Spring 2017

Variable Frequency Drive (VFD) Control Lab

Clayton Eskridge
Montana Tech of the University of Montana

Mitchell Postlethwaite
Montana Tech of the University of Montana

Variable Frequency Drive (VFD) Control Lab

Members: Clayton Eskridge
Mitchell Postlethwaite
Mentor: Tom Moon

INTRODUCTION

A Variable Frequency Drive (VFD) is a device that uses a modulated DC signal to control a motor. This can be done in many ways including turning the knob on the front of the device or by using an external analog signal fed into the VFD.

The VFD (Figure 1.) takes a single-phase AC 120 Volt signal and first inverts the signal to DC and then modulates this signal. This is then split into three phases and fed into the three phases of an AC motor.

The speed and direction of the motor would be determined by an encoder system. This would then be turned into an analog voltage that could be read by the DAQ. This would allow Vissim to determine the speed of the motor and make adjustments as necessary.

Figure 1. Face of the Automation Direct GS2 Variable Frequency Drive.

We plan on controlling the VFD using Vissim which will output an analog voltage through a DAQ. We will tell the VFD which direction and what speed we want the motor to be spinning and will be monitoring it through an encoder wheel which will feedback through our system via the DAQ's inputs. A visual representation of this can be seen in Figure 2.

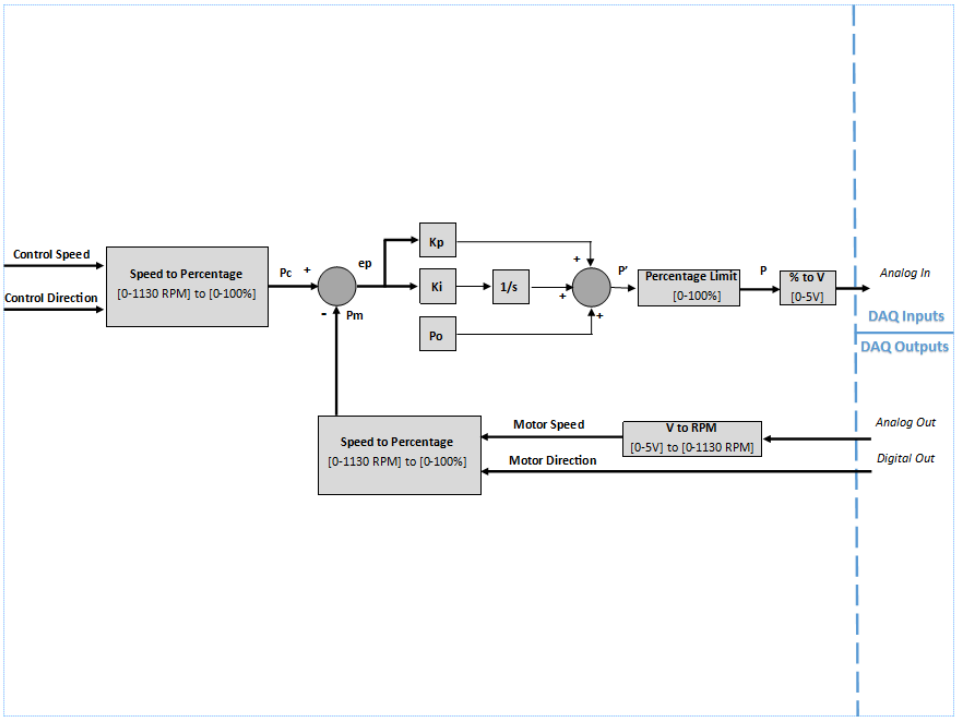


Figure 2. Block Diagram of feedback system in Vissim.

This system will feed into the VFD through an op-amp. The vfd will then control the motors speed and direction which will be read by an encoder chip. This encoder chip will be used to read the speed and direction of the motor using a D Flip Flop to tell direction and a frequency to voltage (F to V) chip to determine the speed. This will require a high pass filter located between the encoder and F to V to eliminate switch noise. The F to V chip would then feed into a voltage divider which will then go back to the DAQ to be read in Vissim.

A visual representation of this is shown in Figure 3.

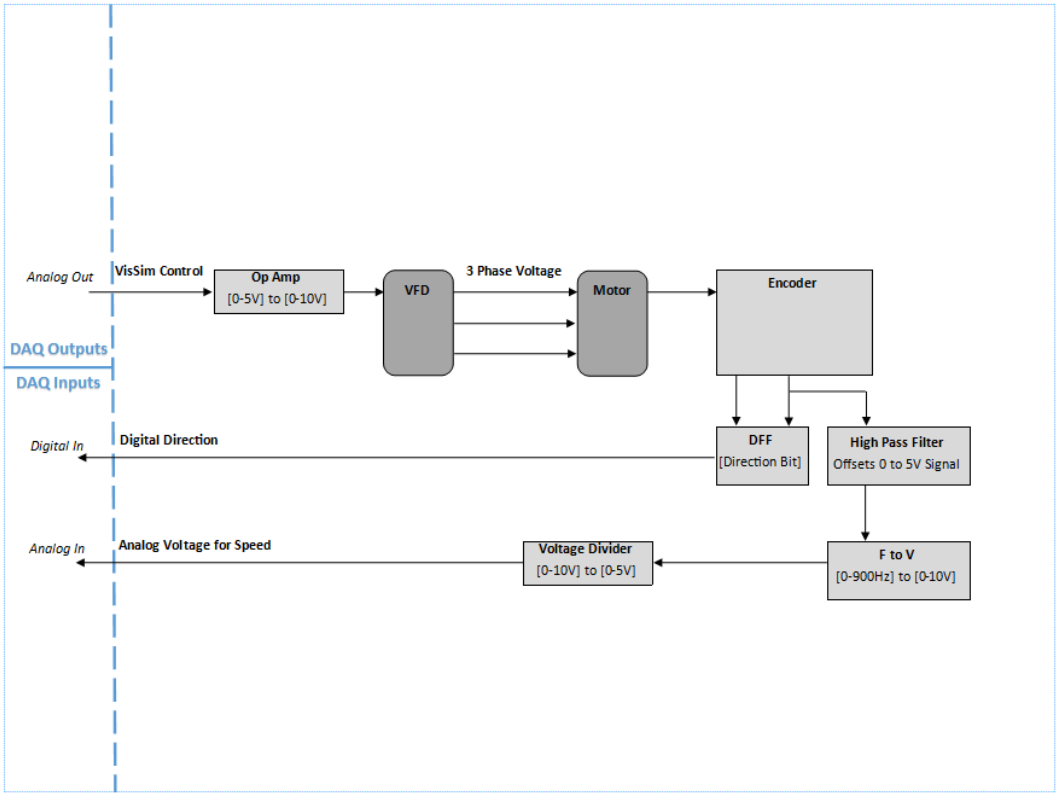


Figure 3. Block Diagram of physical compnents. This includes the VFD, motor, Encoder, F to V chip, D Flip Flop, and associated op amps.

PURPOSE

The purpose of this project was to design a set up for a lab that would serve as an introduction to variable frequency drives. The original plan was to have a digital to analog control (DAQ) scheme using a computer based program such as Vissim to receive an input from a motor and to output a control system to the VFD to regulate a certain speed or pattern. This would give students a chance to practice controlling motors with an automated system using feedback which could be useful in many industrial applications

CONSTRAINTS

Due to the fact that this design is intended for a classroom with students operating the equipment, safety was the main concern. The spinning motor presents threats to loose clothing and long hair, and the voltage involved can easily shock someone who isn't paying attention. Therefore, steps will need to be taken to protect students from harm.

This lab is also intended to be done by upwards of 13 groups of students. As such, any cost incurred would have to be multiplied by 13 to fully implement. So keeping cost low will also be important.

By the same token, any set up or construction would also have to be done 13 times. As a result, keeping the build as simple, and light, as possible is also of high importance. This is of particular importance as the motor will likely be quite heavy regardless. So, any pieces attached to it should be as light as possible to minimize back ache. Also, any pieces that can be bought "off the shelf" will be quite useful as anything custom built will need to be built many times.

DESIGN PROCESS

VFD AND MOTOR

The first step was to decide on a VFD. For our purposes we would need one that could take a single phase AC 120V input and control a three phase motor with it. As we are not connecting the rotor to anything other than an encoder, power could be kept to a minimum. Given this, we wanted the smallest, lightest motor we could get away with. This would also allow us to use a weaker VFD as it would not have to provide large amounts of power to the motor. This would keep cost low, reduce risk, and keep weight to a minimum.

Using these criteria we narrowed our search down to VFD's from Automation Direct, Omega Engineering, and Marshal Wolf Automation. We also narrowed our motor choices down to choices from the same companies. These choices were compared in the following table. Table 1 compares the VFD's from the respective companies. Seeing as power, price, and weight are all things that we hope to keep to a minimum, high values represent poor choices.

		Automation Direct	Omega Engineering	Marshall Wolf Automation
Aspect	Value	VFD	VFD	VFD
Safety	0.5	1	1	1
Price ($)	0.3	9	4	8
Weight	0.1	5	5	5
Power (low)	0.1	5	5	8
Total	1	3.70	2.20	3.70

Table 1. Comparison chart for VFD's

The same process was applied to the VFD's from these companies. Again, low values for weight, power, and price are desirable. These values are shown in Table 2.

		Automation Direct	Omega Engineering	Marshall Wolf Automation
Aspect	Value	motor	motor	motor
Safety	0.5			
Price ($)	0.3	87	201	224.1
Weight	0.1	23	23	30
Power (low)	0.1	0.5	0.5	0.25
Total	1	28.45	62.65	70.26

Table 2. Comparison chart for Motors.

These tables were then put into a table format to more clearly demonstrate the merits of these devices. Figure 4 displays the comparison of the VFD's, and Figure 5 displays the comparison of the motors.

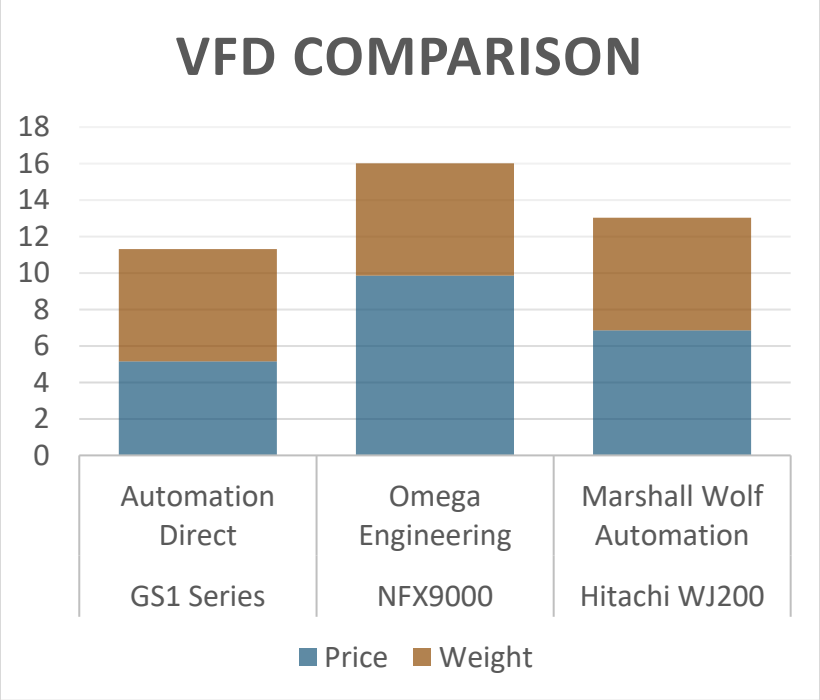


Figure 4. Comparison between possible VFD choices. Again, low values represent a better choice.

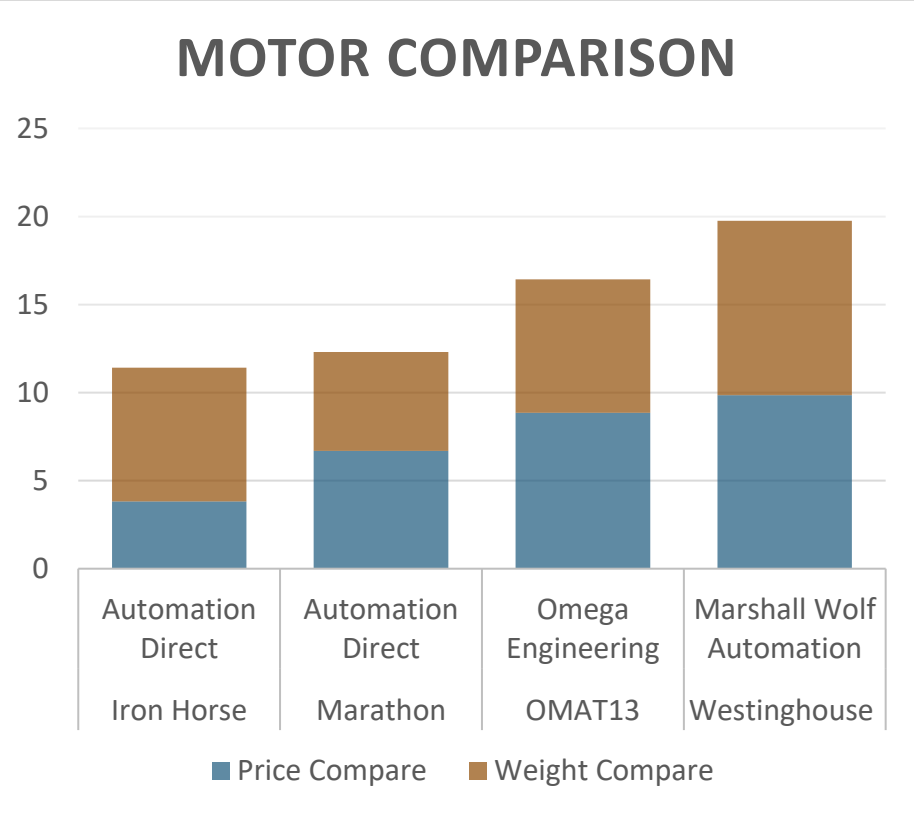


Figure 5. Comparison between possible motor choices. Again, low values represent a better choice.

This led us to decide upon the Automation direct 0.25 hp GS2 AC micro drive. This would be used to control a 0.25hp Marathon MicroMAX series AC induction motor. These choices were based on their relatively light weight, low power, and low cost.

Direction and Speed Sensors

The next step for this project was to design a circuit that could detect the speed and direction of the motor and relay this information in a way that the computer, through the DAQ and Vissim, could read. This would involve using an "off the shelf" optical encoder (Figure 6.) that has two output signals that can be used to determine direction. This design would simply require the construction of a mount to fix the encoder to the motor to prevent to encoder falling off or simply turning with the motor. This mount unfortunately would have to be 3D printed and bolted to the face of the motor. This will be time consuming to produce in sufficient numbers, but the time saved over creating our own encoder circuit, makes the mount and encoder chip the best option.

Figure 6. Encoder (black component in center) shown connected to end of motor.

An example of how you would use to determine direction from the encoders output is shown in Figure 7. In one direction the yellow signal will be leading, in the other direction, the blue signal will lead.

Figure 7. Encoder output with yellow signal leading (left), and blue signal leading (right)

One of these signals is then fed into the frequency-to-voltage (F to V) chip. This chip, pinout shown in Figure 8, takes a signals frequency and outputs a corresponding voltage. This voltage can then be read by the control program.

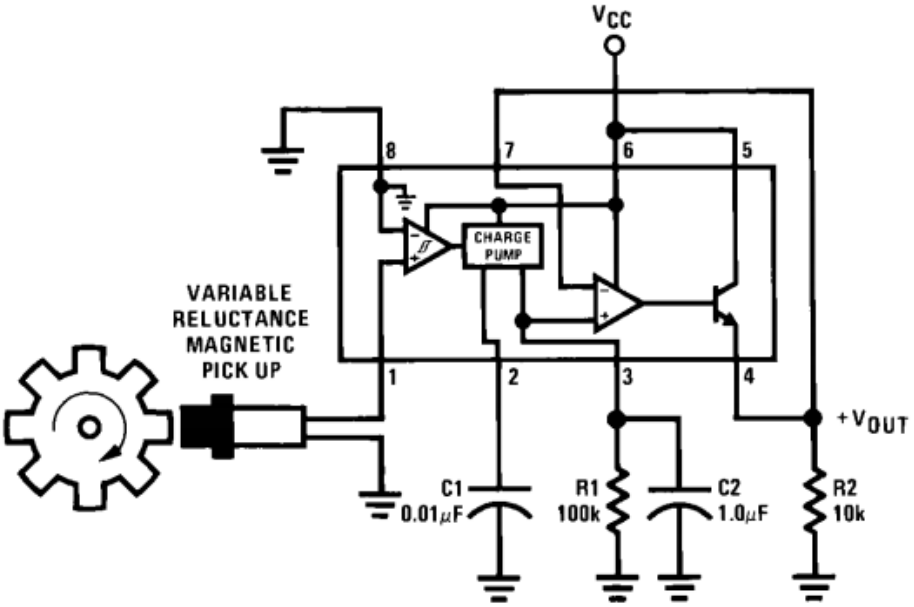


Figure 8. Pinout for the F to V chip. This takes a frequency from our encoder and turns it into a voltage that can be read by the computer and used as a feedback.

The first tests with this piece proved unsuccessful due to the fact that the chip needs the voltage it is reading to cross from a negative to a positive values and vice versa to register the frequency. This presented a small problem as the encoder merely outputs a 0-5V signal. This problem was fixed with the addition of an offset/gain circuit using a 741 op amp set up so to give us a gain of 1 and an offset of -2.5V.

With this piece implemented, we then tested it to determine the range of frequencies we could get a reliable voltage for. We ran the VFD and motor from 0 RPM until the voltage output by the chip would saturate. These results were tabulated and can be seen in Table 3.

The VFD was being controlled by an analog input controlled by a slider in Vissim. The value of the slider was then incremented slowly and measurements were taken at each position. As the slider was moved to full value, the analog signal being sent from the computer, to the VFD, increased as well. This analog signal voltage level is in the second column of the graph. In the third column is the speed displayed by the VFD. This is the speed the VFD predicts the motor will be going. While the motor spins, it turns the encoder wheel. The encoder wheel then outputs a 50:50 duty cycle square wave that peaks once for every time the motor makes a full revolution. The frequency of this signal was read by the oscilloscope and this value is shown in the fourth column. In the fifth column is shown the voltage output by the frequency to voltage chip. It is this voltage that will be fed back into the controlling program (in our case Vissim) and used to monitor the speed of the motor.

Input				Output
VisSim Slider Value	Voltage Input to VFD	Speed As predicted by the VFD (RPM)	Frequency Read by Oscilloscope attached to encoder	Voltage Output by Frequency to Voltage chip
0.0	0.0	0	0	0.2
0.1	0.2	50	0	0.6
0.2	0.4	80	68	1.2
0.3	0.6	120	97	1.4
0.4	0.8	150	114	1.8
0.5	1.0	190	150	2.2
1.0	2.0	370	300	3.8
1.5	3.0	550	450	5.4
2.0	4.0	720	600	7
2.5	5.0	900	750	8.4
3.0	6.0	1080	900	10
3.5	7.0	1250	1050	13.5
4.0	8.0	1440	1200	13.8
4.5	9.0	1600	1350	13.8
5.0	10.0	1750	1500	13.8

Table 3. Values measured during a full, one direction, run of the motor. The voltage output by the frequency to voltage chip is shown in the far right column and this will be used to monitor the speed of the motor during the lab.

As you can see in the table, the encoder and F to V start to saturate at a motor speed of around 1400 RPM. This was deemed acceptable as the point of the proposed lab is to control the motor, not to maximize its speed. Furthermore, the max speed of the motor is rated at 1800 RPM in either direction, so keeping the speeds down to around 1100 RPM will likely avoid unnecessary damage to the motor over extended periods of use. This speed will also be safer for those operating on the equipment as it would minimize any rotating inertias experienced by the motor and its housing. Additionally, this speed can be reached in either direction. Therefore we decided that the student being able to control the motors speed over a span of 1100 RPM in both the clock wise and counter clock wise directions would be sufficient for educational reasons.

Now that we could now read the speed of the motor, it was now time to build a system to read the direction of the motor as well. As stated before, the two signals of the encoder chip could be used to determine the speed of the motor by looking at which of the two signals is leading the other. With this in mind we set up a D flip-flop (Figure 9.) with one of the outputs of the encoder acting as the clock and the other output of the encoder acting as the Data input of the D flip-flop.

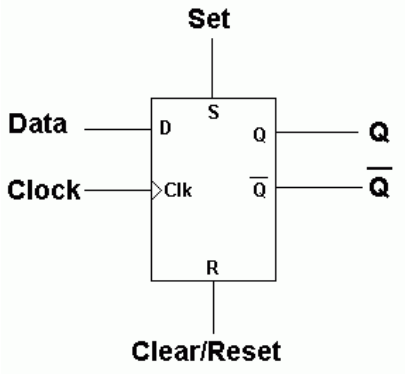


Figure 9. Basic D flip-flop schematic

With this setup, while the motor is spinning in one direction the D flip-flops Q output will be a 1. And while it is spinning in the other direction it will output a 0. Both possibilities are shown in Figure 10. with the D flip-flop outputting a 1 in the top example, and a 0 in the bottom example.

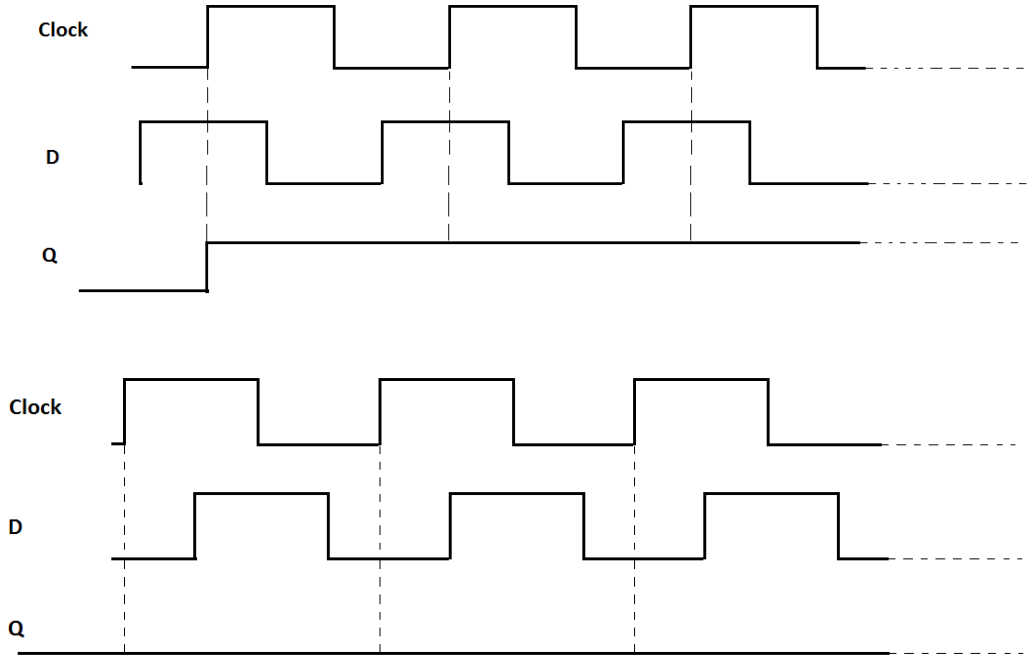


Figure 10. Input and output of D flip-flip, with a Q value of 1 (above) and 0 (below).

Using this, we can now tell the direction of the motor by whether the flip flop is outputting a 1 or a 0. We arbitrarily set the clockwise direction as 1, and the counterclockwise direction as 0. To test this, we set up Vissim to show us the values of the direction bit (output of the D flip flop), and the feedback voltage value. This feedback voltage allows us to determine the speed of the motor. This would result in the Vissim display shown in Figure 11.

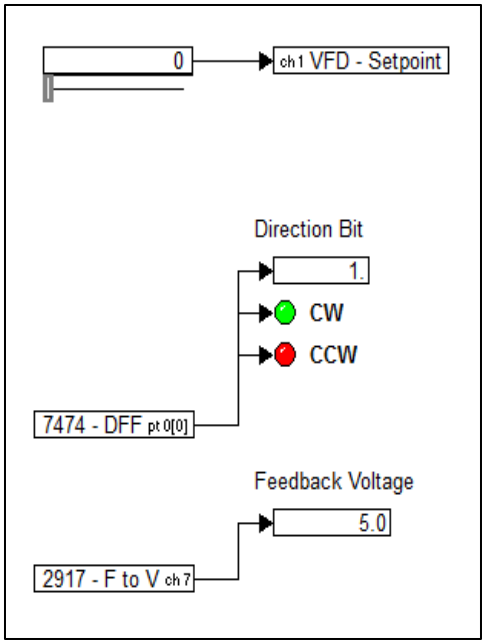
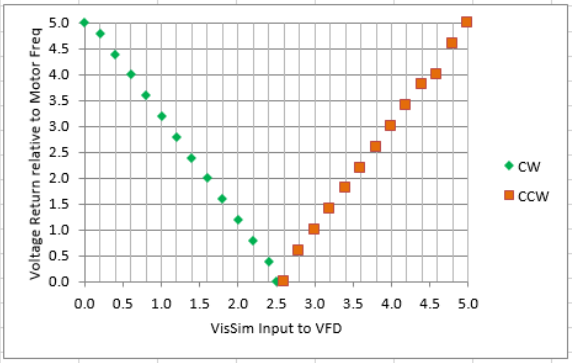


Figure 11. From top to bottom: The slider used to control the VFD. The display of the direction of the motor (shown in the clock wise position). The feedback voltage which will be used to read the speed of the motor (shown at 5V which indicates full speed).

Due to the technical specs of the on-hand DAQ's, we decided to use 5V to indicate that the motor is spinning at 1100RPM in the indicated direction. We also have the slider in Vissim setup to tell the VFD to go full speed clock wise when the slider is at the 0 position (Figure 5.) and full speed counterclockwise when it is at the 5 position. To illustrate this we recorded values of the feedback voltage and direction bit as we stepped from the 0 to 5 position at steps of 0.2. This is shown if Figure 12.

Vissim Slider Value	F to V Return
0.0	5.0
0.2	4.8
0.4	4.4
0.6	4.0
0.8	3.6
1.0	3.2
1.2	2.8
1.4	2.4
1.6	2.0
1.8	1.6
2.0	1.2
2.2	0.8
2.4	0.4
2.5	0.0
2.6	0.0
2.8	0.6
3.0	1.0
3.2	1.4
3.4	1.8
3.6	2.2
3.8	2.6
4.0	3.0
4.2	3.4
4.4	3.8
4.6	4.0
4.8	4.6
5.0	5.0



CW
Stalled
CCW

Figure 12. Values of the Feedback voltage recorded as the slider position is changed from the 0 to 5 position.

As can be seen in the figure above, the relationship between motor speed and voltage is for the most part linear, regardless of which direction the motor is spinning. With this, we can now read the speed and direction of the motor in a way our existing control programs and equipment can read and interact with.

Now that our equipment and instrumentation are working, all that is left is the control software.

CONTROL SOFTWARE

As stated before, we would be using the Vissim control program to control our VFD and motor set up. This is primarily because all the computers that would be used in this lab have this program on it already. Therefore, it will not add any cost to the project to license the software, nor any time to teach the students or Lab T.A.s how to use a new program. Vissim also works well with our existing DAQ's and for those reasons, is ideal for this project.

Our initial plan was to use a PID controller in this lab to reinforce the idea of feedback loops to the students as well as to minimize any damage done to the equipment. However, with the amount of noise present, any derivative component would likely cause instability, so a PI system was decided upon.

While the VFD gradually changes speed on its own, we prefer greater control over the process so we will have a PI control in Vissim as mentioned before. This could also be used for educational reasons as the students could then easily adjust the P and I values and watch the effect it has on the system, this is not our primary goal however.

The system we came up with is shown below in Figure 13. This control system has two user controlled inputs. These are the speed and direction of the motor controlled by the slider and switch in the top left of the figure respectively. The measured speed and direction of the motor is displayed directly below these inputs. The speed is displayed numerically by the dipsplay block connected to the RPM – Actual block. The direction of the motor is displayed by the colored circles appearing below the speed blocks.

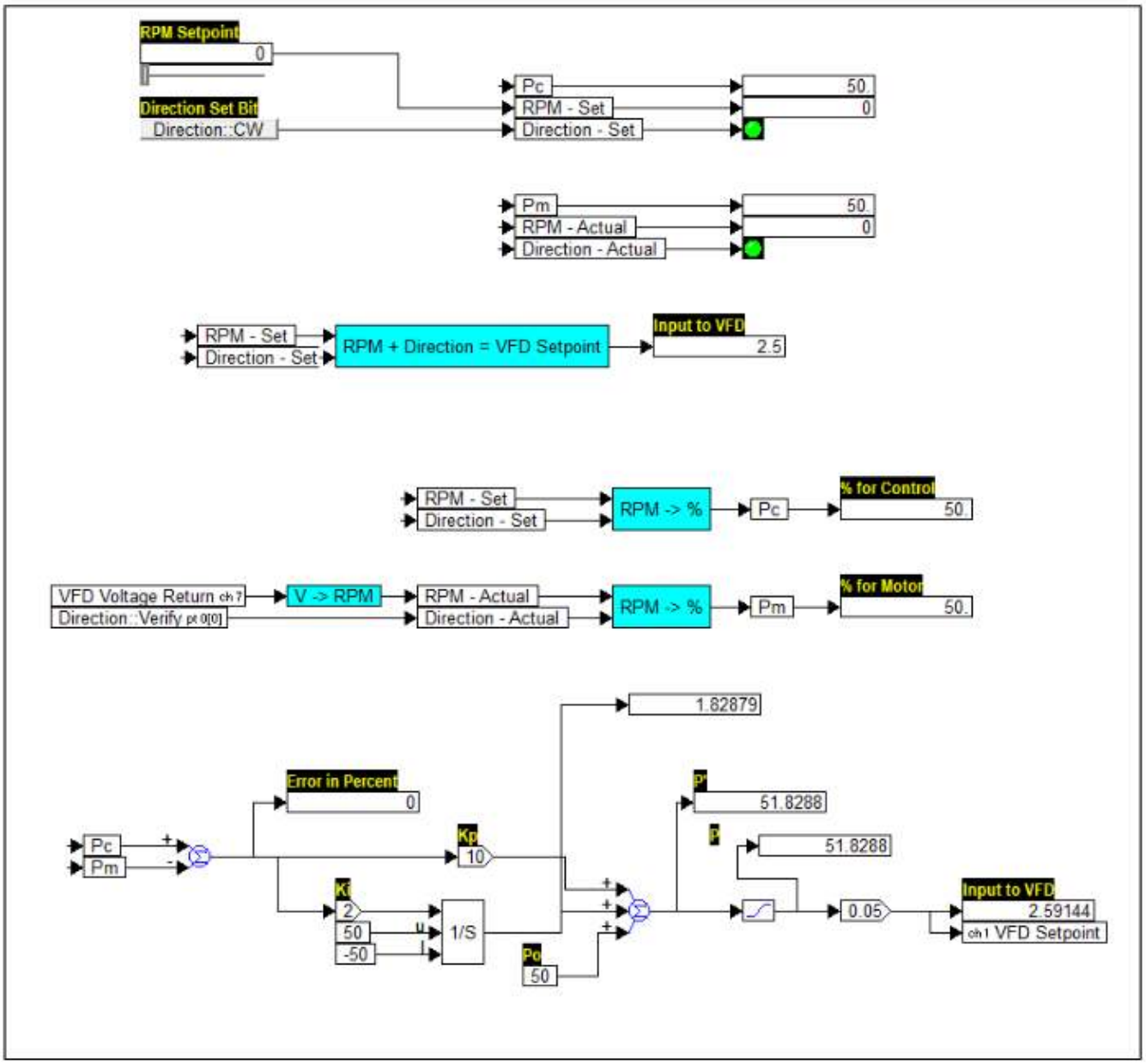


Figure 13. Our control system in Vissim. The user controlled inputs are at the top left of the figure and directly below those set bits are the measurments of the motor. Below these is the actual control and Feedback system.

For the sake of ease, Figure 14. Contains this same system with visual aids separating the different components of the software.

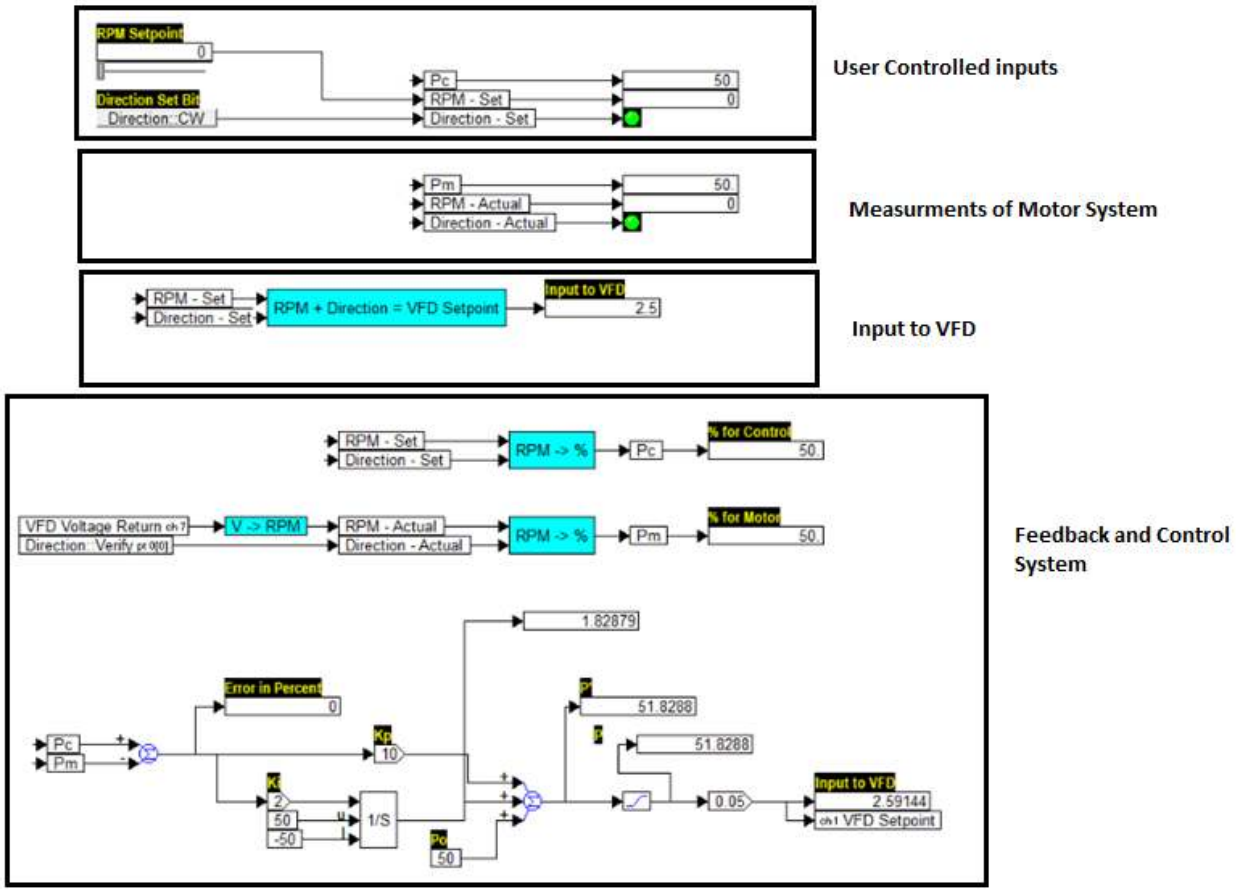


Figure 14. Control system with different elements segregated.

System Response

After we developed the control system, the next step was to determine the systems response to immediate changes in input. To do this we ran the system and instantaneously changed to direction or speed input and graphed the change in speed experienced by the motor. First we used only proportional control, and this result is shown in Figure 15.

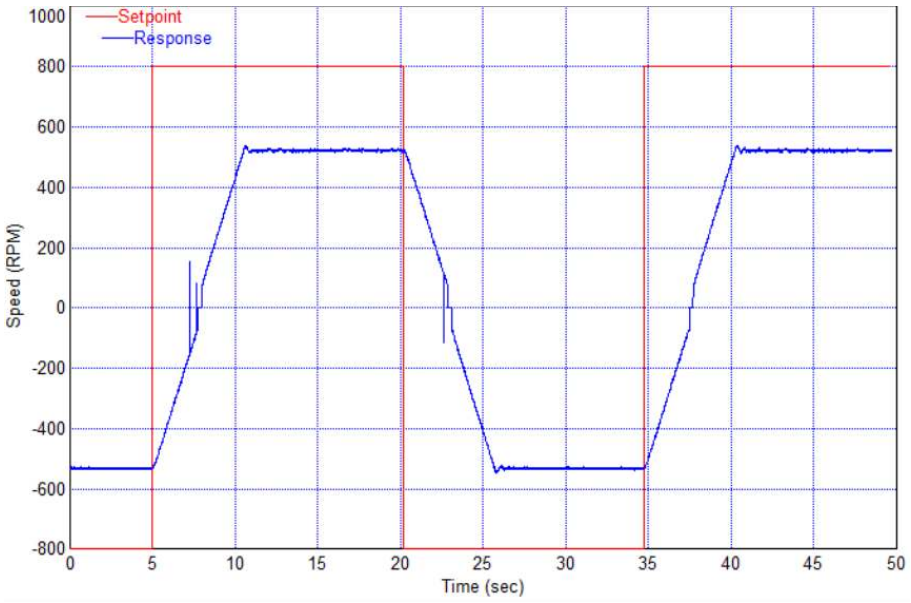


Figure 15. Systems response to input change with only proportional control.

As can be seen in the graph, there is droop present. To remedy this, we implemented integral control. The results of this can be shown in Figure 10. The values used for this are Kp=2 and Ki=0.5. As can be seen in Figure 16. this has eliminated the droop and while there is still some overshoot, it is manageable.

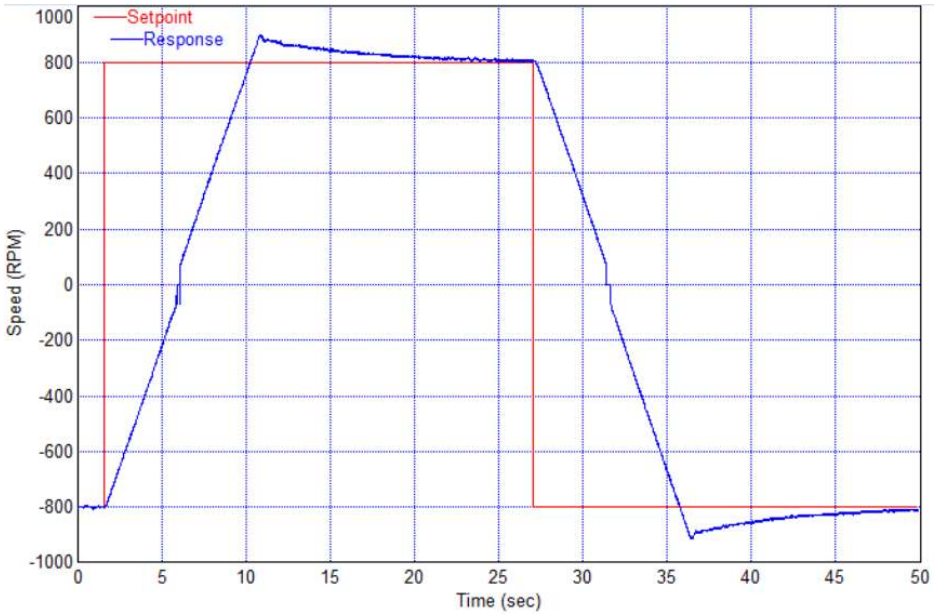


Figure 16. System response with integral and proportional control.

BUILD

All the electrical components can be placed into a circuit board in a simple way this is outlined by the schematics in this report. Therefore, this section will focus primarily on the wiring of the motor and encoder and the mounting of the encoder to the motor.

In order to fix the encoder in such a way as to read the speed and direction of the motor, it had to be fixed to the end of the shaft such that it would not turn with the shaft. This would require building a mount for the encoder that attached to the body of the motor using the four screws on the faceplate. This led us to 3D print a plate to fix to the motor and to a separate piece that would connect to the encoder at the end of the shaft to hold it in place. While we wished to avoid 3D printing to make reproduction as easy as possible, we could find no way around this.

This system would also allow us to easily build a protective case around the moving parts while leaving a clear plastic window to see the moving parts. Designs for the motor and encoder mount are shown in Figure 17 and 18 respectively.

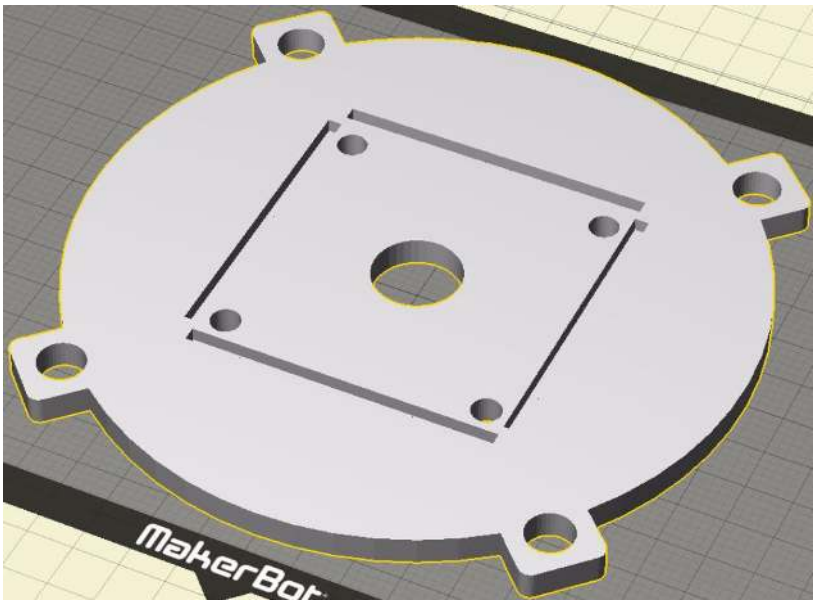


Figure 17. Solid Works design for motor mount.

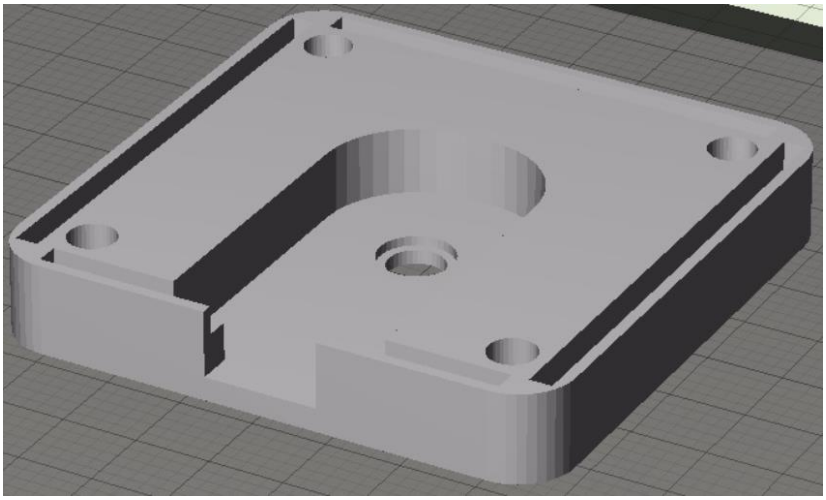


Figure 18. Solid Works Design for encoder mount with recessed space for encoder.

These two elements would then be connected with bolts to the motor and to each other. These pieces mounted to the motor are shown in Figure 19.

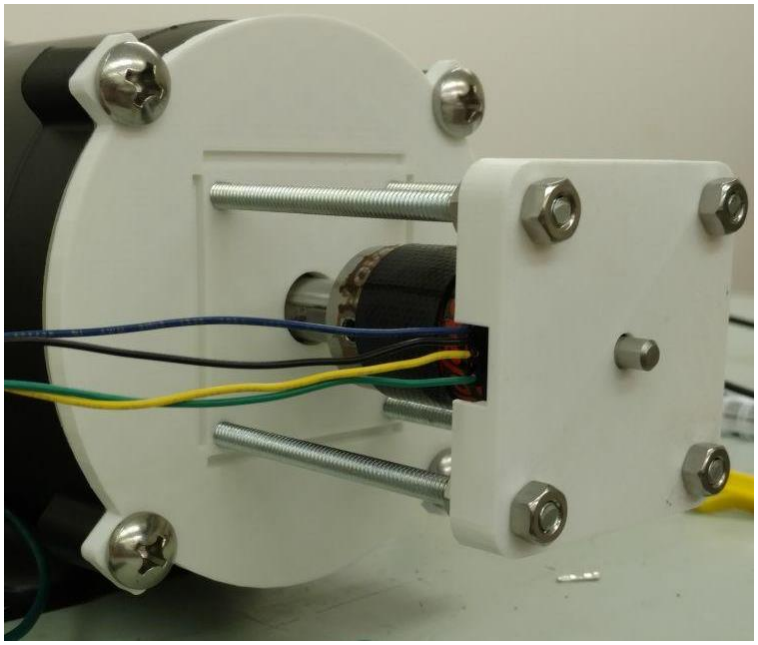


Figure 19. Motor with mounts connected. Notice the wires coming from the bottom of the encoder.

All the components assembled are shown in Figure 20. Notice there is still no protective case around the motors shaft. This is the only component not present in this picture and it is absent strictly for demonstration reasons.

Figure 20. VFD, motor, encoder, and instrumentation circuit assembled.

SCHEMATIC OF SPEED AND DIRECTION SENSING SYSTEM

Because a picture is worth a thousand words a schematic of the system we have built can be seen in Figure 21. As can be seen below, the encoder wheel outputs are fed into the frequency to voltage chip and into the D flip- flop. Only one of these outputs needs to go into the frequency to voltage chip, so the A channel is shown being used for this as it is easier to depict. Both outputs must go to the D flip-flop, however. This is due to the fact that one of the outputs is acting as the clock and the other output is used as the normal data stream.

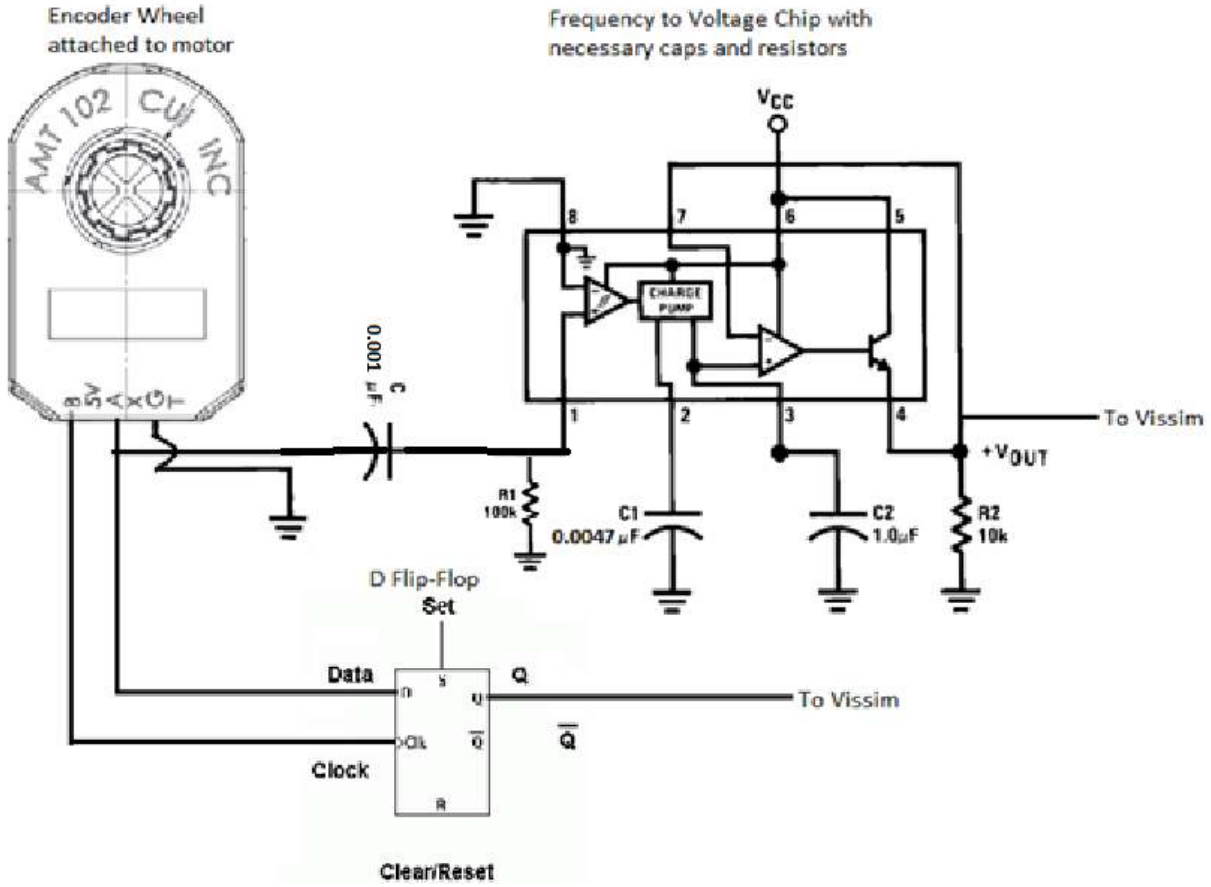


Figure 21. Our speed and direction sensing system. The D flip flop is outputting a 1 or 0 that will be used to indicate direction. And the output of the F to V chip can be correlated to the speed of the motor.

With this system the speed and direction of the motor can be read and transmitted to a computer control system via the DAQ. Once this system was built and tested, our build was complete.

BUDGET

As stated before, any cost incurred during the design would need to be multiplied by the number of lab stations. Given this, the budget for the project was placed at $500. The two most expensive components would be the motor and VFD. Along with these we would also need the encoder as described earlier and a braking resistor was added to decrease the time it took the motor to come to a complete stop. A fuse kit was also added to power the VFD to prevent damage to both the VFD itself, as well as the electrical systems in the building. All of these costs came out to $415.62, well under our $500 budget. These values are also shown on Table 4ss.

Component	Cost
VFD	$156.00
Motor	$152.50
Fuse Kit	$35.00
Braking Resistor	$48.50
Encoder	$23.62
Total	$415.62

Table 4. Cost expenditures of the project

TIME TABLE

The time table for this project is as shown in Figure 22. Seeing as how this was for a class and the deadline for its completion was determined in advance, we had a pretty good idea of when we had to have things done by. As such, we had our work order laid out from the beginning and it was therefore easy to stick to.

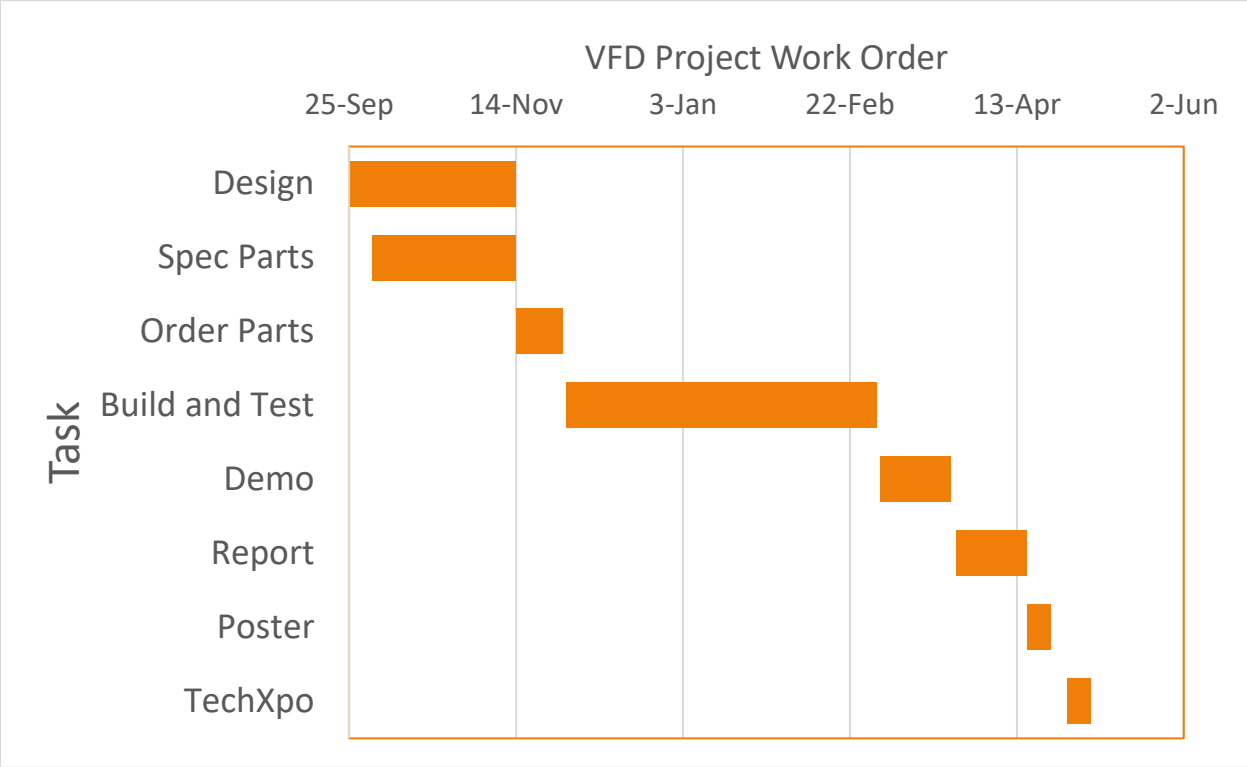


Figure 22. Project Time Table

APPENDIX A

SET UP OF THE VFD

In order to set up the VFD for the purposes of this lab you will start by pressing the program button until the screen displays the correct number next to a P, for example P1.00. Then you will use the arrow keys to scroll through the last two digits. Hitting the up arrow once will give you P1.01, twice will give you P1.02, etc.

Then you will hit enter once you reach your desired setting, in this case P1.02
Once you have done this you will be able to change the parameter value. This will be done with the arrow buttons and once you have the desired value you will hit enter again. For example, for this lab, you will press the up arrows until the screen displays 10, then you will press enter, and that parameter will be set. In this case, the parameter is deceleration time.

You will repeat this until all the values match the ones shown below in Figure 23.

Parameter	Setting
P1.02	10
P4.00	2
P4.01	2
P4.02	60
P4.03	120
P4.04	1
P4.05	0
P4.11	0
P4.12	0

Figure 23. Parameter settings of the VFD

Once this is done, the VFD is ready to go.

On the following pages contains a list of parameters and the associated code to manipulate them. These tables are taken from Chapter 4 of the owner's manual.

The ◆ symbol indicates that these settings can be set during the RUN mode.

Motor Parameters			
GS2 Parameter	Description	Range	Default
P0.00	Motor Nameplate Voltage	115V/230V: 200/208/220/230/240 460V: 380/400/415/440/460/480 575V: 380 to 637	240 480 575
P0.01	Motor Nameplate Amps	Drive Rated Amps X .3 to 1.0	Drive Rated Amps x 1.0
P0.02	Motor Base Frequency	50/60/400	60
P0.03	Motor Base RPM	375 to 9999 RPM	1750
P0.04	Motor Maximum RPM	P0.03 to 9999 RPM	P0.03

Ramp Parameters			
P1.00	Stop Methods	00: Ramp to Stop 01: Coast to Stop	00
◆ P1.01	Acceleration Time 1	0.1 to 600.0 sec	10.0
◆ P1.02	Deceleration Time 1	0.1 to 600.0 sec	30.0
P1.03	Accel S-curve	0 to 7	00
P1.04	Decel S-curve	0 to 7	00
◆ P1.05	Acceleration Time 2	0.1 to 600.0 sec	10.0
◆ P1.06	Deceleration Time 2	0.1 to 600.0 sec	30.0
P1.07	Select method to use 2nd Accel/Decel	00: RMP2 from DI terminal 01: Transition Frequencies P1.08 & P1.09	00
P1.08	Accel 1 to Accel 2 frequency transition	0.0 to 400.0 Hz	0.0
P1.09	Decel 2 to Decel 1 frequency transition	0.0 to 400.0 Hz	0.0
P1.10	Skip Frequency 1	0.0 to 400.0 Hz	0.0
P1.11	Skip Frequency 2	0.0 to 400.0 Hz	0.0
P1.12	Skip Frequency 3	0.0 to 400.0 Hz	0.0
P1.17	Skip Frequency Band	0.0 to 20.0 Hz	0.0
P1.18	DC Injection Current Level	00 to 100 %	00
P1.20	DC Injection during Start-up	0.0 to 5.0 sec	0.0
P1.21	DC Injection during Stopping	0.0 to 25.0 sec	0.0
P1.22	Start-point for DC Injection	0.0 to 60.0 Hz	0.0

Volts/Hertz Parameters			
GS2 Parameter	Description	Range	Default
P2.00	Volts/Hertz Settings	00: General Purpose 01: High Starting Torque 02: Fans and Pumps 03: Custom	00
◆ P2.01	Slip Compensation	0.0 to 10.0	0.0
◆ P2.02	Auto-torque Boost	00 to 10	00
P2.04	Mid-point Frequency	0.1 to 400 Hz	1.5
P2.05	Mid-point Voltage	115V/230V: 2.0 to 255V 460V: 2.0 to 510V 575V: 2.0 to 637V	10.0 20.0 24.0
P2.06	Min. Output Frequency	0.1 to 20.0 Hz	1.50
P2.07	Min. Output Voltage	115V/230V: 2.0 to 50.0V 460V: 2.0 to 100.0V 575V: 2.0 to 130.6V	10.0 20.0 24.0
P2.08	PWM Carrier Frequency	115V/230V/460V 01 to 15 kHz 575V 01 to 10 kHz	12 6

Digital Parameters			
P3.00	Source of Operation Command	00: Operation determined by digital keypad 01: Operation determined by external control terminals, keypad STOP is enabled 02: Operation determined by external control terminals, keypad STOP is disabled 03: Operation determined by RS-485 interface, keypad STOP is enabled 04: Operation determined by RS-485 interface, keypad STOP is disabled	00
P3.01	Multi-function Input Terminals (DI1 - DI2)	00: DI1 - FWD / STOP, DI2 - REV / STOP 01: DI1 - RUN / STOP, DI2- REV / FWD 02: DI1 - RUN momentary (N.O.) DI2 - REV / FWD DI3 - STOP momentary (N.C.)	00

Digital Parameters (cont.)			
GS2 Parameter	Description	Range	Default
P3.02	Multi-function Input (DI3)	00: External Fault (N.O.) 01: External Fault (N.C.) 02: External Reset 03: Multi-Speed/PID SP Bit 1 04: Multi-Speed/PID SP Bit 2 05: Multi-Speed/PID SP Bit 3 06: Reserved 07: Reserved 08: Reserved 09: Jog 10: External Base Block (N.O.) 11: External Base Block (N.C.) 12: Second Accel/Decel Time 13: Speed Hold 14: Increase Speed 15: Decrease Speed 16: Reset Speed to Zero 17: PID Disable (N.O.) 18: PID Disable (N.C.) 99: Input Disable	00
P3.03	Multi-function Input (DI4)		03
P3.04	Multi-function Input (DI5)		04
P3.05	Multi-function Input (DI6)		05
P3.11	Multi-Function Output Terminal 1	00: AC Drive Running 01: AC Drive Fault 02: At Speed 03: Zero Speed 04: Above Desired Frequency 05: Below Desired Frequency 06: At Maximum Speed 07: Over torque detected 08: Above Desired Current 09: Below Desired Current 10: PID Deviation Alarm	00
P3.12	Multi-Function Output Terminal 2		01
◆ P3.16	Desired Frequency	0.0 to 400.0 Hz	0.0
◆ P3.17	Desired Current	0.0 to <Drive Rated Amps>	0.0
◆ P3.18	PID Deviation Level	1.0 to 50.0 %	10.0
◆ P3.19	PID Deviation Time	0.1 to 300.0 sec	5.0

Analog Parameters			
GS2 Parameter	Description	Range	Default
P4.00	Source of Frequency Command	00: Frequency determined by keypad potentiometer 01: Frequency determined by digital keypad up/down 02: Frequency determined by 0 to +10V input on AI terminal with jumpers 03: Frequency determined by 4 to 20mA input on AI terminal with jumpers 04: Frequency determined by 0 to 20mA input on AI terminal with jumpers 05: Frequency determined by RS-232C/ RS-485 communication interface	00
P4.01	Analog Input Offset Polarity	00: No Offset 01: Positive Offset 02: Negative Offset	00
◆ P4.02	Analog Input Offset	0.0 to 100.0%	0.0
◆ P4.03	Analog Input Gain	0.0 to 300.0%	100.0
P4.04	Analog Input Reverse Motion Enable	00: Forward Motion Only 01: Reverse Motion Enable	00
P4.05	Loss of ACI Signal (4-20mA)	00: Decelerate to 0Hz 01: Stop immediately and display error code "EF" 02: Continue operation by the last frequency command	00
◆ P4.11	Analog Output Signal	00: frequency Hz 01: Current A 02: PV	00
◆ P4.12	Analog Output Gain	00 to 200%	100

Presets			
◆ P5.00	Jog	0.0 to 400.0 Hz	6.0
◆ P5.01	Multi-Speed 1	0.0 to 400.0 Hz	0.0
◆ P5.02	Multi-Speed 2	0.0 to 400.0 Hz	0.0
◆ P5.03	Multi-Speed 3	0.0 to 400.0 Hz	0.0
◆ P5.04	Multi-Speed 4	0.0 to 400.0 Hz	0.0
◆ P5.05	Multi-Speed 5	0.0 to 400.0 Hz	0.0
◆ P5.06	Multi-Speed 6	0.0 to 400.0 Hz	0.0
◆ P5.07	Multi-Speed 7	0.0 to 400.0 Hz	0.0

Protection Parameters			
GS2 Parameter	**Description**	**Range**	**Default**
P6.00	Electronic Thermal Overload Relay	00: Constant Torque 01: Variable Torque 02: Inactive	00
P6.01	Auto Restart after Fault	00 to 10	00
P6.02	Momentary Power Loss	00: Stop operation after momentary power loss 01: Continue operation after momentary power loss, speed search from Speed Reference 02: Continue operation after momentary power loss, speed search from Minimum Speed	00
P6.03	Reverse Operation Inhibit	00: Enable Reverse Operation 01: Disable Reverse Operation	00
P6.04	Auto Voltage Regulation	00: AVR enabled 01: AVR disabled 02: AVR disabled during decel 03: AVR disabled during stop	00
P6.05	Over-Voltage Stall Prevention	00: Enable Over-voltage Stall Prevention 01: Disable Over-voltage Stall Prevention	00
P6.06	Auto Adjustable Accel/Decel	00: Linear Accel/Decel 01: Auto Accel, Linear Decel 02: Linear Accel, Auto Decel 03: Auto Accel/Decel 04: Auto Accel/Decel Stall Prevention (limited by P1.01, P1.02, P1.05, P1.06)	00
P6.07	Over-Torque Detection Mode	00: Disabled 01: Enabled during constant speed operation 02: Enabled during acceleration	00
P6.08	Over-Torque Detection Level	30 to 200%	150
P6.09	Over-Torque Detection Time	0.1 to 10.0	0.1
P6.10	Over-Current Stall Prevention during Acceleration	20 to 200%	150
P6.11	Over-Current Stall Prevention during Operation	20 to 200%	150
P6.12	Maximum Allowable Power Loss Time	0.3 to 5.0 sec	2.0
P6.13	Base-Block Time for Speed Search	0.3 to 5.0 sec	0.5
P6.14	Maximum Speed Search Current Level	30 to 200%	150
P6.15	Upper Bound of Output Frequency	0.1 to 400Hz	400
P6.16	Lower Bound of Output Frequency	0.0 to 400Hz	0.0
P6.30	Line Start Lockout	00: Enable Line Start Lockout 01: Disable Line Start Lockout	00

Protection Parameters (cont.)			
GS2 Parameter	Description	Range	Default
P6.31	Present Fault Record	00: No Fault occurred 01: Over-current (oc) 02: Over-voltage (ov) 03: Overheat (oH) 04: Overload (oL) 05: Overload 1 (oL1) 06: Overload 2 (oL2) 07: External Fault (EF) 08: CPU failure 1 (CF1) 09: CPU failure 2 (CF2) 10: CPU failure 3 (CF3) 11: Hardware Protection Failure (HPF) 12: Over-current during accel (OCA) 13: Over-current during decel (OCd) 14: Over-current during steady state (OCn) 15:Ground fault or fuse failure (GFF) 16: Reserved 17: Input power 3-phase loss 18: External Base-Block (bb) 19: Auto Adjust accel/decel failure (cFA) 20: Software protection code (codE)	00
P6.32	Second Most Recent Fault Record		00
P6.33	Third Most Recent Fault Record		00
P6.34	Fourth Most Recent Fault Record		00
P6.35	Fifth Most Recent Fault Record		00
P6.36	Sixth Most Recent Fault Record		00

PID Parameters			
GS2 Parameter	Description	Range	Default
P7.00	Input Terminal for PID Feedback	00: Inhibit PID operation 01: Forward-acting (heating loop) PID feedback, PV from AVI (0 to + 10V 02: Forward-acting (heating loop) PID feedback, PV from ACI (4 to 20mA) 03: Reverse-acting (cooling loop) PID feedback, PV from AVI (0 to +10V). 04: Reverse-acting (cooling loop) PID feedback, PV from ACI (4 to 20mA).	00
P7.01	PV 100% Value	0.0 to 999	100.0
P7.02	PID Setpoint Source	00: Keypad 01: Serial Communications	00
◆ P7.10	Keypad PID Setpoint	0.0 to 999	0.0
◆ P7.11	PID Multi-setpoint 1	0.0 to 999	0.0
◆ P7.12	PID Multi-setpoint 2	0.0 to 999	0.0
◆ P7.13	PID Multi-setpoint 3	0.0 to 999	0.0
◆ P7.14	PID Multi-setpoint 4	0.0 to 999	0.0
◆ P7.15	PID Multi-setpoint 5	0.0 to 999	0.0
◆ P7.16	PID Multi-setpoint 6	0.0 to 999	0.0
◆ P7.17	PID Multi-setpoint 7	0.0 to 999	0.0
◆ P7.20	Proportional Control	0.0 to 10.0	1.0
◆ P7.21	Integral Control	0.00 to 100.0 sec	1.00
◆ P7.22	Derivative Control	0.00 to 1.00 sec	0.00
P7.23	Upper Bound for Integral Control	00 to 100%	100
P7.24	Derivative Filter Time Constant	0.0 to 2.5 sec	0.0
P7.25	PID Output Frequency Limit	00 to 110%	100
P7.26	Feedback Signal Detection Time	0.0 to 3600 sec.	60
P7.27	PID Feedback Loss	00: Warn and AC Drive Stop 01: Warn and Continue Operation	00

Display Parameters			
◆ P8.00	User Defined Display Function	00: Output Frequency (Hz) 01: Motor Speed (RPM) 02: Output Freq. X P8.01 03: Output Current (A) 04: Motor Output Current (%) 05: Output Voltage (V) 06: DC Bus Voltage (V) 07: PID Setpoint 08: PID Feedback Signal (PV) 09: Frequency Setpoint	00
◆ P8.01	Frequency Scale Factor	0.01 to 160.0	1.0

Communications Parameters			
GS2 Parameter	Description	Range	Default
P9.00	Communication Address	01 to 254	01
P9.01	Transmission Speed	00: 4800 baud 01: 9600 baud 02: 19200 baud 03: 38400 baud	01
P9.02	Communication Protocol	00: Modbus ASCII mode 7 data bits,no parity,2 stop bits 01: Modbus ASCII mode 7 data bits,even parity,1 stop bit 02: Modbus ASCII mode 7 data bits,odd parity,1 stop bit 03: Modbus RTU mode 8 data bits,no parity,2 stop bits 04: Modbus RTU mode 8 data bits,even parity,1 stop bit 05: Modbus RTU mode 8 data bits,odd parity,1 stop bit	00
P9.03	Transmission Fault Treatment	00: Display fault and continue operating 01: Display fault and RAMP to stop 02: Display fault and COAST to stop 03: No fault displayed and continue operating	00
P9.04	Time Out Detection	00: Disable 01: Enable	00
P9.05	Time Out Duration	0.1 to 60.0 seconds	0.5
◆ P9.07	Parameter Lock	00: All parameters can be set and read 01: All parameters are read-only	00
P9.08	Restore to Default	99: Restores all parameters to factory defaults	00
◆ P9.11	Block Transfer Parameter 1	P0.00 to P8.01, P9.99	P9.99
◆ P9.12	Block Transfer Parameter 2	P0.00 to P8.01, P9.99	P9.99
◆ P9.13	Block Transfer Parameter 3	P0.00 to P8.01, P9.99	P9.99
◆ P9.14	Block Transfer Parameter 4	P0.00 to P8.01, P9.99	P9.99
◆ P9.15	Block Transfer Parameter 5	P0.00 to P8.01, P9.99	P9.99
◆ P9.16	Block Transfer Parameter 6	P0.00 to P8.01, P9.99	P9.99
◆ P9.17	Block Transfer Parameter 7	P0.00 to P8.01, P9.99	P9.99
◆ P9.18	Block Transfer Parameter 8	P0.00 to P8.01, P9.99	P9.99
◆ P9.19	Block Transfer Parameter 9	P0.00 to P8.01, P9.99	P9.99
◆ P9.20	Block Transfer Parameter 10	P0.00 to P8.01, P9.99	P9.99
◆ P9.21	Block Transfer Parameter 11	P0.00 to P8.01, P9.99	P9.99
◆ P9.22	Block Transfer Parameter 12	P0.00 to P8.01, P9.99	P9.99

Communications Parameters (continued)			
GS2 Parameter	Description	Range	Default
◆ P9.23	Block Transfer Parameter 13	P0.00 to P8.01	P9.99
◆ P9.24	Block Transfer Parameter 14	P0.00 to P8.01	P9.99
◆ P9.25	Block Transfer Parameter 15	P0.00 to P8.01	P9.99
◆ P9.26	Serial Comm Speed Reference	0.0 to 400.0 Hz	60.0
◆ P9.27	Serial Comm RUN Command	00: Stop 01: Run	00
◆ P9.28	Serial Comm Direction Command	00: Forward 01: Reverse	00
◆ P9.29	Serial Comm External Fault	00: No fault 01: External fault	00
◆ P9.30	Serial Comm Fault Reset	00: No action 01: Fault Reset	00
◆ P9.31	Serial Comm JOG Command	00: Stop 01: Jog	00
P9.39	Firmware Version	#.##	#.##
P9.41	GS Series Number	01: GS1 02: GS2 03: GS3	##
P9.42	Manufacturer Model Information	00: GS2-20P5 (230V 1ph/3ph 0.5hp) 01: GS2-21P0 (230V 1ph/3ph 1hp) 02: GS2-22P0 (230V 1ph/3ph 2hp) 03: GS2-23P0 (230V 1ph/3ph 3hp) 04: GS2-25P0 (230V 3ph 5hp) 05: GS2-27P5 (230V 3ph 7.5hp) 06: Reserved 07: GS2-41P0 (460V 3ph 1hp) 08: GS2-42P0 (460V 3ph 2hp) 09: GS2-43P0 (460V 3ph 3hp) 10: GS2-45P0 (460V 3ph 5hp) 11: GS2-47P5 (460V 3ph 7.5hp) 12: GS2-4010 (460V 3ph 10hp) 13: GS2-10P2 (115V 1ph 0.25hp) 14: GS2-10P5 (115V 1ph 0.5hp) 15: GS2-11P0 (115V 1ph 1hp) 16~20: Reserved 21: GS2-51P0 (575V 3ph 1hp) 22: GS2-52P0 (575V 3ph 2hp) 23: GS2-53P0 (575V 3ph 3hp) 24: GS2-55P0 (575V 3ph 5hp) 25: GS2-57P5 (575V 3ph 7.5hp) 26: GS2-5010 (575V 3ph 10hp)	##

APPENDIX B

EQUATIONS USED IN VISSIM

To control our VFD we used a slider in VisSim. We set the slider to go from the 0 to 100 position. This would correspond to a DAQ output of 0 to 5 volts. This would then cause the motor to go from 1130 RPM in the clockwise direction, to 1130 RPM in the counter clockwise direction. This can be seen in Table 5.

Slider Position (in %)	Voltage Output of DAQ (V)	Motor Speed and Direction
0	0	1130 RPM Clockwise
50	2.5	0 RPM
100	5	1130 RPM Counter Clockwise

Table 5. Slider Position and the control voltage and motor speed.

The relation of speed to voltage is described in this equation

$$Vout = \frac{Vrange}{Speed\ Range} * Motor\ Speed \qquad Vout = \frac{5-0}{1130-(-1130)} * Motor\ Speed$$

This then turns into

$$\boldsymbol{Vout = 0.00221 * Motor\ Speed}$$

To control the speed in two directions, the output of the D flip-flop, the direction bit, was used to multiply the motor speed variable by -1 when the motor was spinning CW, when the direction bit was high. This was then offset by 2.5 volts so that the output voltages would be centered around 2.5 Volts. This worked in practice like this (Table 6.)

Clockwise	Counter Clockwise
$Vout = 0.00221 * (-1)\ Motor\ Speed$	$Vout = 0.00221 * Motor\ Speed$

Table 6. Final Equations for output of DAQ

Next, this voltage was converted to a percent range (P_{range}) with the following equation

$$\text{Prange} = (\text{Vout}) * \frac{\text{Range}}{\text{Vrange}} \qquad \text{Prange} = (\text{Vout}) * \frac{(100-0)\%}{(5-0)V}$$

Which simplifies to

$$\boldsymbol{\text{Prange} = (\text{Vout}) * 20}$$

Plugging in the value for Vout, this becomes

$$\textbf{Prange} = \textbf{0.0442} * (\textbf{Motor Speed}) + \textbf{50}$$

And taking into account for direction

Clockwise	Counter Clockwise
$\text{Prange} = 0.0442 * (-1) * (\text{Motor Speed}) + 50$	$\text{Prange} = 0.0442 * (\text{Motor Speed}) + 50$

EQUATIONS USED IN FEEDBACK LOOP

To get the speed of the motor from the feedback voltage, the voltage was put through this equation

$$\text{Speed} = \text{Feedback Voltage} * \frac{1130 - 0}{5 - 0\text{V}}$$

Which becomes

$$\textbf{Motor Speed Actual} = \textbf{226} * \textbf{Feedback Voltage}$$

With this we can display the actual speed of the motor and compare it to the desired speed.

The motor speed is sent into our control loop along with the direction bit and is used to monitor the motor's current state.