MEC 411 Lab

LAB # 1 : Digital PID Speed Control

Of a Turntable

Group #11

Matthew Drzymala

Deeba Farhin

Qian Ting Huang

Date: September 28, 2019

Table of Contents:

Abstract

The purpose of this lab is to implement a PID turntable speed control system that satisfies certain performance specifications. The PID controller is implemented using software with a Data Acquisition interface and the turntable is controlled by a DC motor with a tachometer. The lab tests three different parameters, which results in the three different types of controller; Proportional (P), Proportional-Integral (PI) and Proportional-Integral-Derivative (PID). Experimental results obtained from the lab will be compared to theoretical results calculated using MATLAB.

Introduction

A PID controller is set up in this experiment to control turntable speed. Experimental results are recorded and compared against theoretical values. The proportional factor gives an output that this the product of the gain and measured error. It corrects instantaneous error. The integral factor store all measured error which can be both positive and negative.It corrects the accumulation of error. The derivative factor corrects the current error as compared to the last recorded value of error.

PID controllers control any variable that can be measured and manipulated. The are important in industry as they allow to regulate various processes as part of a control loop. PID controllers are used in industry to control needs such as temperature, flow, pressure and level. A daily example on the use of PID controllers would be in commercial baking ovens. The controller in this case is a part of the temperature control.

The block diagram for the digital PID turntable speed control system is as shown below:

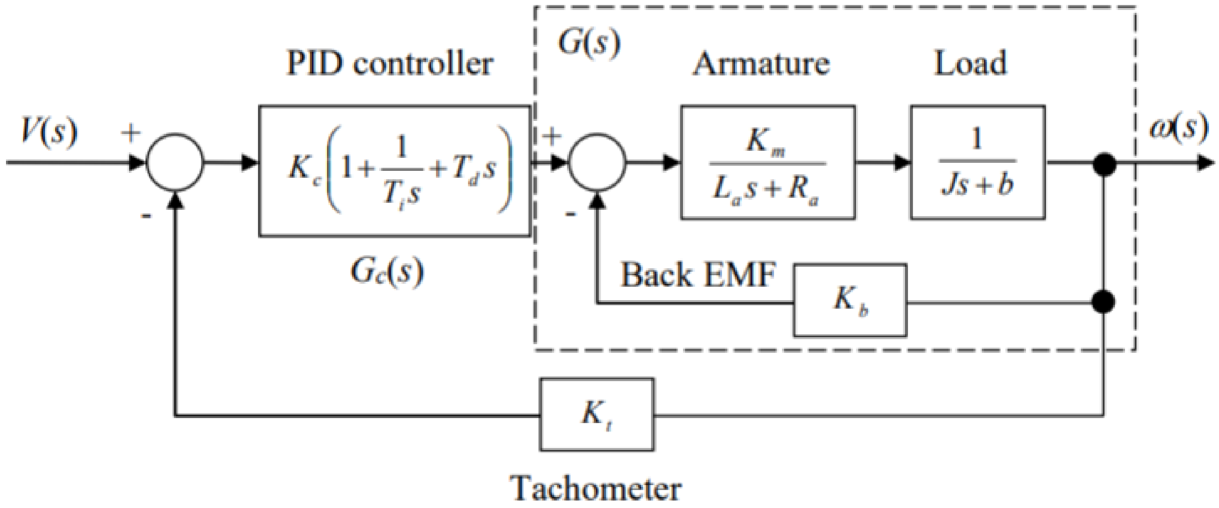

Figure 1: Block Diagram of a PID turntable speed control system

Transfer functions is defined as the voltage input over the output, which is the rotational speed of the tachometer. The closed loop transfer function is given by:

$$T(s)=\frac{\omega(s)}{V(s)}=\frac{G_c(s)G(s)}{1+K_tG_c(s)G(s)} \tag{1}$$

Where

$$G(s)=\frac{K_m}{(L_a s+R_a)(Js+b)+K_bK_m}\approx\frac{K_m}{R_a(Js+b)+K_bK_m} \tag{2}$$

$$G_c(s)=K_c\left(1+\frac{1}{T_i s}+T_d s\right) \tag{3}$$

Experimental Procedure

Equipment:

- DC Motor
- Tachometer
- Breadboard
- LM324D
- TIP 122 (NPN Epitaxial Darlington Transistor)
- TIP 127 (PNP Epitaxial Darlington Transistor)
- Wires
- Screwdriver
- Resistors
- Capacitors
- DAQ
- Computer
- Function Generator
- Oscilloscope
- Diodes

Circuit Diagram:

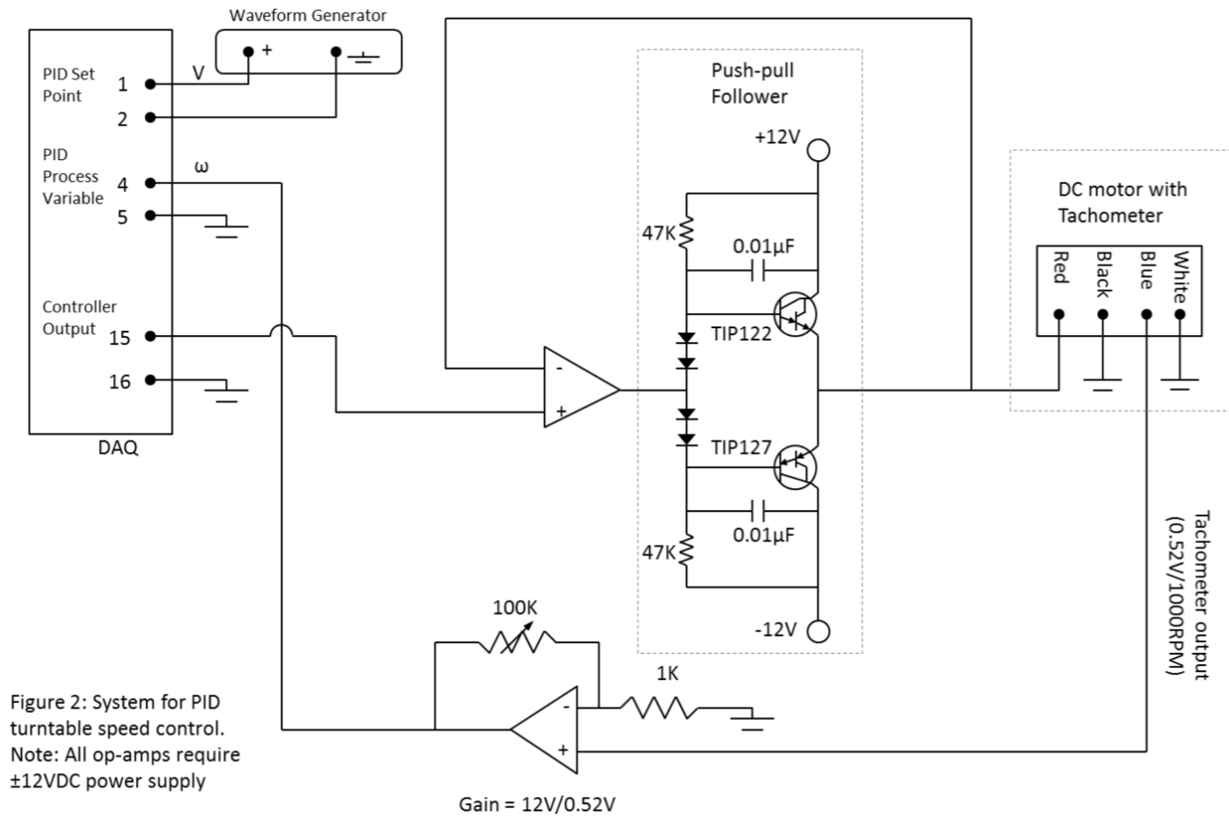

Figure 2: System for PID turntable speed control.

Procedure:

1) Initial step is to build the circuit of the system for PID turntable speed control using supplied circuit components on a breadboard as shown above in Figure 2. Adjust the 100K variable resistor to calculated value, which in this case is approximately 23K because the desired gain of this amplifier is 12V/0.52V.

2) Connect the circuit to the Data Acquisition (DAQ) interface with their respective channels as shown in the diagram. Channel 1 and 2 of the DAQ is connected to the waveform generator.

3) Connect the black and white wires of the tachometer and connect the blue and red wires to the circuit.

4) Connect the input V to the function generator. Set the function generator to a square wave with a frequency of 1 Hz and an amplitude of 2 volts.

5) Connect the oscilloscope to the circuit as shown in the picture. Channel 2 of the oscilloscope is connected to point ω. This point is the feedback voltage indicating the actual speed of the turntable measured by the tachometer.

6) Set system to be a feedback control with a P (Proportional) controller by setting the integration time (T_i) and the derivative time (T_d) to zero. Keeping the system still stable, adjust K_c until the turntable output rotational speed is as close to the input values as possible. Record all system parameters, along with input and output waveforms using the computer software.

7) Set the derivative time to zero and the system becomes a PI (Proportional-Integral) controller. Adjust K_c and T_i so that the turntable output speed is as close to the input as possible while still keeping the system stable. Record all data.

8) Adjust all three parameters to make the system become a PID (Proportional-Integral-Derivative) controller. Adjust the parameters by trial and error until the system performs as required by the given specifications. Record all values of K_c, T_d, and T_i using the computer software. [3]

Results

The input voltage supply was 12.03 volts and the frequency and amplitude supply from the waveform generator was 1 Hz and 2 volts respectively. The variable resistor was set to a value of 22.9Kohms. The parameters were tuned by using trial and error. Both the integral and derivative time were set to zero. It was when K_c is equal to 2.875 that the waveforms started becoming unstable. Ti and Td values remained at zero.

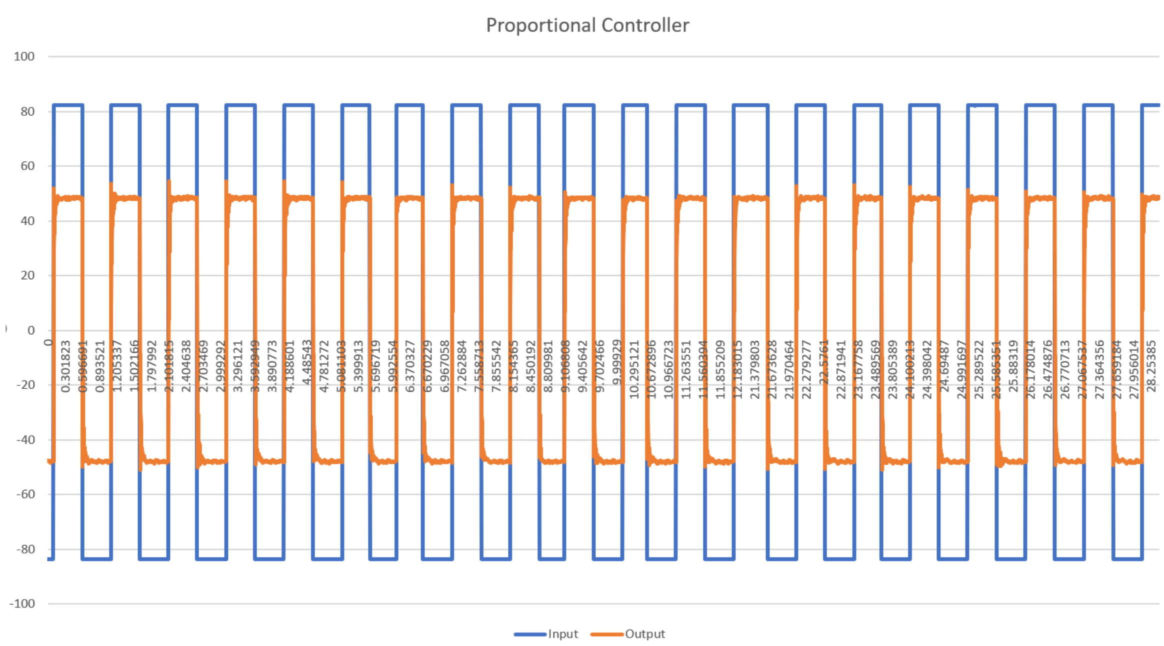

Figure 3: Rotational speed of Turntable from P controller

A simulated output response was made using Matlab PID Tuner. Theoretical response and the corresponding parameters for the controller can be seen in the next figure.

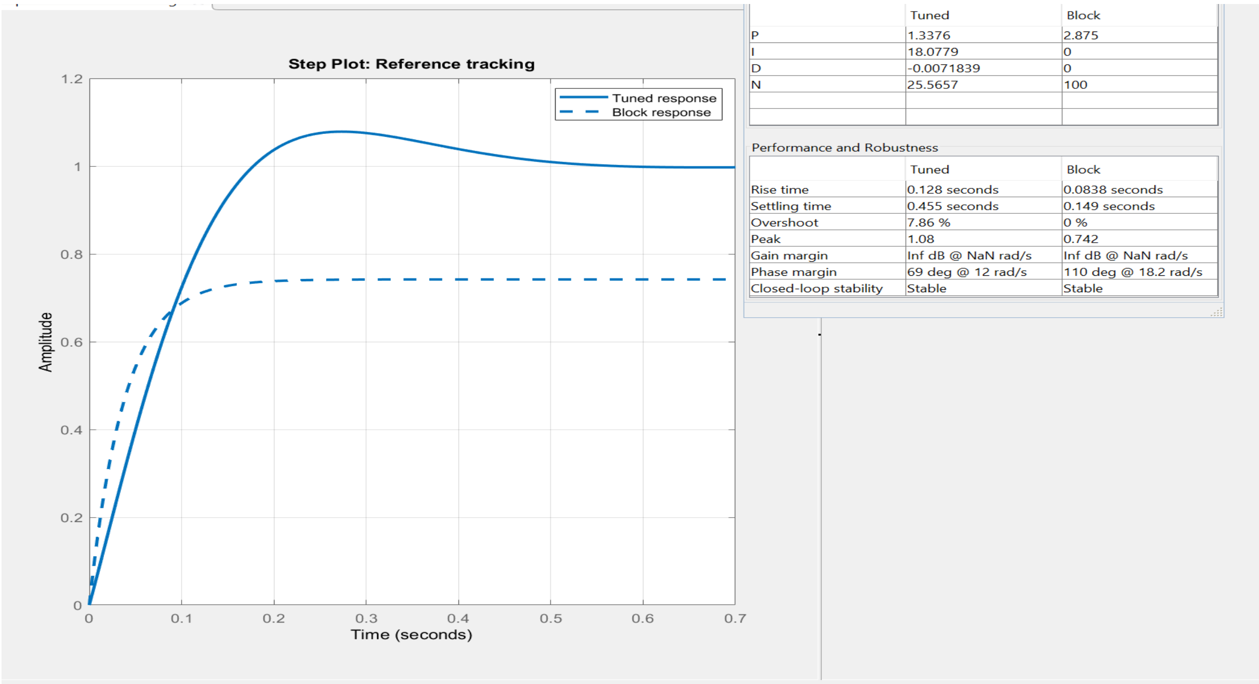

Figure 4: Simulated Response of P Controller

For the next part of the lab the integral component was now adjusted while keeping the proportional component constant and the derivative component remained at zero. These settings resulted in the waveform shown in the next figure.

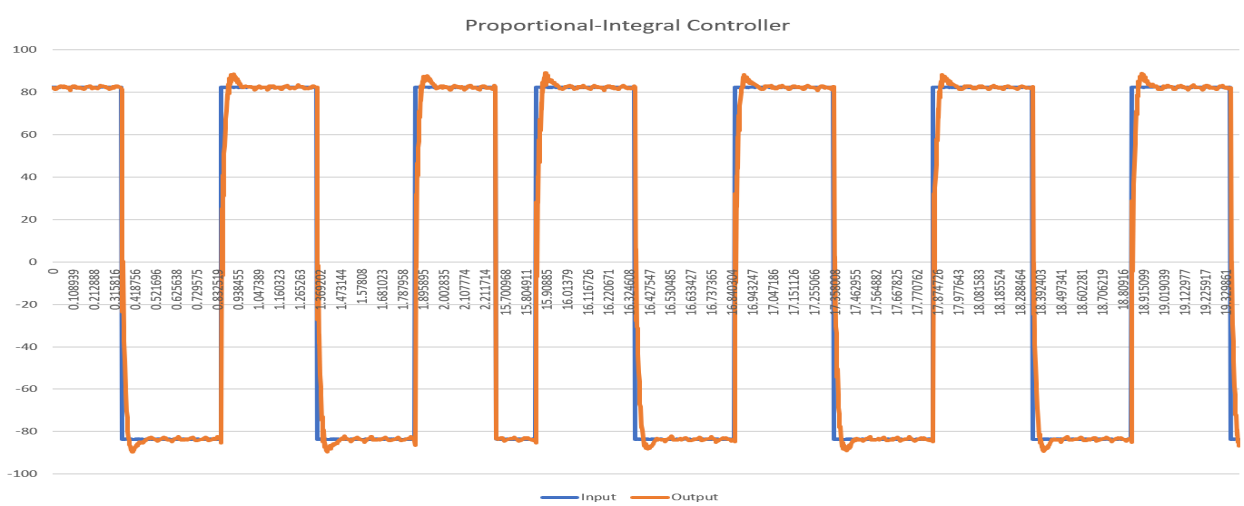

Figure 5: Rotational Speed of Turntable from PI Controller

Another simulated output response was made using Matlab PID Tuner. Theoretical response and the corresponding parameters can be seen in the following figure.

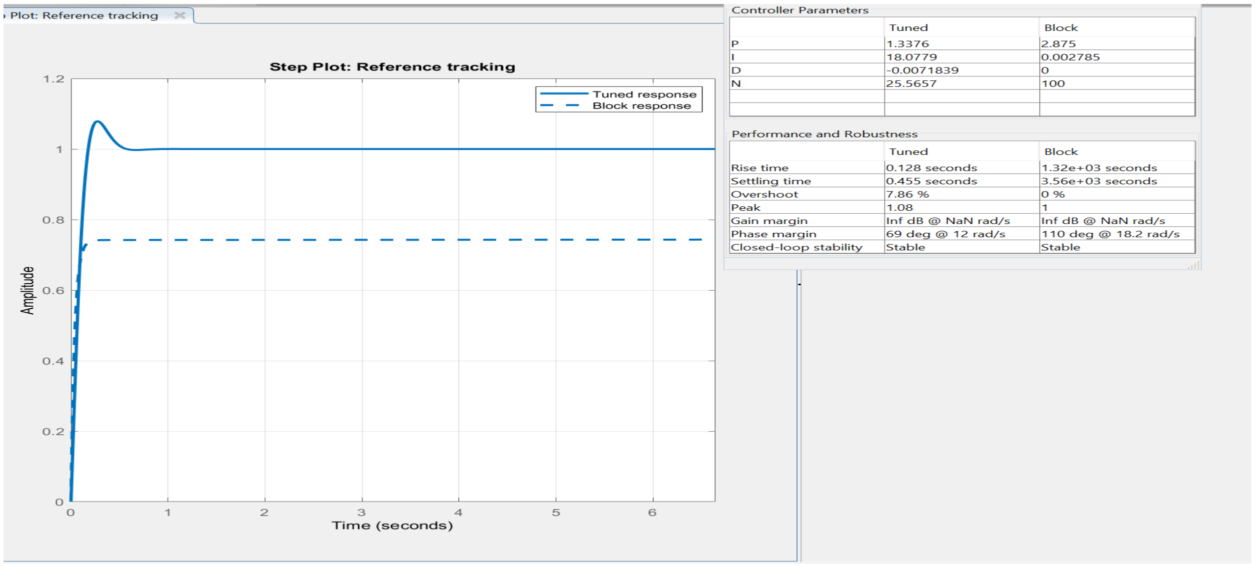

Figure 6: Simulated Response of PI controller

The final step was to now adjust the derivative component of the controller while holding the proportional and integral component values from the previous two steps constant. These values gave the final waveform in the next figure.

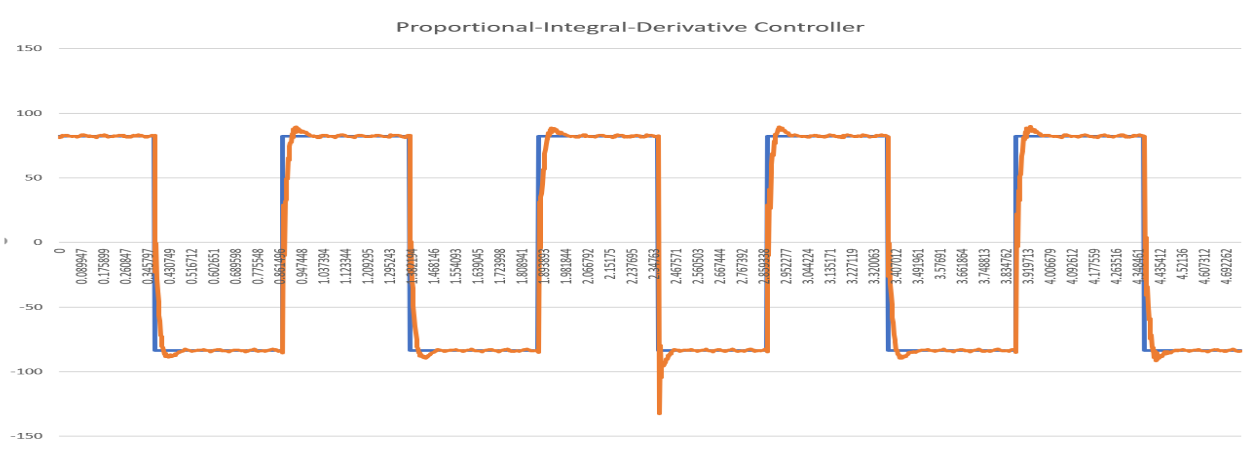

Figure 7: Rotational Speed of Turntable from PID Controller

The final simulated output response was made using Matlab PID Tuner. Theoretical response and the corresponding parameters are displayed in the final figure.

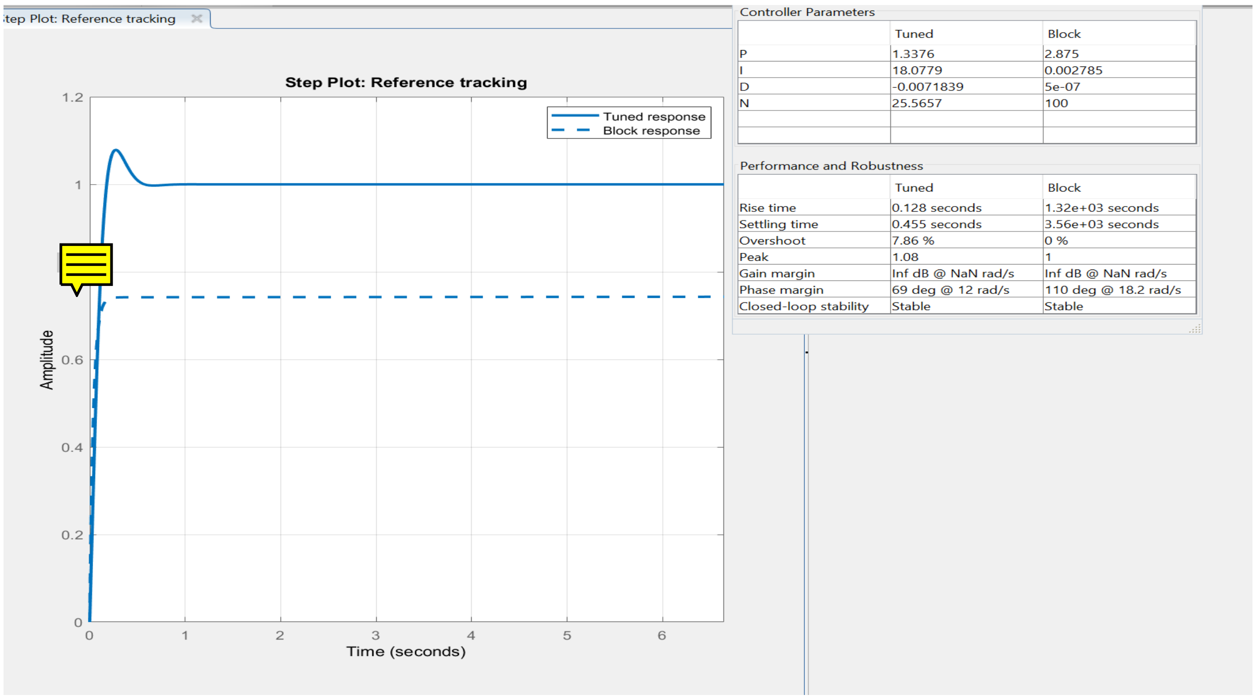

Figure 8: Simulated Response of PID Controller

From the final PID graph of the output the settling time, rise time, and percent overshoot were determined. It is shown as follows:

Settle Time: 0.455 sec <500ms

Rise Time: 0.128 sec <200ms

Overshoot: 7.86% <10%

Discussion

The goal of this lab was to design a PID controller so that the system output satisfies the following performance specifications when subjected to a unit step input:

1) Percentage overshoot is to be less than 10%
2) Settling time to within 2% of the final value is to be less than 500 ms
3) Rise time is to be less than 200 ms.

This apparatus was simulated using a Matlab program which can be seen in the results section of this report. The values we obtained all fell within the desired parameters. This program was found to correlate very closely with the data collected experimentally. The program appeared to be more stable than that of the experimental data and there was almost no error. This is likely due to the fact that analog circuits tend to be highly sensitive to noise due to things such as erroneous electromagnetic interference. Thus this is to be expected as experimental data will always have a certain amount of experimental uncertainty.

Conclusion

In conclusion, the experimental results showed that the PID controller for the turntable speed control system satisfied the performance specifications. The values that we obtained correlated well with the theoretical results found using Matlab. Although they were not a perfect match this can be chalked up to the slight variations in the input voltage as well as the lack of isolation given to the circuit.

References

1. Electrical Technology. “What Is a PID Controller, Their Types and How Does It Work?” *ELECTRICAL TECHNOLOGY*, 16 Nov. 2018, www.electricaltechnology.org/2015/10/what-is-pid-controller-how-it-works.html#real-time-pid-controllers.
2. Engineering, Omega. “Temperature Control Using PID.” *Https://Www.omega.com/En-Us/*, 17 Apr. 2019, www.omega.com/en-us/resources/pid-controller-application-examples.
3. Professor Machtay, MEC 411 Lab 1 Digital PID Speed Control of a Turntable, Department of Mechanical Engineering, State University of New York, Stony Brook, September 2019
4. Yaskawa Electric America. “Introduction to PID Control.” *Machine Design*, 5 Feb. 2016, www.machinedesign.com/sensors/introduction-pid-control.

Appendix:

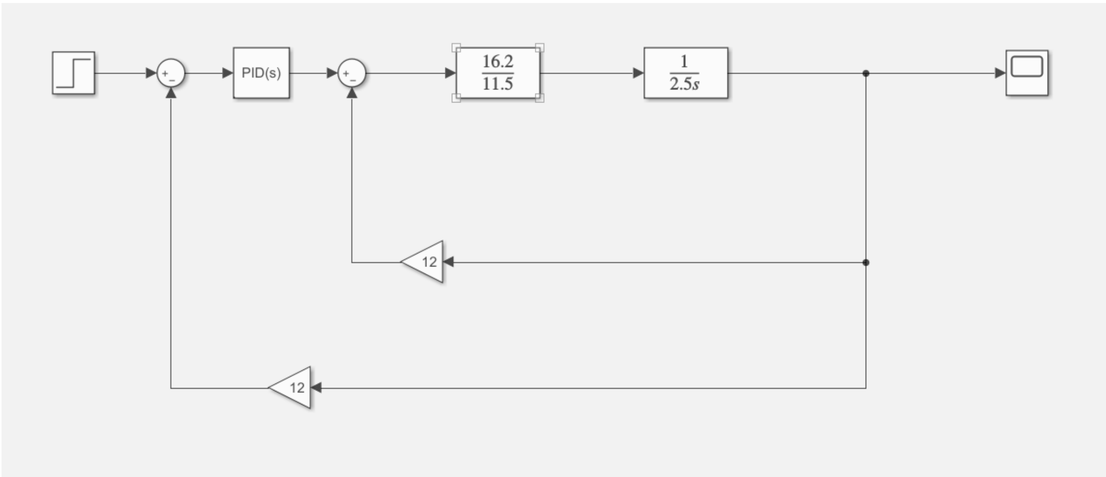

Figure 9: Matlab PID tuner.

Comment Summary

Page 2

1. remove pg

Page 3

2. No mention of your findings.

Page 4

3. Cite handout
4. Derive final expression of TF.

Page 6

5. cite lab handout.

Page 8

6. One complete waveform is enough.

Page 10

7. You do not tune controller parameter in MATLAB plot.

Page 11

8. You do not tune controller parameter in MATLAB plot.
9. Amplitude of what?

Page 12

10. Use bullet point format.

Page 13

11. incorrect format