Olivia Ball

9 December 2019

Leg Exoskeleton Final Project

Introduction

Lower-limb exoskeletal devices have been a growing area of research recently for uses both medically and non-medically. Medical uses for these devices generally fall into one of two categories: assistive or rehabilitative. Assistive devices are designed to help with performance of everyday activities inside and outside of the hospital, such as walking, sitting/standing, and carrying objects. Rehabilitation devices are used to help reduce the limits that can occur with typical physical therapy such as limited time between therapist and patient and bad repeatability [4]. Those with neurological injuries, like stroke and spinal cord, and elderly individuals with muscle degeneration, can strongly benefit from these applications. Additionally, there is more development in nonmedical devices which can be used to assist healthy individuals. One example of a real-world application for these devices is one which provides help to soldiers to allow them to handle heavy equipment while moving through rough terrain. Current models for these devices are very large, extremely costly, and are limited in the amount of stability they provide [3]. In order for leg exoskeletons to be more practical for uses outside of the hospital they need to become more convenient, durable and comfortable for the user.

Methods

In order to obtain the torques for the model, I created two different functions to use the recursive Newton-Euler inverse dynamics algorithm. The first function required was to perform outward recursion which required 7 inputs to get 2 outputs. Some of the inputs came from a given excel sheet, these were the rotation (t), velocity (thetadot) and acceleration (thetadd). The other inputs required were the mass of the link (m), inertia tensor at the center of the mass (ic), a vector to locate the center of mass of each link (s) and a displacement vector (p) that was obtained using the Denavit-Hartenberg transform matrix.

When performing outward recursion, the algorithm starts at the base frame and moves to the end frame. By supplying all the information mentioned above, the outward recursion function then gives you the inertial force (F) and torque(N) acting at the center of mass of each link.

Inward recursion works backwards in order to get the torques at each joint. Due to the reverse order, you need to define a new additional coordinate frame the defines the point on the foot you are trying to describe. The new coordinate frame only affects the displacement vector and the theta values, so they can be accounted for by changing the final row. The final theta value can be assumed as zero to provide the identity matrix and the displacement vector can be changed by using the Denavit-Hartenberg transform matrix. By feeding these newly defined values and the outputs from the outward recursion into the function, you are able to get the torques at each joint.

Finally, in order to implement this at each time step, I used a for loop. The for loop took the required rotation, velocity, and acceleration at each timestep and placed those values into the outward and inward recursion. The final output was a 101X3 matrix where the first column corresponded to torques at joint 1, column two was the torques at joint 2, and the third column was the torques at joint 3. These torques were then written into a csv file and converted to a text file. In order to get the simulation to run, I used the radius and height of my leg to create a model. After defining the COM values, the mass of each link, and applying the initial rotation values to the model I obtained what is seen below in figure 1. By placing a motor at the proximal end of each link that read in torque values from the text file, I was able to create a simulation of my leg moving based on those torques.

Results

The torques required for each joint based on my MATLAB code can be seen below in figure 2. You can see that the torque required is the greatest at the hip and the least at the ankle. These results make sense due to the fact you will need the most input at the hip since it has to move the most amount of mass. When looking at the desired rotation versus the resulting rotation at each the joints, shown in figure

3, you can see that the desired and simulation rotations are most similar throughout joint 1. The figure also demonstrates the idea that the further away you are from the base frame the more variation you have between the simulation and desired values. When comparing all of the velocities, displayed in figure 4, there is the most variation between the desired values and the resulting values at joint 3. Although the torque applied at the foot is much smaller at each timestep, since the foot is such a small mass any input has a great impact on the overall rotation and velocity.

Discussion

Inside of my functions, I had to make several assumptions in order to run the model. In the outward recursion algorithm, I assumed the base was not rotating so the initial rotational velocity (w) and angular acceleration (wd) was zero in the X, Y, and Z directions. In the inward recursion algorithm, I assumed that there were no outside forces (f) or torques (n) that were placed on the end link meaning these were also zero in the X, Y and Z directions. One of the major limitations of the model is that it is unable to account for prismatic joint. Another is that I assumed the base frame velocity was zero however, when looking at the desired values you can see that none of the velocities start at zero.

Overall there are very large errors when using the torque inputs at each joint. You are able to see the large variation between the desired rotation and desired velocity values and the corresponding simulation values. These errors are apparent at the earlier timesteps and increase as time goes on. The future of these devices is going to rely heavily on obtaining ways to quickly adjust the amount of torque that is put into the device. In doing so this will allow for increased safety by allowing users to account for obstacles better while allowing for quick motions to be performed. Lower-limb exoskeletons will be a key technology in helping those affected by disease, trauma or aging maintain their independence and increase their quality of life.

References

[1] J. J. Craig, Introduction to Robotics: Mechanics and Control, Upper Saddle River: Pearson Education, Inc., 2005.

[2] S. Agrawal, S. Banala, S. Kim and J. Scholz, "Robot Assisted Gait Training with Active Leg Exoskeleton (ALEX)," *IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING,* vol. 17, no. 1, pp. 2-8, FEB 2009.

[3] B. Ruoal, S. Rafique, A. Singla, E. Singla, M. Isaksson and G. Virk, "Lower-limb exoskeletons: Research trends and regulatory guidelines in medical and non-medical applications," *SAGE Journals,* vol. 14, no. 6, DEC 2017.

[4] W. Huo, S. Mohammed, J. Moreno and Y. Amirat, "Lower Limb Wearable Robots for Assistance and Rehabilitation: A State of the Art," *IEEE SYSTEMS JOURNAL,* vol. 10, no. 3, pp. 1068-1081, SEP 2016.

<u>Figures</u>

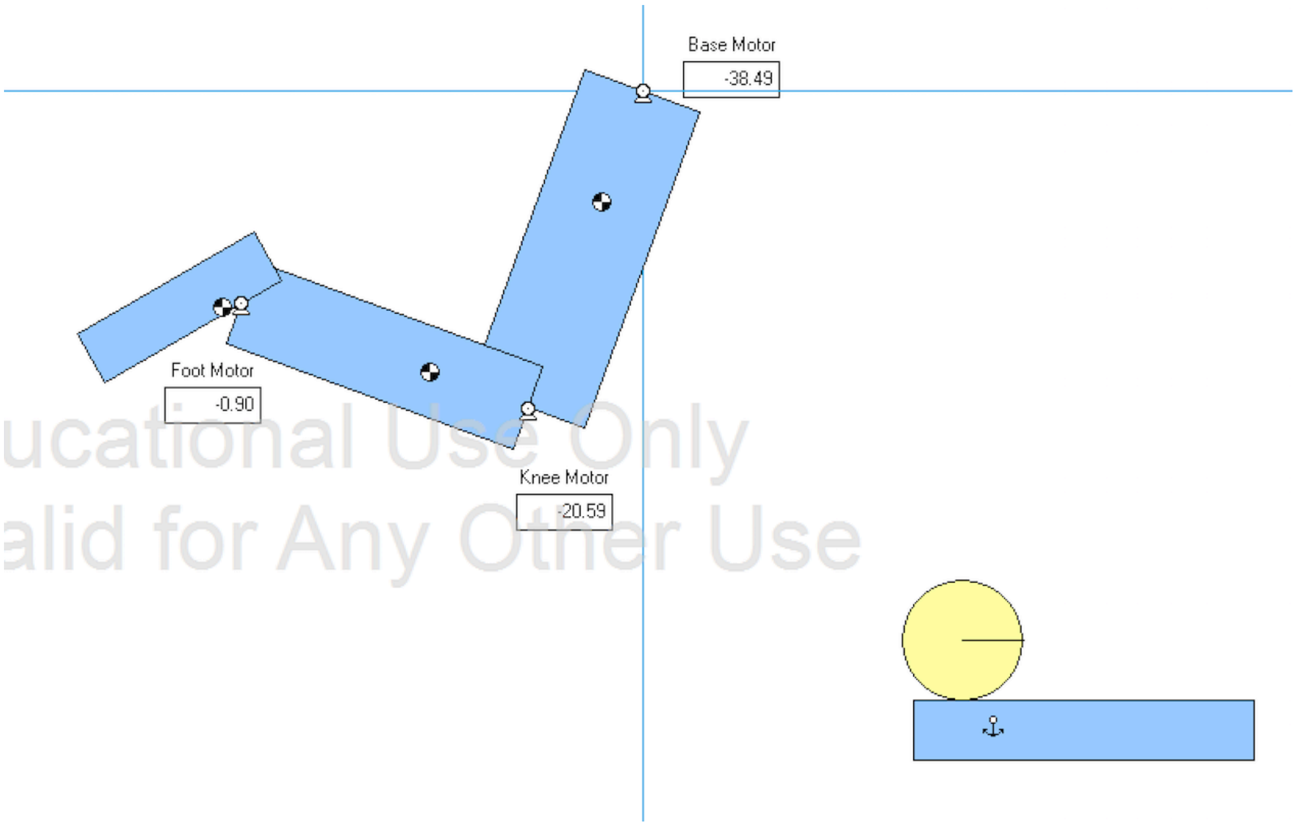

Figure 1: Model with my proportions for the upper leg, lower leg, and foot, showing the motors at each joint and ball held up by small platform, initial rotations are

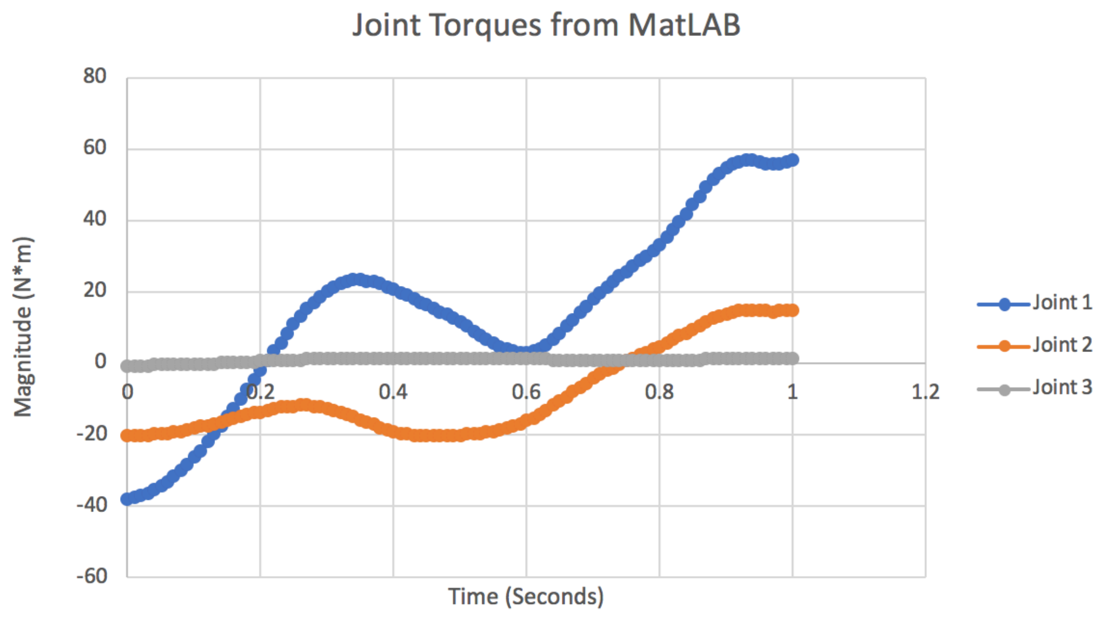

Figure 2: Torque values obtained at each joint every 0.01 seconds for 1 second all displayed on the same graph

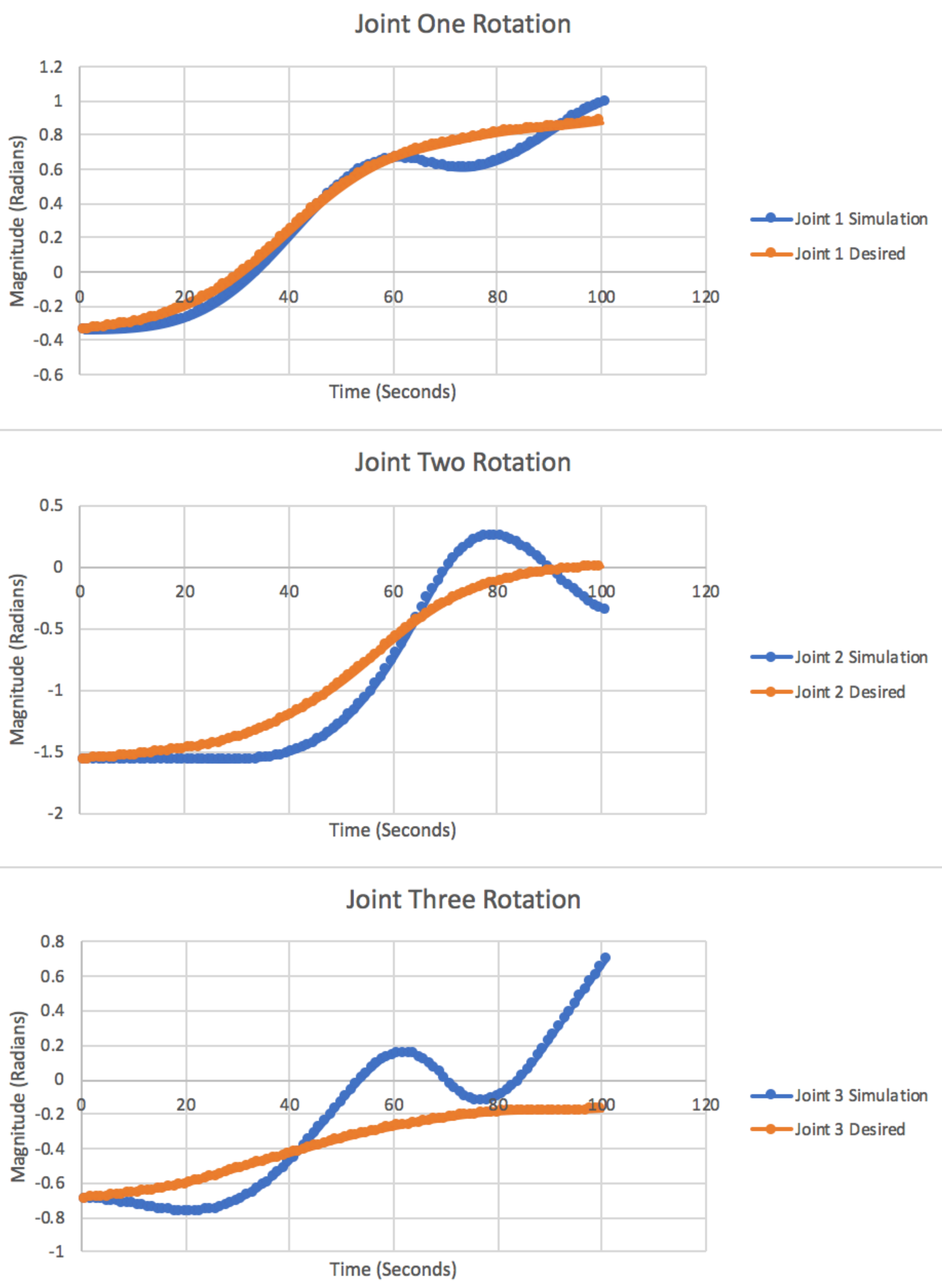

Figure 3: Desired rotation at each joint versus the rotation obtained from simulation when inputting torques at each timestep at each joint

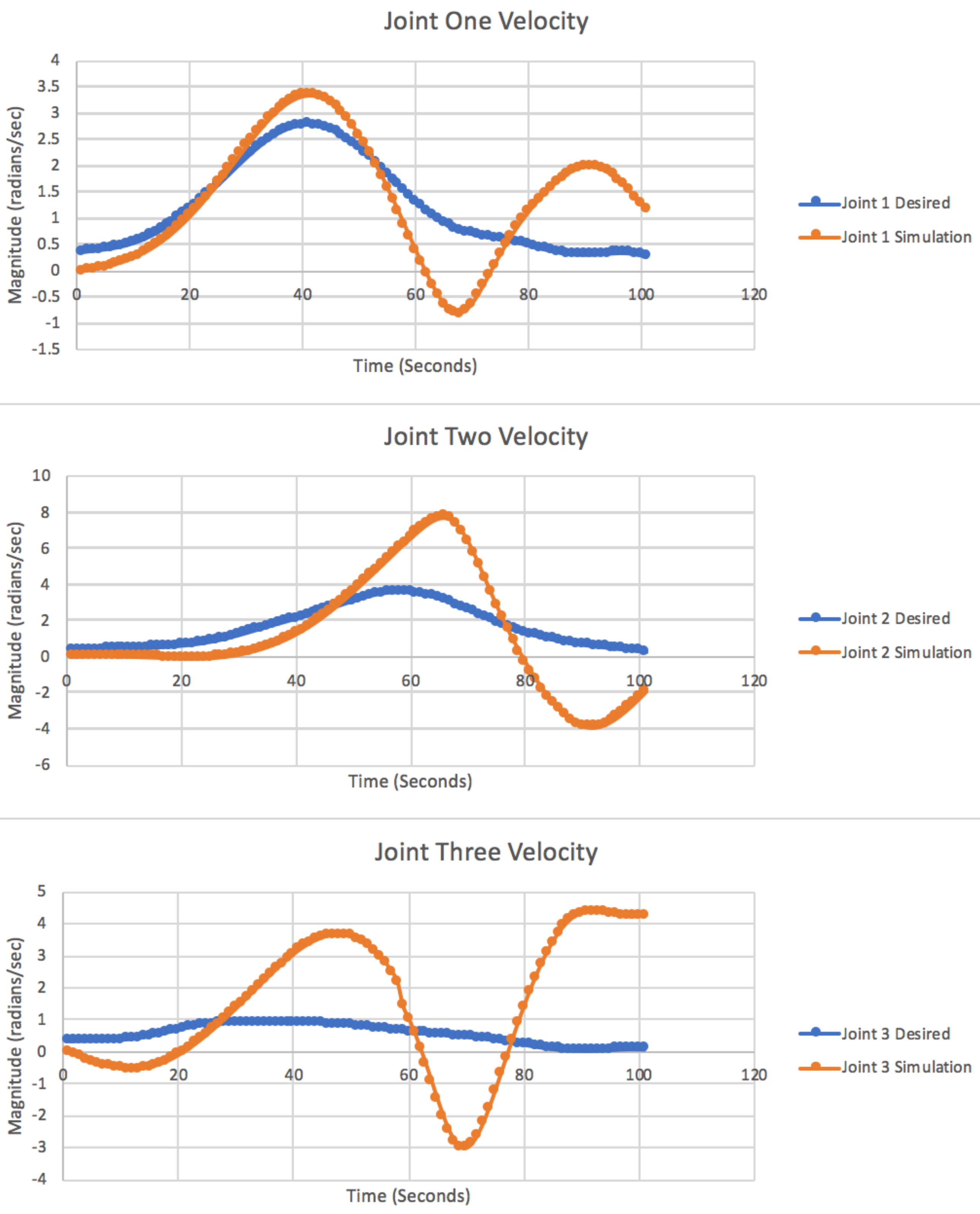

Figure 4: Desired velocity at each joint versus the velocity obtained from simulation when inputting torques at each timestep at each joint