MEC 411: Lab Number 1

Digital PID Speed Control of a Turntable

Group # 12

Meghan McLaren, Gerard Miles, Mariya Zelinska

October 1, 2019

Table of Contents

Abstract……………………………………………………………………………………...….2

Introduction………………………………………………………………………………….2

Experimental Procedure…………………………………………………………………...7

Results…………………………………………………………………………………....10

Discussion……………………………………………………………………………..….22

Conclusion…………………………………………………………………..…..….23

References………………………………………………………………………………..24

Appendix………………………………………………………………………………....24

Abstract

The objective of this lab is to determine the design input variables for a PID controller that will cause a motor to spin according to the design specifications, as well as to compare the capabilities of a P, PI, and PID controller. A PID controller allows a user control over a feedback system through modifying a factor proportional to the error function, the derivative of the error function, and the integral of the error function. In order to perform this experiment, the control system must be analysed in order to obtain the transfer function of the system, (Eq. 7), the transfer function of the PID, (Eq. 9), and the transfer function of the plant, (Eq. 17). For the P controller, a K_C value of 3.0 was obtained, and the theoretical function matched the experimental function. For the PI controller, the design input values were found to be $K_c = 3$ and $T_i = 0.000457$. The theoretical function also matched the experimental function for this system, and it was more accurate than the P controller. For the PID controller, with design inputs of $K_c = 3$, $T_i = 0.000457$, and $T_d = 3.121 \cdot 10^{-10}$, the system met the design specifications with a percentage overshoot, settling time, and rise time of 2.6166 %, 0.4292335 s, and 0.1552 s respectively. In addition, the theoretical function matched the experimental function.

I. Introduction

A closed control system is a type of system in which an input signal passes through some kind of plant, which will produce an output signal and also provide some form of feedback to the input. Closed control systems can be seen in many aspects of life, such as in a thermostat and the cruise control function of an automobile. Another possible closed control system can be seen in a Proportional Integral Derivative (PID) controller. A PID controller operates based on the principle that there is an input signal which goes through the PID controller, then the plant to become an output signal and provide feedback. The PID controller has user input that modifies a factor proportional to the error function, a user input that modifies the derivative of the error function, and an input to modify the integral of the function. The proportional control deals with what is currently happening, while the integral and derivative controls are dealing with the past and future.

The objective of this lab experiment is to implement a PID turntable speed controller for a motor. This will be done using software with a Data Acquisition (DAQ) interface, and comparing these experimental results to results found via MATLAB software. Design specifications include a percent overshoot, or how much the max amplitude of the output signal overshoots the desired value, of less than 10 %. In addition, another specification is to have a settling time of less than 2 % in 500 ms or less, or in other words have the system settle to the desired value with a 2 % tolerance range. Lastly, the rise time, or time it takes for the motor to first reach the desired value, should be less than 200 ms.

In order for this experiment to be performed, a thorough analysis of the control system being dealt with must be performed. The first step of the analysis is to break the system up into a block diagram. This can be seen in the image below:

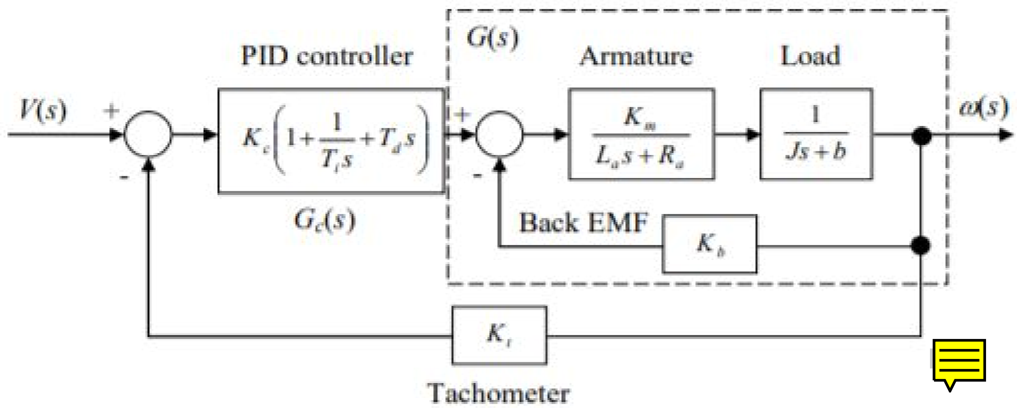

FIG. I. Block diagram of the turntable PID speed control system[1].

From this diagram, it is possible to derive a transfer function, or the ratio of output to input signals, for the system. First it is necessary to derive the transfer signals for the overall system. Note that from point 1 to 5, 1 is at the output, and the points go clockwise until they return to the output, or in other words, point 2 is after the tachometer, point 3 is between the input and the PID controller, point 4 is between the controller and plant, and point 5 is at the output. At point 1 in the above block diagram, the equation is seen below.

$$\omega(s) = \textit{point } 1 \tag{1}$$

Where $\omega(s)$ is equal to the output signal, and s is the laplace variable. From here, the system can be followed to point 2, which goes through the tachometer, where a new equation can be seen.

$$\omega(s)K_t = \textit{point } 2 \tag{2}$$

For which K_t is a constant from the tachometer. Next, point 3 is approached where the equation from point 2 is subtracted from the input function $V(s)$.

$$V(s) - K_t\omega(s) = \textit{point } 3 \tag{3}$$

Following this step, the signal goes through the PID controller, which becomes

$$G_c(s)V(s) - G_c(s)K_t\omega(s) = point\ 4 \tag{4}$$

In which $G_c(s)$ is the function found from the PID controller itself. Lastly, the signal is followed through the plant and returns back to the starting point, which is due to the fact that point 1 and 5 are effectively the same location. Therefore, the equation at point 5 can be written as

$$G(s)G_c(s)V(s) - G(s)G_c(s)K_t\omega(s) = \omega(s) = point\ 5 \tag{5}$$

$G(s)$ is the function from the plant with respect to s. Once this equation is known, it is possible to solve for the transfer function. The basic equation for this is seen below.

$$T = \omega(s)/V(s) \tag{6}$$

Therefore, through factoring out $\omega(s)$ and dividing, the transfer function of the overall system is obtained.

$$T = \omega(s)/V(s) = \frac{G(s)G_c(s)}{1+K_tG(s)G_c(s)} \tag{7}$$

Now that this equation is known, it is necessary to solve for transfer functions of the PID controller and the plant.

The transfer function of the PID controller can be derived from the general equation of a PID controller in the Laplace domain, which is written below.

$$E(s) = k_p + k_I/s + k_Ds \tag{8}$$

In which $E(s)$ is the error function, k_p is the proportional coefficient, k_I is the integral coefficient, and k_D is the differential coefficient. In order to have better control of the motor, it would be preferred to have the derivative and integral coefficients in dimensions of time. This can be done by factoring out the proportional coefficient out of each part of the equation, thus leaving the integral and derivative terms in units of minutes. A new equation, denoted $G_c(s)$ results in the following

$$G_c(s) = K_c(1 + 1/(T_I s) + T_d s) \tag{9}$$

Where K_c is the factored proportional coefficient value, also known as the control coefficient. From this, the equation for the PID controller is known.

As for the plant, the transfer function will need to be derived from the block diagram shown below.

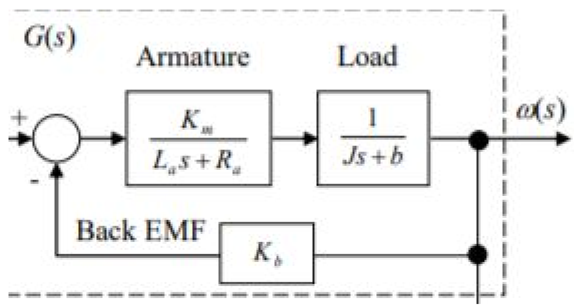

FIG. II. Block diagram of the plant of the speed control system[1].

Using the branch path method again, it is simple to derive the transfer function. The equation at the point 6 starting point is the same as it was for the previous block diagrams point 1. Then point 7 would be right after the back emf, point 8 would be between the input to the load and the armature, denoted R(s), point 9 would be between the armature and the load, and point 10 would be where point 6 is. Point 6 is shown below.

$$\omega(s) = point\ 6 \tag{10}$$

The followed path then goes through the block for back emf until it reaches point 7. The equation for point 7 can then be found.

$$K_b\omega(s) = point\ 7 \tag{11}$$

In which K_b is the back emf constant. The signal is then subtracted from the input, denoted R(s) since it is not the same as the system input at point 8.

$$R(s) - K_b\omega(s) = point\ 8 \tag{12}$$

Next the signal goes to the armature, and thus reaches point 9 in the following way.

$$G_a(s)R(s) - G_a(s)K_b\omega(s) = point\ 9 \tag{13}$$

In which $G_a(s)$ is a function, which is equal to

$$G_a(s) = K_m/(L_a s + R_a) \tag{14}$$

Where K_m, L_a, and R_a are all constants. Finally, the signal goes through the load to reach point 10 as seen below.

$$G_a G_L R(s) - G_a G_L K_b \omega(s) = \omega(s) = point\ 10 \tag{15}$$

$G_L(s)$ is the load function which is equal to the following.

$$G_L = 1/(Js + b) \tag{16}$$

For which J and b are constants. The transfer function can now be solved for by plugging in the values for G_L and G_a into equation 15 and then simplifying and rearranging to get the following transfer function, which will be denoted $G(s)$

$$G(s) = K_m/((L_a + R_a)(Js + b) + K_b K_m) \tag{17}$$

This is the transfer function for the load of the system, which is also the motor itself. Once this is known, all relevant background information for the experiment has been discussed and it is feasible to perform the lab.

II. Procedure

Build the circuit as shown in the diagram below:

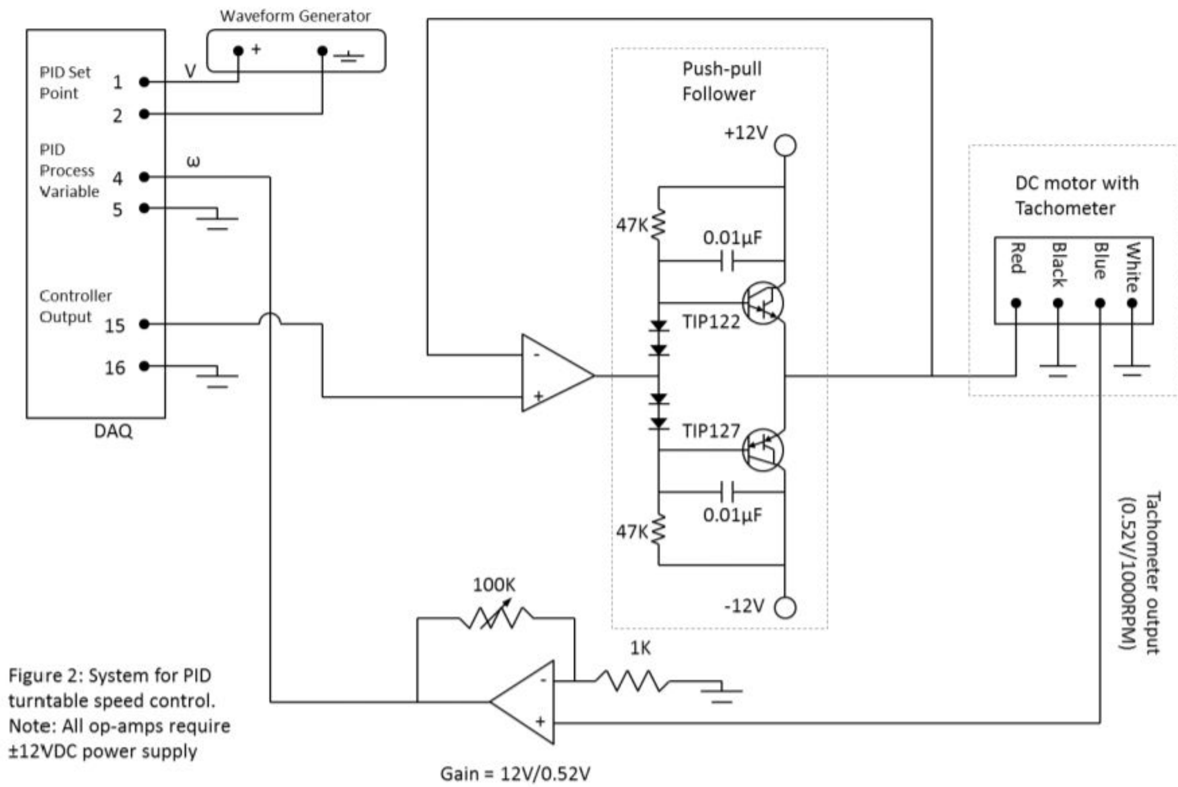

FIG. III. Circuit Diagram of the turntable PID controller[1].

Before proceeding, ensure that all the lab equipment is turned on. Do not connect the circuit to the power supply until the output of both the power supply and the waveform generator are set.

Calculate the resistance value of the potentiometer that will give a gain of 12V/0.52V. Recall that the rules of op-amps are as follows: no current flows through the op-amp, and the potential difference of both the positive and negative sides must be equal[2].

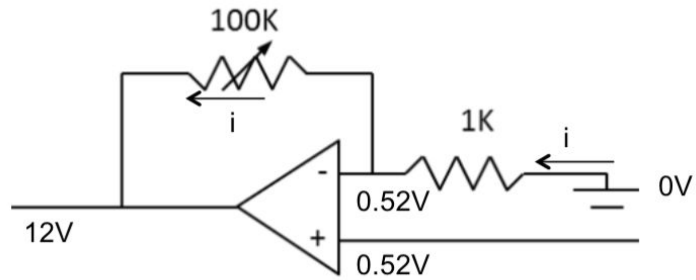

FIG. IV. Circuit Diagram of the operational amplifier with a potentiometer.

Use the fact that the ground has a potential of 0V to calculate the current through the potentiometer.

$$V = iR$$

$$i = \frac{\Delta V}{R} = \frac{0.52V - 0V}{1k\Omega} = 0.52mA$$

Then, use the current and the gain to calculate the resistance of the potentiometer.

$$V = iR$$

$$R = \frac{\Delta V}{i} = \frac{12V - 0.52V}{0.52mA} = 22k\Omega$$

Set the potentiometer to this calculated resistance using a multimeter.

Set the power supply to output 12V and the wave function generator to output a square wave with an amplitude of 1V and a frequency of 1 Hz. Once this is done, make sure the circuit is being powered and open the control software. Set the T_i (integration time) and T_d (derivative time) values to zero and adjust K_C so that the output speed of the motor is as close to the input as possible while maintaining stability. This is the equivalent of a P (proportional) controller. Record the input and output data, as well as the optimal K_C value.

Next, set T_d to zero and adjust K_C and T_i so that the output speed of the motor is as close to the input as possible while maintaining stability. This system is now a PI (potential-integral) controller and should be more accurate than the previous one. Record the input and output data and the optimal K_C and T_i values.

For the final part of the experiment, adjust all three values in order to meet the given specifications for the output of the system. The output function should have a percent overshoot of less than 10 %, a settling time (within 2 % of the final value) of less than 500 ms, and a rise time of less than 200 ms. Record the input and output data and all three design values. This part of the experiment demonstrates the capabilities of a PID (potential-integral-derivative) controller.

III. Results

The signal input and output was measured and collected for the proportional controller, proportional integral controller, and the full PID controller. For the proportional controller, the output that most matched the input signal when the value of K_c was equal to 3.0, with all other inputs equal to zero. This resulted in the plot below, created using Google Sheets.

Proportional Controller:

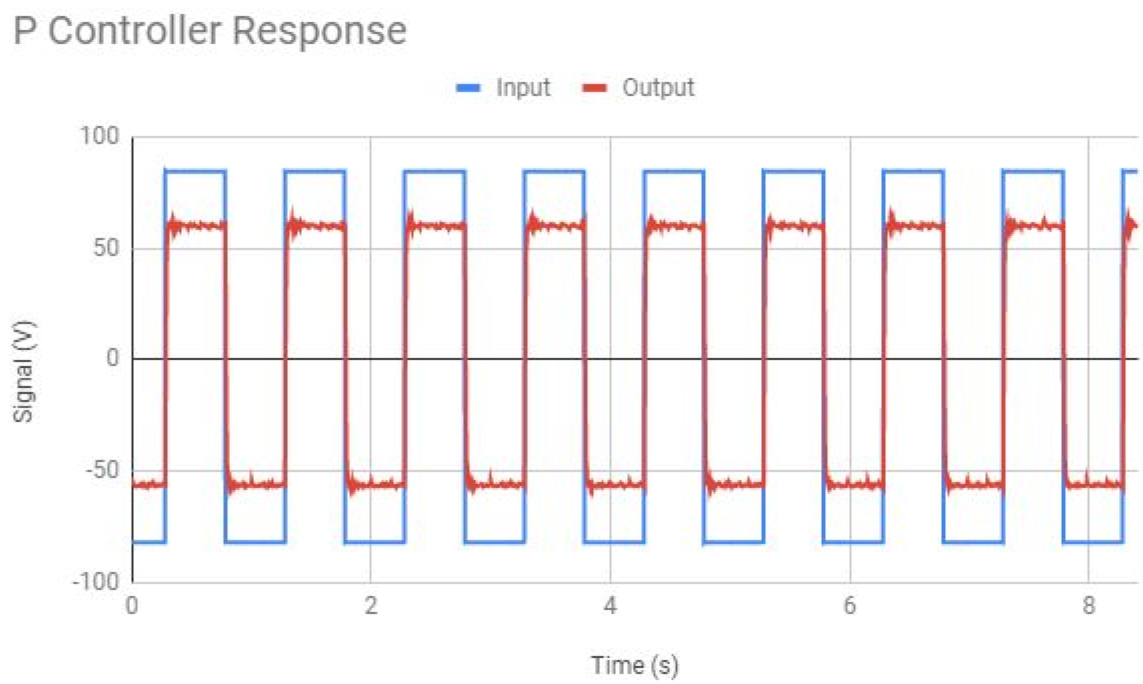

FIG. V. P-controller signal input and output voltage as a function of time.

As seen in the graph above, the output from the P-controller is significantly less than the input function, with the output at an amplitude of 59.97581981V and the input at an amplitude of 84.40572458V (both of these values were obtained by averaging the data points at the top of the first square wave). This was as expected, since the system would become unstable if driven any harder.

In addition to this experimental plot, MATLAB was used to obtain theoretical values of the output signal. The MATLAB plot is reproduced here.

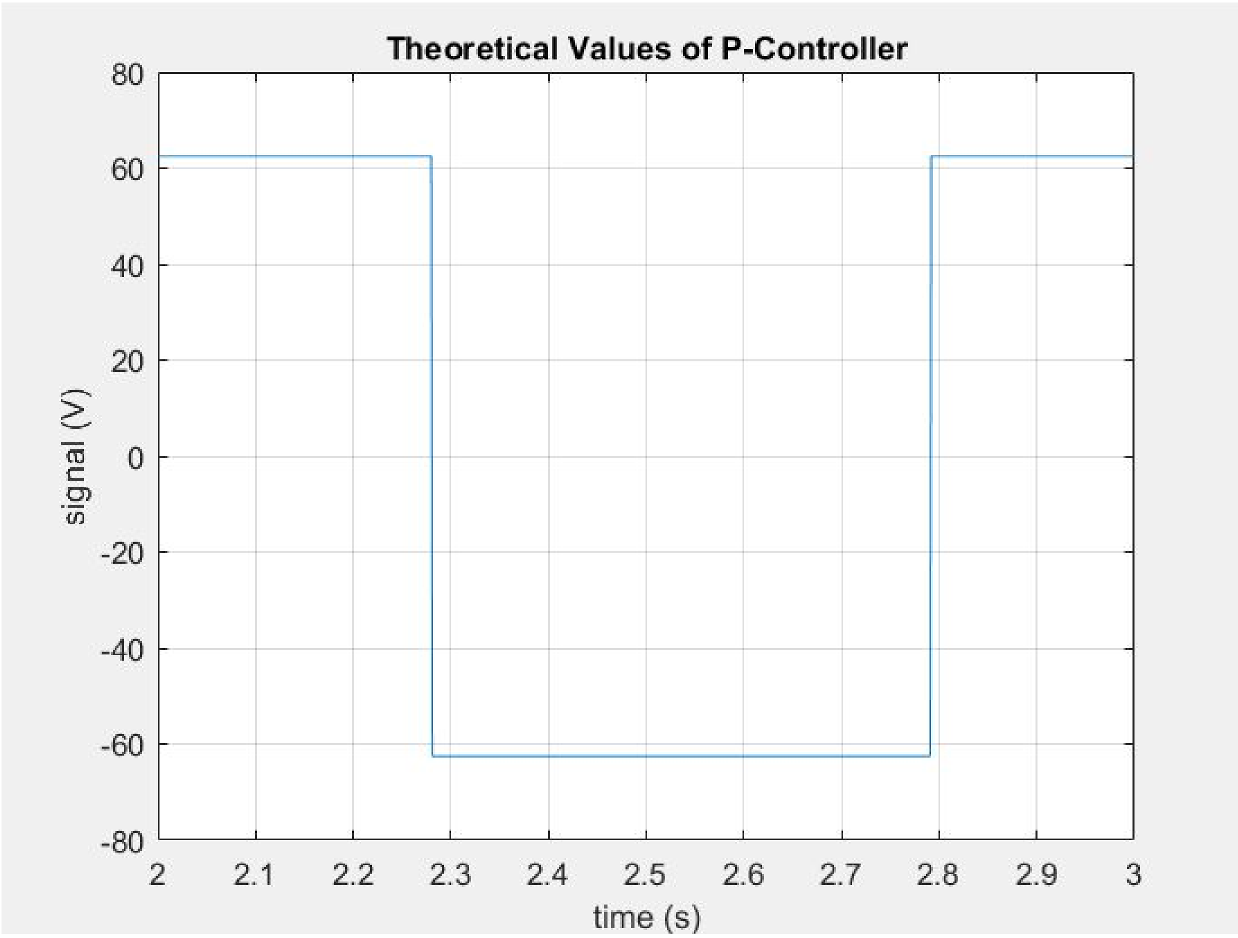

FIG. VI. Theoretical values for the P-controller output voltage as a function of time.

From this graph, it appears that the amplitude, 62.5V, of the theoretical result for the proportional controller is very similar to the result that was achieved in the lab. Therefore, it can be said that the plots agree with one another.

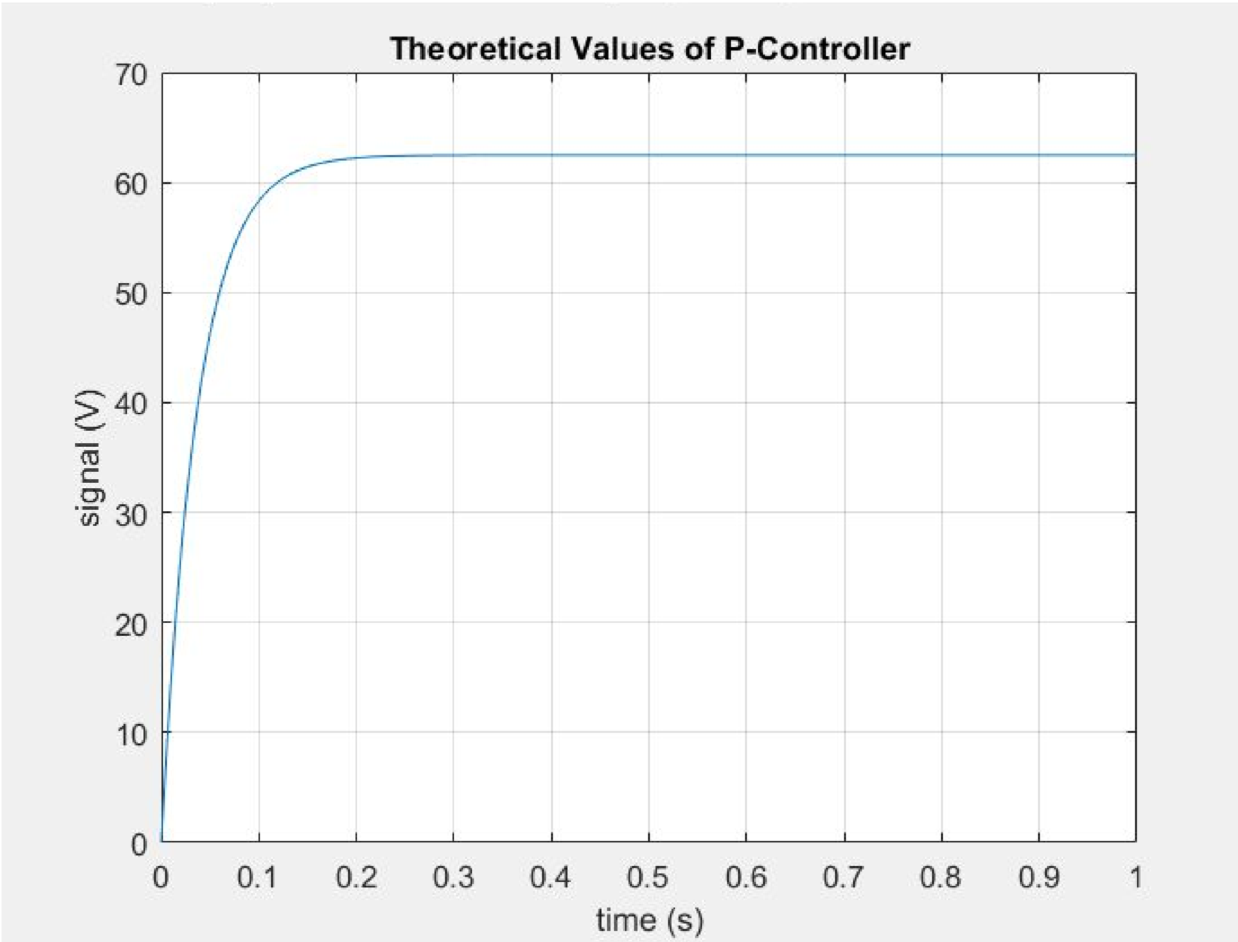

FIG. VII. Theoretical values for the P-controller output voltage as a function of time, starting at t = 0 s.

The plot above illustrates the theoretical output of the system with a P controller from t = 0 s to t = 1s. On this graph, the rise time occurs where the plot first crosses the maximum output of the function, which is 62.5V, and the settling time is where the plot reaches a value within 2 % of 62.5V.

Proportional-Integral Controller:

For the proportional integral controller the input values that had the output most match the input, while remaining stable, were found to be $K_c = 3$ and $T_i = 0.000457$, while $T_d = 0$ due to the nature of a PI controller. The results obtained were then recorded and made into the Google Sheets plot seen below.

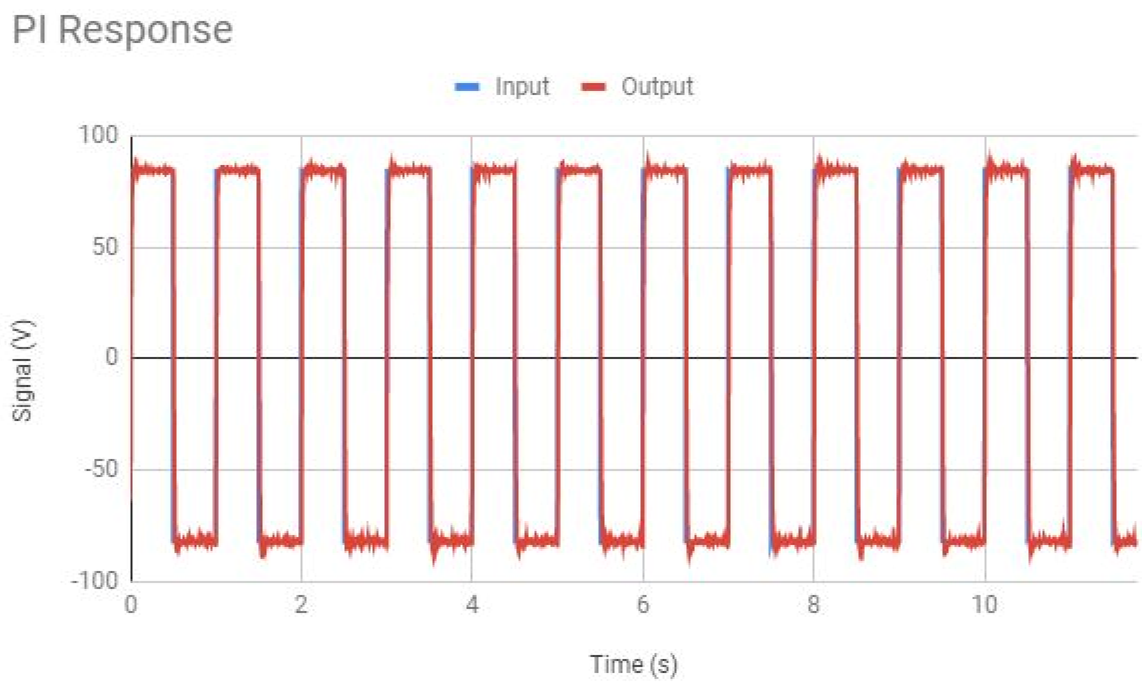

FIG. VIII. PI-controller signal input and output voltage as a function of time.

As seen in the above plot, the output function very nearly matches the input function. The output has an amplitude of 84.34294675V and the input has an amplitude of 84.38907143V (both of these values were obtained the same way as above). This controller was much more accurate than the P controller. This was as expected, since a PI controller gives more variables to adjust than a P controller.

Additionally, theoretical values for the output signal were found using MATLAB, as seen in the following plot.

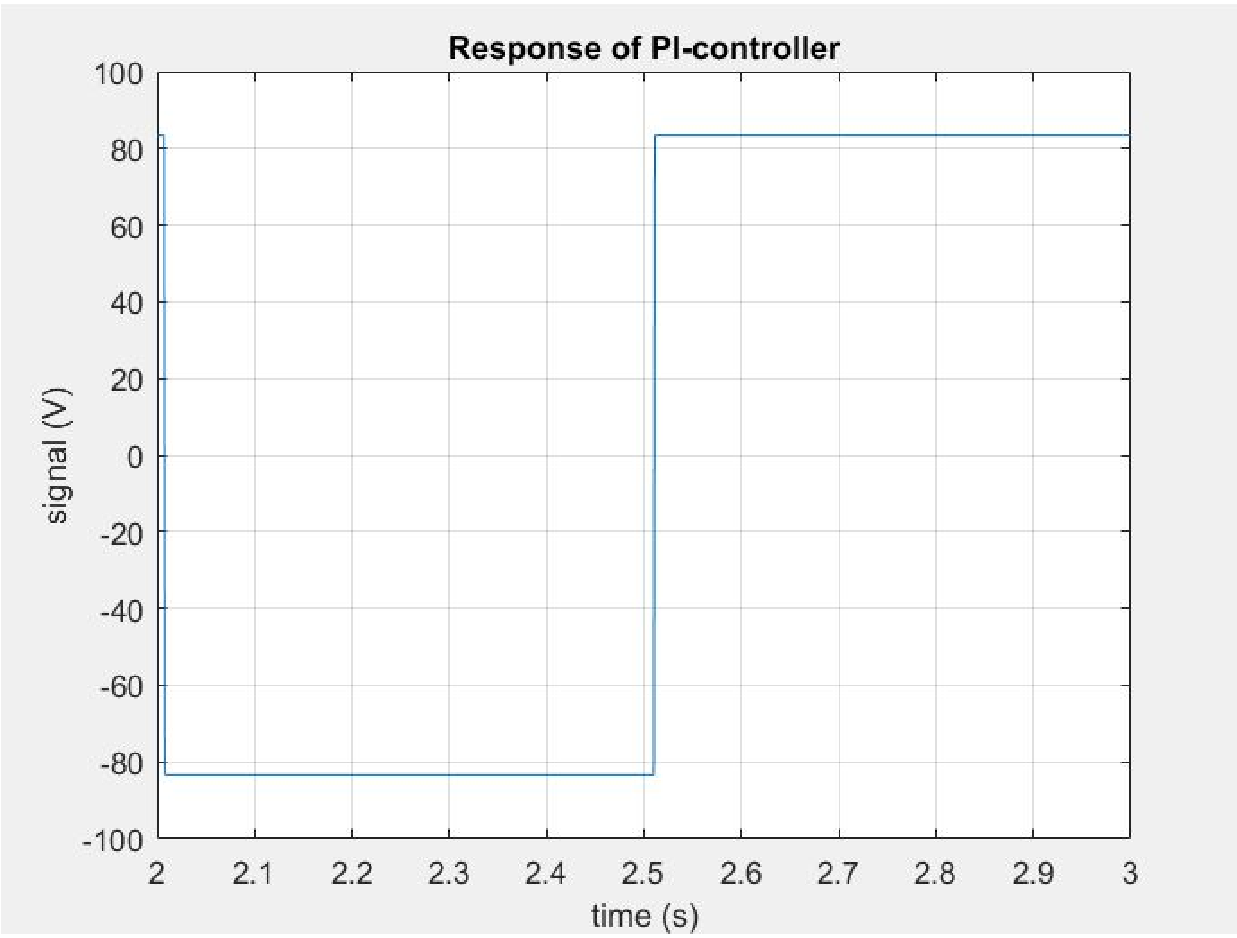

FIG. IX. Theoretical values for the PI-controller output voltage as a function of time,.

As can be seen in the graph above, the theoretical output of the system with a PI controller has relatively the same amplitude as the experimental output, where the experimental output has an amplitude of 84.34294675V and the theoretical output has an amplitude of 83.333V. It can therefore be concluded that the experimental results are reasonable.

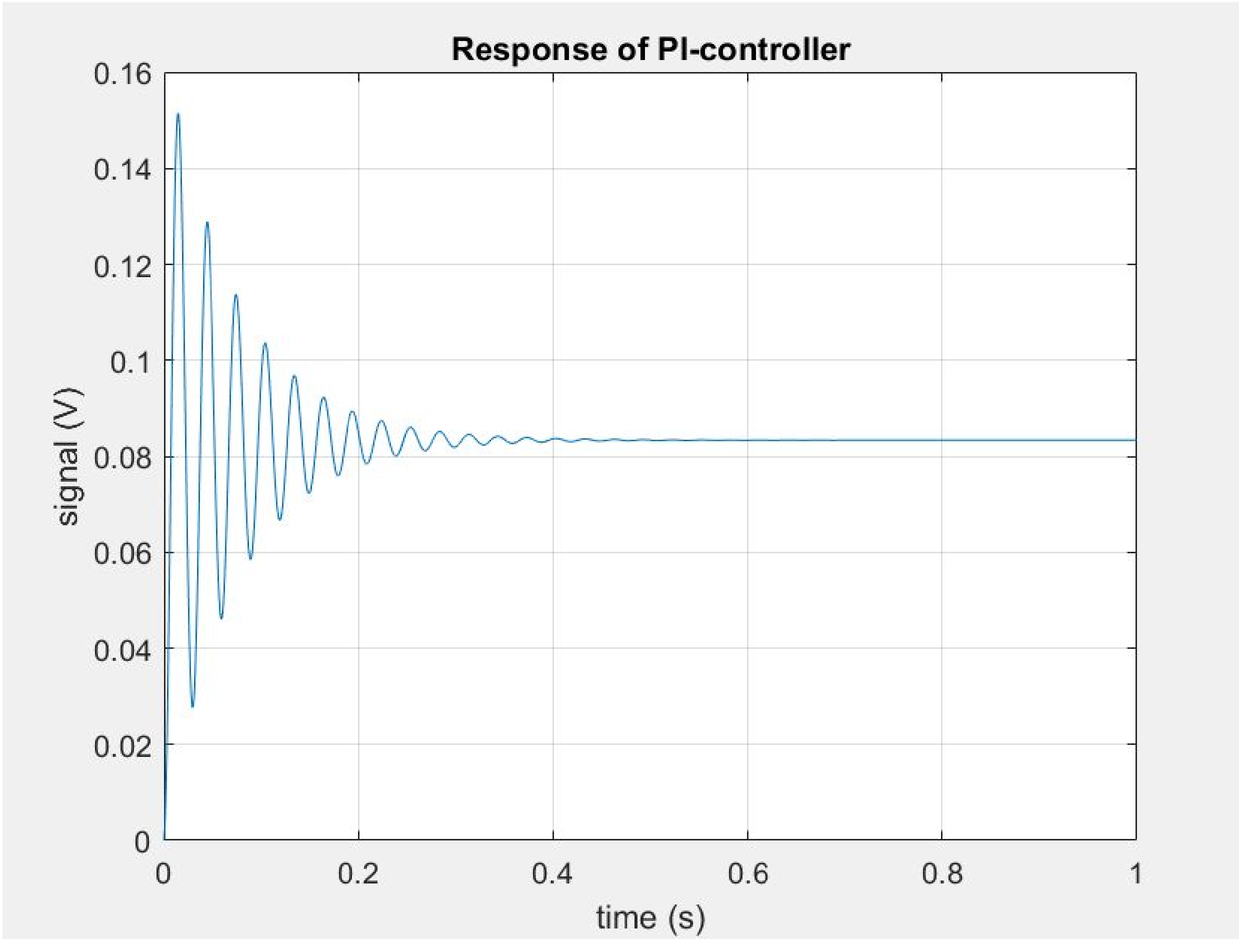

FIG. X. Theoretical values for the PI-controller output voltage as a function of time, starting at t = 0 s.

The above graph shows the theoretical output of the PI controller from t = 0s to t = 1s. The rise time is where the plot first crosses 83.333V, and the settling time is where it starts oscillating within 2 % of 83.333V.

Proportional-Integral-Derivative Controller:

The PID controller, as previously stated, has 3 specific design requirements that were required to be met. The variable inputs that best met a percentage overshoot of less than 10 %, a settling time of less than 2 % to be achieved in 500 ms or less, and lastly a rise time at a maximum of 200 ms were found to be $K_c = 3$, $T_i = 0.000457$, and $T_d = 3.121 \cdot 10^{-10}$. Just like the previous controllers, the measurements made to meet the specifications were plotted against time to more easily display the general trends of the data.

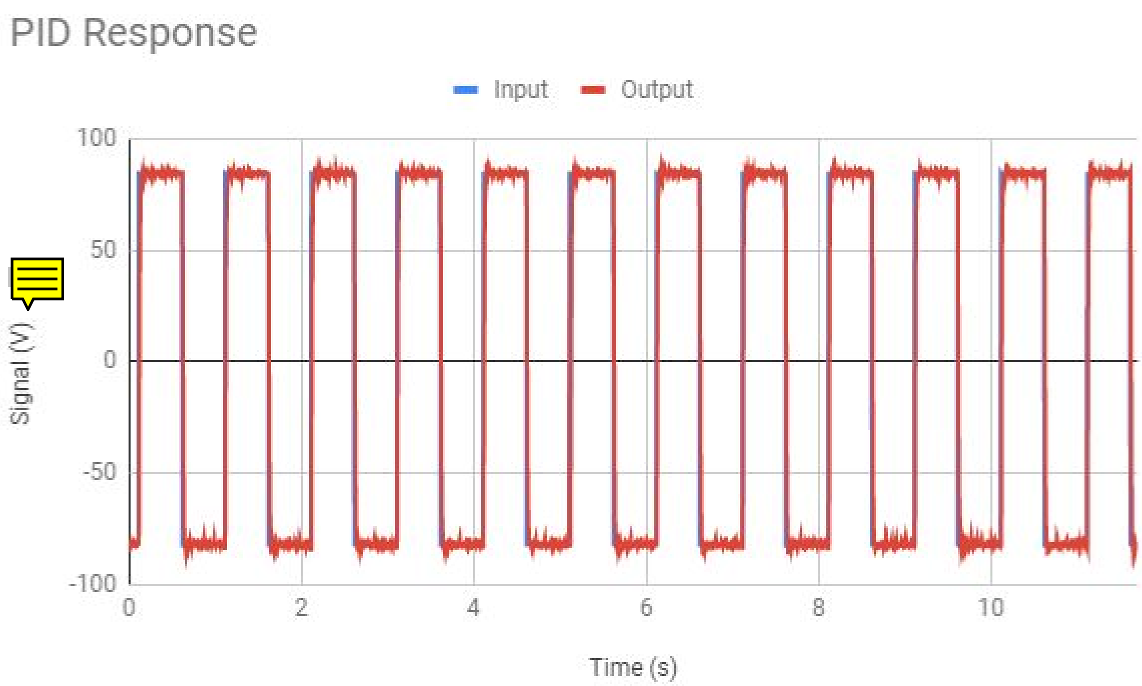

FIG. XI. PID-controller signal input and output voltage as a function of time.

From the figure above, percentage overshoot can be calculated by identifying the maximum amplitude of the output of one of the square waves and comparing it to the input value at the top of the square wave. Looking at the first square wave in the plot, the maximum amplitude was found to be 86.601611V, and the values for the input function at the top of the square wave were averaged together to get 84.39338277V. The percentage overshoot can then be calculated using this formula:

$$PO = \frac{86.601611V - 84.39338277V}{84.39338277V} \times 100$$

$$PO = 2.6166\%$$

Where *PO* is the percentage overshoot.

The settling time can be determined by finding where the function oscillates within 2 % of the input value, in this case between 82.70551512V and 86.08125043V. Based on the data, this happens between t = 0.427732 s and t = 0.430735 s. These two times can be averaged together to get the settling time.

$$t_s = \frac{0.427732\ s + 0.430735\ s}{2}$$

$$t_s = 0.4292335\ s$$

Where t_s is the settling time.

The rise time can be calculated by determining at what time the function first crosses the input value, which is 84.39338277V. Based on the data, this happens between t = 0.154907 s and t = 0.157905 s. Linear interpolation can be used to get the exact time the function reaches the input value.

$$t_r = (84.39338277V - 84.326705V)\left(\frac{0.157905\ s - 0.154907\ s}{84.957101V - 84.326705V}\right) + 0.154907\ s$$

$$t_r = 0.1552\ s$$

Where t_r is the rise time. Based on the values calculated for percentage overshoot, settling time, and rise time, this system meets all the design specifications.

Once again, MATLAB plots were also produced to find theoretical values of the experimental results, such that they may be analyzed and compared.

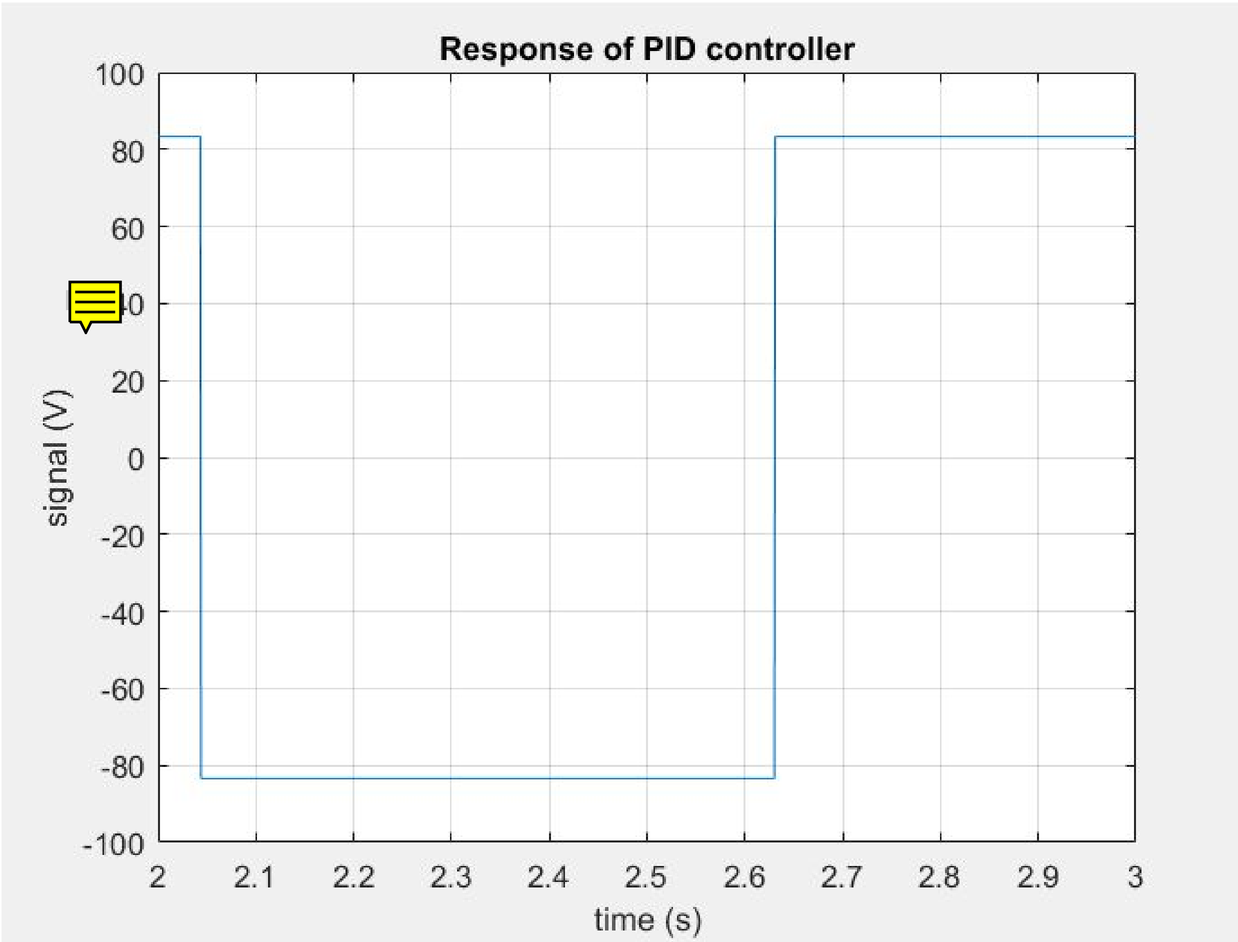

FIG. XII. Theoretical values for the PID-controller output voltage as a function of time.

The plot of the theoretical response of the PID controller above, with an amplitude of 83.333V, very nearly matches the results obtained from the lab, with an amplitude of 84.32281333V. Therefore, the experimental results can be said to be reasonable.

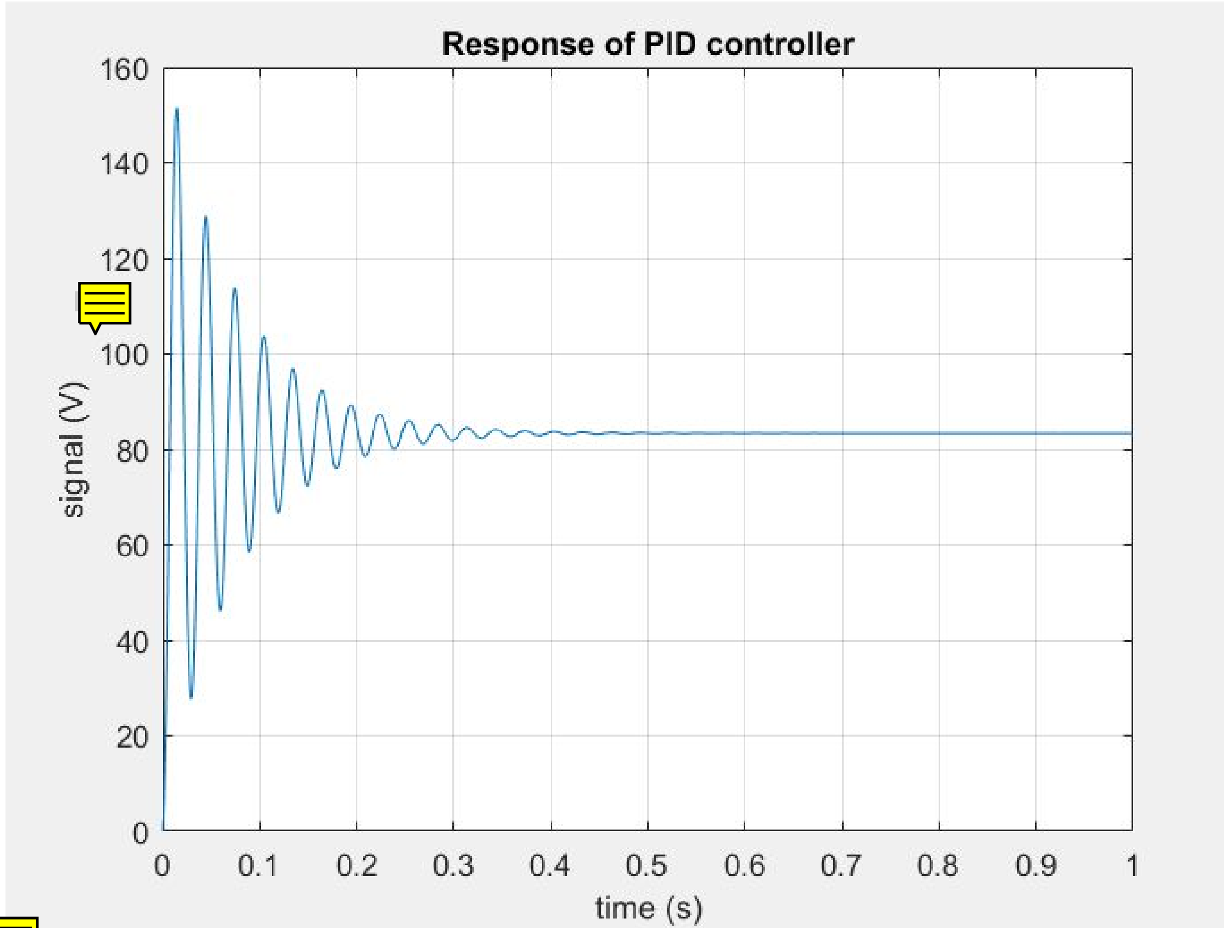

FIG. XIII. Theoretical values for the PID-controller output voltage as a function of time, starting at t = 0 s.

The figure above shows the theoretical output of the system with a PID controller from t = 0 s to t = 1 s. The rise time occurs where the plot crosses 83.333V (from the value of the w vector at t = 1 s), and the settling time occurs within 2 % of the final value, or near where the plot stops oscillating.

Error Analysis:

To obtain an uncertainty in the output voltage, the uncertainty of the gain of the non inverting operational amplifier was calculated. To do so, the instrument uncertainty in the resistance values was determined using the tolerances of the 1K Ω resistor and the potentiometer. The error in the gain was determined as shown below. Additionally, the uncertainty in the DAQ was considered, but since the resolution uncertainty does not change while the voltage is amplified, it is relatively negligible in this case.

- Instrument Uncertainty in 1KΩ Resistor:

  $u_{1K\Omega} = 0.05(1000\ \Omega) = 50\ \Omega$

- Instrument Uncertainty in Potentiometer:

  $u_{potentiometer} = 0.05(22000\ \Omega) = 1100\ \Omega$

- Resolution uncertainty in the DAQ:

  $u_{resolution\ DAQ} = Voltage\ Range/2^{(bit\ precision)} = 24V/(2^{12}) = 5.859\ mV$

- Uncertainty in Gain:

  $\frac{V_{out}}{V_{in}} = 1 + \frac{R_{potentiometer}}{R_{1K}}$

  $\Delta\frac{V_{out}}{V_{in}} = (\frac{V_{out}}{V_{in}})\sqrt{(\Delta R_{potentiometer}/R_{potentiometer})^2 + (\Delta R_{1K}/R_{1K})}$

  $\Delta\frac{V_{out}}{V_{in}} = (\frac{12\ V}{0.52\ V})\sqrt{(1100\ \Omega/22000\ \Omega)^2 + (50\ \Omega/1000\ \Omega)^2} = 1.632\ V$

The following values are the uncertainties in the parameters of the DC motor, tachometer, and load that affect the total uncertainty of the calculated theoretical output voltage (using the inverse Laplace transform of ω(s) = T(s)V(s)) of the PID controller[1]. However, they were not accounted for in the error analysis since error propagation cannot easily be performed on ω(t).

- $u_{K_b} = 0.10(12\ V/KRPM) = 1.2\ V/KRPM$
- $u_{K_m} = 0.10(16.2\ OZ\ IN/A) = 1.62\ OZ\ IN/A$
- $u_{R_a} = 0.15(11.5\ \Omega) = 1.725\ \Omega$

IV. Discussion

For the proportional controller, with a K_C value of 3.0, the output function had a smaller amplitude than the input function, with values of 59.97581981V and 84.40572458V, respectively. This was as expected, however, since any higher of a K_C value would have made the system unstable. In addition, the results obtained in the lab matched the theoretical plot obtained from MATLAB. The experimental plot had an output amplitude of 59.97581981V, while the theoretical plot had an amplitude of 62.5V, which are both reasonably close to one another. Therefore, the experimental results can be said to be reasonable.

For the proportional-integral controller, the output function was very close to the input function with design input values of $K_c = 3$ and $T_i = 0.000457$. Based on the data, the output function had an amplitude of 84.34294675V and the input function had an amplitude of 84.38907143V. This controller was, as expected, more accurate than the P controller. When the output function was compared to the theoretical function, they were both very close with amplitudes of 84.34294675V and 83.333V, respectively. Thus it can be concluded that the results obtained in the lab are reasonable.

The proportional-integral-derivative controller was given a set of design specifications that had to be met. The system had to have a percentage overshoot of less than 10%, a settling time of less than 2% within a minimum of 500ms, and a rise time of 200ms, maximum. With design inputs of $K_c = 3$, $T_i = 0.000457$, and $T_d = 3.121 \cdot 10^{-10}$, the percentage overshoot was calculated to be 2.6166%, the settling time was calculated to be 0.4292335s, and the rise time was calculated to be 0.1552s. All of these values meet the requirements stated above.

When compared with theoretical results, the output function obtained in the lab, with an amplitude of 84.32281333V, was very close to the theoretical function, with an amplitude of 83.333V. Therefore, it can be concluded that the experimental results are reasonable.

Any variance in this experiment would most likely result from loss within the system. Thus, the uncertainty in the 1KΩ resistor, potentiometer, and DAQ had to be accounted for. Based on this, it was calculated that there was a 1.632V uncertainty in the gain. Some of the error could have also resulted from resistance in the wires, friction in the motor, or static electricity from anything that could come into contact with the electrical components.

V. Conclusion

- The experimental results showed that the PID speed control system matched the required design specifications when it was subjected to a unit step input and that the experimental response of the P, PI, and PID controller all relatively matched the theoretical responses for the output signals.
- The relevant values were found to be $K_C = 3$, $T_i = 0.000457$, $T_d = 3.12 \cdot 10^{-10}$ for the PID controller. The percent overshoot was $PO = 2.6166\%$, which is under 10 %, the settling time was $t_s = 0.4292\ s$, which is less than 500 ms, and the rise time was $t_r = 0.1552\ s$, which is less than 200 ms, therefore all values matched the design specifications.

- The experimental results all relatively agreed with the theoretical results according to the plots of the output voltage obtained using MATLAB.
- Any discrepancies can be attributed to uncertainties associated with the voltage gain due to factors such as the resistance in the electrical components, the DAQ, and any other losses within the system.

References

[1] Machtay, Noah, *PID Speed Control of a Turntable - MEC 411 Lab 1*, Stony Brook University (2019).

[2] Machtay, Noah. *MEC 220.01 Practical Electronics Mechanical Engineers,* Stony Brook University, Stony Brook, NY. Lecture (2017).

[3] Franklin, Gene F., David J. Powell and Abbas Emami-Naeini, *Feedback Control of Dynamic Systems,* Upper Saddle River: Pearson Higher Education (2010).

Appendix

MATLAB

```
close all
clc
clear

%general constants
Km=16.2; %OZ-IN/A
R=11.5; %ohm
La=0; %constant
```

```
J=2.5; %OZ-IN^2
b=0; %constant
Kb=12; %V/KRPM
Kt=12; %V/KRPM

%laplace transform of G (note that Gc varies based on setup so it will be
%found later)
syms s %symbolic toolbox
G=Km/(R*(J*s+b)+Kb*Km);
iG=ilaplace(G); %inverse laplace transform

%proportional controller
%knob values
Kc=3; %value of Kc constant
Ti=0; %integral time MINUTES
Td=0; %derivative time MINUTES

%laplace stuff, note that V=1/s (input)
Gcp=Kc; %Kc is the one non zero value for the P controller
t=0:.001:1; %time range of 1 second (1 period)
[r,c]=size(t); %size of time array
T=(Gcp*G)/(1+(Kt*Gcp*G)); %transfer function of the system
T_V=(1/s)*T;
TV=ilaplace(T_V) %no semicolon on purpose, equation of output is equal to transfer function times input (1/s)

%loop to solve for P output
w=zeros(r,c); %zero vector of output
for i=1:c
```

```
    %conditional statement is for what he said in class today, in which we
    %have a piecewise function to represent the square wave from the wave
    %generator. I am not sure how to make this start at the desired values
    %if anyone recorded anything that could help.
    if t(i)>2.280828 && t(i)<2.791512 %below equations copy pasted from symbolic toolbox
        w(i)=-1000*(1/16 - exp(-(15552*t(i))/575)/16); %positive output from the square wave
    else
        w(i)=1000*(1/16 - exp(-(15552*t(i))/575)/16); %negative output from the square wave
    end
end

figure(1)
plot(t,w)
grid on
xlabel('time (s)')
ylabel('signal (V)')
title('Theoretical Values of P-Controller')

%proportional-integral controller
%knob values
Kc=3; %value of Kc constant
Ti=0.000457; %integral time MINUTES
Td=0; %derivative time MINUTES

%laplace stuff, note that V=1/s (input)
Gcpi=Kc*(1+1/(Ti*s)); %Kc is the one non zero value for the P controller
t=0:.001:1; %time range of 1 second (1 period)
[r,c]=size(t); %size of time array
T=(Gcpi*G)/(1+(Kt*Gcpi*G)); %transfer function of the system
```

```
T_V=(1/s)*T;
TV=ilaplace(T_V) %no semicolon on purpose, equation of output is equal to transfer function times input (1/s)

%loop to solve for P output
w=zeros(r,c); %zero vector of output
for i=1:c

    %conditional statement is for what he said in class today, in which we
    %have a piecewise function to represent the square wave from the wave
    %generator. I am not sure how to make this start at the desired values
    %if anyone recorded anything that could help.
    if t(i)>2.007009 && t(i)<2.510694 %below equations copy pasted from symbolic toolbox
        w(i)=-1000*(1/12 -
(exp(-(7776*t(i))/575)*(cos((864*1686032408337053^(1/2)*33009924632371549267^(1/2)*t(i))/969468634793805475
) -
(9*1686032408337053^(1/2)*33009924632371549267^(1/2)*sin((864*1686032408337053^(1/2)*3300992463237154
9267^(1/2)*t(i))/969468634793805475))/66019849264743098534))/12);

    else
        w(i)=1000*(1/12 -
(exp(-(7776*t(i))/575)*(cos((864*1686032408337053^(1/2)*33009924632371549267^(1/2)*t(i))/969468634793805475
) -
(9*1686032408337053^(1/2)*33009924632371549267^(1/2)*sin((864*1686032408337053^(1/2)*3300992463237154
9267^(1/2)*t(i))/969468634793805475))/66019849264743098534))/12);

    end
end

figure(2)
```

```
plot(t,w)
grid on
xlabel('time (s)')
ylabel('signal (V)')
title('Response of PI-controller')

%proportional-integral-derivative controller
%knob values
Kc=3; %value of Kc constant
Ti=0.000457; %integral time MINUTES
Td=3.12*10^-10; %derivative time MINUTES

%laplace stuff, note that V=1/s (input)
Gcpid=Kc*(1+1/(Ti*s)+Td*s); %Kc is the one non zero value for the P controller
t=0:.001:1; %time range of 1 second (1 period)
[r,c]=size(t); %size of time array
T=(Gcpid*G)/(1+(Kt*Gcpid*G)); %transfer function of the system
T_V=(1/s)*T;
TV=ilaplace(T_V) %no semicolon on purpose, equation of output is equal to transfer function times input (1/s)

%loop to solve for P output
w=zeros(r,c); %zero vector of output
for i=1:c

    %conditional statement is for what he said in class today, in which we
    %have a piecewise function to represent the square wave from the wave
    %generator. I am not sure how to make this start at the desired values
    %if anyone recorded anything that could help.
    if t(i)>2.043972 && t(i)<2.63061 %below equations copy pasted from symbolic toolbox
```

```
      w(i)=-1000*(1/12 -
(173783086569602943864012800*exp(-(9400607173323356462515224576*t(i))/695132350677895932571382141)*
(cos((510015580149921683079168*692268299025756749999206781^(1/2)*407353004320243646190166528766 2
060712691613792835937419723424921613465^(1/2)*t(i))/407353004320243646190166528766206071269161379 2
835937419723424921613465) -
(625619107691034856506648233 1*692268299025756749999206781^(1/2)*407353004320243646190166528766 2
060712691613792835937419723424921613465^(1/2)*sin((510015580149921683079168*69226829902575674999 9
206781^(1/2)*407353004320243646190166528766206071269161379283593741972342492161346 5^(1/2)*t(i))/407
3530043202436461901665287662060712691613792835937419723424921613465))/32667017575625226597194 0
4870357144008917625405196594707326841118082620981248 00))/208539705203368779771414642 3);
   else
      w(i)=1000*(1/12 -
(173783086569602943864012800*exp(-(9400607173323356462515224576*t(i))/695132350677895932571382141)*
(cos((510015580149921683079168*692268299025756749999206781^(1/2)*407353004320243646190166528766 2
060712691613792835937419723424921613465^(1/2)*t(i))/407353004320243646190166528766206071269161379 2
835937419723424921613465) -
(625619107691034856506648233 1*692268299025756749999206781^(1/2)*407353004320243646190166528766 2
060712691613792835937419723424921613465^(1/2)*sin((510015580149921683079168*69226829902575674999 9
206781^(1/2)*407353004320243646190166528766206071269161379283593741972342492161346 5^(1/2)*t(i))/407
3530043202436461901665287662060712691613792835937419723424921613465))/32667017575625226597194 0
4870357144008917625405196594707326841118082620981248 00))/208539705203368779771414642 3);

   end
end

figure(3)
plot(t,w)
grid on
xlabel('time (s)')
```

```
ylabel('signal (V)')

title('Response of PID controller')
```

```
TV =

1/16 - exp(-(15552*t)/575)/16

TV =

1/12 -
(exp(-(7776*t)/575)*(cos((864*1686032408337053^(1/2)*33009924632371549267^(1/2)*t)/969468634793805475) -
(9*1686032408337053^(1/2)*33009924632371549267^(1/2)*sin((864*1686032408337053^(1/2)*3300992463237154
9267^(1/2)*t)/969468634793805475))/66019849264743098534))/12

TV =

1/12 -
(173783086569602943864012800*exp(-(9400607173323356462515224576*t)/695132350677895932571382141)*(c
os((510015580149921683079168*692268299025756749999206781^(1/2)*407353004320243646190166528766206
0712691613792835937419723424921613465^(1/2)*t)/4073530043202436461901665287662060712691613792835
937419723424921613465) -
(6256191076910348565066482331*692268299025756749999206781^(1/2)*4073530043202436461901665287662
060712691613792835937419723424921613465^(1/2)*sin((510015580149921683079168*692268299025756749999
206781^(1/2)*4073530043202436461901665287662060712691613792835937419723424921613465^(1/2)*t)/40735
30043202436461901665287662060712691613792835937419723424921613465))/32667017575625226597194048
7035714400891762540519659470732684111808262098124800))/2085397052033687797714146423
```

Published with MATLAB® R2016b

Comment Summary

Page 3

1. integral and drivative with respect to what?

Page 4

2. Using software you just acquire data.

Page 5

3. You need to explicitly show or describe where points 1 to 9 are located.

Page 8

4. Derive the transfer function of the system in expanded form.

Page 9

5. You should have a list of the equipments used and relate them in experimental procedure.

Page 12

6. Recorded output is RPM of the motor not Volt.

Page 16

7. Theoretical result is not correct.

Page 18

8. No unit.

Page 20

9. No unit in Y axis.

Page 21

10. Fig XII and Fig XIII have same caption. I did not understand their use.
11. No unit in Y axis. So I did not understand.
12. Please also provide the experimental result in this figure for ease of comparison.
    i. This is response for unit step which is only part of the waveform in our lab. Please provide response for a complete waveform next time.

Page 24

13. Please provide mathematical justification.