MEC 411: Lab Number 3

Turntable Speed Control System Design

Group # 12

Meghan McLaren, Gerard Miles, Mariya Zelinska

November 19th, 2019

Table of Contents

Abstract

The objective of this lab is to design a PID controller for a turntable driven by a motor according to the provided design specifications, and to compare experimentally obtained results of the system to theoretical results found via MATLAB software. In addition, the designed PID controller must be compared to the previous labs' controllers, as well as have a justifiable cost analysis if it were to be implemented. A PID controller has a parameter that is proportional to its error function, a term that is related to the integral of the error function, and a term that is related to the derivative of the error function, all with respect to time. The transfer function of the system must also be known (Eq. 21). From the experiment, it was found that $K_c = 1.8,\ T_i = 0.00075s,$ and $T_d = 3.4 * 10^{-10}s$. This resulted in PID parameter coefficients of $K_p = 1.8,\ K_i = 2400,$ and $K_d = 6.12 * 10^{-10}$. The experimental results matched the general form of the theoretical results, and any differences between the two can be attributed to noise, disturbances, and the limited number of data points. The percent overshoot was found to be 34.2%, which is greater than the 10% design specification. However if noise is neglected, the corrected overshoot is approximately 8%, which is within the criteria. The rise time and 2% settling time are 168.3 ms and 462.7 ms respectively, both falling within the specifications. This experiment also verified that the PID controller from Lab 1 is the most effective model relative to the specifications, due to it being the only design that met all design specifications. Additionally, implementing the designed Lab 3 PID controller in industry wouldn't be worth the estimated cost of approximately $1,529,550.

I. Introduction

A control system can be considered closed when if and only if the system's input signal passes through some kind of plant, which will then produce an output signal and most importantly provide some form of feedback to the input so that the system can dynamically adapt. The feedback is what directly results in the system being considered closed. Closed control systems can be used in a multitude of everyday applications. Examples can include a thermostat as well as the cruise control function of an automobile. An example of a very important closed control system is the classic Proportional Integral Derivative (PID) controller. A PID controller operates based on the idea that there is an input signal which goes through the PID controller itself, then the plant, and thus becomes an output signal and provides feedback through some form of device such as a tachometer. The PID controller has user input that modifies multiple variables. There is a variable that is proportional to the error function with respect to time, a user input variable that modifies the derivative of the error function with respect to time, and an input variable to modify the integral of the function with respect to time. Essentially, in a PID controller, the proportional control deals with what is currently happening, while the integral and derivative controls are dealing with the past and future responses respectively.

An important thing to take note of when designing a PID controller is the difference between a digital PID controller and an analog PID controller. A digital PID controller typically uses software to control the system, such as LabVIEW, and an analog PID controller uses physical means to control the system, such as potentiometers in an electrical circuit. Generally speaking, a digital PID controller is more accurate than an analog controller because the computer tends to tune out noise and disturbances more efficiently then the circuit. However, in doing this the digital PID controller tends to be slower than it's analog counterparts because of the time it takes to process the data. These differences will be explored in the discussion section of this lab report.

The objective of this lab experiment is to create a PID turntable speed controller system for a turntable driven by a motor using a customized LabVIEW program with a Data Acquisition (DAQ) interface. The experimentally obtained results are then compared to results found via MATLAB software to show how successful the design is based on the specified requirements outlined in the lab manual. The provided design specifications for the controller include as follows: a percent overshoot, or how much the max amplitude of the output signal overshoots the desired value, of less than 10%, 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, and the rise time, or time it takes for the motor to first reach the desired value to be less than 200 ms. Additionally, other requirements relating to the displays programmed in LabVIEW software include: having graphs of the input and output waveforms and being able to record both simultaneously, real time controls for the turntable parameters (K_P, T_I, and T_D), and display indicators showing the user the value for the percent overshoot and if the system passes or fails the previously mentioned specification[7].

The block diagram for the program that will serve as the PID controller is as shown below.

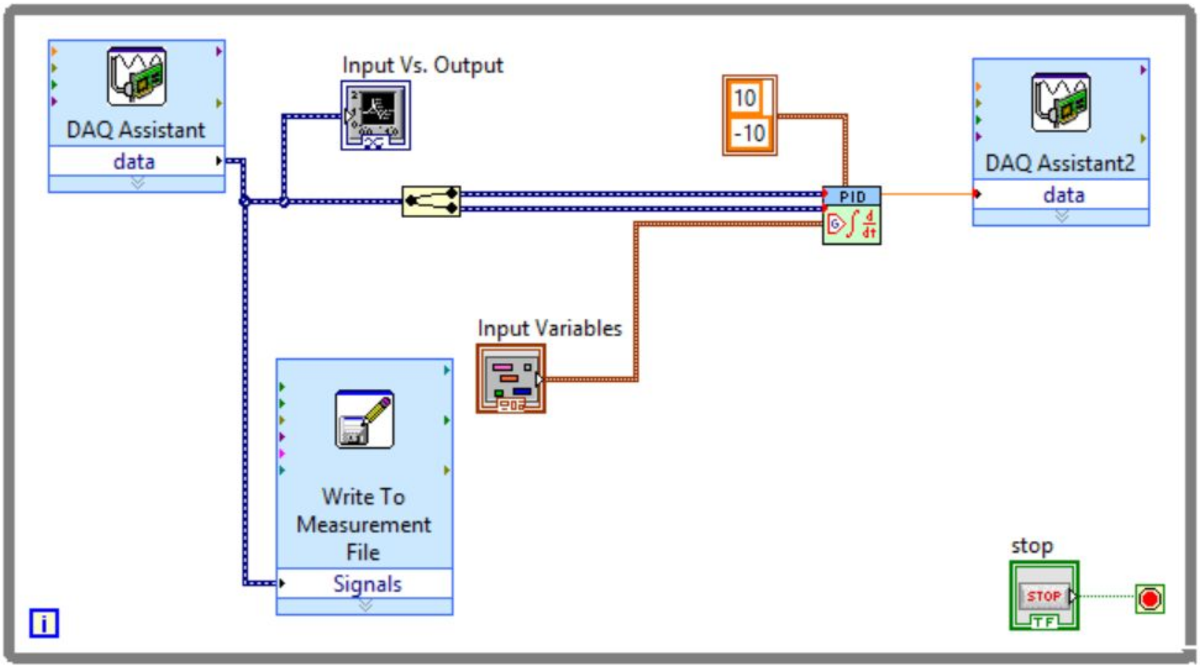

FIG. I. Block diagram of the LabVIEW program.

The DAQ Assistant interfaces with the DAQ to receive both the input function from the wave function generator and the output data from the circuit. It is set to receive one sample on demand so that the system may run continuously. The "Write To Measurement File" function generates an excel file with both sets of data. The name and location of the file can be set within this function. "Input Vs. Output" is a waveform chart that will display the input and output functions in real time on the same axis[5]. The input and output signals are split so that they can be input into the PID function. The input waveform (top branch) goes to the setpoint node since it is the desired output of the system. The output waveform (bottom branch) goes to the process variable node since this is the feedback of the system. "Input Variables" is a cluster of the values input by the user into the control panel. This goes to the PID gains node to specify the proportional gain, integral time, and derivative time values that will be used to process the feedback[5]. The output range of the function is set to (-10,10). Finally, the second DAQ Assistant takes the output of the program and inputs it back into the circuit. Once again, it is set to receive one sample on demand so that the system may run continuously. A stop button was added to the control panel so that the user may end the loop at his/her leisure.

In order for this experiment to be completed, an in-depth analysis of the control system being used is an absolute necessity. The first necessary step of the analysis is to turn the control system into a block diagram. This can be seen in the image below, with numbers to indicate important points of the system for the analysis:

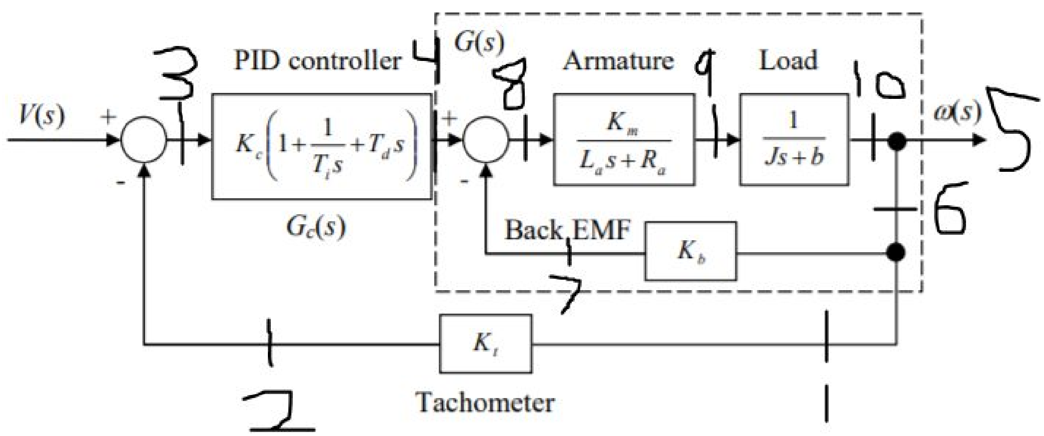

FIG. II. Block diagram of the turntable PID speed control system[11].

From this block diagram, it is possible and quite trivial to derive a transfer function for the system, which is defined as the ratio of output to input signals. For the system derivation, the first mandatory step is to derive the transfer function for the overall block diagram. Nore that the points that will be referred to in this analysis are the ones denoted in the block diagram of the system seen above. At point 1, the equation found is seen below.

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

Where $\omega(s)$ is equivalent to the output signal, and s is known as the laplace variable. From this, the system can be followed via the arrows in the diagram to point 2, and thus goes through the tachometer. The equation found from point 2 can be seen below.

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

For which K_t is a constant found from the tachometer. Next, point 3 is approached. At this point, the equation for point 2 gets subtracted from the input function $V(s)$. This is a result of the negative summer that they system passes through. The point 3 evaluation follows.

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

After point 3, the signal logically goes to point 4, and thus goes through the PID controller as well. The below equation is yielded from this analysis.

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

In which $G_c(s)$ is the transfer function found from the PID controller. The last point that is approached is point 5, in which the signal is traced through the plant, of which the transfer function for will be derived later in this section, and ends up back at the systems output, or in other words, back to the starting point. This is due to the fact that point 1 and 5 are both at the output, and are therefore effectively at the same location. Therefore, the equation at point 5 can be found as

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

Of which $G(s)$ is the transfer function found from the plant, with respect to the Laplace variable once again[13]. After this equation is found, it is then simple to derive the actual transfer function of the system. The basic equation for a transfer function can be seen underneath.

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

Therefore, through factoring out $\omega(s)$ in the equation derived at point 5 and rearranging, the transfer function of the overall system can then be obtained.

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

However, this is not the complete transfer function of the system, primarily because the plant and PID controller transfer functions need to be found. This can be done simply through more block diagram analysis.

The transfer function for the PID controller itself is derived from the use of the typical equation of a PID controller, which is exemplified below. Note that this equation is in the Laplace domain.

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

In which $E(s)$ is the error function, k_p is the proportional coefficient, k_I is the integral with respect to time coefficient, and k_D is the derivative with respect to time coefficient. However, this PID controller is being made digitally, or in other words is being made through digital means via LabVIEW software. Therefore, it is preferred to use an integral time and a derivative time coefficients instead of their typical variable types, and the proportional coefficient is replaced with a constant coefficient that affects all variables. This would result in the following equations.

$$K_p = K_c \tag{9}$$

$$K_i = K_c/(T_i s) \tag{10}$$

$$K_d = K_c T_d s \tag{11}$$

Where K_c is the factored proportional coefficient value which is also referred to as the control coefficient, T_i is the integral time, and T_d is the derivative time. From these equations, the transfer function for the PID controller can be denoted.

$$G_c(s) = K_c(1 + 1/(T_I s) + T_d s) \tag{12}$$

Next, the transfer function for the plant needs to be derived. Just like for the overall system transfer function, the first step is to make the block diagram. An image of it with labeled points can be viewed below.

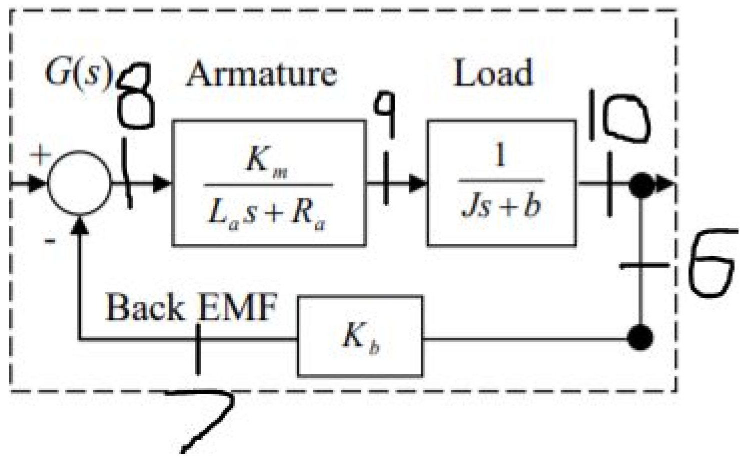

FIG. III. Block diagram of the plant of the speed control system[11].

Once again using the branch path method,the transfer function can be derived. Starting at point 6, the yielded equation can be found and exemplified below.

$$\omega(s) = point\ 6 \tag{13}$$

This is exactly the same as the equation at point 1, which is understandable because point 1 and 6 are effectively the same, being at the output. The system is then followed through the block for the back emf and then goes to point 7 in the diagram. The equation for point 7 is written underneath.

$$K_b\omega(s) = point\ 7 \tag{14}$$

Where K_b is the constant for back emf. This equation is then subtracted from the plant input, once again as a result of a negative summer. In this case, the plant input is denoted R(s) due to the fact that it is different from the overall system input V(s). The operation is demonstrated here.

$$R(s) - K_b \omega(s) = point\ 8 \tag{15}$$

The diagram is then followed to point 9, which is the armature. The following is yielded as a result of this.

$$G_a(s)R(s) - G_a(s)K_b \omega(s) = point\ 9 \tag{16}$$

In which $G_a(s)$ is a function with respect to the Laplace variable, which is defined as

$$G_a(s) = K_m/(L_a s + R_a) \tag{17}$$

Where K_m, L_a, and R_a are all constants specific to the motors used. Finally, the signal is followed through the systems load to reach point 10, which is equal to the output, as seen below.

$$G_a G_L R(s) - G_a G_L K_b \omega(s) = \omega(s) = point\ 10 \tag{18}$$

$G_L(s)$ is the function of the load with respect to s, which is equivalent to the below formula.

$$G_L = 1/(Js + b) \tag{19}$$

For which J and b are constants specific to the used motors. The plant transfer function can now be solved via plugging in the values of G_a and G_L into the equation set at point 10, and then can be easily simplified and rearranged in order to obtain the following plant transfer function, which is what $G(s)$ is equivalent to.

$$G(s) = K_m/((L_a + R_a)(Js + b) + K_b K_m) \tag{20}$$

Now that these are all known, the plant and PID transfer functions can be plugged into the overall system transfer function, and that can be rearranged to be in its simplest form. The fully substituted transfer function for the system is produced below.

$$T = \left(\frac{(K_m/((L_a+R_a)(Js+b)+K_bK_m))K_c(1+1/(T_is)+T_ds)}{1+K_t(K_m/((L_a+R_a)(Js+b)+K_bK_m))K_c(1+1_i/(T_is)+T_ds)}\right) \quad (21)$$

The last bit of theory that needs to be defined for this experiment is how to calculate the percent overshoot of the system. The equation that will be used is stated here.

$$PO = (M_{tp} - F_v)/F_V * 100\% \quad (22)$$

In which PO is the percent overshoot, M_{tp} is the maximum amplitude of the system, and F_v is the desired value of the voltage of the square wave, which is equivalent to 1V. Rise time and settling time can be found by eye from the experimental data plots[13].

At this point, all relevant theoretical and background information required to design and use a PID controller for a turntable motor has been stated and derived, and the experiment is ready to be performed.

II. Experimental Procedure

This experiment requires the following equipment:

- DC motor with a tachometer
- Waveform generator
- DAQ (Data Acquisition software device)
- Data Recording Software (LabVIEW in this case)
- A multimeter
- A power supply
- Resistors
- Op Amps
- Capacitors

- Diodes
- Transistors
- Breadboard
- Wires
- DC Turntable Motor
- Alligator Clips

The necessary equipment will be used to build the circuit according to the diagram below:

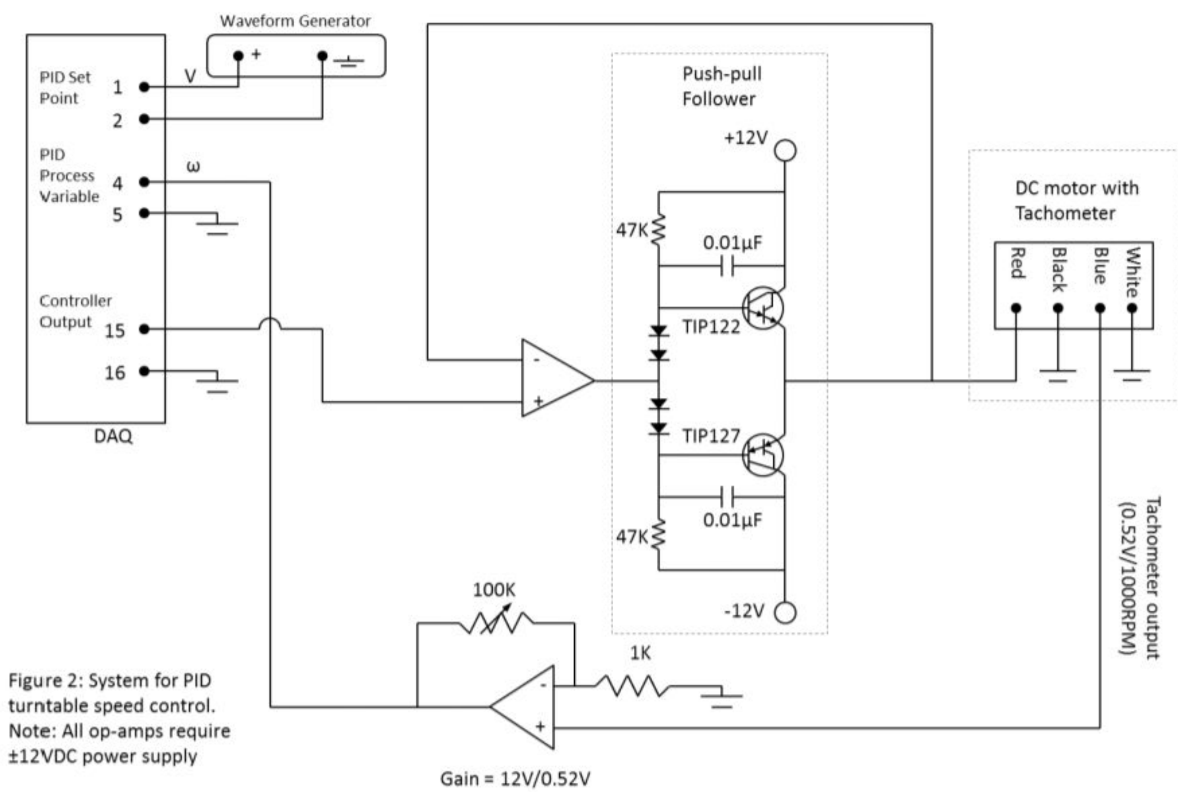

FIG. IV. Circuit Diagram of the turntable PID controller[10].

Before proceeding with the experiment, turn on all the lab equipment. Leave the power supply disconnected until the output of the waveform generator is set.

The resistance value of the potentiometer that will give a gain of 12V/0.52V must be calculated. To determine the relationship between the resistance of the potentiometer and the desired gain, use the fact that no current flows through the op-amp, and that the potential difference of both the positive and negative sides must be equal[12].

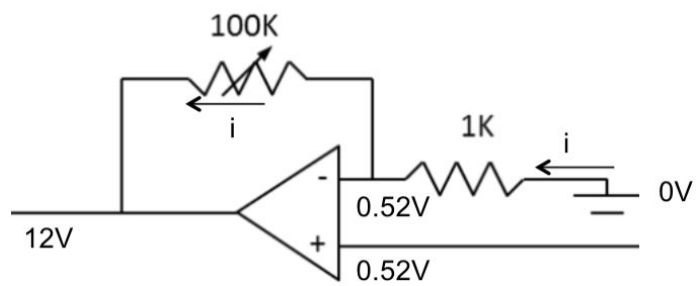

FIG. V. Circuit Diagram of the operational amplifier with a potentiometer[12].

Since the ground has a potential of 0V, the current through the potentiometer can be calculated as follows:

$$V = iR$$

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

Now that both the current and the gain are known, Ohm's Law can be used to calculate the resistance of the potentiometer.

$$V = iR$$

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

Using the multimeter, set the potentiometer to this value.

Now set the power supply to 12V and the wave function generator to a 1 Hz square wave with an amplitude of 1V. Make sure to connect the power supply so that the circuit is being powered. Open the LabVIEW program and set all the input variables to zero as shown in the image below.

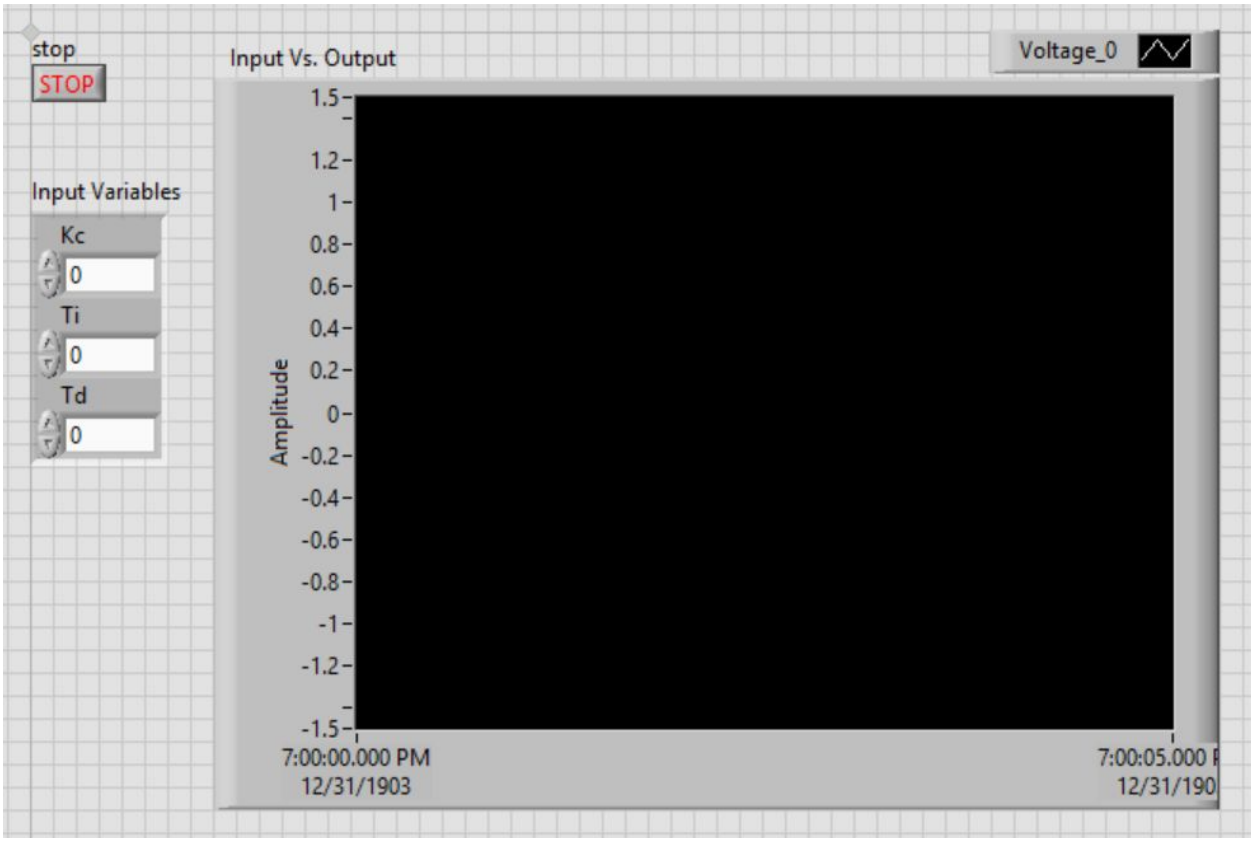

FIG. VI. Control panel of the LabVIEW program.

First adjust K_c so that the output function is as close the input function as possible without generating too much noise. Then adjust T_i so that the output more closely matches the input. Finally, tune T_d to further stabilize the system and record all three values. Make sure that the excel file generated by the program is saved to the correct location. Check the data to ensure that the percent overshoot is less than 10%, the settling time to within 2% of the final value is less than 500 ms, and the rise time is less than 200 ms.

III. Results

The experiment resulted in a K_c of 1.8, a T_i of 0.00075s, and a T_d of $3.4 * 10^{-10} s$. Using the previously found equations for the digital PID controller variables, a value of $K_p = 1.8$, $K_i = 2400$, and $K_d = 6.8 * 10^{-10}$ was found. These values were used to find experimental data value for the turntable motor setup. A plot of the data yielded from these PID parameters can be seen below, made via Google Sheets.

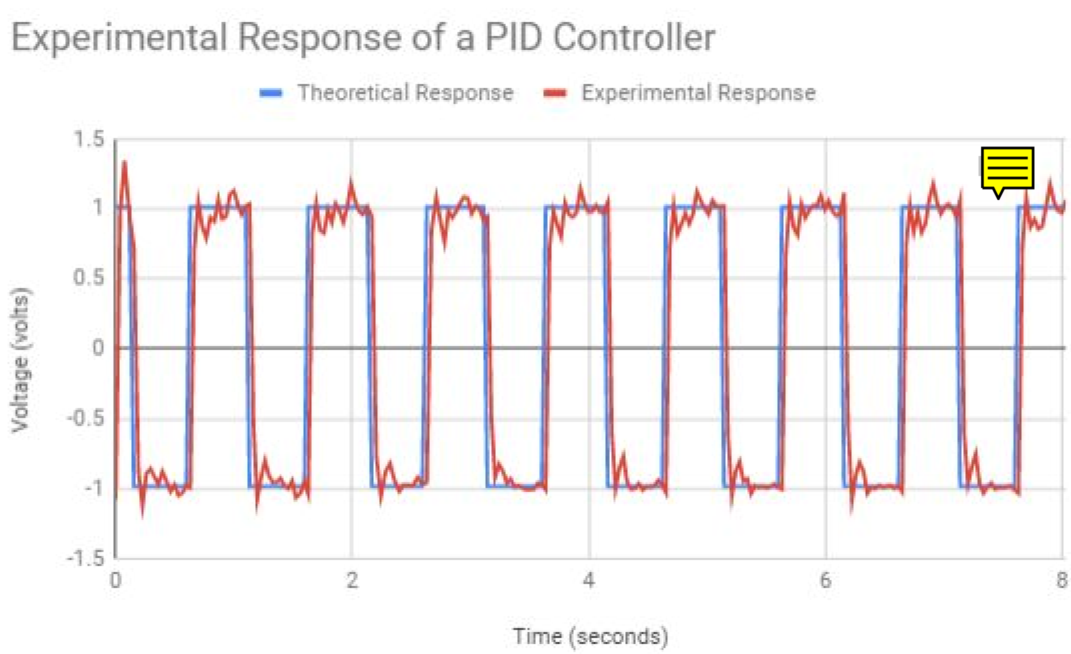

FIG. VII. Plot of experimental inputs and outputs using Microsoft Excel.

The maximum experimental amplitude of this plot appears to be at a value of 1.34V. This value seems rather high however, and also appears irregular. This will be discussed later on in the discussion. The amplitude of the regularly occurring data points appears to be at about 1.08V, which is much more reasonable than at 1.34V. In addition, the experimental response looks rather jagged, but a potential reason for this is that this lab has significantly fewer data points as compared to the others. This will be explored further in the discussion as well.

MATLAB software was used to develop theoretical results of the PID control variables found in the lab. A plot of these theoretical results can be seen below for the full waveform.

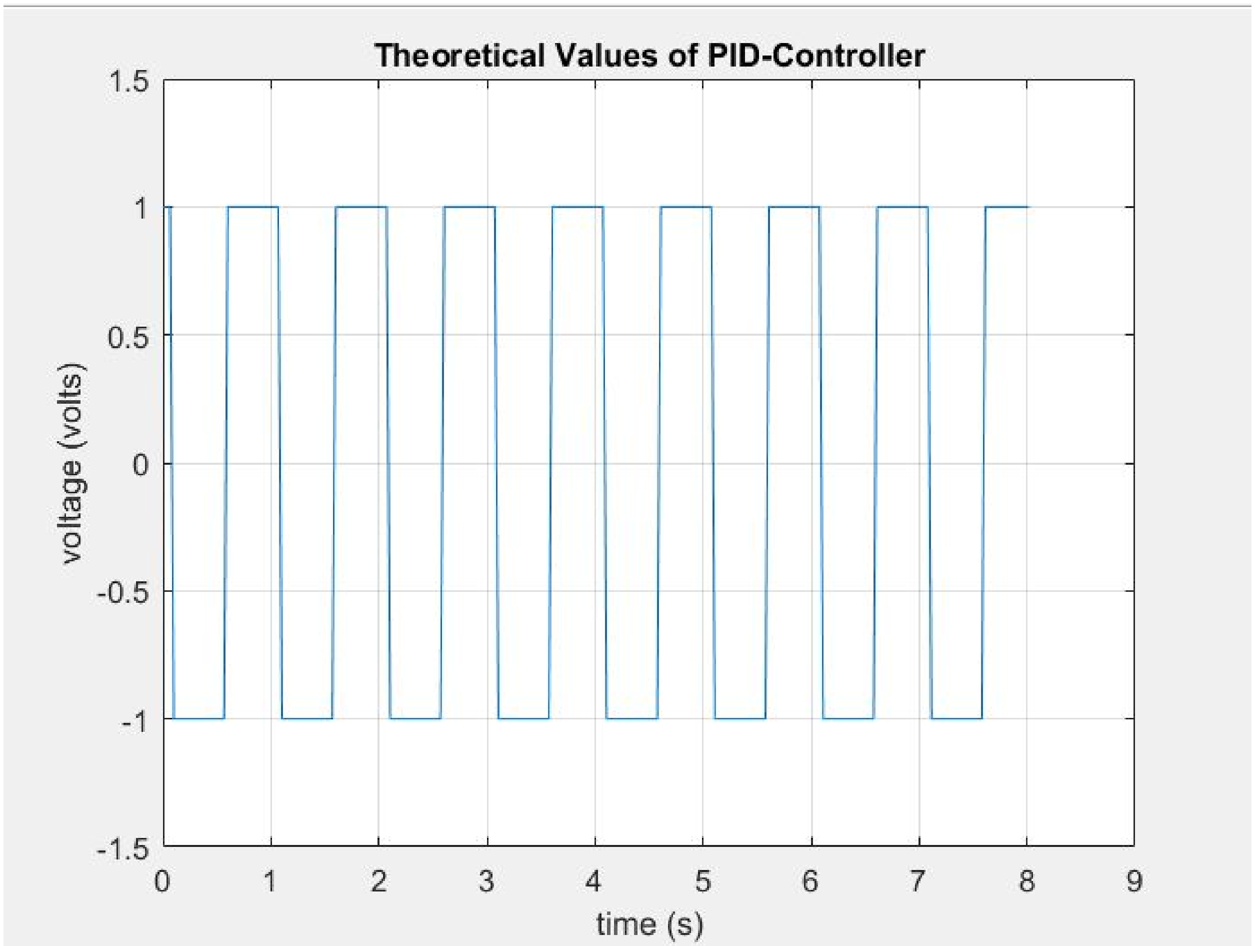

FIG. VIII. Theoretical response of square wave over entire waveform.

The plot matches what a square wave should look like near perfectly. In addition, the theoretical amplitudes max the desired 1V and -1V, so this plot appears to be a pretty accurate representation of what the system should look like, at least from a distance.

The experimental and theoretical results for the full waveforms of the system can be seen in the same plot for comparison below.

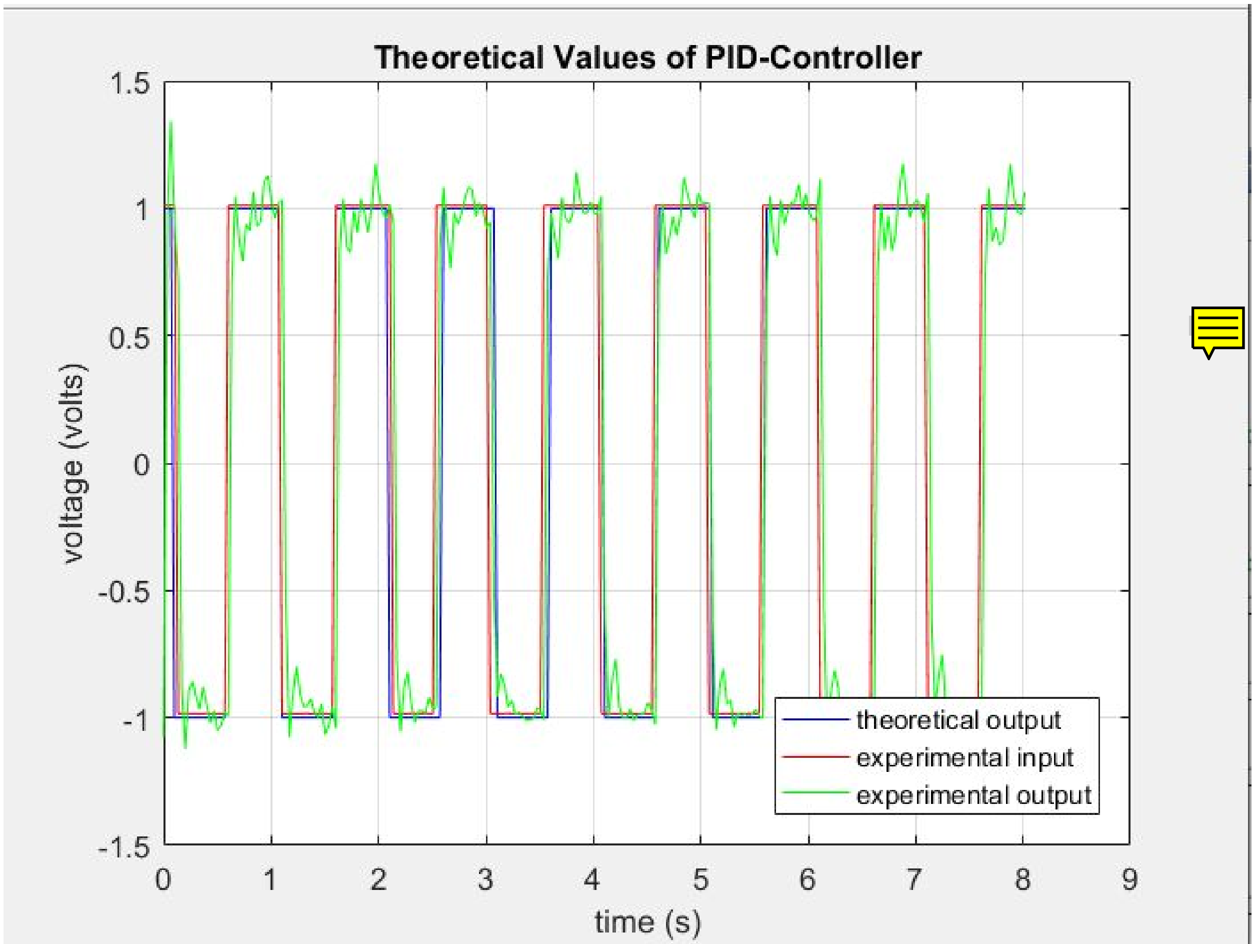

FIG. XI. Plot of experimental data versus theoretical data using MATLAB software.

In this plot, it is clear that the plots mostly line up with each other, despite not being perfect. This is most likely because of noise in the system and fluctuations caused by LabVIEW, which will be explored in the discussion. The plots would also line up better if more data points were recorded. Overall though, the general square wave shape of the data output matches between experimental and theoretical results

Another valuable plot made on MATLAB was the unit step of the theoretical plot. This can be seen below.

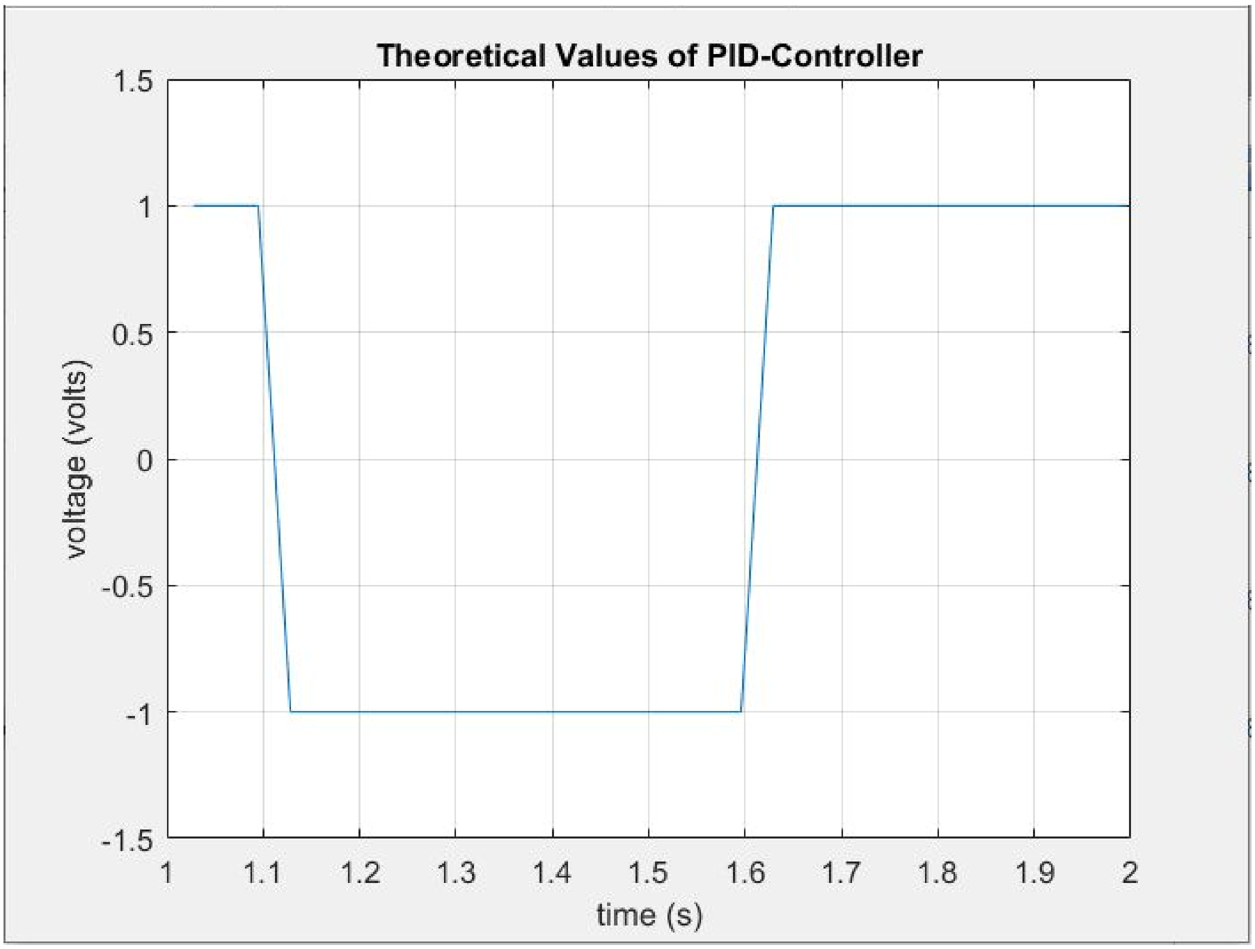

FIG. X. Theoretical plot of unit step data.

This plot ranges from 1 to 2 seconds because the unit step square wave was taken from an arbitrary time interval. The plot looks like it is not a perfect square however, as some of the lines appear at an angle. Most likely, this can again be attributed to not having that many data points, (241 data points for this lab versus several thousand for the previous). This will be explored in the discussion to determine if adding more data points makes data look more salubrious. Below is an image of a plot of the unit step experimental results for comparison sake.

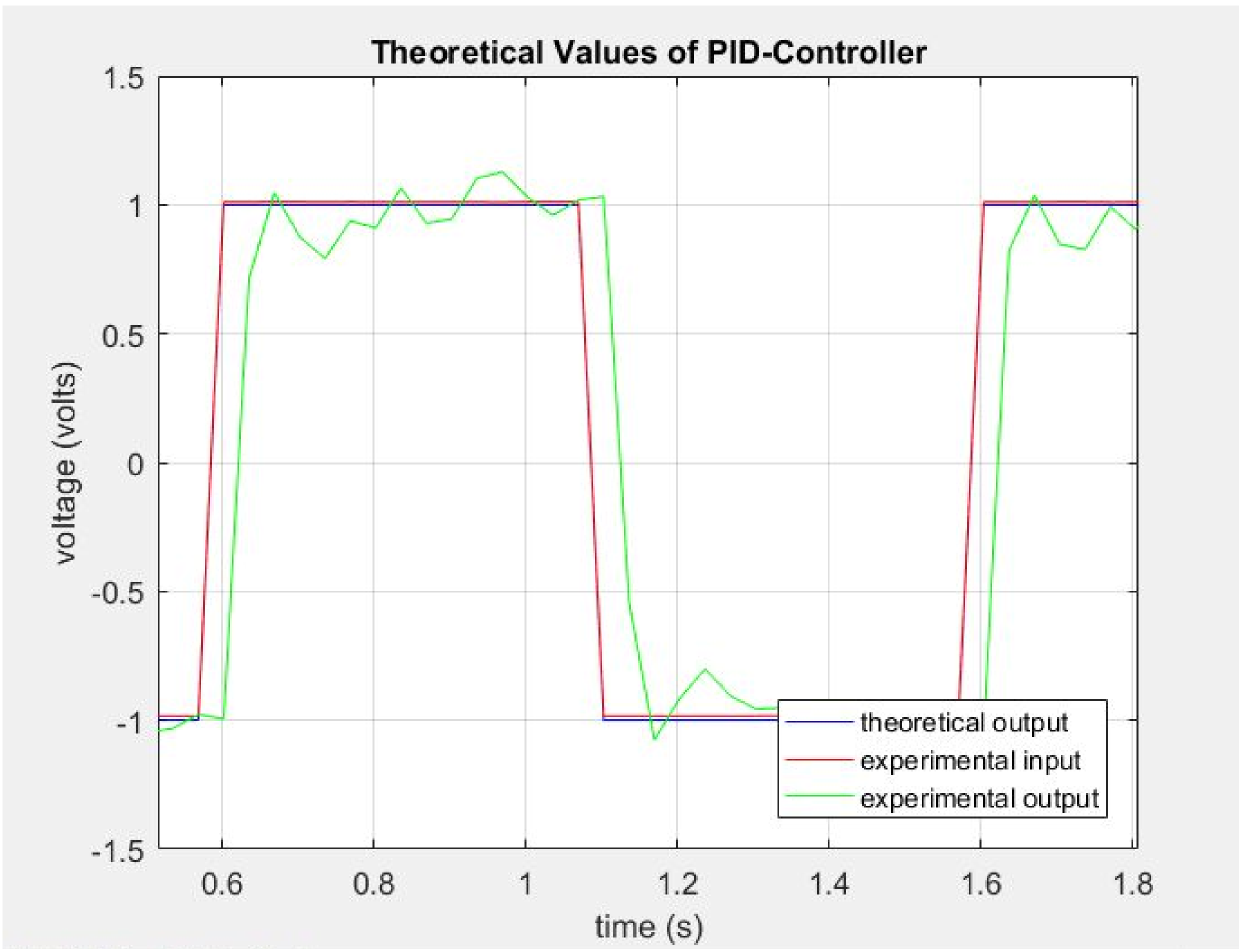

FIG. XI. Unit step of the Experimental Results

This plot demonstrates a similar issue to the theoretical unit step, in which the data lines do not go perfectly in the direction that they should. Because of this, the rise time looks to be significantly longer than it actually is. However, note that the rise time is most likely not a small value. This will be calculated and verified in the discussion for whether or not it meets the 200ms rise time design specification.

Now that the results are known, the next necessity will be to find the performance characteristics of the system. These can be found from the plots with relative ease. For percent overshoot, the maximum voltage is 1.342V. Using the previously stated percent overshoot formula, the calculation is as follows.

$$PO = (1.342 - 1)/1 * 100\% = 34.2\%$$

Therefore, the percent overshoot of the system is at a value of 34.2%. This is higher than the design specification, and reasons for this and their justifications will be discussed in the discussion. The rise time can be found by taking note of the time it takes the motor to go from 0V to the input voltage of 1.0V. In this case, the value was found to be 168.3ms, thus meeting the design specification. Lastly, the 2% settling time needs to be found. This is done by looking for the time in which it takes the system to be within 2% of the input value, which in this case would range from 1.02V to 0.98V. For this experiment, it takes approximately 462.7ms. This parameter meets the design specification of 500ms.

Error Analysis:

In order to obtain the experimental uncertainty in the output voltage, it is first necessary to know the uncertainty of the gain for the non-inverting operational amplifier. To do this, the instrumental uncertainty for the resistance values was determined via use of the tolerances assingned to the 1K Ω resistor and the potentiometer used in the experiment. The error in the voltage gain was determined according to the calculations shown below. In addition, the uncertainty resulting from the DAQ was considered, but since its resolution uncertainty doesn't change while the voltage is being amplified, it is essentially negligible.

- Instrumental Uncertainty in 1KΩ Resistor:

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

- Instrumental Uncertainty in Potentiometer:

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

- Resolution uncertainty in DAQ:

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

- Uncertainty in Voltage 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 given to the parameters of the DC motor, tachometer, and the load that affect the total uncertainty of the PID controllers[7] calculated theoretical output voltage via use of the inverse Laplace transform of ω(s) = T(s)V(s). However, they were not taken into account for the error analysis due to the unfortunate fact that error propagation cannot easily be performed on ω(t) due to it being a complicated inverse laplace transform.

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

- u_{R_a} = 0.15(11.5 Ω) = 1.725 Ω

IV. Discussion

The LabVIEW program was required to be able to display graphs of the input and output waveforms in real time, have user controls for any input parameters that can be adjusted in real time, display the percent overshoot in real time with indicators that show whether the system is passing or failing, detect whether or not the operator is present, and switch off the device if they aren't, and record the input waveform and the turntable speed. The program used for this experiment, due to time constraints, unfortunately only accomplished the first two. If given more time to design the system, the other features could have been added.

Below is a table of the terms that characterize $G_c(s)$. Note that all of these values were previously stated in this lab report and are here for convenience.

Value	PID-Controller
K_C	1.8
T_i	0.00075
T_d	$3.4 * 10^{-10}$

Table 1. Table of Important Coefficients in a PID Controller.

For this PID controller, the values for K_c, T_i and T_d where 1.8, 0.00075, and $3.4 * 10^{-10}$ respectively. This resulted in an experimental maximum amplitude of 1.324267V, which is rather high as compared to the input experimental amplitude of 1.02V. It is also high relative to the theoretical output amplitude which was 1.0V. The high experimental value however only occurs once in the entire plot and can most likely be attributed to noise from the experimental setup. The noise is most likely caused by the LabVIEW program designed for this lab, that given the strict time constraint was not as well-tuned for noise as it could have been. A good way to justify this theory is to use a plot of a similar block diagram and circuit setup. In this case, the best comparison would be the plot from the first lab experiment. This is because the program was much better adjusted for noise due to being made by a professional. The plot from Lab 1 can be seen below[12], with the Lab 3 plot below for better comparison.

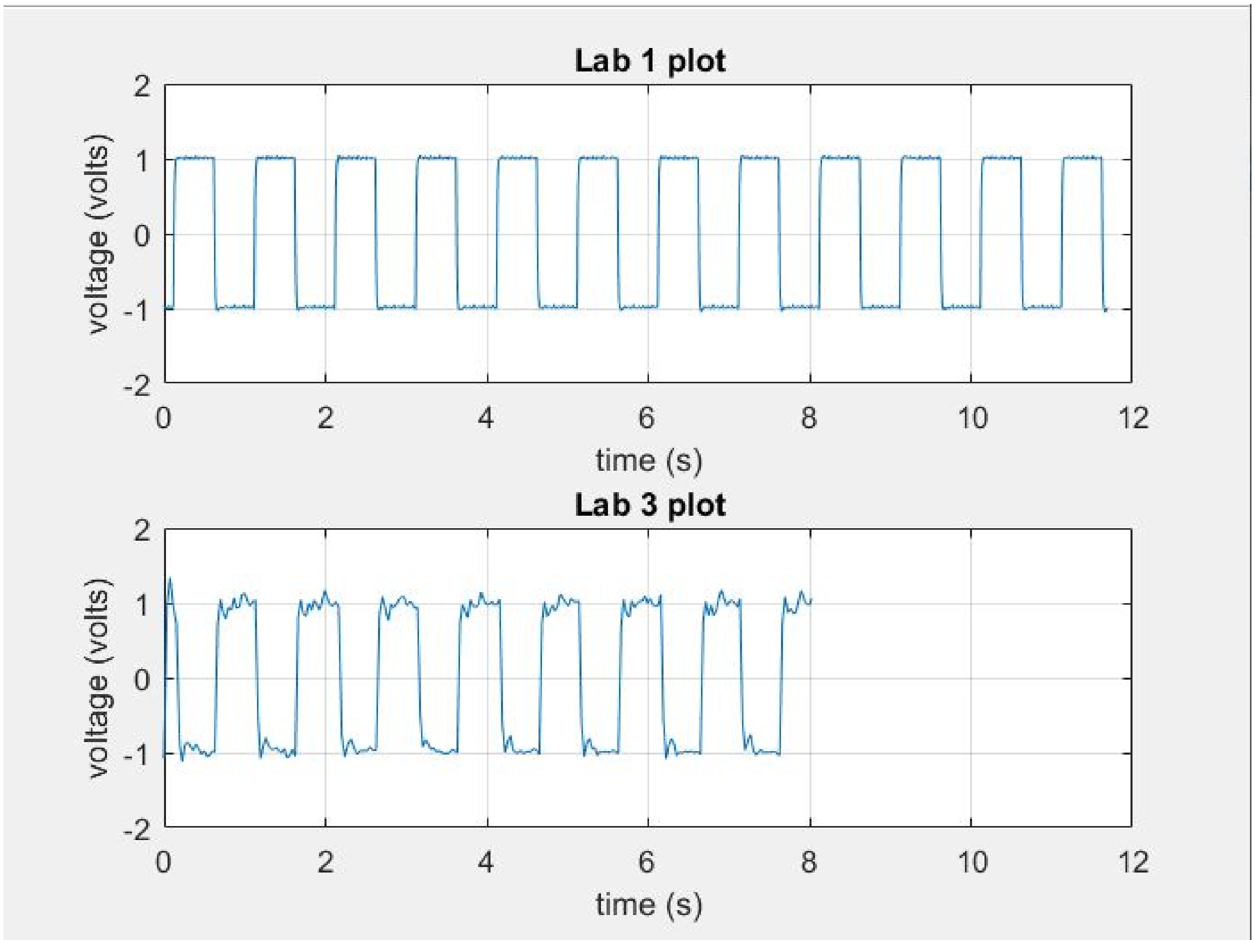

FIG XII. Subplot comparing Lab 1 and 3 Experimental Output.

As is visible by these two plots, the Lab 1 data is much less noisy than the Lab 3 data as expected. However, despite the obvious fact that the motors weren't at the same point in the rotation for when data recording occurred between the two experiments, the plots appear to both be relatively in the same square wave shape, and thus both likely meet the general form of the input square wave. Lab 1 obviously was less affected by noise however, as proven by it's percent overshoot of less than 3%, and thus looks much more precise than the plot of Lab 3, in which the noise makes the data fluctuate heavily. Therefore, it seems safe to assume that if noise was reduced in the lab, the experimental output for Lab 3 would be much smoother and would most likely match the theoretical output values better than it did in the current state.

Another important takeaway from the data plots seen in the results section is that the plots are not a perfect square wave. This is most noticeable in the plot of the theoretical experimental output square wave, in which some of the plots lines go at an angle, as well as how the experimental plots feature more spikes. The most likely source of this is the fact that this lab only had 241 data points taken, as compared to the 3,000 plus collected in the other labs for the course. If a linear interpolation was done on the lab data in order to obtain more data points, then the plots should even out and look more square like. This can be seen below in a plot of 10281 linearly interpolated data points.

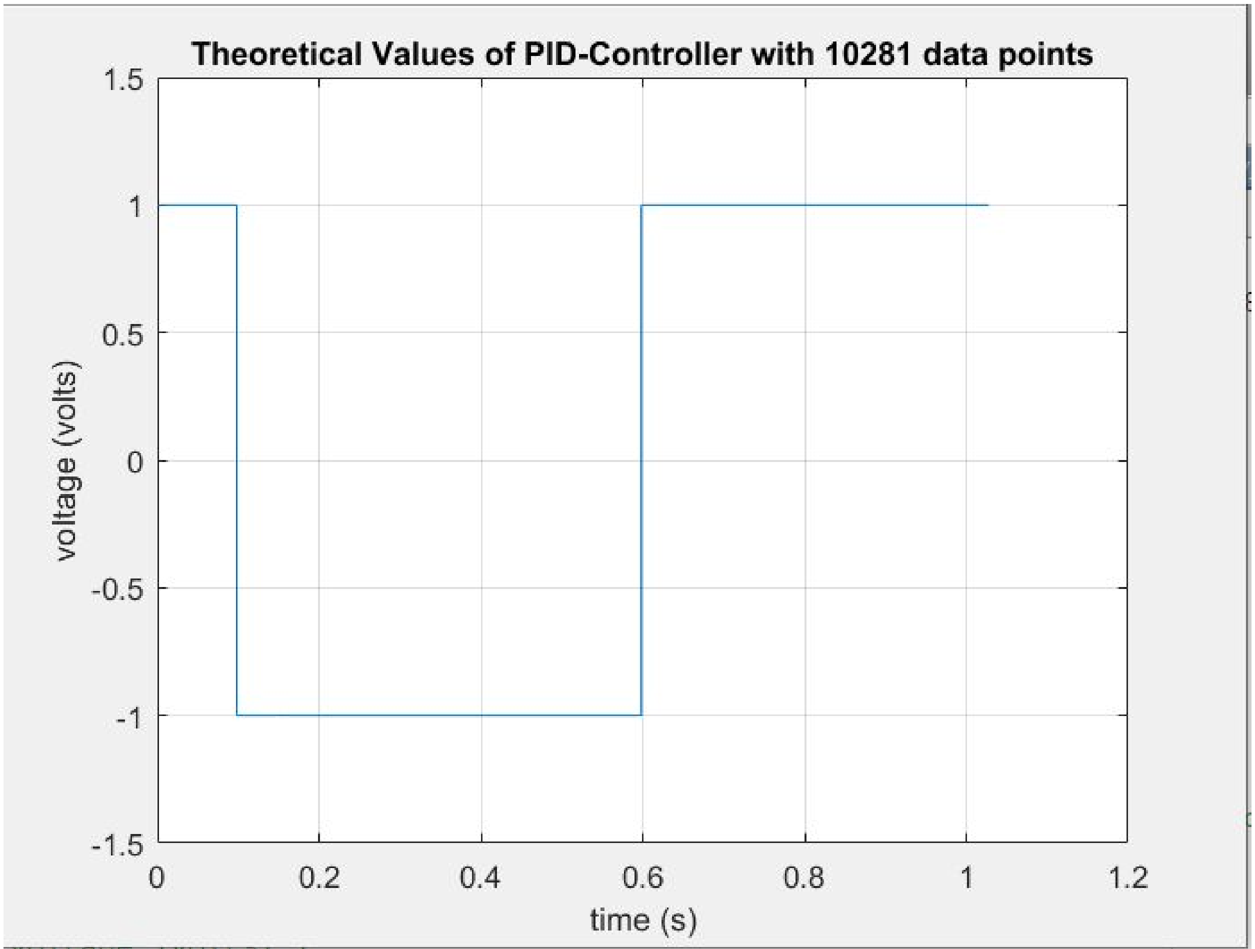

FIG XIII. Square Wave Plot with 10281 Data Points.

From this plot, it should make eminent good sense that the theoretical plots not looking perfect in the results is a direct cause of the limited number of data points, and that the more data points collected in an experiment, the more accurate the data will be. The same reasoning can be used to explain how the experimental plots look jagged, as if there was more data points the spikes would be significantly less noticeable. The best way to justify this is by comparing previous experimental lab results due to the inability to obtain more data points outside of lab. In this case, Lab 2 was chosen for comparison due to a plethora of data points[13]. A subplot comparing the two follows.

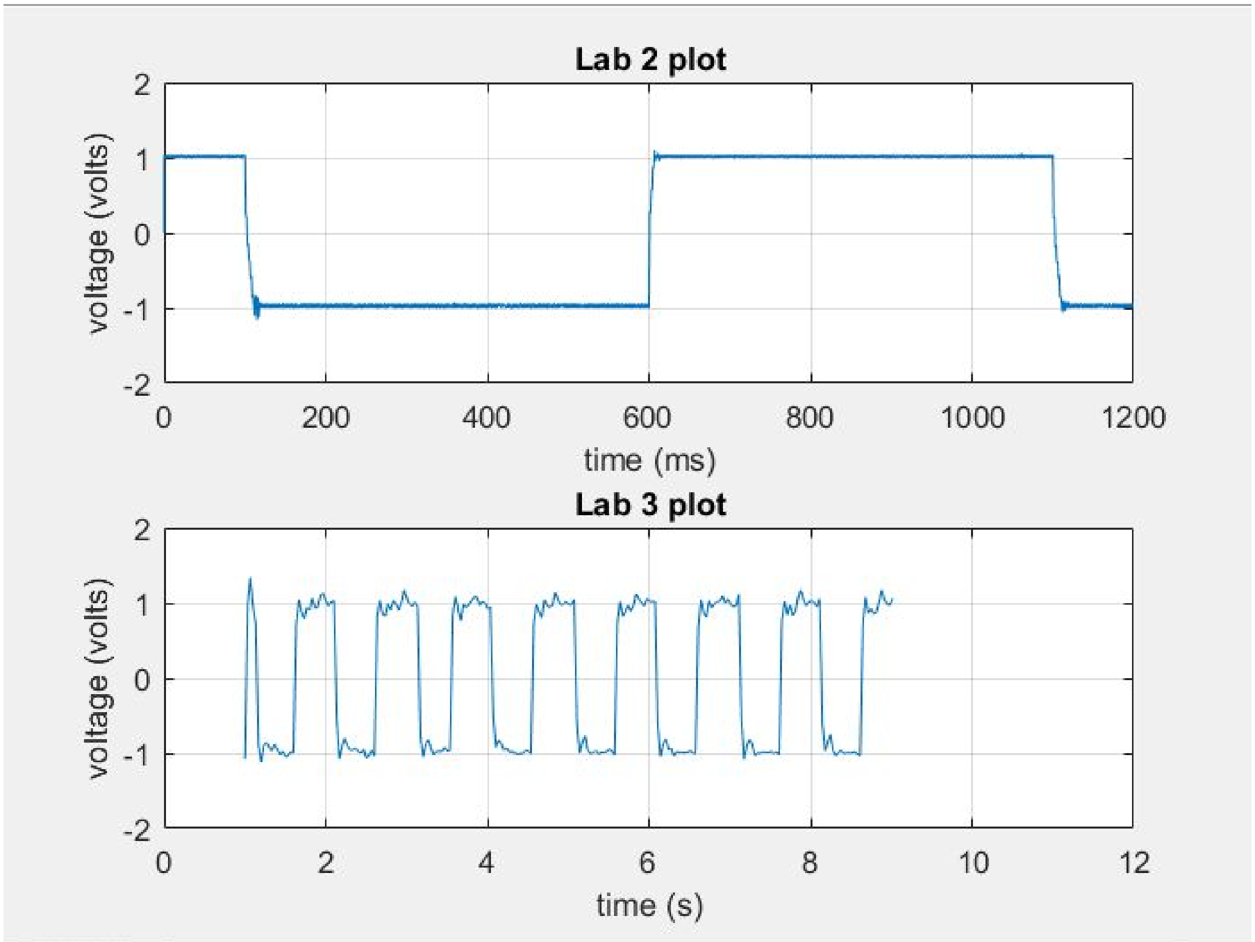

FIG XIV. Comparing Lab 2 and 3 Data in Regards to Data Points Number.

These plots are clearly quite different from each other in terms of the actual data values at given times. However, what is important in this case is how the Lab 2 plot, by having 1202 data points, appears to be much smoother than the data in Lab 3, which as previously stated has 241 data points. By comparing these plots, it seems reasonable to assume that if Lab 3 had more data points, the plot would look much less jagged, despite still being very noisy and with similar overshoot issues. It also stands to reason that for Lab 2, if only a small number of data points were known, the plot would look jagged as well. Admittingly, there are other factors that could affect why the Lab 3 plot is jagged, but based on the evidence provided from lab 2, it appears justified that the limited number of data points played a role in this.

The PID controller was also required to meet several design parameters as previously stated. The parameters that are directly found from the results include the allowable percent overshoot being no more than 10%, the rise time being less than 200 ms, and the time it takes the system to settle to less that 2% be no more than 500 ms. The parameters found for the PID controller can be seen in the table below. Note there values were previously found in the results section and are here for convenience.

Percent Overshoot	34.2%
Rise Time	168.3 ms
2% Setting Time	462.7 ms

Table 2. Performance Parameters of the PID Controller.

The rise time for the motor is at a value of 168.3 ms, which is less than the value of 200 ms set in the design specifications. In addition, the 2% settling time is at a value of 462.7 ms, which again is less than the design specification of 500ms. The percent overshoot of 34.2% however, is significantly higher than the 10% requirement. However, this is believed to be a direct result of the noise of the system. This is because the peak that brings the maximum amplitude to be 1.342V only occurs once and is very irregular. If only data that occurs regularly is considered, or in other words approximate data points that occur multiple times and therefore can not be attributed to noise, the new maximum amplitude of the system will be around 1.08V. This would result in a percent overshoot of approximately 8%, which is indeed within the design specification of less than 10%. Therefore, if noise is to be neglected, the designed PID controller meets all the performance parameter requirements. However, noise is an important factor in design and in the future efforts to avoid significant interference from it should be considered given a large time frame to perform the experiment.

In order to comprehend the full grasp of how the PID Controller works, it is vital to compare the designed digital PID Controller from this lab with the digital one from Lab 1 and analog one from Lab 2. A table comparing the PID controller variables can be seen below.

	K_p	K_i	K_d
Lab 1	3	$6.5646 * 10^3$	$9.39 * 10^{-10}$
Lab 2	5.625	$1.7655 * 10^5$	$2.82 * 10^{-10}$
Lab 3	1.8	$2.4 * 10^3$	$6.12 * 10^{-10}$

Table 3. Comparing PID Variables from Multiple Lab Experiments

From this table it is clear that all the variables are relatively close to each other, with the only significant difference being the integral term for the Lab 2. This term is different by an order of magnitude of two as compared to the others. This however makes sense because it is a different kind of control system, being based off of an analog circuit instead of a digital program. That is the only notable difference between the control variables.

The three different controls systems also each had their own performance parameters. The design specifications were the same between all three experiments. A table comparing these values can be seen below. Note that this table is the table of the originally found parameters, not the ones that neglect noise.

	Percent Overshoot	Rise Time	Settling Time
Lab 1	2.6166%	155.2ms	429.2ms
Lab 2	16%	9.5ms	157.6ms
Lab 3	34.2%	168.3ms	462.7ms

Table 4. Comparing Performance Parameters from all Three Labs.

From the data it is clear that Labs 1 and 3 are as a whole slower than Lab 2. This is seen in their rise times and settling times, which for Lab 1 were 155.2 ms and 429.2 ms respectively, while for Lab 3 were 168.3 ms and 462.7 ms respectively. Lab 2 was faster in both respects with a rise time of 155.2 ms and a settling time of 429.2 ms. On the other hand, Lab 1 was the most accurate with a 2.6166% rise time and settling time of 157.6 ms. This seems reasonable because Labs 1 and 3 were both performed on a digital PID controller, while Lab 2 was done with an analog control system. Typically, a digital control system has a longer rise and settling time because the data has to be processed by a computer which takes extra time, while for the analog control system the use of a circuit negates the need for the computer to process the data, and by extension takes a shorter amount of time to produce the desired wave function. Typically, a digital control system is more accurate than the analog system, or in other words has a smaller percent overshoot. This is because the computer processes the data and makes it more accurate in a digital control system. This does not happen for the analog circuit. Therefore, the percent overshoot of 2.6166% from Lab 1 being much lower than the 16% percent overshoot from Lab 2 makes sense according to theory. However, Lab 3 having its very high percent overshoot of 34.2% does not agree with this theory. As previously stated, this was attributed to noise and justified by using the largest regularly occurring amplitude to obtain a percent overshoot of around 8%. A similar calculation was done in Lab 2 to which the percent overshoot ignoring noise and only taking into account regularly occurring data spikes, was 4%. Therefore, even without noise, the percent overshoot found in Lab 2 is lower and therefore the system as a whole is likely more accurate compared to Lab 1. Since the digital control system should be more accurate than the analog control system, this should not be the case. This can probably be attributed to the fact that the LabVIEW PID function was used for Lab 3. According the Texas Instruments manual on the built in LabVIEW PID Controller block, data that is input into this function can fluctuate by about ten percent of the actual value[5]. This would mean that for a system with an input voltage of 1V, the system can fluctuate by 0.1V in both directions. By taking this into account, the uncertainty range of the Lab 3 LabVIEW function would be between 0.9V to 1.1V, which would allow for the maximum percent overshoot of the system to be 10%. This results in the 8% corrected value landing within the percent overshoot limit that accounts for system fluctuations. Therefore, the corrected percent overshoot value of the Lab 3 controller being larger than that of the Lab 2 value can be attributed to the use of the built-in LabVIEW

PID function. Lab 1 had the smallest percent overshoot because it was a digital PID controller that did not use the built-in PID function, but rather was made by an expert in the field of control systems.

From the comparison of the performance parameters, it is clear that the digital PID controller from Lab 1 appears to be the best. This is because it has the smallest percent overshoot, and additionally is the only PID controller of the three labs that meets the percent overshoot design specification without neglecting noise. It does have a slower rise and settling time than Lab 2, but it is more important to meet all the specifications than to meet some really well and then miss some entirely. In addition, the Lab 1 controller serves as a better PID controller than the digital one in Lab 3 because of the abundance of noise in the latter. This also verifies that it is generally more effective to not use the built-in LabVIEW PID controller because of the abundance of noise and fluctuations that it results in.

The errors in this lab range from being relatively minor to being very large. The large errors can be attributed to noise in the system and fluctuations caused by the PID function on LabVIEW. Minor errors can be attributed to the natural resistance of the wires and the breadboard, the total resistance of the resistors breaking down with age, and the dissipation of heat in resistors and other electrical devices through repeated use. Friction as a result of the rotating motor was also likely a cause of error in the lab experiment because it would have slowed down with time a little bit. Another potential error source is the fact that there could have been an error in the signal produced by the waveform generator, as well as in the data read by the DAQ. In addition, a connection could have been faulty to the point where some extra noise could have been caused. In the future, the best way to avoid these errors is to perform repeated trials, and use newer equipment. Using LabVIEW functions with less associated error would also help improve the experiment.

If the designed PID controller was going to be implemented in industry, the cost, matiniance, installation, and training to use it must be taken into account. One of the costs involved would be the cost of the LabVIEW program, which for current 2019 LabVIEW editions costs either $399 per year for the license, or at a one time retail price of $3,149 dollars for the standard package[6]. Most likely the annual cost would be paid such that if a new, more efficient software came out, it would be replaced with minimal monetary loss. In addition, the cost of electrical supplies, the DAQ, waveform generator, and motor must be considered (making the assumption that a computer is already owned). A typical DAQ costs about $171 dollars, and the cost of the required electrical components can be estimated at around 50 dollars depending on their quality[14]. A waveform generator costs $319 for a typical model[15]. Lastly, the motor looks to cost around $20 for an average model that isn't low quality, but not super high quality either[17]. Therefore, the installation costs look to be around $560, with an additional $1,995 if the program is used for a hypothetical five years. Therefore, the total cost can be estimated at around $2,555. The maintenance required would most likely not be much, just a technician who knows how to work with and fix a waveform generator and LabVIEW software. In addition, an employee will have to maintain the motor, and temporarily replace it if it gets too worn down from repeated usage. Maintenance costs can be estimated at around a $100 per year, meaning that the system is relatively low maintenance over the hypothetical period of five years. Training will also be required for the use of the system, particularly in LabVIEW. The cost of the LabVIEW training course, taught online, is $300 per employee[16]. In addition, the employees legally have to be paid at least \$30,000 per year. For five years of operation, this would be \$150,000. Including these costs, the total cost for one PID controller for a turntable motor is set at \$152,955, assuming one employee uses each system. Therefore, if say 10 systems are needed in total, the cost for the total implementation is \$1,529,550 dollars USD.

The implemented lab would not be a good choice to use in industry. This is because for a cost of \$1,529,550 for 10 systems for 5 years of use, the error with noise and fluctuations in the system are just too high for it to be justifiable. The PID controller used for Lab 1 would be a better choice since it meets all the design specifications, and would cost approximately the same.

Conclusion

- The LabVIEW program found $K_c = 1.8,\ T_i = 7.5 * 10^{-4}s,\ T_d = 3.4 * 10^{-10}s$. This resulted in PID controller variables of $K_p = 1.8,\ K_i = 2400,\ K_d = 6.12 * 10^{-10}$.
- The experimental results yielded a rise time of 168.3 ms, which is less than the 200 ms specification.
- A 2% settling time of 462.7 ms was found, which is less than the 500 ms requirement.
- A percent overshoot of 34.2% was found, which does not meet the design specification of 10%
- The error in the percent overshoot can most likely be attributed to the abundance of noise in the system and is recalculated to be around 8%, which is within the specified 10%.
- When comparing Labs 1, 2, and 3, it is demonstrated that a digital control system that does not use the LabVIEW PID controller block is the most accurate, and that an analog control system generally speaking has a faster response time.

- Since the system from Lab 1 met all the design specifications while the systems from Labs 2 and 3 did not, it can be considered the best of the three.
- Any other error in this lab 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.
- The designed control system was proven to not be a wise implementation in industry.

References

[1] *EXCTECH Laser Tachometer, 2 to 99,999 rpm,* Grainger Industries, Grainger.com (2019) https://www.grainger.com/product/1TZP5?gclid=EAIaIQobChMIlaThmdT05QIVCZyzCh1mOwQrEAQYASABEgKgYfD_BwE&cm_mmc=PPC:+Google+PLA&ef_id=EAIaIQobChMIlaThmdT05QIVCZyzCh1mOwQrEAQYASABEgKgYfD_BwE:G:s&s_kwcid=AL!2966!3!281733071180!!!g!540445061423!

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

[3] *LabVIEW: PID Control Toolset User Manual,* National Instruments (2001). http://experimentationlab.berkeley.edu/sites/default/files/PID_Manual.pdf

[4] *LabVIEW: PID and Fuzzy Logic Toolkit User Manual,* National Instruments (2009). http://www.ni.com/pdf/manuals/372192d.pdf

[5] *LabVIEW: PID VIs - LabVIEW 2018 Help,* National Instruments (2018). http://zone.ni.com/reference/en-XX/help/371361R-01/lvpid/pidtitle/

[6] *LabVIEW: Select Your LabView Edition,* National Instruments (2019) https://www.ni.com/en-us/shop/labview/select-edition.html

[7] Machtay, Noah, *Design Project: Turntable Speed Control System Design - MEC 411 Lab 3*, Stony Brook University (2019).

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

[9] Machtay, Noah. *MEC 411.02 Control System Analysis and Design,* Stony Brook University, Stony Brook, NY. Lecture (2017).

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

[11] Machtay, Noah, *PID Speed Control of a Turntable - MEC 411 Lab 2*, Stony Brook University (2019).

[12] McLaren, Meghan. Miles, Gerard. Zelinska, Mariya. *MEC 411 Lab 1: Digital PID Speed Control of a Turntable,* Stony Brook University (2019)

[13] McLaren, Meghan. Miles, Gerard. Zelinska, Mariya. *MEC 411 Lab 2: Analog PID Speed Control of a Turntable,* Stony Brook University (2019)

[14] *NI Low-Cost, Multifunctioning DAQ Devices,* National Instruments, ni.com (2019). https://www.ni.com/en-us/landing/low-cost-daq.html

[15] Siglent Technologies. *Siglent Technologies SDG1032X Arbitrary Waveform - Function Generator*, Siglent Technologies, Amazon. https://www.amazon.com/Siglent-Technologies-SDG1032X-Arbitrary-Waveform/dp/B01LECZU98/ref=asc_df_B01LECZU98/?tag=hyprod-20&linkCode=df0&hvadid=312249734809&hvpos

=1o3&hvnetw=g&hvrand=13739120798396647881&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9060089&hvtargid=pla-382666487008&psc=1&tag=&ref=&adgrpid=60689741685&hvpone=&hvptwo=&hvadid=312249734809&hvpos=1o3&hvnetw=g&hvrand=13739120798396647881&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9060089&hvtargid=pla-382666487008

[16] *Understanding the Value of NI Training,* National Instruments (2019) http://www.ni.com/training/value/

[17] *78RPM 33/45 DC9-12V 3-Speed Turntables Motor 25mm Mounting Holes,* Alexnd.com. (2019) https://alexnld.com/product/78rpm-33-45-dc9-12v-3-speed-turntables-motor-25mm-mounting-holes/?gclid=EAIaIQobChMIlb6gitP05QIVjZyzCh2LCgyKEAQYBCABEgKLcvD_BwE

Appendix

LabVIEW Program

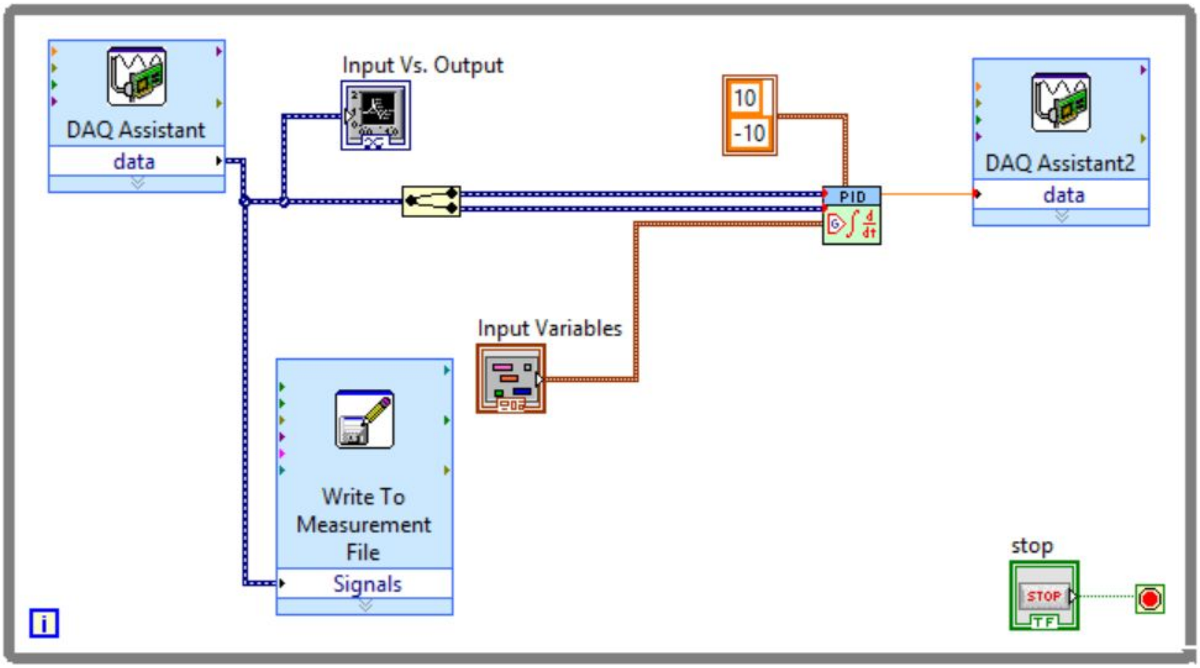

FIG. XIV. Block diagram of the LabVIEW program.

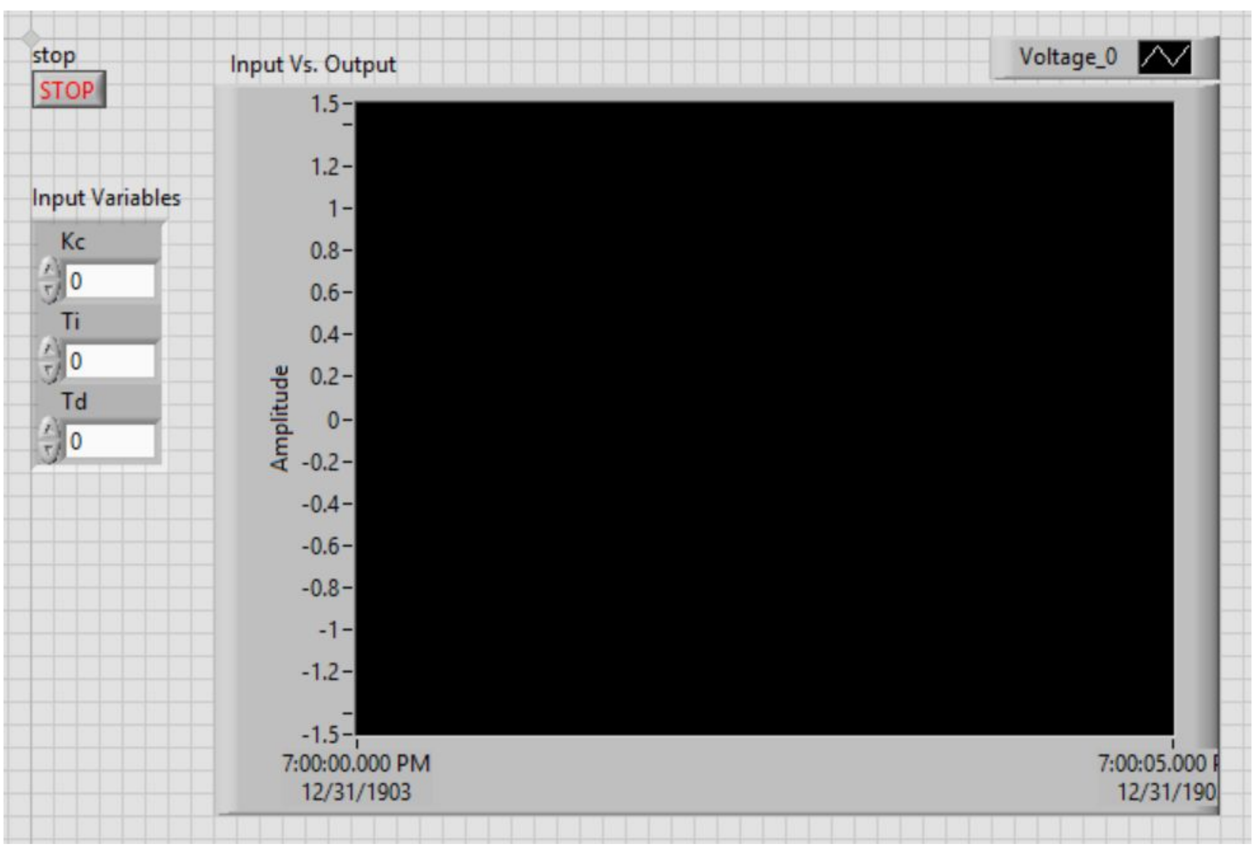

FIG. XV. Control panel of the LabVIEW program.

MATLAB Code

```
close all

clc

clear

exceld=xlsread('Lab 3 Data_1');

excel1=xlsread('MEC_411_Lab1_Data_PID_1');

%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-integral-derivative cotroller

%knob values

Kc=1.8; %value of Kc constant
```

```
Ti=0.00075; %integral time MINUTES

Td=3.4*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

td=0:.0334:8.028066; %time range of 1 second (1 period)

t=1:.0334:9.028066; %in order to get a nicer plot (since the motor is already running when data collection starts)

[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

wd=zeros(r,c); %zero vector of output

wd=zeros(r,c);

win=exceld(:,2);

wot=exceld(:,3);

w1=excel1(:,3);

for i=1:c

    wd(i)=12*(1/12 -
(868915432848014719320064000*exp(-(329021251066317476188032860 16*t(i))/3475661745774987852464992021)*(cos((47
4989023199232*8641172098656964999218391927 06^(1/2)*t(i))/3475661745774987852464992021) -
(125123821035650511815462974 11*8641172098656964999218391927 06^(1/2)*sin((474989023199232*86411720986569649
99218391927 06^(1/2)*t(i))/3475661745774987852464992021))/546311479004995234859502178837735488566067200))/1042
698523732496355739497606 3);


    %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)>8.098324 && t(i)<8.598324 %below equations copy pasted from symbolic toolbox

    wd(i)=-12*(1/12 -
(8689154328480147193200640000*exp(-(3290212510663174761880328601 6*t(i))/347566174577498785246499 2021)*(cos((47
4989023199232*864117209865696499921839192706^(1/2)*t(i))/3475661745774987852464992021) -
(1251238210356505118154629741 1*864117209865696499921839192706^(1/2)*sin((474989023199232*864117209865696 49
9921839192706^(1/2)*t(i))/3475661745774987852464992021))/5463114790049952348595021788377354885660 67200))/1042
6985237324963557394976063);


  elseif t(i)>1.098324 && t(i)<1.598324 %below equations copy pasted from symbolic toolbox

    wd(i)=-12*(1/12 -
(8689154328480147193200640000*exp(-(3290212510663174761880328601 6*t(i))/347566174577498785246499 2021)*(cos((47
4989023199232*864117209865696499921839192706^(1/2)*t(i))/3475661745774987852464992021) -
(1251238210356505118154629741 1*864117209865696499921839192706^(1/2)*sin((474989023199232*864117209865696 49
9921839192706^(1/2)*t(i))/3475661745774987852464992021))/5463114790049952348595021788377354885660 67200))/1042
6985237324963557394976063);

  elseif t(i)>2.098324 && t(i)<2.598324

    wd(i)=-12*(1/12 -
(8689154328480147193200640000*exp(-(3290212510663174761880328601 6*t(i))/347566174577498785246499 2021)*(cos((47
4989023199232*864117209865696499921839192706^(1/2)*t(i))/3475661745774987852464992021) -
(1251238210356505118154629741 1*864117209865696499921839192706^(1/2)*sin((474989023199232*864117209865696 49
9921839192706^(1/2)*t(i))/3475661745774987852464992021))/5463114790049952348595021788377354885660 67200))/1042
6985237324963557394976063);

  elseif t(i)>3.098324 && t(i)<3.598324

    wd(i)=-12*(1/12 -
(8689154328480147193200640000*exp(-(3290212510663174761880328601 6*t(i))/347566174577498785246499 2021)*(cos((47
4989023199232*864117209865696499921839192706^(1/2)*t(i))/3475661745774987852464992021) -
(1251238210356505118154629741 1*864117209865696499921839192706^(1/2)*sin((474989023199232*864117209865696 49
```

```
9921839192706^(1/2)*t(i))/3475661745774987852464992021))/546311479004995234859502178837735488566067200))/1042
69852373249635573949760630);

  elseif t(i)>4.098324 && t(i)<4.598324
      wd(i)=-12*(1/12 -
(868915432848014719320064000*exp(-(3290212510663174761880328601 6*t(i))/3475661745774987852464992021)*(cos((47
4989023199232*864117209865696499921839192706^(1/2)*t(i))/3475661745774987852464992021) -
(1251238210356505118154629741 1*864117209865696499921839192706^(1/2)*sin((474989023199232*86411720986569649
9921839192706^(1/2)*t(i))/3475661745774987852464992021))/546311479004995234859502178837735488566067200))/1042
69852373249635573949760630);

  elseif t(i)>5.098324 && t(i)<5.598324
      wd(i)=-12*(1/12 -
(868915432848014719320064000*exp(-(3290212510663174761880328601 6*t(i))/3475661745774987852464992021)*(cos((47
4989023199232*864117209865696499921839192706^(1/2)*t(i))/3475661745774987852464992021) -
(1251238210356505118154629741 1*864117209865696499921839192706^(1/2)*sin((474989023199232*86411720986569649
9921839192706^(1/2)*t(i))/3475661745774987852464992021))/546311479004995234859502178837735488566067200))/1042
69852373249635573949760630);

  elseif t(i)>6.098324 && t(i)<6.598324
      wd(i)=-12*(1/12 -
(868915432848014719320064000*exp(-(3290212510663174761880328601 6*t(i))/3475661745774987852464992021)*(cos((47
4989023199232*864117209865696499921839192706^(1/2)*t(i))/3475661745774987852464992021) -
(1251238210356505118154629741 1*864117209865696499921839192706^(1/2)*sin((474989023199232*86411720986569649
9921839192706^(1/2)*t(i))/3475661745774987852464992021))/546311479004995234859502178837735488566067200))/1042
69852373249635573949760630);

  elseif t(i)>7.098324 && t(i)<7.598324
```

```
        wd(i)=-12*(1/12 -
(868915432848014719320064000*exp(-(329021251066317476188032860l6*t(i))/3475661745774987852464992021)*(cos((47
4989023199232*864117209865696499921839192706^(1/2)*t(i))/3475661745774987852464992021) -
(125123821035650511815462974l1*864117209865696499921839192706^(1/2)*sin((474989023199232*86411720986569649
9921839192706^(1/2)*t(i))/3475661745774987852464992021))/546311479004995234859502178837735488566067200))/1042
698523732496355739497606 3);

    else
        wd(i)=12*(1/12 -
(868915432848014719320064000*exp(-(329021251066317476188032860 16*t(i))/3475661745774987852464992021)*(cos((47
4989023199232*864117209865696499921839192706^(1/2)*t(i))/3475661745774987852464992021) -
(125123821035650511815462974 11*864117209865696499921839192706^(1/2)*sin((474989023199232*86411720986569649
9921839192706^(1/2)*t(i))/3475661745774987852464992021))/546311479004995234859502178837735488566067200))/1042
6985237324963557394976063);
    end
end
figure(2)
plot(td,wd)
grid on
xlabel('time (s)')
ylabel('voltage (volts)')
title('Theoretical Values of PID-Controller')

figure(3)
plot(td,wd,'b', td,win,'r', td,wot,'g')
legend('theoretical output', 'experimental input', 'experimental output', 'location', 'southeast')
grid on
xlabel('time (s)')
ylabel('voltage (volts)')
```

```
title('Theoretical Values of PID-Controller')

% t1=excel1(:,1);
% t3=exceld(:,1);
% subplot(2,1,1)
% plot(t1,.012*w1)
% grid on
% xlabel('time (s)')
% ylabel('voltage (volts)')
% title('Lab 1 plot')
% subplot(2,1,2)
% plot(t3,wot)
% grid on
% xlabel('time (s)')
% ylabel('voltage (volts)')
% title('Lab 3 plot')
% axis([0 12 -2 2])
```

```
TV =

1/12 -
(868915432848014719320064000*exp(-(329021251066317476188032860l6*t)/34756617457749878524649920 21)*(cos((4749
89023199232*864117209865696499921839192706^(1/2)*t)/34756617457749878524649920 21) -
(125123821035650511815462974 11*864117209865696499921839192706^(1/2)*sin((474989023199232*86411720986569649
9921839192706^(1/2)*t)/34756617457749878524649920 21))/546311479004995234859502178837735488566067200))/104269
85237324963557394976063
```

Published with MATLAB® R2016b

Comment Summary

Page 16

1. use legends.

Page 18

2. Show one complete wave forms for all three types of data. Otherwise it is hard to see the details of the waveforms.

Page 36

3. Please provide proper recommendations.