California State Polytechnic University of Pomona

College of Electrical and Computer Engineering

Control Systems Laboratory

ECE 3709L.02

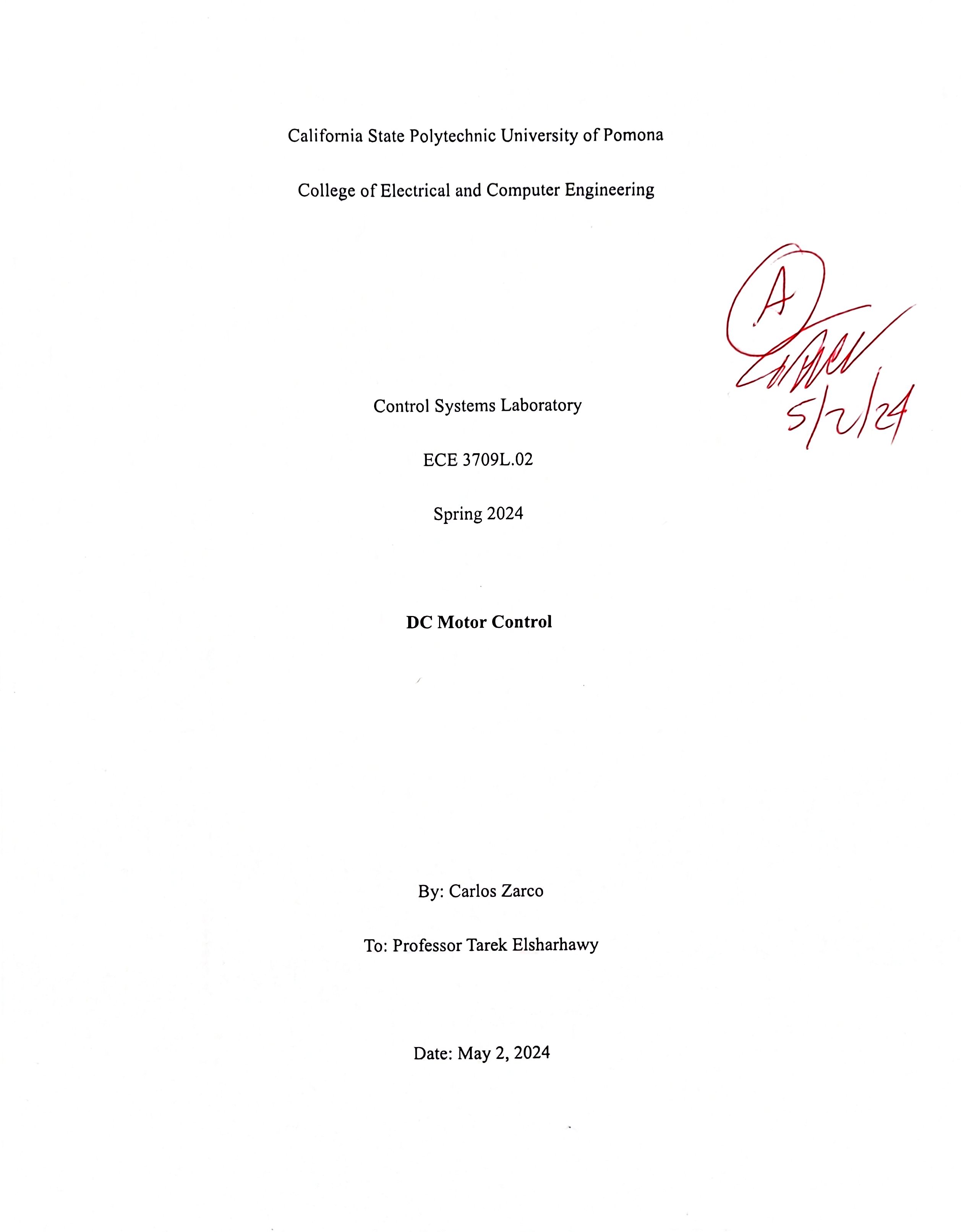
Spring 2024

**DC Motor Control**

By: Carlos Zarco

To: Professor Tarek Elsharhawy

Date: May 2, 2024



**Table of Contents**

1. **Objective……………………………………………………………………………………...1**
2. **Procedure………………………………………………………………………………..…...1**
3. **Conclusion…………………………………………………………………………………....1**

**Appendix A. Hardware…………………………………………………………………………..2**

**Appendix A.1 Control System Schematic…………………………………………………..2**

**Appendix A.2 Determining kp and k­i in from LabVIEW………………………………….3**

**Appendix B. MATLAB…………………………………………………………………………..5**

**Appendix C. Control System Designer App……………………………………………………7**

1. **Objective:**

The goal of the lab was to design, assemble, and test a DC motor control system for speed and rotation. MATLAB was also utilized to simulate and confirm the results of the physical hardware.

1. **Procedure:**
2. Studied open and closed loop control system block diagrams, identifying key parameters and the impact of disturbances.
3. Received and corrected a schematic for a feedback control system.
4. Compared the simulation results with the performance of the physical motor hardware.
5. Performed hardware tuning using a myDAQ controller, motor driver, and LabVIEW software to determine kp and ki values from the step response graph.
6. Implemented the control system in a MATLAB script, plotting step responses and bode plot for open and closed loop transfer functions.
7. Implemented the control system in a MATLAB script, plotting step responses and bode plot for open and closed loop transfer functions.
8. Lastly, we compared the simulation results with the performance of the physical motor hardware.
9. **Conclusion:**

The results from our tests with physical hardware gave us the following values of k­p =0.06 ­and ki = 0.6 for a critically damped graph. The settling time of LabVIEW was approximately 4.1 seconds. In the Control System Designer app, we managed to reduce the settling time to about 3.9 seconds after adjusting the simulation kp and ki values leading to an approximate 5% percent difference between the simulation and hardware results.

1. **Hardware:**
   1. **Control System Schematic:**

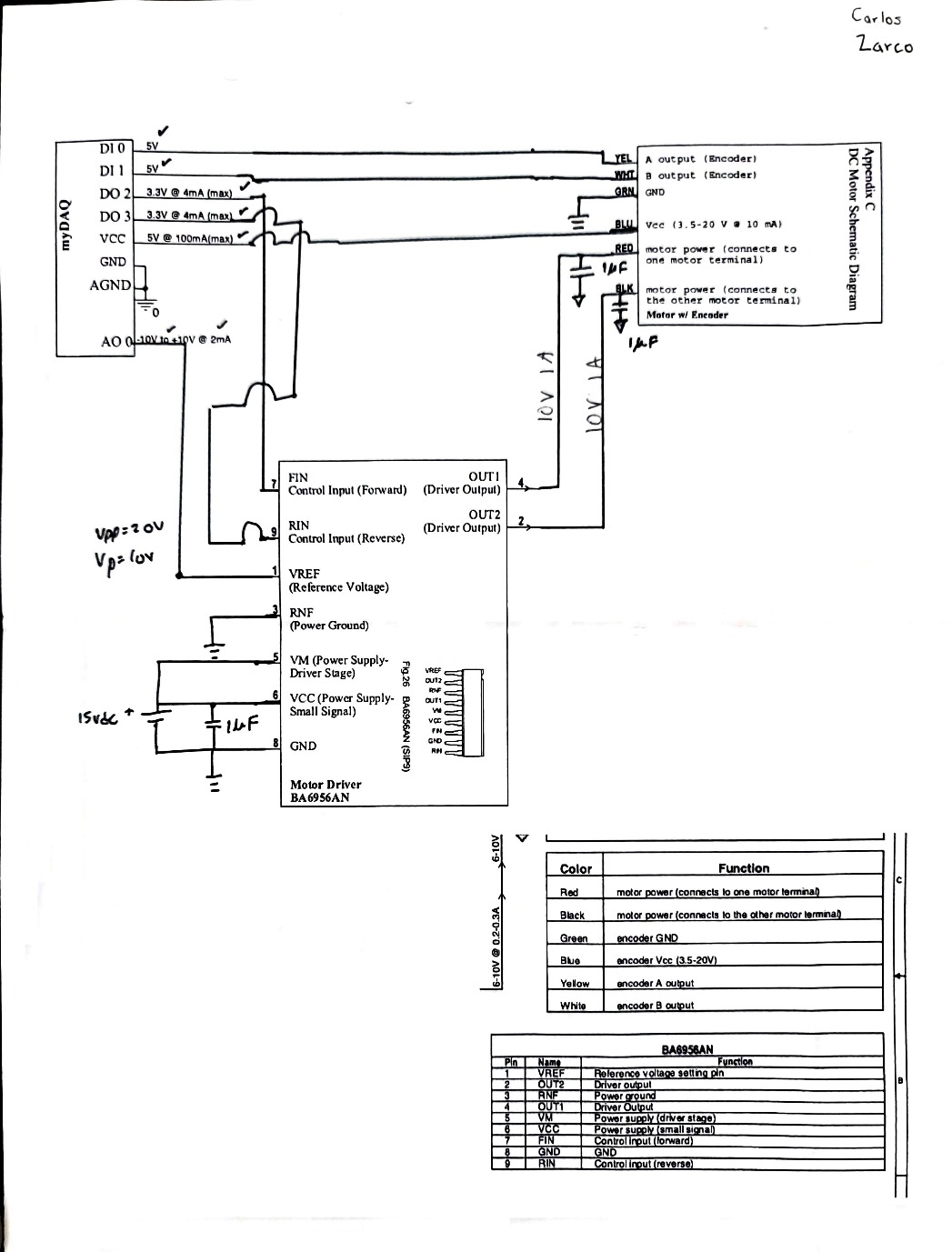
We received the following schematic at the start of this semester which contained errors which needed to be corrected.

A computer screen shot of a diagram

Description automatically generated

*Figure 1: Original given DC Motor Control System Schematic*

Pins 1 and 3 are connected via a voltage divider with R7 and R8. This was removed and the connection between A0 on the myDAQ and pin 1 of the driver was connected directly without resistors. Pin 3 of the driver was connected directly to ground. The resistor R9 on pin 5 of the driver was also removed. The additional power source VCC for the driver was changed from the listed 6.5-15V to a constant 15 Volts because the driver datasheet states that VREF < VM, and VC. Lastly two capacitors were added to the red and black pins of the motor with encoder. The new diagram is shown below in figure 2.



*Figure 2: Updated Control System Schematic*

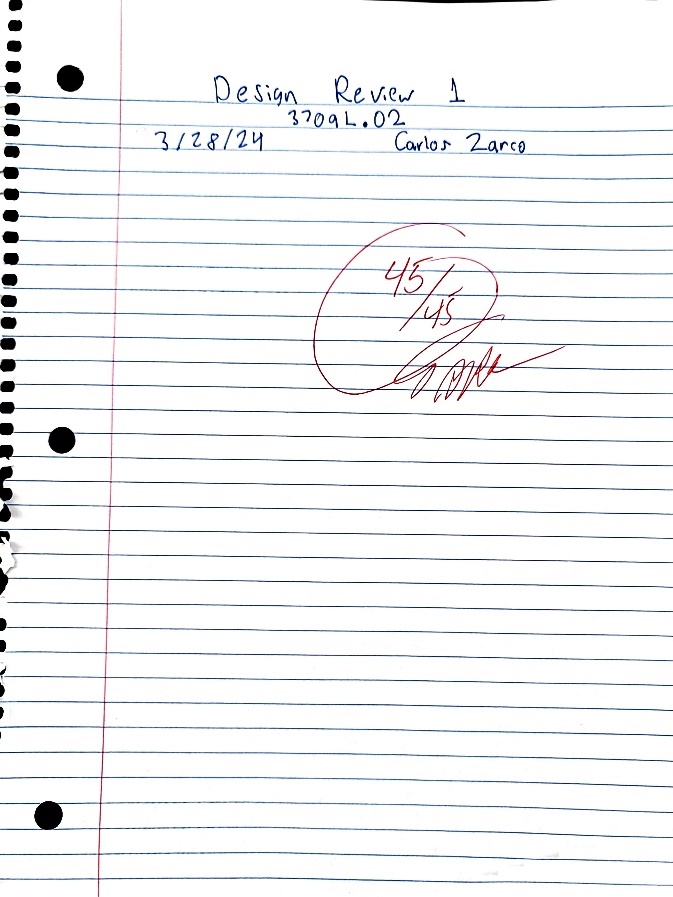
* 1. **Determining kp and k­i in from LabVIEW.**

To evaluate the kp and k­i we started off with a constant value of k­i then incrementally incremented kp until we saw the curve we wanted. Then we proceeded to adjust the ki value until the desired results were achieved as shown in Table 1 below.

|  |  |  |
| --- | --- | --- |
| kp | k­i | Response of the system: |
| 0.02 | 0.01 |  |
| 0.045 | 0.01 |  |
| 0.051 | 0.01 |  |
| 0.055 | 0.55 |  |
| 0.06 | 0.6 |  |

*Table 1: Data for kp and ki values from LabVIEW*

We got 0.06 and 0.6 for kp and k­i respectively. The final graph shows that we have achieved a critically damped response curve that has a settling time of approximately 4.1 seconds. The graphs and values for our integral and proportional gain were approved in Design Review 1 as shown in image 1 below.



*Image 1: Approval for Design Review 1*

1. **MATLAB**

For the MATLAB simulation we used the following equation as the transfer function for the motor, which was provided to us by our professor from a University of Michigan webpage. This derived in part by the 2nd order differential equation of RLC circuits which can be used to model any real life scenario.

A math equation with a number of letters

Description automatically generated with medium confidence

*Equation 1*

With this information we were able to code our step response and bode plots using the information from the site for the J, L, and R values. My partner and I did try to measure our R and L values in the stockroom, but our numbers were off, so we went with the values on the site to continue plotting. The graphs are listed and labeled below for our MATLAB simulation.

A graph of a function

Description automatically generated with medium confidence

*Figure 3: MATLAB Simulation for open, closed loop, and bode plot of each with original kp=0.06and k­i =0.6*

We found that the settling time of this system was off by a significant amount as it reached 63.8 seconds. This meant that our kpand k­i values needed to be adjusted. We kept proportional gain the same kp = 0.06 same and heavily increased integral gain to k­i = 6. After this change was made the simulation results more closely mirrored the results that we saw with the physical hardware using LabVIEW.

A group of graphs showing different steps

Description automatically generated with medium confidence

*Figure 4: MATLAB simulation using adjusted kpand k­i values. (kp= 0.06 and k­i =6)*

This more closely mirrors our measured settling time with LabVIEW of 4.1 seconds. This leads to a percent change of approximately 1.72%.

1. **Control System Designer**

After modeling the control system in MATLAB, we modeled our block diagram in Simulink and had to simulate the results in the Control System Designer application. In the application under the ‘edit architecture’ option, the transfer function of the motor can be directly imported into the program. This uses the modified values for the integral and proportional gain that we found during our MATLAB simulation. The control system designer app automatically graphs the bode plot, root locus, and closed loop step response. This powerful tool also allows for the poles on the root locus to be changed manually on the graph and it will update the bode plot and step response accordingly based on the selected pole locations. Constraints can also be set within the application to further refine the transient time response characteristics of the design. The block diagram model is also available to be exported here and can be viewed in Simulink. A screenshot of a computer

Description automatically generated

*Figure 5: Control System Design Simulation featuring Root Locus, Bode Plot, and Step Response graphs.*

The settling time here was 3.93 seconds. The rise time was 2.5 seconds, and the transient time was 4.03 seconds. Verifying the results of the hardware and MATLAB simulation, as they are similar.

1. **Performance Comparison**

The results that are seen in the MATLAB, Control System Design application, and physical hardware demonstrate that the results have been accurate with a percent change of 5%. The settling time of the control system design application was 3.93 and the result from the hardware was 4.1 seconds. These values were very close. However, the value of 3.93 was achieved because we needed to modify the integral and proportional gain values in the simulation to have comparable performance. Without modification, the MATLAB simulation settling time was 63.8 second which is significantly higher than what was measured in the response curve on LabVIEW. The percentage difference using the original kpand k­i values was 175.85%. This discrepancy is large and could be caused by not using the measure R and L values in the DC motor transfer function for the simulation itself. The values measured in the stockroom were not accurate, so the used values were the given R and L values from the site. The significant difference could also be caused by using the wrong form of analysis. The plots that have been simulated are in the discrete domain, but the z-transform domain would be more accurate in depicting the physical hardware configuration that was simulated. However, by modifying the value for integral gain, the simulations were accurate enough to be able to verify the results of the hardware. The rise time from the control system design app was 2.5 seconds and the transient time was 4.03 seconds. Very closely mirroring the LabVIEW results.

**Works Cited**

[1] “Reversible Motor Drivers for Brush Motors 1.0A Reversible  Motor Drivers (Single Motor)  BA6956AN,BA6287F,BA6285FS,BA6285AFP-Y,BA6920FP-Y.” ROHM Co, Ltd, 2014. [Online]. Available: <https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/325/BA6956AN%2CBA6287F%2CBA6285FS%2CBA6285AFP-Y%2CBA6920FP-Y.pdf>

[2] “myDAQ Specifications.” NATIONAL INSTRUMENTS CORP., May 02, 2024. [Online]. Available: <https://www.ni.com/docs/en-US/bundle/mydaq-specs/page/specs.html>

[3] N. S. Nise, *Control Systems Engineering Eighth Edition Abridged Print Companion with Wiley E-Text Reg Card Set*. 2019.

[4] “DC Motor Speed: System Modeling,” *University of Michigan*. <https://ctms.engin.umich.edu/CTMS/index.php?example=MotorSpeed§ion=SystemModeling>

[5] “DC Motor Speed: PID Controller Design,” *University of Michigan*. <https://ctms.engin.umich.edu/CTMS/index.php?example=MotorSpeed§ion=ControlPID>