INSERT LOGO HERE

Hybrid Dynamical Systems - Legged Locomotion in Robotics

INSERT LOGO HERE

Yusuf Karabacakoğlu Onurcan Yılmaz

Advisor: Assist. Prof. Mustafa Mert Ankaralı

Advisor Assistant: Jean Piere Demir

Middle East Technical University, Department of Electrical & Electronics Engineering

Introduction

Figure 1: Dynamic Walker designed by DARPA Challenge

To decrease the cost of motion, researchers tried to benefit from forces that are already available in the environment and build prototypes called passive walkers. A passive walker consumes very little energy to travel from one point to with the help of gravity. In this project, the dynamic walker designed by DARPA Challenge in Figure 1 (Robertson, Paik, Wu, & Ijspeert, 2017) has inspired us to improve the performance by engineering a better hardware design and implementing a simulation tool to optimize the motion. Their walker was so simple in design and algorithm, but it can walk indeed. We are surprised by how simple this robot is and wanted to try it ourselves.

Aim of the Project

The original robot was controlled by an open-loop control system since its only sensors are two buttons at bottom of the legs. Since the robot's motion can be modelled as a double pendulum, we thought we can close the feedback loop by controlling the legs angular velocity.

To accomplish this, we made its simulations and add new features to its hardware design.

Methodology

HARDWARE IMPROVEMENTS

The first part of the project was construct a hardware that can measure angular velocity. To measure angular velocity of the robot, we introduced two IMU sensors (MPU6050) to the system. We also want to read their measurements, to do this we used Particle Photon instead of Arduino.

These improvements totally changed robot's electronic design. The new design can be seen in Figures 1 and 2.

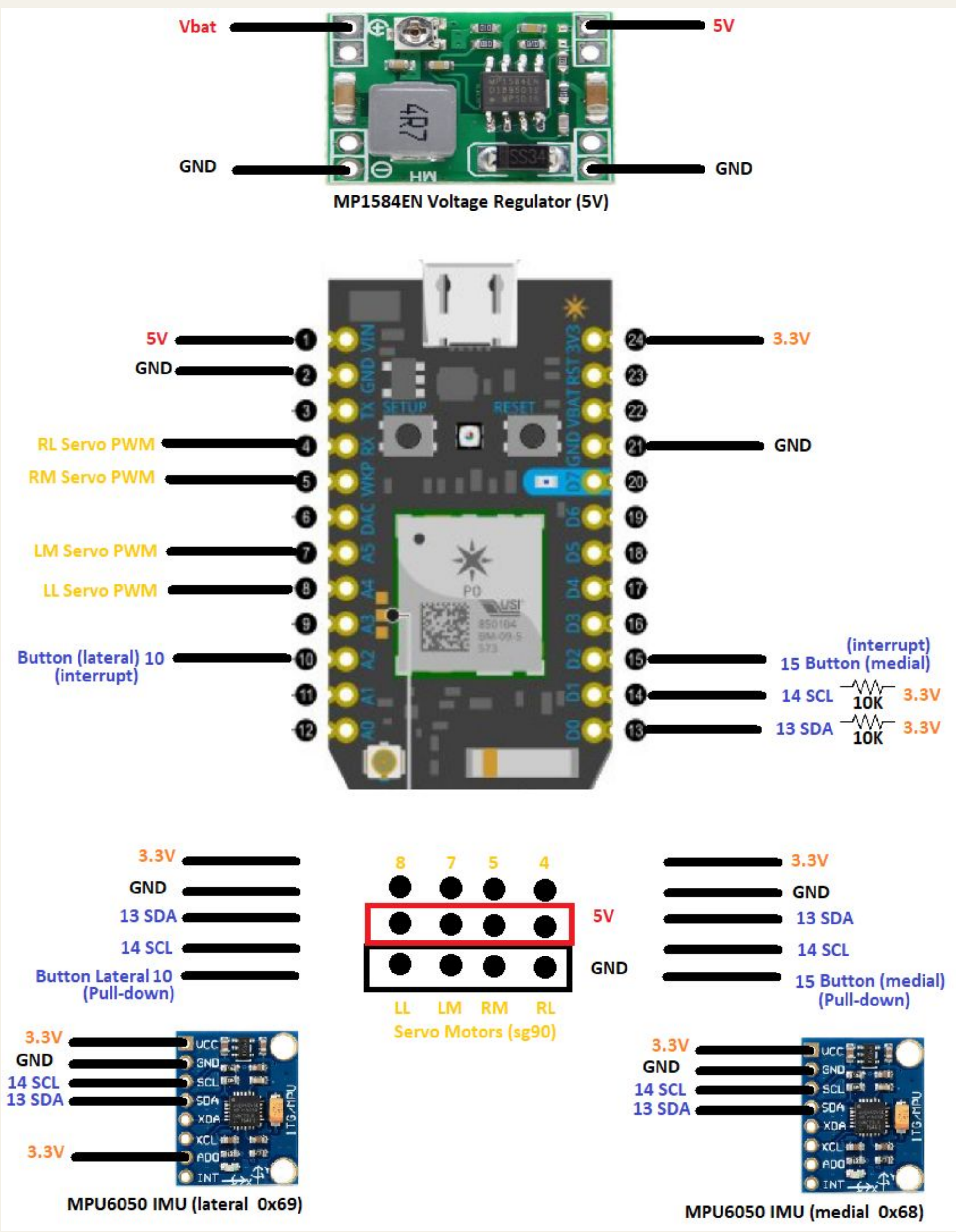

Figure 1: New electronic design of the robot.

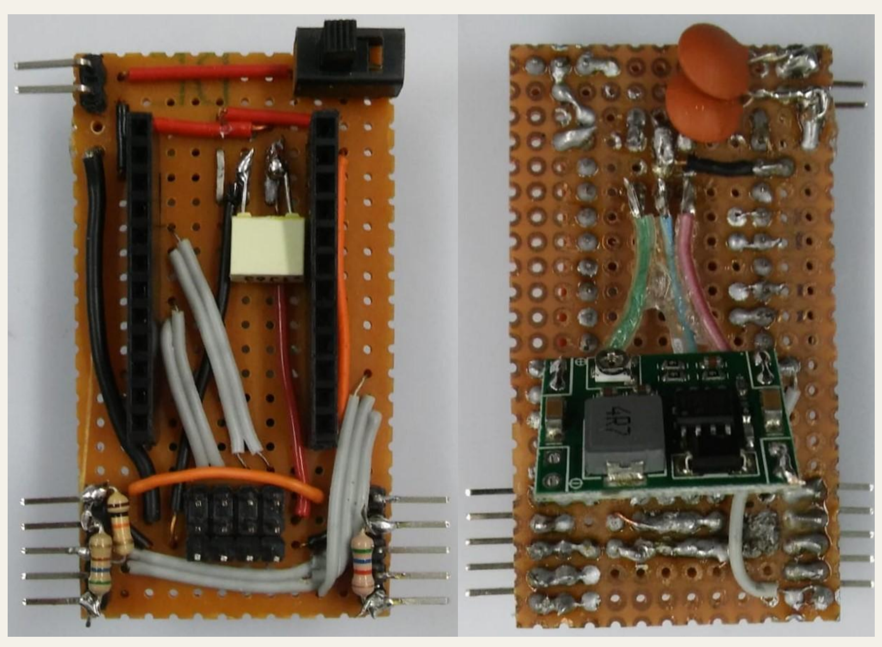

Figure 2: Front and back view of new PCB board.

SIMULATIONS

The second part of the project was to come up with a good simulation that would be close to the real-world environment to optimize the walker controller, because using the real-prototype is not as efficient as using simulation due to its cost and the time it would take to build one. We used Matlab's Simscape for simulation environment.(specialized solid body environment of simulink) Fig. 4 shows the robot model in Simscape. In Fig. 5 hip position w.r.t ground is shown.

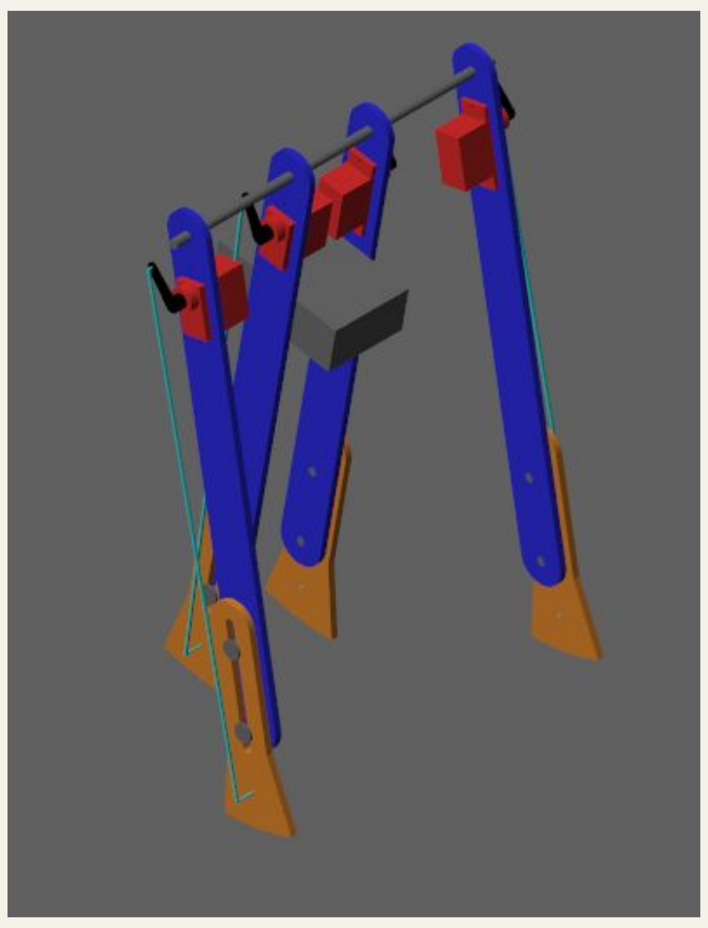

Figure 4: 3D model of robot in simscape

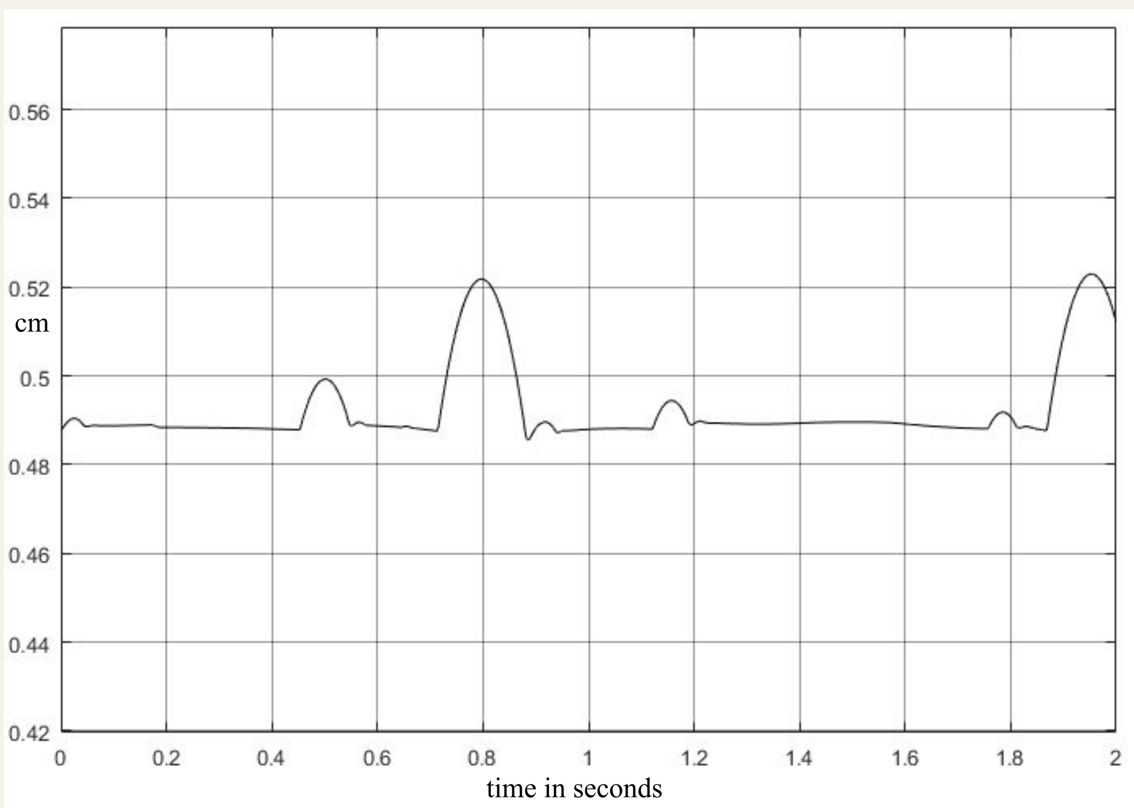

Figure 5: Hip position with respect to ground

CONTACT MODELING OF PASSIVE WALKER

For contact modeling we have chosen force-based penalty force which models the contact with some overlap and uses force law for spring-damper to find the necessary contact forces, so it is continuous.. For this purpose we have used Simscape Multibody Contact Forces Library which is contributed by Steve Miller in 2019. It is defined between two bodies such as in Fig. 6. It models the contact between two solid blocks by putting a really small sphere to the follower side with the desired base, in our case it was between foot and the ground, and we have used 3 sphere contacts for each foot as two edges and mid.

Figure 6: Foot-ground contact model.

CONTROLLER

We are controlling the system with 4 servo motors. We are pushing energy to the system with them, and how much the pushed energy should be is measured by the current energy of the system with measuring the swing time with the help of basic on-off sensors, and the durations of push-off is calculated as follows in Fig. 7. and Fig. 8. shows the states of the controller.

```
% down=60deg mid=30deg time1=new_time time2=old_time
pushoff=mid+(down-mid)*(time1-time2)/max_swing_time;
```

Figure 7: Foot-ground contact equation.

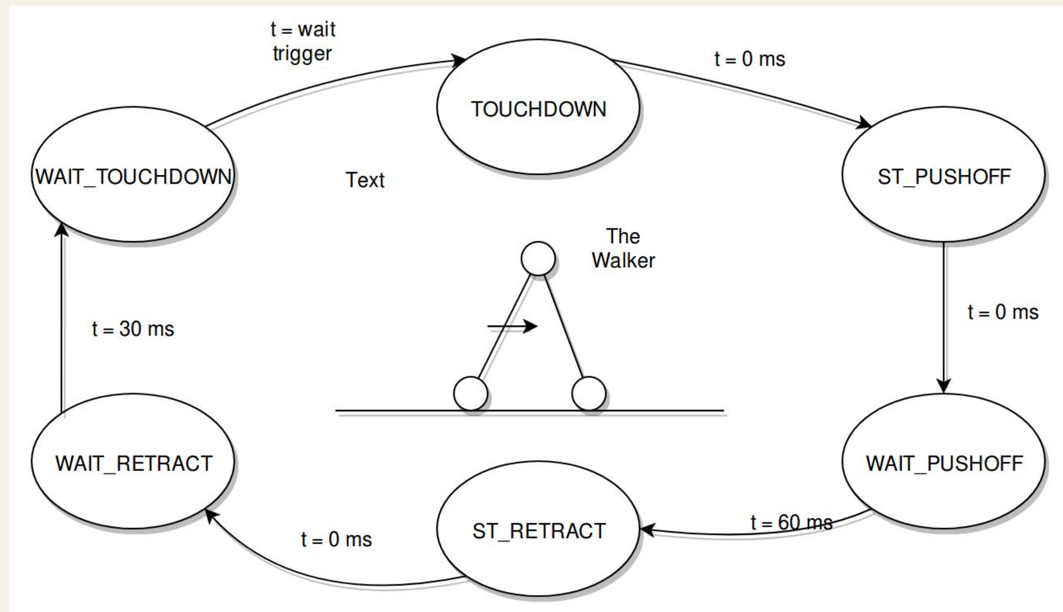

Figure 8: Controller flowchart.

The original controller were using delays for each event and this method was blocking the controller to do other tasks. Since we want to read the IMU sensor outputs, we used timer interrupts instead.

Future Work

Currently, the controller parameters are not optimized and robot is not moving as we expected. So our first focus is to overcome this. After optimization, angular velocity data will be inspected and leg's push-off amount (which is constant now) will be controlled to have a more stable motion.

Conclusion

To conclude, we have tried to model the passive walker more realistically, and instead of calculating the equations of motion we defined the Newtonian Forces between all bodies, then applied input to actuators. We think this is a much better way then using simplified geometry to find equations of motion. However, our realistic approach was taking too much time to simulate the robot(1 second of simulation time done in 20minutes), so we took the intuition that we gained from simulations and applied to the real hardware, and this is what we did so far.

References

Mathworks. (2018). Simscape Multibody Link plug-in. Retrieved from https://www.mathworks.com/help/physmod/smlink/ref/linking-and-unlinking-simmechanics-link-software-with-solidworks.html

Steve Miller. (2019). Simscape Multibody Contact Forces Library. Retrieved from https://www.mathworks.com/matlabcentral/fileexchange/47417-simscape-multibody-contact-forces-library

Robertson, M., Paik, J., Wu, A., & Ijspeert, A. (2017). Rando the Walking Robot. Retrieved from http://bot-hed.com