Temple University

College of Engineering

Department of Electrical and Computer Engineering (ECE)

Student Report Cover Page

Course Number: ECE 3412

Lab 8: PID control realization using analog components

Student Name: Devin Trejo; Robert Irwin

TUid: 914924557; 914980083

Due Date: 4/14/2015

TA Name: Michael Korostelev

Grade: / 100

I. Introduction

In this lab we will be working with an analog PI controller. This is important for understanding of this course because everything up to this point was done in hardware, MatLab was doing a lot behind the scenes that we were unaware of. This was part of the reason we were seeing such noisy outputs in previous lab. The noise came from the fact that the Arduino was getting interrupted while attempting to send commands. This resulted in the count that translated into the speed of the motor to be too high, which caused the noise. To avoid this, we had to build an analog circuit that would free up the Arduino to communicate without being interrupted. It was also important to view what an actual PI controller looked like in hardware. This method is also more robust. If we look at Figure 1, shown below, we can see how we can achieve the proportional and integral gains using analog components. We also note that a change in the value of the components changes the value of the gains.

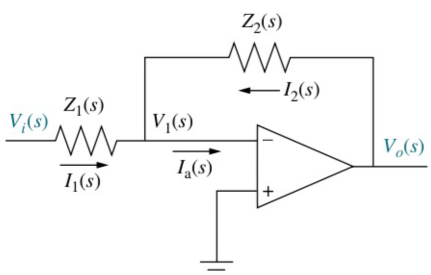

Figure 1: Op-Amp for transfer function realization

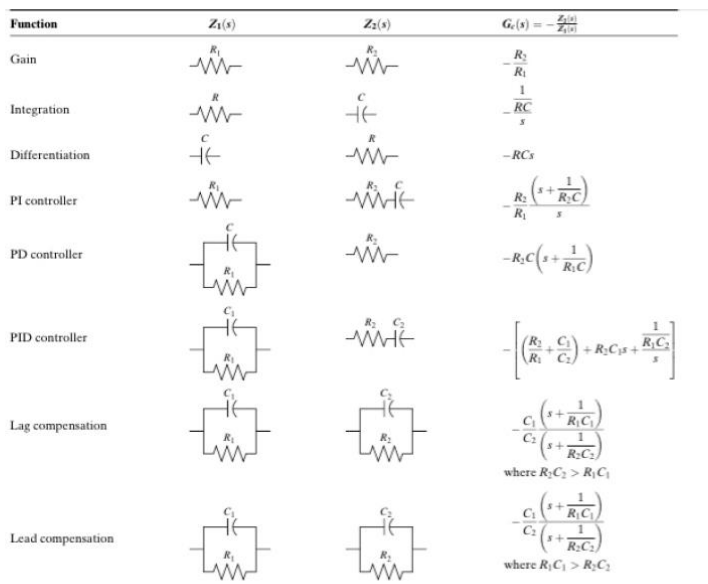

Figure 2: Analog Control Blocks Equivalent

II. Procedure

The majority of the lab is spent building the Microelectronic circuit that will allow us to avoid using the Arduino as the control and data collector. The circuit is laid out below:

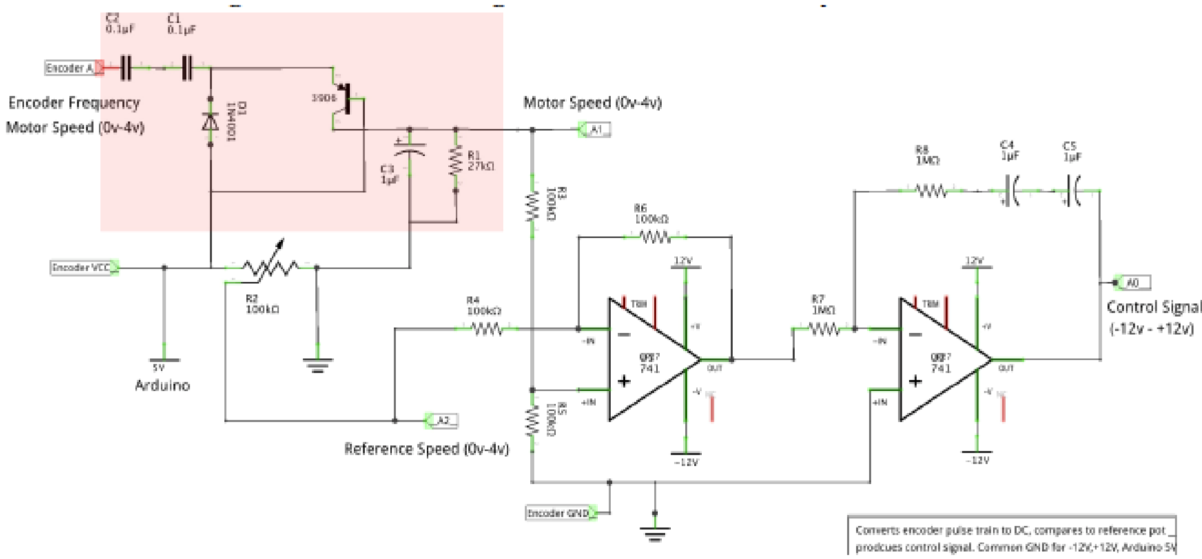

Figure 3: Microelectronic F-V Converter with PID Control

There are three major circuit parts for this circuit. The first thing we take notice of is the frequency to voltage converter circuit (F-V). The encoder which is attached to the motor shaft records speed into different pulse widths. The challenge is to convert this PWM signal into a voltage reference we could use to record the speed of the motor.

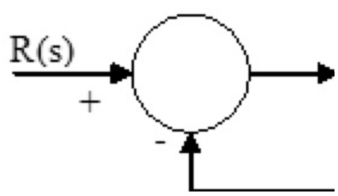

Figure 4: Difference Node

After we have a voltage reference that tells us the speed of our motor we are able to compare this voltage to a reference point set by a potentiometer. We perform this by using a difference amplifier. This amplifier is equivalent to the difference node (see Figure 4) in the block diagram we seen in previous labs which produces our error signal.

Now that we have our difference or error signal we can use a PID controller to feed into the motor. The equivalent circuits for this PID controller in an Op-Amp is seen in Figure 2. In this lab we will change the values of the capacitance and resistors to change our P, I and D gains.

The lab focuses on coming up with an understanding of how this analog circuit compares to the digital equivalent we made in previous labs.

III. Results

If we look at the circuit shown in Figure 2, we can see that the P value is 1 and the I value is 2. This comes from the fact that the

$$P = \frac{R_8}{R_7} = \frac{1M\Omega}{1M\Omega} = 1$$

$$I = \frac{1}{R_8 * (C_4 + C_5)} = 2$$

Now we will begin varying the components of the PID controller. Below we see the reference and speed of the system plotted against time. This corresponds to C4 and C5 being shorted, and R8 = 1MΩ.

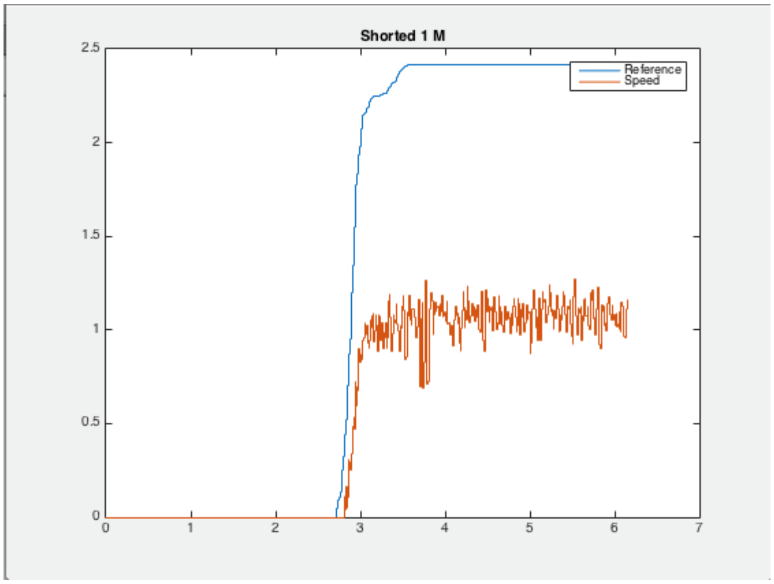

Figure 5: Reference and Speed for R8 = 1MΩ Shorted Capacitors

The figure above shows how we saw a large steady state error of approximately 1.5. The error is because when we short the capacitors, we get rid of the integral gain of the system. The proportional gain alone is not enough to diminish the steady state error. We now look at the control signal for the same combination of components.

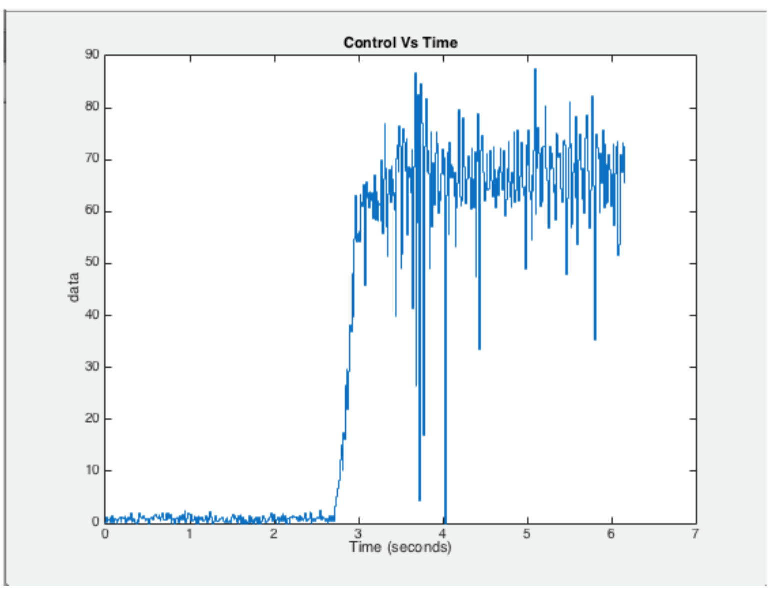

Figure 6: Control Signal for R8 = 1MΩ Shorted Capacitors

We notice that the control signal is very noisy, but still produced a very nice system. We now repeat the process for the same resistor while the capacitors are not shorted.

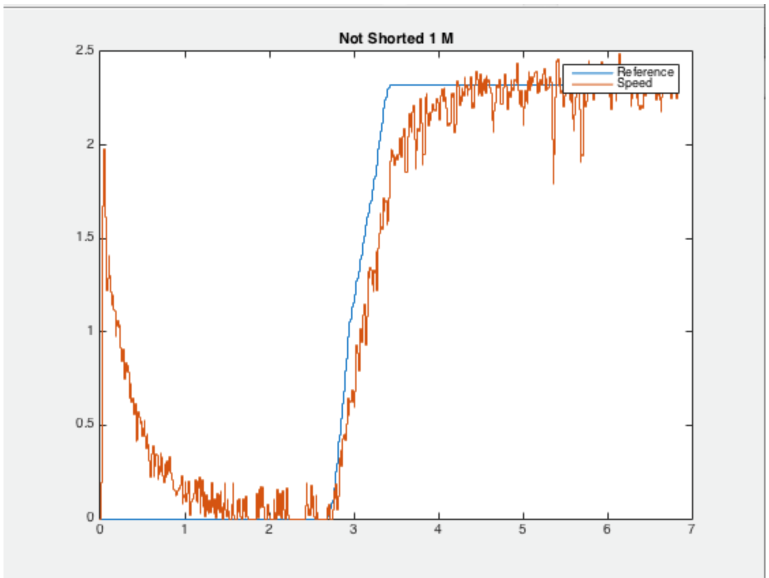

Figure 7: Reference and Speed for R8 = 1MΩ Capacitors NOT Shorted

With the integral gain in the circuit, there is no steady state error. The control signal is also affected by the integral gain.

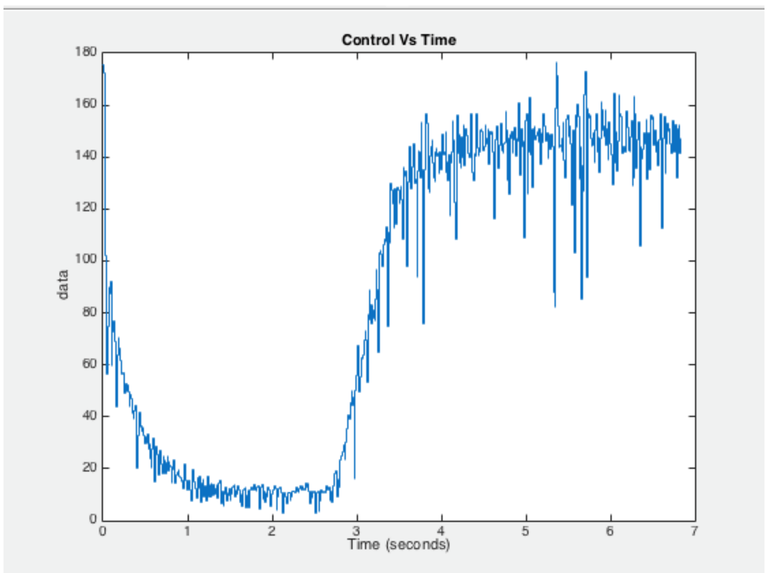

Figure 8: Control Signal for R8 = 1MΩ Capacitors NOT Shorted

Here we notice that the control signal is higher in amplitude that the first control signal shown in this section. Because the control signal is larger, the motor speeds up to the desired steady state value.

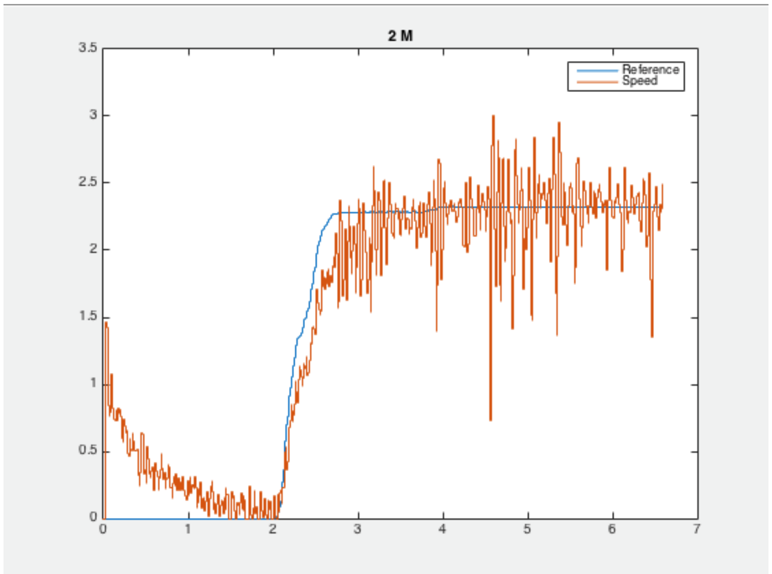

Figure 9: Reference and Speed for R8 =2MΩ

In this figure we see that the rise time is slightly faster, but the settling time and percent overshoot are much larger that the PI controller shown previously. Based on the configurations seen in Figure 1, we can see that if we increase the resistance, the proportional gain rises as R2 increases. The control signal is shown next.

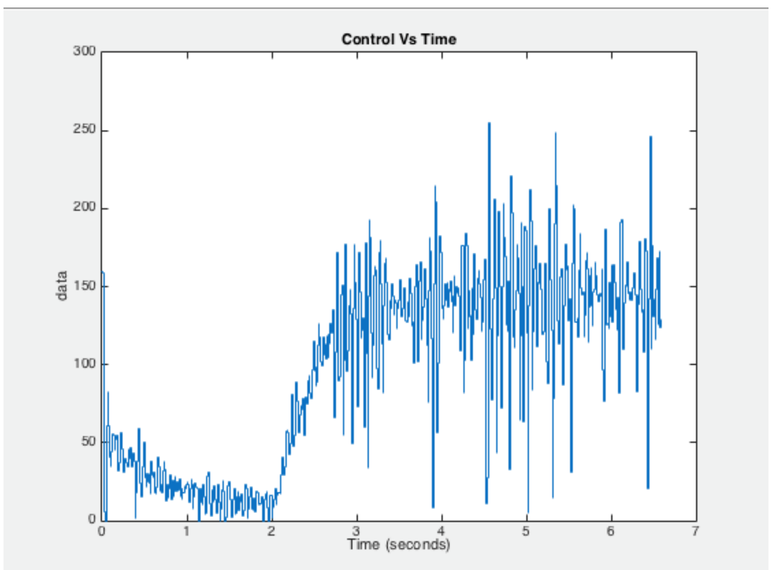

Figure 10: Control Signal for R8 =2MΩ

Looking at this signal, we see there is large fluctuation in the signal. For this reason, the motor is also fluctuating in speed, which is seen in the figure above.

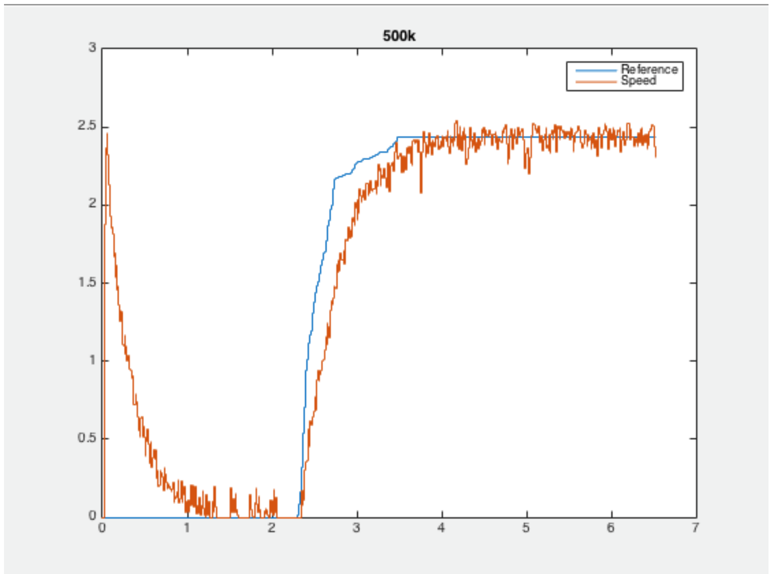

Figure 11: Reference and Speed for R8 = 500kΩ

In this figure, R = 500kΩ. As discussed above, a smaller resistance means a decrease in the proportional gain. For this reason, we see less oscillation in the output, but a slower rise time. The control signal is shown below.

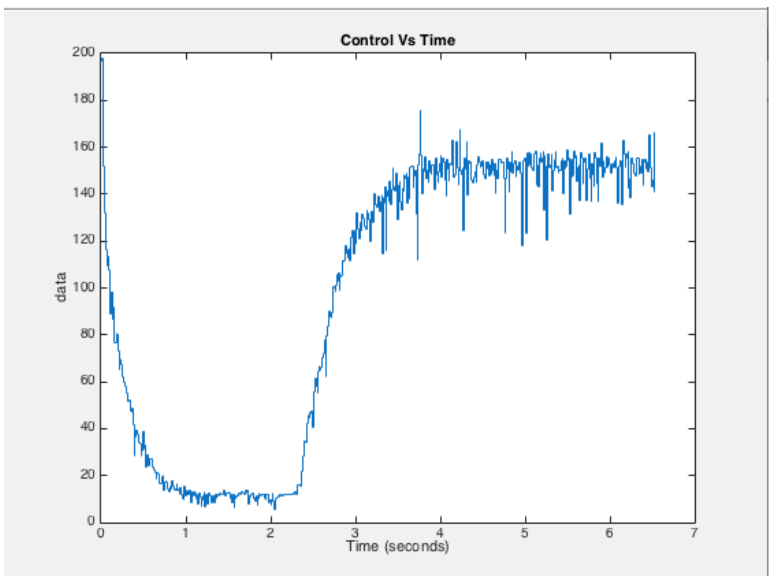

Figure 12: Control Signal for R8 = 500kΩ

We see less fluctuation in the control signal, which results in the less oscillatory output seen above.

Now we introduce a capacitor in parallel with R8. If we look at Figure 1, we see that this will introduce a derivative gain. The output is shown below.

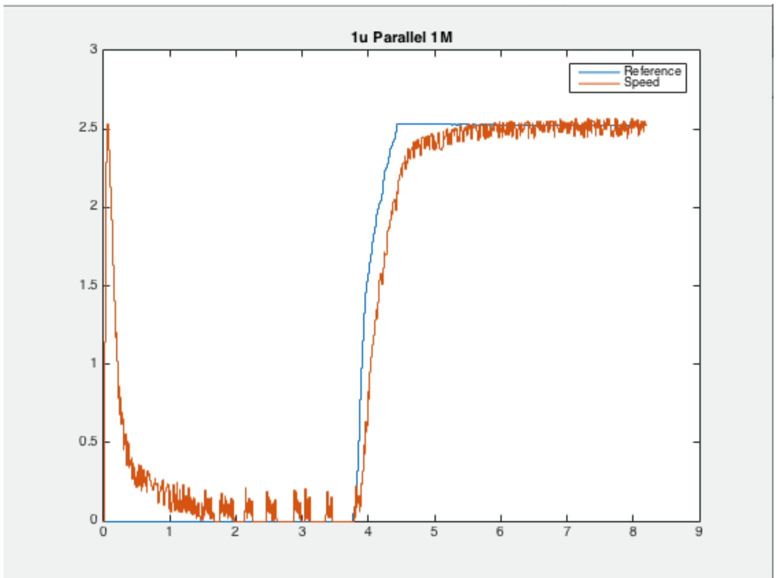

Figure 13: Output for PID Control

We see that with the derivative gain introduced, our output is much less noisy. From this, we can see that PID controllers are effective in obtaining a desired response. However, in noisy systems, derivative gains tend to amplify the noise of the system.

For the circuit described above we can calculate the P, I, and D, gain coefficients. From Figure 2 can calculate the P, I, and D gains.

$$P = \frac{R_2}{R_1} + \frac{C_1}{C_2} = 1 + \frac{1}{.5} = 3$$

$$D = R_2 C_1 = (1M\Omega)(1\mu F) = 1$$

$$I = \frac{1}{R_2 C_2} = \frac{1}{(1M\Omega)(.5\mu F)} = 2$$

To demonstrate further how to obtain specific PI values we practice an exercise with obtaining a P gain of 10 and I gain of 5. Since we only have a PI controller we only take case four from Figure 2. We start by obtaining values that produce an I gain of 5 since I gain is proportional to the resistors seen in P gain. The vice versa is not true. We take $C_2 = 50\mu F$.

$$I = 5 = \frac{1}{R_2 C_2} \rightarrow R_2 = 4k\Omega$$

$$P = 10 = \frac{R_2}{R_1} = \frac{4k\Omega}{R_1} \rightarrow R_1 = 400\Omega$$

In the circuit we built each op-amp provides a special purpose, which we have alluded to throughout out lab thus far. To reiterate the first op-amp is the difference amplifier that takes in the current speed of the motor and compares it to the reference. The result produces an error signal which is fed into our PI(D) controller. The second op-amp produces such controller by combination of resistors and capacitors. In our example above we demonstrated how to obtain specific P and I gains. We can produce P gain by using a ratio of resistors. The reason this produces proportional gain is that op-amps produce gain. Our error signal is too small to drive the motor directly thus we must amplify it to drive the motor. The ratio of the resistors determines how much gain we produce which is what we call proportional gain.

Separating the P gain from the I and D gain we observe how the other two use capacitors. The purpose of the capacitor is to produce either the derivative or integral of our error. Recall how in Laplace that multiplying by s is the same as taking the derivative in the time domain and how dividing by s is the same as integrating in the time domain. Our operational amplifier is using the capacitors to perform this function.

IV. Discussion

Lab 8 differs from other labs performed in that it was the first analog lab we performed. Previously we were only using the Arduino sample our speed, drive the motor, record data, act as our controller, and find the difference of our speed and reference. Overall all functions before this lab were done digitally using the Arduino. In lab 8 we combined analog components and our knowledge of Microelectronics to do the same function. The advantage of using this analog circuit to perform each one of these functions is that we never have issues with sampling. For every lab up until now we required use of the Arduino to sample the speed of our motor which was outputted by the encoder. We learned how the sampling of the speed was performed poorly by the Arduino since it requires constant interrupts of our main program. Our main program was finding the error signal, performing the appropriate gain, and driving the motor. There were too many function going on digitally that our data was always noisy. The noise resulted from the fact PWM signal was not always cleanly captured. The analog circuit has no problem performing all these functions since each component only has one role. We have an F-V portion of our circuit, a PID controller, a difference section. Another advantage of the analog circuit was the cost. Arduino is manufactured to perform many functions, while our analog lab setup can only read speed and server as a controller for our motor. If we change any portion of our circuit we would need to rethink our analog circuit. The Arduino on the other hand is easily reprogrammable to allow for slight changes in hardware.

The FV circuit we created in the lab allows us to convert our PWM signal into a voltage reference. The only way to find the error signal is by using a difference amplifier, but our Op-Amp does not accept a PWM signal as an input. The Op-Amp requires a voltage to compare with another voltage. The problem required us to find a way to convert PWM into voltage that is within a certain range. After we have our speed in terms of voltage we find our error signal. The only limitation to this lab is the controller. For sake of simplicity we combined all three controllers into one Op-Amp (as is seen from Figure 2). Simplifying the controller into one Op-Amp has its limitations in that changing one gain might change another. For example, we performed a calculation to find a P gain of 10 and I gain of 5. We needed to solve for I gain first since it has values that are dependent on P gain. Specifically R_2. Such problems do not exist in digital. The problem be easily solved by separating P, I, and D, but you can see how quickly the circuit can become complicated.

What we have showed is how analog is useful when design constraints limit you what you can do in digital. We had problems sampling our speed via our Arduino board which caused all our previous labs to have lots of noise in our data. The problem with analog is that it is harder to design since you are not simply telling your computer that you want a P gain of 10. Instead you have to pick discrete components that will produce your desired results.