B10 – “The Rocket Men”

CONTROL GROUP

RCGs

Requirement

- 50 ms Instruction time / Rise Time less than 50 ms
- Less than 5% overshoot
  - **Satisfied**
- 2-DOF Spherical Wrist
- Simulink Model
  - **Satisfied**
- Real-Time Controller
  - **Satisfied**

Constraints

- 16 MHz CPU clock speed
- Budget ($1000 UBCD + $400 CAD)

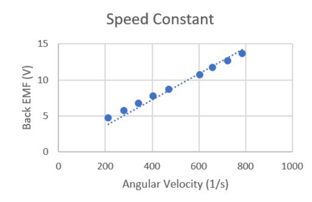
Goals

- Everything on PCBs
- Separate Laser Power – No Dangling Wires from Diode
- Animation
- Optimal Rise – 25 ms

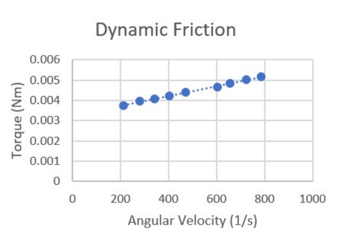
Parameter Identification
Speed Constant and Dynamic Friction Coefficient (B)

$\epsilon = k\omega$ $\qquad\qquad$ $\tau = B\omega$

SPEED CONSTANT (Kw)
0.015547 Vs



Dynamic Friction (B)
2.42E-06 Nms

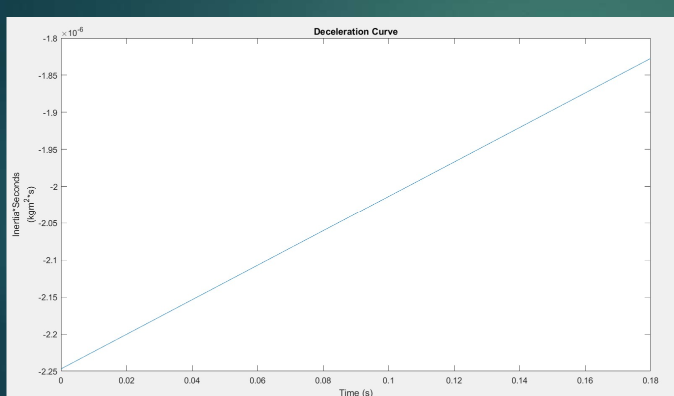


- Steady State Unloaded Voltage Tests
  - Current Draw (I) and Angular Velocity ($\boldsymbol{\omega}$) Recorded
- Back EMF (ϵ) calculated from current & terminal resistance
  - Terminal Resistance & Inductance measured with RLC meter
- Speed Constant (K) as slope of Velocity/Back EMF curve
- Friction Coefficient (B) as Slope of Torque / Speed Curve
  - Torque calculated with: $\tau = kI$

Parameter Identification

Rotor Inertia (J) - Deceleration test of unloaded motor

$$\mathrm{J} = 8.30\ gcm^2$$



- Angular System with no external Torque:

$$0 = J\dot{\omega} + B\omega$$

then:

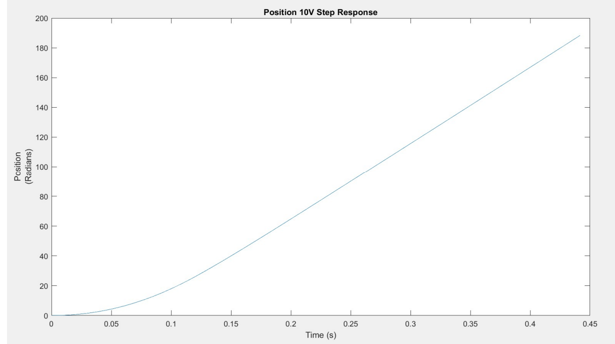
$$J\ is\ Slope\ of\ -\frac{B*\omega(t)}{\dot{\omega}(t)}\ curve$$

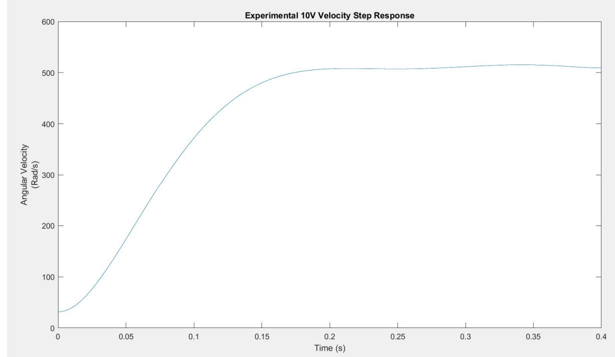
- Power is cut to motor running constant velocity and velocity and acceleration is recorded as motor slows

Experimental Step Response Curve Fitting

- Step test is conducted by applying a 10V step voltage to motor and recording position of motor
- The Velocity Step Response is obtained by differentiating the Position Step Response
- Utilizing MATLAB's System Identification toolkit a transfer function can then be fitted to the Velocity Response
- Transfer Function from Velocity Step Response

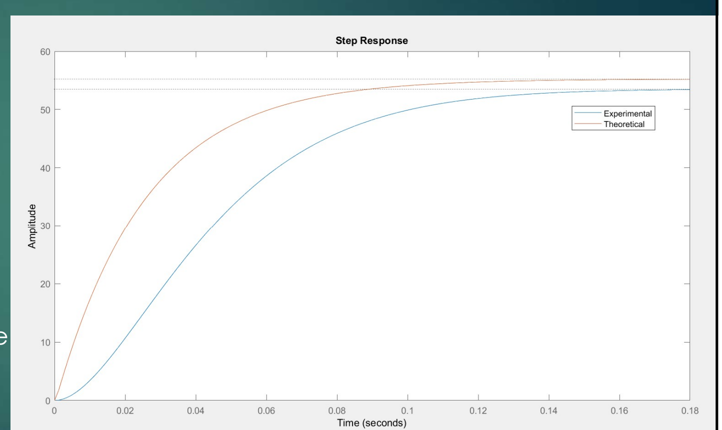
$$\frac{\omega(s)}{V(s)} \frac{2.702x10^4}{s^2+38.02s+531}$$





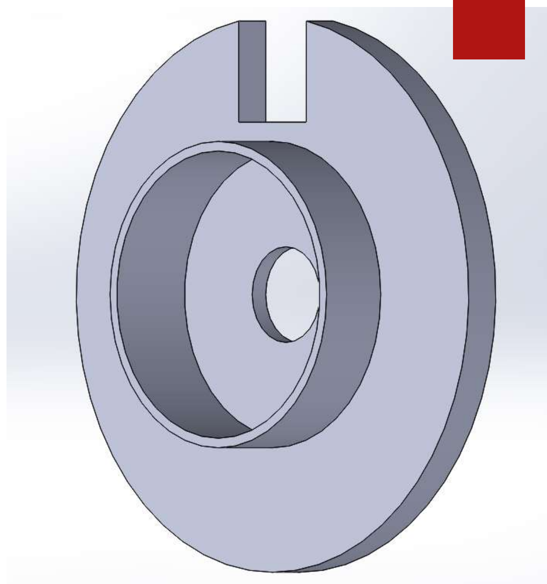
Model Verification – Step Response Comparison

- Comparison between Experimental and Modelled Step Responses helps verify model accuracy
- Possible Causes of Difference between Responses:
  - Static Friction is unmodeled
    - Accounts for initial delay of experimental response
  - Current Driver is unmodeled
    - Resistance/Capacitance/Inductance that could effect response

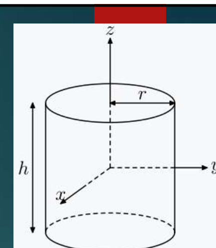


Encoder & Slot Detector Attachment

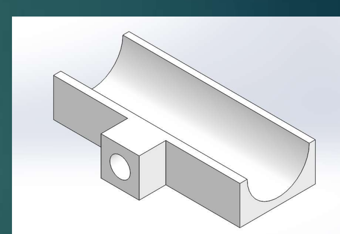
- Fits directly onto face of Jameco Motor.
- Slot Detector friction fitted into slot at tope of attachment
- Advantages:
  - Usable with motor oriented vertically and horizontally
  - Reusable
  - Adjustable



Mechanical System Model



- **Encoder Disk**
  - Modelled as Cylinder rotating about it's Z-Axis
  - Mass (m) calculated from Encoder Dimensions & density of FR4 ($1.850\ \frac{g}{cm^3}$)
    - $J_{Encoder} = \frac{1}{2} * mr^2 = \frac{1}{2} * (5.2719)(2.4)^2 = 15.18\ gcm^2$
- **Laser Diode Bracket**
  - Modelled using SolidWorks Mass Analysis & density of PLA ($1.25\ \frac{g}{cm^3}$)
    - $J_{Bracket} = 0.177 gcm^2$



- **Laser Diode**
  - Modelled as cylinder rotating its y – axis
  - Mass Measured at 2.00 grams
    - $J_{Diode} = \frac{1}{12} m(3r^2 + h^2) = \frac{1}{12} * (2.00)(3(0.7)^2 + (2.1)^2) = 0.98\ gcm^2$

Simulink – Block Diagram + Kinematics

- Allows for real time simulation of system
- Builds transfer function blocks from measured parameters
- Parameters in separate MATLAB files to allow for incremental and documented change
- Single Degree of Freedom for Laser Diode with Constant Y_d

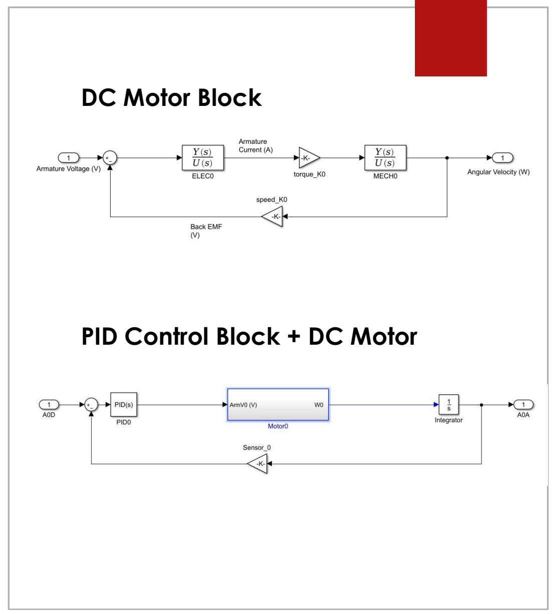
$(X_d, Y_d) \Rightarrow \theta_D$

Inverse

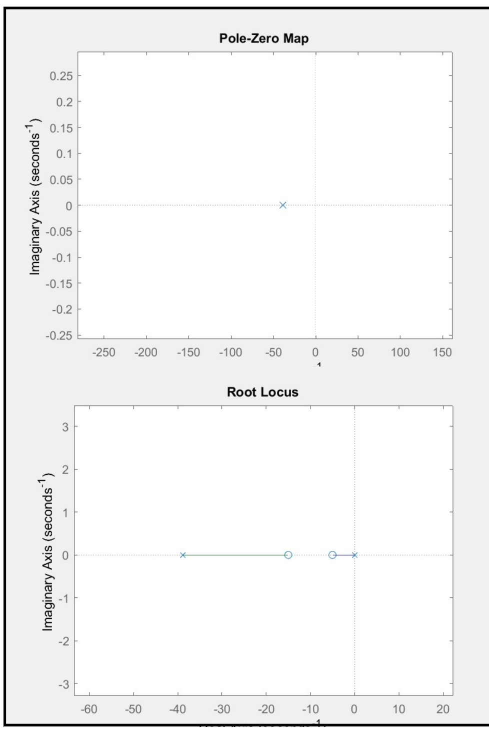
$\theta_D = atan\frac{X_D}{Y_D}$

Direct

$X_A = \tan(\theta_A)^* \, Y_A$



PID – PZ Plot & Zero Placement & Root Locus



PZ Plot

- **Additional Pole at S = -3000**
  - **Assuming contribution is negligible**

Starting Zero Placement

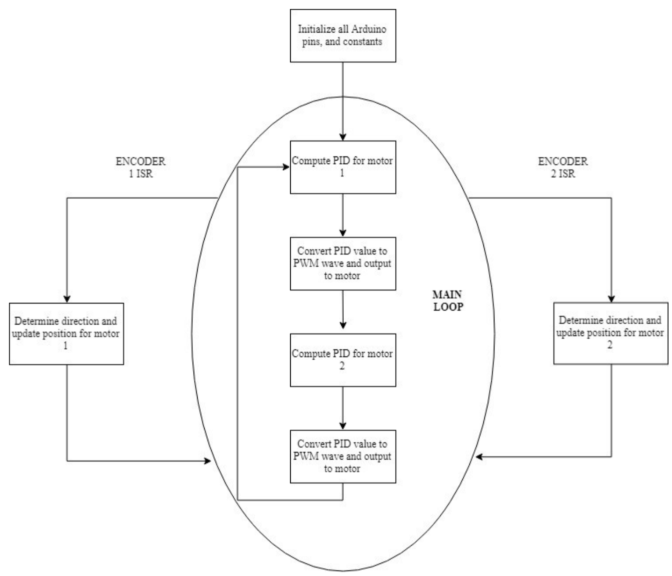
- (s + 5)(s+15)
- Draw Pole at Zero to left for me stability
- Draw Pole on the left to right for improved response time
- No K_U

Starting Gains

$K_D = 0.0400$

$K_P = 0.30$

$K_I = 0.80$



Program Block Diagram

PID Pseudocode

1. Record current program time
2. Calculate dt = time – last time
3. Calculate error = desired position – current position
4. Calculate de
    1. IF Setpoint has changed since the last reading, de = -current position – last error
    2. ELSE de = error – last error
5. Calculate total error = total error + error
6. Calculate output = kp*error + ki*total error + kd*de/dt
7. Constrain output between ±255 for Arduino
8. Store time, total error, and desired position for next calculation

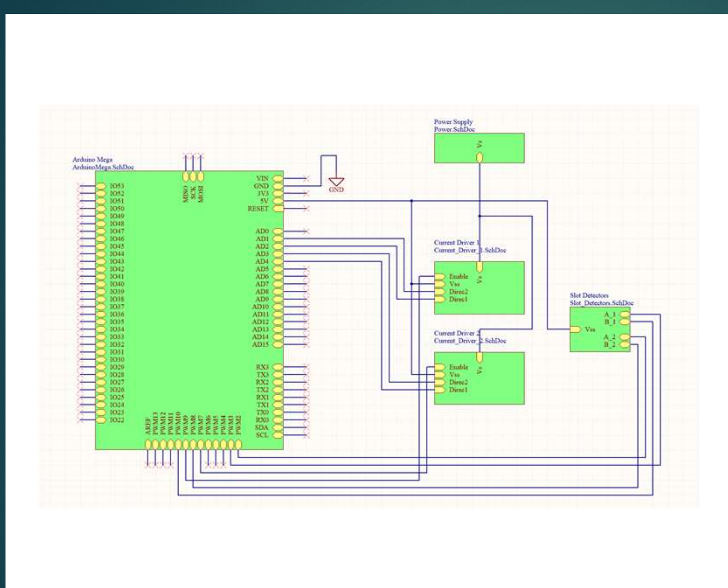
Microcontroller Selection

Arduino Uno

- Interrupt pins: 2
- PWM pins: 6
- RAM: 2kB
- Readily available and easy to build PCB around

Arduino Mega

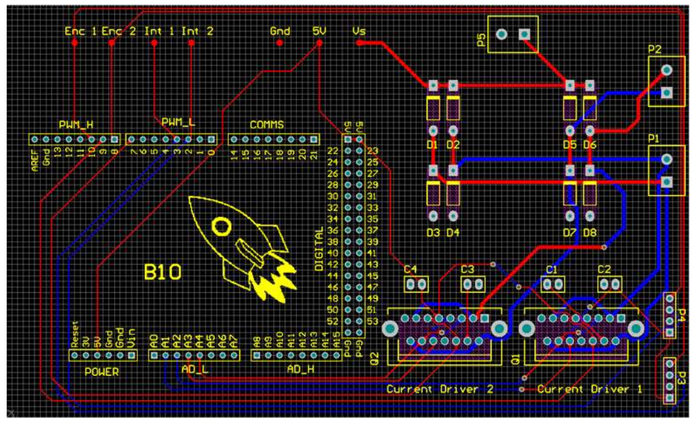
- Interrupt pins:
- PWM pins: 15
- RAM: 8kB
- Need to order in, challenging to build PCB around

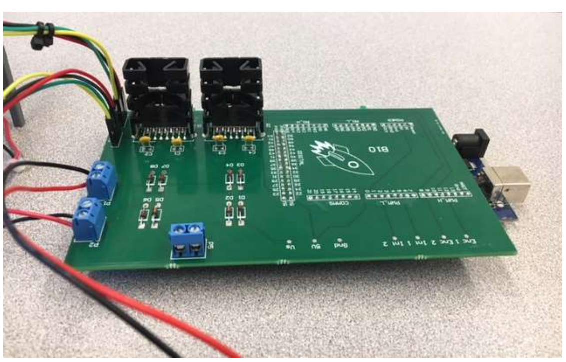


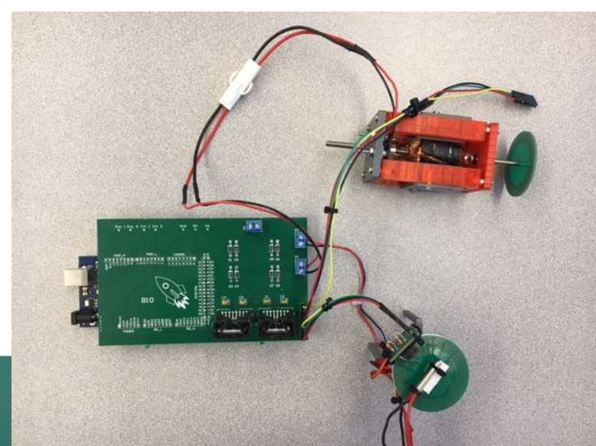
PCB Top Level Design

PCB Layout

- All ground connections connected to polygon pour







PCBA

Encoder ISR Pseudocode

1. ISR triggers on falling edge of Slot Detector A
2. Check state of Slot Detector B
   1. If Slot Detector B is high, direction is forwards
      1. Increment displacement (number of interrupts from initial position)
   2. If Slot Detector B is low, direction is backwards
      1. Decrement displacement
3. Convert displacement to radians:
   $$Displacement_{rad} = Displacement * \frac{3.6}{180}$$

PCB Future Improvements

- Reduce noise from custom motor
  - Add mechanical cutout between motors
  - Replace ground plane with multiple small ground planes that don't come near motors
  - Add capacitor across motor terminals
  - Add small inductors in series with motors to prevent large currents in one motor from affecting the other
- Reduce size of PCB
  - Remove Arduino daughter board
- Add additional power ports
  - Banana plugs, barrel connector

Summary

- Currently Have Operational Real-Time Controlled 1-DOF System
  - Satisfactory Overshoot
- Fully operational Parameter Acquisition Procedure
- Accurate Model of System
- Alpha Version of PCB designed and manufactured