Victor Garcia Flores
Jessica Jiang

Lab 5 Report

1. Introduction

This lab will be using the same closed-loop system previously implemented in lab 3 in conjunction with the VFD assembled in lab 4 to create auto-adjustable speed control for an induction motor. In the past, we have created a VSD system that allows the user to adjust speed through three-phase power using the H-bridge the lab board, but this lab builds beyond that. The first task was to create a Volts/Hertz adjustment system and then create a function that gives the rotor a steady-state ripple band of 15 RPM and a setting time of less than one second. Consistency for different speeds was not a requirement, and a faster set-up time was preferred. Meeting these requirements required a thorough analysis of how the difference between the reference speed and actual speed is processed and adjusted.

2. Design Procedure

Copy and rename lab 4 into a new lab 5 folder and relink the file parameters. Create a PWM subsystem that can receive two inputs voltage and frequency. To do this, use the same sub system created in lab 4 and insert a multiplication where the sine wave is imputed so that we can use the equation V/f = 1.5 as shown in the figure below.

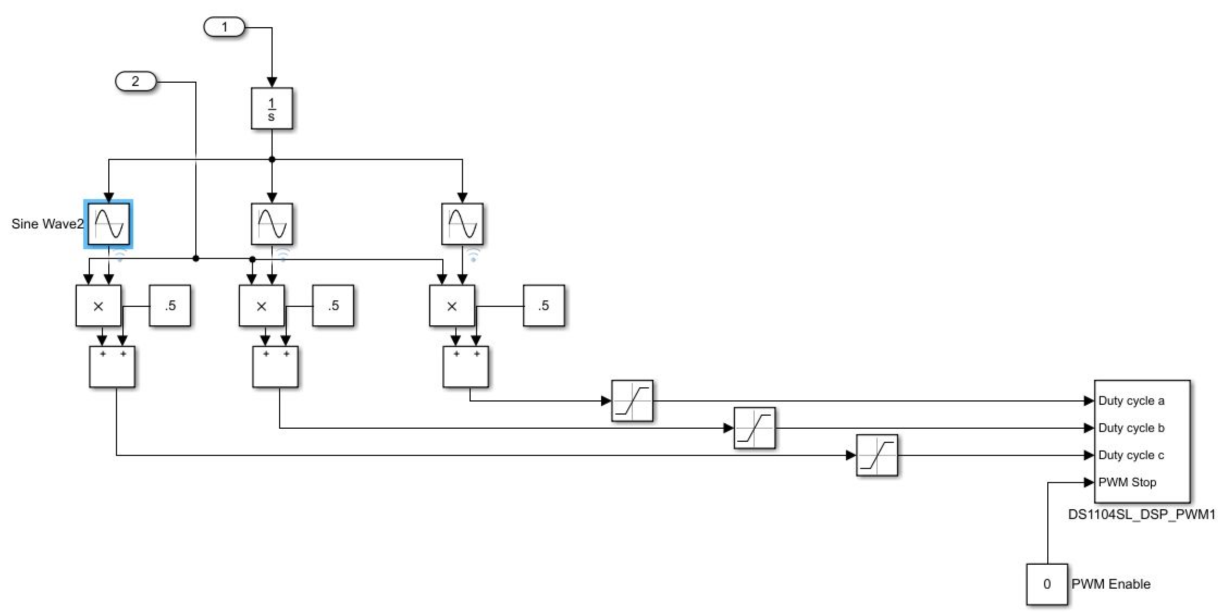

Figure 5.1a: PWM subsystem

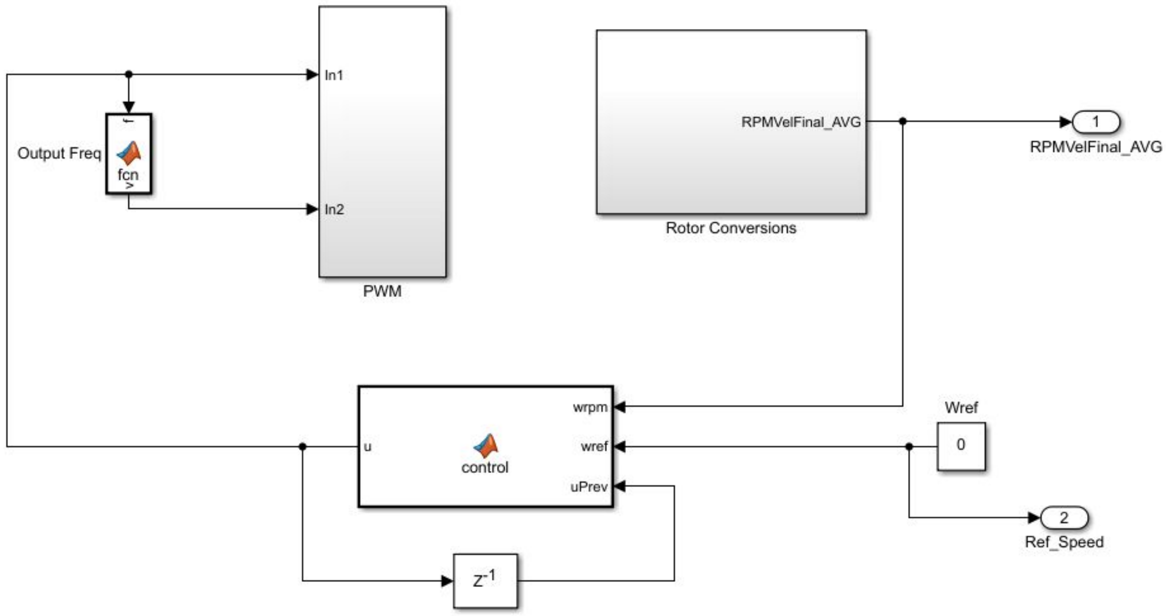

Figure 5.1b: Full subsystem

```
function v = fcn(f)

v = (f*.15)/42;

end
```

Figure 5.1c: V and F parameters

Insert a matlab function and set up the closed loop as done in lab 3. Create a function to link v and f by using the equation v = f*1.5/42 because V is between -42 and 42 and -1 and 1 so that only one output from the matlab function is required (Figure 5.1c). The setting time has to be between one second and the steady state error has to be around zero with the ripple band only having an error of around 15 RPM. To do this, the matlab code is separated into several parts. When the error margin is large, the slope is increased, but when the margin is error, the slope of change is decreased so that the ripple will be smaller.

```
function u = control(wrpm, wref,uPrev)
    err = wref-wrpm;
    du = 0.001;
    if err > 100
     u = uPrev+du*2;
    elseif err > 1
     u = uPrev+du;
    elseif err < 100
     u = uPrev-du*2;
    elseif err < -1
     u = uPrev-du;
    else
      u = uPrev;
    end

    if u > 128
        u = 128;
    elseif u < 0
        u = 0;
    end
end
```

Figure 5.2 Matlab Function

2.1 Errors Encountered

1. Unable to get within the margin of errors: Lower the slope and increase the number of checks.
2. The average graph was shifted down to be below the desired speed. In other words, our actual speed was not centered around our desired speed.

3. Analysis

Before setting up the closed-loop function, input a frequency and record the output:

Frequency (f)	ω_{out}
20	600
25	744
30	900
35	1038
40	1200
45	1338

50	1498
55	1638
60	1780
65	1944
70	2088
75	2238
80	2380
85	2540
90	2682
95	2838
100	3000

Plot the matlab function using f = 20:5:100 to increment from 20 to 100 using increments of five. And every ω_{out} tested for the output graph shown below in figure 5.3.

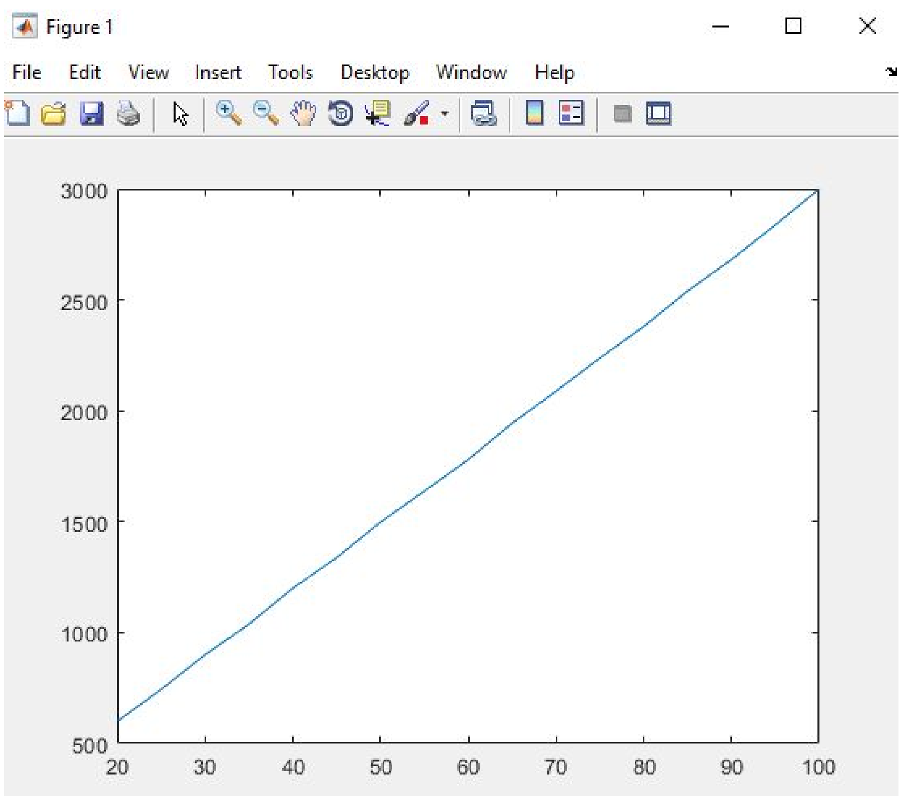

Figure 5.3 Slope of the output

As one may note, the frequency set is directly proportional to the speed; it is a linear relationship. Meaning that this relationship can be further analyzed to achieve the desired speed through a function. This relationship is then implemented in the closed-loop system to meet our needs. It is important to note that this does have a few discrepancies; it is not entirely linear, which can be noted in Figure 5.3.

4. Simulation Results

The final simulation results of this lab may be seen in Figure 5. This depicts the ripple magnitude and setting time of the designed system. The setting time achieved with this implementation is ~0.4 seconds, and a ripple magnitude of ~12 RPM. Although the ripple is close to the customer requirement, it still meets constraints.

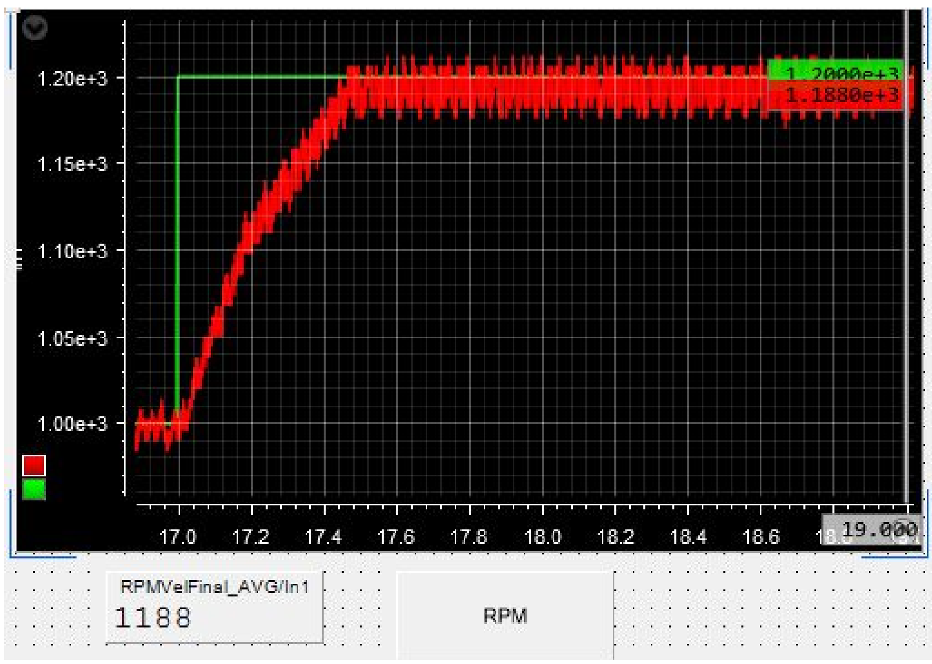

Figure 5.4 Final Output Graph

5.Conclusion

This lab integrated the information learned from the last two sessions into one lab. Here we used concepts ranging from VSD, VFD, and ASD systems to fulfill the task of creating a closed-loop system that sets to a desired speed by adjusting frequency. As one may recall, this lab implements 3-phase pulse-width-modulation to feed into the H-bridge. Aside from this, the lab also gave a more in-depth perspective on how motors worked, as well as how to modify the functions to tune the motor to respond more finely.