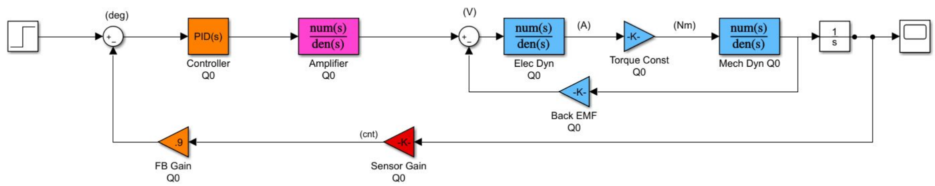
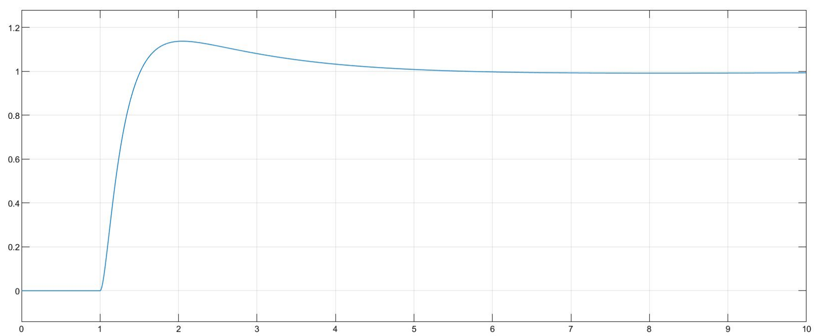
Entire System Together





Zero:

-29.4559

Poles:

$-1.2941*10^4$,

$-0.0031*10^4$,

$-0.0001*10^4$,

0

This is the overall schematic of our model. As seen, there is a PID(s) that gets multiplied by our linearized open loop response (the blue blocks), which outputs the angle (in radians) the 4 bar robotic arm will perform. To make the output in degrees a ‘radians to degrees’ block would be added after the integrator block. The feedback gain, is responsible for converting the analog signal from the sensor to an angular position measurement, as we are working with a linearized system, the FB gain will be 1/(sensor gain) to obtain unity gain.

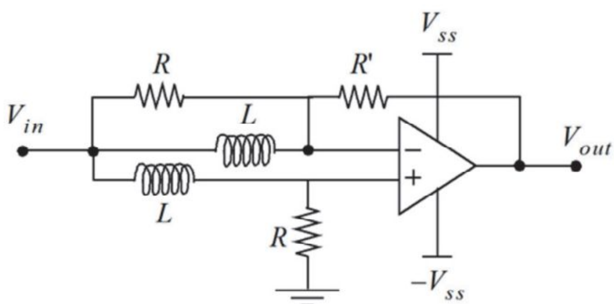
We also used the top block diagram to test and generate the below step response that utilizes our PID gains computed (page 7). Using Matlab’s `zero()` and `pole()` functions we found the characteristics of the open-loop transfer function (provided above).

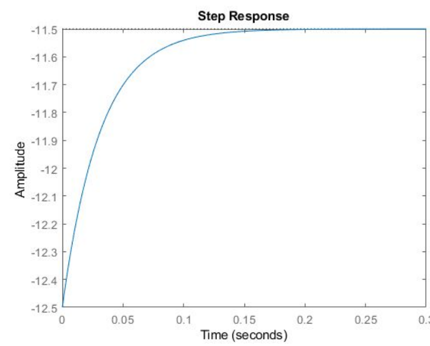
This report will detail the methods used to obtain the transfer functions of each component used in the system.

Power Amplifier Circuit

$$\frac{V_{out}}{V_{in}} = -\frac{L^2R's + LRR' - LR^2}{L^2Rs + LR^2}$$

$$\frac{V_{out}}{V_{in}} = -\frac{12.5s + 368}{s + 32}$$





2

To compute the transfer function of the Power Amplifier circuit, we did the following:

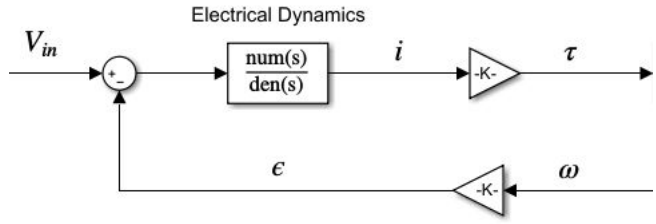
- Transform the circuit into the Laplace domain, replace the components with their equivalent impedances (L becomes Ls for instance)
- Write out the following KCL equations and use MNA
- Also need to recognize that the amplifier has negative feedback therefore we can use the notion that V+ = V-
- Also recognize that there is no current being delivered to the non-inverting and inverting inputs to the OpAmp

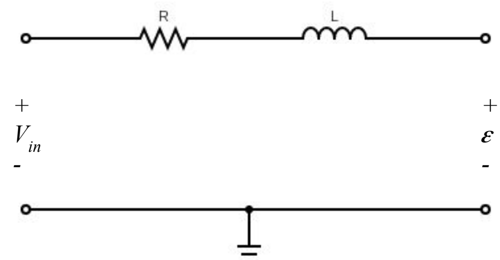
Once this was done, we were able to determine the transfer function equation and simply replace the values with the ones provided in the datasheet. At a high level the transfer function makes sense as since there is negative feedback on the inverting input, the output signal should be flipped (the presence of the minus sign). However, this means that the input to the motor will be negative, so for the entire transfer function of the system we will multiply by -1.

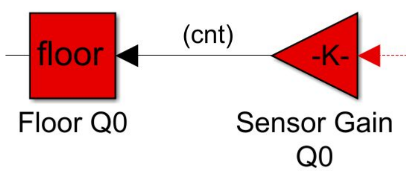
Electrical Motor Dynamics

$$\varepsilon = K_B \omega \quad \tau = K_\tau i$$

$$\frac{i}{V_{in}} = \frac{1}{8.73E(-5)s + 1.13}$$



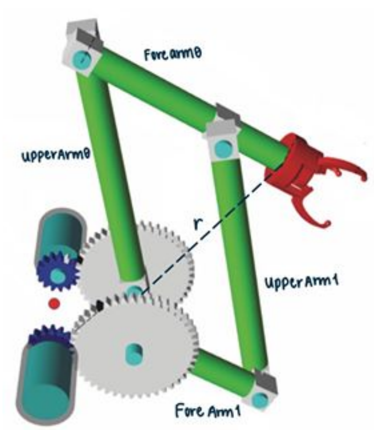


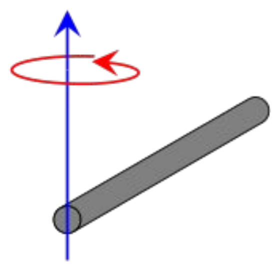


For the transfer function of the motor, being the Maxon ECX19M model, the electrical portion, i/Vin = 1/(Ls+R), was modeled using the simple circuit shown above using circuit analysis. The back emf and torque output was computed using the equations shown above.

The sensor gain was computed by taking the number of windows in an encoder (N = 100), by multiplying by 4 (4 readings), then dividing by 360 (degrees), we get the sensor gain of 1.111. We also linearized this system by setting the value of Floor = 1 which is equivalent to “removing” this block.

Mechanical Dynamics - Inertia





$$J = \frac{1}{3}mr^2$$

$$J = \frac{1}{3}mr^2 + m_{shift}r_{shift}^2$$

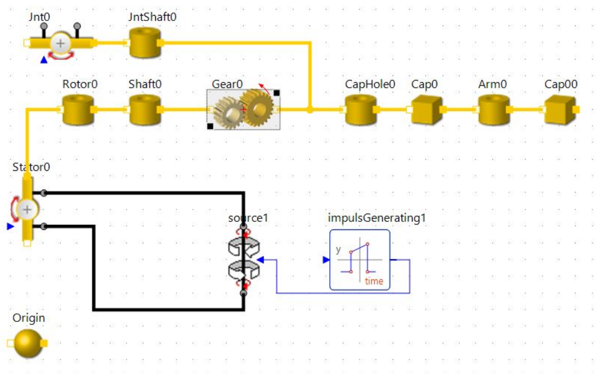
$$J = Mr^2$$

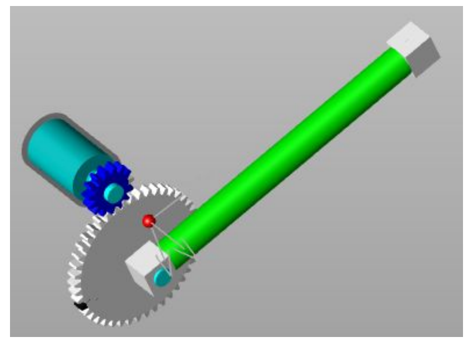
$$J_{total} = 1.2176 * 10^{-4}\frac{kg}{m^2}$$

The inertia of each component on the robot arm was found by setting the Bullgear as the Origin point and performing respective inertia calculations:

- UpperArm0 and ForeArm1 utilized the rod inertia equation
- UpperArm1 and ForeArm0 utilized the shifted rod inertia equation as it has a shift from the origin point
- The gripper utilized the point inertia equation with distance r from the origin
- Pinion inertia was calculated with the estimation of gear inertia provided in the document by Dr. Stocco

Mechanical Dynamics - Linearization

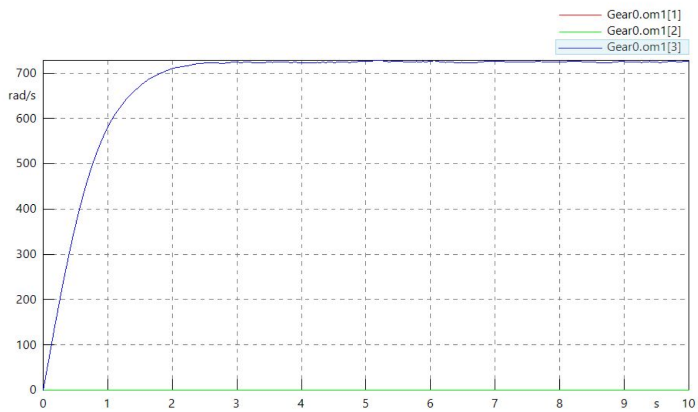


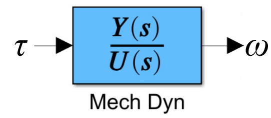


In order to obtain the B value, we had to first linearize the system. This meant blocks that were non-linear were “removed” from the simulation schematic as well as setting our flags in model.m to 0 to turn “off” different models of the system. To find the B in 1 gear, we deleted all of the system except for the gears, motor, UpperArm0 and other connecting elements that were needed. This simple system now experiences half of the total inertia we calculated (1.2176*10^(-4)/2), we then set our UpperArm0 mass and density to 0 so the end point of the rod (Cap00) experiences point inertia. We then calculated the equivalent mass of Cap00 in order to create the inertia needed for our revised system.

We added in an impulse generator block in order to see how our system behaves graphically as well as set our torque in “source 1” to the nominal torque of the motor in order to simplify the simulation. The linearization of the 4 Bar system will assist us in our B estimation between the gears.

Mechanical Dynamics - B Estimation





$$MechanicalTF:$$

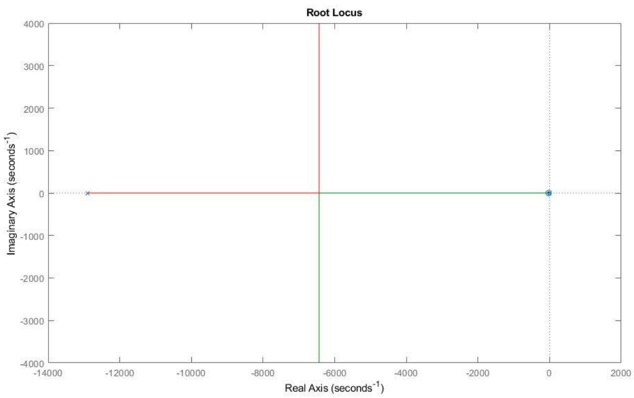
$$\frac{Y(s)}{U(s)} = \frac{\omega}{\tau} = \frac{1}{Js+B}$$

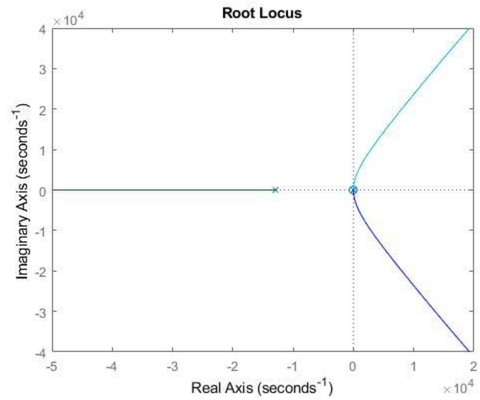
$$f(\infty) = \lim_{s\to 0} s\frac{\frac{1}{J}}{s+\frac{B}{J}} = \frac{\frac{1}{J}}{\frac{B}{J}} = \frac{1}{B}$$

$$\frac{\omega}{\tau} = \frac{1}{1.2176E(-3)s + 1.02E(-5)}$$

The angular velocity vs time graph models the performance of the pinion gear when simulated with the revised system shown in the previous slide. The steady state of the graph reached around 728.2 rad/s. The mechanical transfer function represents output/input which is angular frequency / torque constant (also known as nominal torque = 7.57E-3). Since our angular frequency (728.2 rad/s) is taken at steady state, we must perform final value theorem on the transfer function.
By doing the final value theorem of the mechanical transfer function we obtain (728.2 1/s)/(7.47E-3 Nm) = 1/B, therefore the estimation of B = 1.0258E-5 kgm^2/s.

PID Theoretical Design





Since we have a modelled system, we do not need to use the guess-and-check methods, such as the Ziegler-Nichols method, to compute the initial PID gains. Instead we can use the Root Locus and the open loop transfer function to find the gains systematically.

- First we need to find the open loop transfer function of the entire system (as seen above)
- Then we can plot the Root Locus of the transfer function
- Next we picked two of the zeros included in the PID function to cancel out the two most unstable poles in the open loop transfer function (these being s = -31, and s = -1)
- From there we can equate them to the characteristic equation of the PID transfer function
  - From this we get: Kp = Kd and Ki = 31/Kd
- Now we just need to find Kd - to do this we plotted the Root Locus of the KGH transfer function and picked an s value (being -6440 as it is most stable from the left Root Locus plot), and solved for Kd using the magnitude criteria |KGH| = 1

Along with this method, we also found that in Simulink, there is a PID controller tuning block. We used this to find optimized values for the P, I and D gains by setting requirements such as the rise time, damping, and having no steady state error. In the end we came up with the following gains:

Kp = 0.0672
Ki = 0.0120
Kd = 0.0918

Part I (ELEC 341 Project) Done by:

Mattias Zurković 75106880

Karmen Wang 54144183