ESA Rocky Project

Simrun Mutha
Patrick Ogunbufunmi
Tolulope Oshinowo

Date: 3/22/2021

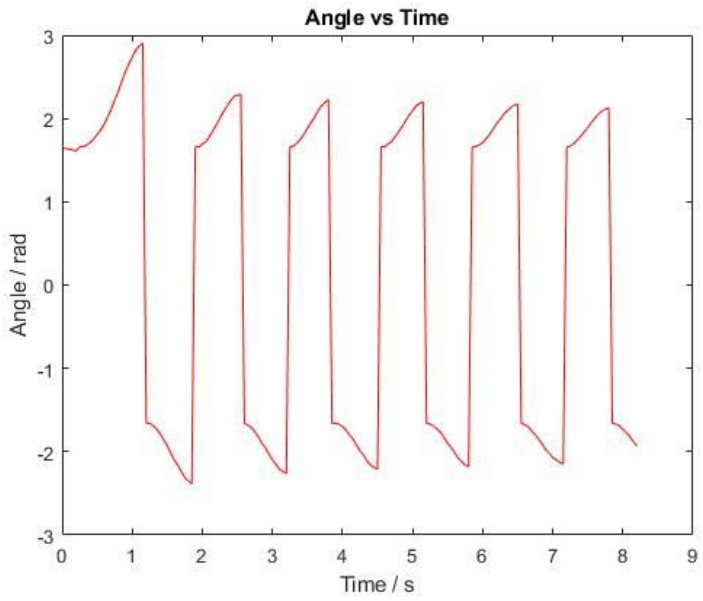
Introduction

Over the past half of a semester we have spent time learning about how real world phenomena can be implemented using a series of different response systems. Using the theoretical experience that we had acquired from applications such as modeling dampening gear and electromagnetic circuits, we were able to begin implementing a real-world application for this project. The goal of this project was to determine how to best balance an inverted pendulum controlled by a motor to stay balanced based on a feedback loop control system. In this paper we will be relaying the inner workings of our project's results accompanied with our group's reasoning and analysis.

This report is laid out to walk the reader through the development process so that one can feel as though they too were a part of our group. First we will talk about how we found the natural frequency and effective length of the pendulum and then we will move into talking about the motor parameters. After that we will move into talking about our base and enhanced model for the system as well as how and why we chose system parameters and poles. Finally we will end this report with a few takeaways and a few ideas for this project could be taken in the future.

Finding natural frequency

In order to find the natural frequency of the system, we did a short procedure with the pendulum. We ran code on the pendulum that prints out the angle in radians every 20 milliseconds. We took off the wheels of the pendulum and held it by its axle and swayed it back and forth for about 8 seconds. Using the serial monitor, we plotted the angle data received from the gyroscope against time. The figure below shows us angle vs time:



Using this figure we were able to find the period by taking the difference between two consecutive peaks in seconds which we got as 1.25 seconds.

The natural frequency is $\frac{2\pi}{period}$ which we calculated to be 5.0265 rad/s.

From this we were able to calculate the effective length of the system by using the equation $L = \frac{g}{w_n^2} = 0.3879\ m.$ Converting to inches, the effective length is 15.27 inches which makes sense given the size of the pendulum.

Finding Motor Constants

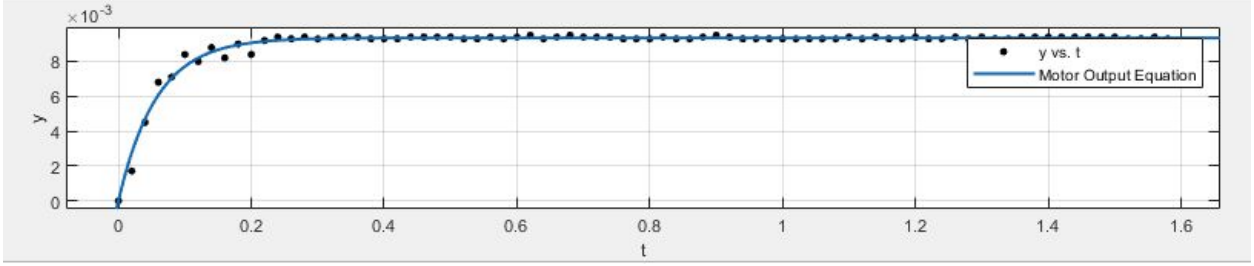
We knew that the transfer function of the motor is

$$M(s) = \frac{ab}{s+a} \text{ where } b = K \text{ and } a = \frac{1}{\tau}.$$

Assuming a step input, by taking the inverse Laplace of the above equation, we will get an equation in the form:

$$M(t) = b - be^{-at}$$

To determine the motor constants, we ran code through Arduino IDE to run both the wheels for 3 seconds and we collected the left and right wheel velocities over that period of time which was in cm/s. Using this data, we used MATLAB curve fitting tool to fit it to the equation for the motor which is in the form $b - be^{-at}$. The figure below shows the raw and data and the curve fit:



Based on the equation of this curve, we got the following values:

$$a = 17.84 \text{ and } b = 0.009341$$

The Base Model

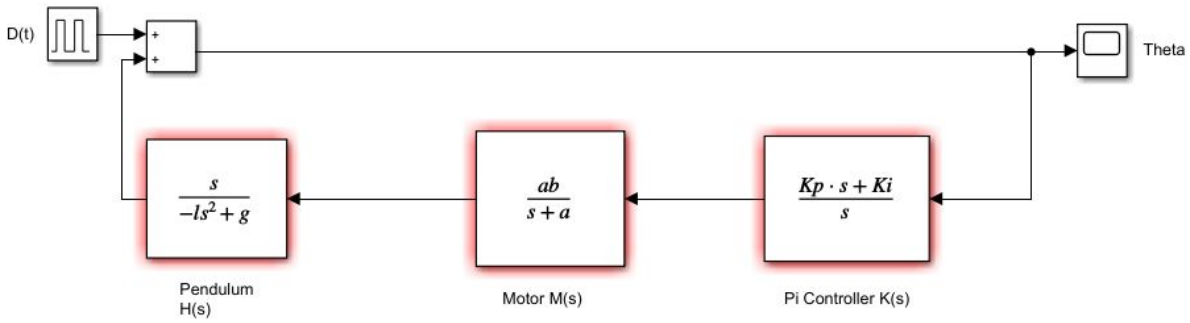
Considering that we wanted the Rocky robot to maintain it's upright balance with some degrees of freedom, we figured that we would need a system that would oscillate in a way that could continuously correct itself. Knowing this we decided to go with an underdamped system consisting of mostly real poles and a complex conjugate pair. Here we will elaborate on how we fleshed out this concept.

The inverted pendulum itself can be described by the following transfer function:

$$\frac{\Theta(s)}{V(s)} = \frac{-s/l}{s^2 - g/l}$$

We added a PI controller,*K(s),* to help make the steady state value of the angle be 0 so that the pendulum will be upright and stable.

We then created a feedback model for the whole system. The input to the system is a disturbance which is shown by D(t). The output of the system is Θ(t) is the angle of the pendum. The following block diagram describes the system we used:

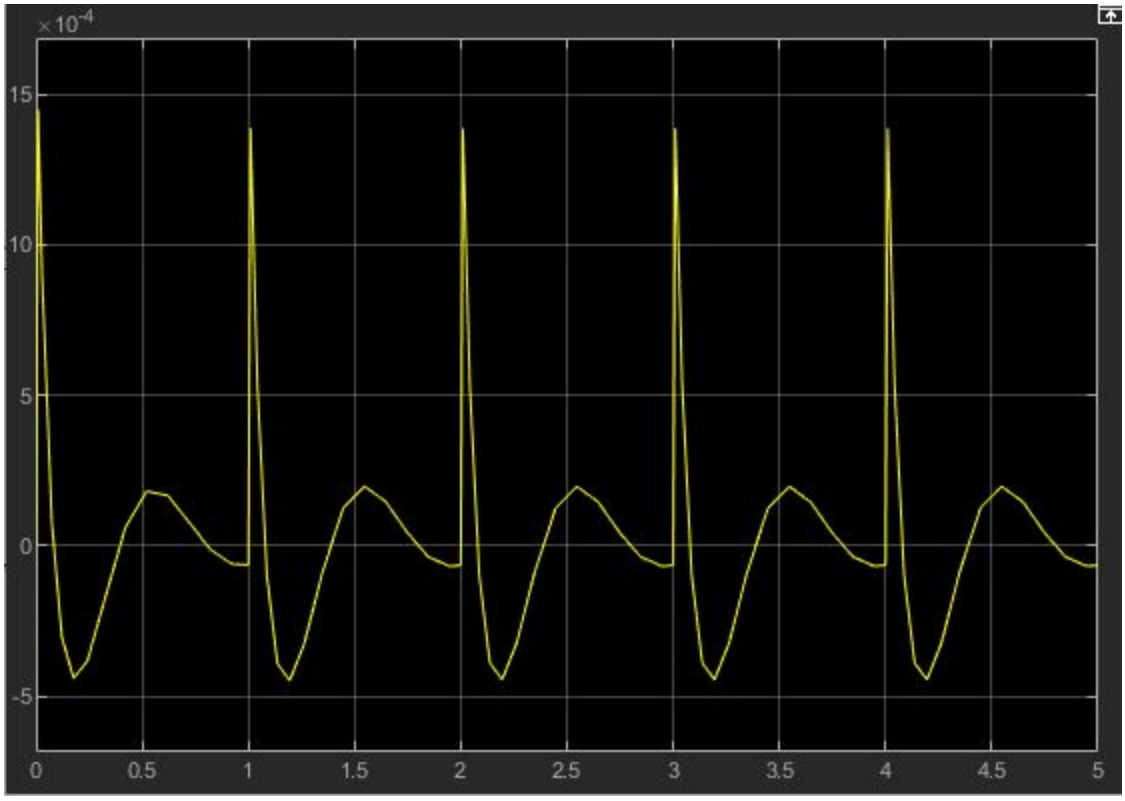


This entire block diagram can be described by the following transfer function:

$$\frac{\Theta(s)}{D(s)} = \frac{1}{1 - M(s)K(s)H(s)}$$

K(s), M(s) and H(s) are depicted in the above block diagram.

The input to this system was a disturbance. The three blocks are the PI controller, the motor transfer function, and the pendulum transfer function. Running this system with a pulse generator input to simulate a disturbance, we got this as the output:



This graph shows that when there was a disturbance, the system would self correct and center back around zero.

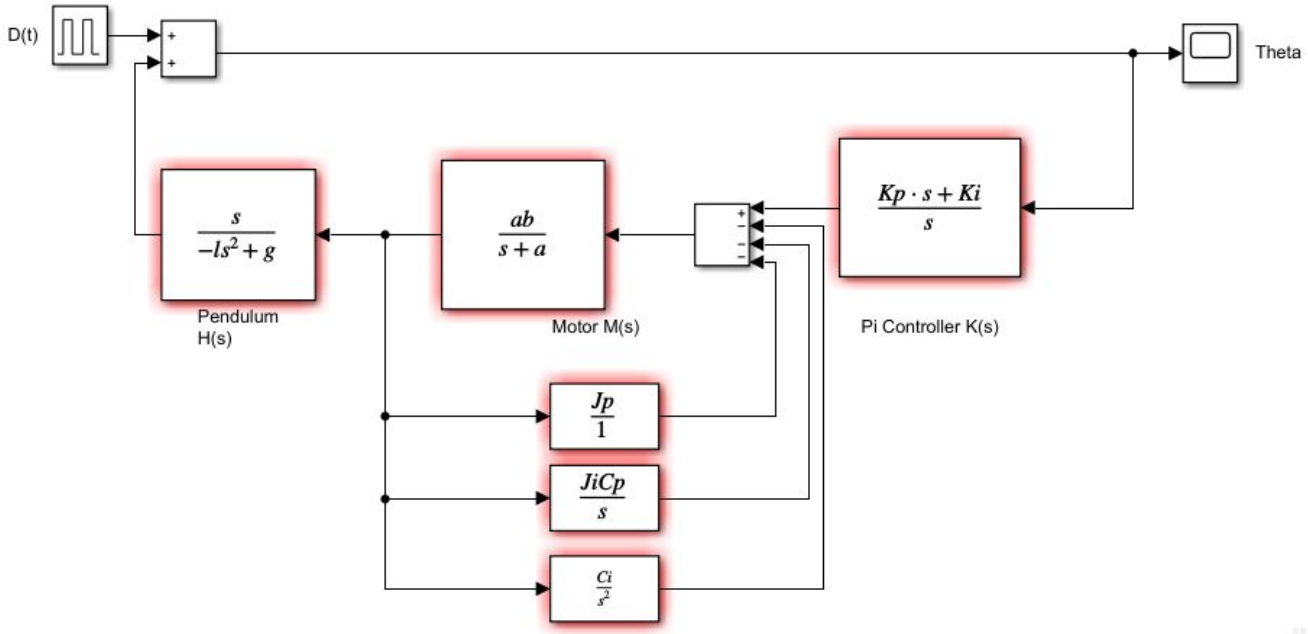
The Enhanced Model

In addition to our base model we realized that not only did the angle of the inverted pendulum needed to be corrected for, but the wheel speeds that allowed for balance needed to be as well. Knowing this our system went from having 3 poles to having 5 poles, adding Ji and Jp variables to our block diagram and its respective equation.

As mentioned in the previous paragraph our original system, its block diagram, and its equation consisted of only one PI controller. While this can make the steady state value of the angle

approach 0, looking at the transfer function of the $\frac{X(s)}{D(s)} = \frac{M(s)K(s)}{s(1 - M(s)(K(s)H(s)}$, we saw that there will be a pole at the origin which will result in an unstable system.

To rectify this issue, we added another PI controller to help the steady state value of the position reach 0. We also added another PI controller around the motor in order to make the steady state velocity to be 0. The following block diagram shows how this was implemented:



This block diagram gave us the transfer function described below:

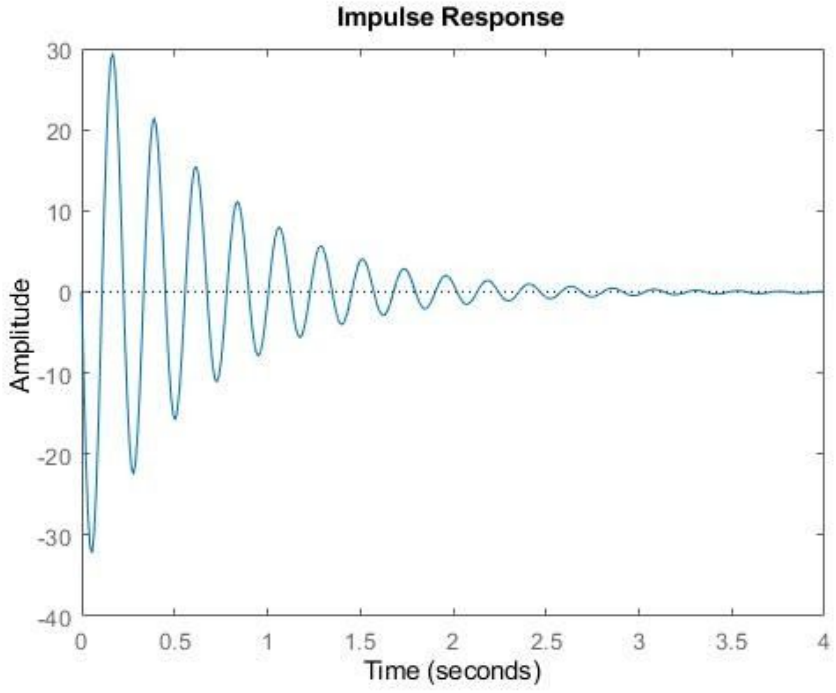
$$\frac{\Theta(s)}{D(s)} = \frac{ls^5 + (al + Jp\ abl)s^4 + (J_i abl - g)s^3 + (C_i abl - ag - J_p abg)s^2 - J_i abgs - C_i abg}{ls^5 + (al + J_p\ abl)s^4 + (K_p ab - J_i abl - g)s^3 + (K_i ab + C_i abl - ag - J_p abg)s^2 - J_i abgs - C_i abg}$$

The denominator of this transfer function shows us that this is a system of the 5th order and therefore there will be five poles. Once we found the five poles we could compare the denominator of this transfer function against the polynomials to solve for the values of Kp,Ki, Jp,

Ji and Ci. The next step in this process was to choose five poles that would work with our system.

Choosing Poles

We wanted to choose poles that brought the system to a stable steady state of 0, with as little amplitude as possible, along with a smooth slow gradual decay. To that end, we had a 5th order Laplace transfer function, with 5 different poles. We opted to go for 2 complex poles to produce oscillation, as well as complex values to produce enough oscillation to allow a gradual decay, and we found that $-1.5+28i$ and $-1.5-28i$ were suitable complex conjugate poles. To achieve a low amplitude, we went with poles with a low magnitude. To that end, we chose -0.5, -1.5 and -2.5 as our poles. They all fell on the left hand plane, in order to prevent instability of the system. The impulse response plot can be found below:



Determining Control Parameters

Once we chose the poles, we were able to find values for Kp, Ki, Ji, Jp and Ci based on the target equation defined by poles. This is the target expression that we got:

$(s + 0.5)(s + 1.5)(s + 2.5)(s + 1.5 - 28i)(s + 1.5 + 28i)$.

By expanding and comparing this expression with the denominator we were able to find values for all our PI controllers as given below:

Jp = -62.0486

Ji = -1074.1

Kp = 2350.5

Ki = 8857.5

Ci = -349.8038

By using these values, we were able to balance the Rocky robot.

Code
https://github.com/toluooshy/Engineering-Systems-Analysis-Rocky-Project