Victor Garcia Flores
Jessica Jiang

# Lab 6 Report

## 1. Introduction

This lab was primarily designed to introduce the concept of a VFD in a closed-loop system. Aside from doing this, the lab also exposed everyone to the Proportional, Integral, Derivative (PID) block, which shortened the enable time, and also significantly decreased the ripple and overshoot value. Aside from this, the system is also different because of how the PWM control is altered. Rather than the usual bounded three-phase PWM, it is taken care of with our designed frequency block. More details regarding the design procedure will be explained in this paper.

## 2. Design Procedure

### 2.1 Frequency Block

This block directly corresponds to the PWM input that drives the motor. The arithmetic performed in this MatLab block is depicted below:

```
function v = fcn(f)

v = (f*.15)/42;

end
```

This function is primarily responsible for processing the PID block output and scaling the other PWM input corresponding to it. Note that this is also scaled/converted to be within -1 and 1.

### 2.2 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 the probed wave hits $0.63X_{ss}$ .

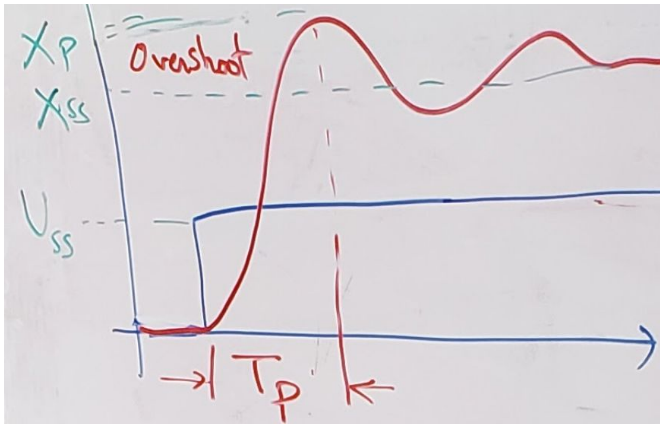
### 2.3 Second-Order Transfer Function Derivation

Second-Order Transfer Function Derivation is used when the graph has a large overshoot ripple before correcting itself. In this lab, the calculated graph would be a second-order transfer derivative can be found using the following equations:

- $K = \frac{X_{ss}}{U_{ss}}$

- $OS = \frac{X_P - X_{ss}}{X_{ss}}$ . Note that OS refers to the overshoot.
- $\zeta = \frac{-ln(OS)}{\sqrt{\pi^2 - [ln(OS)]^2}}$
- $W_d = \frac{\pi}{\tau}$

The graph used to interpret the necessary values is below:




## 2.5 Final Assembled System

In order to determine the final system, the transfer function for this system needed to be determined. To do this, we re-used simulation results from the previous lab and compared the desired speed with the actual speed. Doing this in DSPACE allowed our team to determine our values for $\zeta, X_{ss}, U_{ss}, X_p,$ and $\tau$ . Once all these were determined, the function G(S) was assembled and simulated to test for minimum overshoot and step time.

After having determined the second-order transfer function G(s), a closed-loop system was created so that we could tune the PID block. This small system took the difference between the reference and output speed, fed the output to the PID block, and then connected to the transfer function. After having done this, we interacted with the PID block so that we could tune it to the desired behavior. The tuned result is depicted below:

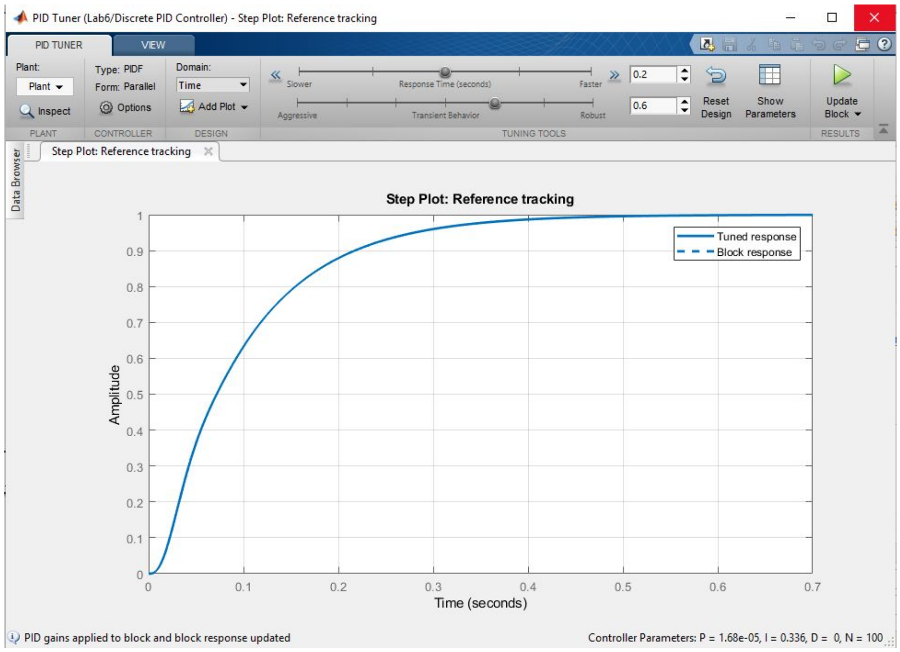



Figure 2.5.1 PID Tuning

As one may note, the output has prevented a large overshoot resulted in a very small setting time. This simulation was obtained using MatLab, but this is the expected result in DSPACE for the motor.

After having tuned this block, we copied it and connected it to the rest of our closed-loop system for this induction motor. The final system is depicted below:

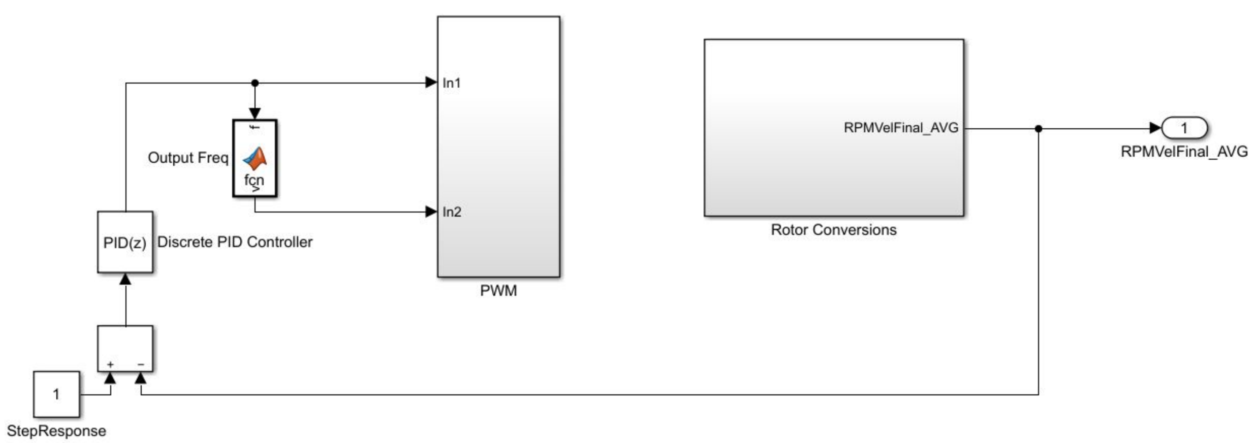
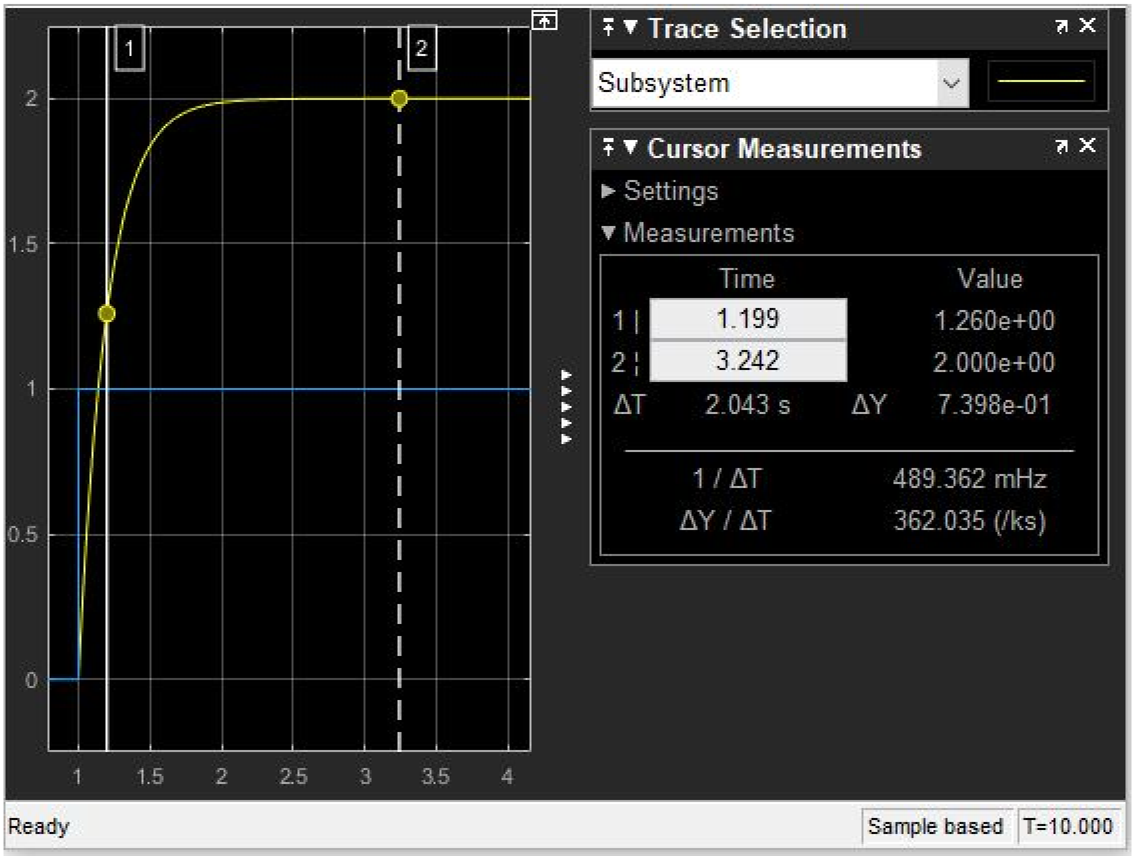



Figure 2.5.2 Final Simulink Design

# 3. Analysis

## 3.1 Practice Mystery Functions

The following mystery function shown in figure 3.1 uses a first order transfer function derivative. Using the formula provided in section 2.2, it is possible to calculate the end result shown in figure 3.2.



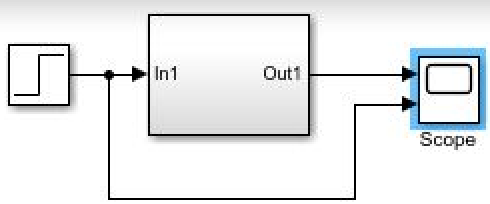




Figure 3.1 Mystery Function 1

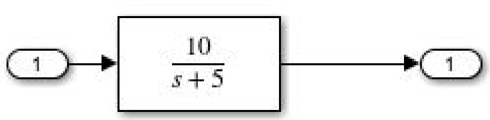



Figure 3.2 Mystery Function 1 Answer

The following mystery function shown in figure 3.3 uses a second order transfer function derivative as defined by the initial overshoot. Using the formula provided in section 2.3, it is possible to calculate the end result shown in figure 3.4.

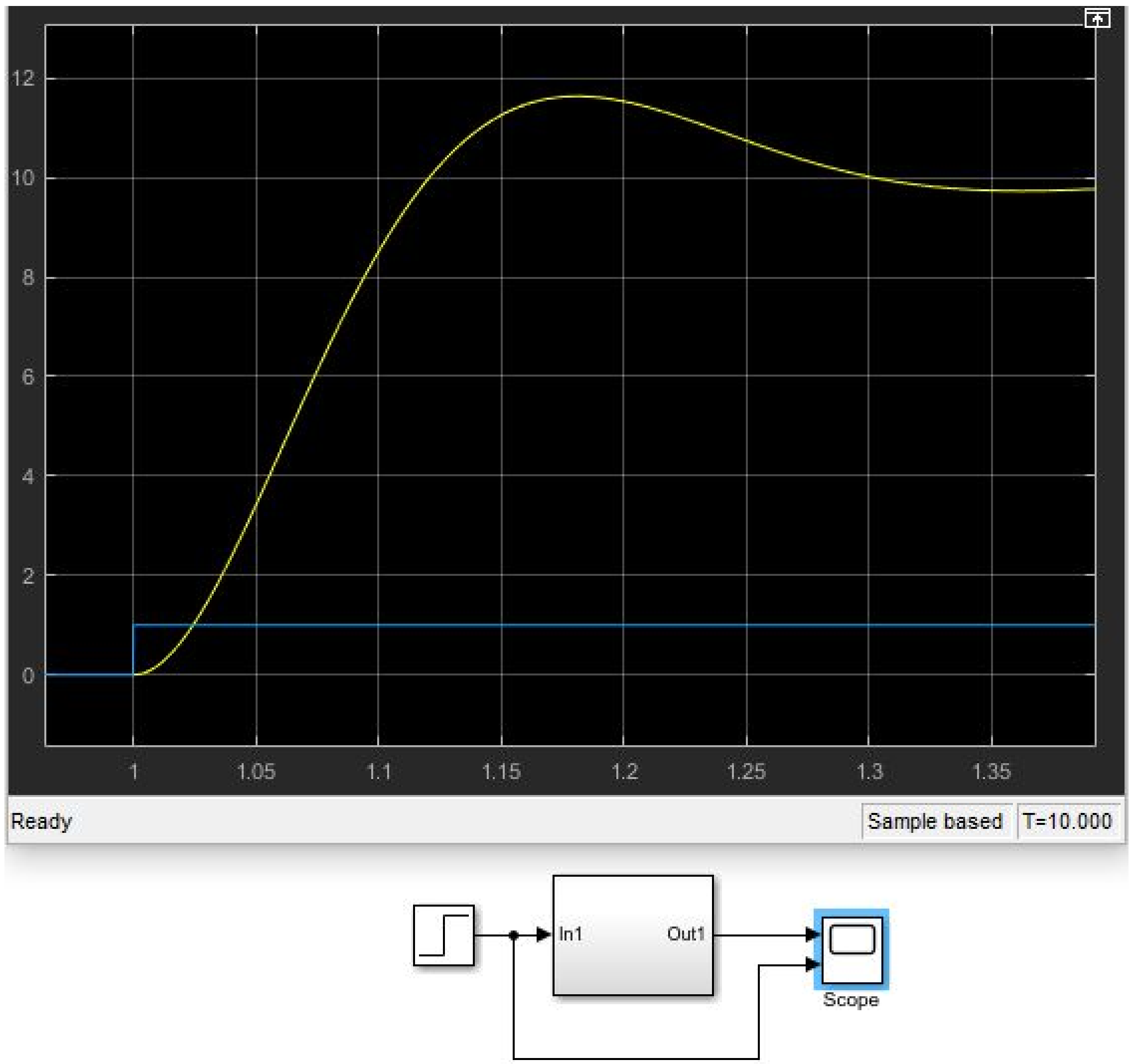



Figure 3.3 Mystery Function 2

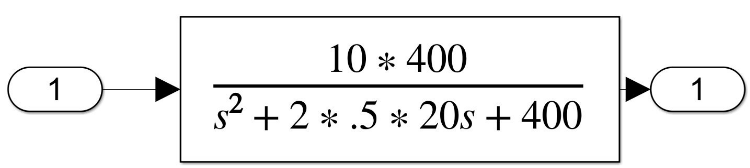



Figure 3.4 Mystery Function 2 Answer

As shown in the above graphs, the first-order derivative function has a rather linear and steady curve while the second-order derivative function starts with an overshoot or ripple before steading out. In this lab, the measured function of the motor has a ripple so the second-order derivative function was used to calculate the function needed to tune the PID.

## 4. Simulation Results

Tested Settling time for the step inputs from lab 5 design gives the following table. This table is used to determine the type of function was implemented in lab 5.

| Speed (ω) | Time (ts) |
|---|---|
| 200 to 500 | 0.9 seconds |
| 500 to 900 | 0.9 seconds |

The graphs from lab 5 were used to solve for the function used to tune the PID. Due to time constraints, the final design was not tested in dSPACE after the PID design shown in figure 2.5.2 but should follow the second-order derivative function shown in mystery function 2.

## 5. Conclusion

This lab served as an opportunity to understand the different ways of motor control and their trade-offs. During this system, we also learned about the importance of using the PID system as a whole rather than the individual components. For example, we learned that using a p-type system does not respond well in steady-state conditions. Meaning that if the reference speed is the same as the actual speed, then the motor would shut down. The system as a whole is important and provides what most companies need by using it as a universal step-size incrementer.