SIMULATION OF WALKING HUMANOID ROBOT BASED ON MATLAB/SIMMECHANICS

Sébastien Corner
scorner@vt.edu

The Robotics & Mechanisms Laboratory, RoMeLa
Department of Mechanical Engineering of the University of Virginia Tech

Abstract

The complexity involved in robotic simulation seems to be an insurmountable obstacle for students to overcome. It usually requires substantial mathematics and programing skills to be master by students. This paper proposes a new approach to humanoid robot simulation that has the potential to reduce the intricacy of creating a biped simulation model. The approach investigates the use of the SimMechanics toolbox to build the computer model of the biped robot. The method operates by assembling body and joints modules to perform the framework of the robot. The control motion of the robot requires neither the mathematical conception of inverse kinematics nor inverse dynamics. It adapts position joints control using PID controller that tracks an angle trajectory file. The methodology of acquiring the corresponding coefficient of the controller, that allows the simulated robot to walk according to the given trajectory file, is described. Another key component of this simulation is a control technique that models the ground based on PID controller. It provides a basis for our analysis of the robot system performance while walking and a better comprehension of the ground reaction forces and impacts. Finally, this paper opens the floor to further projects such as passive dynamic walking.

Introduction

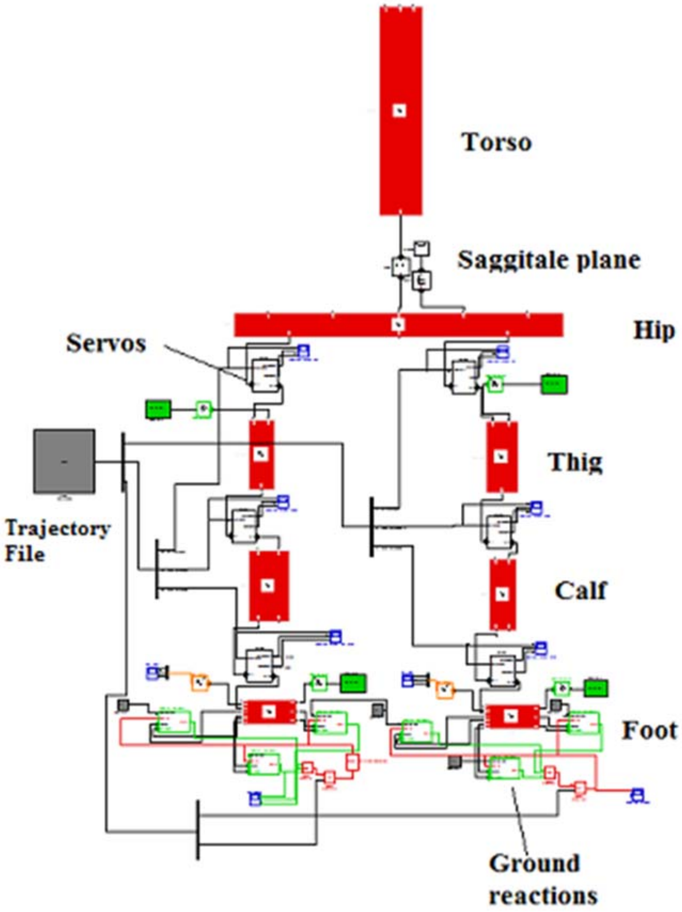
Simulation becomes a strategic tool in the research field of robotics used by many students, researchers, and developers. It supports the development of advanced designs and investigates a wide range of solutions for complex problems [1]. Design solutions and experimental environments are built according to their imagination and creativity. The exponential increase in hard drive capacity has allowed the development of numerous software packages for modeling multibody systems. Some of them have been specifically developed for robots such as Webots, Adams and RoboWorks. These softwares are really convenient for modeling robots and their dynamic motions, but are viewed as inadequate for modeling actuators, sensors, ground contacts, and impact [2]. SimMechanics extends Simulink/Matlab to provide tools for modeling and simulating mechanical systems [3]. With SimMechanics modeling bodies, joints, actuators, external forces, and the interconnections in the robot drive system is quite simple. Previous papers exhibit the simplicity of this software by providing details and simple examples that allows novice users to be familiar with the software [4][5][6]. Physical modeling and parameter settings are presented in detail. In addition, they develop the approach of modeling the nonlinear system, the inverted pendulum, built through SimMechanics. Their approach is to compare this model with the mathematic model based on the derivation of the forward dynamics and differential equations. Results show that both models are almost identical which confirm and validate the SimMechanics model. The main differences results to the simplicity of implementing this nonlinear system. The mathematical approach is quite cumbersome, as it requires the derivation of the nonlinear motion of that mechanism and linearizes the equations using state variables. The SimMechanics model is much simpler. Bodies and joints are linked together to form the model, while joint sensors and actuators are added for simulation.

This paper is about a new approach that simulates biped robotic though SimMechanics that embodies pendulum motions, which has the potential to reduce the intricacy of creating biped simulation model. The approach starts with highlighting the skeleton of the simulated robot made of joints and body links. Once built, the control motion of the robot is explored toward PID controller used to simulate the revolute joints and to simulate ground reactions. This leads the approach to exhibit the results of this research which consists of having the simulated robot accomplish a full step. To validate our approach, screenshot of the simulation will be presented and a website link will be shared to access the video.

Architecture of the simulated robot

The Frame

The structure of the simulated biped robot, presented on figure1a, contains a complete lower body including, two legs and two foot, attached to a torso. The head and arms are not designed to shorten the complexity of the simulation. The hip connects the leg to the torso, using a revolute and weld joint respectively. The knee connects the thighs and the calf which is connected to the foot by the ankle. Both the knee and the ankle are revolute joints. In this research, the robot is free to move in the sagittal plane (y and z axis) and can rotate around the perpendicular axis of the plane (x axis). The overall mass of the robot is 37 kg for a total height of 1.45m. The weight and length are evenly distributed among each link. (The torso weights 10 kg, the hips and each thigh and calf weight 5 kg and each foot weight 1 kg). The moment of inertia of each body is respect to the body’s center of mass. It was assumed that all links were rigid and frictions in the joints were neglected. Finally, the simulated robot is a rigid robot with six degrees of freedom that can move and rotate freely in the sagittal plane.



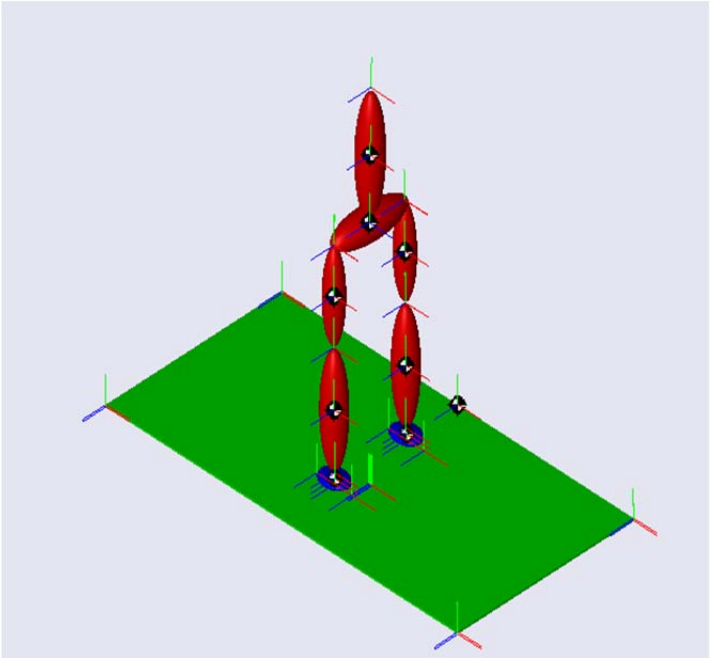


Figure1A: Algorithm structure of the simulation **Figure1b:** Model of the robot.

We note that each revolute joint is controlled by a servo module and the ground contacts as well as the frictions forces are simulated while the foot impacts the ground. SimMechanics does not permit surface contact between objects, therefore, three points of contact are modeled on each foot to simulate the impact. The model of the robot, when the simulation runs, is shown on figure 1b. The green square refers to the ground and has a thickness of 1cm.

The Servo Modules

To simulate the motion of the robot, a module called servo, is performed and the SimMechanics code of this module is presented on figure 2. This code embodies the general structure of a close control system where the given trajectory is tracked. The error between the desired setpoint and the joint angle measured with the sensor is calculated. Then, the controller attempts to minimize the error according to the adjusting coefficients that determines the amount of torque required. The previous and following articulations are connected to Art 1 and Art 2 respectively.

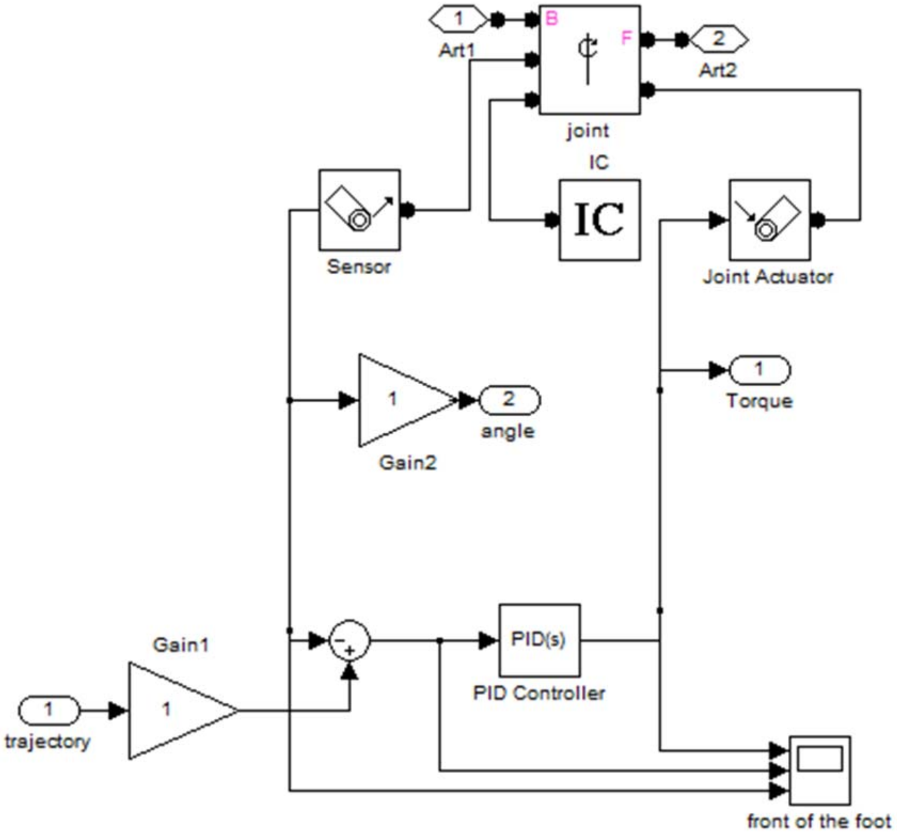


Figure2: Algorithm of the servo module that uses PID controler.

The Ground Reaction Forces

The ground contacts simulation presented on figure 3 is more complex. In this figure, only the reaction force with the ground is simulated. The principal controller is still a PID controller with a feedback mechanism, but activated by a digital signal represented by the red lines. The coordinates of the center of mass of the robot is related to the world coordinate system in the simulator. The body members are connected to the frame of the robot and are related to each other. However, the feet of the robot are observed in the world coordinate system to determine their positions compare to the ground level (plan y=0). If the foot is above the ground, both the error value signal indicated by yellow lines (ground level minus the foot position), and the ground reaction indicated by green lines are forced to zero. Otherwise, the foot is in contact with the ground and the PID controller is activated to generate the appropriate ground forces at each point of the foot.

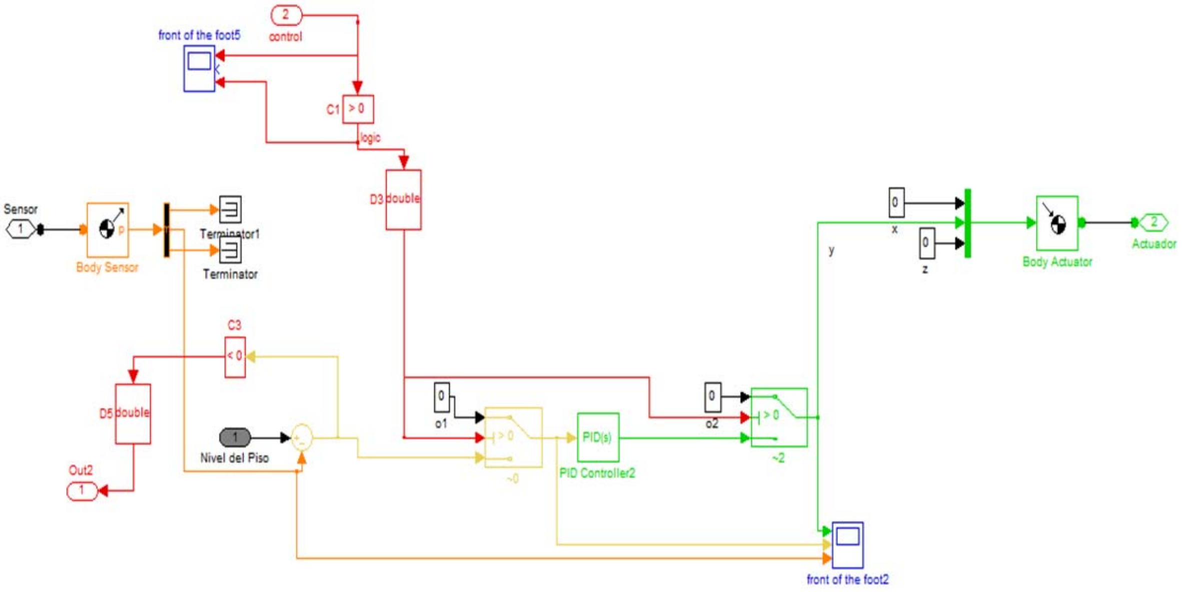


Figure3: Algorithm of the ground force reaction.

The frictional forces

When walking, a real robot is submitted to horizontal frictional forces that prevent the robot from slipping. This research considers the simulation of these forces as a key component to providing realism in the walking motion. The algorithm structure on figure 5 is relatively similar than the previously presented structure. The modification resides in the addition of the horizontal forces, illustrated on the top of the figure. The feature of this part of the code is to obtain the memorization of the contact point on the abscise axis between the foot and the ground using a trigger function. Once the foot impacts the ground, the logic signal changes state and the last position of the foot is recorded and sent to the controller as an input. The PID controller is activated and generates horizontal forces in the opposite direction to the motion. To simplify the overall algorithm of the simulation, the frictional forces are only simulated on the center point of the foot.

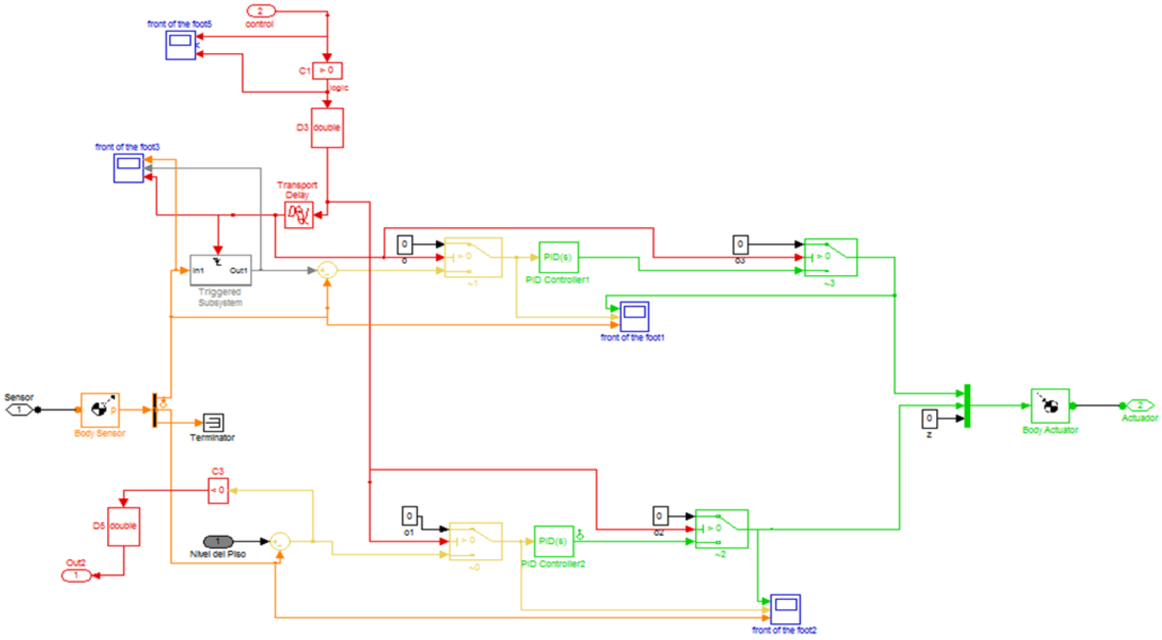


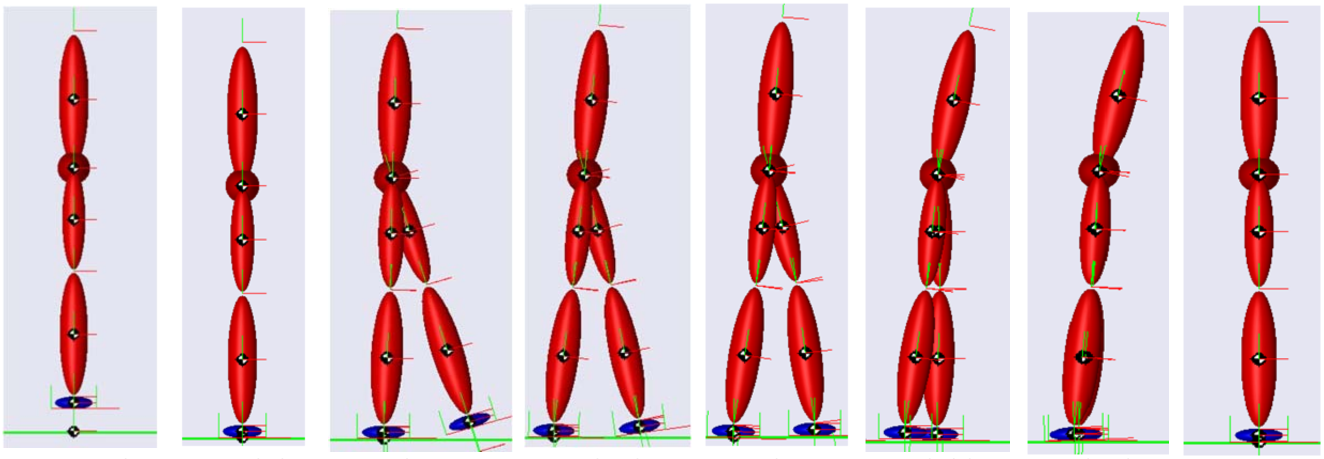
Figure4: Algorithm of the ground reaction force and friction force.

Tuning the coefficients of the PID Controller

The control system toolbox is a wonderful tool for tuning coefficients of the PID controller. It provides, for example, a frequency response diagram that allows the user to analyze the behavior of the system from a step response. The behavior can be directly modified by varying the cursor for each coefficient. The benefit of this system resides in the fact that the user does not have to run the simulator to determine the correct coefficients. Usually, the user runs it and looks at the behavior of the system by visioning the response on the plots. This system toolbox does it directly. To confirm our design, we verify the rise time, overshoot, and settling time. For more information about the toolbox, instructions are well documented on MathWorks[3]. Nevertheless, it is not possible to tune the entire simulated robot, it needs to be decomposed. By making the assumption that the leg is a sequence of simple pendulums, we can find the coefficients of the controller. We create a 10 kg simple pendulum in a separate file and tune it to find the coefficient with a step response. Their values are Kp=241, Kd=110, Ki=55. Similar method is used to find the coefficients for the ground module. We generate a 35 kg body that is dropped 10cm. The goal of this method is to tune the system in order to modelize the ground impact. The body should neither bounce nor sinks on the ground with a maximum error of 1 cm. Thus, we found really high coefficients Kp=$1.8 * 10^6$, Kd=$3 * 10^4$, Ki=$11 * 10^6$ that meet the design requirements. Finally, we model the frictional force by pushing a body by a 400 N horizontal force. If no force is applied to the opposite direction, the body moves to infinite direction. The design control requirement for this model is to obtain opposite force that sticks the robot on the ground with a maximum error of 1 cm. To fulfill this need we found these corresponding coefficient values Kp=$8 * 10^4$, Kd=$5 * 10^3$, Ki=$1 * 10^5$. Once determined, all of these coefficients are inserted in the simulated robot with the assumption that each module will react similarly while independently testing.

Validation of the Simulation and Results

The goal of this research is to obtain a full step of the simulated robot. Illustrated on figure 6 is the sequence of pictures of the simulation which validates the success of the research. The robot starts by a 10 cm jump (A) to verify the consistence of the ground (B). Once standing up the left foot is disconnected to the ground by forcing the digital signal on the left foot to change state. Then, the left leg slangs and the right ankle bend to bascule the robot forward (C). Fixed on the ground, the right foot allows the leg to process a linear inverted pendulum motion. All the mass of the simulated robot is balanced due to the torque applied on the right foot. At the same time, the left leg processes a simple linear pendulum motion to reach the next contact point or impact with the ground (D). At the impact time, the motion of the legs is inverted. The left foot becomes the pivot point (E) and the previous process is repeated (F,G,H). The video of this simulation is available on the following link:



A t=0s B t=0.05s C t=0.45s D t=0.70s E t=0.75s F t=0.80s G t=0.90s H t=1s

Figure5: Sequence of pictures of the simulation.

Below are presented the graphs of the right ankle and the left hip during the first part of the walking motion (from B to E). The first plot refers to the desired angle which is tracked by the controller. The second plot is the torque applied on the joint and the third plot is the measured angle. Usually, the angles of the joints performed sinusoidal signals. In this case, the inputs are linear functions in order to have a better observation of the walking motion. The curve of the measured angle of the ankle matches the input signal that confirms and validates the performance of the controller. There is small discrepancy between these signals because applied on the foot are the ground reaction forces, the frictional forces and the torque which are challenging to simulate in same time. The maximum torque applied on the right angle is 80 N.m.

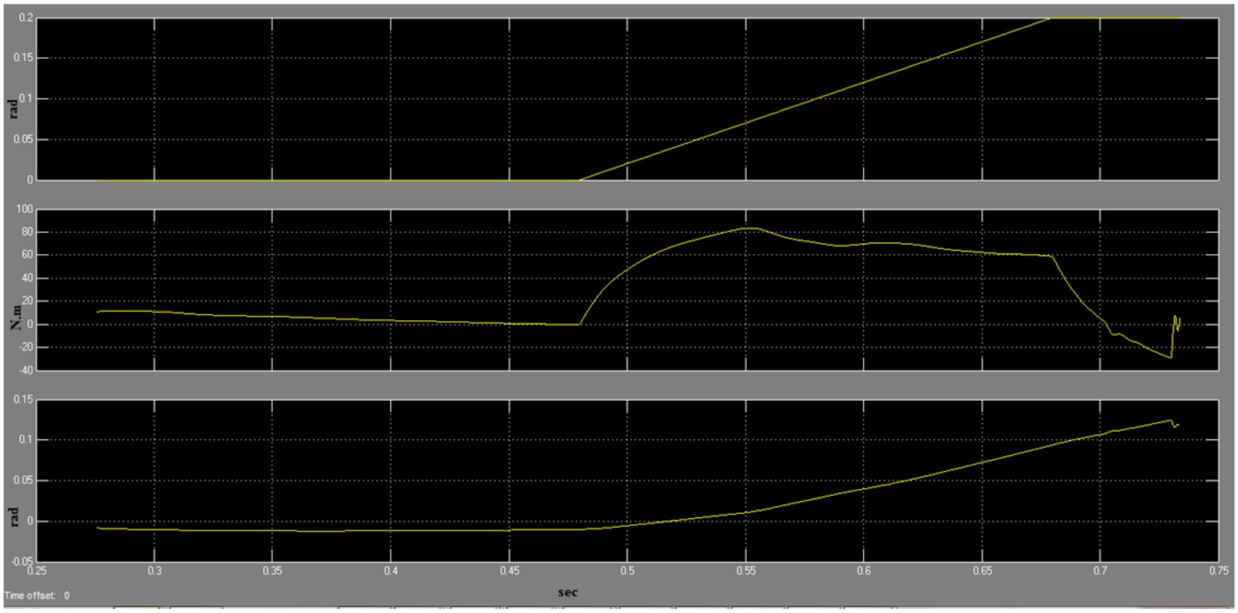


Figure6: Plots of the right ankle during the first part of the walking motion (B to E). From the top to bottom: desired angle, Torque, and measured angle plot.

The curve of the measured angle of the left hip matches perfectly the input signal too. The pendulum motion assumption for the leg foot is verified. The output torque needed to perform this motion varies from 30 N.m to minus 35 N.m . The plots of the ground reaction forces and frictions are situated on the appendix.

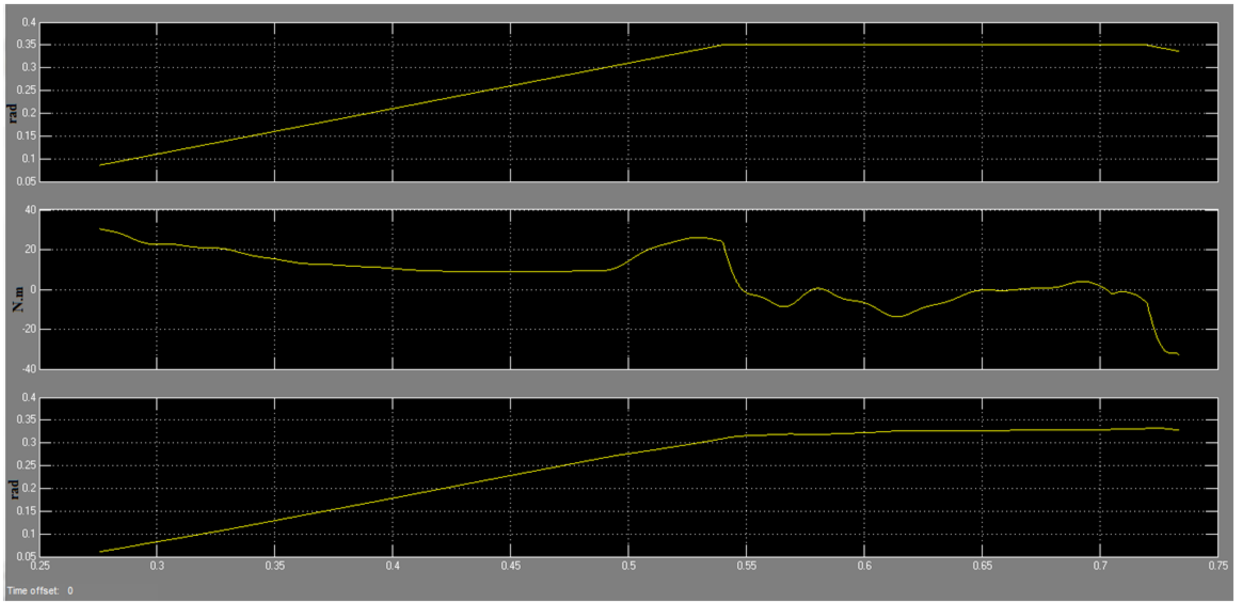


Figure7: Plots of the left Hip during the first part of the walking motion (B to E). From the top to bottom:, desired angle, Torque, and measured angle plot.

Conclusion and Future Works

This paper proofs the accessibility for any scientists to the simulation of walking humanoid robot without any mathematics or physic background relative to this topic. The method and validation of this simulation are detailed along this paper. The simulation shows that a maximum torque of 80 N.m is required on the ankle of a 35 kg humanoid robot to success this particular walking motion. Previous researches used SimMechanics to simulate humanoid robot [7,8]. But they use ZMP calculations to perform the control motion which is typically complex to implement in this model. Further work in walking motion should be undertaken based on the study and implementation of passive dynamic walking in the simulation. This study should aim by minimizing the torque in the ankles through the maximization of the natural motion of the robot. Nevertheless, the issue of this simulation is its run time. One solution will be to explore parallel computing with Matlab in order to run the simulation on multiple processors.