Victor Garcia Flores
Jessica Jiang

Lab 3 Report

1. Introduction

This lab served a great opportunity to build on top of what was put together in lab 1. In that lab, the goal was to get the motor to spin with an isolated DC power supply, and also measure position and velocity in RPMs and radians per second. In this lab, the goal was to assemble an H-bridge using the power electronics interface board. This allowed one to vary the motor's direction in dSPACE and MatLab. This method replaces having to use isolated DC supply because this method doesn't support the H circuit and only allows the motor to spin in one direction. As a whole, this lab allowed one to understand how an H-bridge works, and what happens when supplying the transistors variating base input values. Aside from this, it also allowed the team to understand closed-loop implementations in Matlab, and how to use dSPACE to analyze how the system behaves.

2. Design Procedure

2.1 Hardware

To set up the controller, plug in the PWM, adaptor, and power into the board as shown in *figure 2.1*. The PWM strip should be connected from the board to dSPACE Slave I/O PWM output and the power supply is connected to power and ground on the left side of the controller. Set the power to 42V and .03A and connect the motor to pins A1, B1, and COM ground. The Oscope should be connected to the the same pins to measure the input from the motor. A1 and ground for duty cycle A on the H-bridge, and B1 and ground for duty cycle B on the H-bridge.

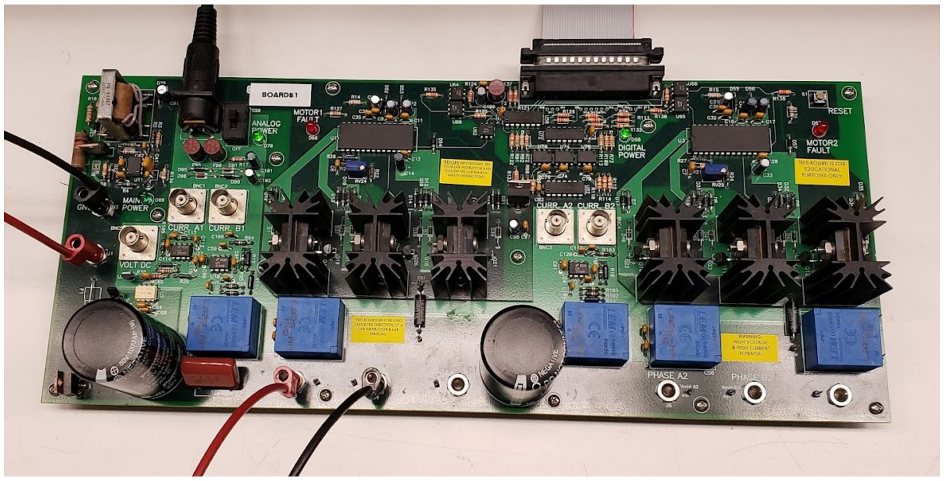


Figure 2.1: Controller setup

For more details on the configuration of the board, please address Figure 2.1.1. As one can see, the main difference between this and Lab 1 is that the power electronics board is used to drive the motor. Note that this is used for the on-board H-bridge.

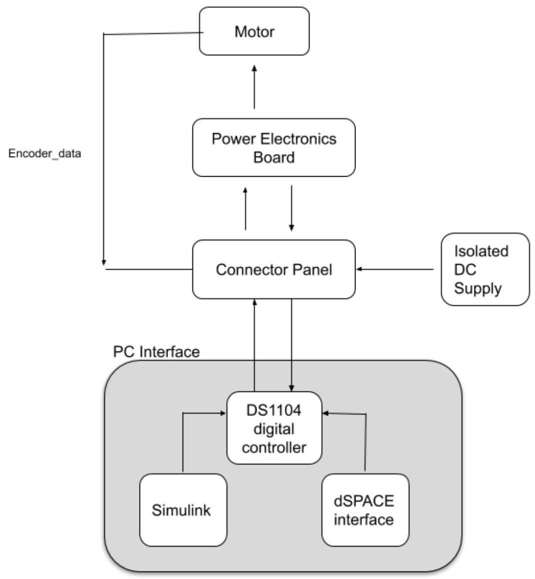


Figure 2.1.1: Hardware Block Diagram

2.2 Software

Create a Lab 2 folder and use the same Simulink file from the previous lab. Do the same for dSPACE, so that the files can be modified while still keeping the experiment results from the first lab. Keep in mind that doing so implies that there will be a new file used by dSPACE for when the simulink is flashed to the board.

2.2.1 Simulink (Non Closed-Loop)

This lab will be using the H-bridge modeled by the *DS1104SL_DSP_PWM3* to control the motor speed and velocity. Right click the H-bridge model and set the frequency to 20,000 Hz. To ensure that there are no problems, the duty cycle for A and B must add to 1 so when inputting values, it would be best to add a function that will ensure this. For this lab, values are only inserted into A. This means that the formula $1 - a = b$ will be used as shown in *figure 2.2*. As we are planning on outputting power to the motor, PWM stop will be set to zero. For this lab, we will not be using duty cycle C so it will be set to 0. Add an output for the raw data coming in Duty cycle A to ensure that there isn't a problem with the input going into the H-bridge and add in a moving average to the RPM result to smooth out the output so that the result is legible in dSPACE and does not jump all over the graph.

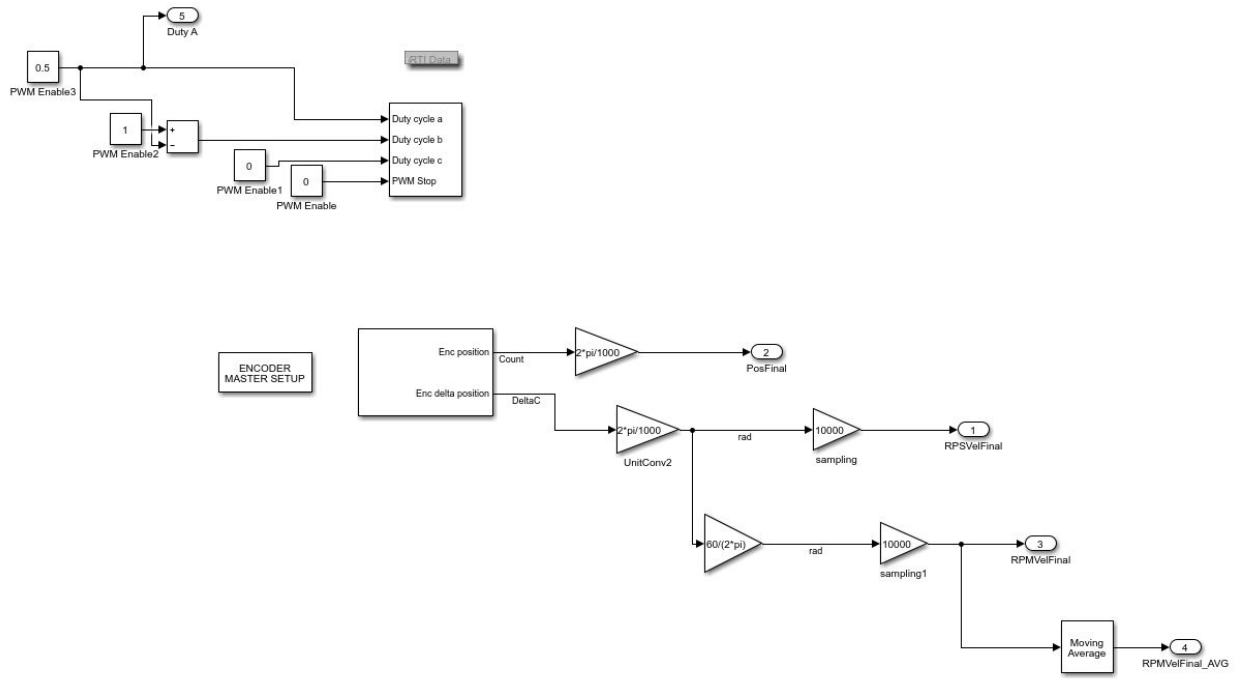


Figure 2.2 Simulink Model Schematic

2.2.2 Simulink (Closed-Loop)

The configuration addressed in 2.2.1 only addresses an open loop system. The figure below depicts the schematic in Figure 2.2 as a closed loop system. For neatness, this team chose to group the encoder blocks into a subsystem block named “Rotor Conversion” and simplified the PWM arithmetic into another subsystem named PWM. Since the goal is for the system to speed up to a value that a user sets, one chose to process the user input, compare it to the current output, and generate a PWM value to have the motor speed up to the desired speed.

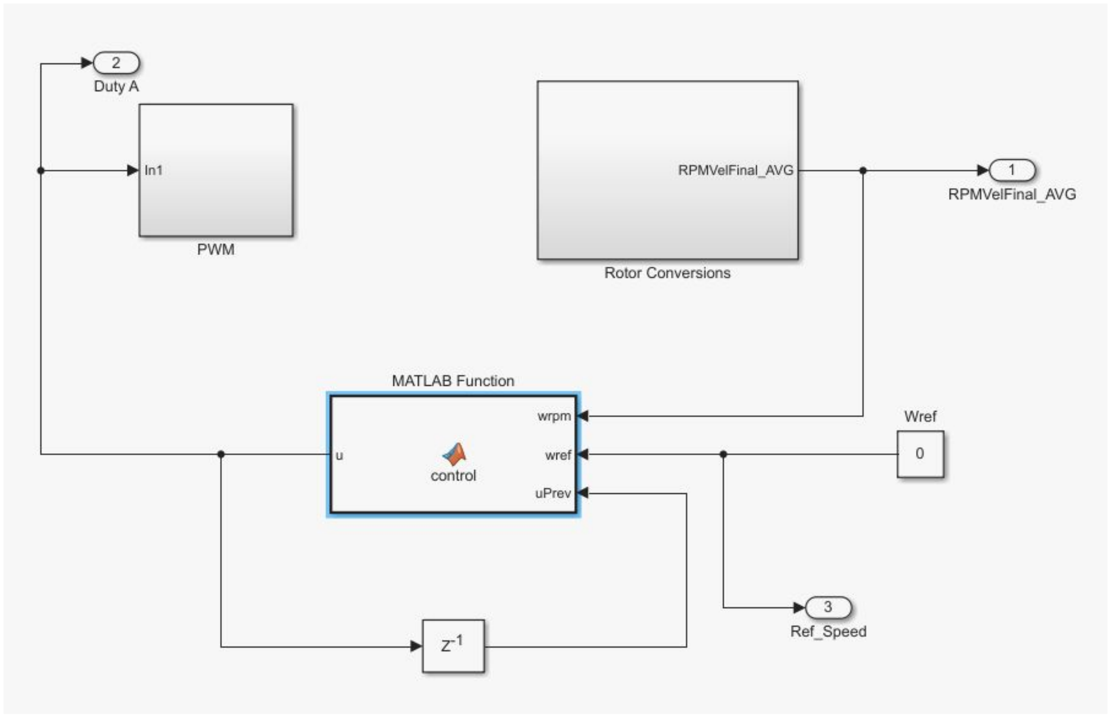
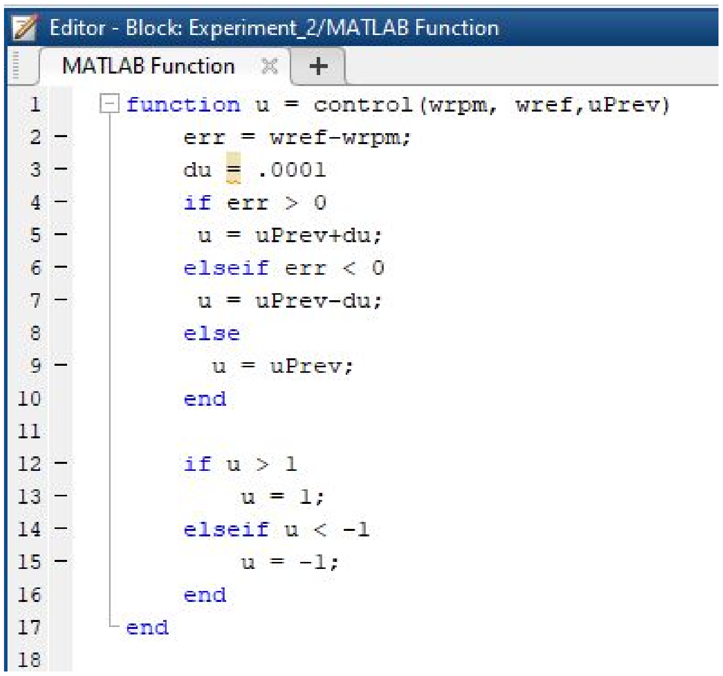


Figure 2.3 Condensed model

To generate a PWM value based on the user input and current speed, the MATLAB function block was implemented. It is important to note that the schematic above references Wref as the reference speed set by the user, and U as the PWM value. Knowing this allowed one to define variables in the MATLAB block and perform the necessary arithmetic to obtain the necessary values. As one can note in the image below, this team chose to find the difference between the desired speed and the current speed, and defined it to be ‘err’. This error then lets one determine whether the speed should remain the same, increase, or decrease. For example, if ‘err’ results as positive, then the block decides to increase its speed by incrementing the PWM value ‘U’. Conversely, if it is negative, then the system decreases its speed by decreasing the same PWM value “U”. Note that the system needs to know it’s previous PWM value for this to work; this is why the Z^{-1} block is used. Once it knows the new PWM value, then the PWM block processes the input and decides the motor speed.



```
function u = control(wrpm, wref,uPrev)
    err = wref-wrpm;
    du = .0001
    if err > 0
     u = uPrev+du;
    elseif err < 0
     u = uPrev-du;
    else
      u = uPrev;
    end

    if u > 1
        u = 1;
    elseif u < -1
        u = -1;
    end
end
```

Figure 2.4 Matlab function snippet

2.2.3 dSPACE

Remove the position plot from dSPACE and reload the .sdf file. Graph the results of the motor. One plot should still display RPS while the other displays both RPM and the average RPM. Both plots should display the current values of the output under them. To do this, go to *Instrument Selector* > *Display* and drag and drop the display into the windows. Then link the displays with their respective output variables.

Change the deadband in dSPACE to 1ms before adding the slider below the RPM and RPS graphs. The slider can be found in *Instrument Selector* > *Slider*. Drag the drop the slide and *Right Click* > *Instrument Properties* > *Range* to set the minimum range to 0 and the maximum range to 1. Then link the slider to the PWM Enable3 from Simulink so that the input of duty cycle A can be controlled by the slider. Use the slider to control the speed of the motor and record the findings.

2.3 Common Errors:

Motor not moving regardless of input: Check that duty cycle A and B are equal to 1. Also check to make sure that duty cycles A or B are not equal to 1 or 0.
Motor 1 fault light: Hit the reset button on the controller a few times.

3. Analysis

There was a lot to understand when supplying the H-bridge with different values. The configuration of an H-bridge in Figure 3.1 was set using the power electronics board. The configuration of this board is addressed in section 2. Note that the frequency measured in

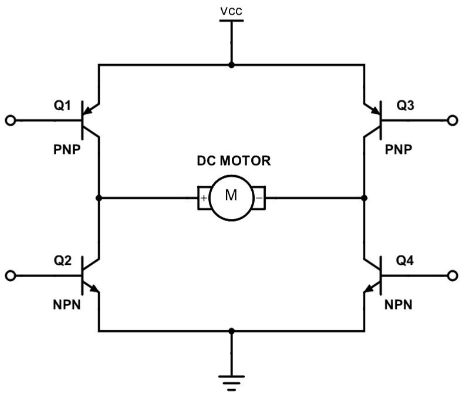
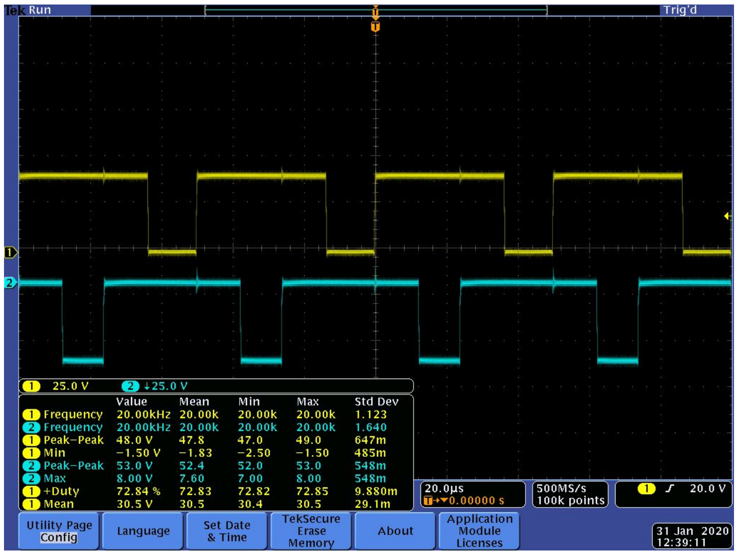


Figure 3.1: Circuit Representation of H-bridge

3.1 Making Motor Rotate in the Positive Direction

In this experiment, this team is controlling the inputs of Q1 and Q3. Doing so will yield different voltages across the motor, which can also change the direction. During testing, settings A and B to 1 and 0 did not make the motor spin, but this was due to the motor not operating in perfect/ideal conditions. Instead, settings A1 to 0.75 and B2 to 0.15 made the motor spin in the conditions depicted in Figure 3.1. The yellow lines in the capture capture corresponds to the potential across the positive side of the motor, and the blue corresponds to the potentials across the negative terminal of the motor.

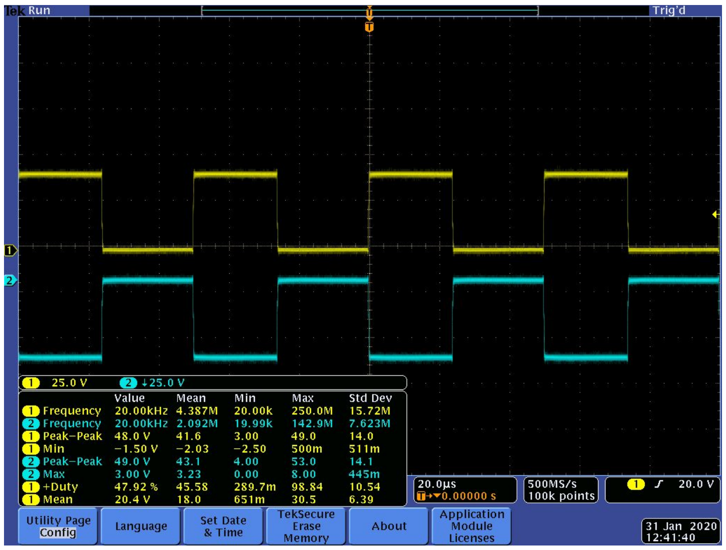


```
1  25.0 V      2 ↓25.0 V
                Value      Mean     Min      Max      Std Dev
1 Frequency    20.00kHz   20.00k   20.00k   20.00k   1.123
2 Frequency    20.00kHz   20.00k   20.00k   20.00k   1.640
1 Peak−Peak    48.0 V     47.8     47.0     49.0     647m
1 Min          −1.50 V    −1.83    −2.50    −1.50    485m
2 Peak−Peak    53.0 V     52.4     52.0     53.0     548m
2 Max          8.00 V     7.60     7.00     8.00     548m
1 +Duty        72.84 %    72.83    72.82    72.85    9.880m
1 Mean         30.5 V     30.5     30.4     30.5     29.1m

20.0µs            500MS/s         1 ʃ 20.0 V
T→▼0.00000 s      100k points

31 Jan 2020
12:39:11
```

Figure 3.1: Oscilloscope capture when $A = 0.75$ and $B = 0.15$.

3.2 Steady State of Motor (No Rotation)

When setting the inputs A and B to be the same, the waveforms were depicted in Figure 3.2. Setting these values to be the same makes the motor be stationary. The waveforms show that the voltages on both sides are inverted, which means that the motor sees a difference of zero; this means it can't spin because of the net voltage it sees.



	Value	Mean	Min	Max	Std Dev
1 Frequency	20.00kHz	4.387M	20.00k	250.0M	15.72M
2 Frequency	20.00kHz	2.092M	19.99k	142.9M	7.623M
1 Peak−Peak	48.0 V	41.6	3.00	49.0	14.0
1 Min	−1.50 V	−2.03	−2.50	500m	511m
2 Peak−Peak	49.0 V	43.1	4.00	53.0	14.1
2 Max	3.00 V	3.23	0.00	8.00	445m
1 +Duty	47.92 %	45.58	289.7m	98.84	10.54
1 Mean	20.4 V	18.0	651m	30.5	6.39

Figure 3.2: Capture When $A = B = 0.75$

3.3 PWM Conversion

During this lab, one had to change the range of PWM values given to the rotor from [0,1] to [-1,1]. In order to do this, this team chose to generate an equation to perform this equation. The equation implemented is as follows: $\frac{input+1}{2}$.

3.4 Motor Observation

Slider was adjusted and then manually moved by hand. When the slider is at 0 and the motor is at a standstill, the shaft of the motor was moved by hand. In figure 3.4.1there are spikes in motor speed when the motor is turned by hand. When the slider was set to 0.1 and the motor stopped by hand, the motor will stop. Figure 3.4.2 shows some stuttering and noise, but the line is centered at 0.

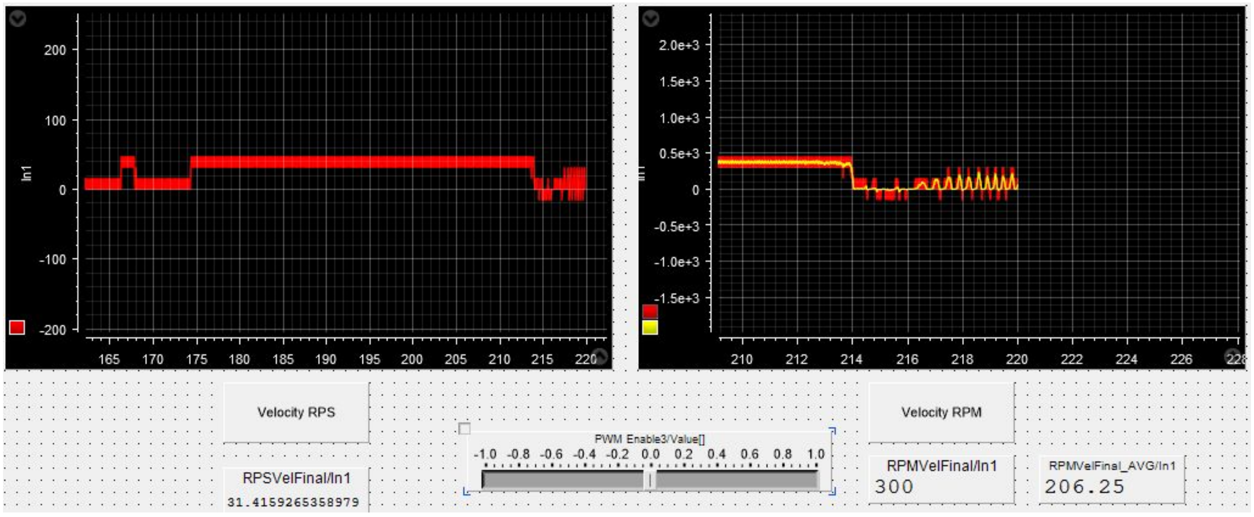


Figure 3.4.1: Duty is set at 0 motor is turned by hand

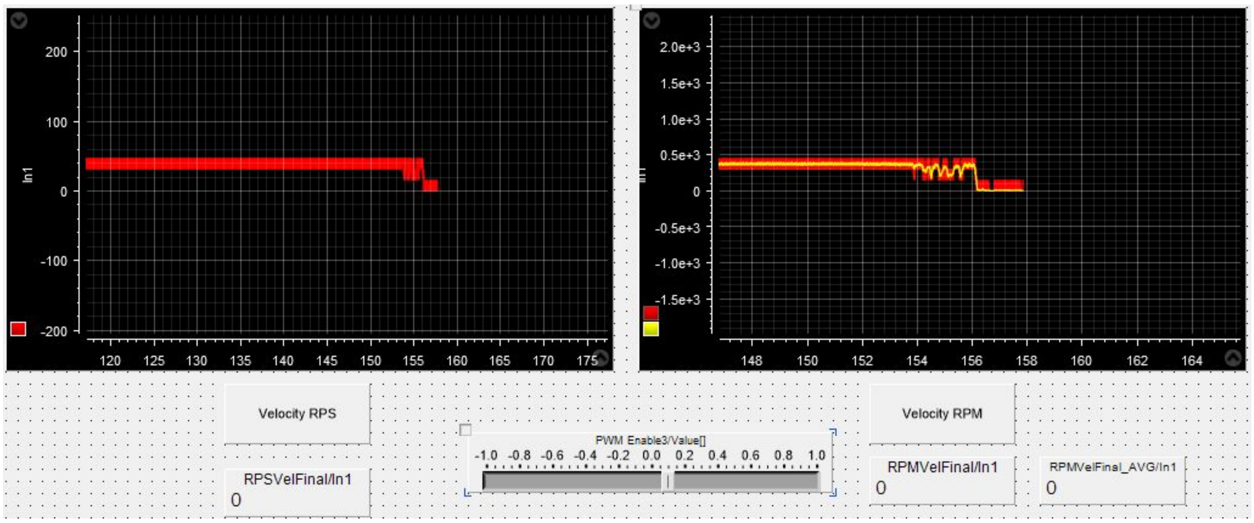


Figure 3.4.2: Duty is set at 0.1 and motor is stopped by hand

3.5 Closed Loop Reflection

After having assembled the closed loop system in MATLAB, one was able to simplify the schematic to a simple block diagram. During the lab one was able to learn that a closed loop is essentially a feedback loop, meaning that the output of the system affects the input in some way. In our case, the value U was the PWM value needed to get the Wref to the desired speed

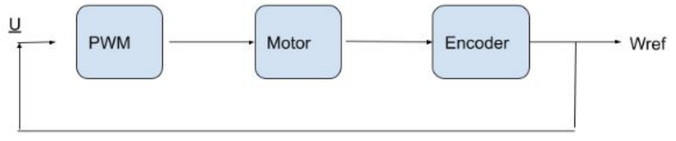


Figure 3.5

4. Simulation Results

Duty A	RPM
0.4 (<0.5 will give negative RPM [negative V])	-750
0.5 (=0.5 RPM will be 0)	0 (stop condition)
0.6 (>0.5 will give positive RPM [positive V])	750

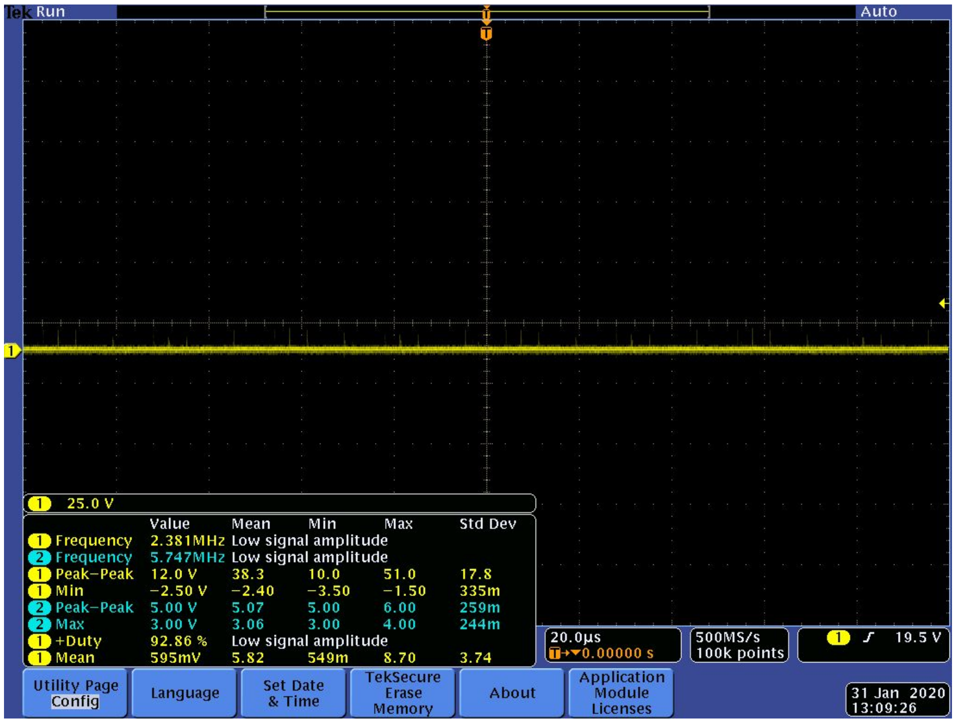
H-bridge rectifier Table

cmd	A	B	RPM	Mean Motor Voltage	G	V_{dc}
0.4	0.4	0.6	-750	-0.85	-0.2	-8.4
0.5	0.5	0.5	0	0	0	0
0.6	0.6	0.4	750	8.25	0.2	8.4

Formulas used for table:

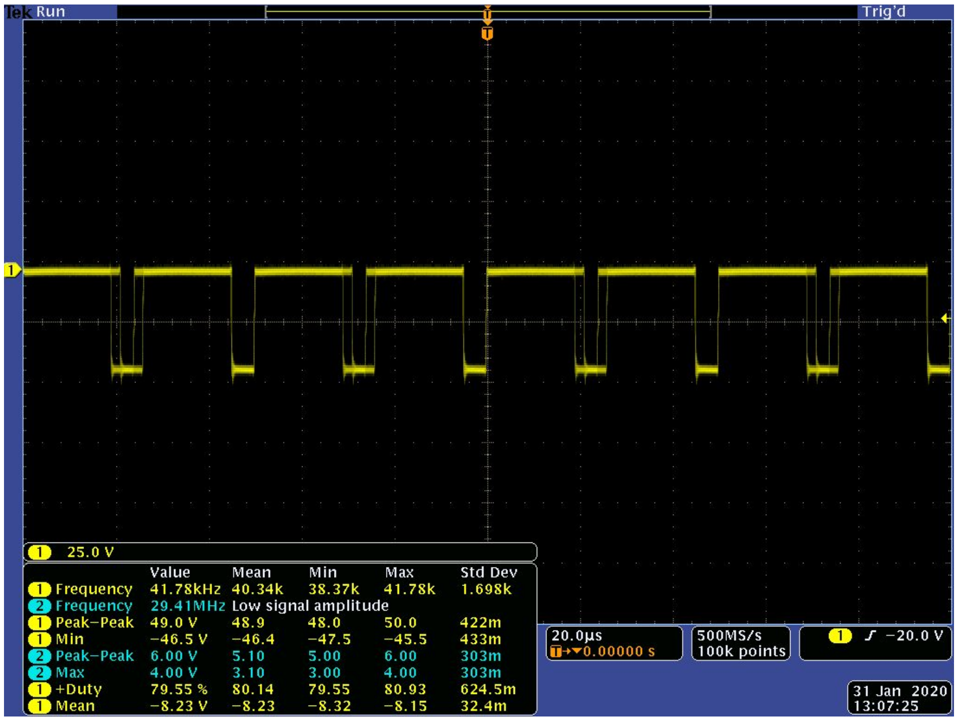
$G = (cmd-0.5)*2$

$V_{dc} = 42G$



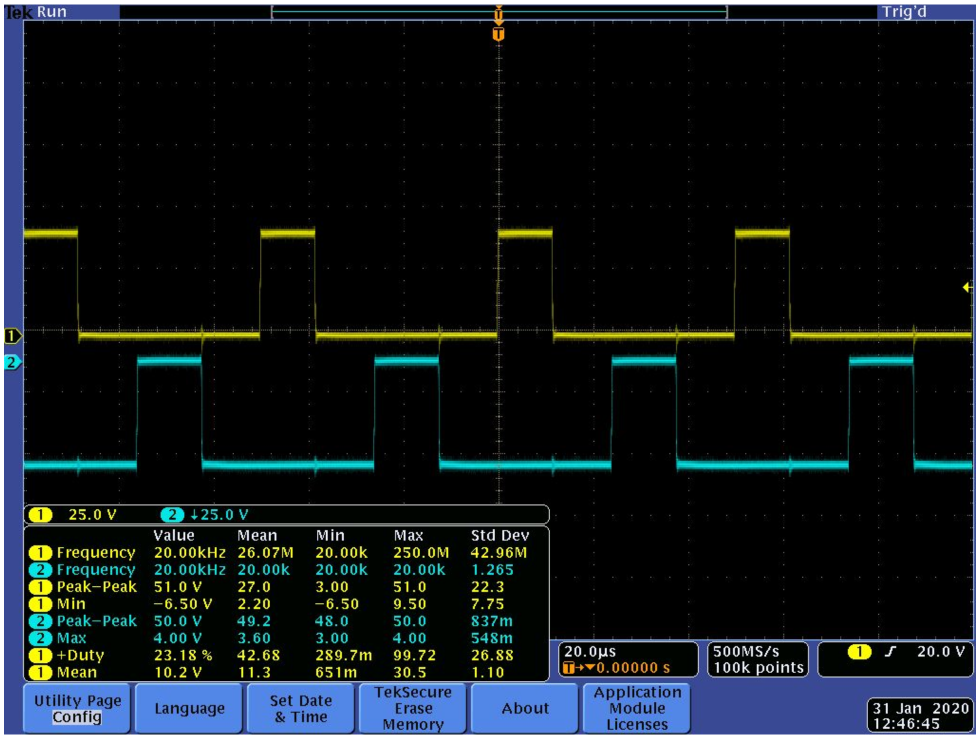
```
1  25.0 V
               Value     Mean     Min     Max     Std Dev
1 Frequency    2.381MHz  Low signal amplitude
2 Frequency    5.747MHz  Low signal amplitude
1 Peak−Peak    12.0 V    38.3     10.0    51.0    17.8
1 Min          −2.50 V   −2.40    −3.50   −1.50   335m
2 Peak−Peak    5.00 V    5.07     5.00    6.00    259m
2 Max          3.00 V    3.06     3.00    4.00    244m
1 +Duty        92.86 %   Low signal amplitude
1 Mean         595mV     5.82     549m    8.70    3.74

20.0µs            500MS/s       1  ƒ  19.5 V
T→▾0.00000 s      100k points

31 Jan 2020
13:09:26
```

Motor at rest



```
1  25.0 V
               Value     Mean     Min     Max     Std Dev
1 Frequency    41.78kHz  40.34k   38.37k  41.78k  1.698k
2 Frequency    29.41MHz  Low signal amplitude
1 Peak−Peak    49.0 V    48.9     48.0    50.0    422m
1 Min          −46.5 V   −46.4    −47.5   −45.5   433m
2 Peak−Peak    6.00 V    5.10     5.00    6.00    303m
2 Max          4.00 V    3.10     3.00    4.00    303m
1 +Duty        79.55 %   80.14    79.55   80.93   624.5m
1 Mean         −8.23 V   −8.23    −8.32   −8.15   32.4m

20.0µs            500MS/s       1  ƒ  −20.0 V
T→▾0.00000 s      100k points

31 Jan 2020
13:07:25
```

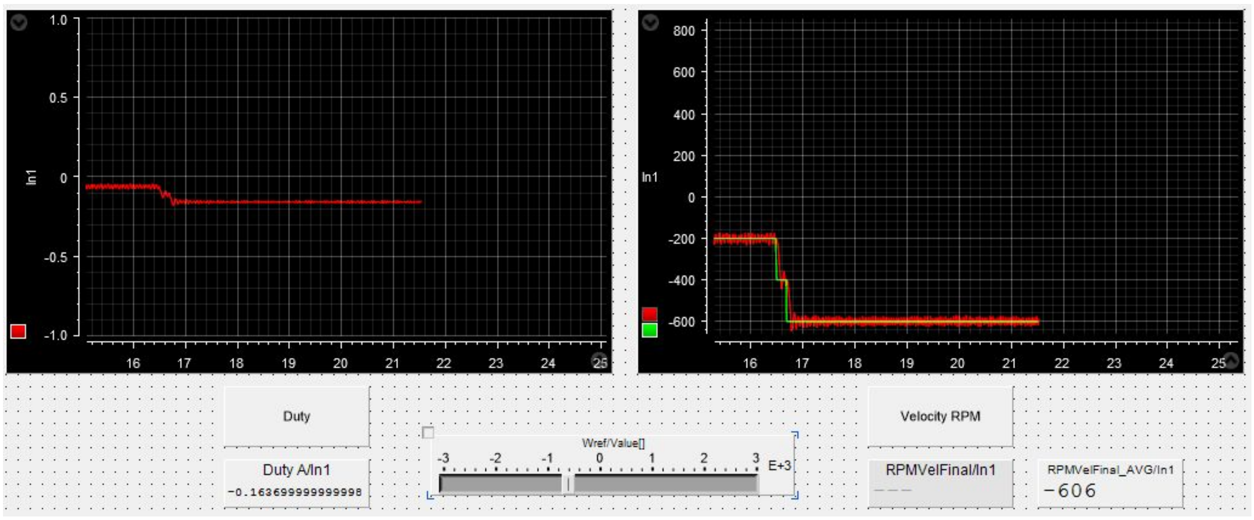
Motor at close to 1 duty cycle



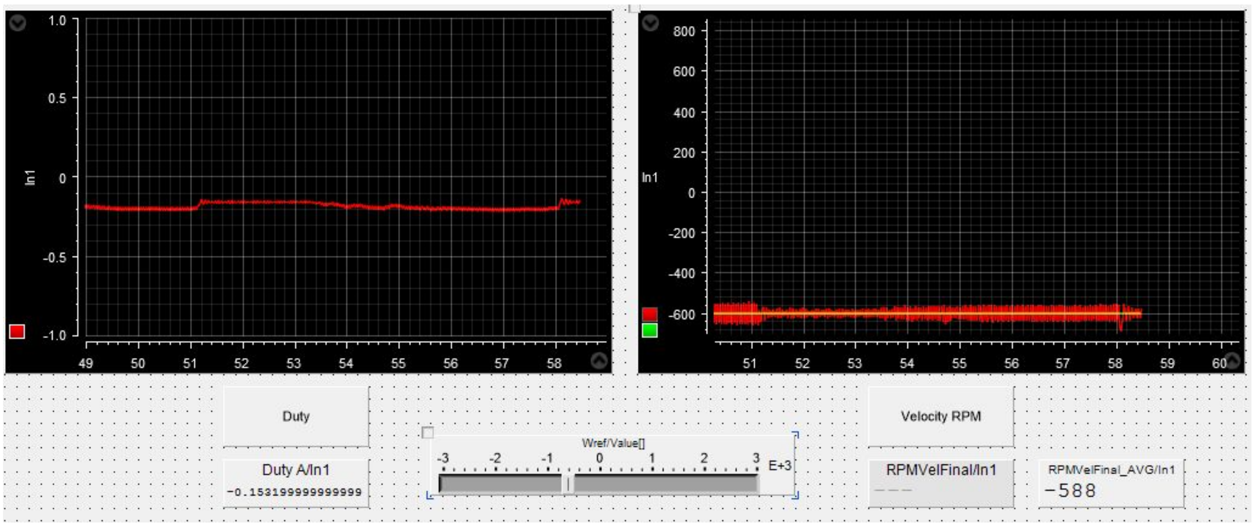
	Value	Mean	Min	Max	Std Dev
1 Frequency	20.00kHz	26.07M	20.00k	250.0M	42.96M
2 Frequency	20.00kHz	20.00k	20.00k	20.00k	1.265
1 Peak−Peak	51.0 V	27.0	3.00	51.0	22.3
1 Min	−6.50 V	2.20	−6.50	9.50	7.75
2 Peak−Peak	50.0 V	49.2	48.0	50.0	837m
2 Max	4.00 V	3.60	3.00	4.00	548m
1 +Duty	23.18 %	42.68	289.7m	99.72	26.88
1 Mean	10.2 V	11.3	651m	30.5	1.10

Motor at 0.75 duty Cycle

Closed Loop Testing:



Speed controlled by slider



Tried to change speed by hand, motor recorrected itself

5.Conclusion

This lab gave a basic overview of H-bridges and how they work. Depending on the input of the H-bridge rectifier, the motor can either give a positive RPM, a negative RPM or no response. Through the oscilloscope, it was possible to see the duty cycles being inputted into the motor for different speeds and modes of operation. In the next lab, we will be working on how to use signal conditioning and the controller with dSPACE and Simulink to create a dynamic model for the motor. The next portion of this lab will also mean that one will control a motor through a closed loop. What this means is that a portion of the output signal is sent to the input to reduce error and improve performance.