Victor Garcia Flores
Jessica Jiang

# Lab 7 Report

## 1. Introduction

This lab is a supplemental lab for lab 3 using the same VFD closed-loop system found in lab 6. This lab was designed as a supplemental lab for lab 3, it was short and mostly used the same steps mentioned in lab 6. Information measured in DSPACE for the Lab 3 design was used to note motor response with a change in reference speed. Using this, a PID block was tuned for improved step time and reduced overshot/ripple. In previous labs, we worked with induction motors, but here the DC voltage is changed for motor speed. More details regarding the design procedure will be explained in this paper.

## 2. Design Procedure

Simulate lab 3 to calculate the function required for tuning the PID. Once simulated, calculate the function using the first-order transfer function to tune the PID in lab 6. Create a new folder named Lab 7 and copy and paste the file for lab 3 into the folder. Delete the matlab function and copy the PID created in lab 6 over to the new file. Use the calculated function shown in figure 2.1 to tune the PID as shown in figure 2.2. This step should be identical to the one done in lab 6 only using the formula in section 2.1 instead of the second-order transfer function.

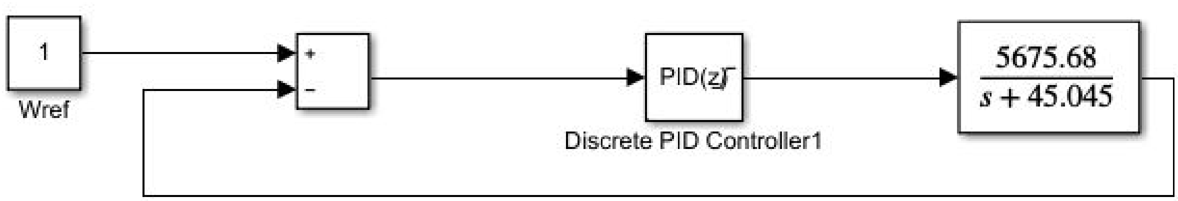



Figure 2.1 Function used to tune the PID.

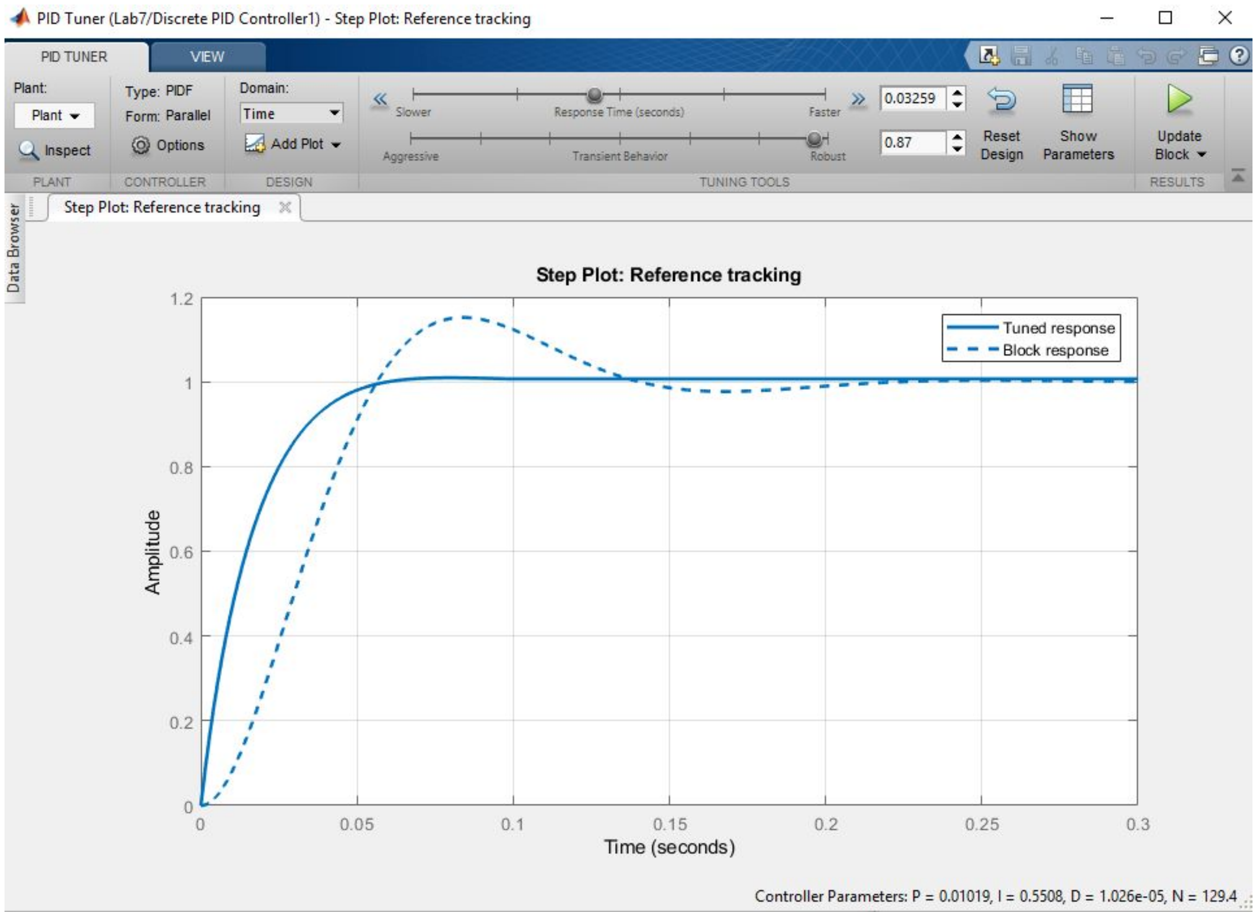


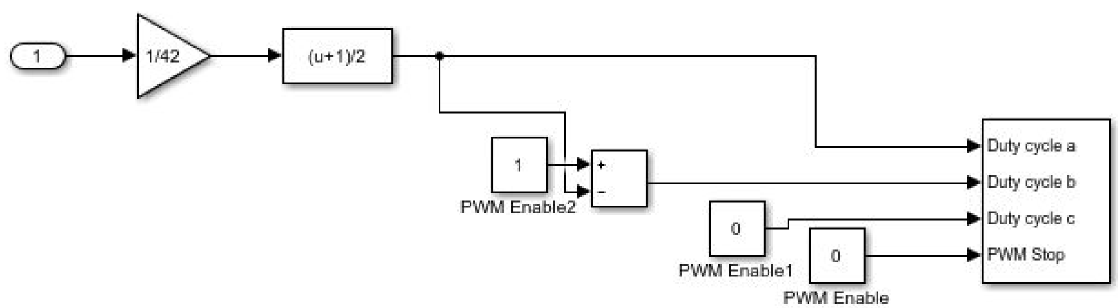

Figure 2.2 Tuning the PID

## 2.1 First-Order Transfer Function Derivation

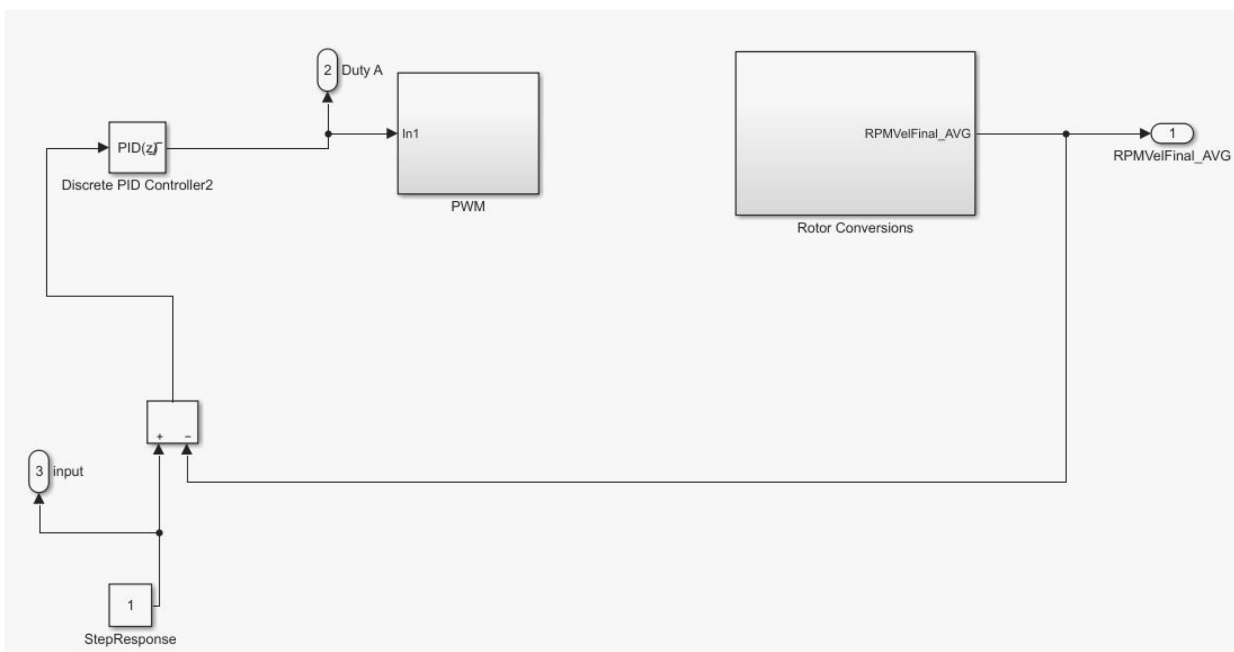
First-Order transfer function derivation is used when the graph looks logarithmic. $G(s) = \frac{\frac{K}{\tau}}{s+\frac{1}{\tau}}$, where $K = \frac{X_{ss}}{U_{ss}}$ and $\tau$ is the distance between the time where the graphs first start deviating to the time where the probed wave hits $0.63X_{ss}$. The reason a first-order transfer function is used here is due to the fact we are using a different motor. In this lab, a DC motor is used instead of the usual 3phase AC motor in the previous labs.

## 2.2 Final Assembled System

As previously mentioned, the final system incorporates ideas from Lab 3 and Lab 6. After having tuned the PID block to output the desired setting time and overshoot, the block was inserted as an input to the PWM block. It is important to note that the voltage being supplied has a magnitude of 42 V, meaning that we needed to take account of this inside the PWM block. As one may remember, the PWM system only allows bounded values [-1,1] and not [-42,42]. To scale down values, a gain block was used; it is depicted below:




After having made this change, the final system looks as follows:




*Note: The value adjusted in DSPACE is the block labeled “StepResponse”. This directly corresponds to the reference speed we are inserting

# 3. Analysis

## 3.1 Issues Encountered

- The motor would spin too slowly: We needed to resolve the scaling
- Overshoot was too large: We needed to adjust the tune settings and sacrifice response time for some transient behavior.

# 4. Simulation Results

As shown below in figure 4.1, the simulated graph fell within the given parameters of ripple-free, 0.1 second settling time. The first time the graph was tuned, it still contained a slight ripple with a small overshoot, this required us to go back and return the PID to dampen the ripple.

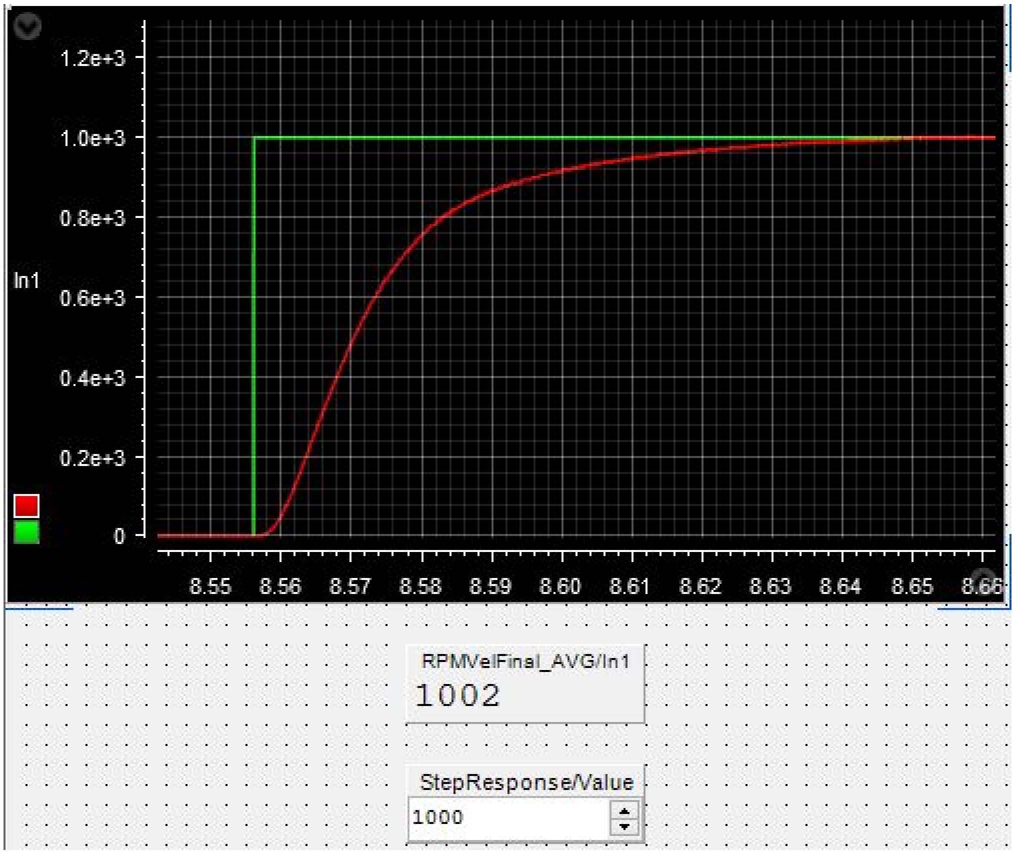


Figure 4.1 Final Simulation

## 5. Conclusion

This lab used the sum of all the previous labs to create a VFD that would take a voltage input and create a fast output that would not require multiple parameters as was done in lab 3 and lab 4. As mentioned in lab 6, the PID created a standard way of stepping up a motor that would give a smooth, mostly ripple-free curve. This standard is used in many industries and is very useful to know about when creating closed-loop systems.