Rocky

Amanda Chang, Jun Park, and Pauline Petersen

March 2024

1 Introduction

This project focuses on the steps taken to achieve a self-balancing inverted pendulum. We used a 32U4 Balancing Robot known as Rocky to stabilize the system.

2 Parameter Identification: Motors

Our Polulu Balboa 32U4 Balancing Robot (Rocky) has DC motors that can be represented as a first-order system where its transfer function $M(s)$, which relates the velocity control signal $V_c(s)$ and the actual velocity output $V(s)$, is:

$$M(s) = \frac{V(s)}{V_c(s)} = \frac{\frac{K}{\tau}}{s + \frac{1}{\tau}} \tag{1}$$

We multiplied M(s) by $V_c(s)$ to find an equation for the actual velocity of the Rocky in the Laplace domain, denoted $V(s)$. We wanted to derive an equation applicable to a real-life velocity-time measurement with a provided velocity signal step input of 300, since that was what was in our calibration script (detailed below). The Laplace transform of a magnitude 300 step input is $V_c(s) = \frac{300}{s}$.

$$V(s) = V_c(s)M(s) = \frac{300}{s} * \frac{\frac{K}{\tau}}{s + \frac{1}{\tau}} \tag{2}$$

We then took the inverse Laplace transform of $V(s)$ using MATLAB's ilaplace() to determine $V(t)$, shown below. This is the theoretical form of our velocity-time equation.

$$V(t) = 300K(1 - e^{\frac{-t}{\tau}}) = 300K - 300Ke^{\frac{-t}{\tau}} \tag{3}$$

2.1 Motor Constants

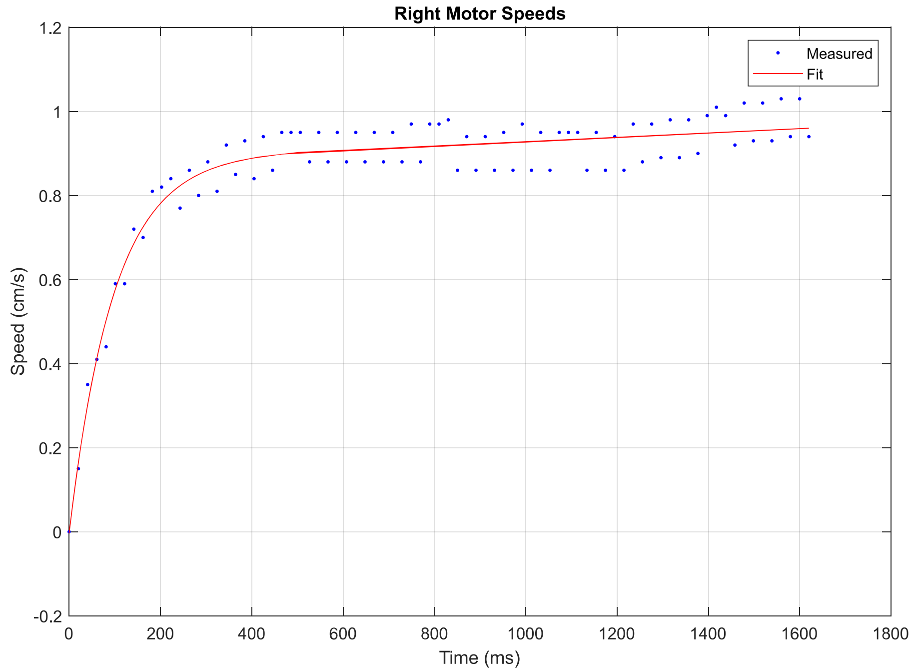

Figure 1: Recorded Right Motor Speed & Fit Curve

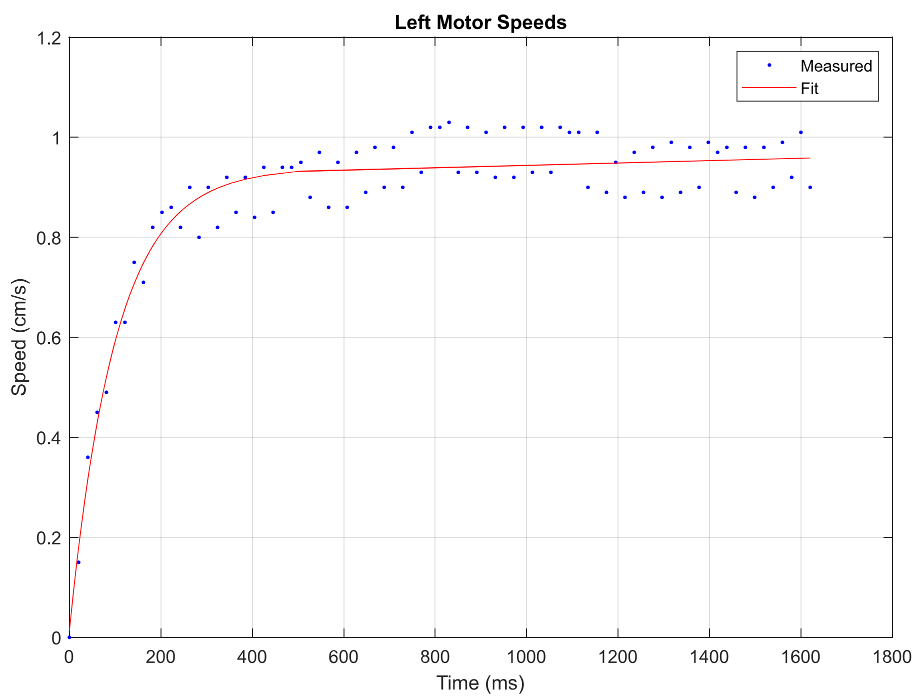

Figure 2: Recorded Left Motor Speed & Fit Curve

To calculate the motor constants K and τ, we ran a calibration script with a step input of 300 as the Rocky's velocity control signal. During the test, both wheels were grounded and its body was gently held upright. While the robot moved forward in response to the step, we made an effort to impart as little force as possible. Using Rocky's internal velocity sensors and its serial port, we sampled the velocities of the two Rocky motors every 20 milliseconds.

To obtain the theoretical fits shown above, we analyzed each of the plots with the curve fit MATLAB tool and this second-degree exponential equation format:

$$f(x) = c_1 * e^{d_1 x} + c_2 e^{d_2 x} \tag{4}$$

Coefficients	Left Motor	Right Motor
c_1	0.9296	0.8849
d_1	$1.904e-05$	$5.032e-05$
c_2	-0.9168	-0.8887
d_2	-0.009922	-0.01033

Table 1: Coefficients are derived from each respective theoretical fits

While our curve fitter gave us fit equations with two exponential terms each, we can observe that the d_1 in each equation is very minimal, essentially making the c_1 constant: independent of the exponential term.

$$V_{step}K - V_{step}K \cdot e^{\frac{-t}{tau}} \tag{5}$$

Therefore, following the velocity-time equation, we can determine that...

$$c_2 = -V_{step}K, d_2 * x = \frac{-t}{tau} \tag{6}$$

K for the left motor is then equated using the coefficient c_1 and the 300 input values used in the calibration code.

$$K = \frac{c_2}{input} = \frac{0.9168}{300} = 3.056 \cdot 10^{-3} \tag{7}$$

τ for the left motor is then equated using the coefficient d_2 and plugged into the curve fit exponential equation

$$\begin{aligned} d_2 * x &= \frac{-t}{\tau} \\ d_2 * t &= \frac{-t}{\tau} \\ \tau &= \frac{-t}{d_2 * t} \\ \tau &= \frac{-1}{-0.009922} = 100.79 \text{ ms} \end{aligned} \tag{8}$$

The right motor is calculated using the same process:

Terms	Left Motor	Right Motor	Average
K	3.10×10^{-3}	2.96×10^{-3}	3.01×10^{-3}
τ (seconds)	0.10079	0.0968	0.0988

Table 2: Coefficients are derived from each respective theoretical fits.

Based on Equation 1, we know that $a = \frac{1}{\tau}$ and that $b = K$. Therefore, based on the averages of the motor parameters, $a = 10.12$ and $b = 0.00301$.

2.2 Natural Frequency and Length of Pendulum Body

To determine the natural frequency, we experimented by putting the Rocky in a pendulum-like motion by swinging it back-and-forth. The recorded response is plotted below:

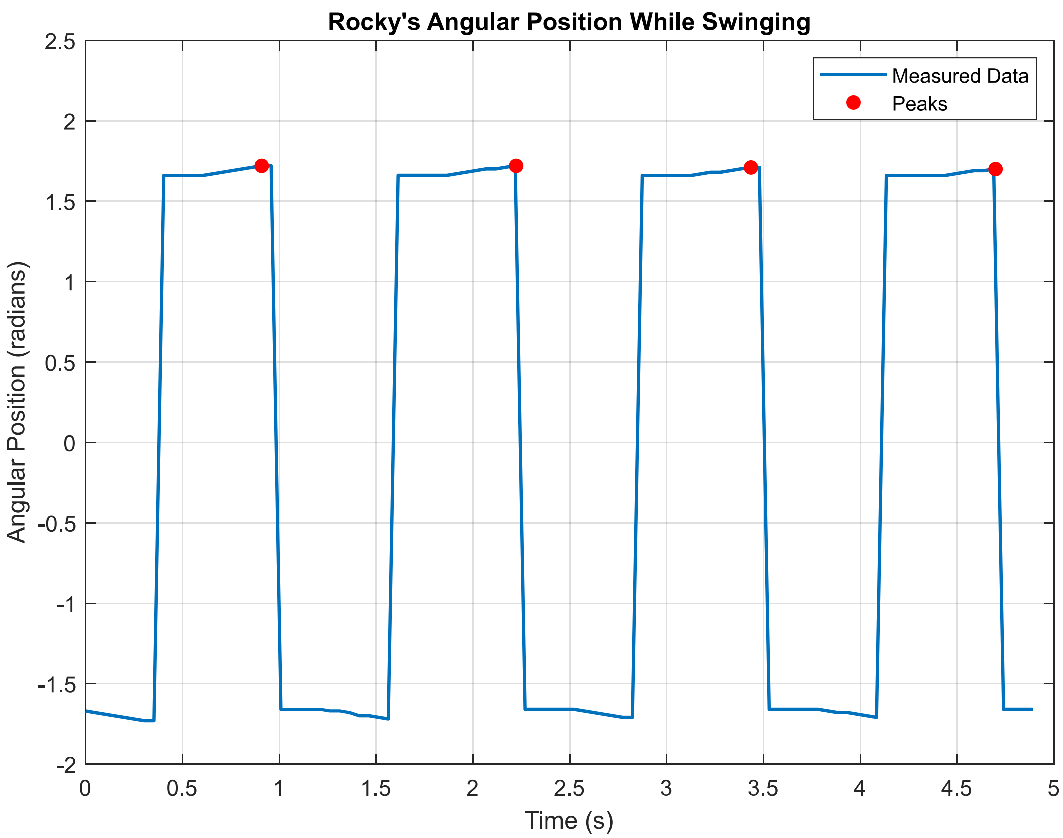

Figure 3: Angular position of Rocky as it is swung back-and-forth in a pendulum-like motion.

To calculate the natural frequency of the Rocky, we first recorded the peaks of our oscillation. Afterwards, we found the time differences between the peaks to find the period.

Finally, the natural frequency is found using the equation:

$$w_n = \sqrt{\frac{g}{l_{eff}}} \tag{9}$$

We use the period to determine the natural frequency of pendulum, which aids us in calculating the effective length of the pendulum body,

$$T = 1.27 \frac{seconds}{cycle} \tag{10}$$

$$w_n = \frac{1}{T} \frac{cycles}{seconds} \tag{11}$$

$$w_n^2 = \frac{g}{l_{eff}} \tag{12}$$

$$l_{eff} = \frac{g}{w_n^2} = 0.40038m \tag{13}$$

3 Initial System

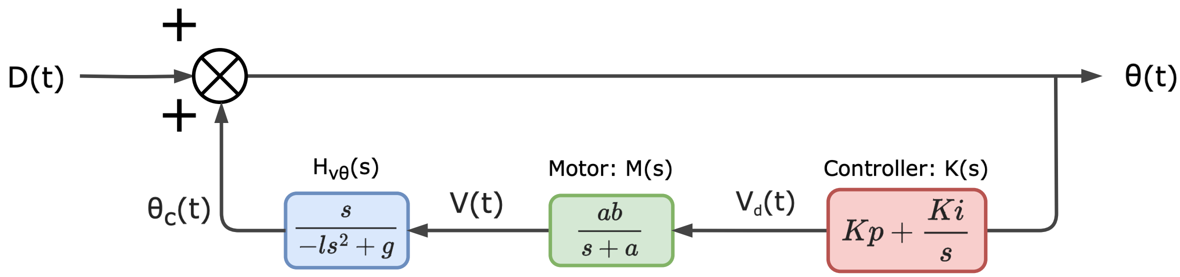

Figure 4: Above is our block diagram of the three pole system. In green is the motor transfer function. In blue is the transfer function representing the dynamics of the **Rocky**, while in red is our transfer function representing the **controller**.

3.1 Pole Values

The poles of a second order system transfer function can be found using the equation

$$p_{1,2} = \underbrace{-\zeta\omega_n}_{\text{Real}} \pm \underbrace{i\omega_n\sqrt{1-\zeta^2}}_{\text{Imaginary}}$$

ζ is the damping ratio and ω_n is the natural frequency of the system. Mathematically, all poles must have negative real components to create a stable system. For our Rocky, we calculated its natural frequency to be 4.9474 radians per second (as explained previously). To balance our robot, we knew we wanted something close to critical damping (ζs near 1). In an ideal world, all poles being critically damped would result in the fastest decay into steady-state (where steady-state is being balanced). However, because of real-world constraints like motor response time and slippage, we thought that the robot would balance best at slightly lower ζ values (a.k.a. slightly underdamped). For our five-pole system with two pairs of complex conjugate poles and one lone pole with no conjugate, we needed three zeta values.

Because the lone pole has no complement and our system's output signal (its movement) is entirely real, it must necessarily have a ζ value of 1 for our system's output to continue to be real. Any other value would cause our lone pole to have an unpaired imaginary component. Thus, $\zeta = 1$ for p_3, which evaluates to $-\omega_n$.

For the first pole pair, denoted $p_{1,2}$, we experimentally determined the ζ_1 value to be 0.9 by slowly incrementing down from 1 until more stable behavior was observed in real life. This pole pair could have a non-1 ζ value and still produce a real output because they are in a pair.

Pole	Expression	Value
p_1	$-\zeta_1\omega_n + \omega_n\sqrt{1-\zeta_1^2}i$	$-4.4527 + 2.1565i$
p_2	$-\zeta_1\omega_n - \omega_n\sqrt{1-\zeta_1^2}i$	$-4.4527 - 2.1565i$
p_3	$-\omega_n$	-4.947

Table 3: Poles 1 and 2 are part of a complex conjugate pair. The expression used to calculate each of the poles is shown in the expression column.

3.2 Calculation of Control Constants

Control Parameter	Value
K_i	4850.8
K_p	1222.9

Table 4: The two control parameters shown above were calculated by plugging in our three pole values into the provided MATLAB script `Rocky_closed_loop_poles.m`.

3.3 Simulink and Response Plots

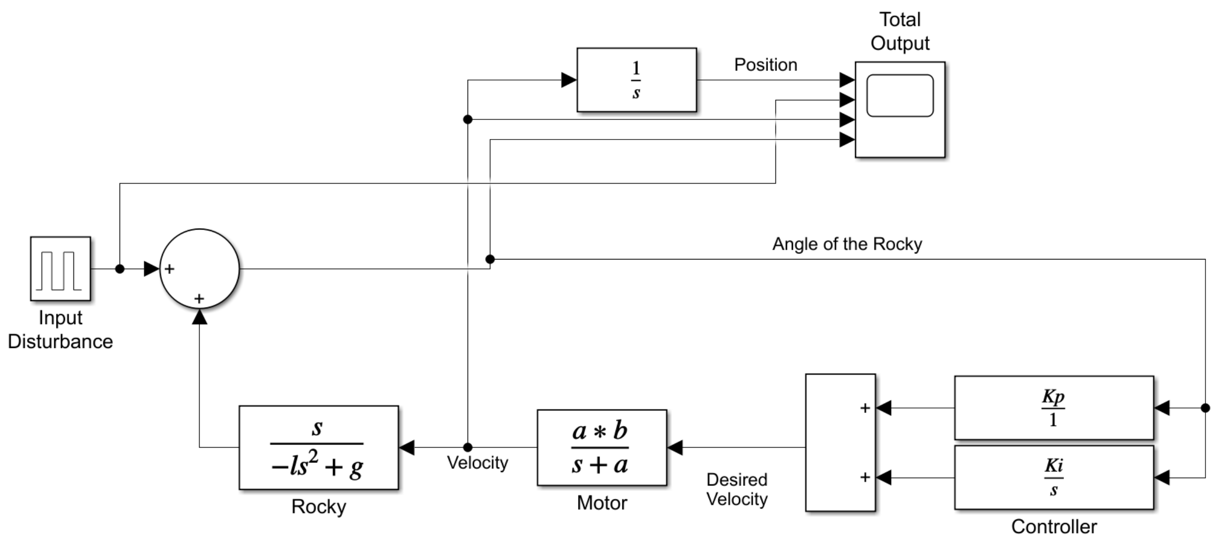

Figure 5: Simulink model of three-pole control system for an inverted pendulum on a cart

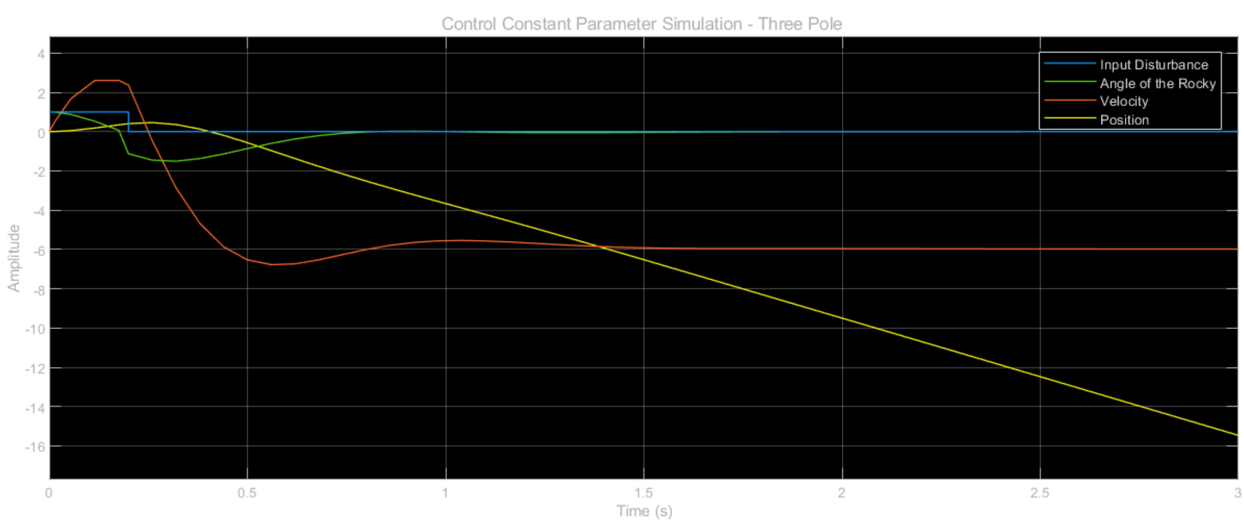

Figure 6: Simulation results showing the position, angle, and velocity of a three-pole control system for an inverted pendulum on a cart according to the input disturbance pulse. We can observe that, while the system stabilizes the angle of the Rocky, the position is unstable. Therefore, now we try to implement two additional poles on our Rocky control system to account for position correction.

4 Stationary Balancing System

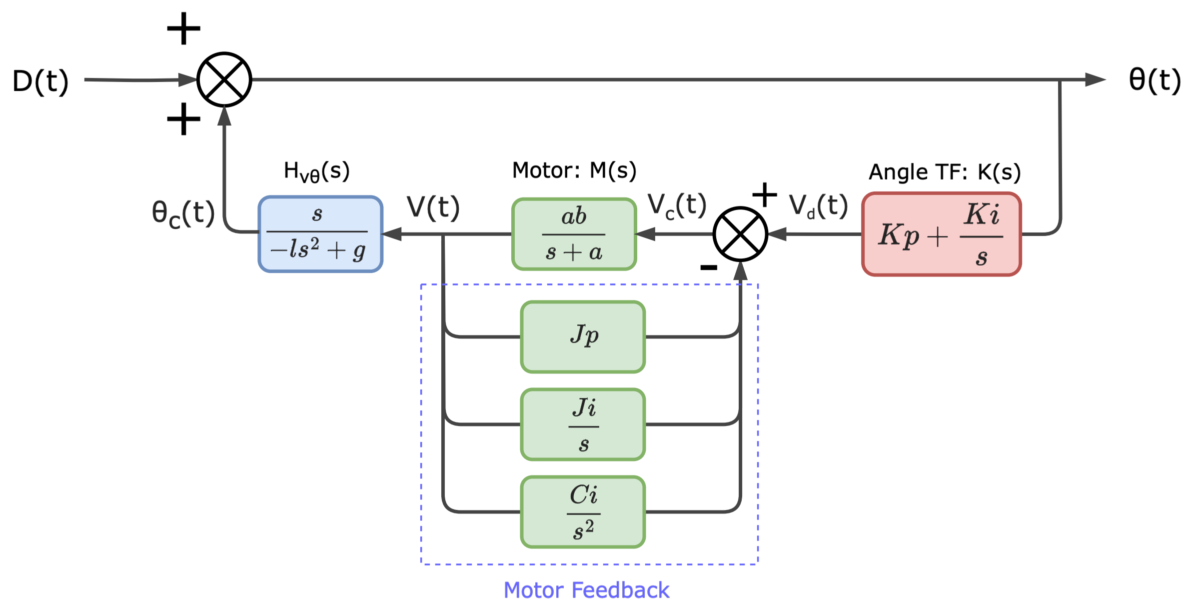

Figure 7: Above is our block diagram of the five pole system. In green is the **motor loop**, where M(s) is the forward path block and J_p, $\frac{J_i}{s}$, and $\frac{C_i}{s^2}$ are the feedback path blocks which are added together and then subtracted from V_d to get V_c. In blue is the transfer function representing the dynamics of the **Rocky**, while in red is our transfer function representing the **controller**. The larger system from D(t) to $\theta(t)$ is represented by $HC_{loop}(s)$.

By adding two additional poles, we were able to add a motor feedback component to the Rocky system. This allowed us to control the position of Rocky as well as the angle, rather than solely angle. The previous model assumed that $V_d(t)$, or desired velocity, would be equal to the velocity control signal $V_c(t)$. Our five pole system did not make this assumption and was thus significantly more stable than our three pole system. It incorporated three new control parameters: J_p, J_i, and C_i, which represent proportional velocity, joint integral velocity/proportional position (J_i, aka J_{iC_p}), and integral position. The actual values of these control parameters are calculated in section 4.2.

4.1 Pole Values

Similar to the three-pole system, we experimentally determined the ζ_2 value to be 0.84 for the second pole pair $p_{4,5}$. Poles 1, 2, and 3 remained the same for the same reasons explained in section 3.1 (nearly critically damped, slightly underdamped for the $p_{1,2}$ pair, and critically damped for p_3).

Pole	Expression	Value
p_1	$-\zeta_1\omega_n + \omega_n\sqrt{1-\zeta_1^2}i$	$-4.4527 + 2.1565i$
p_2	$-\zeta_1\omega_n - \omega_n\sqrt{1-\zeta_1^2}i$	$-4.4527 - 2.1565i$
p_3	$-\zeta\omega_n + \omega_n\sqrt{1-\zeta^2}i$	-4.9474
p_4	$-\zeta_2\omega_n + \omega_n\sqrt{1-\zeta_2^2}i$	$-4.1558 + 2.6844i$
p_5	$-\zeta_2\omega_n - \omega_n\sqrt{1-\zeta_2^2}i$	$-4.1558 - 2.6844i$

Table 5: Poles 1 and 2 are part of a complex conjugate pair, as are poles 4 and 5. The expression used to calculate each of the poles is shown in the expression column.

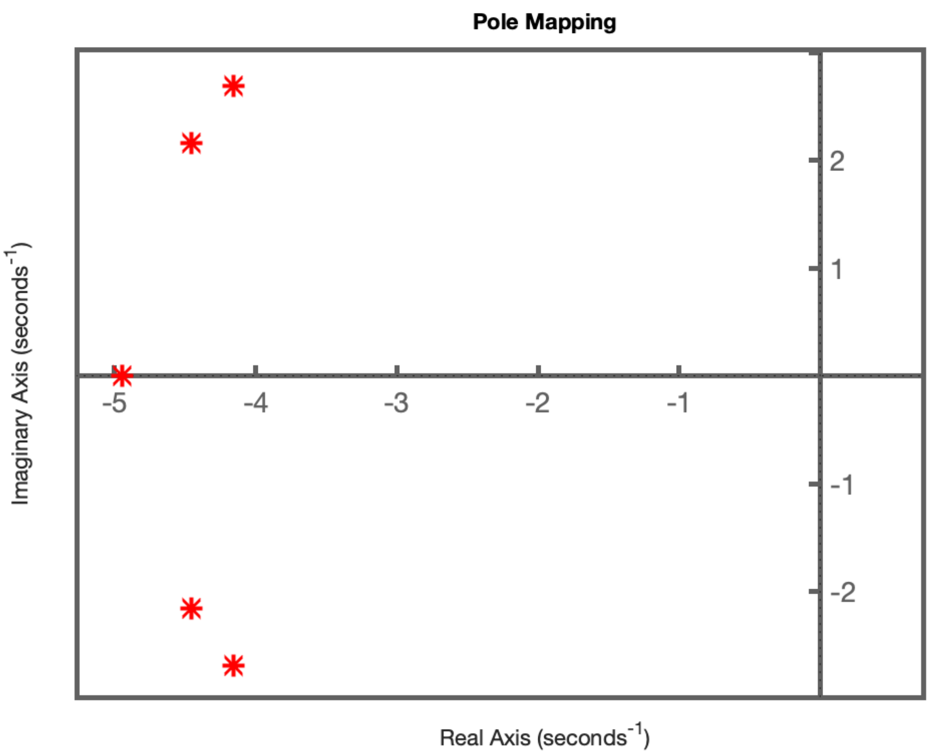

Figure 8: Here is a graph representing the locations of our five poles. As you can tell, all of our poles fall to the left of the imaginary axis and have negative real components.

4.2 Calculation of Control Constants

Control Parameter	Value
K_i	22264
K_p	4497.8
J_{iC_p}	-3596
J_p	395.4
C_i	-3971.3

Table 6: The five control parameters shown above were calculated by plugging in our five pole values into the provided MATLAB script `Rocky_closed_loop_poles.m`.

4.3 Simulink and Response Plots

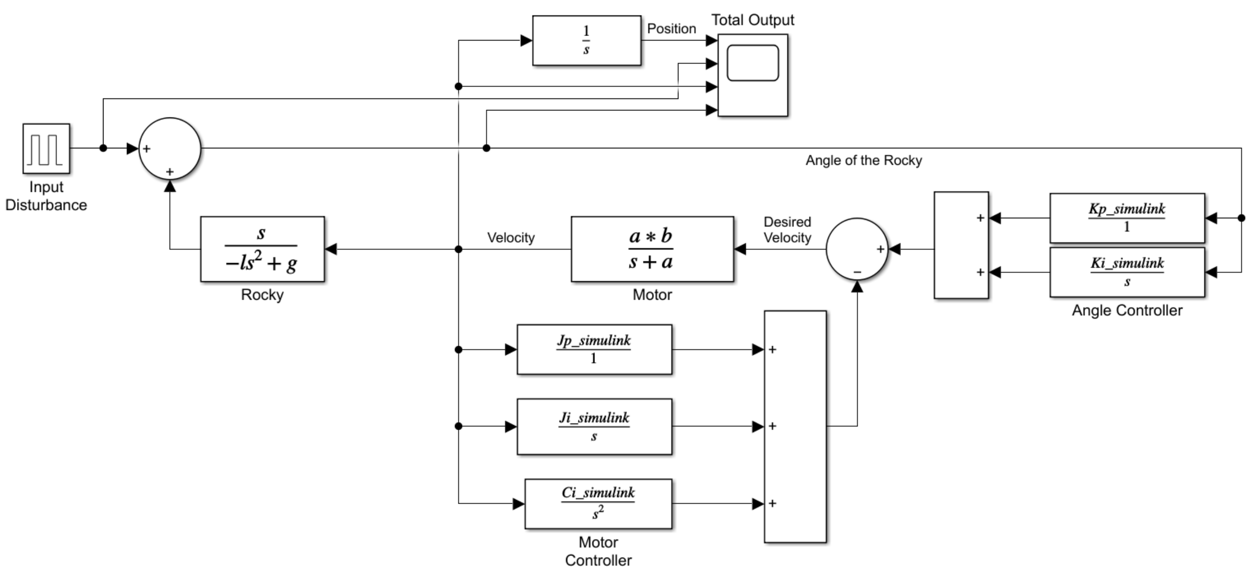

Figure 9: Simulink model of five-pole control system for an inverted pendulum on a cart

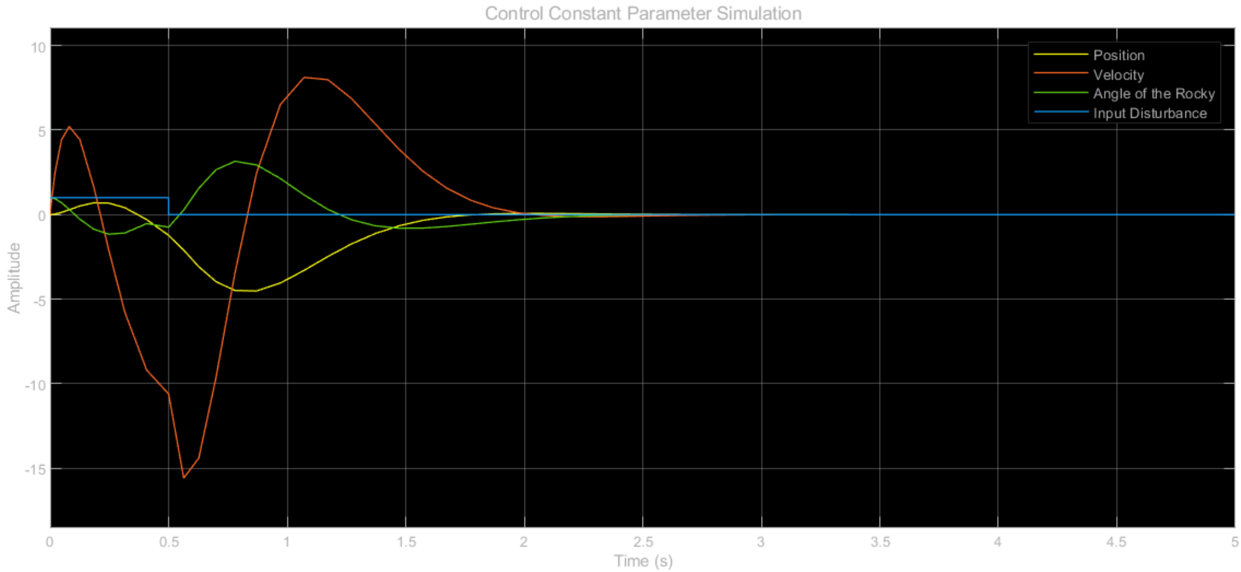

Figure 10: Simulation results showing the position, angle, and velocity of a five-pole control system for an inverted pendulum on a cart according to the input disturbance pulse. The Rocky is visibly much more stabilized than the three-pole control system. The system stabilizes in around two seconds, which closely relates to how we want our Rocky to respond to disturbances.

5 Video

Check out our **demo-video** featured on the Olin College Youtube Channel!

6 Code

Check out our **Github Repo** for all of the code and data we used for this project!