Course Number and Name: ME 747 – S-Lab	
Semester and Year: Fall 2019	**Name of Lab Instructor:** Alireza Ebadi
Lab Section and Meeting Time: Section 3B – W 2:10 pm – 5:00 pm	**Report Type:** Internal Group
Title of Experiment: Velocity Control of a DC Brush Motor Under Load	
Date Experiment Performed: 11/13/2019	**Date Report Submitted:** 11/30/2019
Names of Group Members: Zack Shelby Lucas Simmonds Charlie Nitschelm	**Grader's Comments:** **Grade:**

Table of Contents

Contents

Table of Contents 2

Contents 2

Table of Figures 2

Objectives 3

Executive Summary 4

Results and Discussion 8

Conclusion 19

Lucas 19

Zachary 19

Charlie 19

Reference 20

Appendix 21

Part 2 21

Part 3 21

Part 4 23

Peer Effort 28

Table of Figures

Figure 1 - Sketch of system used during data collection 5

Figure 2 - Electrical Diagram of P, I and PI Controllers 7

Figure 3 - Displays MUT Voltage vs Angular Velocity 10

Figure 4 - Tachometer Output and Motor Input due to step change 11

Figure 5 - Represents decay rate from the disturbance load of 5 Ohms 12

Figure 6 - Root Locus Plot of P Motor Velocity Controller 14

Figure 7 - Root Locus of I Motor Velocity Controller 14

Figure 8 - Root Locus of PI Motor Velocity Controller 15

Figure 9 - Step Response of P Motor Velocity Controller 15

Figure 10 - Step Response of I Motor Velocity Controller 16

Figure 11 - Step Response of PI Motor Velocity Controller 17

Objectives

Proportional, integral and proportional-integral controllers were used to investigate how each impact the rotational speed and response of a DC brush motor during a step change to a desired input velocity. The controller should help the motor hit the desired input and maintain that speed regardless of the torque load on the motor. The parameters of the DC brush motor can be calculated theoretically knowing the system parameters and assuming that a brush motor follows a first order transfer function. The motor will be driven by a power op-amp and the speed of the rotation during the test is calculated using a tachometer. To add a back-torque, another motor will be used which will be used to determine the motor constant experimentally. This information will help us make connections between how the open and closed loop systems compare with speed of response and the steady state error between the desired and actual motor velocity. By knowing how a controller affects the overall motor rotational speed response, you can then select the controller that best fits the situation and environment the motor will be subjected to.

Executive Summary

A primary purpose of this test was to identify the main system parameters including the systems moment of inertia and damping coefficient. Measuring the overall system constants and back solving to find the parameters given the experimental data, the theoretical and experimental values of the moment of inertia and damping coefficient could be compared.

	Moment of Inertia	Damping Coefficient
Theoretical	0.0004 oz-in-sec^2	0.578 oz-in/KRPM
Experimental	0.00035 oz-in-sec^2	0.2 oz-in/KRPM

The moment of inertia values matches up very nicely to one another. However, the damping coefficient seems to be a little over double the specification sheets value given from the manufacturer. The reason behind this may be due to ignoring the electrical time constant.

The next primary purpose of the test was to compare and contrast the overall response of the speed of the motor given a desired velocity speed, and how different controllers deal with getting to that desired speed and maintain it given any torque on the motor. Using the experimental data of each controller in action, the speed of response of the motor speed given the input velocity, the overall settling time, and the steady-state error were calculate for each.

	Speed of Response [s]	Settling Time [s]	Steady State Error [%]
Open Loop	0.0082	0.033	29.49
P Controller	0.0586	0.2346	19.14
I Controller	0.0492	0.1967	0.58
PI Controller	0.0572	0.2286	0.51

It is obvious that an open-loop system offers the quickest response and settling time, but the steady state error of not using a controller is huge. As we integrated a proportional controller, we saw a much slower response time, but the steady state error decreased by nearly half of what it was. The I and PI controllers were similar and had a slightly faster speed of response and greatly reduced the steady state error to less than 1 percent of the desired velocity value. Overall, the PI controller had a slightly better steady state error than the I controller, but also took slightly longer than the I controller to settle.

Theory and Experimental Methods

To begin the investigation of the P, I, and PI control of the rotational speed for a DC brush motor, the parameters of this dc brush electro-mechanical motor will need to be identified. In particular, the motor back emf constant, K_e, will be found by inputting voltages in increments of 1V from 2-10V into the system while utilizing a BK power supply (+/- 15 Volts). This will allow the MUT to operate at different speeds and the output voltages are then recorded. These output voltages from the tachometer and the MUT are used to help find and verify the specification sheet's tachometer sensitivity and torque constant.

A relationship between the angular velocity and output voltage from the MUT can be made to determine the motor voltage constant. Below, Equation 1 displays how the motor constant can be calculated.

$$\frac{K_{tach} * e_m}{Tach\ Output} = K_e \tag{1}$$

Essentially, tachometer outputs are converted using the tachometer's sensitivity given in the specification sheet to find the angular velocities. The slope is calculated from the data between the angular velocity measurements and the MUT voltage outputs to obtain the motor voltage constant. Next, the torque constant, K_t, can be calculated equating this value to the value of the motor voltage constant. However, a unit conversion will have to be made in order to get this value into units of torque. Once converted, the values just described will be used to obtain the system parameters.

The step response can be measured by powering the front motor (MUT) with a power op amp while implementing a function generator amplitude of +/- 8 volts so that the tachometer output will have a +/- 4 volt square wave output. During a step change, the outputs of both the tachometer output and motor input can be observed and analyzed to define the parameters. A system of equations from the MUT will be utilized to help determine the stall torque. Knowing that the back-drive motor is coupled with the MUT, the system of equations along with set up of the system is presented below.

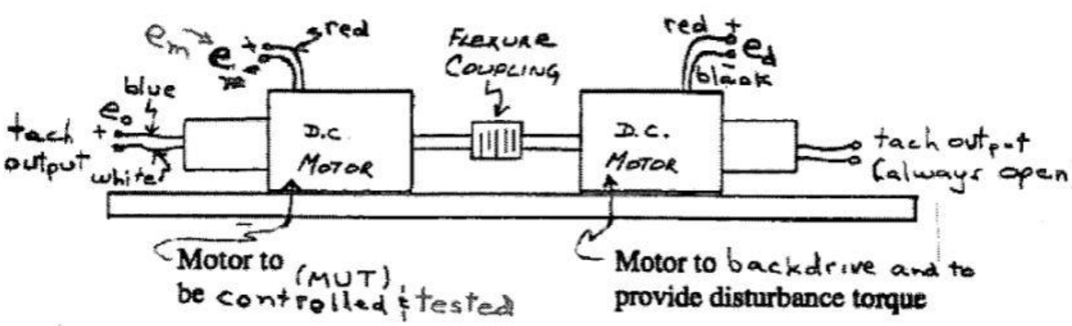

Figure 1 - Sketch of system used during data collection

$$e_b = K_e * \omega \tag{2}$$

$$T_g = K_t * i \tag{3}$$

(4)

$$e_i - iR - K_e\omega = 0$$

$$T_L - 2B\omega - K_t i = 2J\dot{\omega} \tag{5}$$

The stall torque, T_L will occur when the angular velocity value is equal to zero due to a stall in a motor means there is no rotation occurring. Knowing that the angular velocity will be zero, Equation 6 will be simplified as shown below.

$$T_L = K_t * \frac{e_i}{R} \tag{6}$$

Data collected from the step change can be used to determine the K and τ of the motor. System gain will need to be calculated first which will result in the K_{motor} value when divided by the tachometer sensitivity. A ratio of the final input motor voltage and final tachometer output will result in the system gain. Dividing this by the tachometer signal will result in units of KRPM/Volt. Equation 7 displays this relationship.

$$K = \frac{e_{mut}}{e_{tach} * K_{tach}} \tag{7}$$

The time constant of the motor is found by observing the step input taken from the tachometer output voltage. By using the 63.2% method, the time constant will related to 63.2% of the tachometers max amplitude reading. These calculated parameters can be utilized to calculate the moment of inertia, J, and the damping coefficient, B, of the system. When the electrical time constant of the system is ignored, the system of equations can be rearranged as shown below in Equations 8 and 9.

$$B = \frac{K_t - K_e K_t K_{Motor}}{2K_{Motor}R} \tag{8}$$

$$J = \frac{\tau(K_e K_t + RB)}{2R} \tag{9}$$

All variables within the equations have been previously described and accounted for when determining the J and B values. The experimental values can then be compared to those given in the specification sheet for this dc motor to determine the accuracy.

A disturbance load of a 5 Ohm resistor is implemented on the open loop motor which will cause a decay rate in the voltage input of the tachometer once it is turned on. The decay time constant will be calculated in a similar manner as that of the motor. However, the amplitude of the open loop motor from the disturbance implementation will be multiplied by the 36.8% to determine the time constant.

The next steps of the experiment include converting the system from open-loop to closed-loop by implementing a tachometer and using its output voltage as the feedback signal. A 741 Op-Amp with various resistors and capacitors are used alongside the tachometer to provide system compensation in the form of P, I and PI controllers. The diagram below shows the different configurations used to achieve the different controllers.

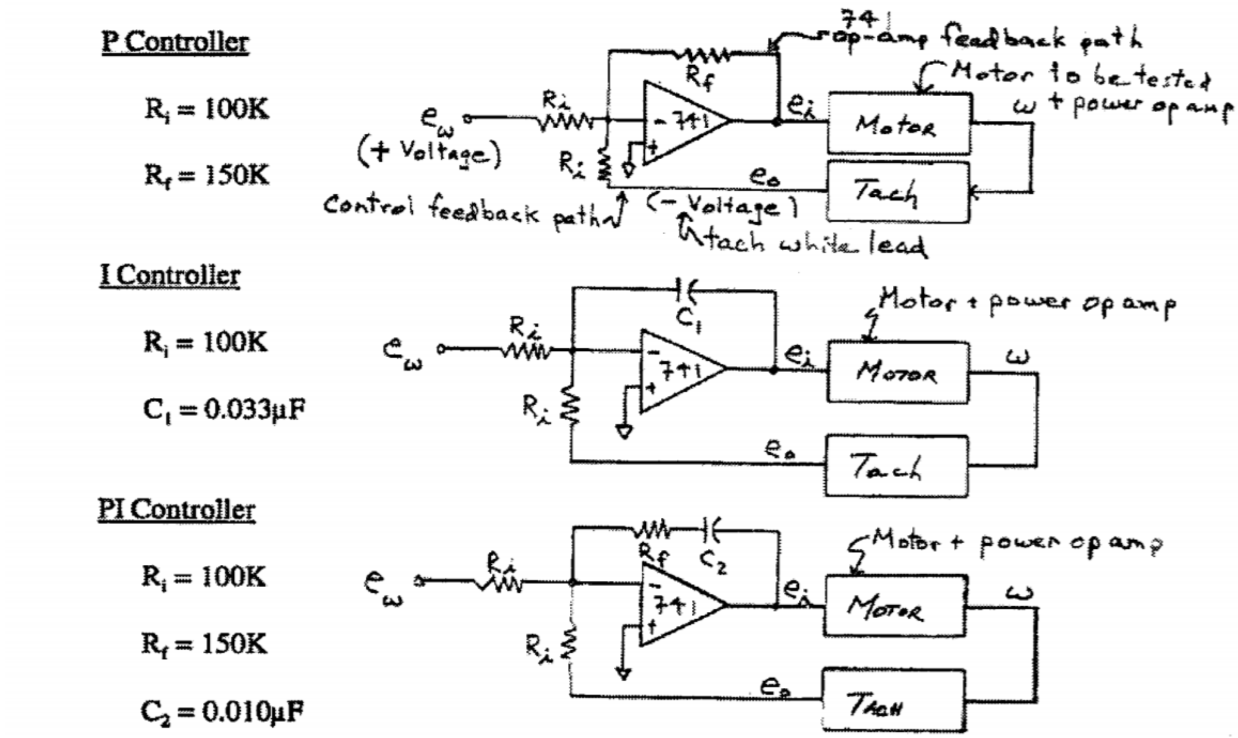

Figure 2 - Electrical Diagram of P, I and PI Controllers

Each configuration was connected to the power supply of +/- 15 volts and the function generator with a 0.4 Hz square wave and a +/- 4 volt amplitude. The trigger was set to capture a step response to the 0.4 Hz square wave being inputted through the function generator. Using the experimentally determined gain and time constant values from the open-loop system, the transfer function of each controller is determined and used to graph the root-locus plot and its corresponding roots and the theoretical step response to the input square wave. The observed step responses can be used to find the motor speed of response for each different controller. Because the P controller exhibits a first order step response, the time constant is found using the 63.2% method. The P and PI controllers exhibit a second-order step response so the time constant must be calculated using the formula below,

$$\tau = \frac{1}{\omega_n \xi} \tag{10}$$

Where the natural frequency is found by using the difference of the first two peaks and dividing it by the period of the first two peaks. The quotient of these two values are multiplied by 2π to achieve units of rad/s. This can be seen by using the formula below,

$$\omega_n = \frac{y_{p,2} - y_{p,1}}{t_{p,2} - t_{p,1}} * 2\pi \tag{11}$$

The damping ratio is found using the percent overshoot method. The percent overshoot can be found by taking the quotient of the difference between the first peak and the final value and the final value and multiplying by 100. This value corresponds to tabulated values in Figure 8-13 from the textbook *System Dynamics* which provides a relationship between damping ratio and percent overshoot. The formula used to calculate percent overshoot can be seen below,

$$\%\,O.S. = \frac{y_p - y_{fv}}{y_{fv}} * 100 \tag{12}$$

Furthermore, the settling time can be calculated by multiplying equation 10 by four as seen below,

$$\tau_s = \frac{4}{\omega_n \xi} \tag{13}$$

To measure the steady state errors of each controller the input voltage is set to 4 volts and frequency to 1 mHz. The 5-ohm resistive load is switch on and the dynamic response is captured. Using the dynamic response, the average of the initial signal is compared to the average of the signal after the response. This was done using the formula below,

$$\%\,Error = \frac{Initial - Final}{Initial} * 100 \tag{14}$$

Results and Discussion

After collecting data for the tachometer and MUT (See Table 1), the data could then be analyzed using the methods described earlier. The tachometer sensitivity of 3 V/KRPM was used to convert the Tach voltages to angular velocities.

Table 1 - Voltage input with resulting outputs

Input, e_i (Volts)	Tach, e_o (Volts)	MUT, e_m (Volts)	Angular Velocity (KRPM)
1.5	0.265	0.390	0.088
2	0.561	0.867	0.1870
3	1.14	1.756	0.38
4	1.72	2.661	0.573
5	2.38	3.85	0.793
6	2.994	4.85	0.998
7	3.579	5.806	1.193
8	4.195	6.784	1.398
9	4.794	7.768	1.598
10	5.405	8.751	1.802

From there, the MUT voltages were plotted against the angular velocities (See Figure 2). Using MATLAB's 'polyfit' command, a line of best fit was created to help determine the slope of the data set. This slope is also equal to the value of the motor voltage constant, K_e. As a result the motor voltage constant equals 4.905 KRPM/V. The specification sheet lists an average constant of 4.88V within 200mV which means the experimental motor voltage constant falls within the range.

The torque constant, K_t was also accounted for and was assumed to be equal to K_e. However, the units had to be checked since the units for the torque constant will be in oz_f-in/A. After checking the units, the motor voltage constant was multiplied by 141.6 to get the units into oz_f-in/A. As a result, the K_t equals 6.63 oz-in/A while the specification lists a constant of 6.6 oz_f-in/A. It can be said that accurate data measurements were taken due to the precision of the constants calculated for the motor voltage and the torque.

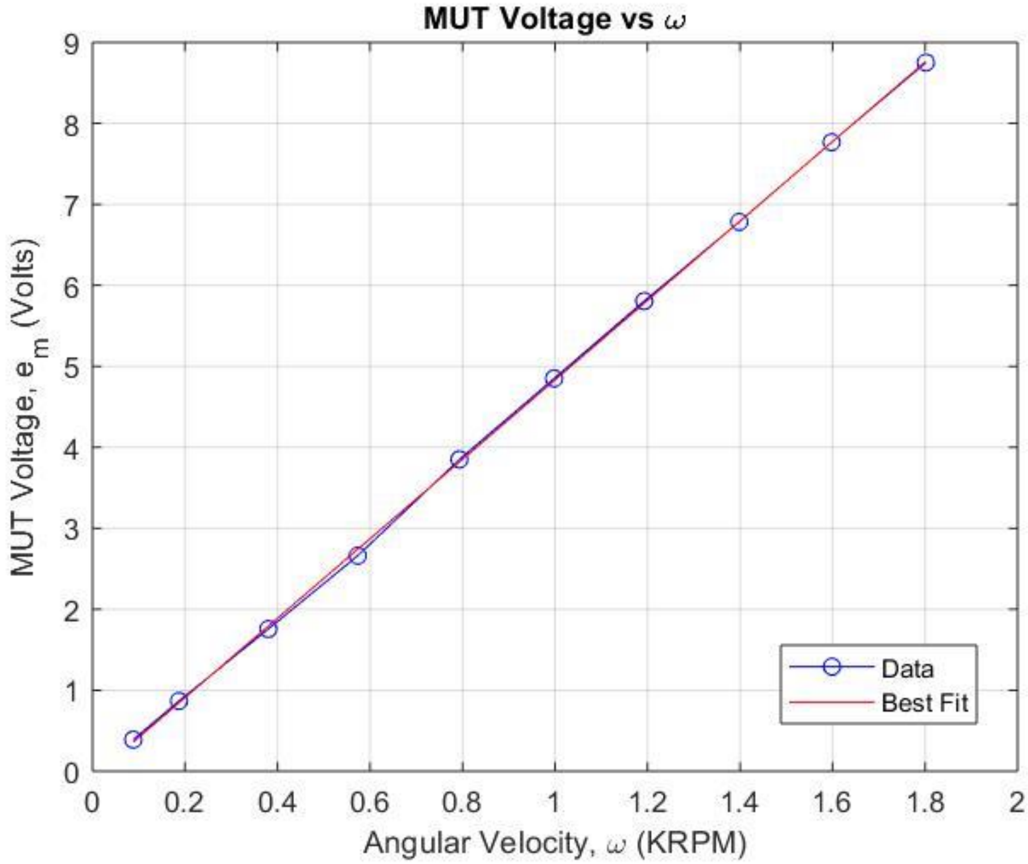

Figure 3 - Displays MUT Voltage vs Angular Velocity

The derived system equations for the motor under test can be found above in Equations 2-4. Using these equations will help determine the torque of the motor when there is zero velocity. This can also be stated as the stall torque and using Equation 6 from above will with the known 6V input, torque constant, and the motor terminal resistance of 3.6 ohms, a stall torque of 11.05 oz_f-in is calculated. The terminal resistance of the motor ranges from 3.6-4.9 Ohms. However, the smallest resistance is desired so the stall torque can be calculated since there will be zero velocity. Using the data collected during the step change (See Figure 3), K_{motor} can be calculated using Equation 7. This results in a value of 0.177 KRPM/V. This value will be needed to determine the damping coefficient as well as the moment of inertia. The time constant was determined by using the 63.2% method of the amplitude from the tachometer. Since the tachometer ranges from the -4V to 4V, the amplitude had to be subtracted by 4 to account for the voltage difference. As a result, the speed of the response came had a resulting time constant value of 0.0089 seconds.

The moment of inertia and damping coefficient values were determined when the electrical time constant was ignored. Equations 8 and 9 were used to solve for these parameters. B was determined to be 0.578 oz-in/KRPM and could then be plugged into Equation 9 to solve for J. The moment of inertia was calculated to be 0.00035 oz-in-sec^2. The table below shows the experimental parameters compared to the theoretical.

Table 2 - Moment of Inertia and Damping Coefficient values

	Moment of Inertia	Damping Coefficient
Theoretical	0.0004 oz-in-sec^2	0.578 oz-in/KRPM
Experimental	0.00035 oz-in-sec^2	0.2 oz-in/KRPM

The moment of inertia values matches up very nicely to one another. However, the damping coefficient seems to be a little over double the specification sheets value. The reason behind this may be due to ignoring the electrical time constant.

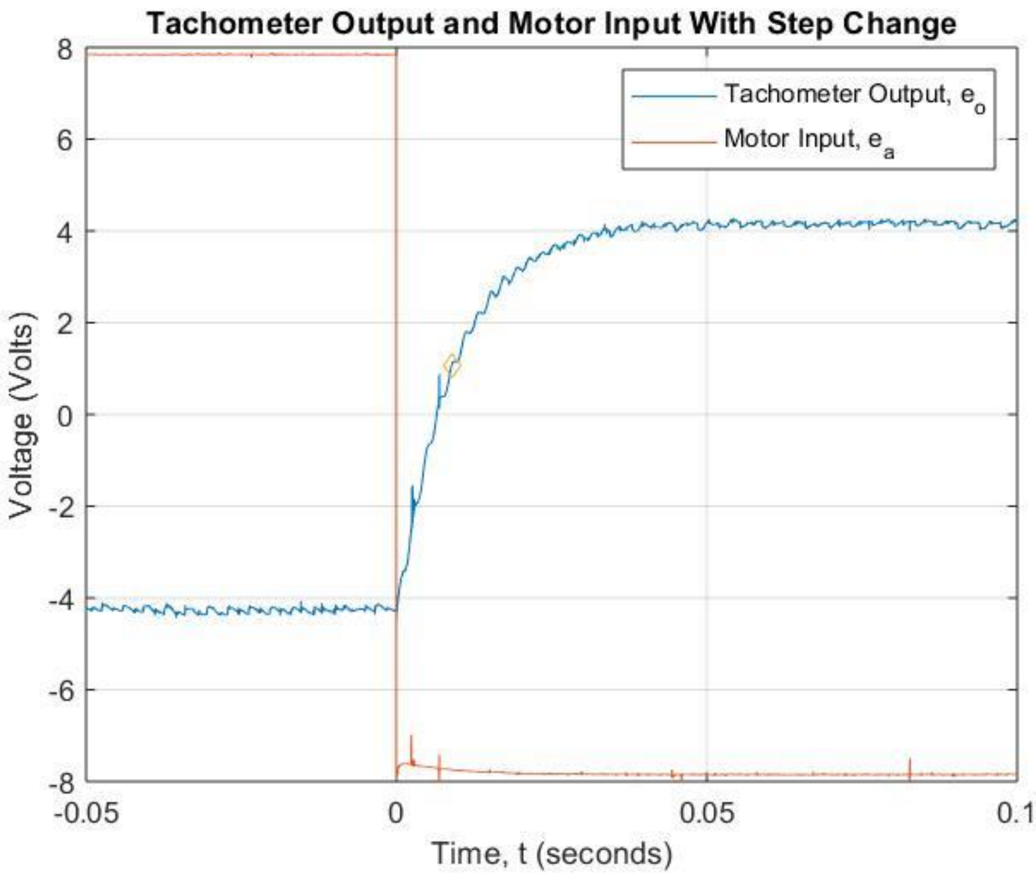

Figure 4 - Tachometer Output and Motor Input due to step change

The figure below (Figure 5) represents the decay rate from the motor when disturbance from the 5 Ohm resistor was used. Using the decay time constant, the time constant of the system when subjected to a 5 Ohm load resistor is 0.0082 seconds. This compares very well to the time constant of the open loop system with disturbance. The reason for any difference could be due to the noise in the decay rate. An aggressive smoothing function was used to try and eliminate noise but still ended up having a significant amount. The settling time was also calculated by multiplying the decay time constant by 4 since that is ideally the time it will take for the response to settle. This resulted in a time of 0.033 seconds which when looking at Figure 3, the response has settled by this time. Utilizing Equation 12 helped calculate the steady state error of the motor when the disturbance load was applied. The steady state error was fairly reasonable by having a value of 29.49%. Since a disturbance load with a 5 Ohm resistor is applied, it would make sense to have this high of an erro.

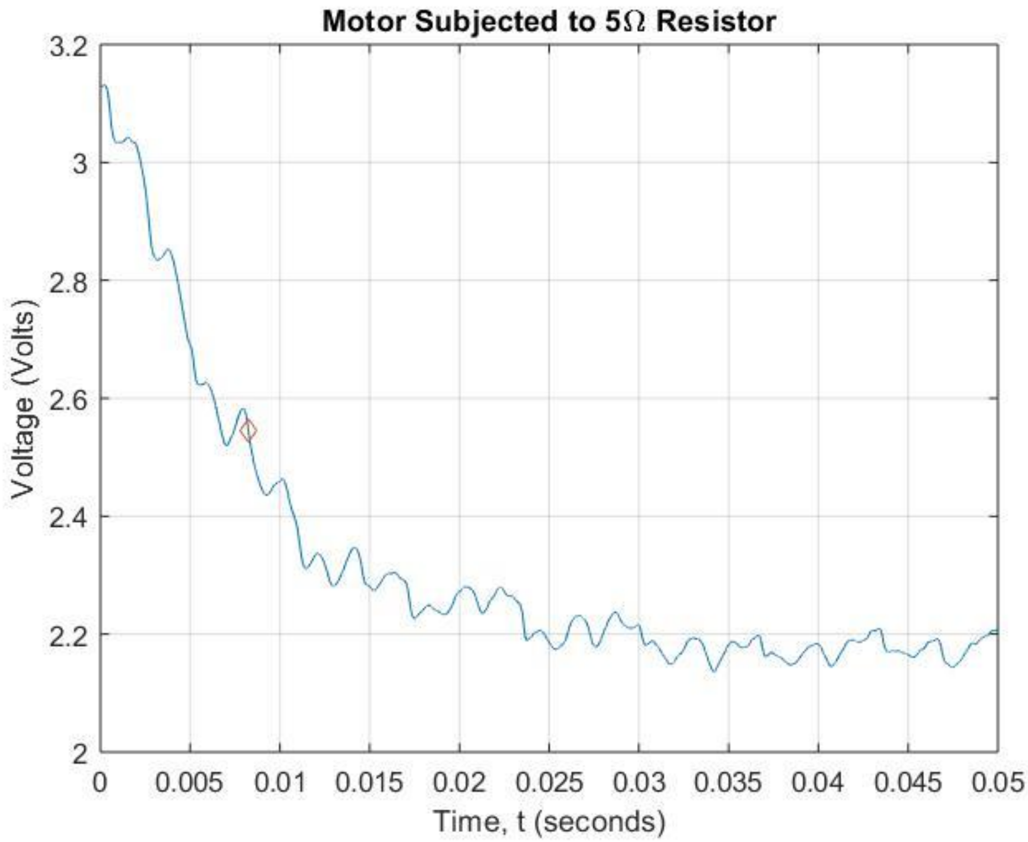

Figure 5 - Represents decay rate from the disturbance load of 5 Ohms

After integrating the tachometer and op-amp circuity to the open loop system, block diagrams were created to simplify the system representation. The picture below shows the block diagram and transfer function for each of the three controllers.

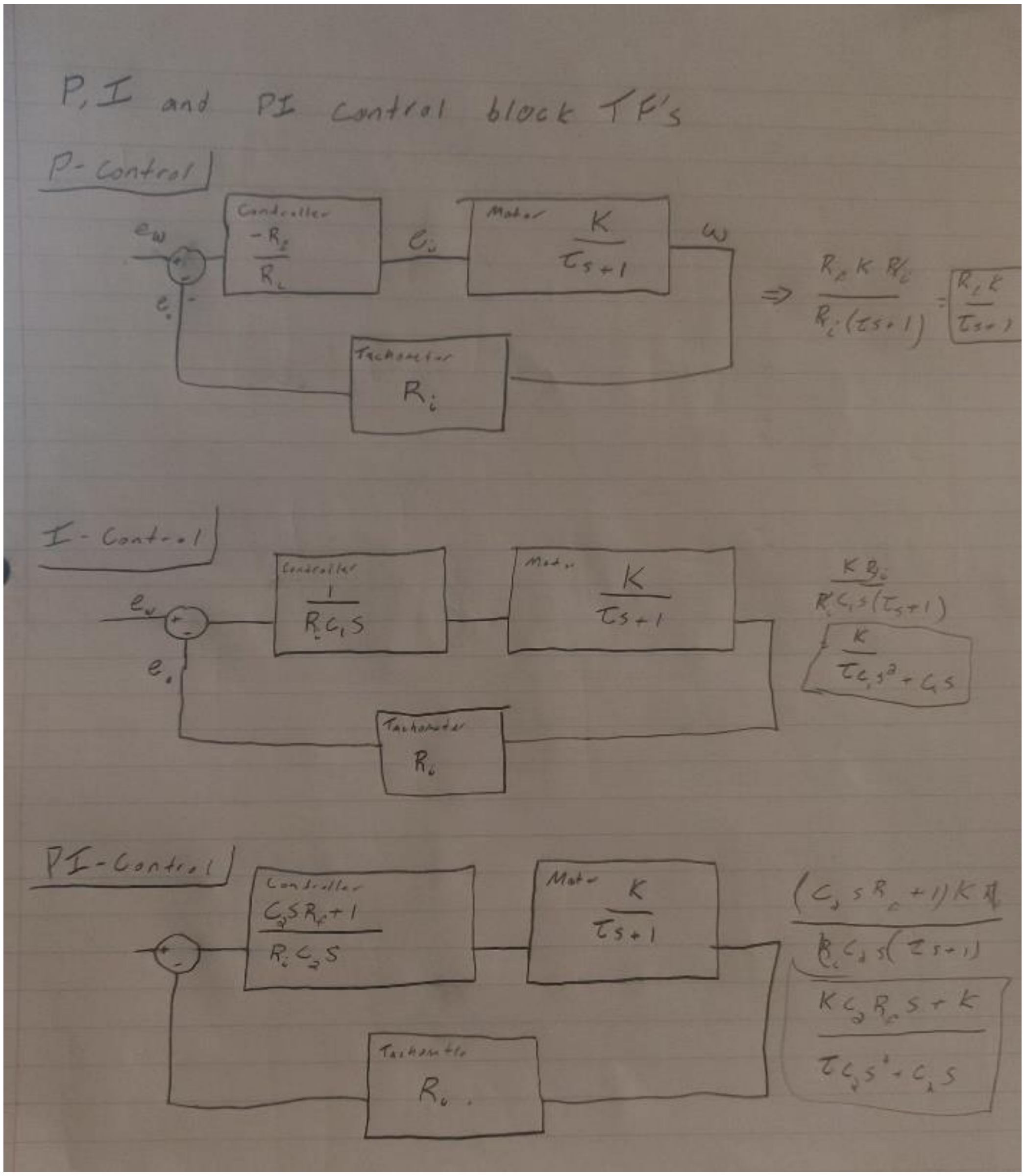

Figure 5 Block Diagram of P, I and PI Controller

Using the experimentally determined values of the gain K, the time constant τ, from the open loop system and the transfer functions, the Matlab command rlocus was used to generate the root locus of each controller assuming there is an imaginary gain that move the root along the locus of each controller. The root locus plots can be seen in figures 6-8 below.

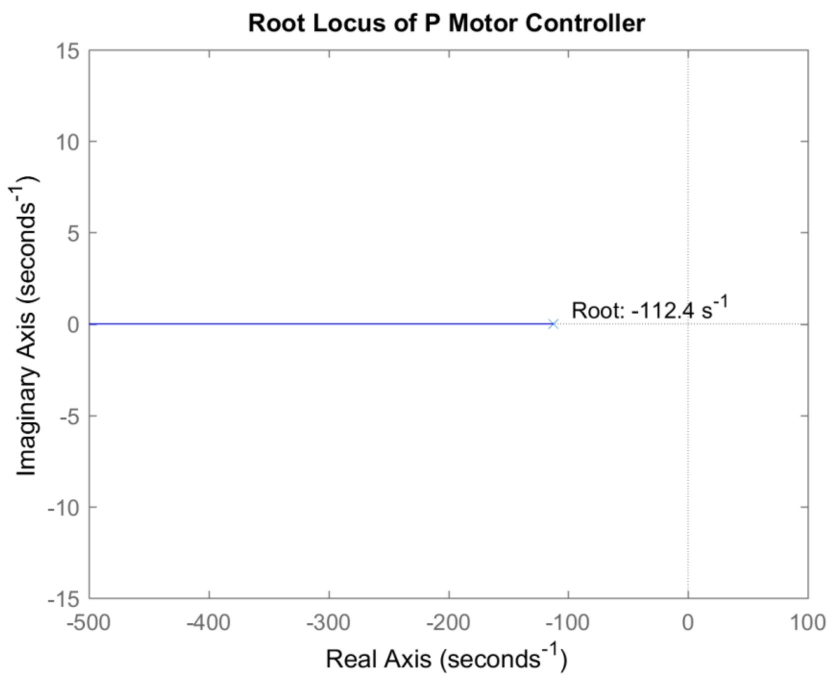

Figure 6 - Root Locus Plot of P Motor Velocity Controller

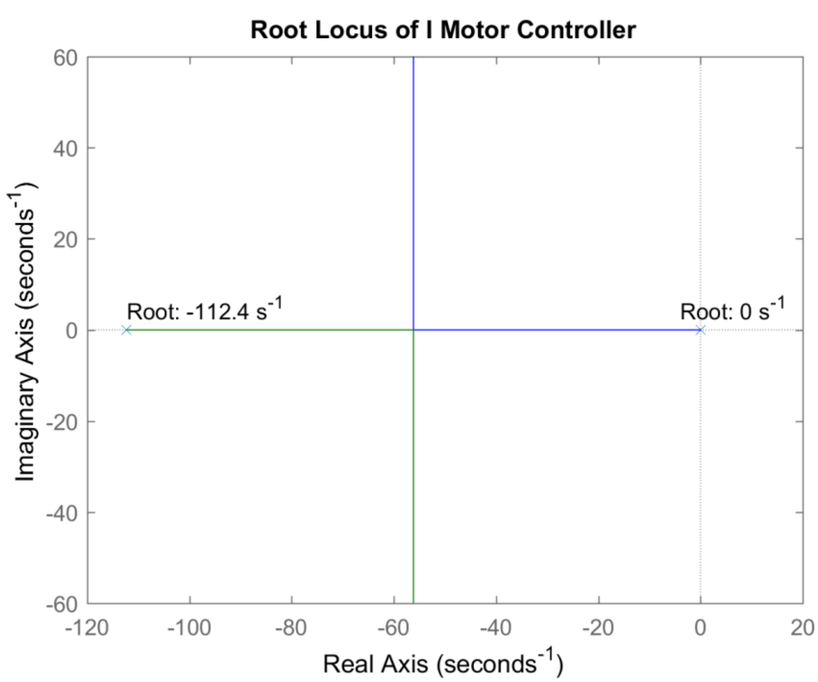

Figure 7 - Root Locus of I Motor Velocity Controller

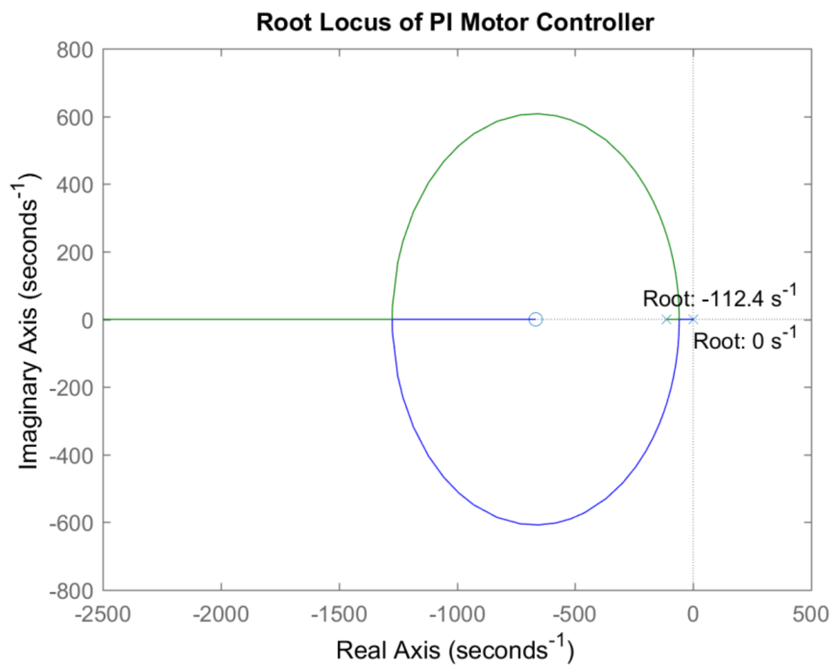

Figure 8 - Root Locus of PI Motor Velocity Controller

Using the transfer function of each controller, the theoretical step response was simulated and compared to the observed step responses captured during the experiment. The figure for the observed and simulated step response for the P controller can be seen below.

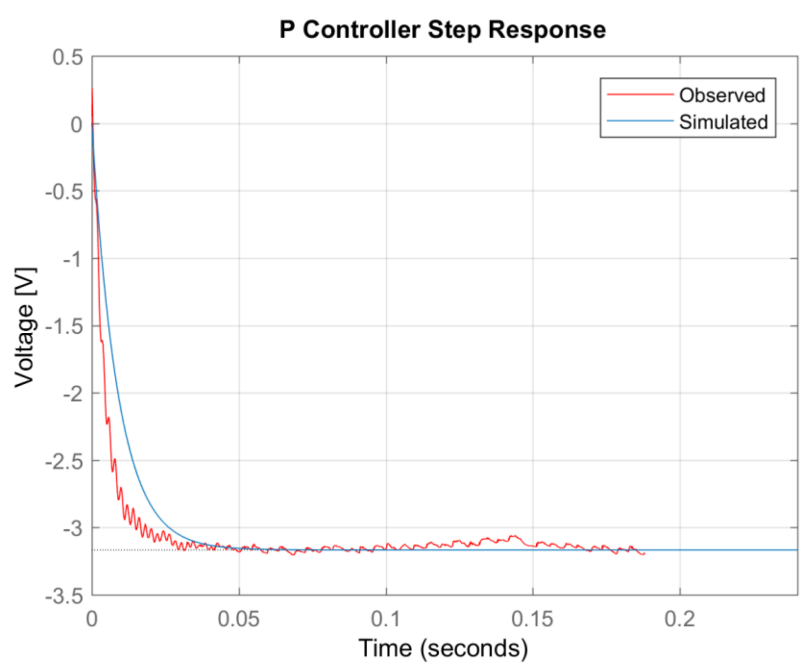

Figure 9 - Step Response of P Motor Velocity Controller

From Figure 9, the response to a step input for this type of controller is first order. This figure also shows that the expected P motor velocity controller system would have a slower response than the response observed. Figure 10 below, shows the observed and simulated step responses for the I motor velocity controller.

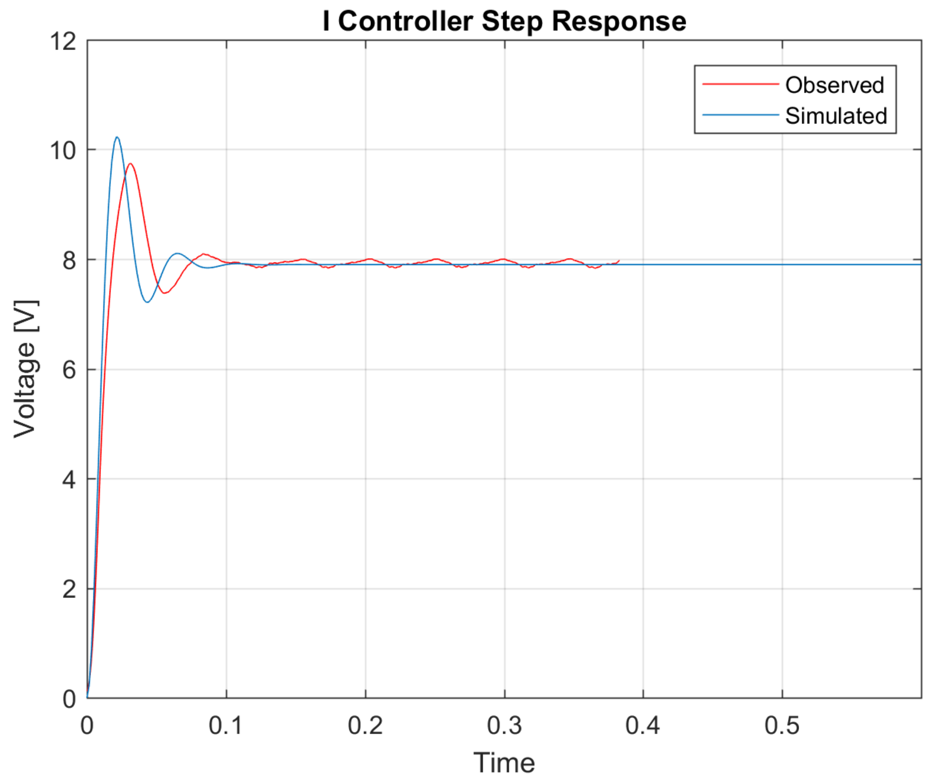

Figure 10 - Step Response of I Motor Velocity Controller

Based on the figure above, the theoretical step response for this controller set up would overshoot the final value more than the observed step response. This would lead to a higher damping ratio along with the same natural frequency and would produce a faster settling time and time constant for the system. The observed response for the I motor velocity controller is slower than the simulated response. This could be due to errors in calculating the open loop motor time constant and gain. The figure below shows the step response for simulated and observed PI controller motor velocity controller.

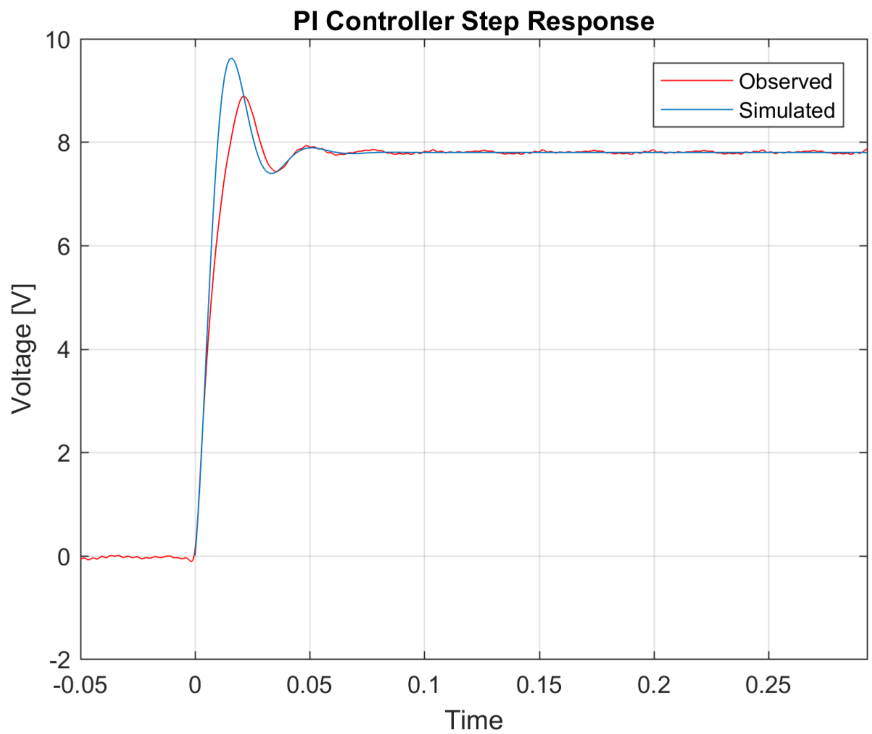

Figure 11 - Step Response of PI Motor Velocity Controller

The findings from this figure are like the findings from Figure 10. Here the simulated step response shows a higher overshoot and a similar natural frequency would lead to a faster time constant or settling time. The observed PI controller step response appears to be slower than the simulated PI controller step response. The speed of response and steady state errors for the observed systems and the open loop motor system can be found in the tables below.

Table 3 - Motor Controller Speed of Response

System	Speed of Response [s]	Settling Time [s]
Open Loop	0.0082	0.033
P Controller	0.0586	0.2346
I Controller	0.0492	0.1967
PI Controller	0.0572	0.2286

Table 4 - Motor Controller Steady State Error

System	Steady State Error [%]
Open Loop	29.49
P Controller	19.14
I Controller	0.58
PI Controller	0.51

From Table 2 the quickest system appears to be the open loop system. The time constant of this system is 0.0082 seconds and the settling time is 0.033 seconds. The next fastest system is the I motor velocity controller, closely behind that is the PI controller and lastly the P controller is the slowest of the three closed loop systems. The full range of time constants and settling times over the three closed loop systems are 0.0094 seconds and 0.0379 seconds respectively. While initially it appears that the open loop system is faster and therefore the best system, the data from Table 3 shows that the open loop system is highly inaccurate with a steady state error of 29.49%. The P controller has the next highest steady state error, but still significantly lower than the open loop system. The I and PI controllers have the lowest values, coming in at under one percent steady state error. However, the PI controller has a lower steady state error than the I controller.

Potential sources of error for the simulated step responses in Figures 9-11 and Tables 2 and 3 could be due to compounding affects from the motor gain and time constant calculations in the transfer function used to simulate the step response. Error affects could also stem from errors in the values for the electrical components such as the resistor and capacitors used in the transfer function to simulate the step response. These values were provided in the lab and not explicitly measured therefore there could be inaccuracies in the values provided.

Conclusion

Lucas

In this lab three closed loop feedback systems were analyzed; a proportional, integral and proportional-integral controller to control the velocity of a DC brush motor as it changes. These closed loop systems were constructed using operational amplifiers, resistors, capacitors and a tachometer to provide a feedback voltage. First another motor was used to back drive the test motor in order to determine the motor constant of the test motor. The motor mechanical properties were determined from the step responses of voltage and load torque on an open loop system. Second, the speed of response and steady state error of the three closed loop systems and the open loop systems were compared to investigate the efficiency of each controller. In conclusion the open loop system yielded the fastest speed of response however it also had the highest steady state error. The P and PI controllers were like, however the I controller had a slightly faster speed of response and the PI controller had a slightly lower steady state error. The I controller should be used in applications where speed of response is desired, and error is not as much of a concern. However, if the application requires higher accuracy and speed of response is not as much of a concern the PI controller should be used.

Zachary

This experiment consisted of measuring the rotational speed of a DC brush motor by utilizing a proportional, integral, and proportional-integral controller. However, an open loop motor was used to compare these three types of controllers so the system parameters could be compared. The time constant of the motor in the open loop system was calculated to compare the controllers to that of a system that did not have a controller. As a result, the open loop system resulted in having the fastest speed of response but ended up having the highest steady state error. If speed of response is desired in a future experiment, then the open loop system will be the recommended test method. However, if the speed of response is not essential, using a controller will better the suit the experiment due the lower steady state errors. Overall, the PI and I controllers tend to have a low steady state error as well as a fast speed of response when the three controllers are compared and should be utilized for future applications similar to this one.

Charlie

A proportional, integral, and proportional-integral controller was used to control a DC brush motor while measuring its rotational speed to seek insight onto the system parameters. Operation amplifiers were implemented into this system to provide the controller, changing the overall circuit slightly to obtain the different control types. An open-loop motor was also inserted into the system for a means of comparing the closed-loop controlled motor being studied. The parameters of both could be calculated and compared. The open-loop motor was calculated to have a faster response rate, meaning a smaller time constant, but had a larger overall steady-state error, which was expected. What this proves to us is that when speed of response is preferred in a design, an open-loop DC brush motor should be used, while if overall steady-state error is preferred to be the smallest, a closed-loop should be utilized. In terms of the effect of the type of controller, the I controller should be used where speed of response is critical to be fast, while the P and the PI controllers offered slower response times, but more accurate steady-state results. The PI seemed to have superior results over the P controller in terms of the speed of response and overall error, and should be used when it is easily implementable.

Reference

Ogata, Katsuhiko. *System Dynamics*. Pearson/Prentice Hall, 2004.

Appendix

Part 2

```
clear all
close all
Input_ei=[1.5 2 3 4 5 6 7 8 9 10]; %Volts
Tach_eo=[.265 .561 1.14 1.72 2.38 2.994 3.579 4.195 4.794 5.405]; %Volts
MUT_em=[.390 .867 1.756 2.661 3.85 4.85 5.806 6.784 7.768 8.751]; %Volts

%a
Ktach=3; %Volts/KRPM

%b
for i=1:length(Tach_eo)
Omega(i)=Tach_eo(i)/Ktach;
end
MUTfit=polyfit(Omega,MUT_em,1);
Ke=MUTfit(1); % Ke motor voltage constant (Volts/KRPM)
Keactual=(4.39+5.37)/2;
figure(1)
bestfit=(MUTfit(1)*Omega)+MUTfit(2);
plot(Omega,MUT_em,'bo-',Omega,bestfit,'r-')
ylabel('MUT Voltage, e_m (Volts)')
xlabel('Angular Velocity, \omega (KRPM)')
title('MUT Voltage vs \omega')
grid on
legend('Data','Best Fit')
%c

Kt=((Ke/1000)/(pi/30))*141.6; %oz-in/A
ktactual=6.6;%oz-in/A
```

Part 3

```
File37 = '3.7.xlsx';
data37 = xlsread(File37); %import data
CH037 = data37(5:100004,1); %Ch0
CH137 = data37(5:100004,2); %Ch1
t37 = data37(5:100004,3)-.2935; %time
File39 = '3.9.xlsx';
data39 = xlsread(File39); %import data
CH039 = data39(5:100004,1); %Ch0
CH139 = data39(5:100004,2); %Ch1
t39 = data39(5:100004,3); %time
%a
% system of equations in document

%b
```

```
ei=6; %Volts
R=3.6; %Ohms
TStall=(Kt*ei)/R; %ozf-in

%c
Ksys=abs(mean(CH037(end-100:end))/mean(CH137(end-100:end))); %K
Kmot=Ksys/Ktach; %KRPM/Volt for c
tau_Voltage=(.632*(8))-4;
for i=1:length(t37)
    if CH037(i)  >= tau_Voltage
        tau632_motor = t37(i); %Tau for part c in seconds
        break

    else
    end
end
figure(2)
plot(t37,CH037,t37,CH137,tau632_motor,tau_Voltage,'d')
axis([-.05 .1 -8 8])
grid on
xlabel('Time, t (seconds)')
ylabel('Voltage (Volts)')
legend('Tachometer Output, e_o','Motor Input, e_a')
title('Tachometer Output and Motor Input With Step Change')

%d
R42=4.2;%Ohms
B=.5*(Kt-(Kmot*Ke*Kt))/(Kmot*R42); %oz-in/KRPM
J=.5*((tau632_motor*((Ke*Kt)+(R42*B)))/R42)*(.001/(pi/30));

%e

t39new=t39(79200:end)-.19;
CH039new=CH039(79200:end);
CH139new=CH139(79200:end);
figure(3)
CH039S=wsmooth(CH039new,t39new,6);%Smoothed
CH139S=wsmooth(CH139new,t39new,6);%Smoothed
meanCH039=mean(CH039S(end-100:end));
AMP=(CH039S(1)-CH039S(end));
plot(t39,CH039,t39,CH139)%-meanCH039)%tau632_motor,tau_Voltage,'d')
%plot(t39,CH039)
%axis([-.05 .1 -8 8])
grid on
xlabel('Time, t (seconds)')
ylabel('Voltage (Volts)')
title('Motor Subjected to 5\Omega Resistor')

% for i = 1:length(t39new)
%
```

```
%     if CH039S(i)<= (.368*AMP)+meanCH039
%         tau39 = t39new(i);
%         break
%
%     else
%     end
% end
tau39=find((abs((.368*AMP)+meanCH039)-CH039S) <= .001);
plot(t39new,CH039S,t39new(tau39(end)),CH039S(tau39(end)),'d')
grid on
xlabel('Time, t (seconds)')
ylabel('Voltage (Volts)')
title('Motor Subjected to 5\Omega Resistor')
tau39=t39new(tau39(end));
Settling_Time=4*tau39;
sserror=(CH039new(1)-meanCH039)/CH039new(1); %(Initial Voltage - Final
Voltage)/(Initial Voltage)
```

Part 4

Reading in Data for c,d,e

```
close all;
P_TachOut = xlsread('P-Control.csv', 'A7:A100007');
P_MotorIn = xlsread('P-Control.csv', 'B7:B100007');
P_Time    = xlsread('P-Control.csv', 'C7:C100007');
figure(1)
plot(P_Time,P_TachOut)
hold on
plot(P_Time,P_MotorIn)
legend('Tach Out','Motor In')

I_TachOut = xlsread('I-Control.csv', 'A7:A100007');
I_MotorIn = xlsread('I-Control.csv', 'B7:B100007');
I_Time    = xlsread('I-Control.csv', 'C7:C100007');
figure(2)
plot(I_Time,I_TachOut)
hold on
plot(I_Time,I_MotorIn)
legend('Tach Out','Motor In')

PI_TachOut = xlsread('PI-Control.csv', 'A7:A100007');
PI_MotorIn = xlsread('PI-Control.csv', 'B7:B100007');
PI_Time    = xlsread('PI-Control.csv', 'C7:C100007');
figure(3)
plot(PI_Time,PI_TachOut)
hold on
plot(PI_Time,PI_MotorIn)
legend('Tach Out','Motor In')
```

a

```
% drawing the block diagrams of each controller attached in  folder
```

b

Root Locus Plots for each controller

```
Ri  = 100000;  %ohm
Rf  = 150000;  %ohm
C1  = .033e-6; %farad
C2  = .010e-6; %farad
K   = 11.9;      %unitless
KI  = 1e-4;
tau = 0.0089;  %sec
K_m = 0.1774;  %krpm/v

P_sys  = tf([K_m*Rf*K], [tau 1]);
I_sys  = tf([K_m*KI], [(tau*C1) C1 0]);
PI_sys = tf([K_m*K*C2*Rf K_m*K], [tau*C2 C2 0]);



% c - finding roots
close all;

P_TachOut = xlsread('P-Control.csv', 'A7:A100007');
% P_MotorIn = xlsread('P-Control.csv', 'B7:B100007');
P_Time    = xlsread('P-Control.csv', 'C7:C100007');

I_TachOut = xlsread('I-Control.csv', 'A7:A100007');
% I_MotorIn = xlsread('I-Control.csv', 'B7:B100007');
I_Time    = xlsread('I-Control.csv', 'C7:C100007');

PI_TachOut = xlsread('PI-Control.csv', 'A7:A100007');
% PI_MotorIn = xlsread('PI-Control.csv', 'B7:B100007');
PI_Time    = xlsread('PI-Control.csv', 'C7:C100007');


[rp,kp] = rlocus(P_sys);
Proot1_p = rp(1);
Proot1_k = kp(1);

[rp,kp] = rlocus(I_sys);
Iroot1_p = rp(1,1);
Iroot1_k = kp(1);
Iroot2_p = rp(2,1);
Iroot2_k = kp(1);

[rp,kp] = rlocus(PI_sys);
```

```
PIroot1_p = rp(1,1);
PIroot1_k = kp(1);
PIroot2_p = rp(2,1);
PIroot2_k = kp(1);

figure(4)
rlocus(P_sys); title('Root Locus of P Motor Controller')
text(Proot1_p + 15, 1, strcat({'Root: '}, num2str(Proot1_p,4), ' s^{-1}'));
figure(5)
rlocus(I_sys); title('Root Locus of I Motor Controller')
text(Iroot1_p-4, 5, strcat({'Root: '}, num2str(Iroot1_p,4), ' s^{-1}'));
text(Iroot2_p, 5, strcat({'Root: '}, num2str(Iroot2_p,4), ' s^{-1}'));
figure(6)
rlocus(PI_sys); title('Root Locus of PI Motor Controller')
text(PIroot1_p, -50, strcat({'Root: '}, num2str(PIroot1_p,4), ' s^{-1}'));
text(PIroot2_p-100, 75, strcat({'Root: '}, num2str(PIroot2_p,4), ' s^{-1}'));

% step responses

figure()
P_TachOut = wsmooth(P_TachOut, P_Time, 3);
plot(P_Time - .052, P_TachOut - P_TachOut(1), 'r')
hold on
step(P_sys* -1e-5, P_Time(end));
xlabel('Time'); ylabel('Voltage [V]');
legend('Observed', 'Simulated')
title('P Controller Step Response'); grid on;
xlim([0 P_Time(end)])

tau_index  = .632*(mean(P_TachOut(end-30000:end)));

for i = 1:length(P_Time)
    if P_Time(i) > .052
        basetime = i;
        break
    end
end

baseline = mean(P_TachOut(1:basetime));
basedev = std(P_TachOut(1:basetime));
threshold = 5*basedev;

for i = 1:length(P_Time)
    if (abs(P_TachOut(i) - baseline) > threshold)
        starttime = i; % start of event
        break
    end
end

P_Time = P_Time(starttime:end);
P_TachOut = P_TachOut(starttime:end);
```

```
% figure()
% plot(P_Time, P_TachOut)

for ii = 1:length(P_TachOut)
   if P_TachOut(ii) <= 0.95*tau_index && P_TachOut(ii) >= 1.05*tau_index
       P_tau = P_Time(ii);
       break
   end
end

% I Controller
I_TachOut = wsmooth(I_TachOut, I_Time, 8);
num = [K_m*4e-5];
den = [Ri*C1*tau C1*Ri 0];
sys = tf(num, den);
sys = feedback(sys, Ri);
[resp, time] = step(sys, I_Time(end));
figure()
plot(I_Time -.217 , I_TachOut - I_TachOut(1), 'r', time, resp*7.9e5)
xlim([0 I_Time(end)])
xlabel('Time'); ylabel('Voltage [V]');
legend('Observed', 'Simulated')
title('I Controller Step Response'); grid on;

I_wn   = abs((5.735 - 4.092)/(0.03009 - 0.2484))*2*pi;
I_fv   = mean(I_TachOut(end-3000:end)) - I_TachOut(1); % I_TachOut(1) is negative
I_pv   = 9.739;
I_OS   = (I_pv - I_fv)/I_fv;
I_zeta = 0.43; % determined from % overshoot

I_SetTime    = 4/(I_zeta*I_wn); % settling time?
I_tau        = I_SetTime/4;

% PI Controller
PI_TachOut = wsmooth(PI_TachOut, I_Time, 8);
PI_TachOut = wsmooth(PI_TachOut, PI_Time, 2);
KPI = 2e-5;
num = [KPI*K_m*Rf*C2 KPI*K_m];
den = [Ri*C2*tau Ri*C2 0];
sys2 = tf(num, den);
sys2 = feedback(sys2, Ri);
[res, t] = step(sys2, PI_Time(end));
figure()
plot(PI_Time -.307 , PI_TachOut - PI_TachOut(1), 'r', t, res*7.8e5)
xlabel('Time'); ylabel('Voltage [V]');
legend('Observed', 'Simulated')
title('PI Controller Step Response'); grid on;
xlim([-.05 PI_Time(end) - .307])

PI_wn   = abs((5.01 - 4.067)/(0.3559 - 0.3277))*2*pi;
PI_fv   = mean(PI_TachOut(end-3000:end));
```

```
PI_pv  = 5.01;
PI_OS  = (PI_pv - PI_fv)/PI_fv;
PI_zeta = 0.37; % determined from % overshoot

PI_SetTime   = 4/(PI_zeta*I_wn); % settling time?
PI_tau       = PI_SetTime/4;
```

d.)

```
% P controller
ptach = xlsread('disturbanceP.csv','A8:A100007');
ptime = 0:2.4e-6:(2.4e-6)*1e5;
ptime = ptime(1:end-1);

ptach = wsmooth(ptach, ptime, 3);
ptachInt = mean(ptach(1:7000));
ptachFin = mean(ptach(end-80000:end));
pSSerror = ((ptachInt - ptachFin)/ptachInt)* 100;
figure()
% plot(ptime, ptach);

% I controller
itach = xlsread('disturbanceI.csv', 'A8:A100007');
itime = 0:6e-6:(6e-6)*1e5;
itime = itime(1:end-1);

itach = wsmooth(itach, itime, 3);
itachInt = mean(itach(1:35000));
itachFin = mean(itach(end-35000:end));
iSSerror = abs((itachInt - itachFin)/itachInt)* 100;
figure()
% plot(itime, itach);

% PI controller
pitach = xlsread('disturbancePI.csv', 'A8:A100007');
pitime = 0:6e-6:(6e-6)*1e5;
pitime = pitime(1:end-1);

pitach = wsmooth(pitach, pitime, 3);
pitachInt = mean(pitach(1:10000));
pitachFin = mean(pitach(end-50000:end));
piSSerror = abs((pitachInt - pitachFin)/itachInt)* 100;
figure()
% plot(pitime, pitach);
```

Peer Effort

By electrically signing this document I agree to the percentages of effort put towards Lab 5: Velocity Control of a DC Brush Motor Under Load.

Zach Shelby – 33%

Lucas Simmonds – 33%

Charlie Nitschelm – 33%

X ZS

Zachary Shelby

X LS

Lucas Simmonds

X CN

Charlie Nitschelm