

Lab 2 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 varying base input values.

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 Oscopce 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.

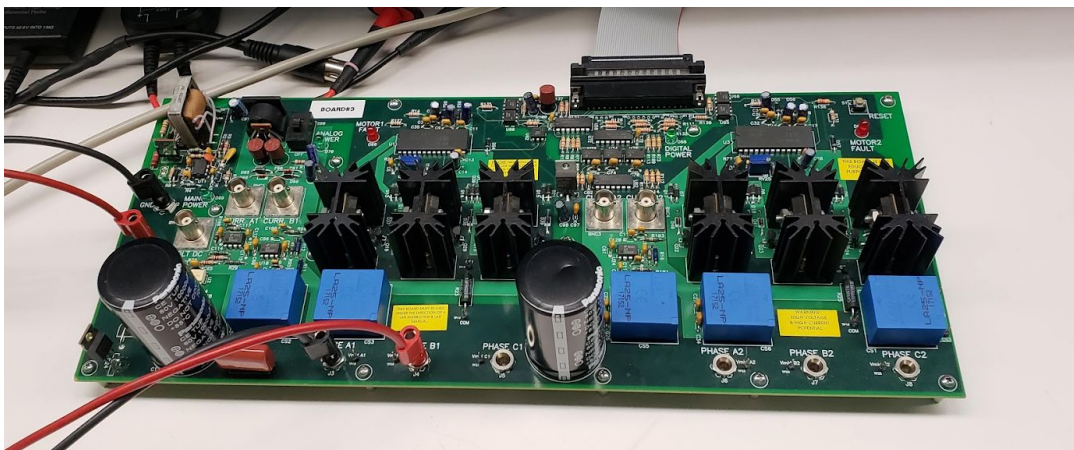


Figure 2.1 Controller setup

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

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.

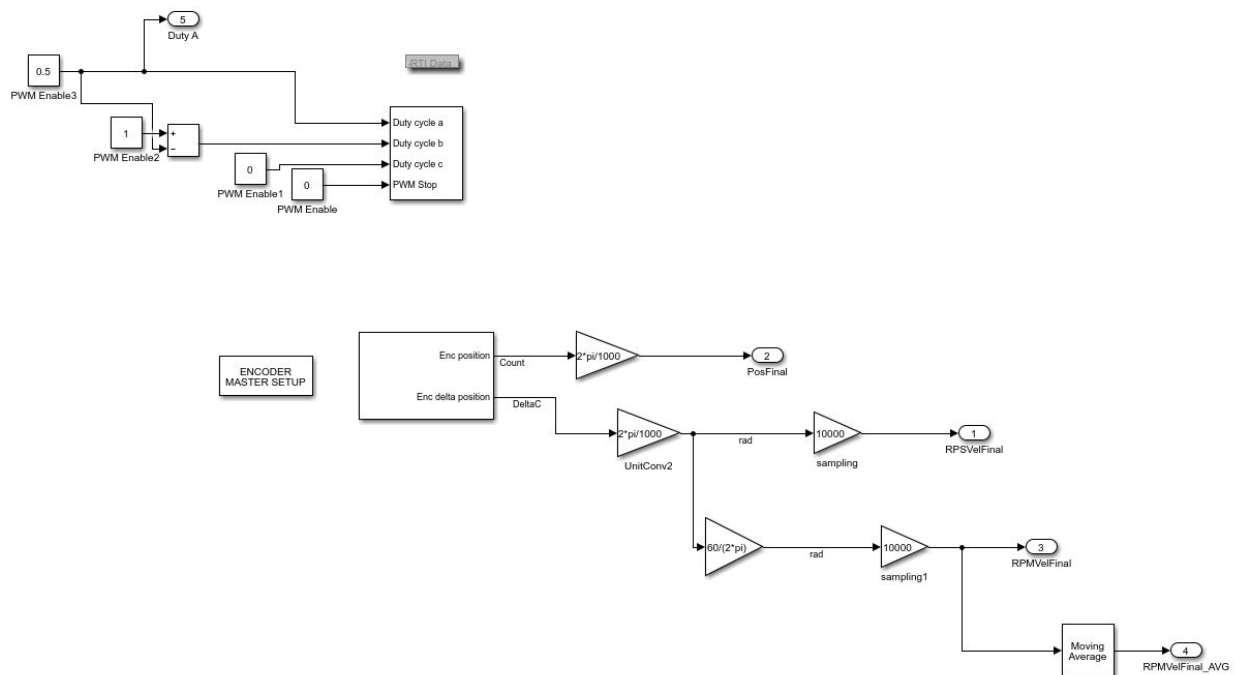


Figure 2.2 Simulink Model Schematic

2.2.2 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

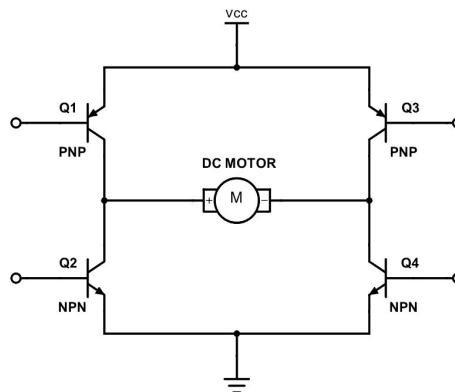


Figure 3.1: Circuit Representation of H-bridge

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 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.



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.

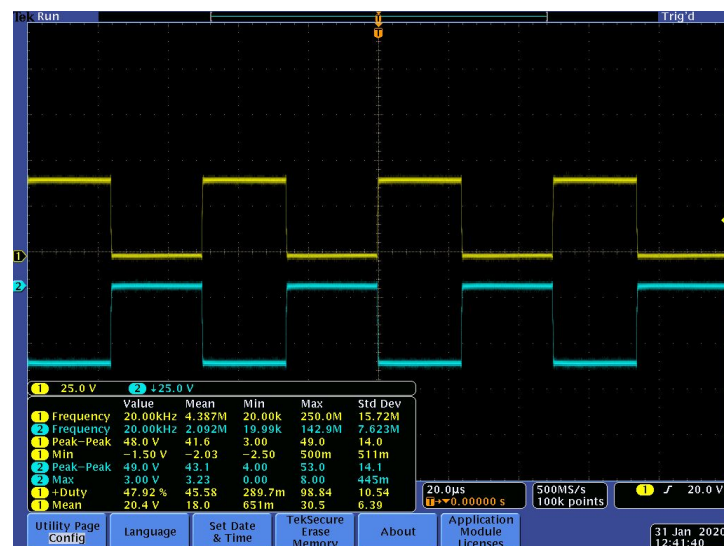


Figure 3.2: Capture When $A = B = 0.75$

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

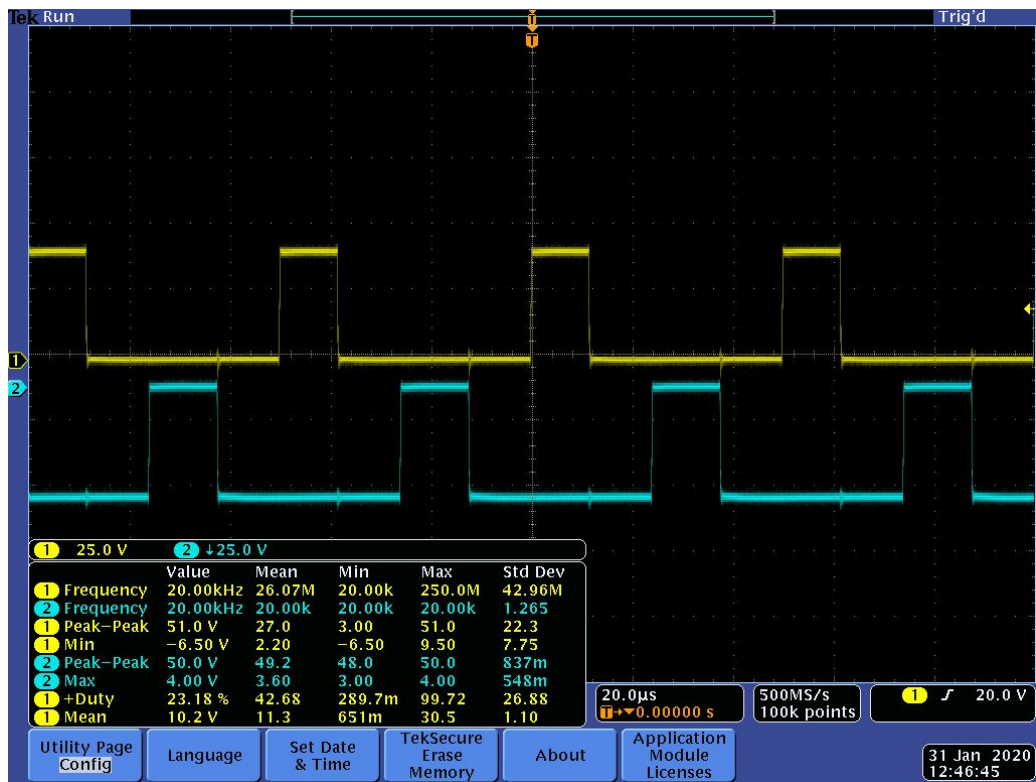
cmd	A	B	RPM	Mean Motor Voltage	G	V _{dc}
0.4	0.4	0.6	-750	-0.85	-0.2	-8.4
0.5	0.5	0.5	0	0	0	0
0.6	0.6	0.4	750	8.25	0.2	8.4

Formulas used for table:

$$G = (\text{cmd} - 0.5) * 2$$

$$V_{dc} = 42G$$

Screenshot of Testing Done Above:





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.