MEC 411

LAB #3 : Turntable Speed Control

System Design

Group #11

Matthew Drzymala

Deeba Farhin

Qian Ting Huang

Date: November 26, 2019

Table of Contents:

Abstract

In this experiment, a PID turntable control system is implemented. A combination of an analog circuit and a Labview code with a Data Acquisition (DAQ) interface is used to carry out the experiment. The system is to satisfy the following design and feature specifications: Percentage overshoot is to be less than 10% of the unit step input, the settling time to within 2% of the final value is to be less than 500ms for a unit step input and the rise time is to be within 200ms for a unit step input. The Labview code is to display both the input and output waveforms, include a way of displaying the value of percentage overshoot in real time along with a display element to indicate whether the system is satisfying the percentage overshoot requirement or not. The system is to also record the required values associated with the input and output waveforms using a method that prevents synchronisation of the waves with respect to time. The system should only be able to operate when there is someone present to do so. A 1Hz square wave is supplied to the circuit and the input and output data are collected using the Labview software.[2] The experimental results are compared with MATLAB generated values and also with labs 1 and 2. The system is evaluated to see if it meets the specified design and feature requirements. The end values of K_c, T_i, T_d are 0.75, 0.00045, and 0.0000001 respectively.

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. [1]

The turntable is controlled by a DC motor with a tachometer. The tachometer has the parameters listed in table 1.

Table 1: DC Motor and Tachometer.

Performance Parameters	Value	Units	Tolerance
Rated voltage D.C.	@12.0	Volts	-
Rated current	0.944	AMPS	-
Rated continuous torque	14	OZ-IN	-
Peak (momentary) torque	70	OZ-IN	-
Rated speed	800	RPM	± 15%
Rated continuous power out	8.3	Watts	± 15%
No load speed	1000	RPM	MAX
No load current	0.340	AMPS	MAX
Back EMF constant (K_b)	12.0	V/KRPM	± 10%
Torque constant (K_m)	16.2	OZ-IN/A	± 10%
DC armature resistance (R_a)	11.5	OHMS	± 15%
DC armature inductance (L_a)	3.16	mH	± 15%
Armature temperature	155	DEG. C	MAX
Unit weight	12	OZ	MAX
Tachometer voltage gradient	0.52	V/KRPM	

Table 1: DC Motor and Tachometer [1]

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

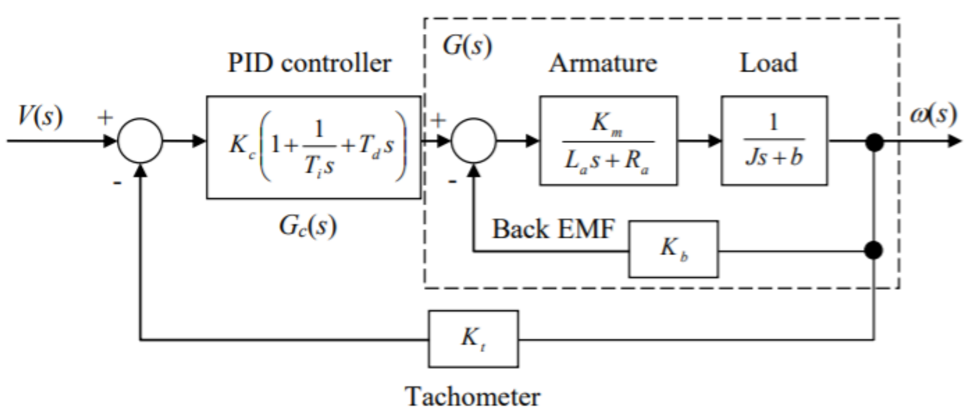


Figure 1: Block Diagram of a PID turntable speed control system [1]

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:

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

Where

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

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \tag{3}$$

The analog circuit is built as shown in figure 2.

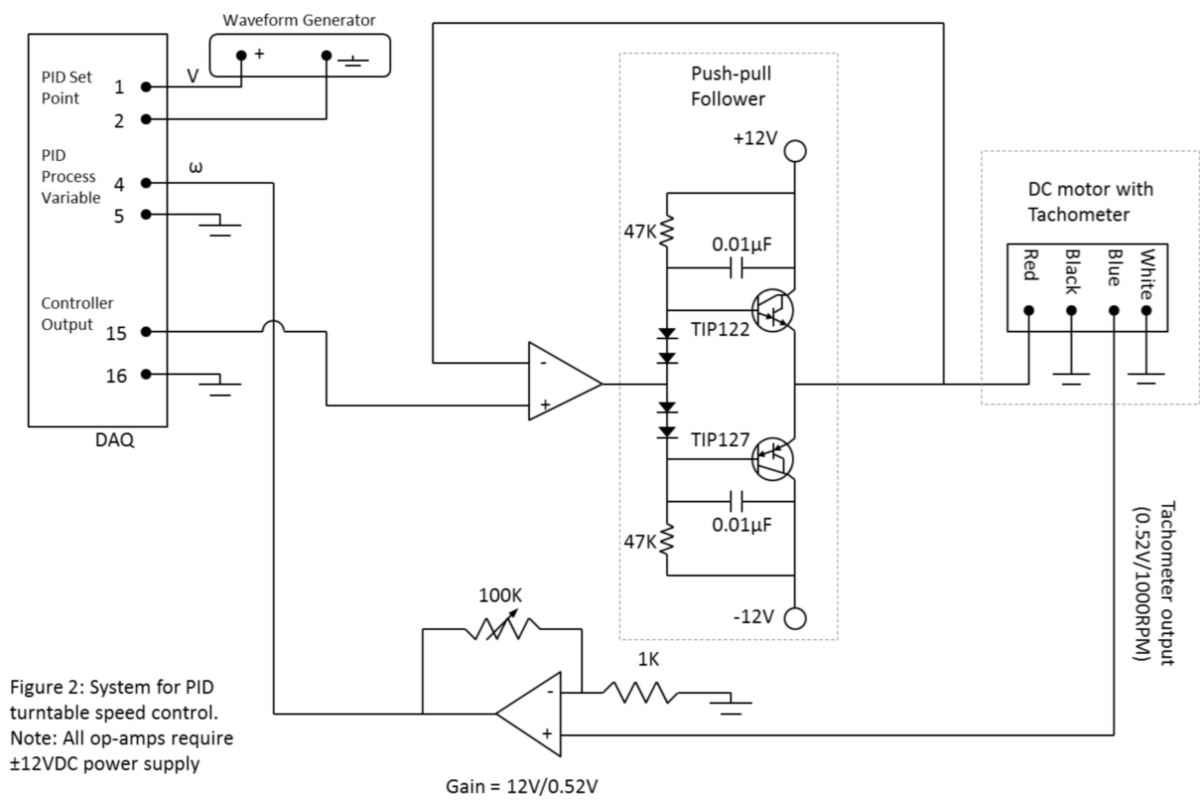


Figure 2: Circuit for PID Turntable Speed Control.[1]

The PID turntable speed control system is comprised of several components; a DC motor, EMF, and tachometer. The DC motor has a rotating armature and there is both a load and back EMF. The tachometer then acts as a sensor for the rest of the system while the EMF is simply a sensor for the DC motor. Now to derive the transfer function for this system begin by analyzing the plant section first. Combining the load and armature gives the following equation.

$$G = \frac{K_m}{L_a s+R_a} \times \frac{1}{Js+b} = \frac{K_m}{(L_a s+R_a)(Js+b)} \tag{4}$$

Now assuming that both La and b are equal to zero the equation can be rewritten which is shown below.

$$G(s) = \frac{\frac{K_m}{(L_a s+R_a)(Js+b)}}{1+\frac{K_m}{(L_a s+R_a)(Js+b)}} = \frac{K_m}{(L_a s+R_a)(Js+b)+K_b K_m} \approx \frac{K_m}{R_a Js+K_b K_m} \tag{5}$$

From the above equation it is determined that the transfer function of the digital PID controller is this equation below.

$$G_c(s) = K_c(1 + \frac{1}{T_i s} + T_d s) \tag{6}$$

By substituting $G_c(s)$ and $G(s)$ into the transfer function, it will get the following results:

$$T(s) = \frac{\omega(s)}{V(s)} = \frac{G_c(s)G(s)}{1 + K_t G_c(s)G(s)} = \frac{K_c\left(1 + \frac{1}{T_i s} + T_d s\right)\left(\frac{K_m}{R_a Js + K_b K_m}\right)}{1 + K_t K_c\left(1 + \frac{1}{T_i s} + T_d s\right)\left(\frac{K_m}{R_a Js + K_b K_m}\right)} \tag{7}$$

$$= \frac{\frac{K_c K_m(1 + \frac{1}{T_i s} + T_d s)}{R_a Js + K_b K_m}}{\frac{R_a Js + K_b K_m + K_m K_t K_c(1 + \frac{1}{T_i s} + T_d s)}{R_a Js + K_b K_m}}$$

$$= \frac{K_c K_m (1 + \frac{1}{T_i s} + T_d s)}{R_a J s + K_b K_m + K_m K_t K_c (1 + \frac{1}{T_i s} + T_d s)}$$

$$= \frac{K_c K_m T_i s + K_c K_m + K_c K_m T_i T_d s^2}{T_i R_a J s^2 + K_b K_m T_i s + K_m K_t K_c + K_m K_t K_c T_i s + K_m K_t K_c T_d T_i s^2}$$

$$= \frac{K_c K_m T_i T_d s^2 + K_c K_m T_i s + K_c K_m}{(T_i R_a J + K_m K_t K_c T_d T_i) s^2 + (K_b K_m T_i + K_m K_t K_c T_i) s + K_m K_t K_c}$$

This transfer function will be used in MATLAB to determine theoretical results.

Moving onto the amplifier or gain portion of the circuit. If the input is v_1 and the output is v_o then the following equation is valid. This is the portion of the circuit between the 1k and 100k resistors.

$$i = \frac{v_i}{1K} \tag{8}$$

Knowing that the total current at the node is equal to zero the next equation is derived as follows:

$$\sum i = \frac{v_i}{1K} + \frac{v_o - v_i}{R_1} = 0 \tag{9}$$

In the circuit provided the gain is defined as 12V/0.52V. Using this information v_o can then be found using the following equation.

$$v_o = 23.08 \times v_i \tag{10}$$

Now moving onto the Push-Pull Follower. This portion along with the gain were constructed in the first lab and used again for this lab. The Push-Pull Followers main job is to amplify the current. 12 volts traveled through the follower and it amplified the transistors within. The diodes are used to bias the transistors towards the given voltage. While this was able to function correctly, there is a downfall to this set up and that is the crossover distortion. This occurs due to the cut-off region of the transistor/diode. The diagram for this portion is shown in the figure below.

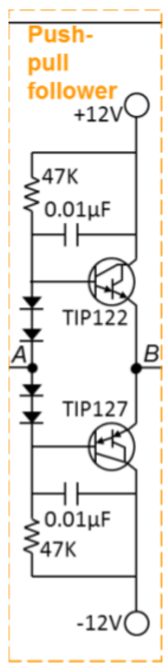


Figure 3: Push-pull follow [1]

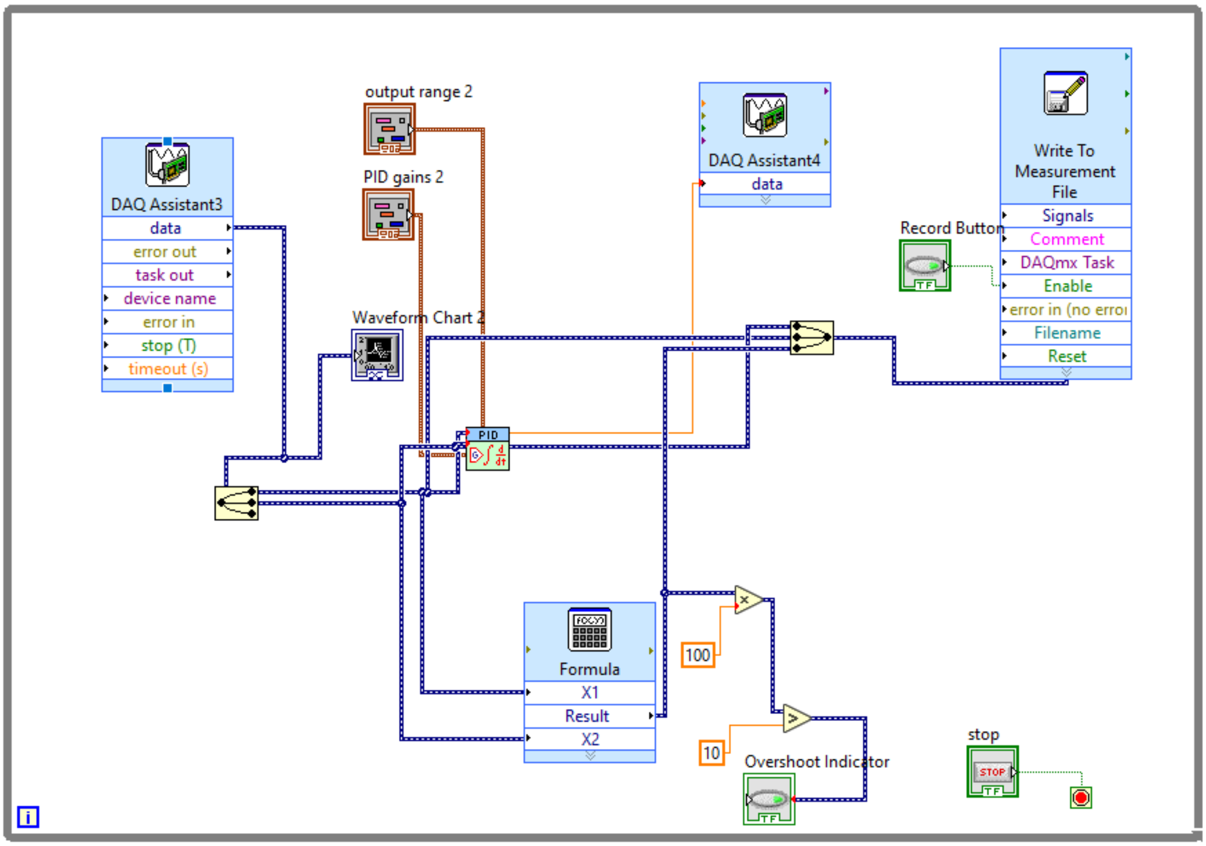


Figure 4: Labview Circuit (Labview)

The figure above shows the Lab view portion of our lab. We tackled the given tasks by using the circuit from lab 1 and combining it with this circuit. This portion begins at DAQ Assistant3. Here both the input wave and the output wave of the breadboard circuit are brought into labview. They are then immediately displayed onto a chart by use of the Waveform Chart 2 function. This allowed us to display the two waves over one another and make live adjustments.

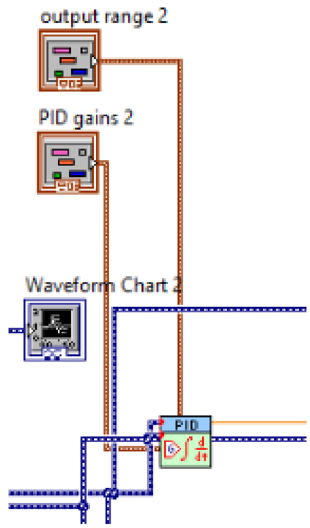


Figure 5: PID Portion of Labview Circuit (Labview)

The image above displays the PID portion of the Labview circuit. Using these components, we were able to manipulate the output wave from the breadboard to then meet the criteria of the lab. Output range 2 was set to 10 for the upper and -10 for the lower. This set the bounds of the controller. PID gains 2 was used to adjust the Ki, Kc, and Kd values which is ultimately what manipulated the output wave. This wave was then output to DAQ Assistant4 which was used to plug the wave into the Push-Pull follower of the breadboard.

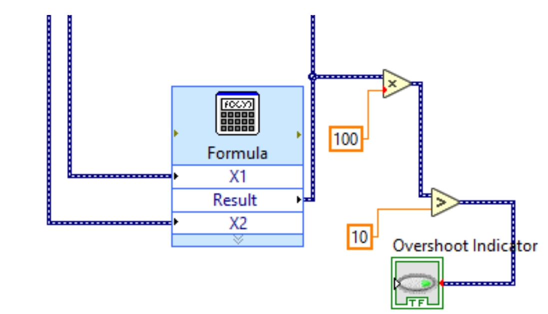


Figure 6: Overshoot Indicator Portion of Labview (Labview)

This portion of the circuit was used to create an overshoot indicator. Our goal was to create a light that would show green when the overshoot was within the desired parameters and red when it was not. Using the formula box we took the absolute value of the input wave and subtracted it from the absolute value of the output wave. When the two are within 10% of each other the light shows as green and red when not.

The final portion of the labview program is the Write to Measurement File. This was used to take the values of the waves and overshoot with respect to time and record it to a ".lvm" file. This data was then used to calculate the rest of the desired info for this presentation.

Experimental Procedure

Equipment:

- DC Motor
- Tachometer
- Breadboard
- Screwdriver
- LM324D
- TIP 122 (NPN Epitaxial Darlington Transistor)
- TIP 127 (PNP Epitaxial Darlington Transistor)
- Wires
- Resistors
- Potentiometer
- Capacitors
- Diodes

- DAQ
- Function Generator
- Computer with Labview software

Procedure:

1. The analog circuit is set up as shown in figure 2 and the labview code was set up as shown in Figure 4.
2. The op-amp and the push-pull followers are given a power supply of +/- 12V and the control circuit is supplied with a 1Hz square wave.
3. The system was inspected to see if all components were wired correctly.
4. Once verified, the system was turned on and the required data was recorded. The values of Kc, Ti and Td were adjusted to provide an output wave as close to the input wave as possible and these values were recorded. Once the desired output wave was obtained, the system was turned off by the operator and all recorded values were saved as a txt file.
5. Excel was then used to analyse any data.
6. Theoretical values were generated using MATLAB and experimental results were compared to this and to results from previous labs.

Results

The goal of this lab is to design a PID controller for the turntable speed control system by selecting potentiometer values for the analog circuit so that the system will meet the following specifications: Percentage overshoot is to be less than 10%. Settling time to within 2% of the final value is to be less than 500ms. Rise time is to be less than 200ms.

After building the Labview program, the K_c (proportional gain), T_i (integral time), T_d (derivative time) parameters were adjusted until the system satisfies the required specifications. The values obtained was 0.7500, 0.00045, 0.0000001 respectively. These values will be used in MATLAB to stimulate responses. Table 2 below shows the output performance parameters produced by these values.

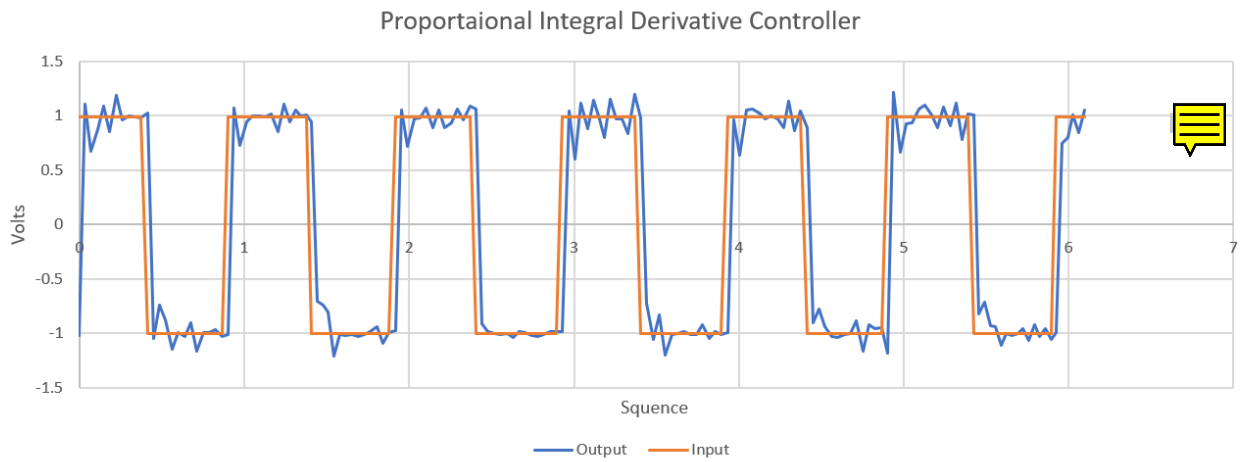


Figure 7: Rotational Speed of Turntable from PID Controller (Excel)

Table 2: Estimated specifications of the experimental output response

PID Controller	Rise Time	Settling Time	Overshoot	Peak Voltage
Values	0.05s	0.327s	9.52%	1.2542 Volts

The following graph was produced by running the MATLAB simulation code (appendix 2) with the transfer equation (equation 7).

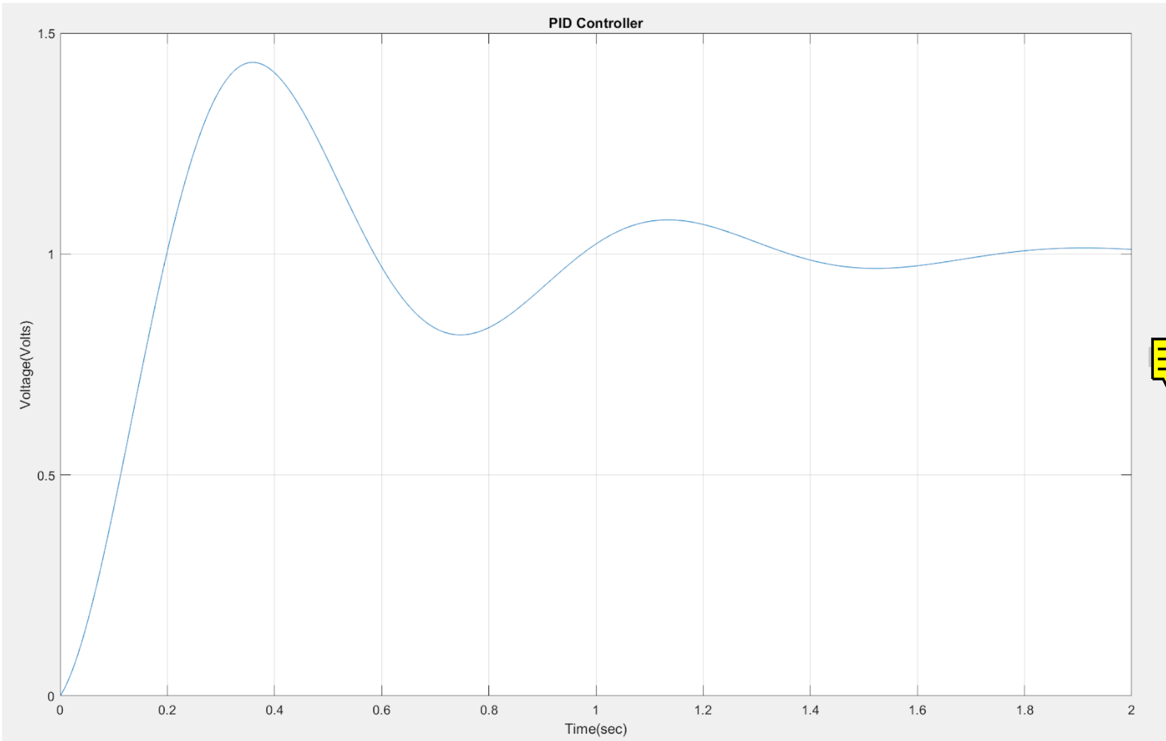


Figure 2: Stimulated Response of PID Controller (MATLAB)

Table 3: System Output Performances

	PID Controller
Rising Time (sec)	1.47028
SettlingTime (sec)	1.69603
SettlingMin (volts)	0.81690
SettlingMax (volts)	1.43370
Overshoot (%)	41.89050
Undershoot (%)	0.00000
Peak (volts)	1.43370
PeakTime (sec)	0.36000

Error Analysis

One of the reasons for the large discrepancy in the percentage overshoot values could be because many of the components used in the speed control system have high tolerance values. Therefore, there is more room for errors to occur. Potentiometers used are prone to mechanical and electrical degradation affecting the performance and accuracy so they are not suitable for continuous use. [4]

Below is a table of the tolerances of all the parameters used in the control system:

Table 4: Tolerances of performance parameters

Performance Parameters	**Tolerance**
Rated speed	±15%
Rated continuous power out	±15%
Back EMF constant (K_b)	±10%
Torque constant (K_m)	±10%
DC armature resistance (R_a)	±15%
DC armature inductance (L_a)	±15%
All resistors	±5%

Discussion

Compare the experimentally determined results with those found from labs 1 and 2, compare and explain the differences between the three control systems.

In each of the labs we had three requirements in common with each other.

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.

In some of the labs this criteria was met while in others they were not. In lab 1 the results obtained met all three of the desired parameters. In lab 2, each of the P, I, and D controller portions were broken into parts. All the parameters were met with the P controller but the PI and PID controllers did not meet the desired values for the overshoot. The overshoot was as high as 42% which was determined to be most likely due to the adjustable resistor within the derivative gain to not be adjusted properly.

In lab 3 the percentage overshoot was greater than 10% at times and the settling time was greater than 2%. This is determined to be due to the labview circuit that was made. This can be determined by knowing that the breadboard circuit from lab 1 was reused. In this scenario all three of the given design specifications were met while utilizing the given software to adjust the PID controller. Upon some investigation there appears to have been significant noise when using the DAQ assistants. Even after spending hours adjusting the Kc, Ti, and Td values this could not be eradicated. None of the group members had much previous time using the labview software and it is most likely that this was the reason for failure. It is also possible that there were excess leading wires that could have caused the excess noise which could have been removed with more experimentation.

From this comparison it can be seen that lab 1 was the best set up. This one was the easiest to construct and simplest to adjust the PID controller values. This is likely why it was done first. Lab 2 required much more breadboard work and thus taught much more about the components that were used. PID was easily adjusted but it was hard to tell if it was being done correctly. Finally, lab 3 was again simple with the breadboard work but brought a new wrench into the equation known as labview. This software is extremely versatile and so requires much effort to learn all the ins and outs.

Cost Analysis:

The table below is a detailed and thorough cost analysis on each of the components used to build the analog circuit used in Lab 3.

Table 5: Detailed cost analysis of components as designed and implemented

Component	Cost per unit	Quantity	Cost
Breadboard	\$5.50	1	\$5.50
Op Amp	\$0.42	1	\$0.42
Diode	\$1.32	4	\$5.28
47K Resistor	\$0.35	2	\$0.70
1K Resistor	\$0.35	1	\$0.35
0.01 μF Capacitor	\$3.10	2	\$6.20
100K Potentiometer	\$2.00	1	\$2.00
TIP 122 Transistor	\$0.90	1	\$0.90
TIP 127 Transistor	\$0.90	1	\$0.90
Wire Kit	\$6.00	1	\$6.00
Alligator Wire Kit	\$4.95	1	\$4.95
		Total Cost:	**\$33.20**

Next, the cost of the Labview program, the cost to train someone to use the program, as well as installation and maintenance fees will be determined. An annual fee of \$100 per license will allow use of the most current LabVIEW Professional Development System suit of software.[6] A function generator like the one used in the lab costs approximately $1,413.00. A metal tachometer is approximately $75. The USB Data Acquisition (NI USB-63430) costs approximately $1645. [5] It will take approximately two hours to train a person to use Labview. Assume an hourly rate of $15 (minimum wage), it would cost $30 for the total cost of training one person.. If it takes approximately 5 hours for installation by a professional that is paid at a rate of $35 per hour, then the cost of installation is $175. Let's assume a maintenance fee of $150 per year. However, maintenance fee really depends on how complicated or how severely damage the system is. Severely damaged systems require the assistance of a professional that needs to be paid at a high hourly rate.

Conclusion

- Not all design requirements were met due to failure to properly build labview software.
- Significant noise that comes from using DAQ assistant should be able to be eradicated using the correct PID values.
- Lab 1 was the best set up due to its simplicity.
- The parameters for K_c, T_i, and T_d are 0.75, 0.00045, and 0.0000001 respectively. Percent overshoot is equal to 41.89050%, settling time is 1.69603 and rise time is 0.81690.
- The experimental results did not agree with the theoretical values. Only some specifications were met.
- By comparing the results of Lab 1, Lab 2, and Lab 3, the control system used for Lab 2 was found to be better. The analog PID controller used in LAb 2 worked more effectively

than the digital software built on Labview because the analog circuit was built and used correctly. And there was the least noise level in Lab 2.

- The discrepancy can be attributed to the noise level from the DAQ device. Since MATLAB simulation does not factor in physical noise of the surrounding, there is a large relative error because the data acquired from the DAQ collected a lot of noise.
- In future investigations, a different Labview program could help produce better experimental results and a different analog circuit arrangement can help reduce noise level collected by the DAQ device.

References

[1] Machtay, Noah. "PID Speed Control of a Turntable." *MEC 411 Lab 1.* Stony Brook University, Fall 2019. Print. November 16, 2019

[2] Machtay, Noah. "Design Project: Turntable Speed Control System Design" *MEC 411 Lab 3.* Stony Brook University, Fall 2019. Print. November 16, 2019

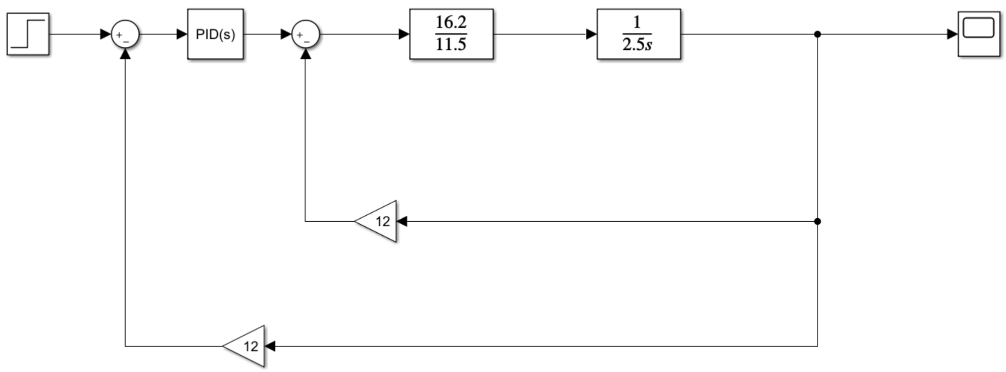
[3] Collins, Danielle. "Overshoot and Undershoot in Servo Control Systems." *Motion Control Tips*, www.motioncontroltips.com/how-to-address-overshoot-in-servo-control/. October 26, 2019

[4] "Potentiometers, Preset Potentiometers and Rheostats." *Basic Electronics Tutorials*, 23 Sept. 2019, www.electronics-tutorials.ws/resistor/potentiometer.html.

[5] Sine.ni.com,. 'NI USB-6343 - National Instruments'. N.p., 2015. Web. 16 Nov. 2019.

[6] National Instruments Corporation, 2012, "Using the DAQ Assistant to Automatically Generate LabVIEW Code", Austin, TX

Appendix



Appendix 1: MATLAB PID block diagram

PID Controller

```
clc
clear all
close all

km = 16.2*0.02835*9.81*0.0254;
Ra = 11.5;
J = 2.5*0.02835*9.81*0.0254^2;
b = 0;
kb = 12/(1000*2*3.14/60);
kt = 12/(1000*2*3.14/60);

kc = 0.75;
Ti = 0.00045*60;
Td = 0.00000001*60;

t = 0:0.001:2;
num = [kc*Td*km kc*km kc*km/Ti];
den = [Ra*J+kt*km*kc*Td Ra*b+kb*km+kt*km*kc kt*km*kc/Ti];
sys = tf(num,den);
step(kt*sys,t);
y = step(kt*sys,t);
PID_Controller = stepinfo(y)

plot(t,y);
title('PID Controller');
xlabel('Time(sec)');
ylabel('Voltage(Volts)');
grid on
```

Appendix 2: MATLAB Simulation Code for PID Controller

Comment Summary

Page 3

1. Unit?

Page 13

2. Legends?

Page 14

3. Show response with square wave.

Page 15

4. Please compare experimental and theoretical results in this section.

Page 18

5. A brief summary of the lab is needed.