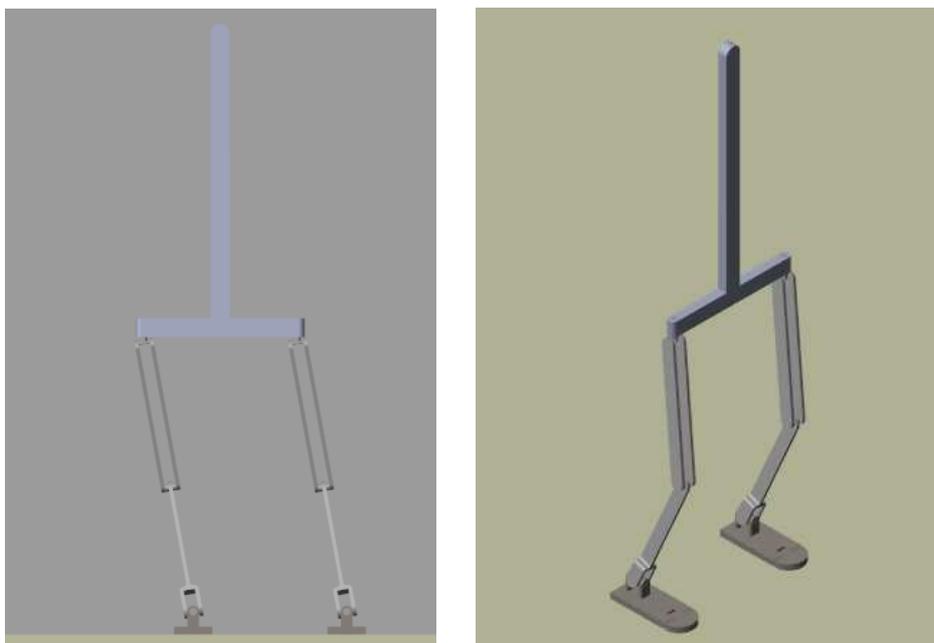


Figure 3.5 Initial manoeuvre stance isometric view (left) posterior view (right)



Figure 3.6 Mid-swing isometric view (left) posterior view (right)

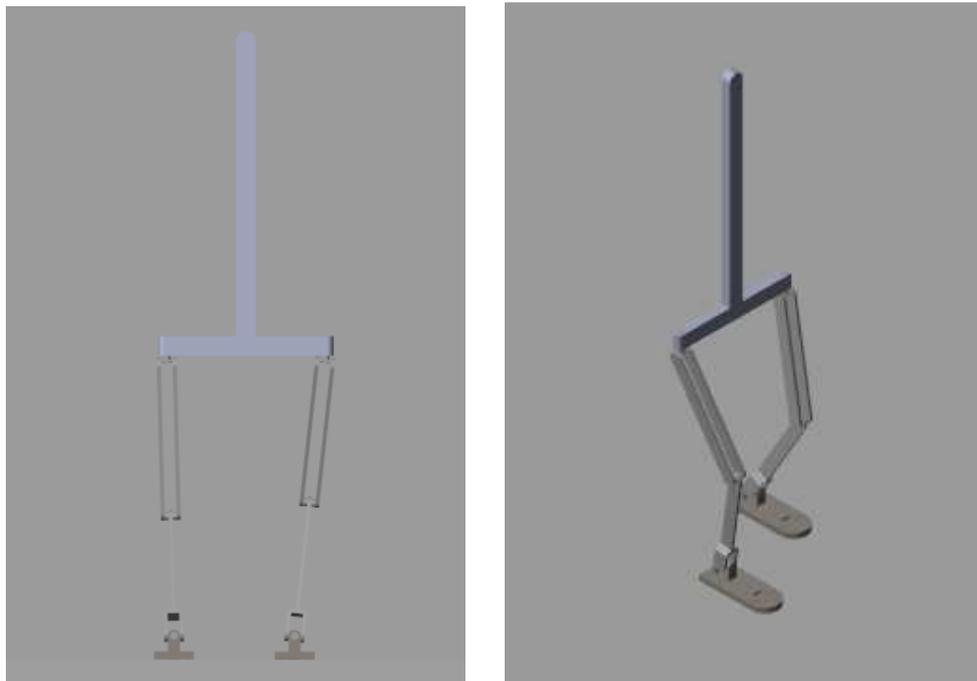


Figure 3.7 Foot switch isometric view (left) posterior view (right)

We can see that the CoG has a sinusoidal behaviour. This happens because the CoP is supposed to always lay within the support polygon defined by the feet. Hence, the result is correct given that each peak in the yellow line corresponds to the foot touching the ground.

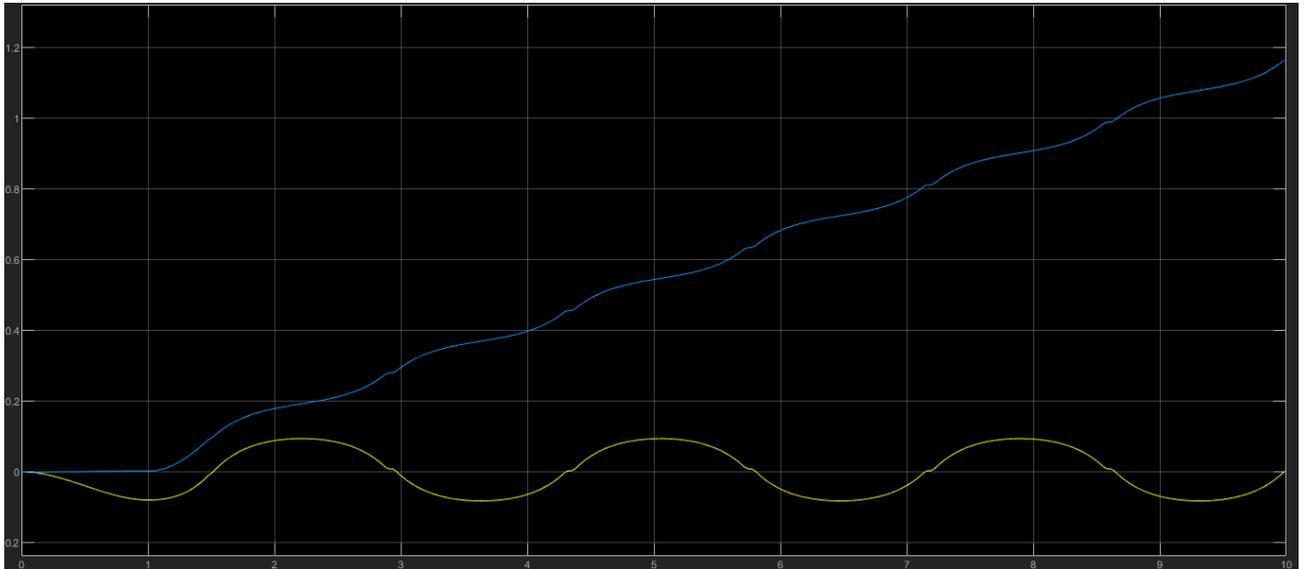


Figure 3.8 CoG behaviour during gait cycle (yellow line along y-axis and blue line along x-axis)

3.2 Simulations with human measurement

In this paragraph we will analyse how the system acts when joints are actuated according to human measurement. We have previously shown how our robot moves when its control variables are calculated through a basic inverse kinematics approach, with no proper control system on the variables and on the postural equilibrium. Here, we will go through the measurement techniques adopted in recording data and, later on, we will show the importance of giving constraints, to make the robot oriented to a future exoskeleton's development. The environment in which we will carry simulations out is, as the previous one, the Simscape Multibody platform^[16].

3.2.1 Measurement setup

Data were recorded at the University of Texas in Dallas [x] on a Bertec instrumented treadmill.

The treadmill is dual belt model and it gives the possibility to record data thanks to the force plate installed in it. The following figure shows the treadmill used in the data recording session.



Figure 3.9 Bertec treadmill

All gait trials were performed in barefoot condition, at natural cadence. The marker-set used is shown in figure 3.4 and it is aimed to the recording of all the angular joints position. In our experiment, we will only use the hip rpy, knee and ankle rp. Before going through the trial, each patient was asked to stand straight for one second, to calibrate the axes of the reference frame with respect to which measurements were recorded.

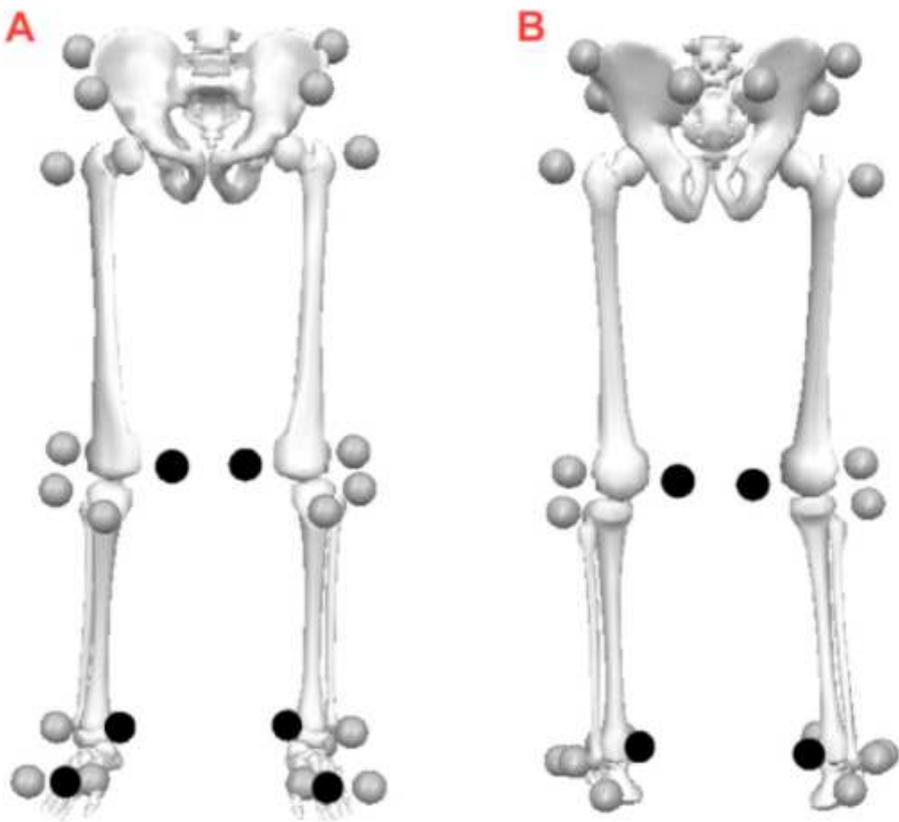


Figure 3.10 marker set used in the recording session (A anterior view, B posterior view)

3.2.2 Simulation with complete dataset

In order to simulate an exoskeleton linked to the roof, to avoid any kind of failure in the self-equilibrium of the patient, we decided to link our robot to the world frame through a prismatic joint, avoiding any oscillation of the robot itself. The aim of this choice is to simulate, with good approximation, the behaviour of an exoskeleton with a proper control system meant for EMG signals recording and online computation of the torque at each joint. In our case, input torque at each joint is automatically computed by the Simscape platform in order to reach the angular position received as input. In addition we will be able to simulate variables such as ZMP, baso signals and CoG position, to find information about the patient's walk.

In this simulation is possible to pick all the gait-cycle's phases, including the double stance one. Let us now go through the simulation and the recognition of gait-cycle's phases. In addition we can notice the importance of the passive toe joint that gives support to the whole body extending the support polygon. In other words, if the foot was modelled as a unique body, during the single stance phase in which there is the revolution around the contact point with the ground, the whole body would have been relying on a single point to support its weight, while with the mobile tip, the point is extended into a surface. Another pro of the toe joint is its adaptability to the difference of each step. This means that regardless the angle the foot spaces with respect to the ground, the tip will always be flat on the floor, sustaining the robot. In the following figure it is possible to see the different phases of the gait cycle performed by the robot, with particular emphasis to the tip.

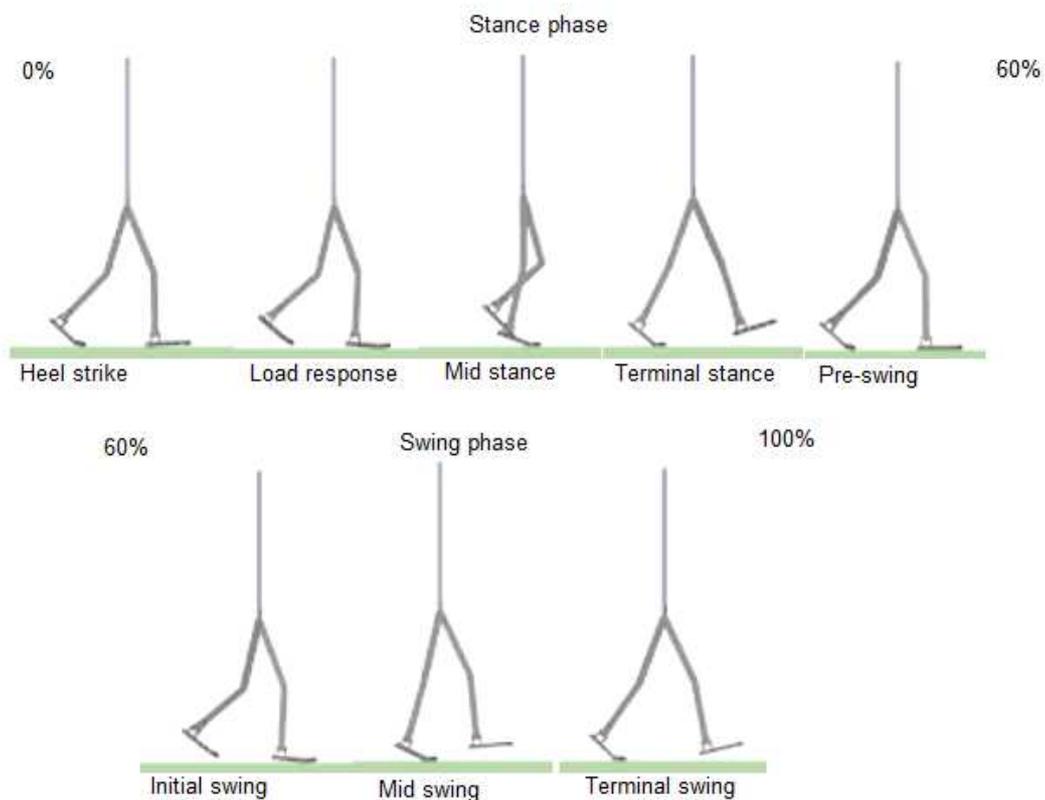


Figure 3.11 Robot's gait cycle

Through the basometric signals, recorded on the robot, we can see that all the four states are carried through, even though they may show different time duration. Thanks to the following graph, it is possible to assume that the patient on whom the measurements were recorded has no significant walking issue and his feet perfectly interact with the ground.

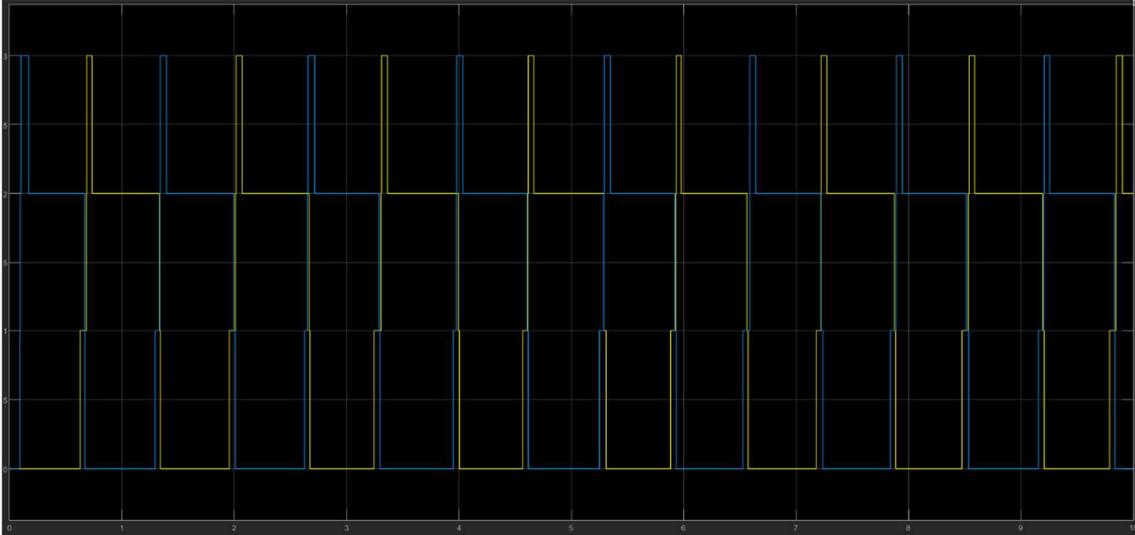


Figure 3.12 Baso signals from simulation (yellow left foot, blue right foot)

Let us now go in-depth of the contact analysis. With the baso signals another important variable to analyse in order to describe the quality of a walking pattern is the Zero Moment Point calculated with the algorithm exposed in chapter 2. In the graph below we can see the different curves of the ZMP, one along the x-axis (yellow) and the other one along the y-axis (blue). We can see that the robot, moved through the signals measured on the patient, effectively progresses forward. The small oscillation at the beginning of the plateau are given by small non-linearities at the heel-strike and they appear also in recorded data on the force plate. Talking about the y-axis, we can see that the same non-linearities recorded along the x-axis appear as well. In addition, it is evident that, while walking, the robot tends to curve to its left, showing a small walking flaw of the patient. This flaw was not evident because during the recording session the patient was forced to walk on a straight line, being him constrained to the handle of the treadmill. So we can deduce that, even though the patient was constrained to walk on a straight line, the resultant of the forces he exchanges with the floor makes him tend to walk towards his left. This is the behaviour we want to avoid when creating an exoskeleton for rehabilitation because we want to constraint the patient to make him accomplish the most accurate postural task, training him to activate the correct muscles depending on the postural task. In this example, we can deduce that the patient shows a hyper-activation of the left ischiocrural muscles, with respect to the right ones. So, by giving some constraints

aimed to limit this hyper-activation we can train the patient to straighten his walk, finding more symmetrical activations.

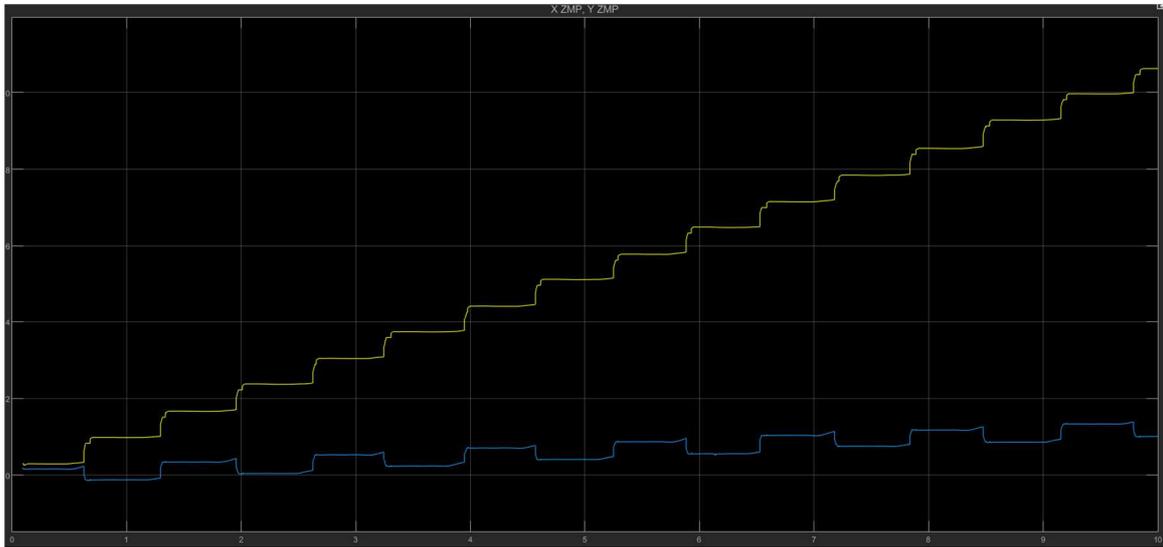


Figure 3.12 Unconstrained ZMP (along y-axis blue, along x-axis yellow)

We can find confirmation of the walking flaw described above in the analysis of the CoG position. In the following graph we can see the CoG moving towards negative value of y –left of the robot. While the x behaviour remains the expected one.

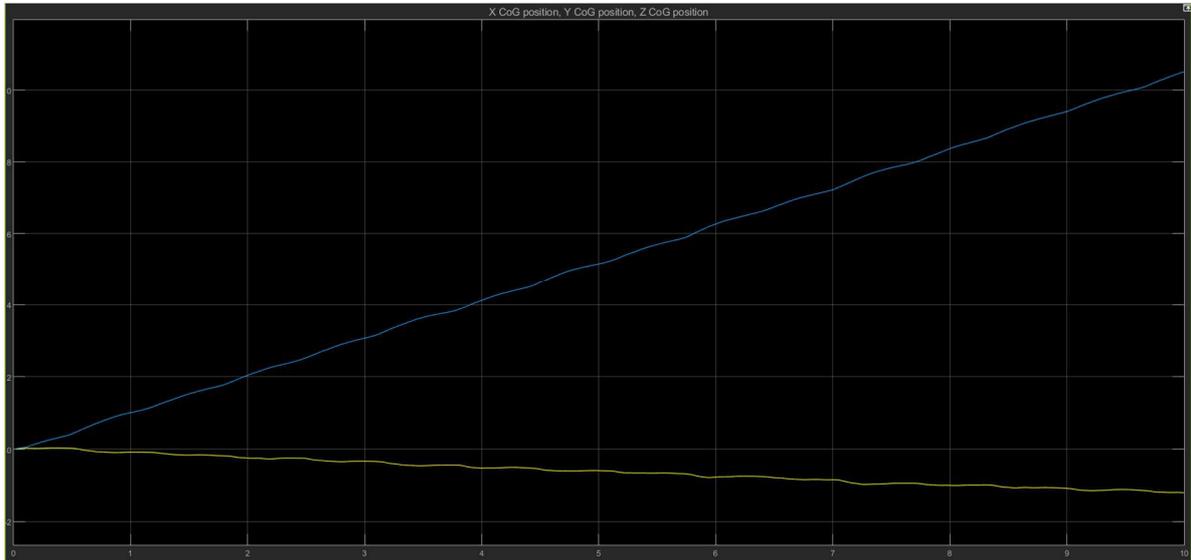


Figure 3.13 Unconstrained CoG (along y-axis blue, along x-axis yellow)

3.2.3 Simulation with complete dataset and joint constraints

In the previous paragraph we have discussed about the simulation carried through with the complete dataset and all of the 14-dof controlled by input variables represented by angular measurement taken on the patient. Here we will propose a solution to the little flaws of the patient's walking pattern. We have already seen that his problem is that, even in walking on a straight line, the resultant forces exchanged with the floor, cause him to slightly turn to his left. The solution adopted in order to make the patient walk properly is the constraining of some of the degrees of freedom the eventual rehabilitation exoskeleton will have. This will force the patient to activate certain muscles in order to symbiotize with the machine and, eventually, after many hours of training, to show a proper walk on a straight line.

The joint variables that will be set to zero are all the roll angles (hip and ankle) and the yaw angle at the hip. This will force the patient to keep his feet and legs straight, avoiding any lateral oscillation of the lower limbs and, in particular, avoiding the generation of a yaw moment through the contact with the ground that is the dynamical parameter that causes turning. The gait cycle remains exactly equal to the previous one, as the baso signals. The differences are evident when looking at the ZMP and CoG position graphs. The ZMP shows a behaviour very similar to the theoretical one, with no oscillations and step width and length constant. The CoG, on the other hand shows no oscillation along the y-axis.

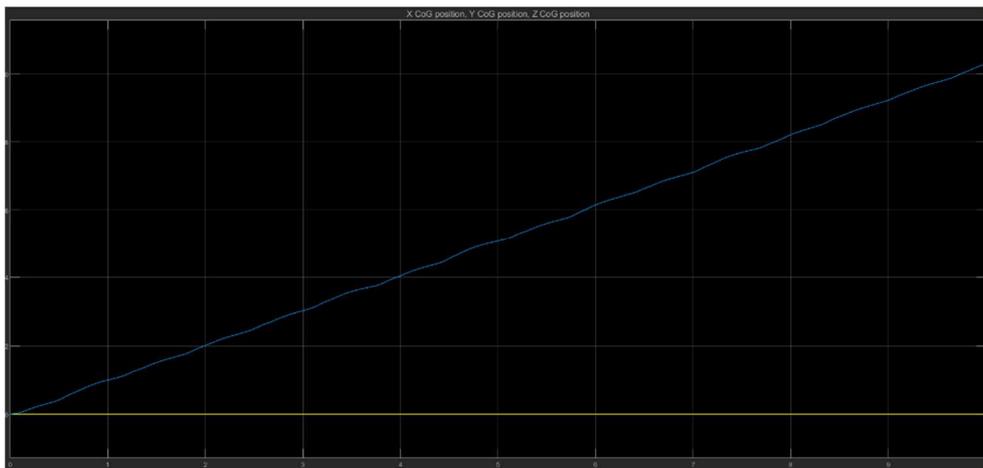


Figure 3.14 Constrained CoG (along y-axis blue, along x-axis yellow)

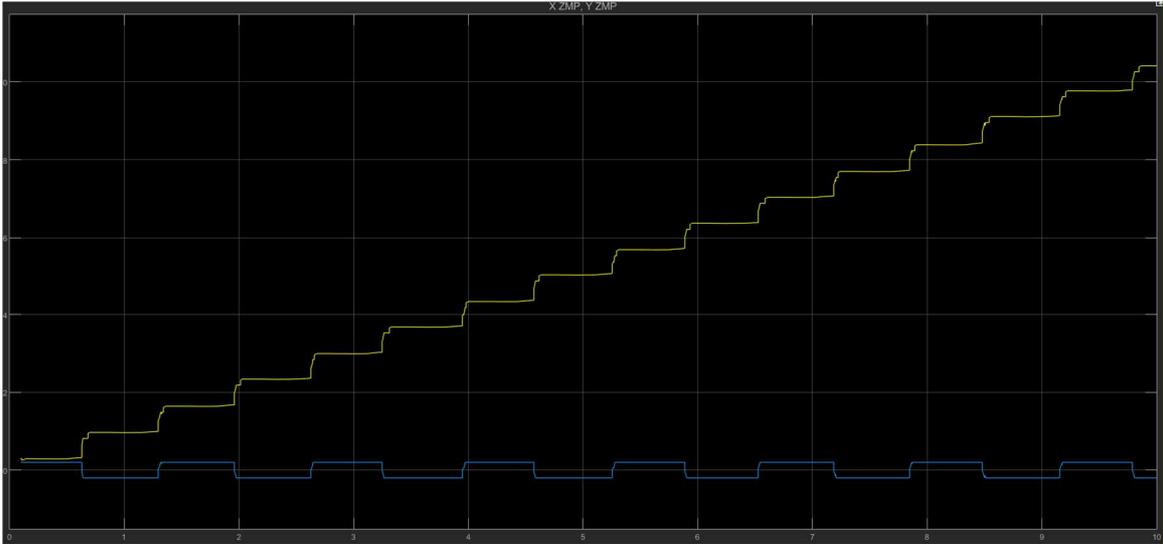


Figure 3.13 Constrained ZMP (along y-axis blue, along x-axis yellow)

The same technique could be adopted for many different cases. Let us assume our patient suffered of a knee injury and needs his tendons to reactivate, with quads and gastrocnemius. By giving the roll and yaw constrains, we force the patient to activate only the large muscular groups in order to accomplish the complex postural task which is walking. At the same time the exoskeleton could give support to the injured leg, by accepting a part of the bodyweight and maintaining the postural equilibrium.

4 Conclusion and future development

An exoskeleton may be built for two main reasons which are empowering of human capabilities, and rehabilitation of patients affected by different pathologies. In this thesis project we have discussed the modelling of an exoskeleton oriented to rehabilitation. Let us begin giving the assumption that regardless the complexity of a robot, it will not be able, at the moment, to show all the degrees of freedom our body has. So, a rehabilitation exoskeleton has to be modelled on the patient and for the patient, to safeguard his/her health. The first step to make is surely the recognition of the patient's pathology, comparing recorded data, if present, with a standard database holding data of a large variety of subjects. After that, it will be possible to give emphasis to the certain postural task the patient struggles to carry out. In this thesis, after having analysed the human behaviour during walking, we have presented the first steps necessary to build a robot. In order to make our robot behave as a proper exoskeleton, future improvements are needed. First of all, there is the need of a complex deep-learning tool, able to measure the EMG signals while the patient is moving, translate 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.

Secondly, another addition to the simple project we have presented can be a control system for postural equilibrium, working together with the neural net, to correct input torques at each joint. The idea will be to measure the acceleration of the CoG, compare it to a given threshold and, if it is exceeded, compensate joints' torque. This system could be improved inserting a prevision horizon, to have a preview of the future state of the robot and correct in advance the control variables.

Thirdly, CAD modelling has to be adapted according to the patient, to follow his/her shapes and capabilities.

In conclusion, this work was merely carried out to move the first steps into the exoskeletons' world, following some childish dream through the mean of knowledge. The results obtained are a good starting point for future studies, that can be improved through on-field work together with a strong-willed team, following the same objective of giving new life to some people that, unfortunately do not have the chance to live theirs as they would and could. Lastly, as engineers, we have to keep in mind that our role in the world

is to put in act ideas, regardless the person giving birth to them, to make our planet a better place to live in, without leaving anybody behind.

5 Appendices

5.1 Anatomy recalls

Ankle: trochlea joint between the tibia, fibula and talus it only allows movements of flexion and extension of the foot

Flexion of the Ankle (Dorsiflexion or Dorsal-extension): movement which brings the back of the foot closer to the anterior part of the leg. The breadth of movement is 20° - 30° .

Ankle Extension (Plantar Flexion): movement that moves away the back of the foot from the anterior aspect of the leg, while the foot stretches to arrange themselves in the extension of the leg. The range of motion is of 30° - 50° .

Hamstring muscles: muscle group (hamstring, semitendinosus, and semimembranosus muscle) which are part of the posterior part of the thigh, and which perform the common action of hitting the leg on the thigh and extend the thigh to the hip.

Quads: parts of the quadriceps femoris muscle, which with its action extends the leg on the thigh and puts the thigh on the pelvis.

Gracilis muscle: adductor muscle of the thigh. It is a bi-articular muscle, that is, which acts on two joints, the only one among the adductor muscles to have this particularity.

The Sartorius muscle is a long ribbon-like muscle located in the part front of the thigh, and is in fact the longest muscle in the human body. It acts both at the level of the knee, allowing internal rotation of the tibia and leg flexion, both at the hip, allowing the thigh on the pelvis and external rotation.

Soleus: originates on the head of the tibia, and is inserted as a heel tendon (Achilles tendon). With its action it plantarily affects the joint of the ankle.

Gastrocnemius: The two heads join together and fit into the tuberosity calcaneal with the tendon of the soleus muscle. With its action flexes plantarily the ankle joint and collaborates in the flexion of the knee.

5.2 Robot's XML code

```
% Simscape(TM) Multibody(TM) version: 7.2

% This is a model data file derived from a Simscape Multibody Import XML file using the smimport
function.

% The data in this file sets the block parameter values in an imported Simscape Multibody model.
% For more information on this file, see the smimport function help page in the Simscape Multibody
documentation.

% You can modify numerical values, but avoid any other changes to this file.
% Do not add code to this file. Do not edit the physical units shown in comments.

%%VariableName:smiData

%===== RigidTransform =====%

%Initialize the RigidTransform structure array by filling in null values.
smiData.RigidTransform(30).translation = [0.0 0.0 0.0];
smiData.RigidTransform(30).angle = 0.0;
smiData.RigidTransform(30).axis = [0.0 0.0 0.0];
smiData.RigidTransform(30).ID = "";

%Translation Method - Cartesian
%Rotation Method - Arbitrary Axis
smiData.RigidTransform(1).translation = [0 0 0]; % m
smiData.RigidTransform(1).angle = 3.1415926535897931; % rad
smiData.RigidTransform(1).axis = [1 0 0];
smiData.RigidTransform(1).ID = 'B[-:MfmKeEzS13xr1hWd5]';

%Translation Method - Cartesian
%Rotation Method - Arbitrary Axis
smiData.RigidTransform(2).translation = [0 -0.014999999999999999 -0.50794981699999997]; % m
smiData.RigidTransform(2).angle = 3.1415926535897931; % rad
smiData.RigidTransform(2).axis = [1 0 0];
smiData.RigidTransform(2).ID = 'F[-:MfmKeEzS13xr1hWd5]';
```

```

%Translation Method - Cartesian
%Rotation Method - Arbitrary Axis
smiData.RigidTransform(3).translation = [0 -0.01 0.085000000000000006]; % m
smiData.RigidTransform(3).angle = 2.0943951023931953; % rad
smiData.RigidTransform(3).axis = [0.57735026918962584 -0.57735026918962584 -
0.57735026918962584];
smiData.RigidTransform(3).ID = 'B[M/HUMrhCayAMK8bxE:-:M+hCgn0+RxotHQWiT]';

```

```

%Translation Method - Cartesian
%Rotation Method - Arbitrary Axis
smiData.RigidTransform(4).translation = [0 -0.01 -0.036865652000000003];%-
0.056865652000000003]; % m
smiData.RigidTransform(4).angle = 2.0943951023931953; % rad
smiData.RigidTransform(4).axis = [0.57735026918962584 0.57735026918962584
0.57735026918962584];
smiData.RigidTransform(4).ID = 'F[M/HUMrhCayAMK8bxE:-:M+hCgn0+RxotHQWiT]';

```

```

%Translation Method - Cartesian
%Rotation Method - Arbitrary Axis
smiData.RigidTransform(5).translation = [0 0 0]; % m
smiData.RigidTransform(5).angle = 3.1415926535897931; % rad
smiData.RigidTransform(5).axis = [0 0.70710678118654746 0.70710678118654757];
smiData.RigidTransform(5).ID = 'B[MJU3k80US1zHsLRky:-:M/HUMrhCayAMK8bxE]';

```

```

%Translation Method - Cartesian
%Rotation Method - Arbitrary Axis
smiData.RigidTransform(6).translation = [-0 -0.05999999999999998 -0.080000000000000002]; % m
smiData.RigidTransform(6).angle = 3.1415926535897931; % rad
smiData.RigidTransform(6).axis = [1 0 -0];
smiData.RigidTransform(6).ID = 'F[MJU3k80US1zHsLRky:-:M/HUMrhCayAMK8bxE]';

```

```

%Translation Method - Cartesian
%Rotation Method - Arbitrary Axis
smiData.RigidTransform(7).translation = [0 -4.479999999999999e-07 0.0050000000000000001]; % m
smiData.RigidTransform(7).angle = 1.5707963267948968; % rad

```

```

smiData.RigidTransform(7).axis = [-1 -0 -0];
smiData.RigidTransform(7).ID = 'B[M1rvh1rDGAc6aYvT7:-:MAehd/LFx/ZxMmv8F]';
%Translation Method - Cartesian
%Rotation Method - Arbitrary Axis
smiData.RigidTransform(8).translation = [0 -0.01 0]; % m
smiData.RigidTransform(8).angle = 1.5707963267948968; % rad
smiData.RigidTransform(8).axis = [-1 -0 -0];
smiData.RigidTransform(8).ID = 'F[M1rvh1rDGAc6aYvT7:-:MAehd/LFx/ZxMmv8F]';

%Translation Method - Cartesian
%Rotation Method - Arbitrary Axis
smiData.RigidTransform(9).translation = [0.20000000000000001 -0.014999999999999999 -
0.45794981699999998]; % m
smiData.RigidTransform(9).angle = 3.1415926535897931; % rad
smiData.RigidTransform(9).axis = [1 0 0];
smiData.RigidTransform(9).ID = 'B[MfmKeEzS13xr1hWd5:-:M1rvh1rDGAc6aYvT7]';

%Translation Method - Cartesian
%Rotation Method - Arbitrary Axis
smiData.RigidTransform(10).translation = [0 0 0.059999999999999998]; % m
smiData.RigidTransform(10).angle = 3.1415926535897931; % rad
smiData.RigidTransform(10).axis = [1 0 0];
smiData.RigidTransform(10).ID = 'F[MfmKeEzS13xr1hWd5:-:M1rvh1rDGAc6aYvT7]';

%Translation Method - Cartesian
%Rotation Method - Arbitrary Axis
smiData.RigidTransform(11).translation = [0 -0.029999999999999999 -0.17499999999999999]; % m
smiData.RigidTransform(11).angle = 3.1415926535897931; % rad
smiData.RigidTransform(11).axis = [0 -0.70710678118654746 0.70710678118654757];
smiData.RigidTransform(11).ID = 'B[M6n692nmwUBZGWvfH:-:MJU3k80US1zHsLRky]';
%Translation Method - Cartesian
%Rotation Method - Arbitrary Axis
smiData.RigidTransform(12).translation = [-0.025000000000000001 -0.029999999999999999 0]; % m
smiData.RigidTransform(12).angle = 2.0943951023931953; % rad
smiData.RigidTransform(12).axis = [-0.57735026918962584 -0.57735026918962584
0.57735026918962584];
smiData.RigidTransform(12).ID = 'F[M6n692nmwUBZGWvfH:-:MJU3k80US1zHsLRky]';

```

%Translation Method - Cartesian

%Rotation Method - Arbitrary Axis

smiData.RigidTransform(13).translation = [0 0 -0.20000000000000001]; % m

smiData.RigidTransform(13).angle = 3.1415926535897931; % rad

smiData.RigidTransform(13).axis = [-0 0.70710678118654746 0.70710678118654757];

smiData.RigidTransform(13).ID = 'B[MrTdipo7NpBGdEaYl:-:M6n692nmwUBZGWvfH]';

%Translation Method - Cartesian

%Rotation Method - Arbitrary Axis

smiData.RigidTransform(14).translation = [0 -0.0050000000000000001 0.17499999999999999]; % m

smiData.RigidTransform(14).angle = 1.5707963267948968; % rad

smiData.RigidTransform(14).axis = [1 0 0];

smiData.RigidTransform(14).ID = 'F[MrTdipo7NpBGdEaYl:-:M6n692nmwUBZGWvfH]';

%Translation Method - Cartesian

%Rotation Method - Arbitrary Axis

smiData.RigidTransform(15).translation = [-0.20000000000000001 -0.01499999999999999 -0.45794981699999998]; % m

smiData.RigidTransform(15).angle = 3.1415926535897931; % rad

smiData.RigidTransform(15).axis = [1 0 0];

smiData.RigidTransform(15).ID = 'B[MfmKeEzS13xr1hWd5:-:M8u/DfEoU5o42gr9m]';

%Translation Method - Cartesian

%Rotation Method - Arbitrary Axis

smiData.RigidTransform(16).translation = [0 0 0.059999999999999998]; % m

smiData.RigidTransform(16).angle = 3.1415926535897931; % rad

smiData.RigidTransform(16).axis = [1 0 0];

smiData.RigidTransform(16).ID = 'F[MfmKeEzS13xr1hWd5:-:M8u/DfEoU5o42gr9m]';

%Translation Method - Cartesian

%Rotation Method - Arbitrary Axis

smiData.RigidTransform(17).translation = [0 -4.479999999999999e-07 0.0050000000000000001]; % m

smiData.RigidTransform(17).angle = 1.5707963267948968; % rad

smiData.RigidTransform(17).axis = [-1 -0 -0];

smiData.RigidTransform(17).ID = 'B[M8u/DfEoU5o42gr9m:-:Mn1nOcwC08SrJSuUX]';

%Translation Method - Cartesian

%Rotation Method - Arbitrary Axis

smiData.RigidTransform(18).translation = [0 -0.01 0]; % m

smiData.RigidTransform(18).angle = 1.5707963267948968; % rad

smiData.RigidTransform(18).axis = [-1 -0 -0];

smiData.RigidTransform(18).ID = 'F[M8u/DfEoU5o42gr9m:-:MnInOcwC08SrJSuUX]';

%Translation Method - Cartesian

%Rotation Method - Arbitrary Axis

smiData.RigidTransform(19).translation = [-0.025000000000000001 -0.01 -0.01]; % m

smiData.RigidTransform(19).angle = 2.0943951023931953; % rad

smiData.RigidTransform(19).axis = [0.57735026918962584 -0.57735026918962584 -
0.57735026918962584];

smiData.RigidTransform(19).ID = 'B[MAehd/LFx/ZxMmv8F:-:MKxmLYjmpQJjOE8LP]';

%Translation Method - Cartesian

%Rotation Method - Arbitrary Axis

smiData.RigidTransform(20).translation = [0 -0.025000000000000001 0.20000000000000001]; % m

smiData.RigidTransform(20).angle = 1.5707963267948968; % rad

smiData.RigidTransform(20).axis = [1 0 0];

smiData.RigidTransform(20).ID = 'F[MAehd/LFx/ZxMmv8F:-:MKxmLYjmpQJjOE8LP]';

%Translation Method - Cartesian

%Rotation Method - Arbitrary Axis

smiData.RigidTransform(21).translation = [0 0 0]; % m

smiData.RigidTransform(21).angle = 3.1415926535897931; % rad

smiData.RigidTransform(21).axis = [0 0.70710678118654746 0.70710678118654757];

smiData.RigidTransform(21).ID = 'B[MKHV3mBwZGVI+NY88:-:MiRbW1IwB53fv34fK]';

%Translation Method - Cartesian

%Rotation Method - Arbitrary Axis

smiData.RigidTransform(22).translation = [-0 -0.059999999999999998 -0.080000000000000002]; % m

smiData.RigidTransform(22).angle = 3.1415926535897931; % rad

smiData.RigidTransform(22).axis = [1 0 -0];

smiData.RigidTransform(22).ID = 'F[MKHV3mBwZGVI+NY88:-:MiRbW1IwB53fv34fK]';

%Translation Method - Cartesian

%Rotation Method - Arbitrary Axis

```
smiData.RigidTransform(23).translation = [0 0.02 -0.17499999999999999]; % m
smiData.RigidTransform(23).angle = 3.1415926535897931; % rad
smiData.RigidTransform(23).axis = [0 -0.70710678118654746 0.70710678118654757];
smiData.RigidTransform(23).ID = 'B[MIANUfg2UPIZqXCU8:-:MKHV3mBwZGVI+NY88]';
```

%Translation Method - Cartesian

%Rotation Method - Arbitrary Axis

```
smiData.RigidTransform(24).translation = [0.025000000000000001 -0.02999999999999999 0]; % m
smiData.RigidTransform(24).angle = 2.0943951023931953; % rad
smiData.RigidTransform(24).axis = [-0.57735026918962584 -0.57735026918962584
0.57735026918962584];
smiData.RigidTransform(24).ID = 'F[MIANUfg2UPIZqXCU8:-:MKHV3mBwZGVI+NY88]';
```

%Translation Method - Cartesian

%Rotation Method - Arbitrary Axis

```
smiData.RigidTransform(25).translation = [0 0 -0.20000000000000001]; % m
smiData.RigidTransform(25).angle = 1.5707963267948968; % rad
smiData.RigidTransform(25).axis = [1 0 0];
smiData.RigidTransform(25).ID = 'B[MKxmLYjmpQJjOE8LP:-:MIANUfg2UPIZqXCU8]';
```

%Translation Method - Cartesian

%Rotation Method - Arbitrary Axis

```
smiData.RigidTransform(26).translation = [0 -0.0050000000000000001 0.17499999999999999]; % m
smiData.RigidTransform(26).angle = 1.5707963267948968; % rad
smiData.RigidTransform(26).axis = [1 0 0];
smiData.RigidTransform(26).ID = 'F[MKxmLYjmpQJjOE8LP:-:MIANUfg2UPIZqXCU8]';
```

%Translation Method - Cartesian

%Rotation Method - Arbitrary Axis

```
smiData.RigidTransform(27).translation = [0 -0.01 0.085000000000000006]; % m
smiData.RigidTransform(27).angle = 2.0943951023931953; % rad
smiData.RigidTransform(27).axis = [0.57735026918962584 -0.57735026918962584 -
0.57735026918962584];
smiData.RigidTransform(27).ID = 'B[MiRbW1IwB53fv34fK:-:MoyKIxtsieEkdXtfj]';
```

%Translation Method - Cartesian

%Rotation Method - Arbitrary Axis

smiData.RigidTransform(28).translation = [0 -0.01 -0.036865652000000003]; % m

smiData.RigidTransform(28).angle = 2.0943951023931953; % rad

smiData.RigidTransform(28).axis = [0.57735026918962584 0.57735026918962584
0.57735026918962584];

smiData.RigidTransform(28).ID = 'F[MiRbW1IwB53fv34fK:-:MoyKlxtsieEkdXtfj]';

%Translation Method - Cartesian

%Rotation Method - Arbitrary Axis

smiData.RigidTransform(29).translation = [-0.025000000000000001 -0.01 -0.01]; % m

smiData.RigidTransform(29).angle = 2.0943951023931953; % rad

smiData.RigidTransform(29).axis = [-0.57735026918962584 -0.57735026918962584
0.57735026918962584];

smiData.RigidTransform(29).ID = 'B[MnInOcwC08SrJSuUX:-:MrTdipo7NpBGdEaYI]';

%Translation Method - Cartesian

%Rotation Method - Arbitrary Axis

smiData.RigidTransform(30).translation = [0 0.025000000000000001 0.20000000000000001]; % m

smiData.RigidTransform(30).angle = 1.5707963267948968; % rad

smiData.RigidTransform(30).axis = [-1 -0 -0];

smiData.RigidTransform(30).ID = 'F[MnInOcwC08SrJSuUX:-:MrTdipo7NpBGdEaYI]';

```
%===== Solid =====%
%Center of Mass (CoM) %Moments of Inertia (MoI) %Product of Inertia (PoI)
```

```
%Initialize the Solid structure array by filling in null values.
```

```
smiData.Solid(15).mass = 0.0;
smiData.Solid(15).CoM = [0.0 0.0 0.0];
smiData.Solid(15).MoI = [0.0 0.0 0.0];
smiData.Solid(15).PoI = [0.0 0.0 0.0];
smiData.Solid(15).color = [0.0 0.0 0.0];
smiData.Solid(15).opacity = 0.0;
smiData.Solid(15).ID = '';
```

```
%Inertia Type - Custom
```

```
%Visual Properties - Simple
```

```
smiData.Solid(1).mass = 0.71140341900000004; % kg
smiData.Solid(1).CoM = [-0 -0.0099990119999999998 -0.024246253999999998]; % m
smiData.Solid(1).MoI = [0.00023778699999999999 0.000605999999999999998
0.000416543000000000001]; % kg*m^2
smiData.Solid(1).PoI = [-2.0000000000000001e-09 0 0]; % kg*m^2
smiData.Solid(1).color = [0.650980392 0.619607843 0.588235294];
smiData.Solid(1).opacity = 1.000000000;
smiData.Solid(1).ID = 'JFb*:*4aa1748ffbc6c6df020f2393';
```

```
%Inertia Type - Custom
```

```
%Visual Properties - Simple
```

```
smiData.Solid(2).mass = 3.20589605000000002; % kg
smiData.Solid(2).CoM = [-0 -0.011969459 -0.0065391729999999997]; % m
smiData.Solid(2).MoI = [0.01027812 0.012541817 0.002931225]; % kg*m^2
smiData.Solid(2).PoI = [-0.000274380999999999997 0 0]; % kg*m^2
smiData.Solid(2).color = [0.650980392 0.619607843 0.588235294];
smiData.Solid(2).opacity = 1.000000000;
smiData.Solid(2).ID = 'JFL*:*4aa1748ffbc6c6df020f2393';
```

%Inertia Type - Custom

%Visual Properties - Simple

```
smiData.Solid(3).mass = 1.4241845099999999; % kg
smiData.Solid(3).CoM = [0 -0.0050000000000000001 -0.029235403]; % m
smiData.Solid(3).MoI = [0.016682611999999999 0.016693268000000001
0.00037529099999999998]; % kg*m^2
smiData.Solid(3).PoI = [0 0 0]; % kg*m^2
smiData.Solid(3).color = [0.752941176 0.752941176 0.752941176];
smiData.Solid(3).opacity = 1.000000000;
smiData.Solid(3).ID = 'JFv*:*4aa1748ffbc6c6df020f2393';
```

%Inertia Type - Custom

%Visual Properties - Simple

```
smiData.Solid(4).mass = 1.4241845099999999; % kg
smiData.Solid(4).CoM = [0 -0.0050000000000000001 -0.029235403]; % m
smiData.Solid(4).MoI = [0.016682611999999999 0.016693268000000001
0.00037529099999999998]; % kg*m^2
smiData.Solid(4).PoI = [0 0 0]; % kg*m^2
smiData.Solid(4).color = [0.752941176 0.752941176 0.752941176];
smiData.Solid(4).opacity = 1.000000000;
smiData.Solid(4).ID = 'JFf*:*4aa1748ffbc6c6df020f2393';
```

%Inertia Type - Custom

%Visual Properties - Simple

```
smiData.Solid(5).mass = 0.22940936200000001; % kg
smiData.Solid(5).CoM = [-1.0999999999999999e-08 -0.029999984 0]; % m
smiData.Solid(5).MoI = [5.2302000000000002e-05 3.4217000000000001e-05 7.4628000000000004e-
05]; % kg*m^2
smiData.Solid(5).PoI = [0 0 0]; % kg*m^2
smiData.Solid(5).color = [0.752941176 0.752941176 0.752941176];
smiData.Solid(5).opacity = 1.000000000;
smiData.Solid(5).ID = 'JFj*:*4aa1748ffbc6c6df020f2393';
```

%Inertia Type - Custom

%Visual Properties - Simple

```
smiData.Solid(6).mass = 80;%13.9210285; % kg
smiData.Solid(6).CoM = [0 -0.014999999999999999 -0.216149533]; % m
smiData.Solid(6).MoI = [0.98080007199999997 1.051782065 0.073084404000000006]; % kg*m^2
smiData.Solid(6).PoI = [0 -0 0]; % kg*m^2
smiData.Solid(6).color = [0.792156863 0.819607843 0.933333333];
smiData.Solid(6).opacity = 1.000000000;
smiData.Solid(6).ID = 'JFD*:*4aa1748ffbc6c6df020f2393';
```

%Inertia Type - Custom

%Visual Properties - Simple

```
smiData.Solid(7).mass = 0.14024099100000001; % kg
smiData.Solid(7).CoM = [-8.9999999999999995e-09 9.5999999999999999e-08
0.031377897000000002]; % m
smiData.Solid(7).MoI = [4.2713999999999998e-05 4.2930999999999999e-05 7.1210000000000001e-
06]; % kg*m^2
smiData.Solid(7).PoI = [-0 0 0]; % kg*m^2
smiData.Solid(7).color = [0.752941176 0.752941176 0.752941176];
smiData.Solid(7).opacity = 1.000000000;
smiData.Solid(7).ID = 'JFT*:*4aa1748ffbc6c6df020f2393';
```

%Inertia Type - Custom

%Visual Properties - Simple

```
smiData.Solid(8).mass = 0.026891558199999999; % kg
smiData.Solid(8).CoM = [5.8000000000000003e-08 -0.010000019000000001 -
0.0068568140000000001]; % m
smiData.Solid(8).MoI = [1.381e-06 3.027e-06 3.1049999999999999e-06]; % kg*m^2
smiData.Solid(8).PoI = [0 -0 -0]; % kg*m^2
smiData.Solid(8).color = [0.752941176 0.752941176 0.752941176];
smiData.Solid(8).opacity = 1.000000000;
smiData.Solid(8).ID = 'JFn*:*4aa1748ffbc6c6df020f2393';
```

%Inertia Type - Custom

%Visual Properties - Simple

smiData.Solid(9).mass = 2.6532374600000002; % kg

smiData.Solid(9).CoM = [0 0 -0.0059115390000000004]; % m

smiData.Solid(9).MoI = [0.040244371000000001 0.039525382999999997 0.0013984990000000001]; % kg*m^2

smiData.Solid(9).PoI = [0 0 0]; % kg*m^2

smiData.Solid(9).color = [0.752941176 0.752941176 0.752941176];

smiData.Solid(9).opacity = 1.000000000;

smiData.Solid(9).ID = 'JFz*:4aa1748ffbc6c6df020f2393';

%Inertia Type - Custom

%Visual Properties - Simple

smiData.Solid(10).mass = 0.026891558199999999; % kg

smiData.Solid(10).CoM = [5.8000000000000003e-08 -0.010000019000000001 -0.0068568140000000001]; % m

smiData.Solid(10).MoI = [1.381e-06 3.027e-06 3.1049999999999999e-06]; % kg*m^2

smiData.Solid(10).PoI = [0 -0 -0]; % kg*m^2

smiData.Solid(10).color = [0.752941176 0.752941176 0.752941176];

smiData.Solid(10).opacity = 1.000000000;

smiData.Solid(10).ID = 'JFP*:4aa1748ffbc6c6df020f2393';

%Inertia Type - Custom

%Visual Properties - Simple

smiData.Solid(11).mass = 0.14024099100000001; % kg

smiData.Solid(11).CoM = [-8.999999999999995e-09 9.599999999999999e-08 0.031377897000000002]; % m

smiData.Solid(11).MoI = [4.2713999999999998e-05 4.2930999999999999e-05 7.1210000000000001e-06]; % kg*m^2

smiData.Solid(11).PoI = [-0 0 0]; % kg*m^2

smiData.Solid(11).color = [0.752941176 0.752941176 0.752941176];

smiData.Solid(11).opacity = 1.000000000;

smiData.Solid(11).ID = 'JFr*:4aa1748ffbc6c6df020f2393';

%Inertia Type - Custom

%Visual Properties - Simple

```
smiData.Solid(12).mass = 10;%0.71140341900000004; % kg
smiData.Solid(12).CoM = [-0 -0.0099990119999999998 -0.024246253999999998]; % m
smiData.Solid(12).MoI = [0.0002377869999999999 0.0006059999999999998
0.00041654300000000001]; % kg*m^2
smiData.Solid(12).PoI = [-2.0000000000000001e-09 0 0]; % kg*m^2
smiData.Solid(12).color = [0.650980392 0.619607843 0.588235294];
smiData.Solid(12).opacity = 1.000000000;
smiData.Solid(12).ID = 'JF7*:*4aa1748ffbc6c6df020f2393';
```

%Inertia Type - Custom

%Visual Properties - Simple

```
smiData.Solid(13).mass = 2.6532374600000002; % kg
smiData.Solid(13).CoM = [0 -0 -0.0059115390000000004]; % m
smiData.Solid(13).MoI = [0.040244371000000001 0.039525382999999997
0.0013984990000000001]; % kg*m^2
smiData.Solid(13).PoI = [0 0 0]; % kg*m^2
smiData.Solid(13).color = [0.752941176 0.752941176 0.752941176];
smiData.Solid(13).opacity = 1.000000000;
smiData.Solid(13).ID = 'JFX*:*4aa1748ffbc6c6df020f2393';
```

%Inertia Type - Custom

%Visual Properties - Simple

```
smiData.Solid(14).mass = 0.22940936200000001; % kg
smiData.Solid(14).CoM = [-1.0999999999999999e-08 -0.029999984 0]; % m
smiData.Solid(14).MoI = [5.2302000000000002e-05 3.4217000000000001e-05 7.4628000000000004e-
05]; % kg*m^2
smiData.Solid(14).PoI = [0 0 0]; % kg*m^2
smiData.Solid(14).color = [0.752941176 0.752941176 0.752941176];
smiData.Solid(14).opacity = 1.000000000;
smiData.Solid(14).ID = 'JFH*:*4aa1748ffbc6c6df020f2393';
```

%Inertia Type - Custom

%Visual Properties - Simple

smiData.Solid(15).mass = 10;%3.2058960500000002; % kg

smiData.Solid(15).CoM = [-0 -0.011969459 -0.0065391729999999997]; % m

smiData.Solid(15).MoI = [0.01027812 0.012541817 0.002931225]; % kg*m^2

smiData.Solid(15).PoI = [-0.00027438099999999997 0 0]; % kg*m^2

smiData.Solid(15).color = [0.650980392 0.619607843 0.588235294];

smiData.Solid(15).opacity = 1.000000000;

smiData.Solid(15).ID = 'JF3*:*4aa1748ffbc6c6df020f2393';

```
%===== Joint =====%
%X Revolute Primitive (Rx) %Y Revolute Primitive (Ry) %Z Revolute Primitive (Rz)
%X Prismatic Primitive (Px) %Y Prismatic Primitive (Py) %Z Prismatic Primitive (Pz) %Spherical
Primitive (S)
%Constant Velocity Primitive (CV) %Lead Screw Primitive (LS)
%Position Target (Pos)
```

```
%Initialize the RevoluteJoint structure array by filling in null values.
```

```
smiData.RevoluteJoint(14).Rz.Pos = 0.0;
```

```
smiData.RevoluteJoint(14).ID = "";
```

```
smiData.RevoluteJoint(1).Rz.Pos = 1.2776660921661053e-14; % deg
```

```
smiData.RevoluteJoint(1).ID = '[M/HUMrhCayAMK8bxE:-:M+hCgn0+RxotHQWiT]';
```

```
smiData.RevoluteJoint(2).Rz.Pos = -5.7295778624998042e-07; % deg
```

```
smiData.RevoluteJoint(2).ID = '[MJU3k80US1zHsLRky:-:M/HUMrhCayAMK8bxE]';
```

```
smiData.RevoluteJoint(3).Rz.Pos = 0; % deg
```

```
smiData.RevoluteJoint(3).ID = '[M1rvh1rDGAc6aYvT7:-:MAchd/LFx/ZxMmv8F]';
```

```
smiData.RevoluteJoint(4).Rz.Pos = 0; % deg
```

```
smiData.RevoluteJoint(4).ID = '[MfmKeEzS13xr1hWd5:-:M1rvh1rDGAc6aYvT7]';
```

```
smiData.RevoluteJoint(5).Rz.Pos = -1.8334649434535793e-06; % deg
```

```
smiData.RevoluteJoint(5).ID = '[M6n692nmwUBZGWvfH:-:MJU3k80US1zHsLRky]';
```

```
smiData.RevoluteJoint(6).Rz.Pos = 5.7295784253767679e-08; % deg
```

```
smiData.RevoluteJoint(6).ID = '[MrTdipo7NpBGdEaYl:-:M6n692nmwUBZGWvfH]';
```

```
smiData.RevoluteJoint(7).Rz.Pos = 0; % deg
```

```
smiData.RevoluteJoint(7).ID = '[MfmKeEzS13xr1hWd5:-:M8u/DfEoU5o42gr9m]';
```

```
smiData.RevoluteJoint(8).Rz.Pos = 0; % deg
```

```
smiData.RevoluteJoint(8).ID = '[M8u/DfEoU5o42gr9m:-:Mn1nOcwC08SrJSuUX]';
```

smiData.RevoluteJoint(9).Rz.Pos = 1.799193426557978e-14; % deg
smiData.RevoluteJoint(9).ID = '[MAehd/LFx/ZxMmv8F:-:MKxmLYjmpQJjOE8LP]';

smiData.RevoluteJoint(10).Rz.Pos = 2.5444437451708134e-14; % deg
smiData.RevoluteJoint(10).ID = '[MKHV3mBwZGVI+NY88:-:MiRbW1IwB53fv34fK]';

smiData.RevoluteJoint(11).Rz.Pos = 1.799193426557978e-14; % deg
smiData.RevoluteJoint(11).ID = '[MIANUfg2UPIZqXCU8:-:MKHV3mBwZGVI+NY88]';

smiData.RevoluteJoint(12).Rz.Pos = 0; % deg
smiData.RevoluteJoint(12).ID = '[MKxmLYjmpQJjOE8LP:-:MIANUfg2UPIZqXCU8]';

smiData.RevoluteJoint(13).Rz.Pos = 1.799193426557978e-14; % deg
smiData.RevoluteJoint(13).ID = '[MiRbW1IwB53fv34fK:-:MoyKIxtsieEkdXtfj]';

smiData.RevoluteJoint(14).Rz.Pos = 1.799193426557978e-14; % deg
smiData.RevoluteJoint(14).ID = '[Mn1nOcwC08SrJSuUX:-:MrTdipo7NpBGdEaYI]';

6 List of figures

Fig 1.1 Frontal, trasversal and sagittal planes

Fig 1.2 Step length and width and stride

Fig 1.3 step cycle phases

Fig 1.4 stance and swing phases

Fig 1.5 Initial contact on the ground

Fig 1.6 Load response

Fig 1.7 Intermediate stance

Fig 1.8 Terminal stance

Fig 1.9 initial swing to following stance manoeuvre

Fig 1.10 7-links human model

Fig 1.11 a) Heel position and velocity b) Meta position and velocity c) Representation of the body displacement at heel stike

Fig 1.12 Position and velocity of the COM during the gait cycle

Fig 1.13 angular measures of the hip during gait cycle

Fig 1.14 angular measures of the knee during gait cycle

Fig 1.15 angular measures of the ankle during gait cycle

Fig 1.16 angular measures at natural cadence

Fig 1.17 angular measures at natural cadence on a set of patients (left); angular measures at different cadences

Fig 1.18 Support polygons during one gait cycle

Fig 1.19 Progress of the COP under the foot during walking

Figure 1.20. Support foot and the influence of the ankle by the force , moment , ground reaction by force and moment and gravity.

Figure 1.21 Ideal ZMP

Figure 1.22 Ideal baso signals

Figure 2.1 Robot assembly

Figure 2.2 Robot's trunk

Figure 2.3 Robot's hip joint

Figure 2.4 Robot's thigh

Figure 2.5 Robot's leg

Figure 2.6 Robot's ankle

Figure 2.7 Robot's foot

Figure 2.8 Robot's tip

Figure 2.9 Mechanism configuration, solver configuration and world reference frame blocks

Figure 2.10 Ground Simulink block

Figure 2.11 Trunk Simulink block

Figure 2.11 Right leg whole Simulink block

Figure 2.12 Particular on the leg's blocks cascade (right thigh linked to the right leg through the knee)

Figure 2.13 Knee joint block implementation

Figure 2.14 Friction coefficient in function of tangential velocity

Figure 2.15 Sensor placed under the foot for baso extraction

Figure 2.16 Simulink implementation of the contact with the ground

Figure 2.17 Contact spheres under the robot's foot

Figure 2.18 Passive toe joint representation

Figure 2.19 Passive toe joint Simulink implementation

Figure 2.20 ZMP Simulinks block

Figure 2.21 Baso signal algorithm simulinks' implemetation

Figure 3.1 Linear 3D Inverted Pendulum

Figure 3.2 Link coordinate frames of the robot's right leg

Figure 3.3 LIPM animation (left) and feet trajectories (right)

Figure 3.4 Initial stance isometric view (left) posterior view (right)

Figure 3.5 Initial manoeuvre stance isometric view (left) posterior view (right)

Figure 3.6 Mid-swing isometric view (left) posterior view (right)

Figure 3.7 Foot switch isometric view (left) posterior view (right)

Figure 3.8 CoG behaviour during gait cycle (yellow line along y-axis and blue line along x-axis)

Figure 3.9 Bertec treadmill

Figure 3.10 marker set used in the recording session (A anterior view, B posterior view)

Figure 3.11 Robot's gait cycle

Figure 3.12 Baso signals from simulation (yellow left foot, blue right foot)

Figure 3.12 Unconstrained ZMP (along y-axis blue, along x-axis yellow)

Figure 3.13 Unconstrained CoG (along y-axis blue, along x-axis yellow)

Figure 3.14 Constrained CoG (along y-axis blue, along x-axis yellow)

Figure 3.15 Constrained ZMP (along y-axis blue, along x-axis yellow)

7 Bibliography

- [1] Capozzo, Analysis of the linear displacement of the heel and trunk during walking at different speeds. _ J. Biomech 14: 411-425, 1981.
- [2] D.A. Winter, Overall principal of lower limb support during stance phase of gait _ J. Biomech. 13: 923-927, 1980.
- [3] D.A. Winter, Energy generation and absorption at the ankle and knee during fast natural and slow cadences. Clin. Orthop. Rel. Res. 175, 1983
- [4] D.A. Winter, Biomechanical motor patterns in normal walking. J. Mot. Behav. 1983
- [5] D.A. Winter, Foot trajectory in human gait :a precise and multifactorial control task. Physiotherapy, 72: 45-56, 1992.
- [6] Ettore Etenzi, “Formulazione e Sviluppo di un Modello teorico di Locomozione Bipedale Passiva in Due Dimensioni.”, master degree thesis, 2013.
- [7] M. Vukobratović, B. Borovac and D. Surdilović, “Zero-Moment Point - Proper Interpretation and New Applications,” IEEE-RAS Int. Conf. on Humanoid Robots, pp. 237-244, 2001.
- [8] M. Vukobratovic and B. Borovac, “Zero-Moment Point – Thirty Five Years of its Life,” International Journal of Humanoid Robotics, vol. 1, pp. 157-173, 2004.
- [9] Bertomeu-Motos, “Biomechanics of human walking and stability descriptive parameters,” Revista Doctorado UMH, vol. 1 no. 1, p4, 2015.
- [10] S. Kajita, The 3D Linear Inverted Pendulum Mode: A simple modelling for a biped walking pattern generation, 2001.
- [11] Sebastian Castro, MathWorks, ” Modeling and Simulation of Walking Robots”.
- [12] M. Ali, “Closed-Form Inverse Kinematic Joint Solution for Humanoid Robots”, 2010
- [13] Omer Eldirdiry, “Modeling of a biped robot for investigating foot drop using MATLAB/Simulink”, 2019.
- [14] R. P. Paul, B. E. Shimano, and G. Mayer, “Kinematic control equations for simple manipulators,” in IEEE Transactions on Systems, Man and Cybernetics, vol. 11, 1981, pp. 449–455.
- [15] D. L. Pieper, “The kinematics of manipulators under computer control,” Ph.D. dissertation, Stanford Univ., 1968.

[16] Claudiane A. Fukuchi, "A public dataset of overground and treadmill walking kinematics and kinetics in healthy individuals", university of texas, 2018.

[17] D. A. Winter," Biomechanics and Motor Control Of Human Movement", 2009.