

POLITECNICO DI TORINO

---

Master's Degree Course in Mechatronic Engineering

Master's Degree Thesis

# Modelling and Simulation of a Four Legged Robot



**Advisors:**

Prof. Giovanni Gerardo Muscolo

**Candidate:**

Francesco Tracuzzi Spadaro  
242082

---

Academic Year 2019 - 2020



*Alla mia famiglia*

## **Abstract**

Legged animals are capable of rapid movement, are efficient from the point of view of the energy and are able to adapt their gait depending on environmental conditions. Motions like walking, trotting, galloping and jumping, are challenging to evaluate and replicate due to their being consequence of tricky interaction of different systems (like the musculoskeletal system and the central and peripheral nervous systems, not to mention the influence of the environment). In this thesis, is presented developed and studied a four-legged robot with 1 DOF for each leg and passive springs in knees and spine. Our objective is to increase the motion performances of the robot modifying the stiffness value of springs. The results obtained from the simulation underline how the spine influences the performance of the robot by increasing the speed and reducing the energy required by the motors for the same input signal.



# Contents

List of Tables	4
List of Figures	5
1 Introduction	7
<b>I Software used</b>	<b>9</b>
2 Mathworks	11
2.1 Matlab . . . . .	11
2.2 Simulink . . . . .	14
2.2.1 Simscape multibody . . . . .	16
2.3 SolidWorks . . . . .	20
<b>II Four legged Robot Model</b>	<b>27</b>
3 Robot planar model	29
4 3D Virtual Model	33
4.1 Central Architecture . . . . .	33
4.2 Simscape Model . . . . .	35
4.3 SolidWorks Model . . . . .	36
<b>III Results</b>	<b>39</b>
5 Simulation, results and discussion	41
5.1 Simulation . . . . .	41
5.1.1 Test 1: higher stiffness in all joints legs and spine . . . . .	42
5.1.2 Test 2: high stiffness for legs - spine compliant . . . . .	42
5.1.3 Test 3: Compliance in the legs - spine with high stiffness . . . . .	47

5.1.4	Test 4: legs with high stiffness - spine sinusoidal . . . . .	47
5.1.5	Model 2 . . . . .	47
5.1.6	Test 5: legs sinusoidal behaviour - spine maximum stiffness .	49
5.2	Discussion . . . . .	49
5.3	Conclusion . . . . .	54
<b>A</b>	<b>Script</b>	<b>55</b>
<b>B</b>	<b>Simscape simulation results</b>	<b>63</b>
B.1	Test 1: higher stiffness in all joints legs and spine . . . . .	63
B.2	Test 2: high stiffness for legs - spine compliant . . . . .	66
B.3	Test 3: compliance in the legs - spine with high stiffness . . . . .	68
B.4	Test 4: legs with high stiffness - spine sinusoidal . . . . .	70
B.5	Test 5: legs with sinuroidal behaviour - spine maximum stiffness . .	72
	<b>Bibliography</b>	<b>75</b>

# List of Tables

4.1	Data of model torso . . . . .	36
4.2	Data of legs . . . . .	36
4.3	Data of model torso . . . . .	37
4.4	Data of legs . . . . .	37
5.1	Test 1 data . . . . .	42
5.2	Test 2 data . . . . .	42
5.3	Test 3 data . . . . .	47
5.4	Test 4 data . . . . .	47
5.5	Test 5 data . . . . .	50
5.6	Resume data . . . . .	51

# List of Figures

2.1	Screen of MatLab environment . . . . .	12
2.2	Genetic Algorithm a step output . . . . .	14
2.3	Screen of MatLab environment . . . . .	15
2.4	Simulink library . . . . .	15
2.5	Selection of new Simscape Multibody model . . . . .	16
2.6	Simscape multibody initial model . . . . .	17
2.7	Simscape multibody initial model . . . . .	17
2.8	main view of the simscape model . . . . .	18
2.9	view of the torso subsystem . . . . .	19
2.10	Sensor subsystem . . . . .	19
2.11	view of the leg subsystem . . . . .	19
2.12	Assembled trunk with motors, SolidWorks model; trimetric view . .	20
2.13	Tronco . . . . .	21
2.14	SolidWorks model of anterior right leg . . . . .	22
2.15	Foot coverage . . . . .	22
2.16	SolidWorks model of upper anterior right leg . . . . .	23
2.17	SolidWorks model of lower anterior right leg . . . . .	23
2.18	Complete SolidWorks model, of quadruped robot, clearly visible we see legs, motors and motors support. . . . .	24
2.19	In the figure is shown a window of a solid block of Simscape Multi- body, in which are visible the parameters and the files used for the import of the model from SolidWorks to Simscape . . . . .	24
3.1	Robot planar model. $k$ indicates the stiffness of the springs for each leg, the damping factor is indicated $b_r$ or $b_f$ . $l_0$ indicates the resting length of the springs. While the leg angle is characterized by $\beta$ . The orientation angle of the front part of the body is characterized by $\theta_f$ . The orientation angle of the rear part of the body is characterized by $\theta_r$ . $L$ indicates the length of each part half part of the body. $F$ represents the force of the leg, and its orientation coincides with the orientation of the leg. . . . .	31
3.2	Robot planar model presented in this thesis . . . . .	31
4.1	General system architecture . . . . .	34
4.2	Angular input of hip motors . . . . .	34

4.3	Example of walking in two different instant of time. . . . .	35
4.4	Hip motor control system . . . . .	35
4.5	Simscape model robot side view . . . . .	36
4.6	SolidWorks model robot side view . . . . .	37
5.1	Evolution of point A during gait in test 1 . . . . .	43
5.2	Spine oscillation, test 1 (x axis) . . . . .	44
5.3	Legs anterior torque, test 1 . . . . .	44
5.4	Legs posterior torque, test 1 . . . . .	44
5.5	Legs anterior power, test 1 . . . . .	44
5.6	Legs posterior power, test 1 . . . . .	45
5.7	Evolution of point A during gait, test 2 . . . . .	45
5.8	Spine oscillation, test 2 (x axis) . . . . .	46
5.9	Legs torque, test 2 (right anterior) . . . . .	46
5.10	Evolution of point A during gait, test 3 . . . . .	48
5.15	Legs torque, test 4 (Right anterior) . . . . .	48
5.11	Spine oscillation, test 3 (x axis) . . . . .	49
5.12	Legs torque, test 3 (Right anterior) . . . . .	49
5.13	Evolution of point A during gait, test 4 . . . . .	50
5.14	Spine oscillation, test 4 (x axis) . . . . .	51
5.16	Evolution of point A during gait, test 5 . . . . .	52
5.17	Spine oscillation, test 5 (x axis) . . . . .	52
5.18	Legs torque, test 5 (Right posterior) . . . . .	53
5.19	Trend of velocities for each test . . . . .	53
B.1	Evolution of point A during gait in Simscape model, test 1 . . . . .	64
B.2	Spine oscillation of Simscape model, test 1 . . . . .	64
B.3	Hip joints anterior torque of Simscape model, test 1 . . . . .	65
B.4	Hip joints posterior torque of Simscape model, test 1 . . . . .	65
B.5	Legs anterior power of Simscape model, test 1 . . . . .	65
B.6	Legs power of Simscape model, test 1 . . . . .	65
B.7	Evolution of point A, of Simscape model, during gait, test 2 . . . . .	67
B.8	Spine oscillation of Simscape model, test 2 . . . . .	67
B.9	Legs torque of Simscape model, test 2 . . . . .	67
B.10	Evolution of point A during gait, test 3 . . . . .	69
B.11	Spine oscillation, test 3 . . . . .	69
B.12	Legs torque, test 3 . . . . .	69
B.13	Evolution of point A, of Simscape model, during gait, test 4 . . . . .	71
B.14	Spine oscillation of Simscape model, test 4 . . . . .	71
B.15	Legs torque of Simscape model, test 4 . . . . .	71
B.16	Evolution of point A during gait, test 5 . . . . .	73
B.17	Spine oscillation, test 5 . . . . .	73
B.18	Legs torque, test 5 . . . . .	73

# Chapter 1

## Introduction

Nowadays, in a society where everything is influenced by the innovation and every area of human knowledge can be supported and improved by the use of technology, robotics has a central role in our life.

Mobile robots have a lot of applications from military to space exploration, from industrial use to their implementation in daily life. There are two main branches of mobile robots depending of their structure, the wheeled and the legged one, the combination of these two branches gives the opportunity to conceive novel robotic architectures. The wheeled mobile robots are famous for their massive application in land exploration, the advantage of their structure is to be more simple to construct and easy to control, but certainly less flexible.

In legged robots, mobility and versatility are fundamental characteristics to be considered allowing more complex functions. For this reason, researchers, inspired by animals, work on legged robots that carrying out incredible performance of movement in a hard environment. However, legged locomotion presents some disadvantages: these systems are more complex, unstable and harder to control.

The mechanism of mammalian movement has been studied from the biological point of view [1] showing how big mammals preserve much energy during their stride through an elastic structure in their legs, So the spring-mass model was proposed [2] [3]. On the same topic, the authors of [11] are focused examining the passive dynamics of four-legged robots, with a discussion of the importance of forwarding speed versus touchdown angle in the self-stabilized Spring-Loaded Inverted Pendulum (SLIP).

Additionally, in [4], simple control laws for asymmetric gaits implemented on SCOUT II robot are presented. Authors of [5] studied the effect of musculoskeletal spring in legged animals, they modelled trotting and hopping as a spring-mass system, in different animals.

Furthermore analyzing a wide size range of animals at equivalent speed, it appears that larger animals have stiffer leg spring, proportional with the body mass ( $M^{0.67}$ ),

but the angle swept by the leg spring is nearly independent of body mass. This involves a resonant period of vertical vibration of the system, longer in larger animals. Authors of [6] proposed a control framework for a quadrupedal robot able to drive locomotion with different gaits. Also, the studies of [7] are based on SLIP. Authors presented a stability criterion on which the necessary and sufficient condition of the galloping stability for the quadruped robot is proposed to attain a controlled thrust. The results demonstrated that imbalance posture of the trunk could be stabilized by adjusting the stiffness of the four legs. In article [8] authors presented a new "cheap" method to achieve bounding gait in four-legged robots with a minimalistic approach. They showed that body velocity could be controlled using a straightforward method with only one parameter to vary. Another simple control method is presented in [9], in which the controller works through an adaptive frequency oscillator that autonomously tunes the intrinsic frequency of the oscillators to the resonant frequency of a compliant four-legged robot. A computational model that unifies the well-known scale effects in running quadrupeds is presented in [10]. The model is made through a physics-based simulation of six running mammals of different size. Fascinating work is presented by [11], where the basis for a systematic approach to design legged robots is presented. A four-legged robot with dynamical stable running is used, is introduced a non-dimensional criterion based on the robot's forward speed and the required power to sustain a passive motion. An important aspect to study on quadruped motion is the contribution provided by the presence of the spine in the system. Its effect and leg behaviour are presented in [12] and [13], where researchers proposed a simplified model, on the sagittal plane, of quadruped mammals with articulated spine joint. Then control method was formulated in order to obtain symmetrical bounding of the model. As a result of the growing number of researches in the robotic field, the cost and the time employed for prototyping is high. For this reason, scientists are motivated to use different tools for simulation testing in a 3D environment, there are many tools on the market and the authors of [14] presented a four-legged robot modelled with SimMechanics. Also in [15] is illustrated how to develop a mechatronic system and how dynamic model can be used to communicate between design groups and with suppliers.

In this works, the design of a symmetric four-legged robot is developed. The virtual model of the robot has a flexible spine, it is able to walk with a stable stride and the robot present 1 DOF for each leg. The aim of this thesis is to study the variation of the system response when the stiffness of each spring joint is modified, furthermore, is shown the impact that brings the flexible spine on the system. The thesis is structured as follows: in part II are presented the analytical equation, the dynamics of the model, the 3D model used for simulation; in part 4, section 4.1 the control architecture are explained while in 5.1 the simulation tests are shown. Section 5.2 and 5.3 show respectively discussion and conclusion.

# Part I

## Software used





## Chapter 2

# Mathworks

With the purpose to create a model in a 3D environment, I used the software produced by MathWorks®, which is a multinational corporation specialized in development of programs with the purpose of mathematical calculations. The core products are MATLAB 2.1 and Simulink 2.2. These programs comply with the requirement necessary for the thesis goals. In this chapter, are presented with a brief description the principal elements of these programs, from the Matlab script in section 2.1 to the 3D model with the help of Simulink in section 2.2 and Simscape Multibody in section 2.2.1.

### 2.1 Matlab

MATLAB is the abbreviation of MATrix LABoratory, and it was created at the end of the 70s. MATLAB is an environment, developed in C, with the specific purpose to simplify the operation of numerical calculation and statistic analysis. It contains the same-named programming-language that allows the manipulation of the matrix, the visualization of the functions and data with ease. The software part of this thesis work is divided into different files for a better organization of the work. One file was created for the robot's parameters, one for starting the simulation and one file for each robot configuration. In order to compute the better angular position as reference input of motor hip joints, a genetic algorithm was used, thanks to the Global Optimization Toolbox developed by Mathworks. At the start of MATLAB we will be in front of the classic default window of the program that is visible in the figure 2.1. Here we find the Command Window, the workspace, the current folder and the details. At the top, there is a menu with several icons. Through the command window, it is possible to carry out interactive programming, and in this case, all the written instructions will be saved in the command history that can be activated by going to the command window and using the up and down arrows. It can be useful to write instructions for a particular program, through scripts, which

are MATLAB files of type M (M-file). When an M-file is executed, the instructions written in it are executed in sequence. In addition, to better organize the work you can call M-file, in other scripts such as this one in which it was defined a script for the parameters of the robot that was called in other scripts. To create a script file, click on the first icon on the left of the command bar. An essential tool for searching for and correcting errors in scripts is the debugger. The errors it recognizes are syntax errors and runtime errors. Also, thanks to breakpoints, you can stop the execution of a script and execute the same instruction by instruction and analyze the behaviour of variables.

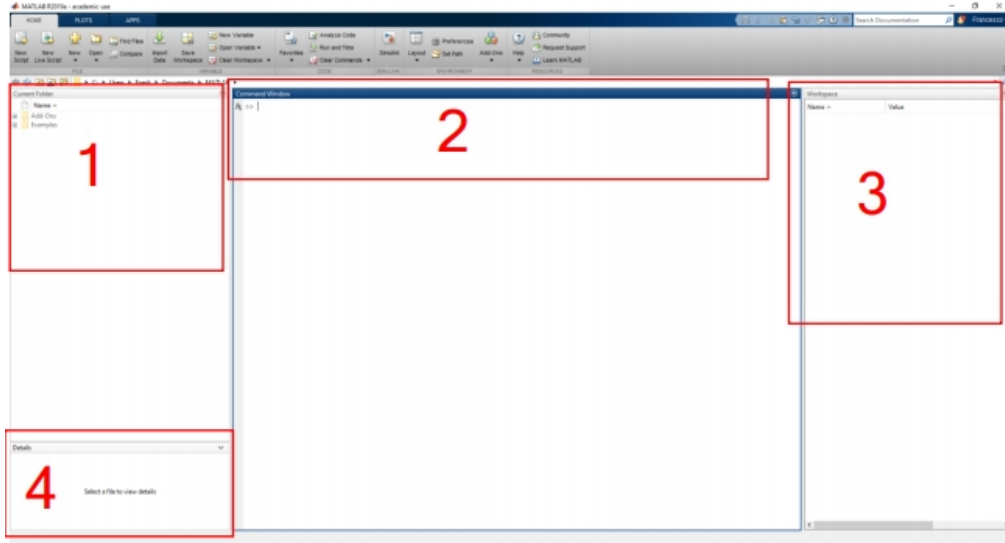


Figure 2.1: Screen of MatLab environment

In appendix A are reported all the scripts used for modelling and simulate the quadruped-robot. The script A.1 was used to define the parameters of the Simscape 3D model. The script sets all the variables useful for setting the parameters of solid blocks and for rigid translations. Also, the coefficients of friction for the modelling of the plane are defined while the parameters of the SolidWorks model are imported in Simscape through file generated directly by the Simscape Multibody Link. In A.2 is shown the script used to start the model simulation directly from MATLAB. In the first part of this script are defined the directories from which the script can take the different m-files, then are set the values of stiffness and damping of the springs of the model and the data to achieve the angular input for the movement of the motors are loaded. These data are interpolated according to the period chosen for the movement of the robot and finally with the `sim` the simulation is started directly from the script. The data obtained from the simulation are saved in a variable called `out_spring` and are processed by the script `DataPlot` A.3, which takes care of producing the graphs shown in the following chapters.

To calculate the angular values to be sent to the joints on the hip and shoulder;

it was decided to use an optimization algorithm. It is necessary to make some assumptions for the model, to define the search space of the algorithm better.

- Input:
  - have a fixed number of waypoints for our walking trajectory;
  - discretize the angle values that these trajectories can take. We're going to use 2.5 degrees;
  - fixed gait period, so each waypoint is going to run at periodic interval;
  - two legs going to mimic the motion of the other two legs exactly but with a lag of half period;
  - A cubic interpolation of the trajectory of the spline is performed;
- Output:
  - The output of the optimization algorithm will be a cost function set by us, which mathematically can provide feedback on how well the optimization algorithm is working with a certain type of input parameters.

Based on what was said about the output, it was necessary to assign good and bad scores on some simple choices. The good costs are affected by the reached final distance, and if the robot does not fall during the simulation. Among the elements that penalize or give a negative result we find, the angular speed of the robot, in fact, we want the robot to move on a straight line so if it starts to deviate is given a negative score. Another case in which a negative score is given is that of aggression. This means that a signal with a minimum number of oscillations is preferable. The script [A.4](#) that takes care of the penalty count is shown in the appendix, in this case, to make it work in symbiosis with the optimization algorithm it is necessary to write it as a function.

The choice of the optimization algorithm falls on the Genetic Algorithm, provided by MathWorks within the Global Optimization toolbox. Among the reasons why this algorithm was chosen, we find the high non-linearity of the walking problem. It is possible to find several local minima during the optimization. Furthermore, It is possible to see how the cost function shows a stepped trend, and each step represents a local minimum for the algorithm as you can see in [fig. 2.2](#). In any case, the genetic algorithm or a global optimization algorithm will never guarantee that the solution provided is the optimal one, so it is necessary to perform multiple simulations to try to extract the best gait. The optimization options that have had the most impact on this problem are:

- Population size, (how many different samples are going to keep track of as we run the algorithm):

- Initial population (initial condition, we want to start from a completely unknown gate or do we want to start from an existing gate that we know works which can help us converge quicker, but at the same time it might help us converge to a local minimum that was guided by that initial condition;
- Max/stall generation;
- Parameter bounds ( this is useful again in narrowing down the search space as well as giving realistic trajectories, one of the parameters that I enforced was that the hip angle of legs could not be greater or smaller than +45 or -45 degrees);

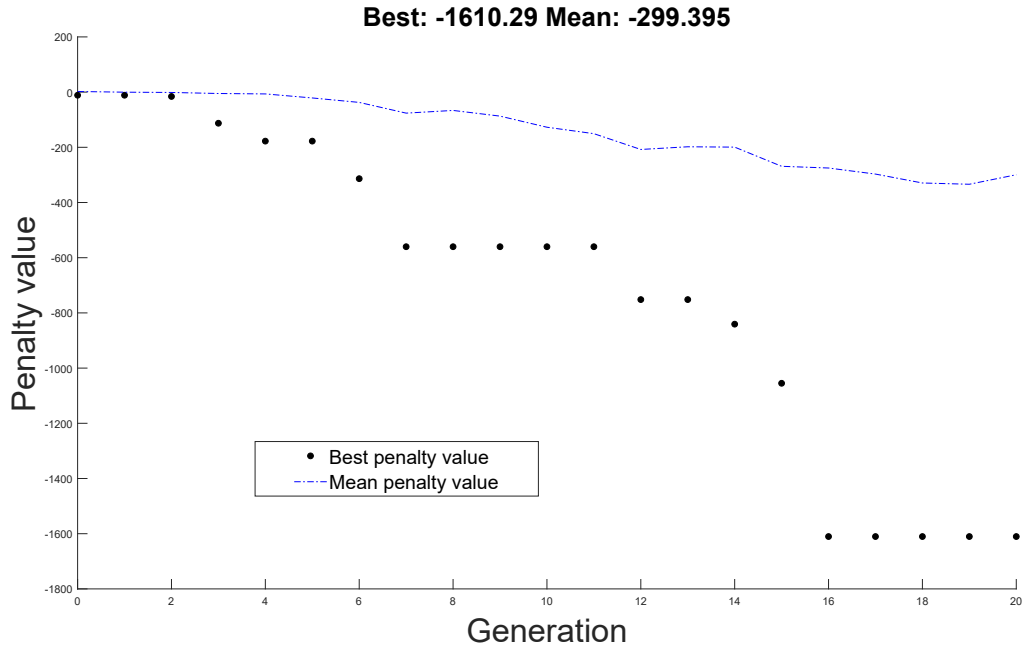


Figure 2.2: Genetic Algorithm a step output

## 2.2 Simulink

Simulink is a MATLAB environment designed to model, simulate and analyze multi-domain dynamical systems, it is a graphical diagramming tool, and his key point is the complete integration with MATLAB environment. The main field of use of Simulink are signal processing, automatic control and model-based design. Opening a new blank model of Simulink we are faced with a completely white screen with the classic menu bars and icons as you can see in the figure 2.3. One of the fundamental elements of the program is the library, which you can open from the View/Library Browser menu or with the dedicated icon. Opening it, we find a

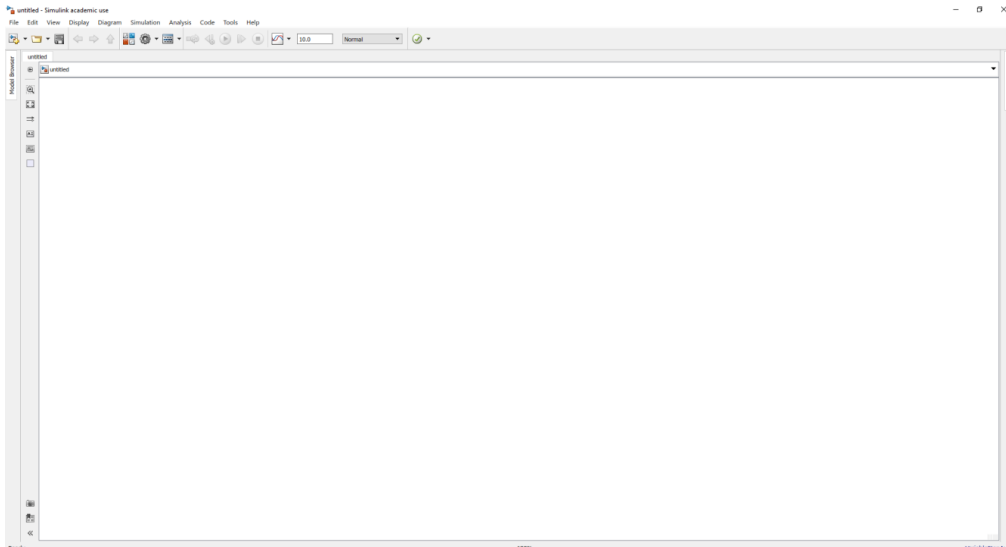


Figure 2.3: Screen of MatLab environment

screen as in figure 2.4. On the left side, there are all the libraries of the program. These are very varied and allow you to deal with most of the problems of modelling and simulation. Among the most used libraries among those provided are those with blocks of frequent use, those for modelling in discrete time and continuous time and others such as that of sources or sinks. It is possible to add new libraries by installing dedicated toolboxes, such as the Simscape one, used in the thesis. To

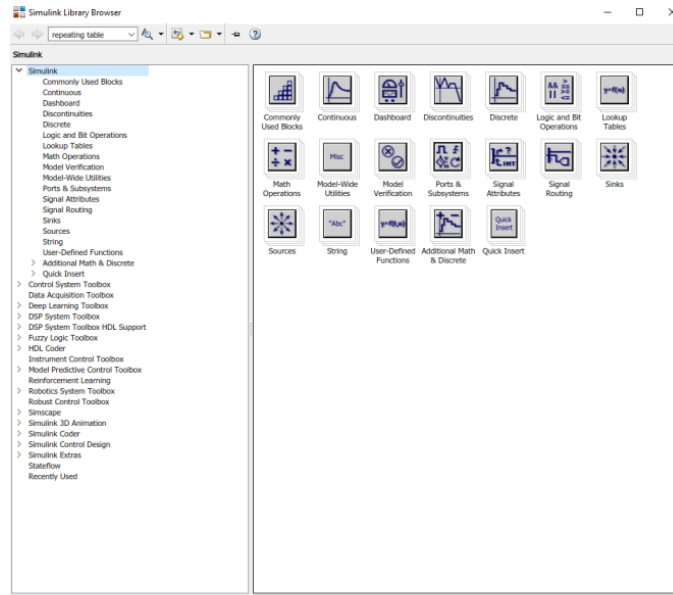


Figure 2.4: Simulink library

have consistent and comparable results, it is necessary to set, correctly, the method of resolution of the differential equations of the system. In this case, to simulate the quadruped in the different tests, we used the ode15s algorithm, choosing as a type of integration step, a variable step. Moreover, it was necessary to modify the basic parameters, following the indications of the contact force library, in order to obtain the best possible results in the simulation. The recommended parameters are: as solver ode15s, relative tolerance  $1e - 4$ , absolute tolerance  $1e - 5$  and maximum step size  $1e - 2$ .

### 2.2.1 Simscape multibody

Simscape Multibody (formerly called SimMechanics) is a toolbox developed by Mathworks that provides an environment for multibody simulation of a 3D mechanical system like robots. It is a handy tool because, after the creation of a multibody system by means of simple blocks, it compute and solve the equation of motion of the entire system and is a handy tool to create a control system or to test the performance of a system. The model designed for this thesis is created as the first step with Simscape Multibody using simple blocks, such as spheres or cylinders. To start designing a new model with Simscape Multibody, it is sufficient, starting from Simulink, selecting the simscape option and from there choosing multibody, as shown in the fig 2.5. Alternatively, you can create the new model by typing from the command line `smnew`. The new model created will appear as shown in the fig 2.6.

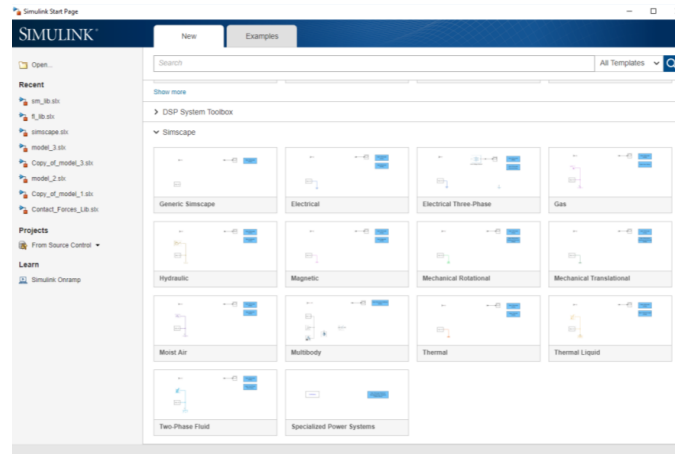


Figure 2.5: Selection of new Simscape Multibody model

In the new Simscape model created, we find some fundamental blocks for the creation of the quadruped robot. In the figure 2.7 we find highlighted the main blocks useful for this purpose. The first block, highlighted with the number 1, is the solver configuration, that defines the solver settings used for the simulation. The block

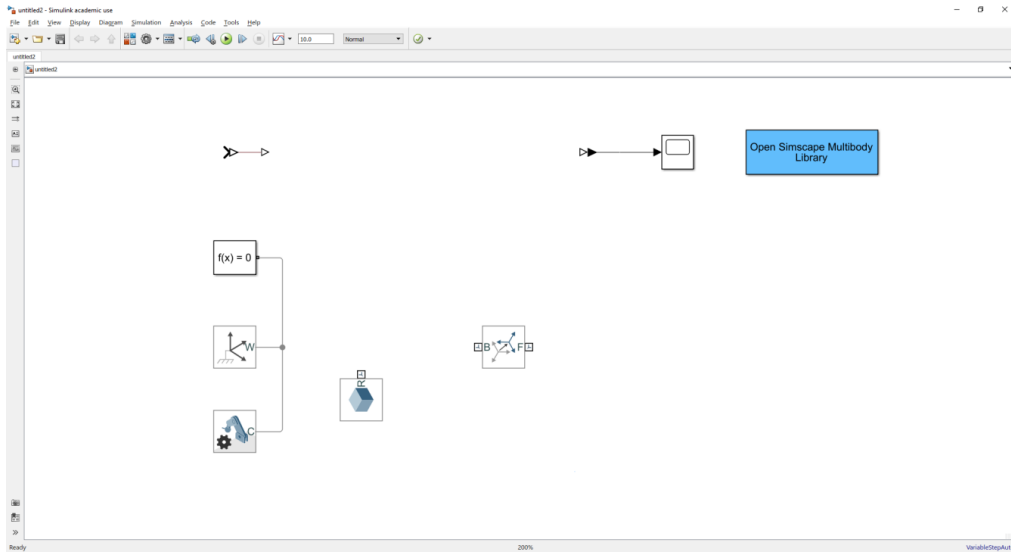


Figure 2.6: Simscape multibody initial model

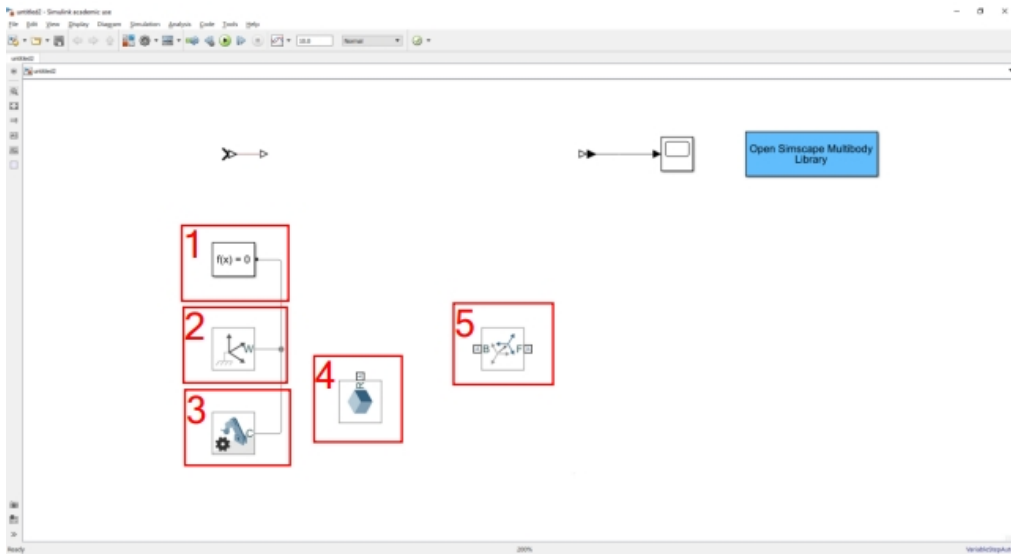


Figure 2.7: Simscape multibody initial model

highlighted with the number 2 is the world frame. This block provides access to the world or the ground frame, a single frame of coordinates orthogonal, motionless, orthogonal, right-handed, predefined in any mechanical model. The world frame is the basis of all frame networks in a mechanical model. A model may have multiple World Frame blocks, but all represent the same frame. The third block is the mechanism configuration, and this allows you to set the mechanical and simulation parameters that apply to an entire machine, the target machine to which the block



is attached. In the Properties section below, you can specify uniform gravity for the entire mechanism and also set the linearization delta. The linearization delta specifies the perturbation value that is used to calculate the numerical partial derivatives for linearization. Blocks 4 and 5 are fundamental for the system design. The first represents a solid. This combines geometry, inertia and mass, a graphic component and rigidly fixed frames in a single unit. A solid is the typical construction block of rigid bodies. Solid block calculates inertia from geometry and density or geometry and mass or by a user-specified inertia tensor. The second is a rigid Transform block, which defines a rigid 3-D transformation between two frames. The transformation rotates and translates the follower port frame (F) with respect to the base port frame (B). Connecting the frame ports in reverse causes the transformation itself to reverse. In the transform block is possible to specify, independently, the translational and rotational parts of the transformation. Different translations and rotations can be freely combined.

The model created with Simscape has been divided into subsystems to improve the organization and reading of the same. In figure 2.8 is shown the main screen that is displayed when the Simscape model of quadruped robot is opened. In the figure 2.8

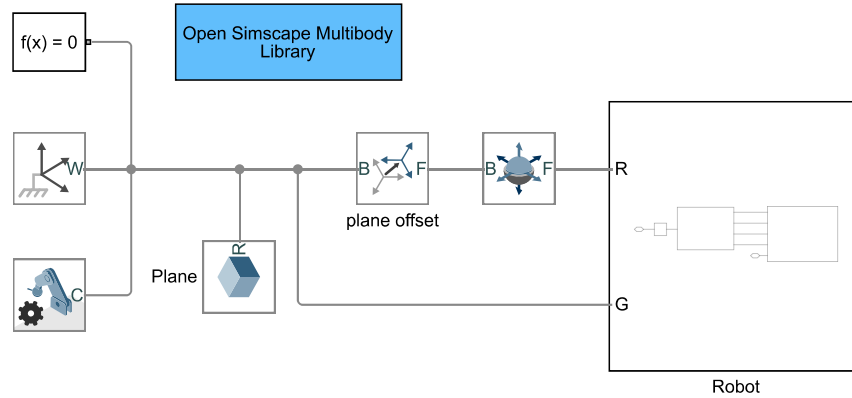


Figure 2.8: main view of the Simscape model

above, the basic blocks (world frame, solver configuration and mechanism configuration) seen in the figure 2.7 are shown. Besides, there is a joint with 6 DOFs. This joint has 3 degrees of translational freedom and 3 degrees of rotational freedom. It is fundamental for the movement of the robot in space. The figure also shows a subsystem called "Robot", within which there are other subsystems that represent the robot's torso and legs. The figure 2.9 shows the model with which the torso of the robot was realized. As you can see in the figure, the trunk is made up of simple blocks, blocks of rigid transformations and simple solids. In addition, in this case, there is a joint of revolution needed for the relative movement of the two parts of the trunk. In the trunk subsystem, another subsystem exists. This, visible in

the figure 2.10, is called "Sensor". This subsystem is used to calculate the position along the x, y and z axes. It is also used to measure the speed of rotation along the z-axis of point A relative to the world frame. Figure 2.11 shows the model used

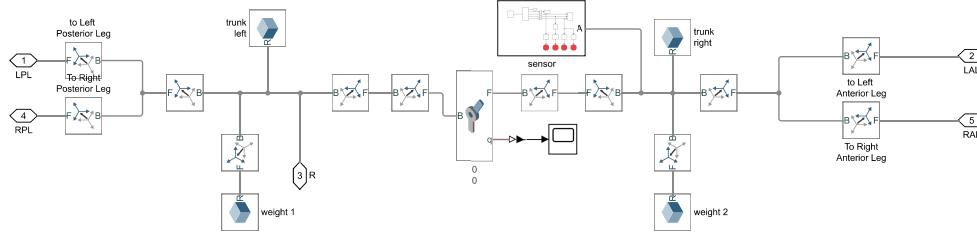


Figure 2.9: view of the torso subsystem

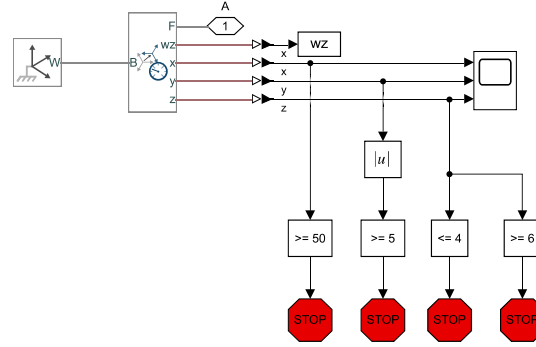


Figure 2.10: Sensor subsystem

to make the robot's legs. All 4 legs have the same structure made of blocks for rigid transformations alternated to a solid block. An active hip joint receives the position of the reference leg at the entrance. A second joint, in this passive case, is used in the knee, in which the stiffness and damping values for the torsion springs are set. In the model, there are three solid blocks, and these are used to make the upper and lower leg and also the foot.

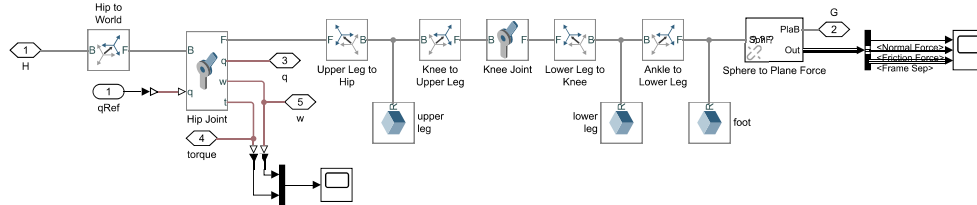


Figure 2.11: view of the leg subsystem

## 2.3 SolidWorks

This section presents the different parts of the quadruple, modelled in three dimensions using SolidWorks. These components represent, at a schematic Simulink level, what was shown in the previous section (2.2). Analysing the Simulink 2.9 scheme presented above, it can be seen that the trunk can be divided into two parts located on the left and right of the rotary joint; these two parts, left and right, are identical and mirrored and are connected to the rotary joint by means of a rotational spring necessary for the purposes of the thesis and useful for reducing the relative movement of the two parts. The 3D model of the trunk, including motors, can be viewed in fig 2.12. In the figure just mentioned (2.12), it is clearly visible

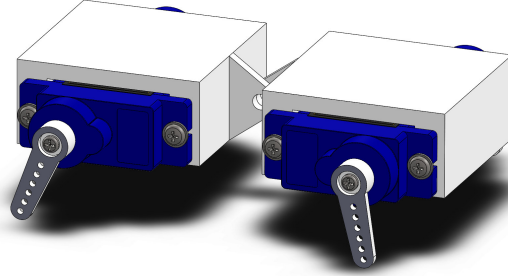
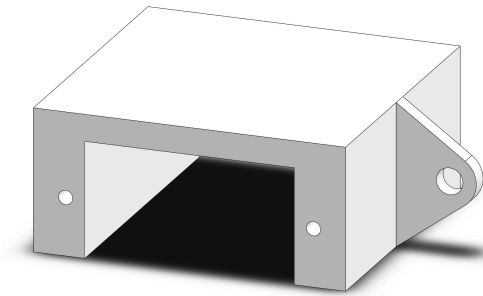
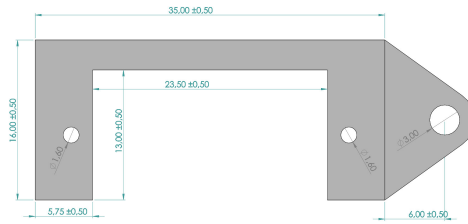


Figure 2.12: Assembled trunk with motors, SolidWorks model; trimetric view

the structure used for the realization of the torso, it is evident the division into two symmetrical parts of the model that are connected to each other by means of a rotational joint that is not present in the 3D modelling but that is represented in the Simulink scheme. More in detail, in the fig 2.13a you can see, with a trimetric perspective, the housing made for the engines and in fig 2.13b with front view and relative dimensions. The following figures show the union of the two parts of the



(a) Trimetric view of the left side of the trunk



(b) Frontal view of the left side of the trunk with dimensions

trunk. In fig 2.13a, with trimetric view, and in fig 2.13b, with front view. Also, the figs 2.13c, 2.13d clearly show the relative movement that the two halves of

the structure make with respect to each other. The Sim scape scheme visible in

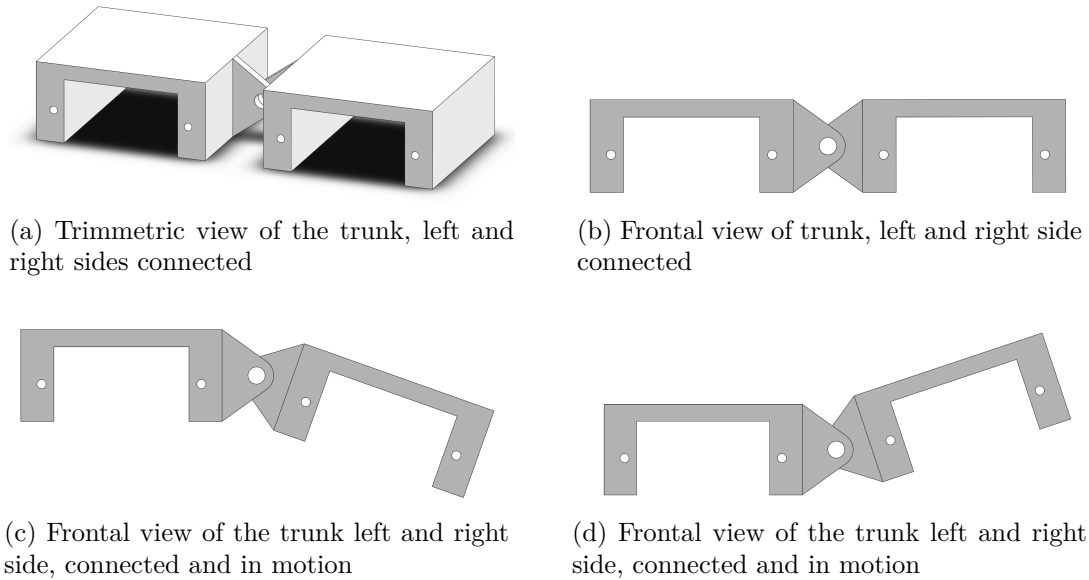


Figure 2.13: Tronco

fig 2.11, as already mentioned, represents the model of the robot leg used for the simulation. The 3D realization of this leg is shown in fig 2.14 visible below, where the leg is shown, with a dimetric view. As you can see also in the same figure (2.14) the leg is shown in two different relative positions, in fact, the lower part can rotate through a passive rotation joint placed in the knee. And move from a minimum starting position (fig 2.14a) set at 40 deg from the vertical of the leg to a maximum position (fig 2.14b) of 50 deg from the starting position or 90 deg from the vertical. In this case, the figure shows the leg used in the front right or rear left position while the legs in the front left and rear right position are mirrored to this (2.14a) The following figures show the individual sections that make up the robot leg. These are made from 3 parts or upper- lower- leg and foot, but in the case of the model SolidWorks the lower leg and the foot are joined. What is displayed as a foot in the Simscape scheme 2.11 is represented by the visible foot cover in fig 2.15. The upper leg is modeled as seen in fig 2.16, you can see here a dimetric view in (2.16a), a front view with the dimensions in 2.16b, and always with a dimension, in 2.16c the lateral view. From the 3D model (2.16) just presented, it became clear that the upper hole is used to connect the leg to the motor, while the lower hole is used to connect the lower leg to the motor. As previously mentioned, the lower leg is, unlike the Simmechanics model, incorporated with the foot, except for the rubber coating that has already been seen above (2.15). In the following group of figures 2.17, the 3D model of the lower leg is shown. Also in this case we find from left to right: dimetric (2.17a), frontal (2.17b) and lateral (2.17c) view (2.17c). Ul-

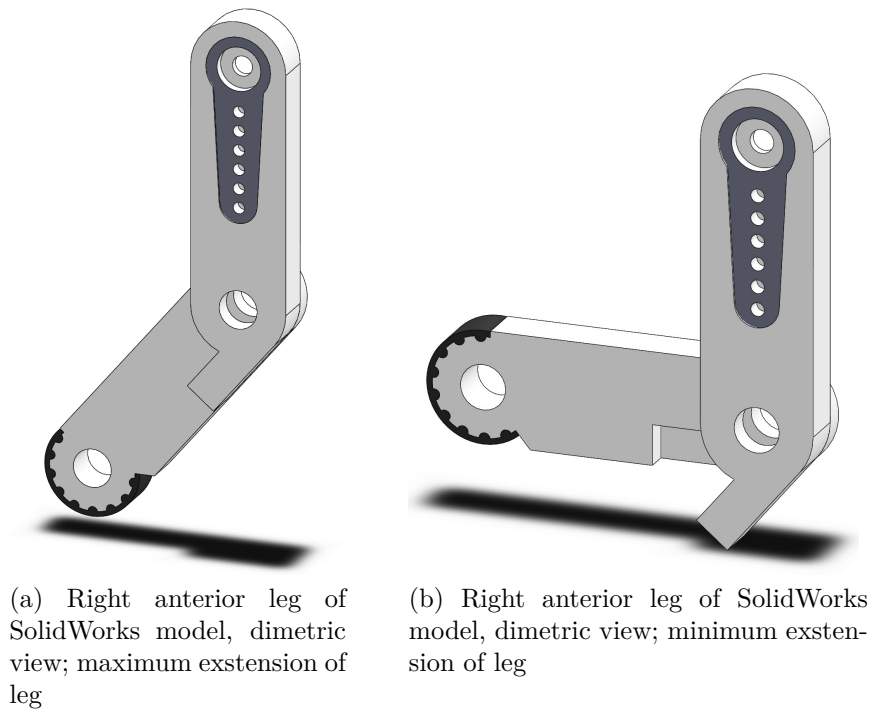


Figure 2.14: SolidWorks model of anterior right leg

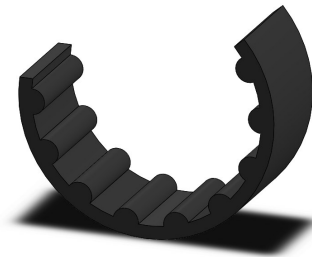


Figure 2.15: Foot coverage

timately, by combining, in the environment of SolidWorks, all the elements seen so far, thanks to the coupling methods provided by SolidWorks, you can go to create a complete three-dimensional model, formed by the union of several parts, visible in fig 2.18. From the last figure (2.18) it is possible to see that the right and left front and back legs are identical and rotated by 180 deg, the same is true for the left front and right back legs. After you have made all the parts of the robot, you need to couple them correctly to make the final 3D model (2.18). Pairing is not a complex operation, but it is essential to prepare the model for export to Simscape Multibody correctly.

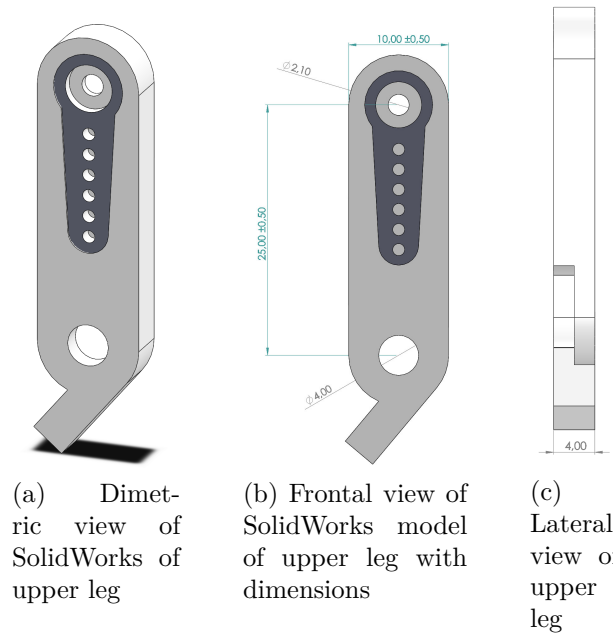


Figure 2.16: SolidWorks model of upper anterior right leg

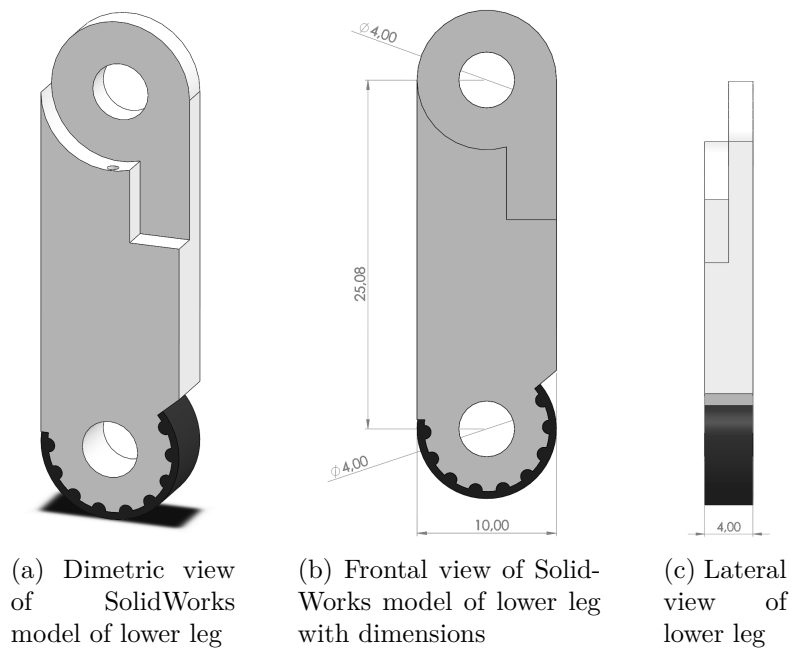


Figure 2.17: SolidWorks model of lower anterior right leg

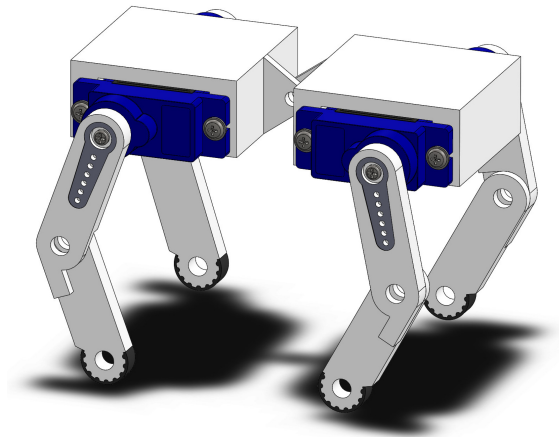


Figure 2.18: Complete SolidWorks model, of quadruped robot, clearly visible we see legs, motors and motors support.

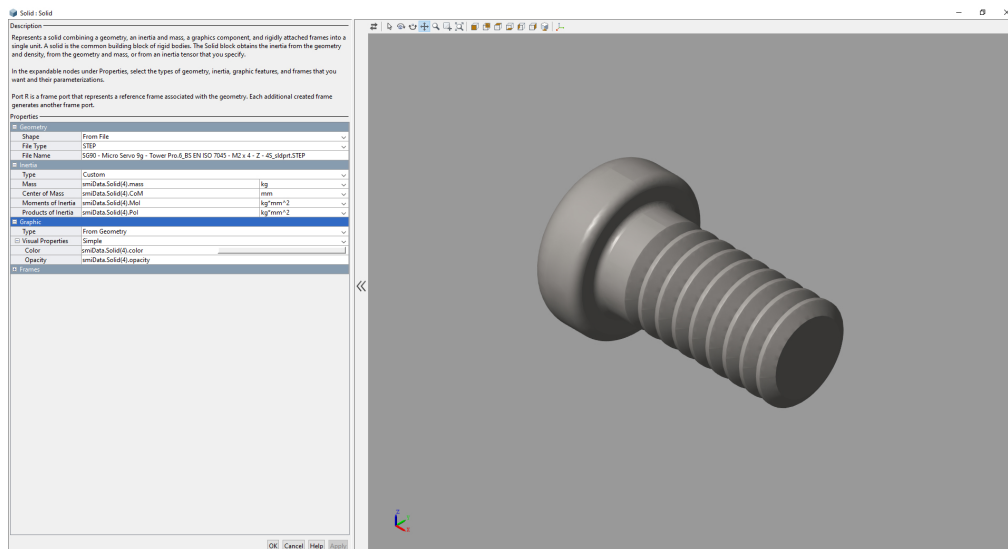


Figure 2.19: In the figure is shown a window of a solid block of Simscape Multibody, in which are visible the parameters and the files used for the import of the model from SolidWorks to Simscape

From the SolidWorks quadruped robot model, it is, therefore, possible through the Simscape Multibody Link plug-in, provided by MathWorks, to export the model, providing the user with files in *.STEP* format for each part of the 3D model. These files will be used within Simscape in solid blocks to take the exact shape of the SolidWorks model as it will be possible to load the solid block geometry from this file. Also, an *.m* file will be provided containing the main information of the different blocks, such as inertia matrices, mass, colour, etc., as well as the

information on the solid block geometry. In fig [2.19](#) you can see how a simple screw can be configured with the parameters just described. After the exportation, to be able to carry out simulations in line with those made with Simscape, it is necessary to make some changes to the exported model, this to ensure that the model of the quadruped, can move in the 3D environment.





## Part II

# Four legged Robot Model



## Chapter 3

# Robot planar model

To understand the dynamic properties of a quadruped robot, a simple planar model is presented and used to study the motion of the system. The four-legged robot is divided into two symmetrical parts linked by a passive joint with torsional springs as shown in fig. 3.2. The model may also be modelled with the sketch of 3.1. In equation (3.1), the dynamic behaviours of the system are shown:

$$A(\mathbf{q})\ddot{\mathbf{q}} = C(\mathbf{q}, \dot{\mathbf{q}}, \hat{\beta}, \hat{\lambda}) + Q(\mathbf{q}, \dot{\mathbf{q}}, \hat{\beta}, \hat{\lambda}) \quad (3.1)$$

this equation is widely described in [12] and [13]. The constituent elements of equation 3.1 are:  $A(\mathbf{q})$  that is the mass matrix,  $C(\mathbf{q}, \dot{\mathbf{q}}, \hat{\beta}, \hat{\lambda})$  is a matrix that includes the velocity-dependent forces, elastic spring and the gravitational forces. The vector  $\mathbf{q} = (x, y, \theta_r, \theta_f)$  contains the generalized coordinates of the model. The orientation angles of legs are  $\hat{\beta} = (\beta_r, \beta_f)^T$ . The vector  $\hat{\lambda} = (\lambda_r, \lambda_f)^T$  is used to indicate the state of legs. The value of  $\lambda = 1$  or  $(0)$  represents the legs landing, leg lands (or does not land) on the ground.  $Q(\mathbf{q}, \dot{\mathbf{q}}, \hat{\beta}, \hat{\lambda})$  is the matrix containing the friction forces, driving forces and driving torques.  $Q$  represents the actuation and dissipation elements this terms are ignored from (3.1), so the dynamics is described by (3.2).

$$A(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}, \hat{\beta}, \hat{\lambda}) \quad (3.2)$$

$\mathbf{q} = [x \ z \ \theta_f \ \theta_r]^T$ ;

$A$  the mass matrix is:

$$A = \begin{bmatrix} 2m & 0 & 0 & 0 \\ 0 & 2m & 0 & 0 \\ 0 & 0 & \frac{1}{8}mL^2 + J & \frac{1}{8}mL^2 \cos(\theta_f - \theta_r) \\ 0 & 0 & \frac{1}{8}mL^2 \cos(\theta_f - \theta_r) & \frac{1}{8}mL^2 + J \end{bmatrix} \quad (3.3)$$

where  $m$  is the mass of the robot,  $J$  represent the inertia of the body, and the other terms are showed in fig. 3.2.

$$\mathbf{C} = [C_1 \ C_2 \ C_3 \ C_4]^T$$

$$C_1 = -k[\lambda_f(l_0 - l_f) \sin(\beta_f) + \lambda_r(l_0 - l_r) \sin(\beta_r)] \quad (3.4)$$

$$C_2 = k[\lambda_f(l_0 - l_f) \cos(\beta_f) + \lambda_r(l_0 - l_r) \cos(\beta_r)] - 2mg \quad (3.5)$$

$$\begin{aligned} C_3 = & kL \left[ \frac{3}{4} \lambda_f(l_0 - l_f) \cos(\theta_f - \beta_f) \right. \\ & - \frac{1}{4} \lambda_r(l_0 - l_r) \cos(\theta_r - \beta_r) \left. \right] \\ & - \frac{1}{8} m L^2 \sin(\theta_f - \theta_r) \dot{\theta}_r^2 - k_s(\theta_f - \theta_r) \end{aligned} \quad (3.6)$$

$$\begin{aligned} C_4 = & kL \left[ \frac{1}{4} \lambda_f(l_0 - l_f) \cos(\theta_r - \beta_f) \right. \\ & - \frac{3}{4} \lambda_r(l_0 - l_r) \cos(\theta_r - \beta_r) \left. \right] \\ & + \frac{1}{8} m L^2 \sin(\theta_f - \theta_r) \dot{\theta}_f^2 - k_s(\theta_f - \theta_r) \end{aligned} \quad (3.7)$$

$$l_r = \sqrt{(x_r - 0.5L \cos(\theta_r) - x_r^{td})^2 + (y_r - 0.5L \sin(\theta_r))^2} \quad (3.8)$$

$$l_f = \sqrt{(x_f + 0.5L \cos(\theta_f) - x_f^{td})^2 + (y_f - 0.5L \sin(\theta_f))^2} \quad (3.9)$$

$$\beta_r = \arctan \left[ \frac{x_r^{td} - x_r + 0.5L \cos(\theta_r)}{y_r - 0.5L \sin(\theta_r)} \right] \quad (3.10)$$

$$\beta_f = \arctan \left[ \frac{x_f^{td} - x_f + 0.5L \cos(\theta_f)}{y_f + 0.5L \sin(\theta_f)} \right] \quad (3.11)$$

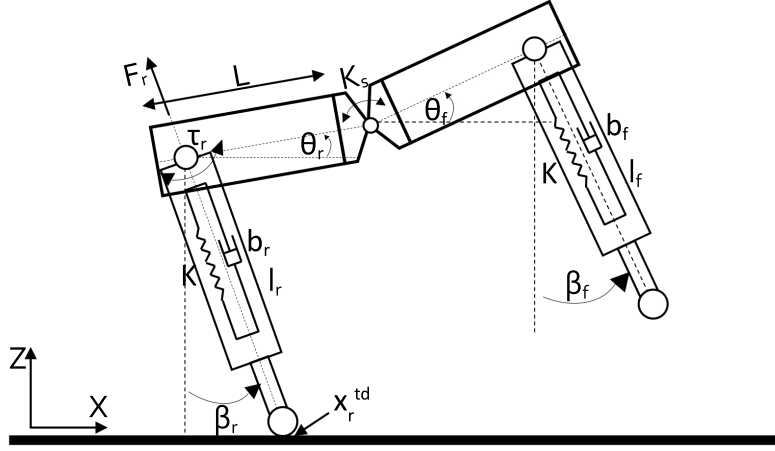


Figure 3.1: Robot planar model.  $k$  indicates the stiffness of the springs for each leg, the damping factor is indicated  $b_r$  or  $b_f$ .  $l_0$  indicates the resting length of the springs. While the leg angle is characterized by  $\beta$ . The orientation angle of the front part of the body is characterized by  $\theta_f$ . The orientation angle of the rear part of the body is characterized by  $\theta_r$ .  $L$  indicates the length of each part half part of the body.  $F$  represents the force of the leg, and its orientation coincides with the orientation of the leg.

The structure of the model presented in this work as you can see fig. 3.2 below is slightly different from that one showed above in fig. 3.1. Indeed the model in this thesis presents a torsional spring in the knee that creates momentum on the legs structure so in this case, then in the case of previous model, the forces acting parallel with the line between the contact point and the motor, must be computed.

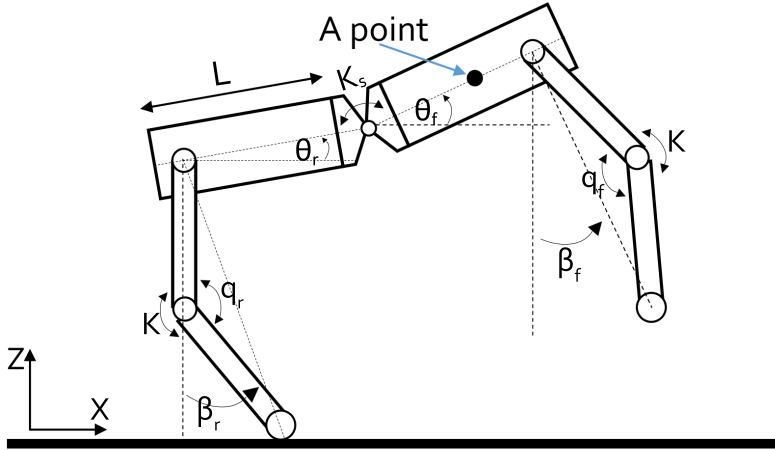


Figure 3.2: Robot planar model presented in this thesis



## Chapter 4

# 3D Virtual Model

To modelling and design the 3D model of the four-legged robot, at the beginning i used Simscape Multibody™, developed by MathWorks®, that provides a simulation environment for 3D mechanical systems. The model of four-legged robot was made using simple blocks such as sphere-, cylinder- and brick-shape. It is possible to split the model into two sections: torso and legs. In a second phase, we carry out a realistic 3D model designed with Solidworks. This was imported in Simscape Multibody and simulated. The trunk is divided into two identical parts separated by the spine. The four legs are identical and are composed by upper-leg, lower-leg and foot. The actuation is provided by 4 motors located at the connection point between hip and trunk. These motors drive the 4 hip joints for walking. We take the servo motor SG90 as a real model of motors for potential future prototyping. This is a small servo motor capable to provide a torque of 2.5 Kgcm which is 0.2452 Nm. The material selected for the model is ABS, is a techno-polymer, that can be extruded in 3D printing. The density of ABS is 1.12 g/cm<sup>3</sup> while for the foot was chosen rubber to reduce sliding with density of 1.7 g/cm<sup>3</sup>.

### 4.1 Central Architecture

The Simscape model is made by subsystems division. In the main section lies the world environment composed by the plane and the Robot subsystem, which contains torso and legs subsystems. In figure 4.1, a general architecture of the system is shown. In the World block, the basic settings like the direction of gravitational force and the origin in the system are defined. The robot system block contains the 3D model, composed by torso and four legs with all the blocks that define them. The stop condition is the logic block that compares the input received from the sensor with the stop condition defined by us. These conditions compare the displacement along the x, y and z axes, measured at point A defined in fig. 3.2, with the world reference. An example of a stop condition is the one defined along the



x-axis, i.e. the displacement of point A by a maximum of 50 cm; in this case, the simulation ends. If the stopping conditions are not met at the moment  $t$ , the simulation continues, in this case, the contact force library returns the reaction of the plane that allows the system moving to respect the initial reference frame defined by the world.

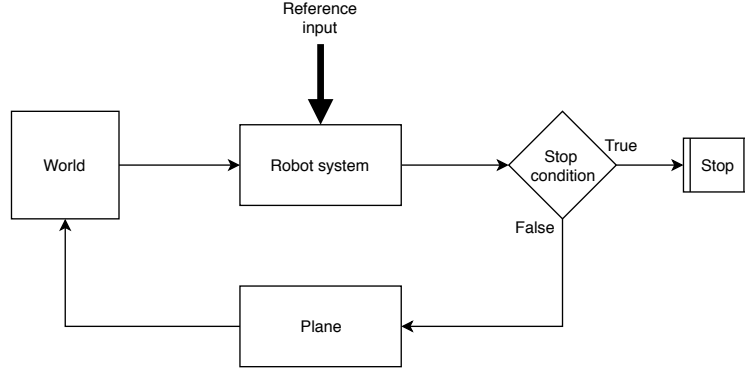


Figure 4.1: General system architecture

To allow robot movement, four hip joint actuators are used to connect legs and torso, while the spine and knees are passive joints with rotational spring. The gait is achieved, sending the angular joints position directly to the actuator with a gait period of 0.8 s. During the walk, right anterior and left posterior legs are activated simultaneously, while the other two legs are shifted of half period. To find good values of angular position to drive hip joints, we used the Genetic Algorithm, thanks to Matlab Global Optimization Toolbox. Fig 4.2 shows the trend of reference angular joint.

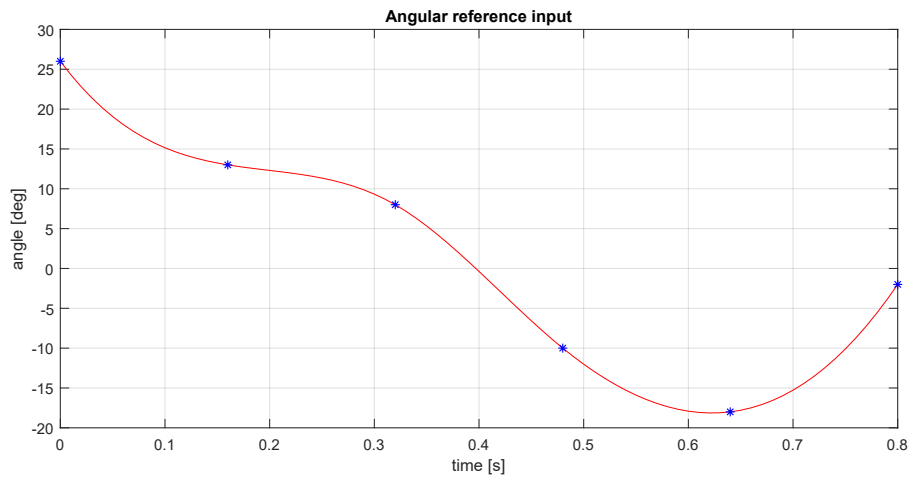


Figure 4.2: Angular input of hip motors

In fig. 4.3 it is possible to see robot walking in two different instant of time while in 4.3a is shown the robot after 1 second of simulation. It shows that anterior right and posterior left legs start to push backwards, after another half period, fig. 4.3b, anterior left and posterior right legs, doing the same. So after half period, legs are switched. For the scope of this thesis is sufficient that the robot is capable

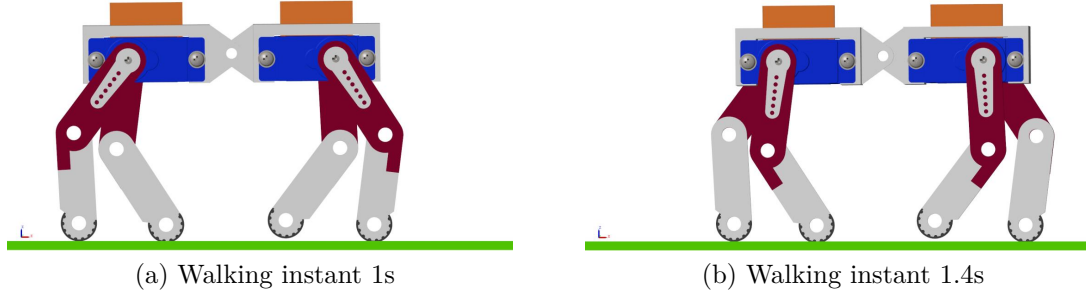


Figure 4.3: Example of walking in two different instant of time.

to walk on straight line. So to fulfil this requirement is not necessary to overload the system with a dedicated control system; however, is sufficient computed the best sequence of the angular position that allow the correct walk in a line. Six angular position showed behind in fig. 4.2, are computed to complete one stride, through the genetic algorithm. The only check of robot stride is done to stopping the simulation when the point A is arrived at 50 [cm] while a PID controller is put on each leg to act if there is some discrepancy between the reference angular input and the output. The parameters of the PID controller are computed through PID tune in Simulink. A general scheme of this controller is presented below.

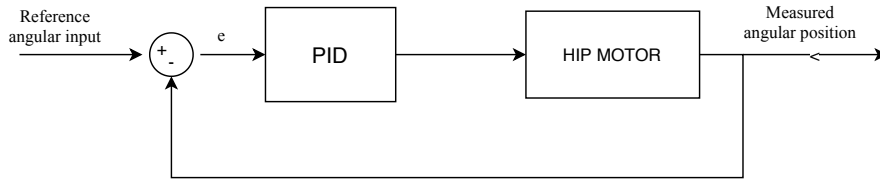


Figure 4.4: Hip motor control system

## 4.2 Simscape Model

The dimensions of this model are small, and the total weight is 63.59 [g]. In the tables 4.1 and 4.2 is possible to see the data of model, while in the figure 4.5, some images of the quadruped in simulation environment are shown.

Table 4.1: Data of model torso

	Torso				
	Length [m]	Wide [m]	Thickness [m]	Density [Kg/m <sup>3</sup> ]	Mass [Kg]
Torso	0.032	0.057	0.012	1120	0.0272
Torso1	0.032	0.057	0.012	1120	0.0272
Spine	0.01	-	-	-	-

Table 4.2: Data of legs

	Legs and foots			
	Length [m]	Radius [m]	Density [Kg/m <sup>3</sup> ]	Mass [Kg]
Upper leg	0.025	0.004	1120	0.0012
Lower leg	0.025	0.004	1120	0.000993
Foot	–	0.005	1700	7.8749e-05

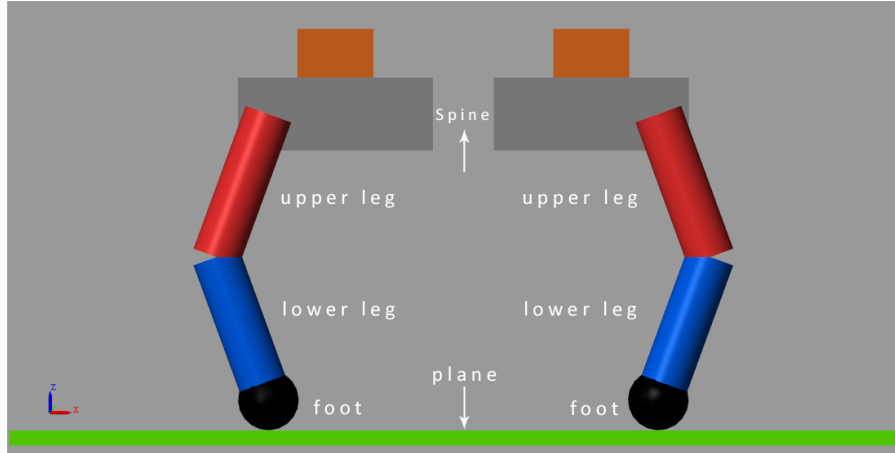


Figure 4.5: Simscape model robot side view

### 4.3 SolidWorks Model

The dimensions of this model are small, and the total weight is 63.5 [g]. In the tables 4.3 and 4.4 is possible to see the data of model, while in the figure 4.6, some images of the quadruped in simulation environment are shown.

Table 4.3: Data of model torso

	Torso				
	Length [m]	Wide [m]	Thicknness [m]	Density [Kg/m <sup>3</sup> ]	Mass [Kg]
Torso	0.035	0.034	0.016	1120	0.0271
Torso1	0.035	0.034	0.016	1120	0.0271
Spine	0.0012	0.002	-	-	-

Table 4.4: Data of legs

	Legs and foots			
	Length [m]	Thicknness [m]	Density [Kg/m <sup>3</sup> ]	Mass [Kg]
Upper leg	0.025	0.004	1120	0.0012
Lower leg	0.025	0.004	1120	0.000993
Foot	–	0.004	1700	7.8749e-05

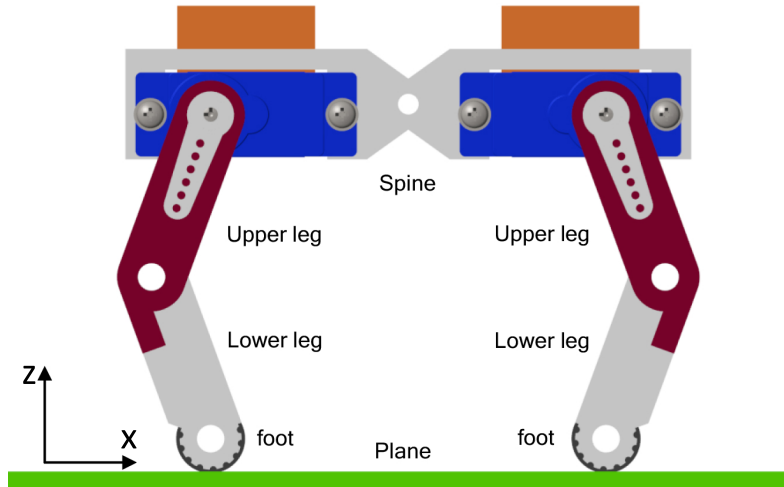


Figure 4.6: SolidWorks model robot side view



# Part III

## Results



## Chapter 5

# Simulation, results and discussion

### 5.1 Simulation

To test the performances of the robot, we carried out different trials. In this chapter, it is explained the process carried out to test the mechanic model of the quadruped. The measured performances are the time spent to travel from 0 cm to 50 cm of displacement of point A, the torque supplied by motors and the power spend. To understand the upcoming graphs is important to know the direction of  $x$ ,  $y$  and  $z$  axes. Moreover, in some graphs, the movement of point A is shown, the position of this point is showed in fig 3.2.

Five tests are done to analyze the response of the system at different stress. In the first test, in (section 5.1.1) was chosen to simulate the walking robot with a higher value of stiffness. In the second test (section 5.1.2), is shown the behaviour of the system when the spring legs had infinity stiffness while the spine spring is compliance. The third test (section 5.1.3) shows the response of the robot when all the springs are compliance. The other two tests show the behaviours of quadruped when the stiffness of the springs varies with a sinusoidal trend. So in the fourth test (section 5.1.4), the stiffness of the springs on the legs is maxima while the stiffness of spine spring varies with the sinusoidal trend. In the last test, the fifth (section 5.1.6), it is shown the opposite scene of the fourth test, in fact in this case the spine had maximum stiffness and the stiffness of the spring in the legs had a sinusoidal. Behaving these tests allowed us to underline the contribution of the spring behaviour a the four-legged robot motion. In this chapter are presented only the results obtained with the SolidWorks model, while appendix shown the results obtained from the same tests but with the Simscape model.



### 5.1.1 Test 1: higher stiffness in all joints legs and spine

For this test, the value of spring stiffness on legs and spine stiffness are chosen to have a minimal joint oscillation. The values are shown in table 5.1.

Table 5.1: Test 1 data

-	Stiffness	Damping
<i>Legs</i>	1 [N * m/deg]	1e-3 [N * m/deg/s]
<i>Spine</i>	100 [N * m/deg]	1 [N * mdeg/s]

In fig 5.1 the trend of point A during the gait is showed. In particular we can see that the time spent to travel from 0 [cm] to 50 [cm] is 8.8523 [s] the maximum deviation on  $y$  axis is 0.5621 [cm] and the trend on  $z$  axis is between 6.1w005 [cm] and 6.2776 [cm]. The fig 5.2 shows the angular displacement of spine. These oscillations are really small and goes from  $-3.82e - 4$  [deg] to  $2.59e - 5$  [deg]. In fig 5.3 the torque produced by anterior leg hip joints is plotted, while fig 5.4 shows the posterior legs torque. The maximum torque, in absolute value, is produced by anterior legs. Its value is 0.2212 [Nm] can see by fig 5.3, the maximum torque generated by hip joint of posteriors leg is 0.0771 [Nm]. For what concerns the power spent for walking, the max power is provided by the left anterior leg. Its value is 1.9381 [W] (see fig 5.5), while the maximum value of posterior leg power is 0.8560 [W] (fig 5.6).

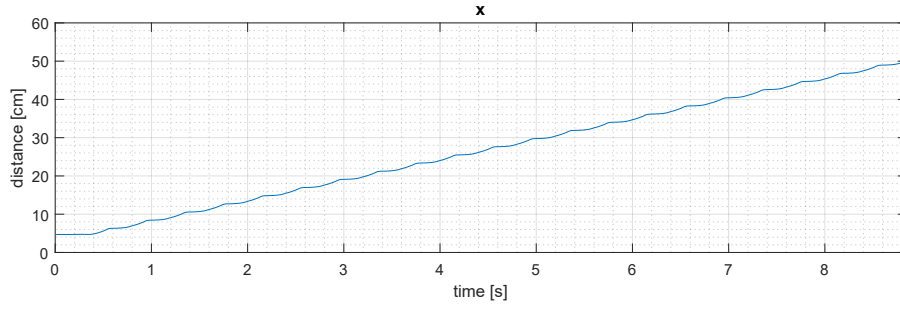
### 5.1.2 Test 2: high stiffness for legs - spine compliant

The value of legs spring stiffness and spine, for this test, are showed in the table 5.2: The outcome of this second test, for the SolidWorks model, are outlined here.

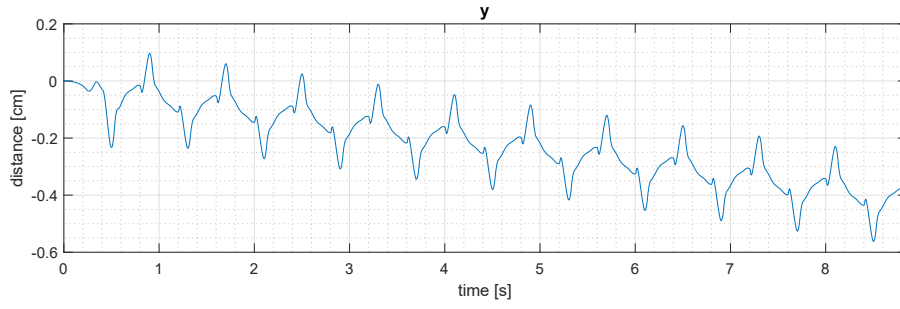
Table 5.2: Test 2 data

-	Stiffness	Damping
<i>Legs</i>	1 [N * m/deg]	1e-3 [N * m/deg/s]
<i>Spine</i>	5e-3 [N * m/deg]	1e-5 [N * mdeg/s]

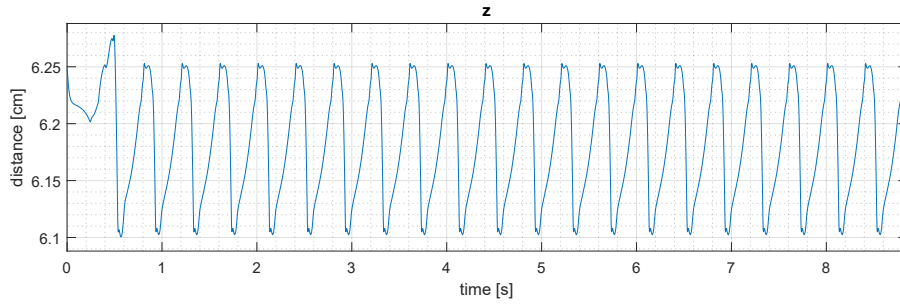
Fig 5.7 shows the trend of point A during the gait. In particular, we can see that the time to travel from 0 [cm] to 50 [cm] is 8.6448 [s] the maximum deviation in module on  $y$  axis is 1.1627 [cm] and the trend on  $z$  axis is among 6.0846 [cm] and 6.2485 [cm]. The fig 5.8 present the displacement of spine. These oscillations are bigger than the ones in test 1 and go from  $-8.2228$  [deg] to 0 [deg]. The maximum torque is produced by the Left anterior leg and its trend is shown in figure 5.9 its



(a) Point A movement - x axis



(b) Point A movement - y axis



(c) Point A movement - z axis

Figure 5.1: Evolution of point A during gait in test 1

maximum value is 0.2110 [Nm]. In this test the maximum value of power spent for walking is provided by right posterior leg. Its value is 1.2829 [W].

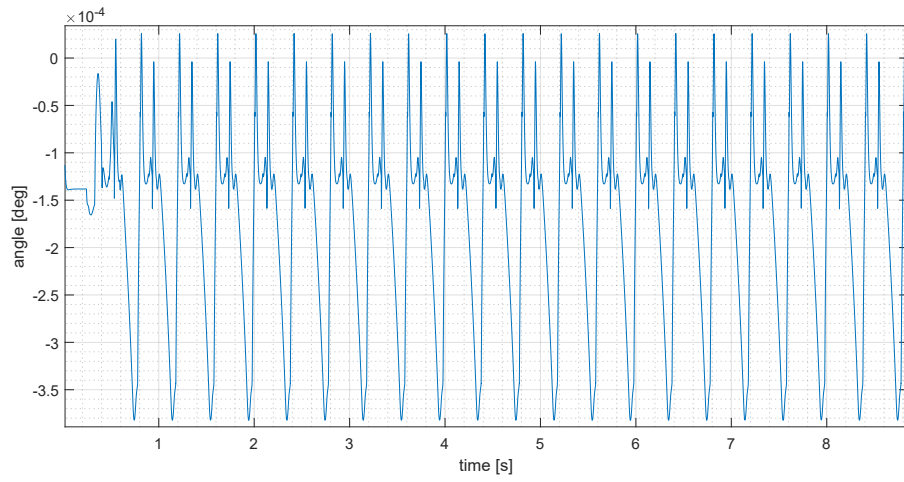


Figure 5.2: Spine oscillation, test 1 (x axis)

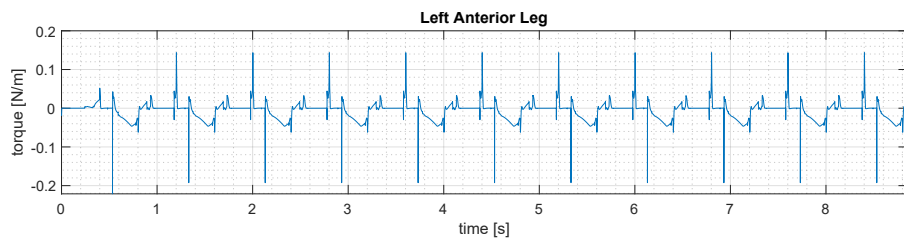


Figure 5.3: Legs anterior torque, test 1

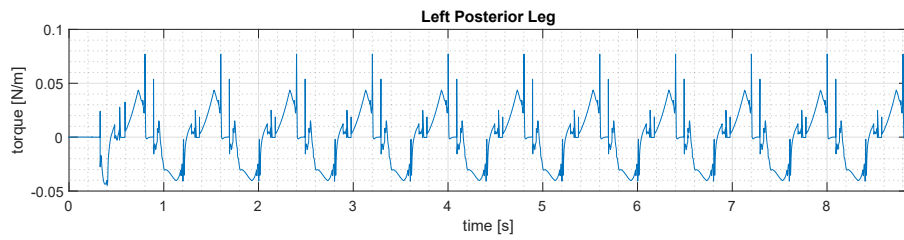


Figure 5.4: Legs posterior torque, test 1

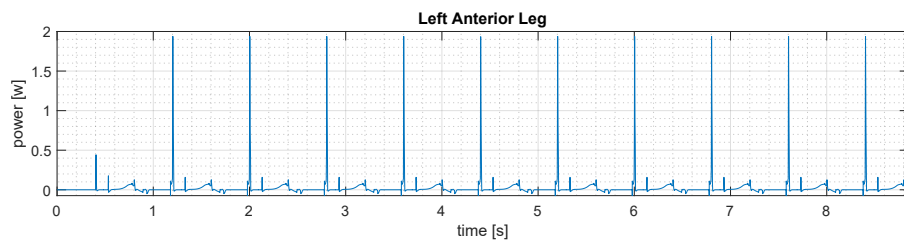


Figure 5.5: Legs anterior power, test 1

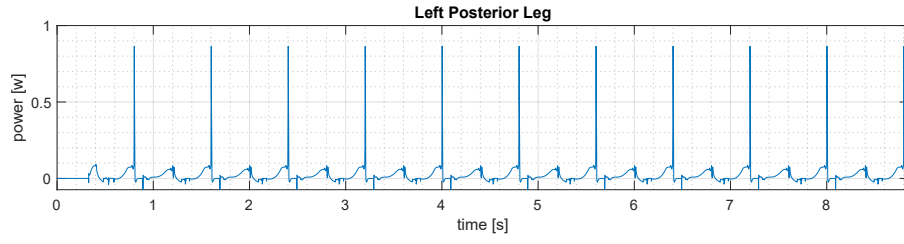
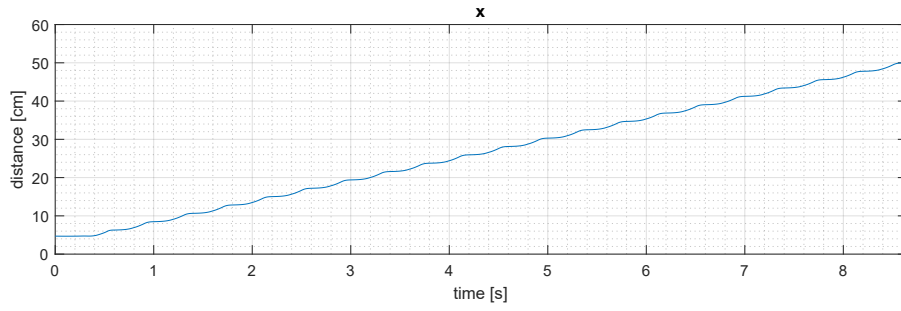
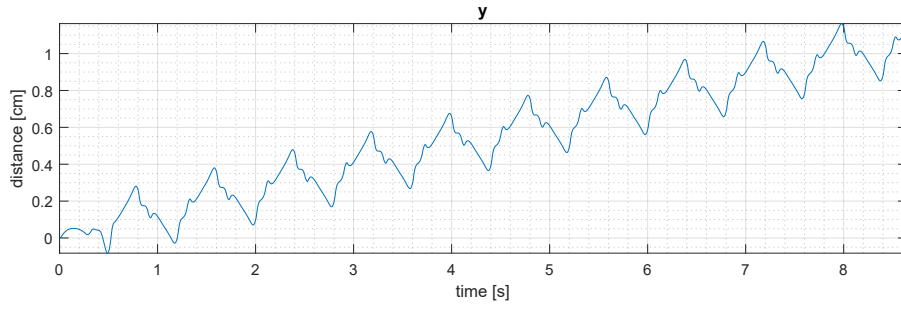


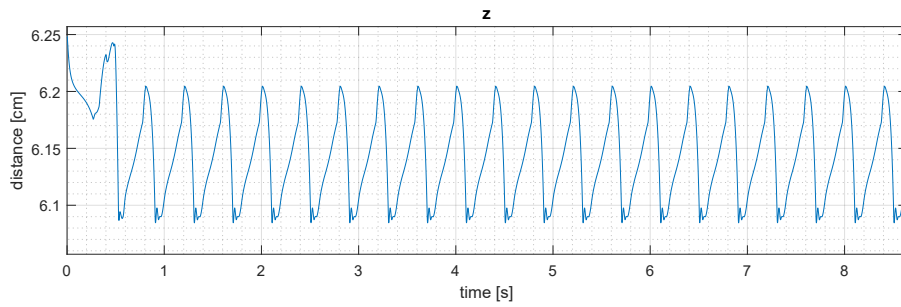
Figure 5.6: Legs posterior power, test 1



(a) Point A movement - x axis



(b) Point A movement - y axis



(c) Point A movement - z axis

Figure 5.7: Evolution of point A during gait, test 2

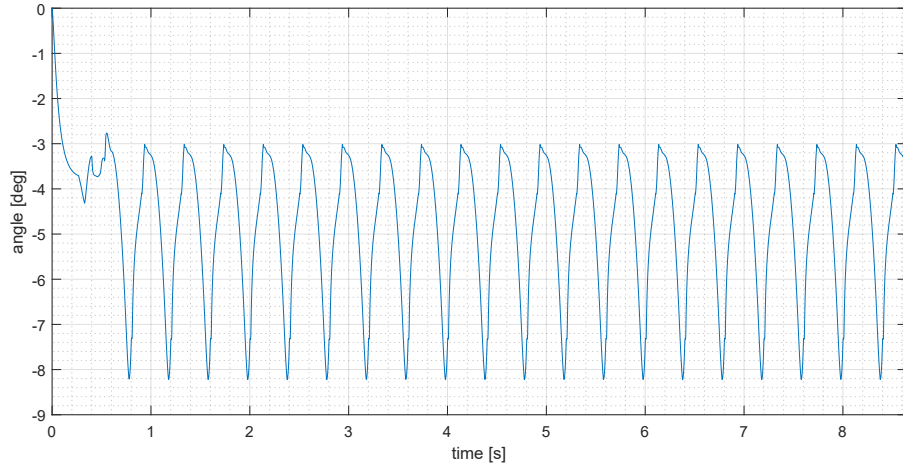


Figure 5.8: Spine oscillation, test 2 (x axis)

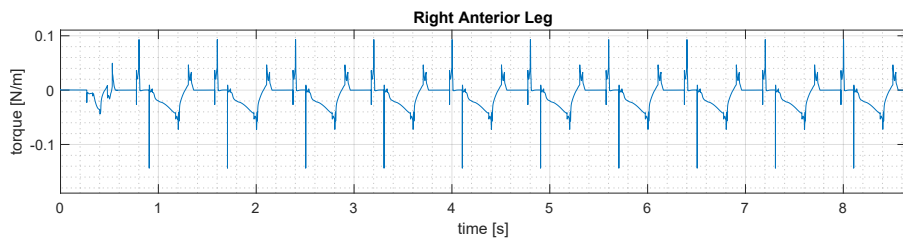


Figure 5.9: Legs torque, test 2 (right anterior)

### 5.1.3 Test 3: Compliance in the legs - spine with high stiffness

The third simulation is the opposite of the test 5.1.2. In table 5.3 below, the values of spring and spine stiffness are showed. For the SolidWorks model, the results of

Table 5.3: Test 3 data

-	Stiffness	Damping
<i>Legs</i>	0.0040 [N * m/deg]	5e-4 [N * m/deg/s]
<i>Spine</i>	100 [N * m/deg]	1 [N * m/deg/s]

this test are illustrated in this subsection. In fig 5.10, like in previous sections, the trend of point A during the gait is shown. We can see that the time to travel from 0 [cm] to 50 [cm] is 15.3855 [s] the maximum deviation on  $y$  axis is 3.3962 [cm] and the trend on  $z$  axis is between 6.0480 [cm] and 6.3150 [cm]. Fig 5.11 present the displacement of spine. These oscillation are really small and go from  $-2.1677 * 10^{-4}$  [deg] to  $1.5007 * 10^{-4}$  [deg]. In fig 5.12 the torque produced by the Right Anterior leg is showed, its value is 0.2347 [Nm]. The max power in absolute value is provided by Right Anterior leg. It's value is 2.5362 [W].

### 5.1.4 Test 4: legs with high stiffness - spine sinusoidal

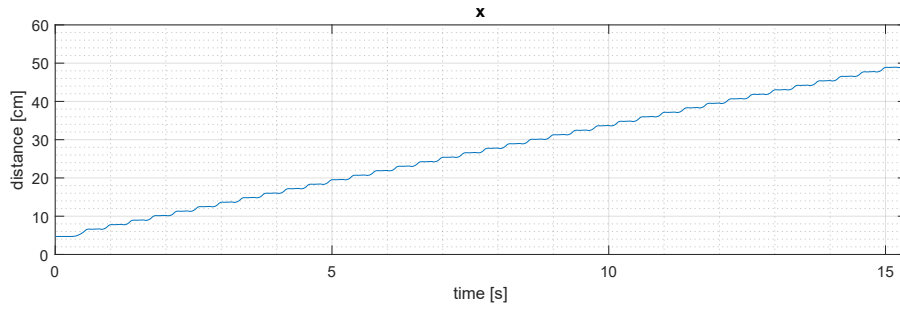
The values of this test are shown in table 5.4 below:

Table 5.4: Test 4 data

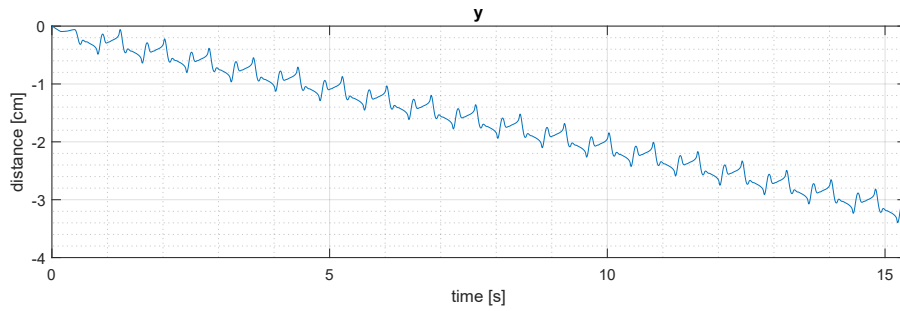
-	Stiffness	Damping
<i>Legs</i>	1 [N * m/deg]	1e-3 [N * m/deg/s]
<i>Spine</i>	4.9e-3 to 0.5 [N * m/deg]	1e-5 [N * m/deg/s]

### 5.1.5 Model 2

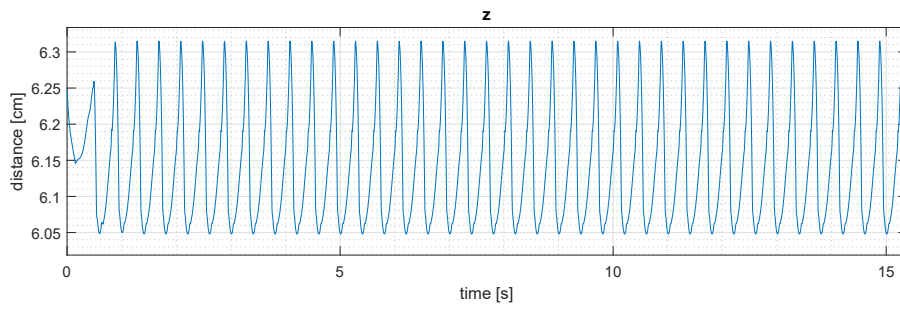
The results of SolidWorks model are reported in this subsection. Figure 5.13 shows the trend of point A during the gait. In particular we can see that the time spent to travel from 0 [cm] to 50 [cm] is 8.8504 [s], the maximum deviation on  $y$  axis is 0.5581 [cm], and the trend on  $z$  axis is between 6.1007 [cm] and 6.2780 [cm]. Fig 5.14 present the displacement of spine in degrees. These oscillation are small and go from  $-0.1323$  [deg] to  $0.0$  [deg]. In fig 5.15 the maximum torque produced by robot's hip joints is plotted. The maximum value is provided by the right anterior leg, and it is 0.2209 [Nm]. The maximum power spent for walking is provided by right anterior leg. Its value is 1.9280 [W].



(a) Point A movement - x axis



(b) Point A movement - y axis



(c) Point A movement - z axis

Figure 5.10: Evolution of point A during gait, test 3

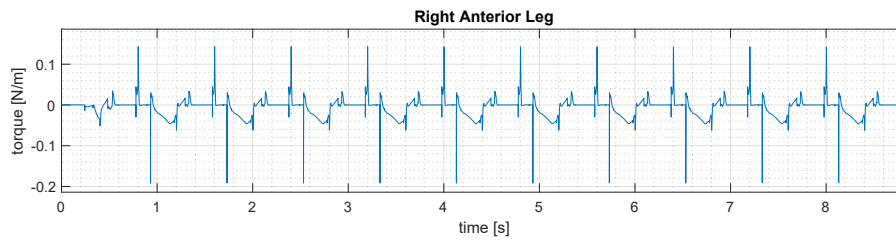


Figure 5.15: Legs torque, test 4 (Right anterior)

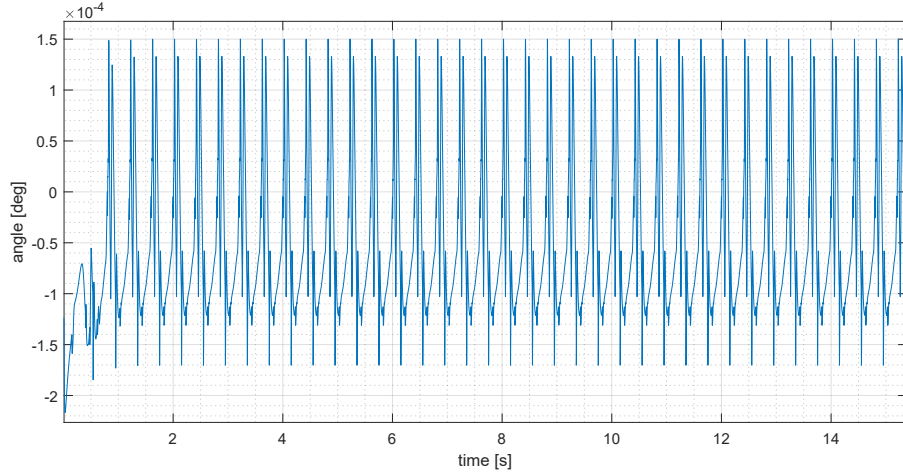


Figure 5.11: Spine oscillation, test 3 (x axis)

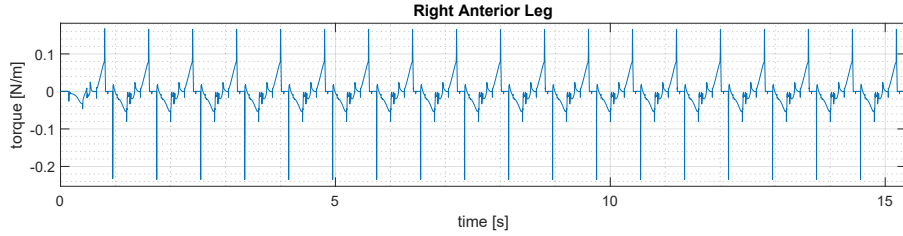


Figure 5.12: Legs torque, test 3 (Right anterior)

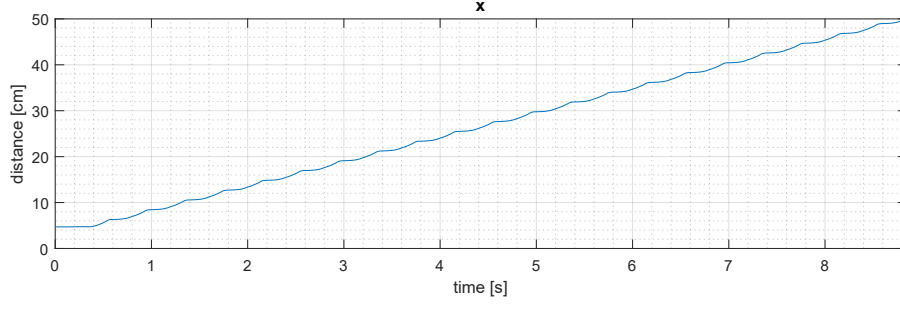
### 5.1.6 Test 5: legs sinusoidal behaviour - spine maximum stiffness

The fifth test is the opposite of test four 5.1.4 in the tab 5.5 we can see all values: Fig 5.16 shows the trend of point A during the gait. The time taken to travel from 0 [cm] to 50 [cm] is 8.8507 [s], the maximum deviation on  $y$  axis is 0.5591 [cm], and the trend on  $z$  axis is between 6.1007 [cm] and 6.2780 [cm]. Figure 5.17 presents the displacement of spine in degrees. These oscillation are small and go from  $-0.0450$  [deg] to 0 [deg]. Fig 5.18 the maximum torque, produced by left anterior leg is plotted its, value is 0.2209 [Nm]. The maximum power spent for walking is provided by the right anterior leg. Its value is 1.9737 [W].

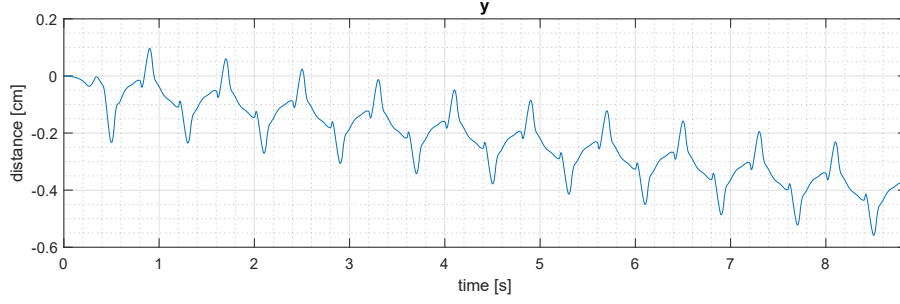
## 5.2 Discussion

Based on the results obtained in this section, a general overview and some considerations on the results are presented. As can be seen from the graphs shown in the previous sections and from the resulting data, the behaviour of the system varies slightly in the five tests. In the table below 5.6, you can see a summary of the data

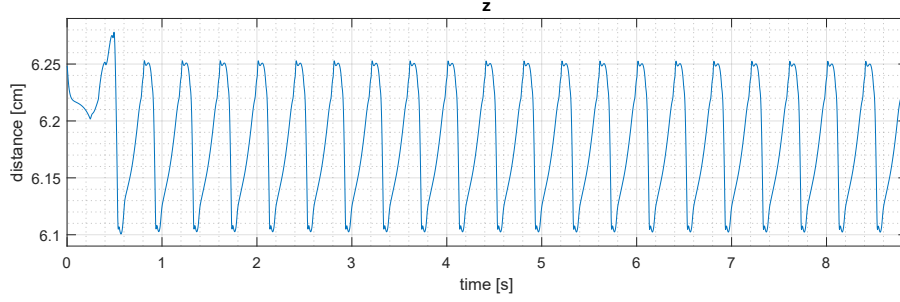




(a) Point A movement - x axis



(b) Point A movement - y axis



(c) Point A movement - z axis

Figure 5.13: Evolution of point A during gait, test 4

Table 5.5: Test 5 data

-	Stiffness	Damping
<i>Legs</i>	0.004 to 1 [N * m/deg]	5e-4 [N * m/deg/s]
<i>Spine</i>	100 [N * m/deg]	1 [N * m/deg/s]

collected. From table 5.6 is clear that test 1, 2, 4 and 5 have at least the same behaviour with about the same forward speed and the same maximum torque. While the performance in test 3 are a little bit different, for example it is possible to see that in this test the system is approximately twice slower than the other and the

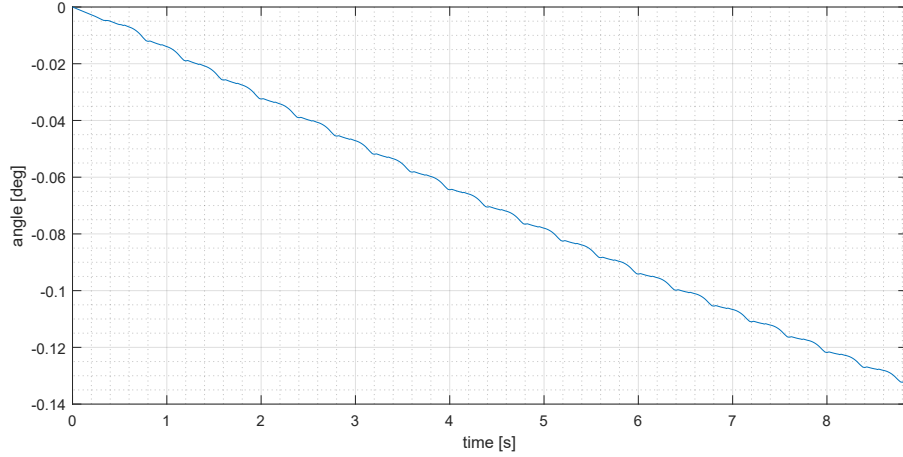
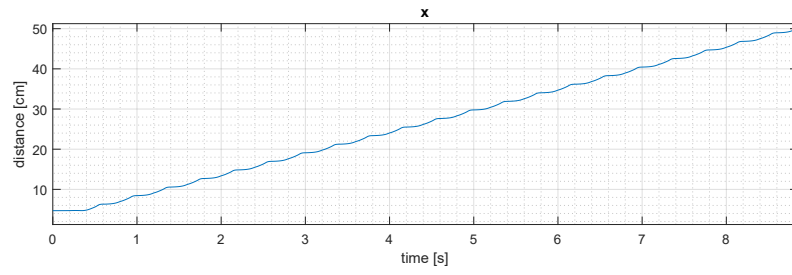


Figure 5.14: Spine oscillation, test 4 (x axis)

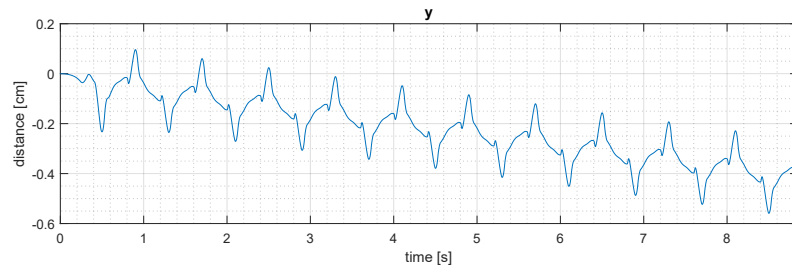
Table 5.6: Resume data

Test	1	2	3	4	5
Simulation Time	8.8523 [s]	8.6448 [s]	15.3855 [s]	8.8504 [s]	8.8477 [s]
Anterior torque	0.2212 [Nm]	0.2110 [Nm]	0.2347 [Nm]	0.2209 [Nm]	0.2200 [Nm]
Posterior torque	0.0771 [Nm]	0.1061 [Nm]	0.1303 [Nm]	0.0773 [Nm]	0.0804 [Nm]
Anterior power	1.9381 [W]	1.1994 [W]	2.5362 [W]	1.9280 [W]	1.7176 [W]
Posterior power	0.8560 [W]	1.2829 [W]	1.0380 [W]	0.8688 [W]	0.8803 [W]
Average speed	5.6482 [cm/s]	5.7838 [cm/s]	3.2498 [cm/s]	5.6495 [cm/s]	5.6512 [cm/s]

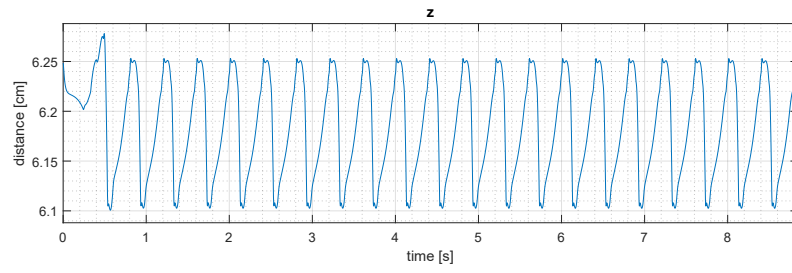
maximum torque provided is higher than the other. It is important to notice that in all the tests, there is an important difference in torque generated by anterior motors respect posteriors. For example, if we take test 1 and compare the graph of one anterior and posterior legs, it is clear that the peak torque generated by the first one is twice the peak torque of the second one as is it possible to see in the figure, 5.3 and 5.4. Observing the data it is possible to see that the worst case is represented by test 3, where we find an increment of all measured data. We can see that, respect the average of other tests, the increment of time is of 42%, the increased torque is about 7% on the posterior legs and with the average values in anterior legs. In the graphs 5.19, six lines with different slope are presented. These describe the different performance in terms of forwarding speed in the 5 tests presented in the paper. Three of these lines are overlapped in fact table 5.6 shows that performance of test 5, 4 and 1 are comparable. Test 2 with stiff legs and compliance spine present a slightly higher performance, this because the introduction of compliance spine assists the walking during lift off and the landing of the foot with the effect to increment the forwarding speed and to provide a better distribution of torque and power.



(a) Point A movement - x axis



(b) Point A movement - y axis



(c) Point A movement - z axis

Figure 5.16: Evolution of point A during gait, test 5

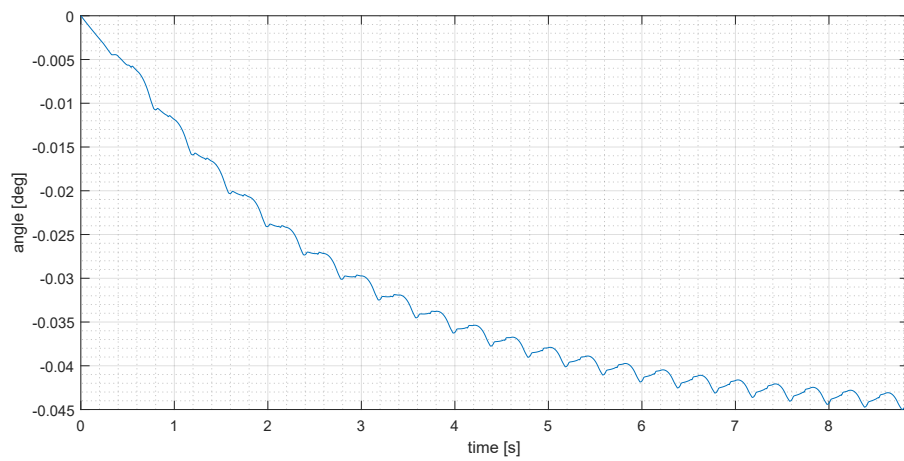


Figure 5.17: Spine oscillation, test 5 (x axis)

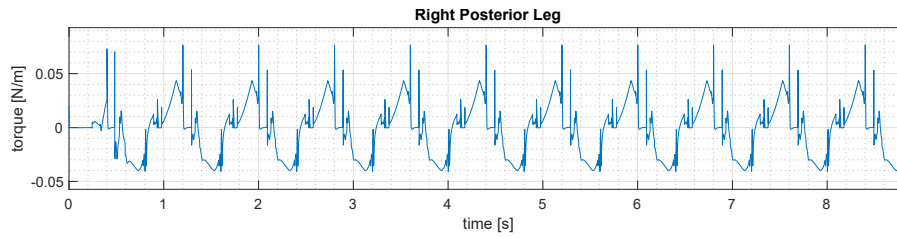


Figure 5.18: Legs torque, test 5 (Right posterior)

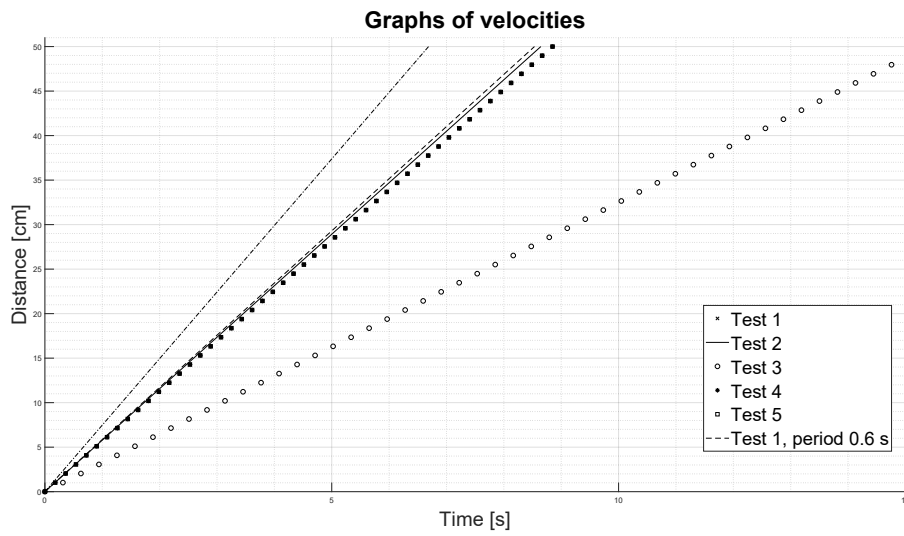


Figure 5.19: Trend of velocities for each test

### 5.3 Conclusion

In this work, was presented the design and simulation, in a 3D environment, of a simple four-legged robot modelled with Solidworks and Matlab. The quadruped holds passive knees and spine made by passive joints provided by rotational spring. A servo motor with one DOF actuates each leg. It is showed how a basic quadruped robot could perform a simple action like walking with a simple control law. The effects of spring in legs and spine allow modifying the behaviour of the system with a reduction of the energy required for the movement and increase the performance of the entire system like forwarding speed for the same energy.

*In the future* it will be useful to try the same test with a real model, to see if the overall performance is comparable. It is possible to build a real model printed in 3D to compare the response of the real model with the simulated one. Finally, the implementation of different gaits such as trot and gallop could be realized, and a control system could be introduced to ensure a smooth transition between the different strikes.

# Appendix A

## Script

Listing A.1: Model parameters of quadruped robot, for Simscape model

```
1 %% model_parameters
2 %% PARAMETERS of Robot
3
4 motion_time_constant = 0.01;
5
6 %% Trunk dimension
7 trunk_L = 3.2;           % [cm]
8 trunk_W = 2.85*2;       % [cm]
9 trunk_T = 1.2;          % [cm]
10 density = 1.12;         % [g/cc] - Density of ABS
11 spine_length = 1;       % [cm]
12
13 % leg dimension
14 upper_leg_L = 2.5;       % [cm]
15 lower_leg_L = 2.5;       % [cm]
16 leg_R = 0.4;            % [cm]
17 foot_radius = leg_R+0.1; % [cm]
18 foot_density = 1.7;      % Rubber - avoid slipping
19
20 %% Contact/friction parameters
21 mu_k = 0.6;              % [N/m] - Coefficient of Kinetic ...
    Friction
22 mu_s = 0.8;              % [N/(m/s)] - Coefficient of ...
    Static Friction
23 mu_vth = 0.1;           % [m/s] - Velocity Threshold
24
25 contact_stiffness = 5e3;  % [N/m] - Contact Stiffness
26 contact_damping = 1e2;   % [N/(m/s)] - Contact Damping
27 sphere_contact = foot_radius/100+1e-6;
28
29 %% plane position
```

```
30 height_plane = 0.25e-2;      % [m]
31 plane_x = 1.2;               % [m]
32 plane_y = 30e-2;            % [m]
33 pos_plane2 = upper_leg_L*cosd(beta) + ...
    lower_leg_L*cosd(-40+beta) + foot_radius +height_plane*100 ...
    +1.6/2;  % [cm]
```

Listing A.2: Listed for starting the simulation of Simscape model

```
1  %%% Script for test 1
2  %% all systems is rigid.
3  clear all; close all; path(pathdef); clc;
4  HomeDir = pwd;
5  cd ../
6  ModelDir = pwd;
7  addpath(pwd)
8  addpath([pwd filesep 'SM_Contact_Forces_Lib_R18a_v4p1' filesep ...
    'Libraries']);
9  addpath([pwd filesep 'assieme_motori_solidworks']);
10 cd([pwd filesep 'SM_Contact_Forces_Lib_R18a_v4p1' filesep ...
    'Libraries']);
11 open('Contact_Forces_Lib')
12 cd([ModelDir filesep 'assieme_motori_solidworks']);
13 nuovoAssieme_motori_DataFile;
14 cd([HomeDir]);
15
16 model_parameters;
17
18 spring_stiffness = 1;
19 spring_stiffness_ant = spring_stiffness;
20 spring_damping = 1e-3;
21 spine_stiffness = 100;
22 spine_damping = 1;
23
24
25 %%
26 load('optimizedData_14Sep19_0344.mat', 'hip_motion');
27 hip_stop = deg2rad(zeros(length(hip_motion),1)); ...
    % stop
28
29 period = 0.8;
30 time = linspace(0,period,length(hip_motion));
31 t = 0:0.01:period;
32 smooth_hip_motion = interp1(time,hip_motion,t,'spline');
33 smooth_hip_stop = interp1(time,hip_stop,t,'spline');
34
35 %%
36 out_spring = sim('Simscape_model', 'StopTime', '15');
37 % out_spring = sim('solidWorks_model', 'StopTime', '15');
```

```

38
39 %%
40 DataPlot

```

Listing A.3: Script to print the data calculated from the model.

```

1  %% Print results
2  time = out_spring.simulation_time;
3  CM_x = out_spring.CM_position.signals(1).values;
4  CM_y = out_spring.CM_position.signals(2).values;
5  CM_z = out_spring.CM_position.signals(3).values;
6
7  time_end = time(end)
8  %% Movemete of point A, plot
9  figure('Name','Movement of point A','NumberTitle','off'),
10 subplot(3,1,1), plot(time, CM_x), axis([0 time(end) 0 51]), ...
    axis('auto y')
11 grid on, grid MINOR,
12 xlabel('time [s]'), ylabel('distance [cm]'), title x
13 subplot(3,1,2), plot(time, CM_y), axis([0 time(end) -0.34 ...
    0.51]), axis('auto y')
14 grid on, grid MINOR,
15 xlabel('time [s]'), ylabel('distance [cm]'), title y
16 subplot(3,1,3), plot(time, CM_z), axis([0 time(end) 4.91 ...
    5.32]),axis('auto y')
17 grid on, grid MINOR,
18 xlabel('time [s]'), ylabel('distance [cm]'), title z
19 sgtitle('Movement of point A along three axes', 'Color','red')
20
21 y_max = max(abs(CM_y))
22 z_min = min(CM_z)
23 z_max = max(CM_z)
24
25 %% Spine oscillation
26 spine_osc = out_spring.spine_movement.signals.values ;
27
28 figure('Name','Spine oscillation','NumberTitle','off'),
29 plot(time, spine_osc), axis([0 time(end) -0.0565 0.028]), ...
    axis('auto y')
30 grid on, grid MINOR,
31 xlabel('time [s]'), ylabel('angle [deg]')
32
33 s_min = min(spine_osc), s_max = max(spine_osc)
34
35 %% Sphere to plane Data
36 % normal Force RAL
37 NFral = out_spring.sphere2planeDataRAL.signals(1).values;
38 figure('Name','Normal force - anterior right ...
    leg','NumberTitle','off'),

```



```

39 plot(time,NFral), axis([0 time(end) -0.2 14.5]), axis('auto y')
40 grid on, grid MINOR,
41
42 % foot position RAL
43
44 RALfootX = ...
    out_spring.sphere2planeDataRAL.signals(3).values(:,1)*100;
45 RALfootY = ...
    out_spring.sphere2planeDataRAL.signals(3).values(:,2)*100;
46 RALfootZ = out_spring.sphere2planeDataRAL.signals(3)....
47 values(:,3)*100-0.61128;
48
49 figure('Name','Rigth anterior foot position','NumberTitle','off'),
50 subplot(3,1,1), plot(time, RALfootX), axis([0 time(end) -0.11 ...
    0.22]), axis('auto y')
51 grid on, grid MINOR,
52 title ('X position')
53 subplot(3,1,2), plot(time, RALfootY), axis([0 time(end) -0.2 ...
    0.12]), axis('auto y')
54 grid on, grid MINOR,
55 ylabel('Position [cm]'), title('Y position')
56 subplot(3,1,3), plot(time, RALfootZ), axis([0 time(end) -0.11 ...
    0.22]),axis('auto y')
57 grid on, grid MINOR, xlabel('time [s]'), title('Z position')
58
59 figure,
60 plot(time, RALfootZ), hold on, plot(time,NFral),axis([0 ...
    time(end) -0.2 14.5]), axis('auto y')
61 grid on, grid MINOR, title('RAL Normal ground reaction VS foot ...
    movment')
62
63
64 % normal Force LPL
65 NF1pl = out_spring.sphere2planeDataLPL.signals(1).values;
66 figure('Name', 'Normal force - left posterior ...
    leg','NumberTitle','off'),
67 plot(time,NF1pl), axis([0 time(end) -0.2 14.5]), axis('auto y')
68 grid on, grid MINOR,
69
70 % foot position LPL
71
72 LPLfootX = ...
    out_spring.sphere2planeDataLPL.signals(3).values(:,1)*100;
73 LPLfootY = ...
    out_spring.sphere2planeDataLPL.signals(3).values(:,2)*100;
74 LPLfootZ = out_spring.sphere2planeDataLPL.signals(3)....
75 values(:,3)*100-0.61128;
76
77 figure('Name','Left posterior foot position','NumberTitle','off'),

```

```

78 subplot(3,1,1), plot(time, LPLfootX), axis([0 time(end) -0.11 ...
    0.22]), axis('auto y')
79 grid on, grid MINOR,
80 title('X position')
81 subplot(3,1,2), plot(time, LPLfootY), axis([0 time(end) -0.2 ...
    0.12]), axis('auto y')
82 grid on, grid MINOR,
83 ylabel('Position [cm]'), title('Y position')
84 subplot(3,1,3), plot(time, LPLfootZ), axis([0 time(end) -0.11 ...
    0.22]), axis('auto y')
85 grid on, grid MINOR, xlabel('time [s]'), title('Z position')
86
87 figure,
88 plot(time, LPLfootZ), hold on, plot(time, NFlpl), axis([0 ...
    time(end) -0.2 14.5]), axis('auto y')
89 grid on, grid MINOR, title('LPL Normal ground reaction VS foot ...
    movment')
90
91
92 %% Torque plot
93 LPL = out_spring.leg_hip_torque.signals.values(:,1);
94 LAL = out_spring.leg_hip_torque.signals.values(:,2);
95 RPL = out_spring.leg_hip_torque.signals.values(:,3);
96 RAL = out_spring.leg_hip_torque.signals.values(:,4);
97
98 figure('Name','Legs torque','NumberTitle','off'),
99 subplot(4,1,1), plot(time, LPL), axis([0 time(end) -0.11 ...
    0.22]), axis('auto y')
100 grid on, grid MINOR,
101 xlabel('time [s]'), ylabel('torque [N/m]'), title('Left ...
    Posterior Leg')
102 subplot(4,1,2), plot(time, LAL), axis([0 time(end) -0.2 0.12]), ...
    axis('auto y')
103 grid on, grid MINOR,
104 xlabel('time [s]'), ylabel('torque [N/m]'), title('Left ...
    Anterior Leg')
105 subplot(4,1,3), plot(time, RPL), axis([0 time(end) -0.11 ...
    0.22]), axis('auto y')
106 grid on, grid MINOR,
107 xlabel('time [s]'), ylabel('torque [N/m]'), title('Right ...
    Posterior Leg')
108 subplot(4,1,4), plot(time, RAL), axis([0 time(end) -0.2 ...
    0.12]), axis('auto y')
109 grid on, grid MINOR,
110 xlabel('time [s]'), ylabel('torque [N/m]'), title('Right ...
    Anterior Leg')
111 sgtitle('Legs torque', 'Color','red')
112
113 LPL_max = max(abs((LPL)))
114 LAL_max = max(abs((LAL)))

```

```

115 RPL_max = max(abs((RPL)))
116 RAL_max = max(abs((RAL)))
117
118 %% Power plot
119 LPLw = out_spring.hip_out_power.signals.values(:,1);
120 LALw = out_spring.hip_out_power.signals.values(:,2);
121 RPLw = out_spring.hip_out_power.signals.values(:,3);
122 RALw = out_spring.hip_out_power.signals.values(:,4);
123
124 figure('Name','Legs power','NumberTitle','off'),
125 subplot(4,1,1), plot(time, LPLw), axis([0 time(end) -0.42 ...
    0.67]), axis('auto y')
126 grid on, grid MINOR,
127 xlabel('time [s]'), ylabel('power [w]'), title('Left Posterior ...
    Leg')
128 subplot(4,1,2), plot(time, LALw), axis([0 time(end) -1.3 ...
    1.65]), axis('auto y')
129 grid on, grid MINOR,
130 xlabel('time [s]'), ylabel('power [w]'), title('Left Anterior Leg')
131 subplot(4,1,3), plot(time, RPLw), axis([0 time(end) -0.5 ...
    1.75]),axis('auto y')
132 grid on, grid MINOR,
133 xlabel('time [s]'), ylabel('power [w]'), title('Right Posterior ...
    Leg')
134 subplot(4,1,4), plot(time, RALw), axis([0 time(end) -1.3 ...
    1.4]), axis('auto y')
135 grid on, grid MINOR,
136 xlabel('time [s]'), ylabel('power [w]'), title('Right Anterior ...
    Leg')
137 sgtitle('Legs power', 'Color','red')
138
139 LPLw_max = max(abs((LPLw)))
140 LALw_max = max(abs((LALw)))
141 RPLw_max = max(abs((RPLw)))
142 RALw_max = max(abs((RALw)))

```

Listing A.4: Cost function.

```

1 function penalty = simulateWalkingRobot(params,mdlName, ...
    scaleFactor, period)
2 % Load parameters into function workspace
3 model_parameters;
4 % Apply variable scaling
5 params = scaleFactor*params;
6 % Extract simulation inptus from parameters
7 hip_motion = deg2rad(params);
8 time = linspace(0,period, length(hip_motion));
9 tt = 0:0.01:period;
10 smooth_hip_motion = interp1(time, hip_motion, tt, 'spline');

```

```
11 out_spring = ...
    sim mdlName, 'StopTime', '10', 'SrcWorkspace', 'current', ...
12 'FastRestart', 'on');
13 % Unpack logged data
14 wAvg = mean(out_spring.wz);
15 xEnd = out_spring.CM_position.signals(1).values(end);
16 tEnd = out_spring.simulation_time(end);
17 % Calculate penalty from logged data
18 %   Distance traveled without falling
19 %   (endingsimulation early) increases reward
20 positiveReward = sign(xEnd)*xEnd^2*tEnd;
21 %   Angular velocity and trajectory aggressiveness
22 %   (number of times the derivative flips signs) decreases reward
23 %   NOTE: Set lower limits to prevent divisions by zero
24 aggressiveness = 0;
25 diffs = [diff(smooth_hip_motion)];
26 for idx = 1:numel(diffs)-1
27     if (sign(diffs(idx)/diffs(idx+1))<0) && mod(idx,numel(params))
28         aggressiveness = aggressiveness + 1;
29     end
30 end
31 negativeReward = max(abs(wAvg),0.01) * max(aggressiveness,1);
32 %   Negative sign needed since the optimization minimizes cost ...
    function
33 penalty = -positiveReward/negativeReward;
34 end
```



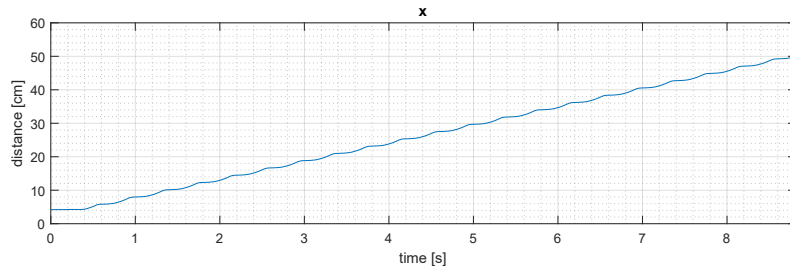
## Appendix B

# Simscape simulation results

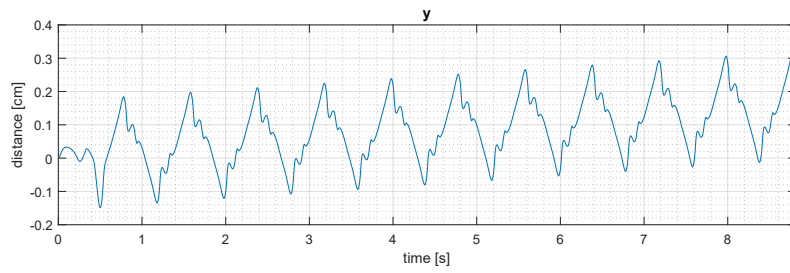
In this appendix are shown the results of the simulation, presented in section 4.2, but obtained with the Simscape model presented in 4.2.

### B.1 Test 1: higher stiffness in all joints legs and spine

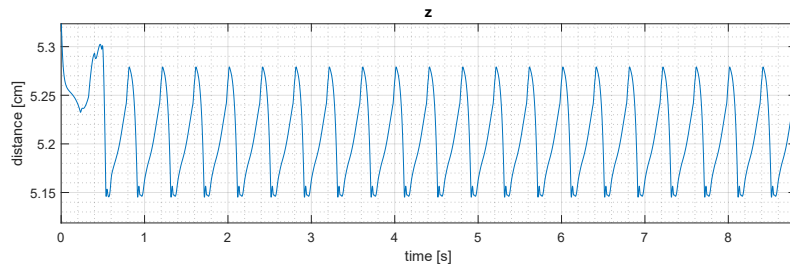
The figure B.1 shows the trend of point A of the SimScape model during the gait. In particular we can see that the time taken to travel from 0 [cm] to 50 [cm] is 8.9518 [s] the maximum deviation on the  $y$  axis is 0.6777 [cm] and the trend on the  $z$  axis is between 5.1509 [cm] and 5.3235 [cm]. In figure B.3 the torque produced by the front hip joints of the legs is plotted while in figure B.4 the torque of the back legs is plotted. The maximum torque, in absolute value, is produced by the front legs. Its value is 0.1997 [Nm]. As you can see from the graphs B.3, the maximum torque generated by the posterior leg hip joint is 0.0822 [Nm]. As for the power spent on walking, the maximum power is provided by the right front leg. Its value is 2.0806 [W], see figure B.5, while the maximum value of the rear leg power is 0.8266 [W] fig. B.6. The latest graph B.2 shows the angular displacement of the spine. These oscillations are very small and range from  $-3.8478 * 10^{-4}$  [deg].



(a) Point A movement - x axis



(b) Point A movement - y axis



(c) Point A movement - z axis

Figure B.1: Evolution of point A during gait in Simscape model, test 1

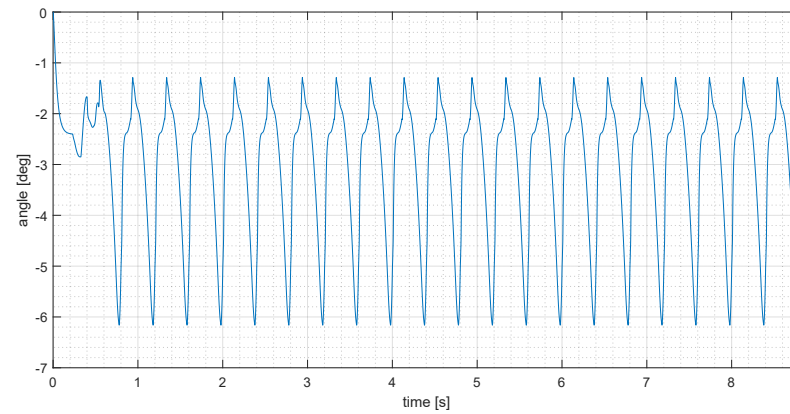


Figure B.2: Spine oscillation of Simscape model, test 1

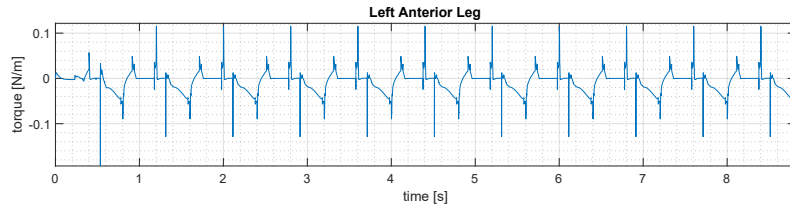


Figure B.3: Hip joints anterior torque of Simscape model, test 1

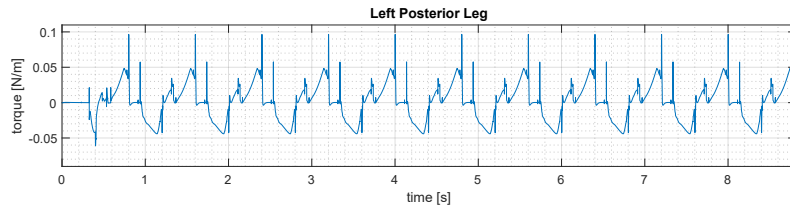


Figure B.4: Hip joints posterior torque of Simscape model, test 1

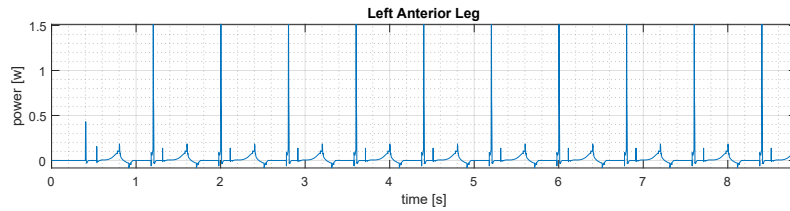


Figure B.5: Legs anterior power of Simscape model, test 1

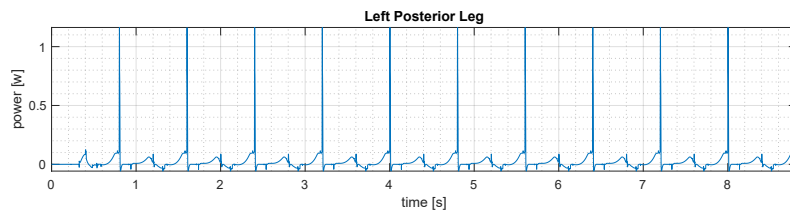
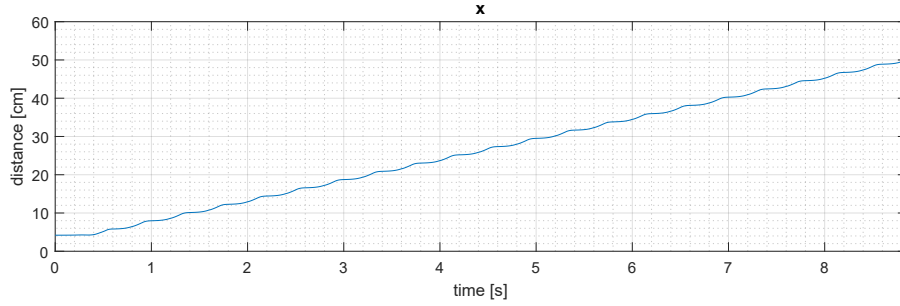


Figure B.6: Legs power of Simscape model, test 1

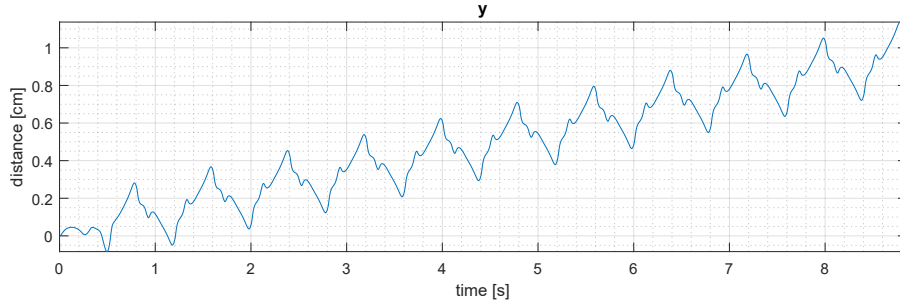


## B.2 Test 2: high stiffness for legs - spine compliant

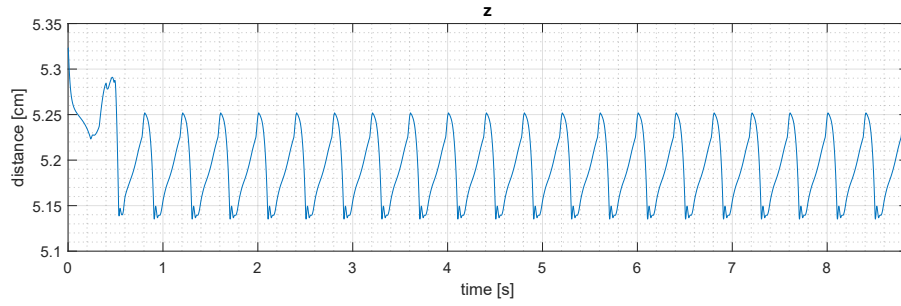
The outcome of this second test, for the Simscape model, are outlined here. Figure B.7 shows the trend of point A during the gait. In particular we can see that the time to travel from 0 [cm] to 50 [cm] is 8.8543 [s] the maximum deviation in module on  $y$  axis is 1.1370 cm and the trend on  $z$  axis is among 5.1353 [cm] and 5.3235 [cm]. The maximum torque is produced by the Left anterior leg and its trend is shown in figure B.9 its maximum value is 0.1931 [Nm], while the maximum value of power spent for walking is provided by right posterior leg. Its value is 1.2628 [W]. The last graph B.8 present the displacement of spine. These oscillations are bigger than ones in test 1 and go from  $-8.5739$  [deg] to 0 [deg].



(a) Point A movement - x axis

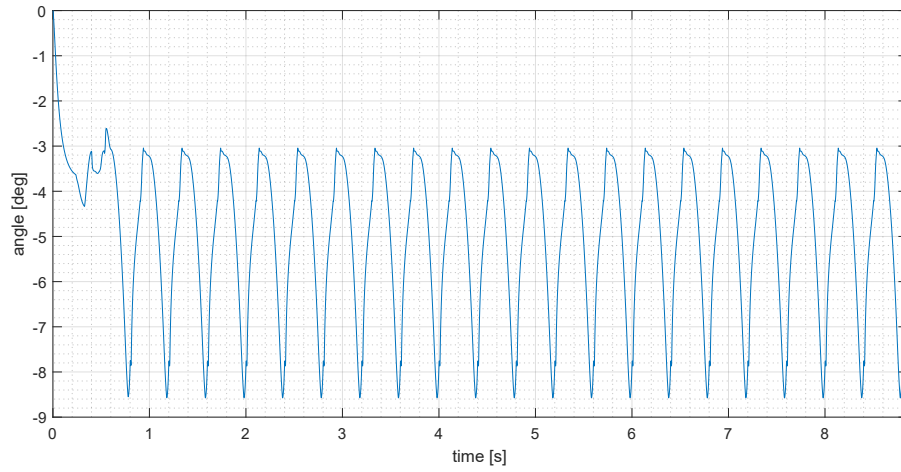


(a) Point A movement - y axis



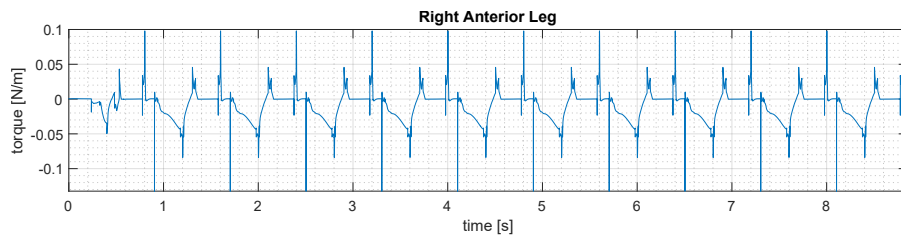
(a) Point A movement - z axis

Figure B.7: Evolution of point A, of Simscape model, during gait, test 2



(a) Spine oscillation - x axis

Figure B.8: Spine oscillation of Simscape model, test 2

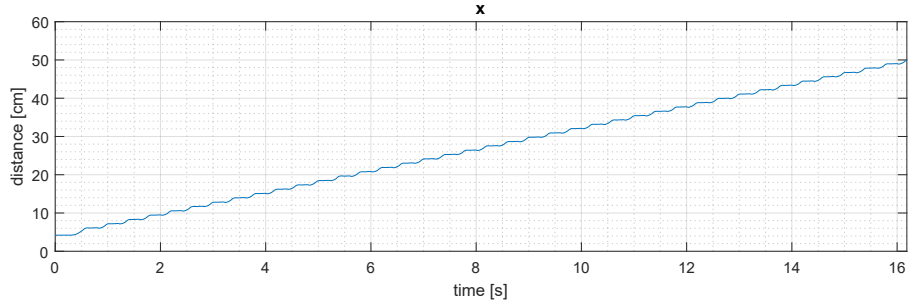


(a) Right anterior leg torque

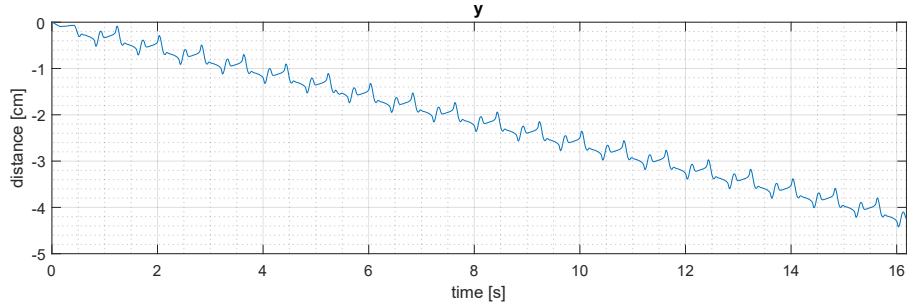
Figure B.9: Legs torque of Simscape model, test 2

### B.3 Test 3: compliance in the legs - spine with high stiffness

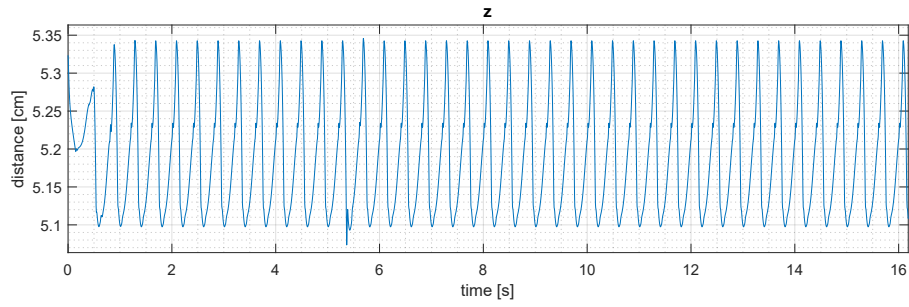
For the Simscape model, the result of this test are illustrated in this subsection. In fig. B.10, like in previous sections, show the trend of point A during the gait. We can see that the time to travel from 0 [cm] to 50 [cm] is 16.1851 [s] the maximum deviation on  $y$  axis is 4.4196 cm and the trend on  $z$  axis is between 5.0735 [cm] and 5.3456 [cm]. In figure B.12 the torque produced by the Left Anterior leg is showed, its value is 0.3300 [Nm]. The max power in absolute value is provided by Right Anterior leg. It's value is 2.8778 [W]. The last graph in figure B.11 present the displacement of spine. These oscillation are really small and go from  $-2.2054 * 10^{-4}$  [deg] to  $1.8062 * 10^{-4}$  [deg].



(a) Point A movement - x axis

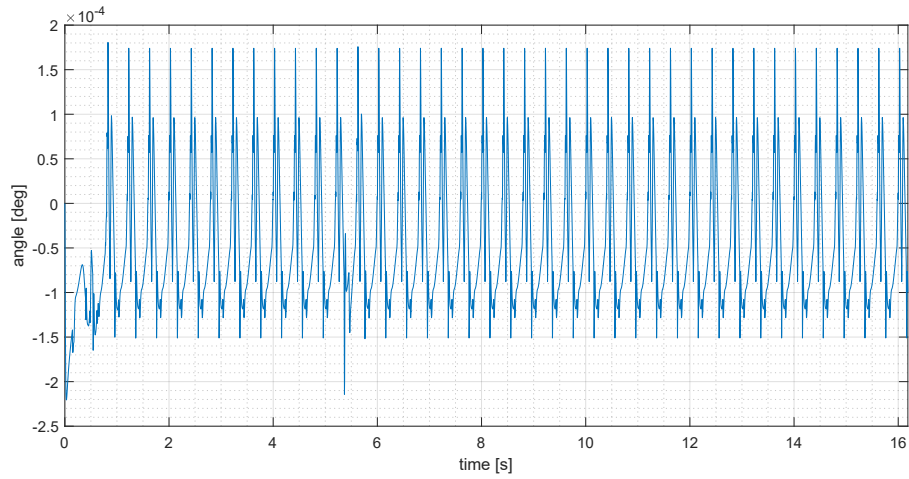


(a) Point A movement - y axis



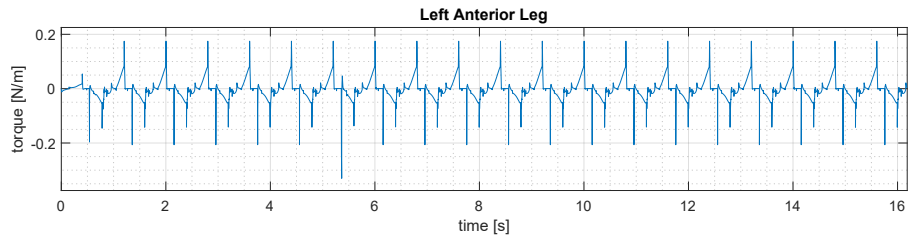
(a) Point A movement - z axis

Figure B.10: Evolution of point A during gait, test 3



(a) Spine oscillation - x axis

Figure B.11: Spine oscillation, test 3

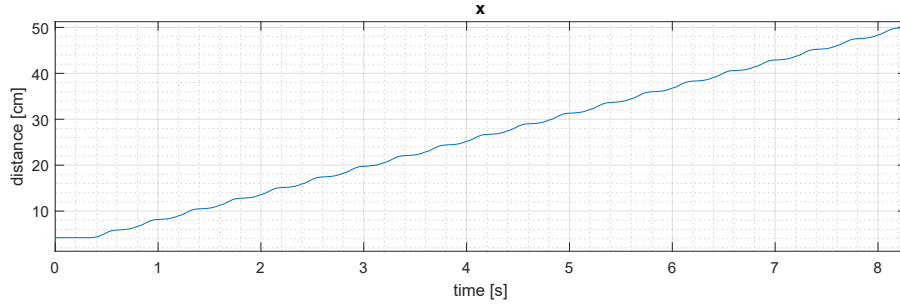


(a) Right anterior leg torque

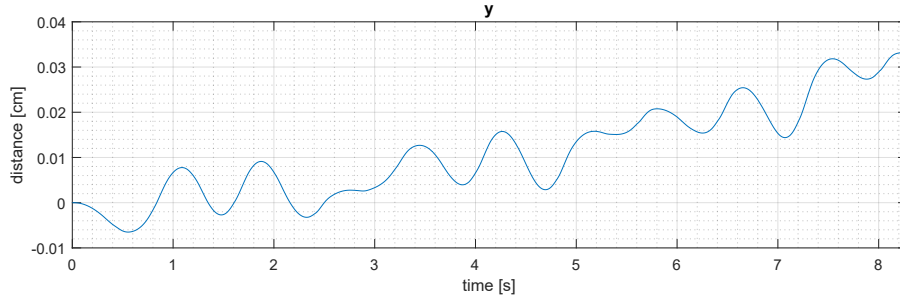
Figure B.12: Legs torque, test 3

## B.4 Test 4: legs with high stiffness - spine sinusoidal

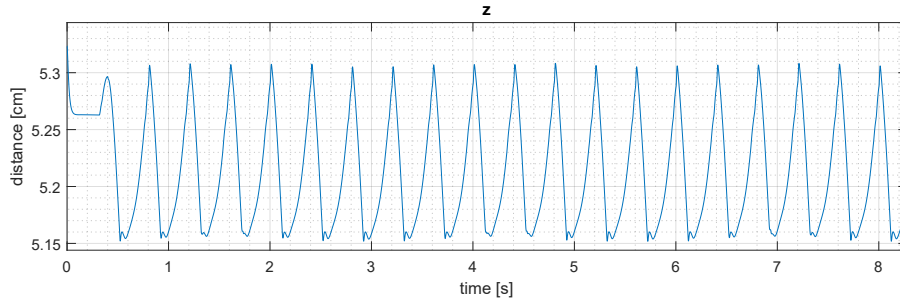
In this subsection are reported the results of Simscape model, figure B.13 shows the trend of point A during the gait. In particular we can see that the time spent to travel from 0 [cm] to 50 [cm] is 8.8950 [s] the maximum deviation on  $y$  axis is 0.6144 [cm] and the trend on  $z$  axis is between 5.1507 [cm] and 5.3235 [cm]. In the second figure B.15 is plotted the maximum torque produced by robot's hip joints. The maximum value is provided by the Left Anterior leg, and it is 0.2314 [Nm]. The maximum power spent for walking, is provided by right anterior leg. Its value is 2.0017 [W]. The last graph B.14 present the displacement of spine in degrees. These oscillation are small and go from  $-1.0644$  [deg] to  $0.2284$  [deg].



(a) Point A movement - x axis

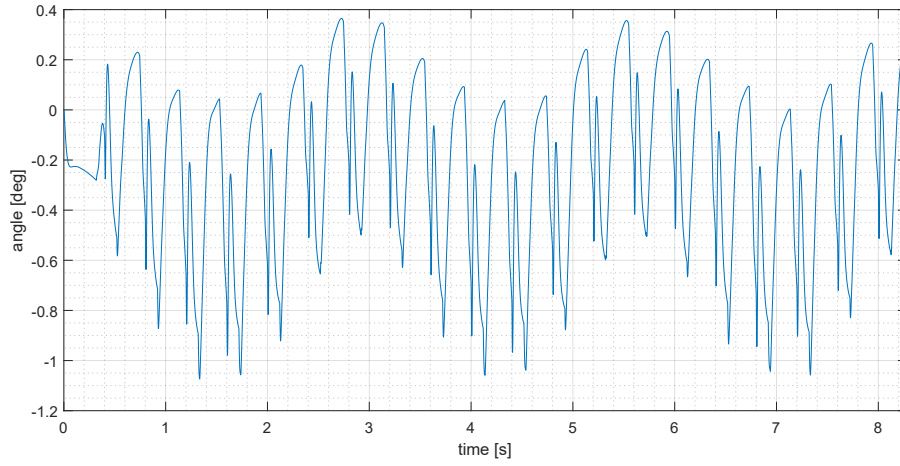


(a) Point A movement - y axis



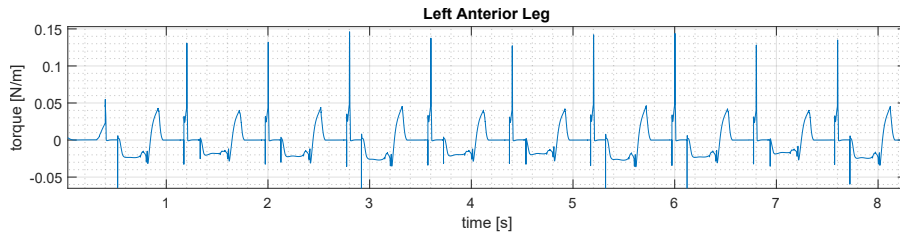
(a) Point A movement - z axis

Figure B.13: Evolution of point A, of Simscape model, during gait, test 4



(a) Spine oscillation - x axis

Figure B.14: Spine oscillation of Simscape model, test 4

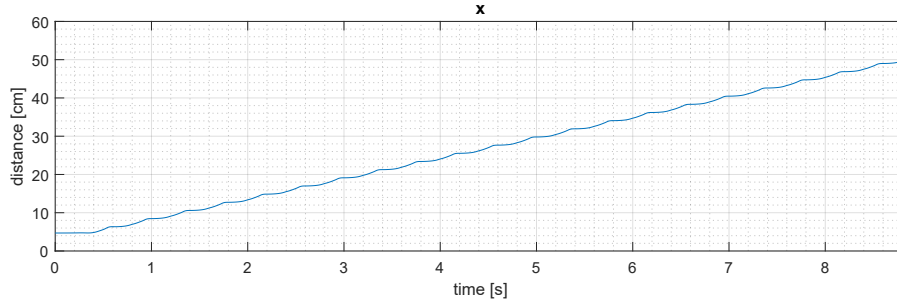


(a) Right anterior leg torque

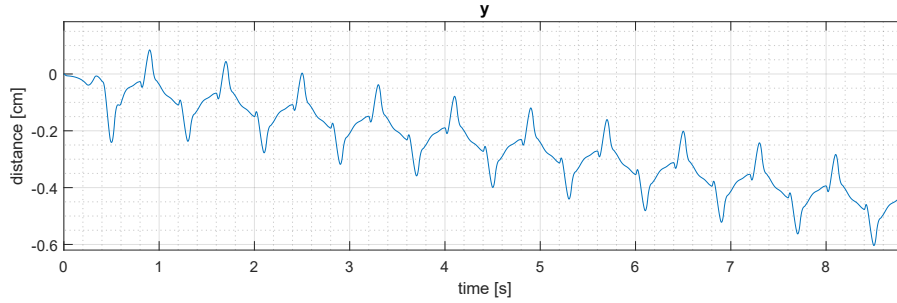
Figure B.15: Legs torque of Simscape model, test 4

## B.5 Test 5: legs with sinuoidal behaviour - spine maximum stiffness

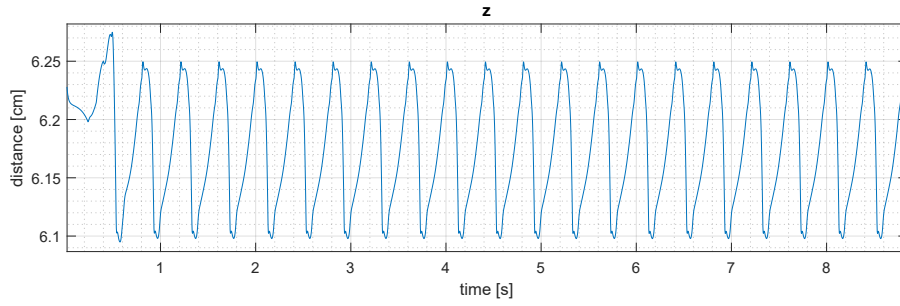
The results of Simscape model are reported here. In figure B.16 is showed the trend of point A during the gait. The time taken to travel from 0 [cm] to 50 [cm] is 8.9221 [s], the maximum deviation on  $y$  axis is 0.6347 [cm] and the trend on  $z$  axis is between 5.1517 [cm] and 5.3235 [cm]. In the figure B.18 is plotted the maximum torque, produced by left anterior leg its value is 0.2387 [Nm]. The maximum power spent for walking is provided by the right anterior leg. Its value is 1.6007 [W]. The last graph in figure B.17 present the displacement of spine in degrees. These oscillation are small and go from  $-0.0370$  [deg] to 0 [deg].



(a) Point A movement - x axis

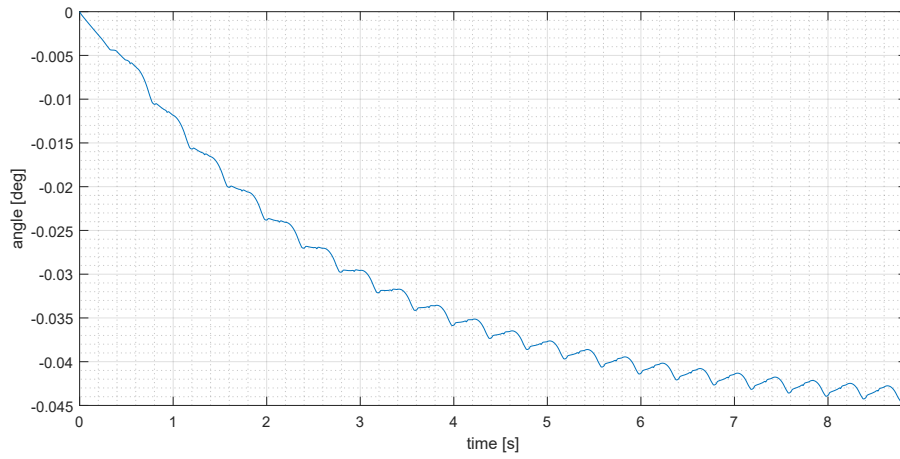


(a) Point A movement - y axis



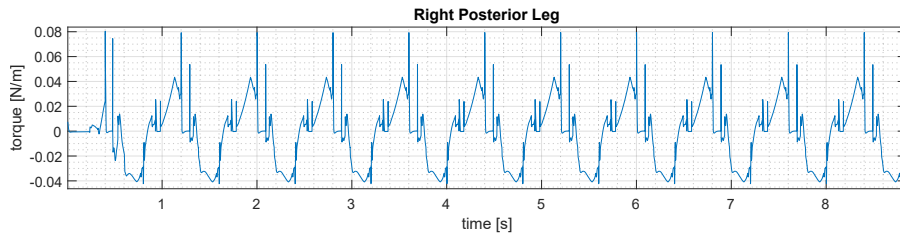
(a) Point A movement - z axis

Figure B.16: Evolution of point A during gait, test 5



(a) Spine oscillation - x axis

Figure B.17: Spine oscillation, test 5



(a) Right posterior leg torque

Figure B.18: Legs torque, test 5





# Bibliography

- [1] R McNeill Alexander. Elastic energy stores in running vertebrates. *American Zoologist*, 24(1):85–94, 1984.
- [2] RMcN Alexander. Three uses for springs in legged locomotion. *International Journal of Robotics Research*, 9(2):53–61, 1990.
- [3] Reinhard Blickhan. The spring-mass model for running and hopping. *Journal of biomechanics*, 22(11-12):1217–1227, 1989.
- [4] Ioannis Poulakakis, James Andrew Smith, and Martin Buehler. On the dynamics of bounding and extensions towards the half-bound and the gallop gaits. In *In Proceedings of the 2nd International Symposium on Adaptive Motion of Animals and Machines (AMAM)*. Citeseer, 2003.
- [5] Claire T Farley, James Glasheen, and Thomas A McMahon. Running springs: speed and animal size. *Journal of experimental Biology*, 185(1):71–86, 1993.
- [6] Christian Gehring, Stelian Coros, Marco Hutter, Michael Bloesch, Markus A Hoepflinger, and Roland Siegwart. Control of dynamic gaits for a quadrupedal robot. In *2013 IEEE international conference on Robotics and automation*, pages 3287–3292. IEEE, 2013.
- [7] Chunlei Wang, Ting Zhang, Xiaohui Wei, Yongjun Long, and Shigang Wang. Dynamic imbalance analysis and stability control of galloping gait for a passive quadruped robot. *Applied bionics and biomechanics*, 2015, 2015.
- [8] Fumiya Iida and Rolf Pfeifer. Cheap rapid locomotion of a quadruped robot: Self-stabilization of bounding gait. In *Intelligent Autonomous Systems*, volume 8, pages 642–649. IOS Press Amsterdam, The Netherlands, 2004.
- [9] Jonas Buchli, Fumiya Iida, and Auke Jan Ijspeert. Finding resonance: Adaptive frequency oscillators for dynamic legged locomotion. In *Intelligent robots and systems, 2006 IEEE/RSJ international conference on*, pages 3903–3909. IEEE, 2006.
- [10] Hugh M Herr, Gregory T Huang, and Thomas A McMahon. A model of scale effects in mammalian quadrupedal running. *Journal of experimental biology*, 205(7):959–967, 2002.
- [11] Panagiotis Chatzacos and Evangelos Papadopoulos. Parametric analysis and design guidelines for a quadruped bounding robot. In *2007 Mediterranean Conference on Control & Automation*, pages 1–6. IEEE, 2007.

- [12] Qi Deng, Shigang Wang, Wei Xu, Jinqiu Mo, and Qinghua Liang. Quasi passive bounding of a quadruped model with articulated spine. *Mechanism and Machine Theory*, 52:232–242, 2012.
- [13] Xiaohui Wei, Chunlei Wang, Yongjun Long, and Shigang Wang. The effect of spine on the bounding dynamic performance of legged system. *Advanced Robotics*, 29(15):973–987, 2015.
- [14] MF SILVA, RS Barbosa, and TS Castro. Development of a quadruped robot model in simmechanicstm. In *Nature-Inspired Mobile Robotics*, pages 709–716. World Scientific, 2013.
- [15] RA Hyde and J Wendlandt. Tool-supported mechatronic system design. In *2008 34th Annual Conference of IEEE Industrial Electronics*, pages 1674–1679. IEEE, 2008.