

Inverted Pendulum Project Report

Olivia Jo Bradley and Nabih Estefan

Engineering System Analysis

Olin College of Engineering

3/20/2021



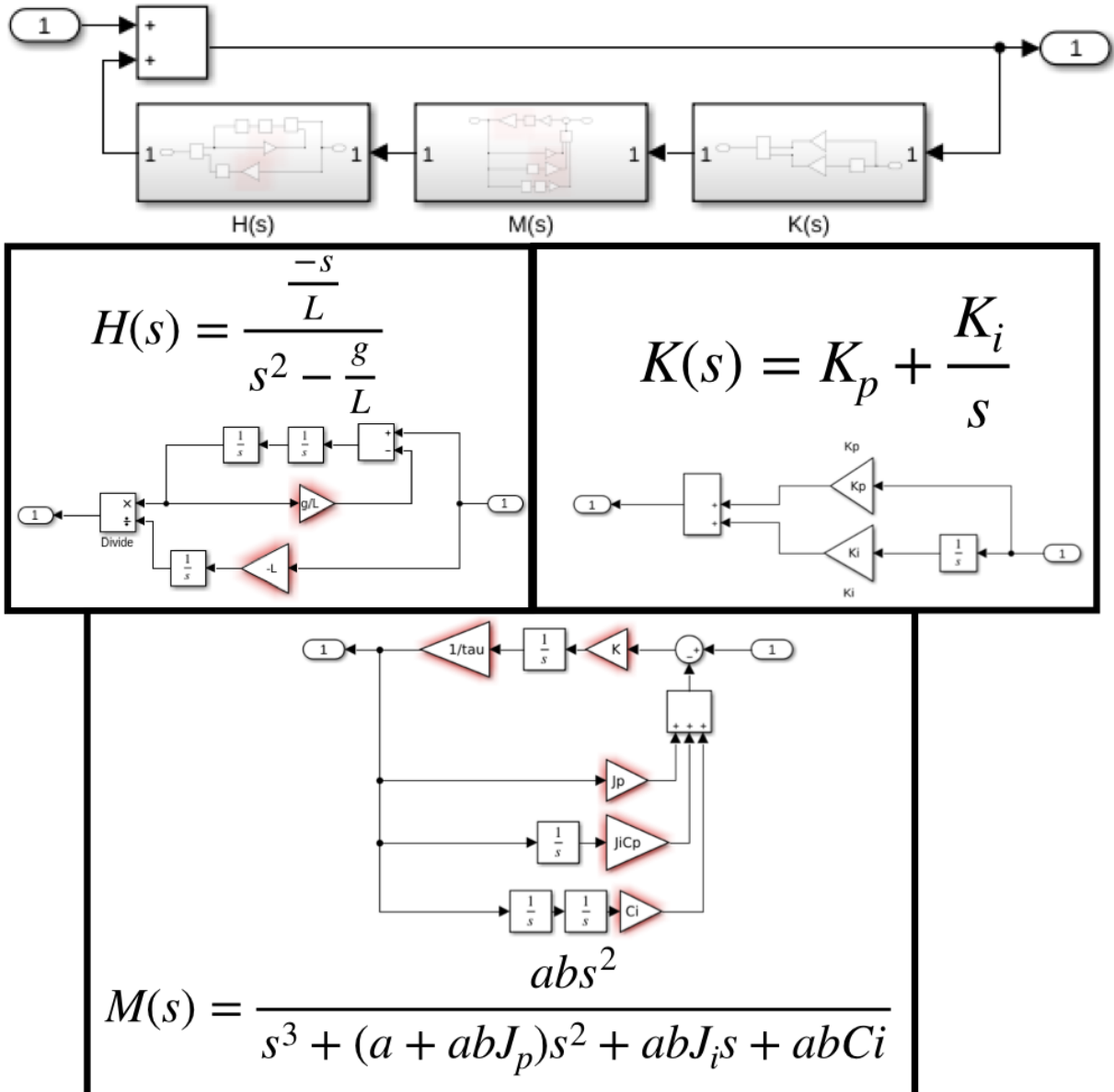
Introduction

For this project, we used a 32U4 Balancing Robot Kit from Pololu Robotics and Electronics, also known as Rocky, to mimic a balancing inverted pendulum (seen above). Below, we will explain how we created our control system to make our Rocky balance in a single location.

Base model & Block Diagram

Below you can see the base model, and the zoom into the specific parts we abstracted. As you can see there are 3 distinct parts of the model:

- $H(s)$ which is the main transfer function between angle and velocity.
- $K(s)$ which is the main steady-state error function.
- $M(s)$ which is the Motor transfer function with a feedback loop to change expected velocity to actual velocity.



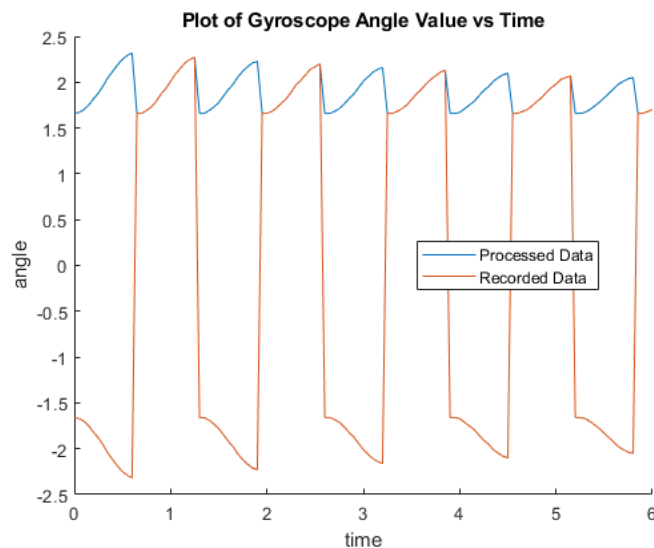
As you can see above, there are multiple variables which are: g , L , τ , K , K_p , K_i , J_p , J_i , and C_i . g is the value of gravity, which is 9.81 m/s. The other values we find throughout the process of our calculations which you will see below. L is the effective length, and the process to get it can be seen in the "Natural Frequency and Effective Length" section. K and τ are Motor Constants, and you can see the process to get them in the "Motor Constants" section. Finally, K_p , K_i , J_p , J_i , and C_i are the steady-state variables which are calculated in the "Controlled Parameters" section.

Finally, each of the main sections (H(s), K(s), and M(s)) has a transfer function equation defined for each. These can be seen in the image above in each of the breakdown for each subsystem.

Natural Frequency and Effective Length

To find effective length, we first had to collect frequency data using the provided Rocky_Gyro_Calibration code. We carried our Rocky by the axles, and recorded the gyroscope data as it rocked back and forth.

To make the data look more like a sine curve, we absolute valued the data. Below is a graph of the data before and after the absolute value.



From this data, we are able to find the natural frequency using Fast Fourier Transforms (FFT), which plots the data we collected in the frequency domain.

$$\omega_n = \sqrt{\frac{g}{l_{eff}}}$$

With the equation we are able to find the effective length of our inverted pendulum, which we then subtracted from the total length, to get about 17 inches.

Motor Constants

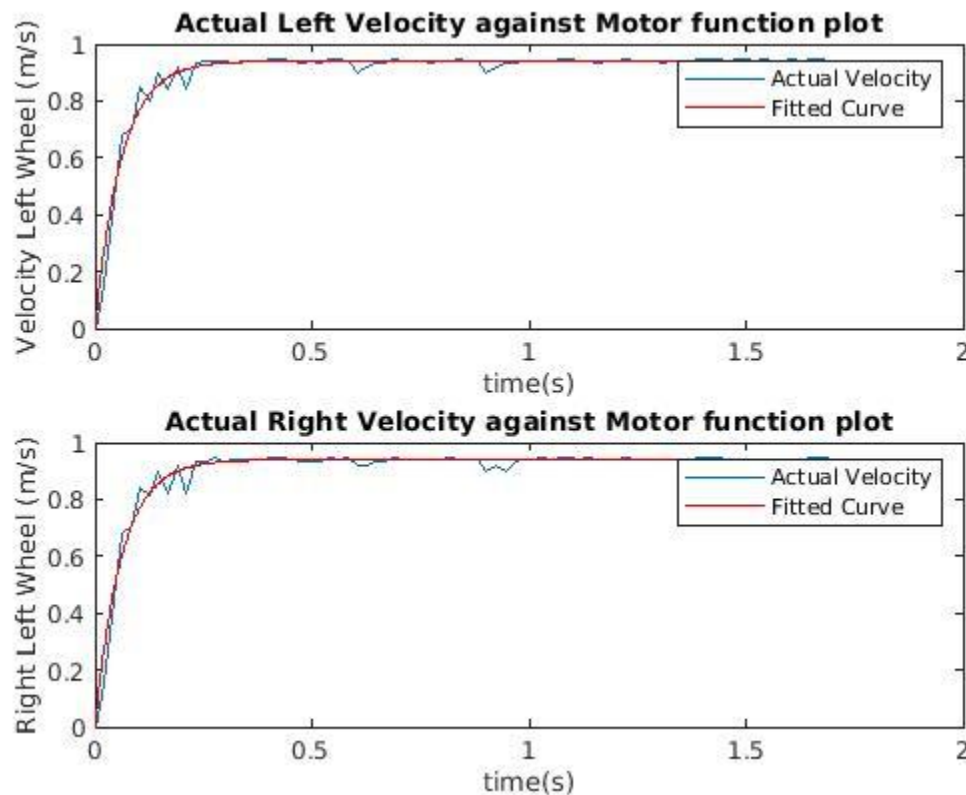
For this section, we need to calculate two separate motor constants: K and tau. To calculate these, we measured the actual values of the left and right wheels when we give them a 300 input and plotted how those wheels reached that speed. Then, we used MATLAB's fit() function to find a function that would fit this graph. For the base of this function, we used the motor

$$M_{base}(s) = \frac{\frac{K}{\tau}}{s + \frac{1}{\tau}}$$

function we have which is . Which. By inverse laplacing, we can turn into a

function of time: $M_{base}(t) = 300K(1 - e^{\frac{-t}{\tau}})$ where 300 is the value of our input when testing, and K and tau are the values we are fitting for.

The graphs for these fits look like this:



And the K and tau values (when averaged between both wheels) end up being:

K = 0.0031, tau = 0.0602.

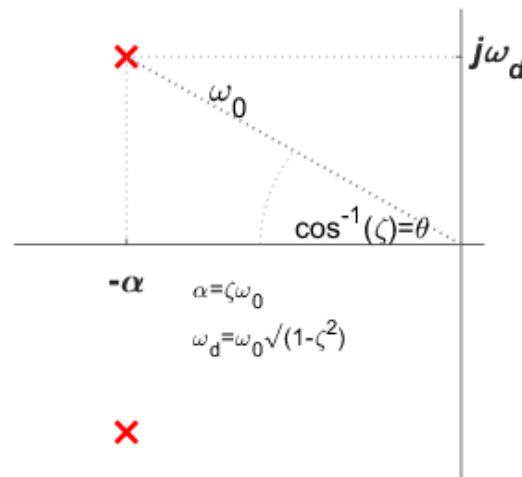
Performance Specifications

Enhanced model and the control system performance specifications and why you chose them. We are planning to create a slightly underdamped system. We want dampening values (zeta) of around .90 and .85. We have two values because it is a fifth degree equation, so we are separating it into two values for more stability. This means we have a high decay of angles and a frequency of oscillations relatively close to the natural frequency. The disturbance we are working against is any type of push towards the rocky, specifically pushes that change his angle

of tilt but do not turn him. This could theoretically be represented as a constant step input with boundaries in the time domain.

Poles of the System

For the poles of the system, we decided to use simple calculations based on the graphic below:



For our five poles, we chose one pole to be a constant value of $-w$ (where w is the natural frequency of the system), which is a critically dampening system. Then for the other 4 poles, we separate them into 2 pairs: for both of them, the points are: $w(-\cos(\theta) \pm \sin(\theta))$ where the angle is defined by the terms we said above, which lead to us using angles of 25° and 30° . This gives us the pole values of:

Constant: -9.3469

Angle of 25° : -4.5210 & -12.4213

Angle of 30° : -3.4212 & -12.7681

Controller Parameters

For determining the controller parameters (K_p , K_i , J_p , J_i , and C_i) we use a very simple script. This script uses MATLAB's `syms` function to create a simulation of the system using the values we already calculated. It also creates a target characteristic polynomial using the poles we gave the system, and using both of these it calculates the values of K_p , K_i , J_p , J_i , and C_i . Using the values we showed to calculate above, we get controller parameters of:

$K_p = 13046$, $K_i = 61176$, $J_p = 497.7807$, $J_i = -15643$, $C_i = -20073$.