Lab 3 - DC Motor Driver and Feedback Controller

Abhinav Avula, Will Boenning, Dominic Menassa

3/16/2022

Introduction and Objectives

In this lab experiment, we made a DC motor driver and a feedback control system, building off the circuit we used to measure DC motor parameters and the speed sensor circuit from the last lab experiment.

For part 3A, we studied the H-bridge circuit, which is used to drive a DC motor in both directions as opposed to the circuit from the last lab which only used one direction. We built an H-bridge circuit in LTSpice to explore how it works and simulated it with given armature resistances in both the forwards and backwards directions. Then, we used the motor parameters from experiment 2A to make a model for the motor driver circuit and use that to take more measurements on the motor when it is connected to the motor driver circuit. We then built one part of the H-bridge on the robot breadboard to test it before soldering, and after verifying that it works, we soldered the motor driver onto a protoboard and then tested it with the motor. After verification of functionality, we attached the motor driver to the board. To summarize, the goals of part 3A were to simulate and build a fully functioning DC motor driver circuit using an H-bridge to control the direction that the motor spins, effectively controlling the direction in which the robot moves.

The objective of 3A was to gain more understanding of how transistors function and act as the switches in an H-bridge, and how the H-bridge can be used to control direction of the motor. The materials we used do this were LTSpice to simulate the motor circuit, two MJE200G BJT transistors, two MJE210 BJT transistors, two 100 nanofarad capacitors, two 100 microfarad capacitors, two screw terminals (serve as DC+ and DC- to the motor), header pins (serve as Vcc, Vb1, Vb, and ground), and solder.

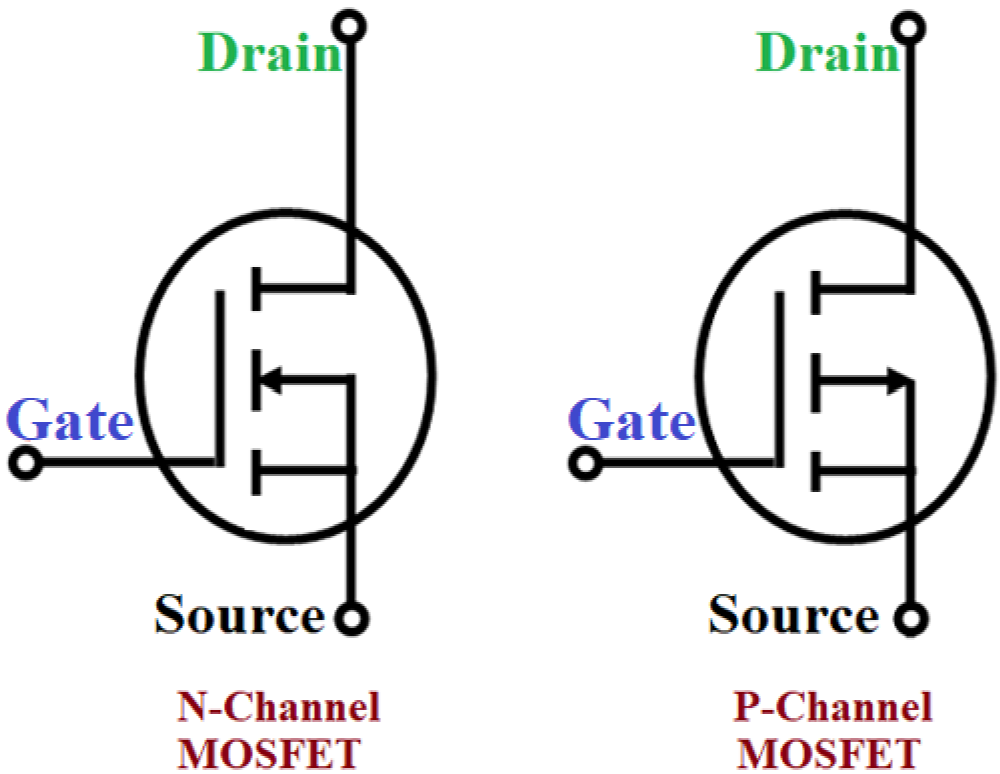
For 3B, we worked off the DC motor driver from the last part and the speed sensor circuit we built in the last experiment to build a feedback controller. First, we explored the idea of having a virtual ground and built one in LTSpice, and ran some analysis. Next, we constructed the I-compensator circuit in LTSpice and then simulated it in an open loop. When we verified this works, we moved on to building a closed loop circuit in LTSpice. After we verify this also works, we build the compensator circuit on the robot and test it in comparison to the LTSpice simulations to ensure that we get the correct output. After building the compensator circuit, we finished the direction control. As usual, we first built the circuit on LTSpice, and then added it to the closed-loop circuit implementation. After finishing the SPICE simulation and verifying that it works, we built the circuit on the breadboard and ran some tests.

The objectives of 3B were to build a compensator circuit that acts as a feedback control system to control the speed of the motors and ensure that they are all moving at the same speed, no matter what happens to one. The materials used were LTSpice to do simulations prior to building the Analog Discovery 2 with the BNC adapter and scope probe setup to test the physical circuit, resistors, and capacitors to build the feedback system.

3.A.1 - Prelab and Real Estate Planning

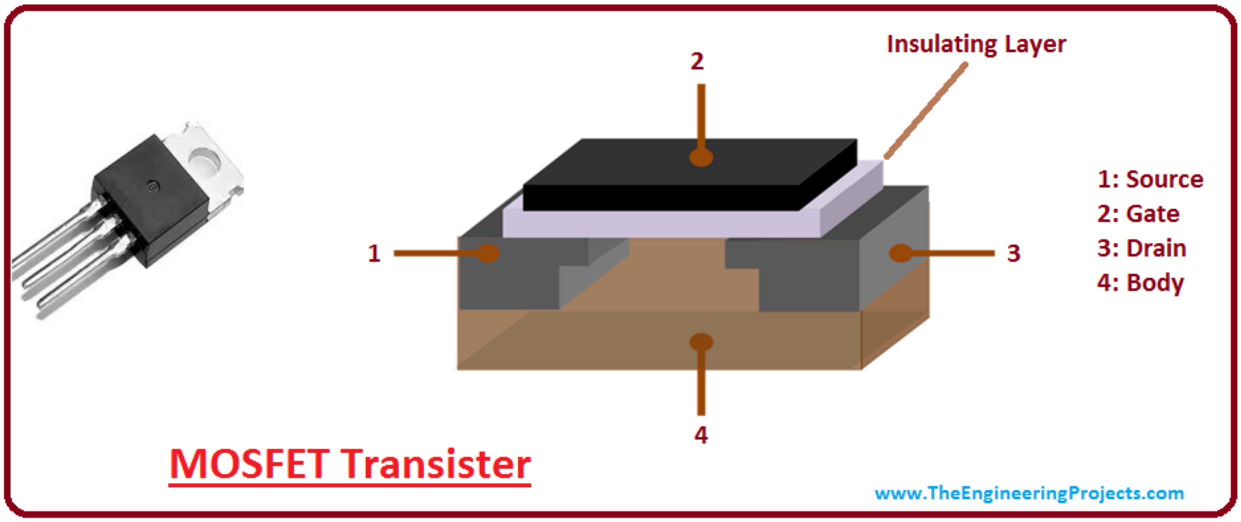
MOSFETs operate in a depletion mode and an enhancement mode, which are both available as N or P channels. Below is the schematic symbol for an N channel and P channel MOSFET:

N vs. P Channel MOSFETs



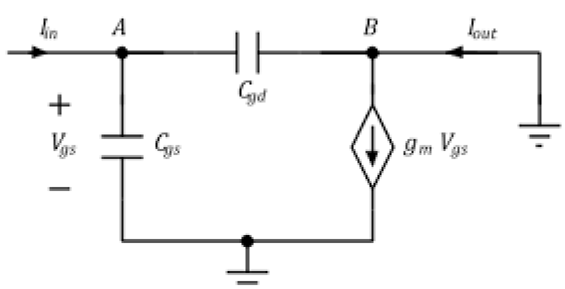
The P channel is used for high side switching, whereas the N channel is used for low side switching. MOSFETs are often made from silicon. Below is a diagram of their physical structure:

MOSFET Structure



MOSFETs allow or prevent current from flowing. A positive at the gate attracts electrons, which creates a connection from the source to the drain. This allows current to flow. The opposite is also true - a negative at the gate pushes away electrons, creating an open from source to gate and preventing current flow. Below is a simple equivalent circuit of a MOSFET:

MOSFET Equivalent Circuit

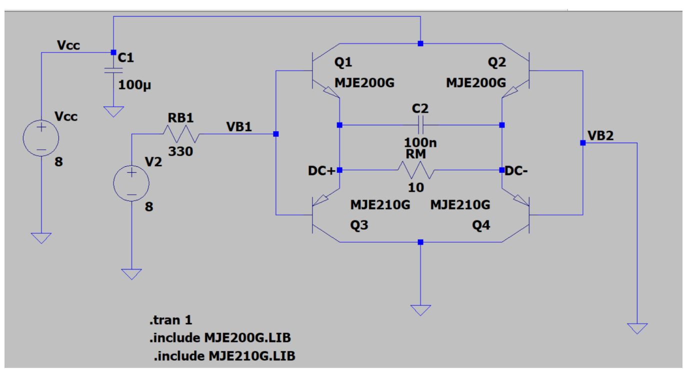


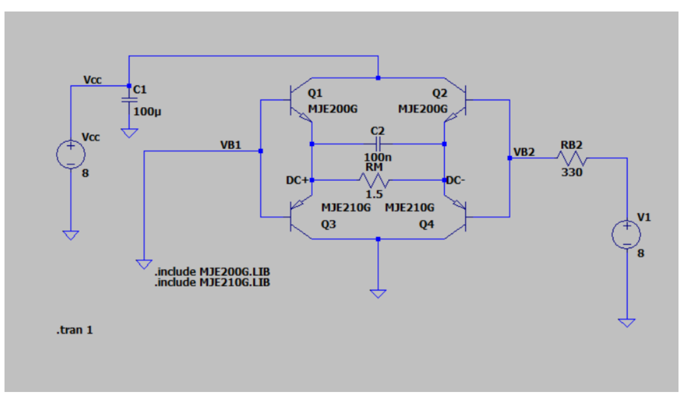
MOSFETs operate similarly to BJTs, although there are some differences. MOSFETs are voltage controlled, whereas BJTs are current controlled. MOSFETs also have a source, drain, and gate, whereas BJTs have a base, emitter, and collector. MOSFETs also consume less power and switch at a higher frequency. In addition to learning more about MOSFETs, we also made our speed sensors more compact prior to the beginning of this lab.

3.A.2 - Getting to Know the H-Bridge Circuit

For this experiment, we had to build the following two circuits in LT Spice:

H-Bridge Simulation Forward (Top) and Backward (Bottom)





We had to download the necessary libraries in order to use the correct transistors that we will use in our physical circuit. The two simulations are showing how the H-Bridge will drive the car forwards or backwards. The MJE200G is active high and the MJE210G is active low. So, when base 1 is high and base 2 is low, the top left and bottom right transistors will be active, and the opposite is true when base 1 is low and base 2 is high. We then had to fill out the following table by running the simulations in LT Spice:

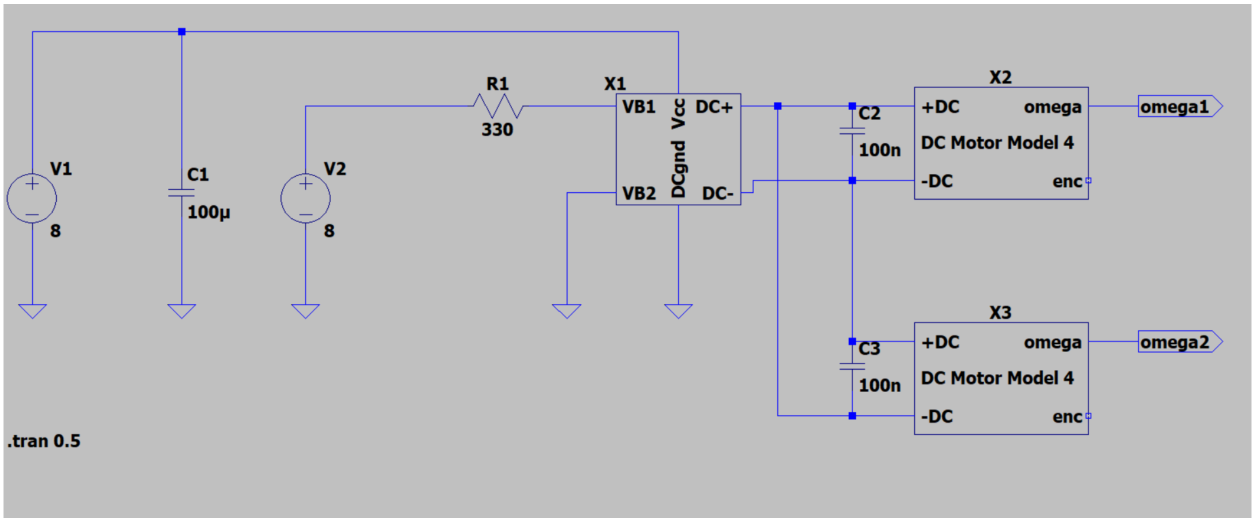
Table of Simulated Values for the H-Bridge

R_M (Ω)	V_{B1} (V)	V_{B2} (V)	DC^+ (V)	DC^- (V)	I_{RM} (A)	P_{Q1} (W)	P_{Q2} (W)	P_{Q3} (W)	P_{Q4} (W)
1.5	3.534	0	2.771	0.7644	1.337	6.933	0	0	1.022
10	6.370	0	5.650	0.7204	0.4930	1.150	0	0	0.3551
1.5	0	3.534	0.7644	2.771	-1.337	0	6.933	1.022	0
10	0	6.370	0.7204	5.650	-0.4930	0	1.150	0.3551	0

3.A.3 - Simulate the Motor Driver Circuit

This experiment required us to create a model of the motor driver and use it to simulate the motor drive connected with the two DC motors. The LT Spice for this (in this case in the forward direction) is pictured below:

Simulated Motor Driver Circuit in the Forward Direction



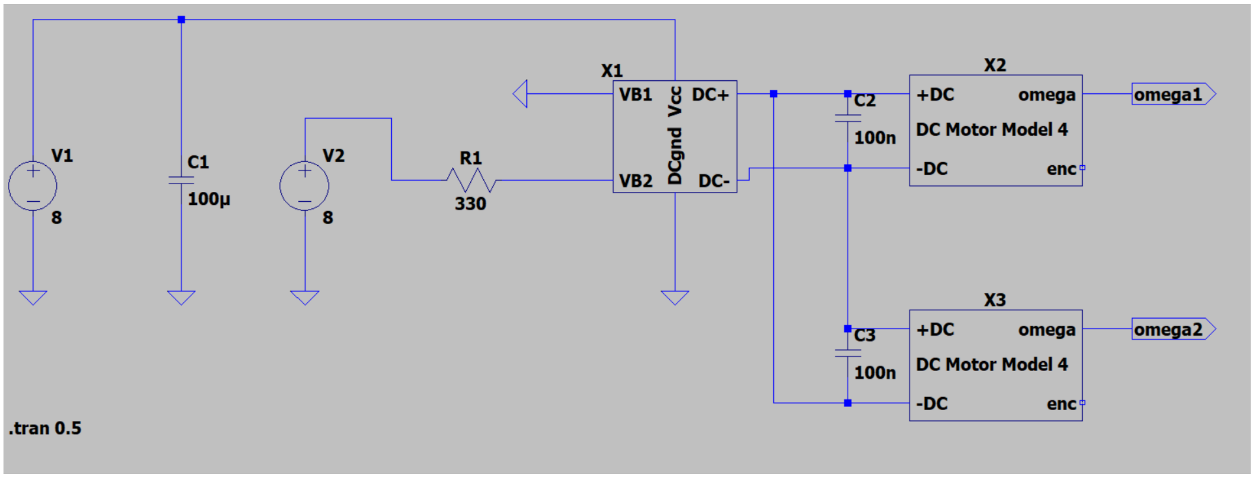
X2 runs forward and X3 runs backwards as the motors are mirrored from each other, so therefore their voltages will need to be opposite in order for the wheels to spin in the same direction. We then measured the current through both motors, and the omega values of both motors in the cases that both are free running, one is stalled and one is free running, and both are stalled. Below are the results:

Simulated Voltages and Currents of Free Running and Locked Motors for the Forward Direction

	I_{RM} (A)	ω_1 (V)	ω_2 (V)
Both Free Running	0.645	-11.01	11.01
One Stalled One Free	1.202	0	5.714
Both Stalled	1.3975	0	0

After this, we had to perform the same simulation but in the backwards direction. Below is the LT Spice model used to simulate this:

Simulated Motor Driver Circuit in the Backward Direction



This time, X2 will run backward and X3 will run forward. This is for the same reason as previously stated - the motors are mirrored from each other, so they need to run opposite directions in order for the wheels to spin in the same direction. We then again measured the total motor current and the two omega values for the three cases, shown below:

Simulated Voltages and Currents of Free Running and Locked Motors for the Backward Direction

	I_{RM} (A)	ω_1 (V)	ω_2 (V)
Both Free Running	-0.645	11.01	-11.01
One Stalled One Free	-1.202	0	-5.714

Both Stalled	-1.3975	0	0

As we can see, we get nearly identical results, except the signs are flipped. This makes sense as we would anticipate that flipping the signs would change the direction of the car, since in previous labs we have tested this out by switching the red motor wire with the black motor wire.

3.A.4 - Testing Motor Driver Circuit on Breadboard

In this section of the experiment, we tested our H-Bridge by physically building the circuit on our breadboard. We then took measurements of the different values in order to compare it with our values from 3.A.2. Note that we only tested the circuit on the breadboard for one direction (meaning forward or backward), as we felt this would be sufficient in determining if the circuit worked properly. Below are the results:

Measured Values from Motor Driver Circuit Breadboard

R_M (Ω)	V_B (V)	DC^+ (V)	DC^- (V)	I_{RM} (A)
1.5 (stalled)	4.01	2.66	1.19	0.98
10 (free running)	5.91	4.85	0.87	0.398

As we can see, our results are relatively similar to our simulated values, giving us confidence in our circuit. There is a relatively notable difference in the currents, but since we used a multimeter to measure our current it makes sense that the measured version would be slightly lower due to an internal resistance in the multimeter itself.

3.A.5 - Construct the Motor Driver Circuit

This experiment involved actually constructing our motor driver circuit and soldering it to the perfboard. After testing our circuit on the breadboard, we were confident in translating the final H-Bridge onto the perfboard. This was a tedious process, but with careful planning we were able to properly connect and solder the circuit. The next experiment involved actually testing this circuit, so we will elaborate on how well the circuit worked there.

3.A.6 - Test the Motor Driver Circuit

For this experiment, we took physical measurements of our motor driver circuit. We based which measurements we decided to take off of the ones from 3.A.2, excluding power. Below are the results:

Measured Values from Actual Motor Driver Circuit

R_M (Ω)	V_{B1} (V)	V_{B2} (V)	DC^+ (V)	DC^- (V)	I_{RM} (A)
1.5 (stalled)	3.30	0	2.58	0.84	0.93
10 (free	5.70	0	5.17	0.74	0.383

running)					
1.5 (stalled)	0	3.20	0.83	2.38	-0.945
10 (free running)	0	5.80	0.75	5.03	-0.403

Evidently, our numbers are quite close to our simulated values. Again, there is a notable difference in current, which we can conclude might be due to the internal resistance when measuring with a multimeter. However, as a whole our measured numbers give us confidence in the effectiveness of our circuit. As a visual check, we also observed our wheels when connected to the motor driver and saw that both spun in the same direction, which is obviously our intended goal as the robot would not move otherwise.

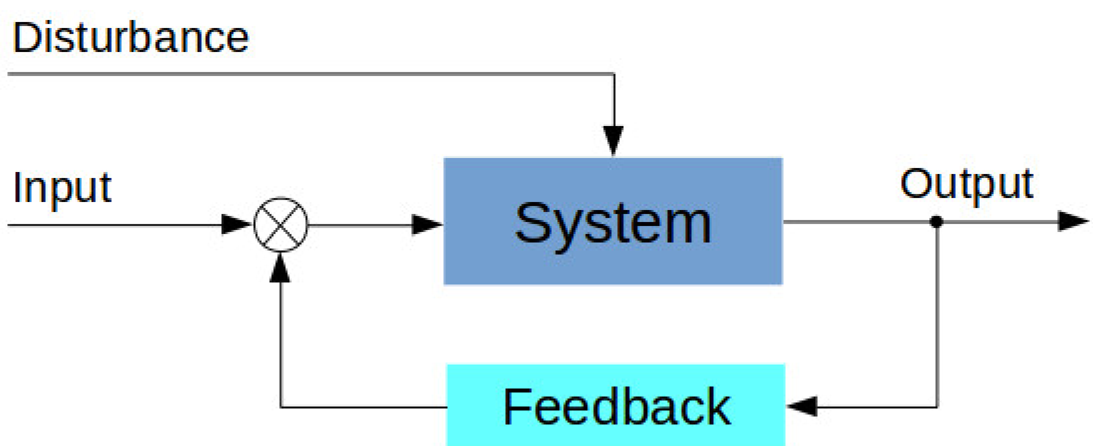
3.A.7 - Mount Motor Driver Circuit Inside Robot

This experiment does not require much analysis, as we simply were just velcroing our motor drivers to the bottom of where our breadboard is mounted.

3.B.1 - Prelab

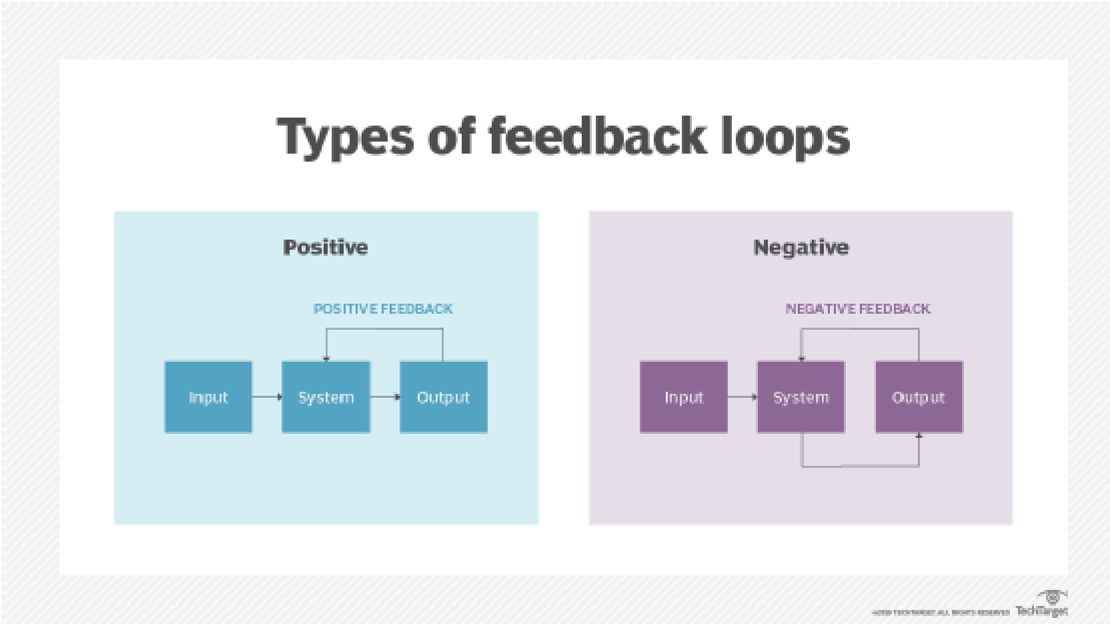
Q: What are some common examples of feedback control systems?

A: A feedback control system is a system whose output is measured and then used as an input to correct any error that may be present. The output signal is fed back into the system and measured against a reference signal, and the difference signal is filtered to match the desired output. Some examples of feedback control systems are a home furnace control system (to keep the temperature of a home at a desired rate), an automobile's cruise control (controls the speed to some value preset by the driver), et. Some non circuit examples are the human brain (constantly sending signals to the brain to keep constantly correct behavior in completing tasks such as picking up an item, talking, basic comprehension, etc.), and soil erosion (how soil reacts to how much water is present).



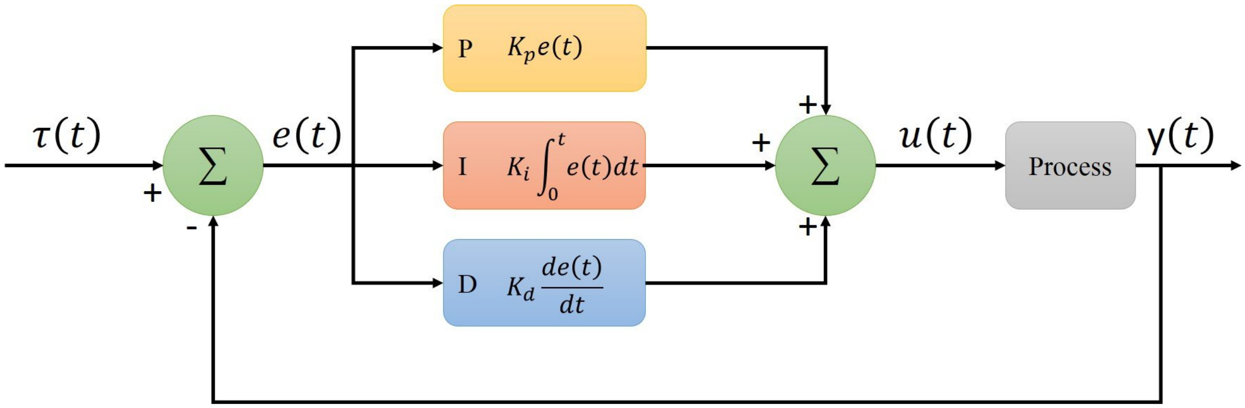
Q: How do the functions of positive and negative feedback systems differ?

A: Positive feedback systems amplify and cause more change in a system, whereas negative feedback systems aim to stabilize the output of a system. Going back to the previous examples, the home temperature regulator is a negative feedback control system because with each input signal it takes, it’s job is to minimize the error and keep the output in a stable state while an example of a positive feedback system is the



Q: What is proportional, integral, and derivative control?

A: Proportional control is a type of control system in which there is a correction to the variable that is user controlled. This correction is proportional to the difference between the measured output that is fed back to the system and the desired output. Proportional controllers are used to stabilize the system and reduce steady-state errors. Integral control is often used with proportional control. It's job is to also correct errors, but unlike the proportional controller which can't detect trends, the integral controller detects and corrects trends. The derivative controller is in charge of detecting and correcting sudden changes that occur in the system. All of these controllers can work together in a PID system configuration to automatically adjust the output signal.



Q: What is a good block diagram that shows the main principle of feedback control?

A: A block diagram represents the flow of the signals in a feedback control system. It shows only the flow of signals, not the energy transfer between each of the different parts of the circuit. A good block diagram is simple both graphically and mathematically, and should obviously resemble feedback and show the path from the output signal through the different stages of the feedback control system.

Q: What are the advantages of feedback control systems?

A: Some of the advantages of feedback systems are the ability to correct the error of a system by automatically adjusting the controllers that are used, instant increase in stability, reduction of external disturbances, less human interaction with the system, and the reliability that they provide .

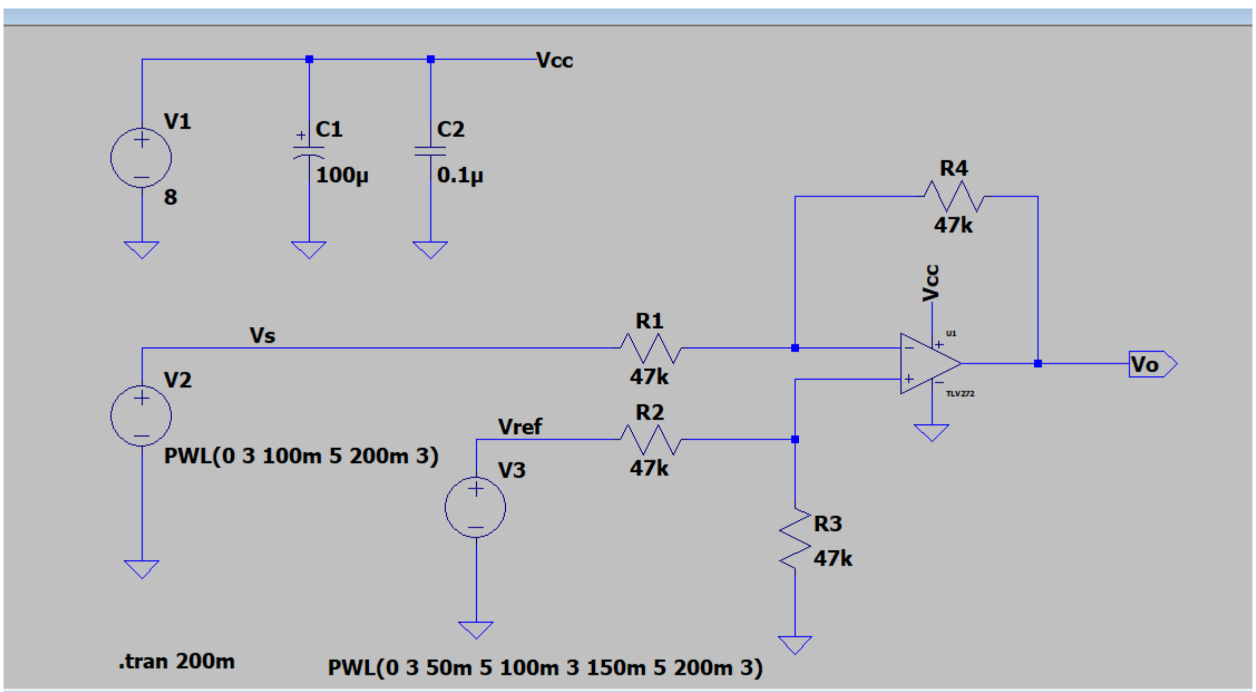
Q: What are the disadvantages of feedback control systems?

A: Some of the disadvantages of feedback control systems are that they must be constantly undergo changes and this could cause issues with reliability and some feedback systems are extremely complex which can make them hard to implement.

3.B.2 - Making the Case for Virtual Ground

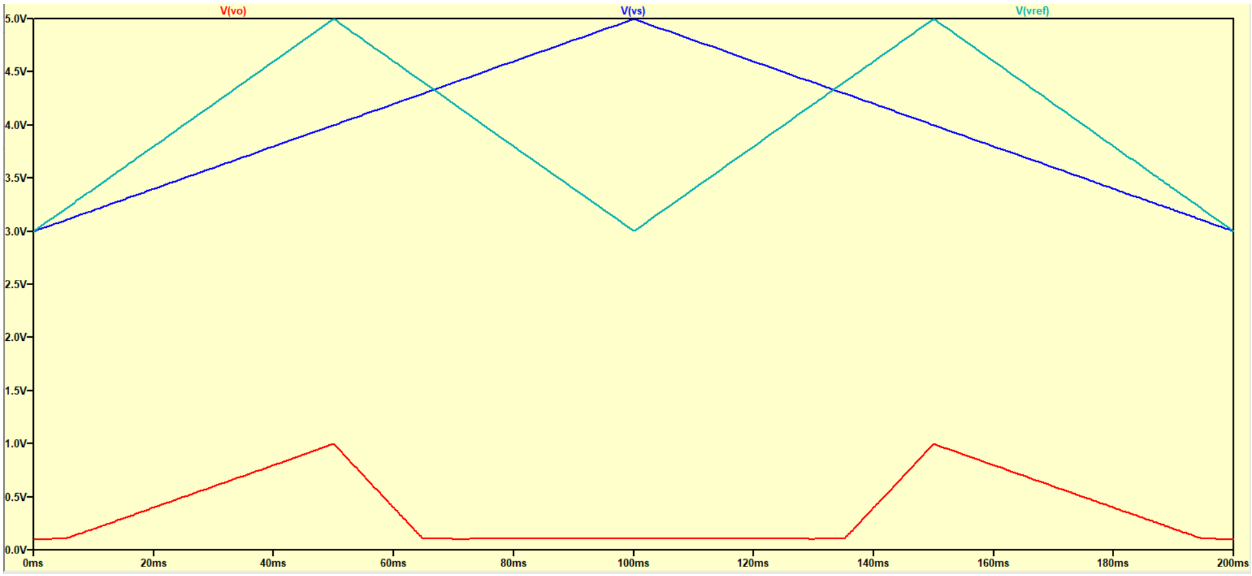
This experiment was intended to prove why a virtual ground may be needed. Below is the circuit we are analyzing:

Circuit using Regular Ground



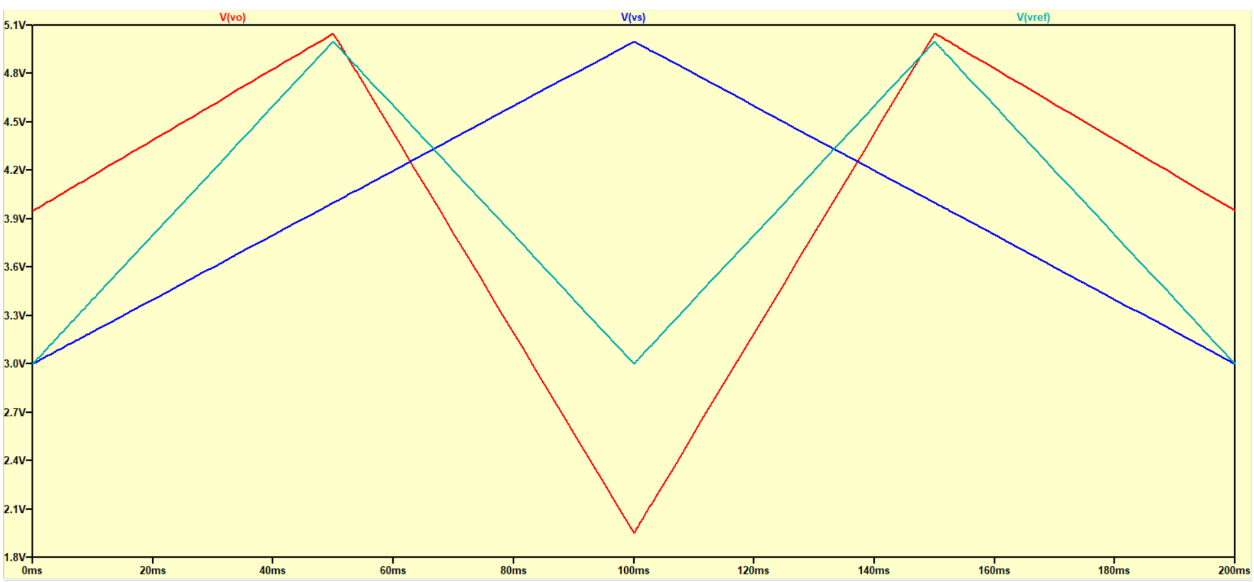
This is a difference amplifier. It should be taking the difference of these input voltages. However, as we can see in the following simulation, our output is saturating:

Normal Ground Output



However, with the virtual ground, we can see we get a clean output and no saturation:

Virtual Ground Output

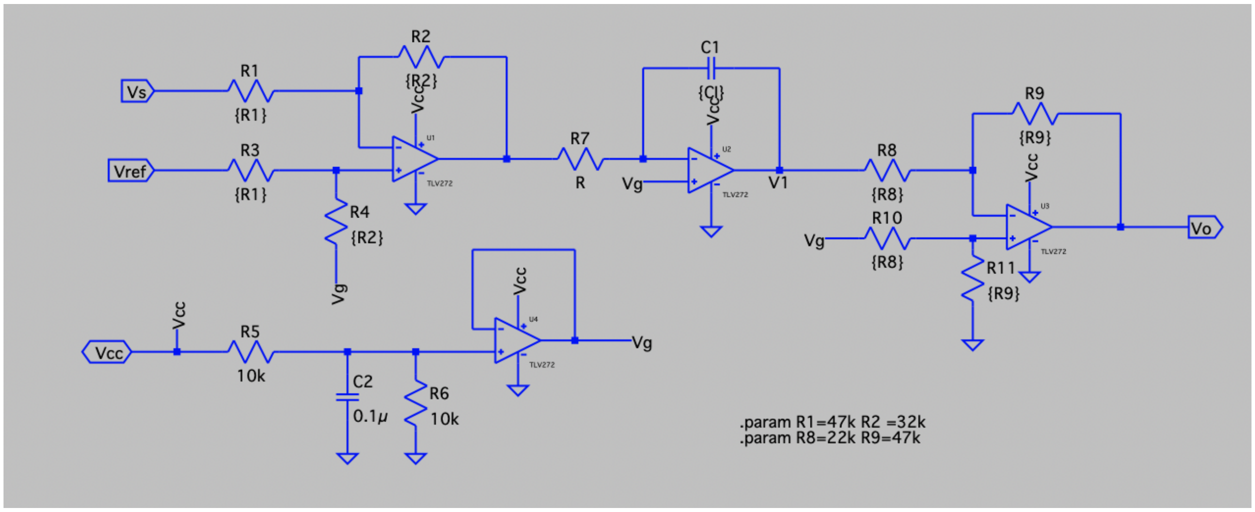


The virtual ground basically changes our ground reference. In this specific example, our virtual ground is a voltage divider of Vcc between two 10k Ω resistors, so our virtual ground is 4 V. This is also the same virtual ground we use in our actual compensator circuit.

3.B.3 - I-Compensator Circuit in LTspice

For the rest of this lab we primarily focused on the compensator circuit, a feedback loop circuit used to maintain the speed of the motors, as external factors often effect the overall speed. With two input voltages, Vs and Vref, (voltage of speed sensor output and a reference voltage respectively) the compensator compares Vref, which is obtained directly from the motor control circuit with the potentiometer, to the speed sensor circuit output. The compensator helps maintain the desired speed by using three blocks, all of which use a virtual ground of 4V. The first block takes the difference between Vs and Vref, and is offset by 4V by the virtual ground. The second integrates the result, once again offset by the same 4V virtual ground. The last block takes the difference between the integrated output and the 4V virtual ground to essentially “return” ground back to 0V. This was our circuit in LTspice:

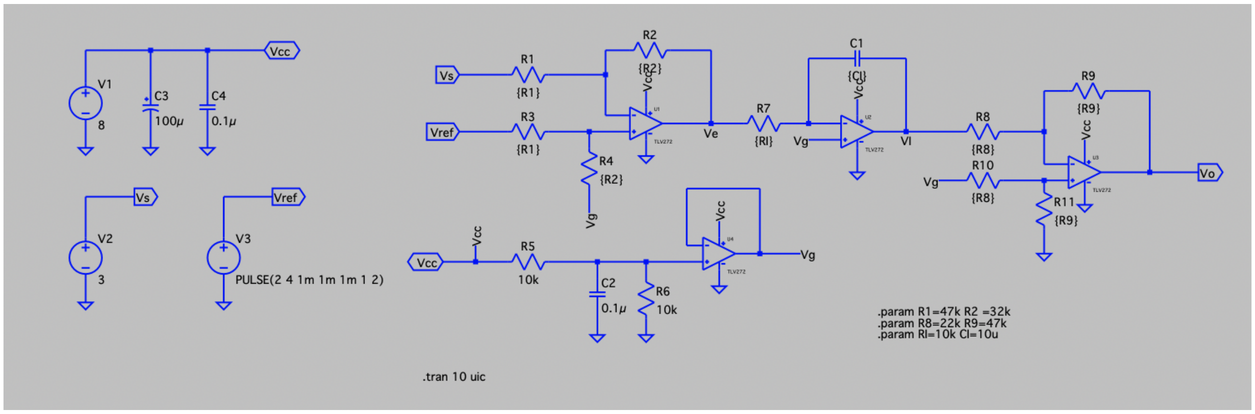
Compensator Circuit LTspice Simulation



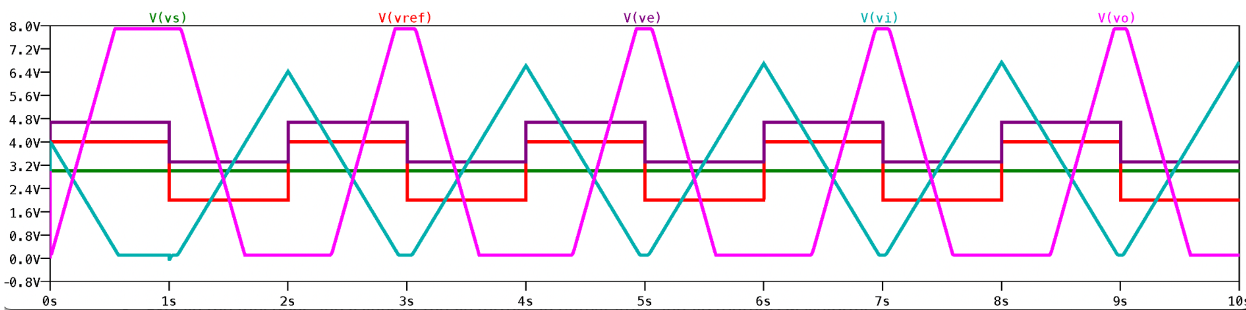
3.B.4 - Compensator circuit test (open-loop) in LTspice

To ensure each part of the compensator was working properly we needed to check each output to see that the results were what we expected. We tested the compensator as an open loop with a PWM input for Vref and a DC input of 3V at the speed sensor. For the first block output, Ve for error, (between Vs and Vref) we expected to see slightly higher PWM as it simply takes the difference between a constant voltage and a PWM and then is shifted up 4V. At the integrator output, Vi, we expected to see a triangular shape as the integral of a straight horizontal line is a straight diagonal line, so the integrated rectangular PWM should take the form of triangles. Finally, taking the difference between the output of the integrator and the virtual ground shifts ground back to 0V instead of the virtual ground of 4V which in turn increases the magnitude of the Vi so much so that the output saturates at Vcc (8V) and at real ground (0V). Below is the circuit and the results of its simulation:

Open Loop Compensator Circuit



Open Loop Compensator Circuit Outputs



3.B.5 - Closed-Loop with I-Compensator Circuit in LTspice

Since we knew that our open loop compensator was working, we needed to simulate the compensator on the actual speed sensor and motor circuit. Before doing this, we had to change Ri and Ci, the resistor and capacitor used in the integrator block. This is necessary because we wanted to obtain a damping constant (ζ) as close to 1 as possible. We know from a previous experiment the most reasonable motor parameters consisted of armature resistance was about 2.5716, k of about 0.3244, and J of about 0.0063. This was necessary because in the prelab we derived that Ri*Ci = $1.28\zeta^2 G_0/\omega_m$ where G_0 = 1/k and $\omega_m = k^2/RJ$. Solving for ζ we found it to be ideal to have a time constant of about 0.6s.

R_I and C_I Calculation

$\zeta = \frac{\omega_m}{2\sqrt{\frac{K_{sense}}{R_I C_I} G_o \omega_m}}$, $\zeta = 1$

$R_I C_I = \frac{4K_{sense} Go\, \omega_m}{\omega_m^2}$, $K_{sense} = 917 * t_{on} = 0.32$

$R_I C_I = \frac{1.28 Go}{\omega_m}$, $G_o = \frac{1}{k}$, $\omega_m = \frac{k^2}{RJ}$

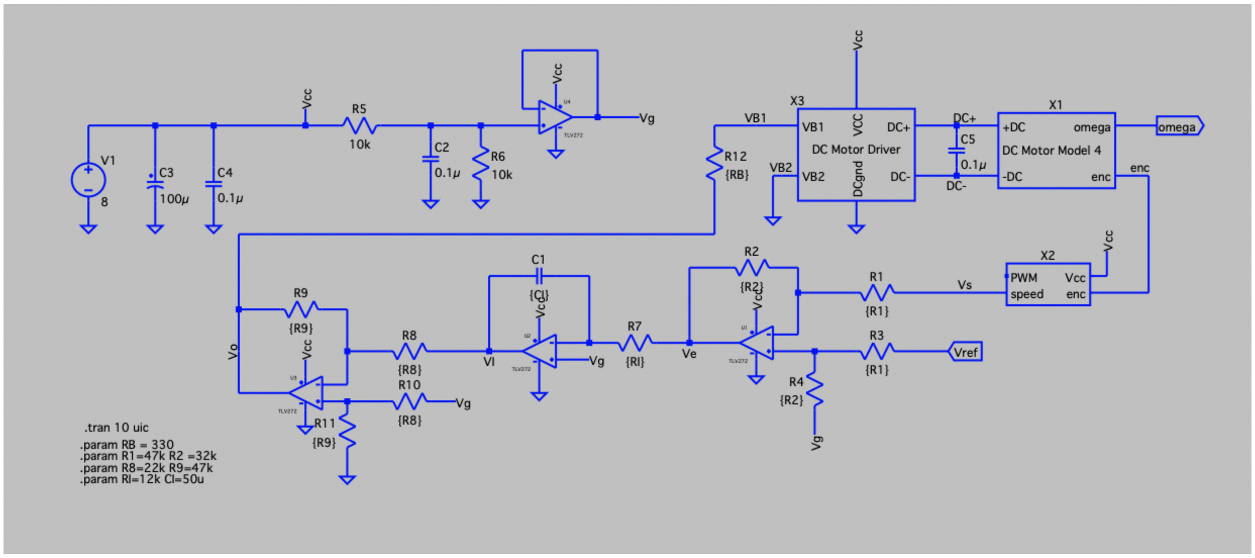
$R_I C_I = \frac{1.28 RJ}{k^3}$, $R_m = 2.5716$, $J = 0.0063$, $k = 0.3244$

$R_I C_I = 0.6075$, choose $C_I = 50\ \mu F$

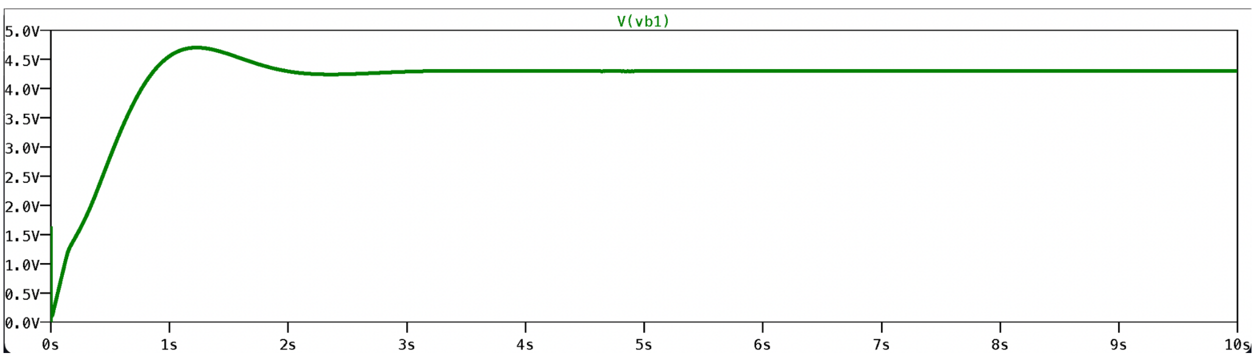
$R_I = 12150\ \Omega$

We realized this would change depending on how ideal each of our circuits were, and on our different motor parameters. For the simulation though, we decided to use a 12kΩ resistor and a 50μF capacitor to obtain the 0.6 second time constant. Below is our circuit and the final output of the compensator (Vout) which goes into a 330Ω resistor then into the previously built motor driver (at VB1):

Closed Loop Compensator Circuit



VB1 Voltage for Powering Motor



The result looked reasonably close to critical damping, so we concluded the values we found for Ri and Ci made sense to go off of for building the real circuit.

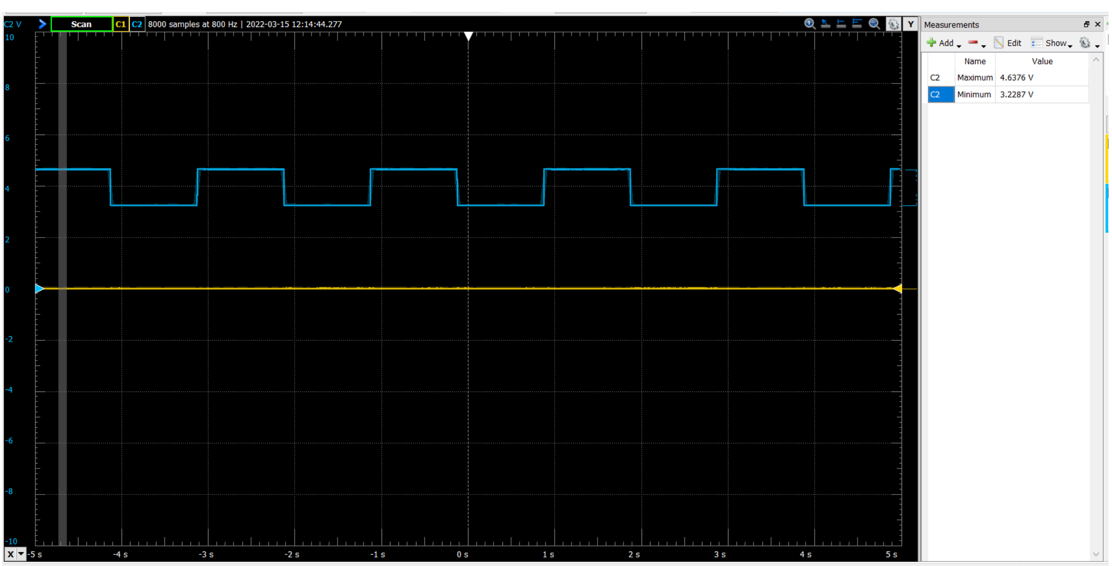
3.B.6 - Build Compensator Circuit

This experiment called for simply building the compensator circuit. This was done using two TLV272 opamps, which contain one opamp per side. For R_I, a 10k Ω and 2.2k Ω resistor were used in series in order to achieve our desired resistance of 12150 Ω. For our capacitor, we used two 100 μF resistors in series with the negative ends facing each other in order to create a non-polarized capacitor with an effective capacitance of 50 μF. Testing of this circuit was done in the following experiments.

3.B.7 - Compensator implementation test (open-loop)

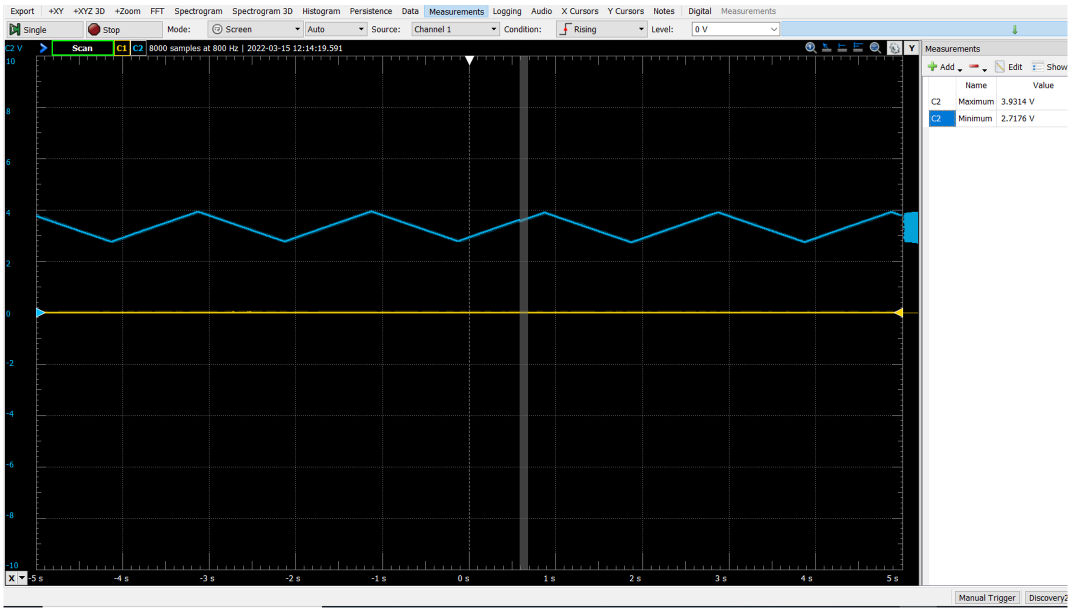
This experiment involved testing the compensator before connecting it to the other parts of our circuit. In order to perform this test, we generated a fixed 3 V Vs and a rectangular Vref that oscillated between 2 V and 4 V, and compared the resulting V_E, V_I, and V_o waveforms to our LTspice simulations from experiment 3.B.4. Below are the resulting waveforms we saw:

Hardware Ve Measurement



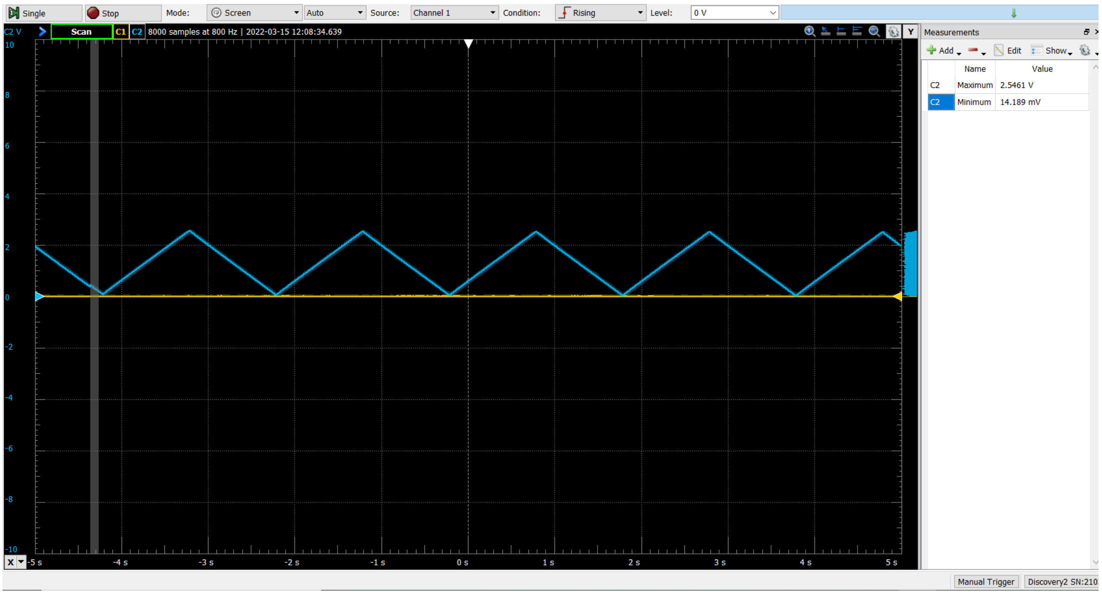
As we can see, our Ve is a rectangular waveform that oscillates between about 4.6 V and 3.3 V. This is very similar to what we expected from simulation, which is a square waveform oscillating between 4.7 V and 3.3 V. So, this is a promising sign that our first difference amplifier and our virtual ground are working properly.

Hardware V_I Measurement



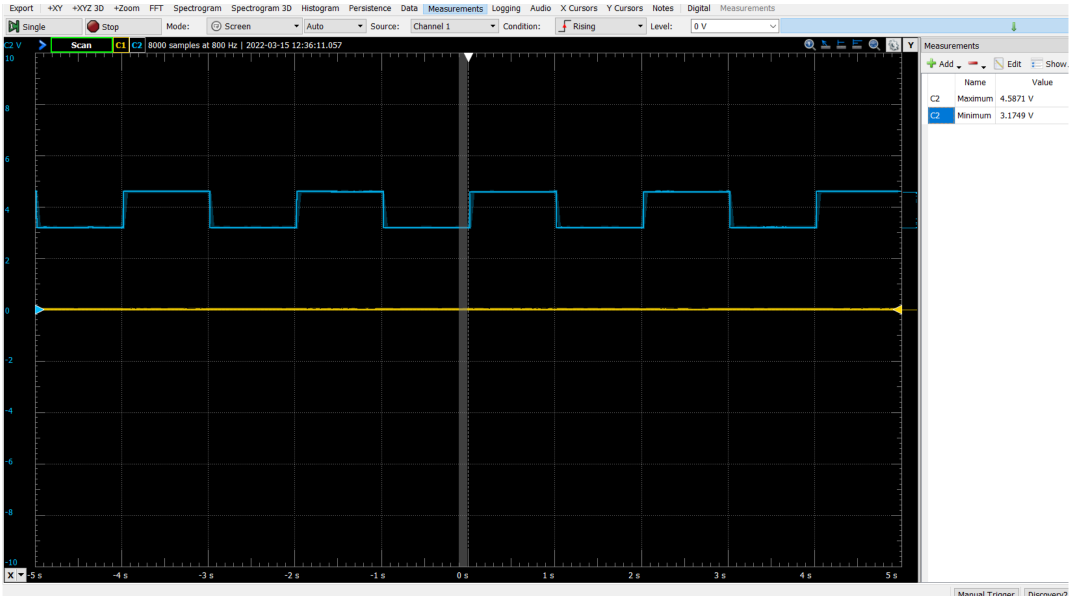
Here we can see that our V_I waveform is a sawtooth shape that oscillates between about 3.9 V and 2.7 V. Our LTspice simulation, as previously mentioned, is an identical waveform that oscillates between 4 V and 2.85 V. Evidently, our measured output is very similar to our simulated output, indicating that our integrator is working properly.

Hardware V_o Measurement



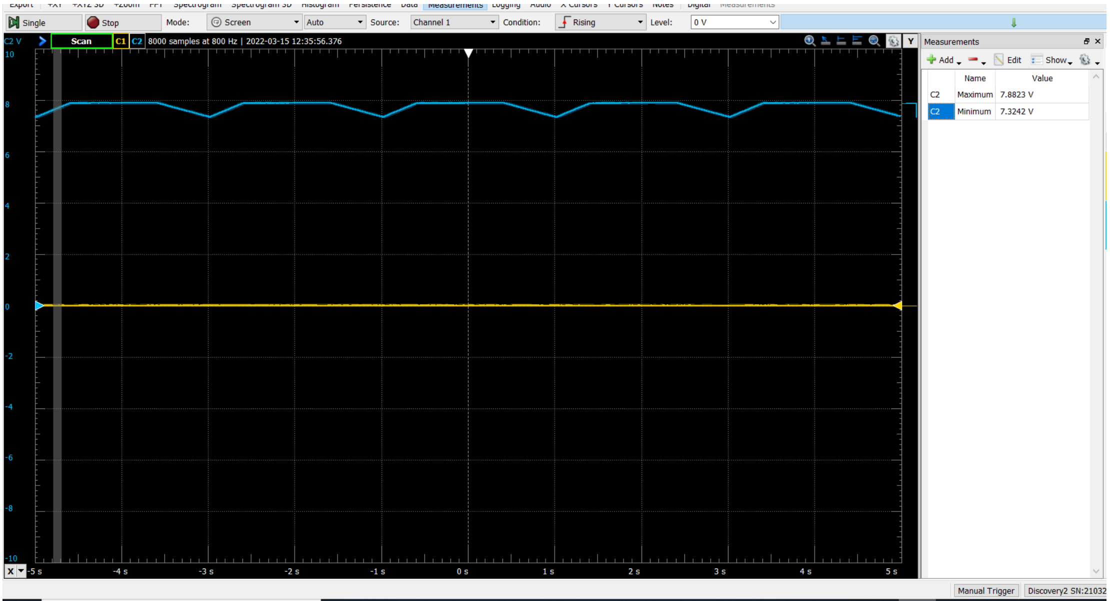
Lastly, we have our V_o waveform. This is also a sawtooth type waveform, however this time oscillating between 0 V and a max of about 2.5 V. Our LTspice is of the same shape and oscillates between 0.1 V and 2.47 V, giving us confidence that our entire compensator is working properly. We next changed V_s to 2.8 V while keeping V_{ref} the same in order to observe what happens to V_o, V_I, and V_E. Below are the resulting waveforms:

Hardware V_E with 2.8 V V_s



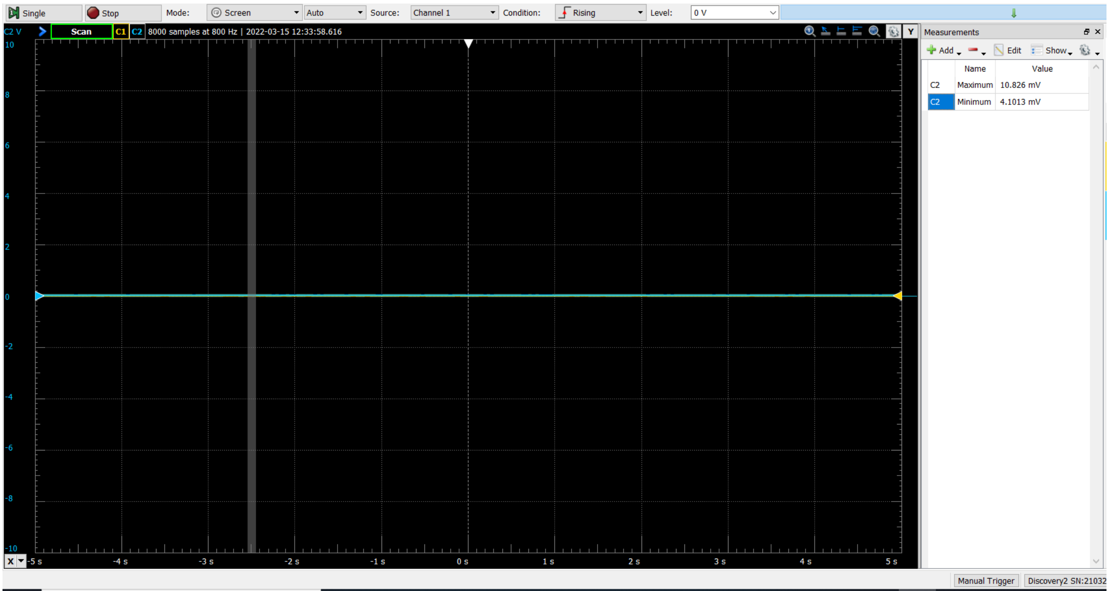
As we can see, V_E stays nearly identical, although the min voltage has dropped by about 0.1 V and the max voltage has dropped by about 0.05 V. This tiny change would make sense as our differential amplifier is dependent on V_s, and a tiny change to it would likely have a tiny change on its output.

Hardware V_I with 2.8 V V_s



Our waveform recorded at V_I, however, has had quite a drastic change. This may be occurring as the compensator is trying to overcorrect for the incorrect speed voltage, causing it to basically rail out at 8 V.

Hardware V_o with 2.8 V V_s



Here our output waveform has dropped to zero volts. We believe this is also due to the compensator overcorrecting and the compensator output railing out at 0 V.

3.B.8 - Test Closed-loop with Compensator Implementation

This experiment called for us to connect our integrator to the rest of the circuit. A few different changes had to be made in order for this to be done. Instead of connecting our V_s input to the compensator to a 3 V DC waveform, we now connect it to the speed sensor output on our breadboard. The V_{ref} would also be changed from a wavegen input to an input from our potentiometer circuit, allowing us to control V_{ref} by twisting the potentiometer knob. Lastly, our motor driver would have one base now connected to ground, and one base connected to the output of the compensator circuit. After finishing these changes, we first tested our circuit by turning on and checking that when twisting our potentiometer knob, the speed of the wheels would change. After observing that this does indeed occur, we took measurements at V_{ref} and V_s and compared them:

Comparing V_{ref} and V_s at Different Wheel Speeds

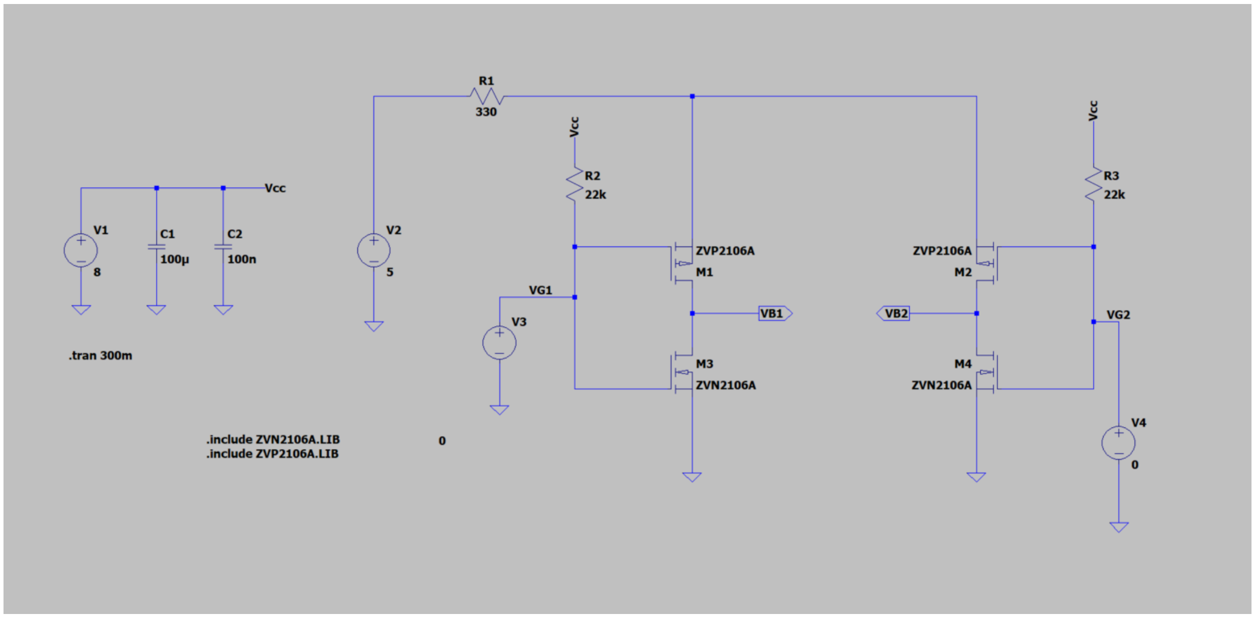
V_{ref} (V)	V_s (V)
3.52	3.5
2.8	2.82
2.1	2.02
1.4	1.41
0.98	0.98

As we can see, our V_{ref} is nearly identical to our V_s, indicating our circuit is working properly.

3.B.9 - Direction Control in LTspice

This experiment required us to build and test the direction control circuit in LTspice. Below is the LTspice circuit we created for this test:

LTspice for Direction Control



The V2 source in the schematic is the V_B, which comes from the output of our compensator. For the purpose of this test, we fixed it at 5 V. The p channels mosfets are active low, and the n channel mosfets are active high. So, when V_{G1} is high, we should see that current flows from V_B to V_{B1}, indicating that we are in the forward direction as V_{B1} will in turn be driven high. This is because the p channel mosfet connected to V_{B1} will be on (since its gate is at 0 V). V_{B2} will be 0 V, as it will be shorted to ground through its n channel mosfet. Since the p channel transistors are active low, they will be default off since V_{G1} and V_{G2} are connected to V_{cc} through the 22k Ω pull up resistors. We can then intentionally ground them in order to turn on the respective n channel mosfet. Below is a table summarizing this.

Measured V_{B1}, V_{B2}, and V_B (Output from Compensator)

V_{G1}	V_{G2}	V_{B1}	V_{B2}	V_B

0	7	5	0	5
7	0	0	5	5

As we can see, when gate 1 is low and gate 2 is high, V_{B1} will be driven high, indicating we are in the forward direction. When gate 2 is low and gate 1 is high, we will instead see V_{B2} is driven high, indicating we are in the backward direction. In order to test this in hardware, we will use a power source in order to act as our compensator output. We will step this down for three different voltages with one gate grounded, and take measurements of the voltage at the drain. We then ground the other gate and perform the same three measurements.

3.B.10 - Add Direction Control to Closed-loop Circuit Implementation

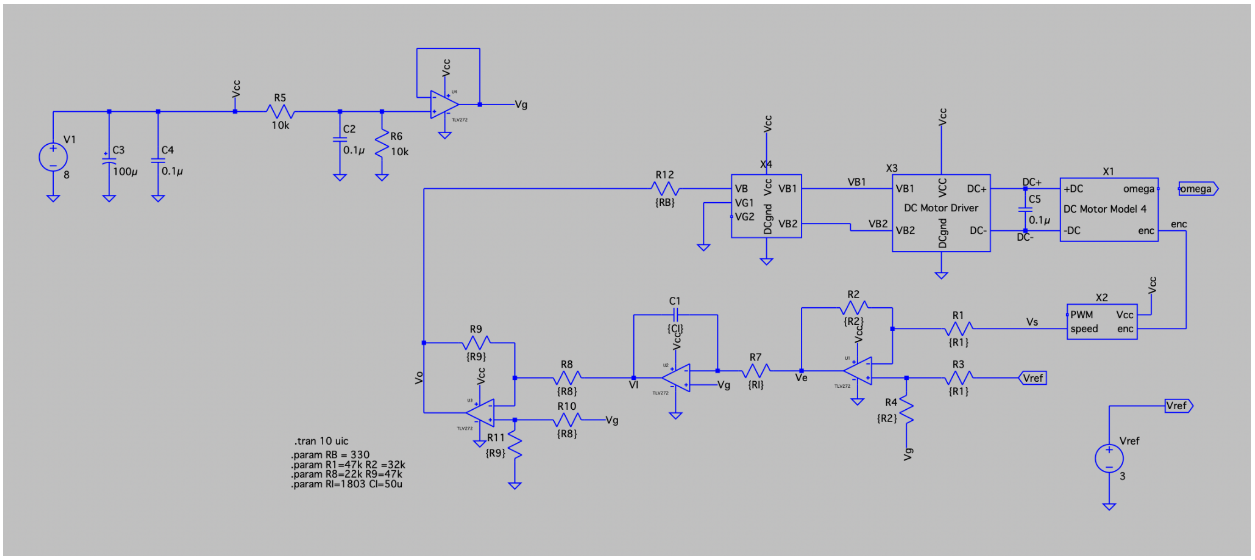
This experiment required us to build and test the direction control circuit on our breadboard. This was a relatively simple circuit to build, and we did not run into too many problems. In order to test the circuit, we followed our plan from 3.B.9. Our compensator output was replaced with a voltage source from the power supply, and our bases were replaced with probes in order to read their respective voltages. Below is a table of the measurements:

V_{G1} (V)	V_{G2} (V)	V_{B1} (V)	V_{B2} (V)	V_B (V) (from power supply)

0	7	6.477	0.7	7.013
0	7	4.13	0.5	4.5
0	7	2.4	0.4	2.9
7	0	0 V	6.543	7.139
7	0	0 V	4.217	4.63
7	0	0 V	2.412	3.186

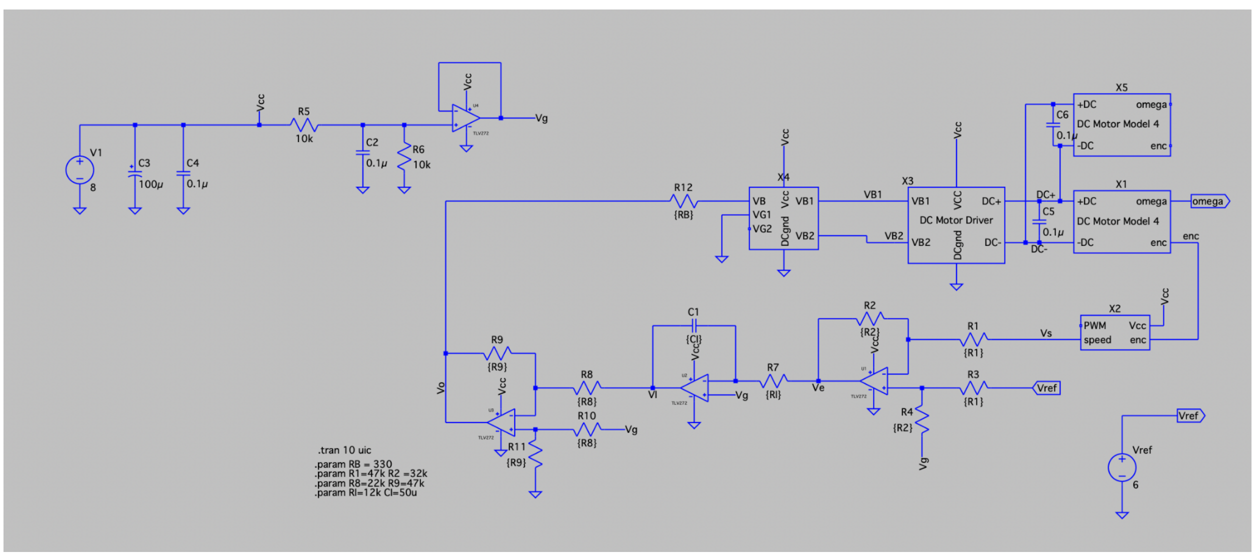
As we can see, our results are about what we expected and similar to our simulation data. When gate 1 is low, V_{B1} is high and when gate 2 is low, V_{B2} is high, as we anticipated. However, we also see that when gate 1 is low and gate 2 is high, V_{B2} has a little bit of voltage, which is unexpected. It is not enough to drive our motor, however, so we decided not to worry about it. Overall, it appears that our direction circuit is working as intended, so we can connect it with the rest of our circuit. The LTspice circuit including direction control is shown below:

3.B.10 LTspice Schematic



3.B.11 - Test Overall Speed Control Loop Implementation

3.B.11 LTspice Schematic for Verifying Results



This experiment involved implementing our final completed circuit. Our direction control was connected to our compensator output via the V_B node, and our motor bases were connected to the respective drains of the two sides of the direction control circuit. We then took several data points in order to see if our circuit was behaving as intended:

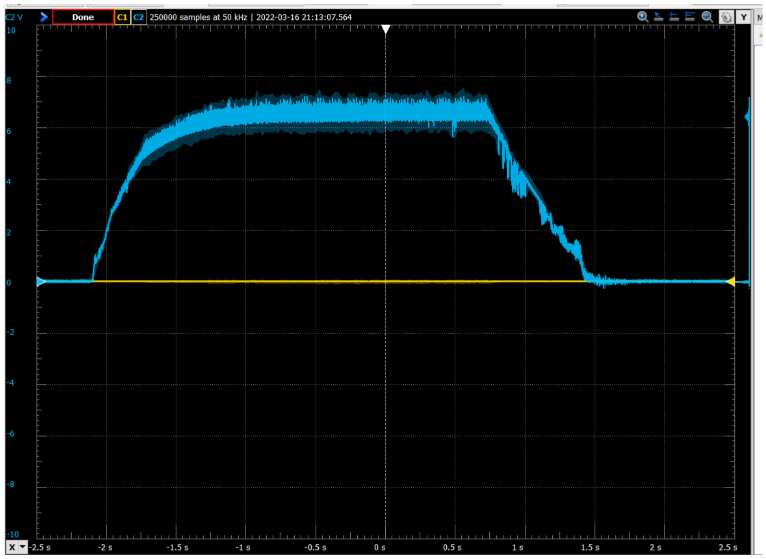
Voltages at V_{ref}, V_E, and V_I

V_{ref} (V)	V_E (V)	V_I (V)
7.8	5.843	0
3.27	3.858	1.6
1.74	3.901	2.488
0	3.806	8

As seen in the table, we measured the V_E and V_I values using a scope at varying V_{ref} voltages. As V_{ref} decreased, we can see that V_E stayed roughly the same, while V_I increased. Evidently, V_{ref} and V_I appear to be almost inversely proportional.

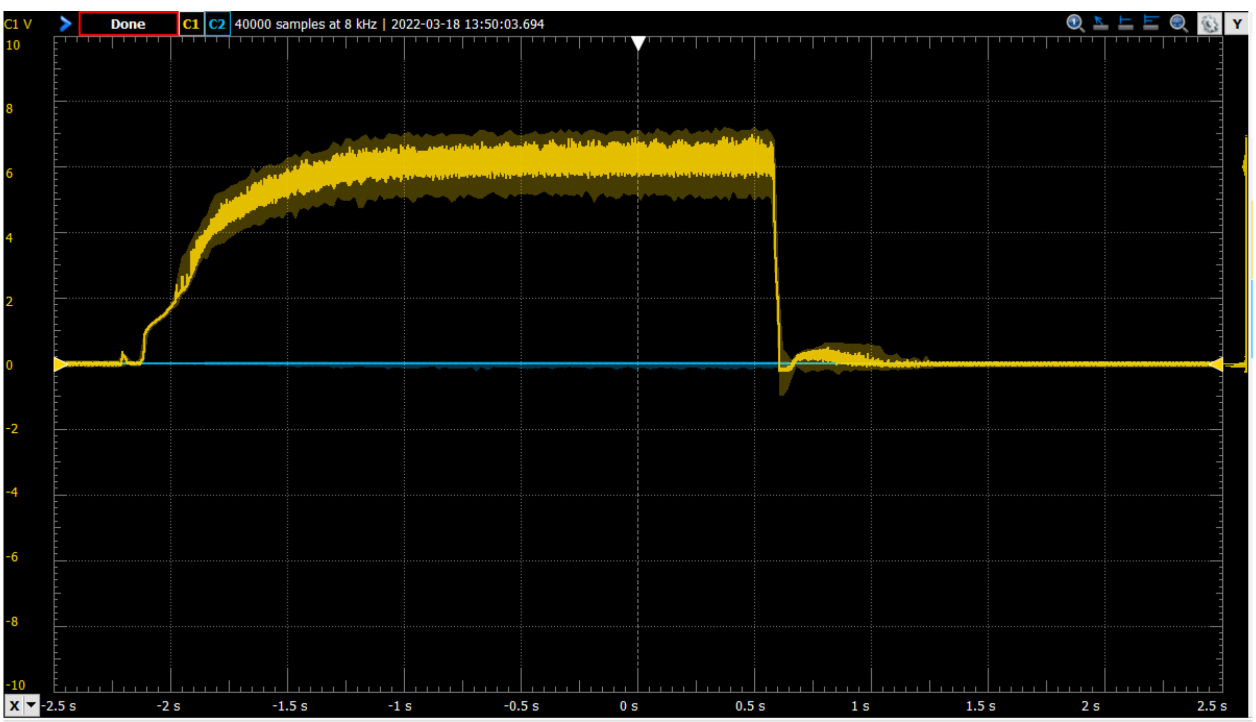
We next tested the static properties by turning the motor on then quickly off and recording the transient response of the motor. Below is the resulting output:

Static Test In Forward Direction

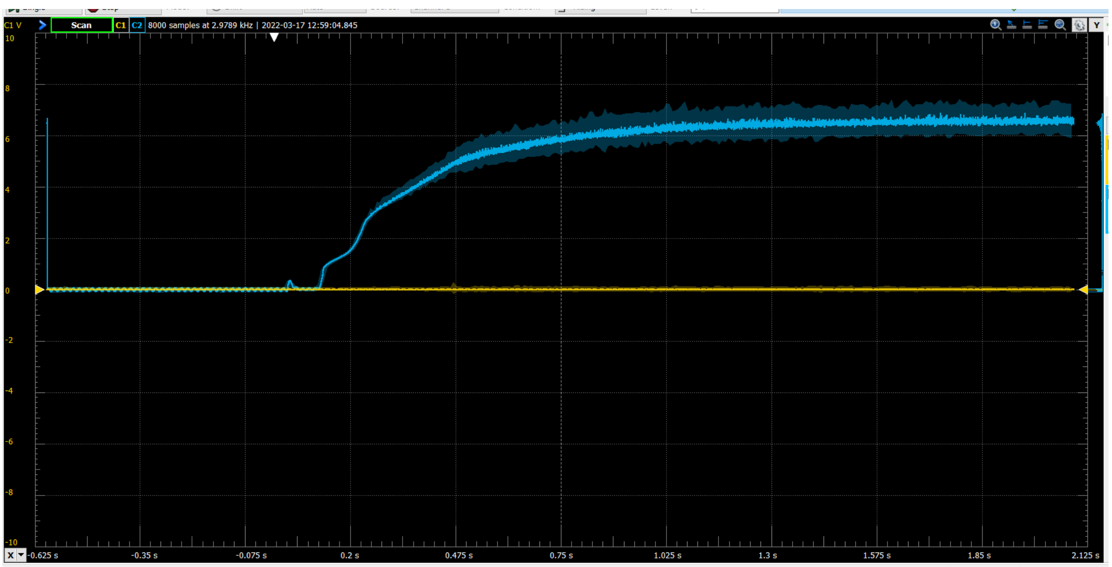


We also performed this test in the backward direction, shown below:

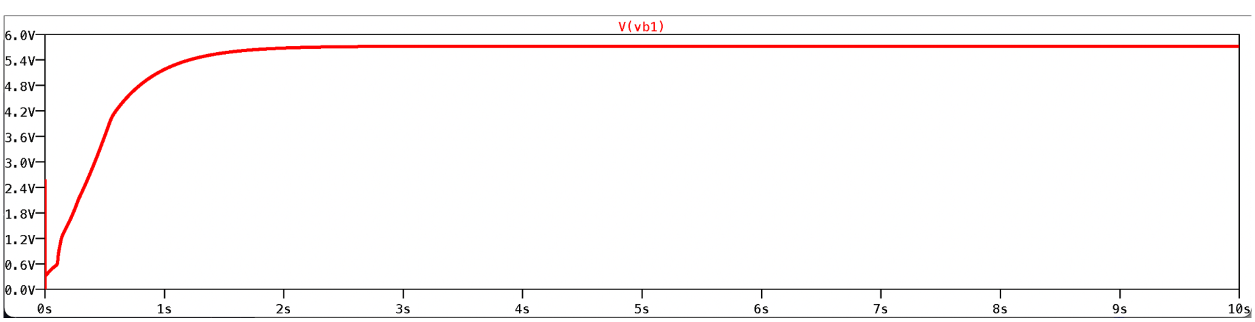
Static Test in the Backward Direction



As we can see, both our forward and backward direction yield similar results, indicating both directions are working. The forward test was done by turning the potentiometer up and down, whereas the backward test was done by shutting the power supply off completely, which is why our backward test shows a much quicker turn off time. We also did a transient analysis at the motor base voltage after turning on our circuit:

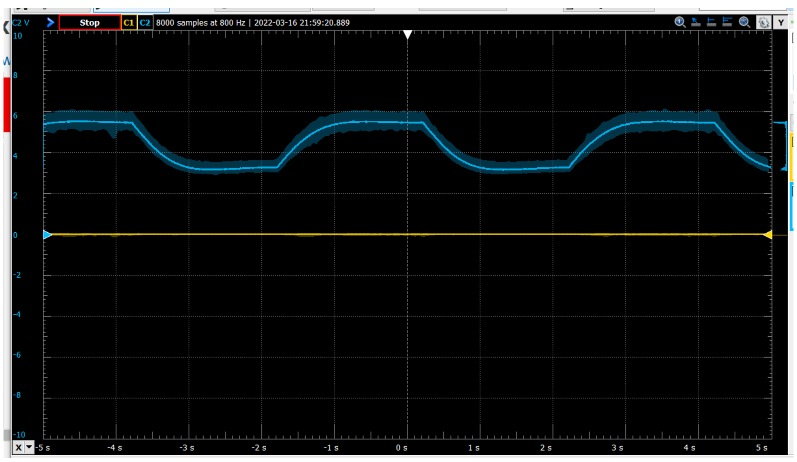


LTspice For Static Test With 6V for Vref

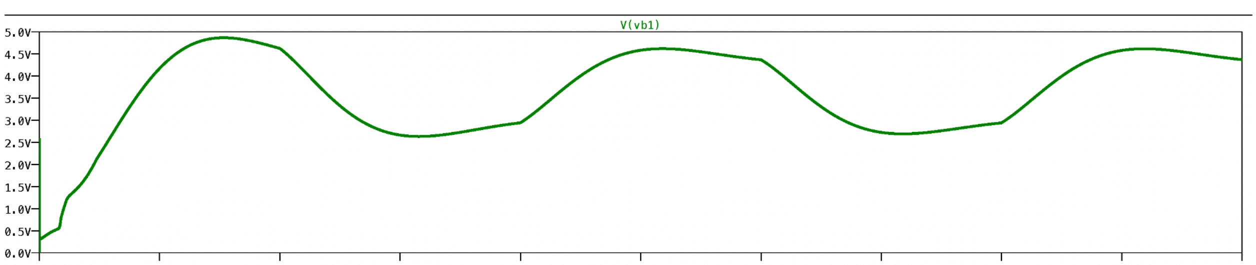


As we can see from the above waveforms, our motor responds quickly and turns on and off reasonably quickly, and is able to go continuously from a standstill to maximum speed. Our measured waveforms are also very similar to our LTspice waveforms. This gives us confidence in the effectiveness of our circuit. We next tested the dynamic properties by using a square wave from 2 to 4 V with a period of 4 s as our V_{ref} and recording the motor's response:

Dynamic Test

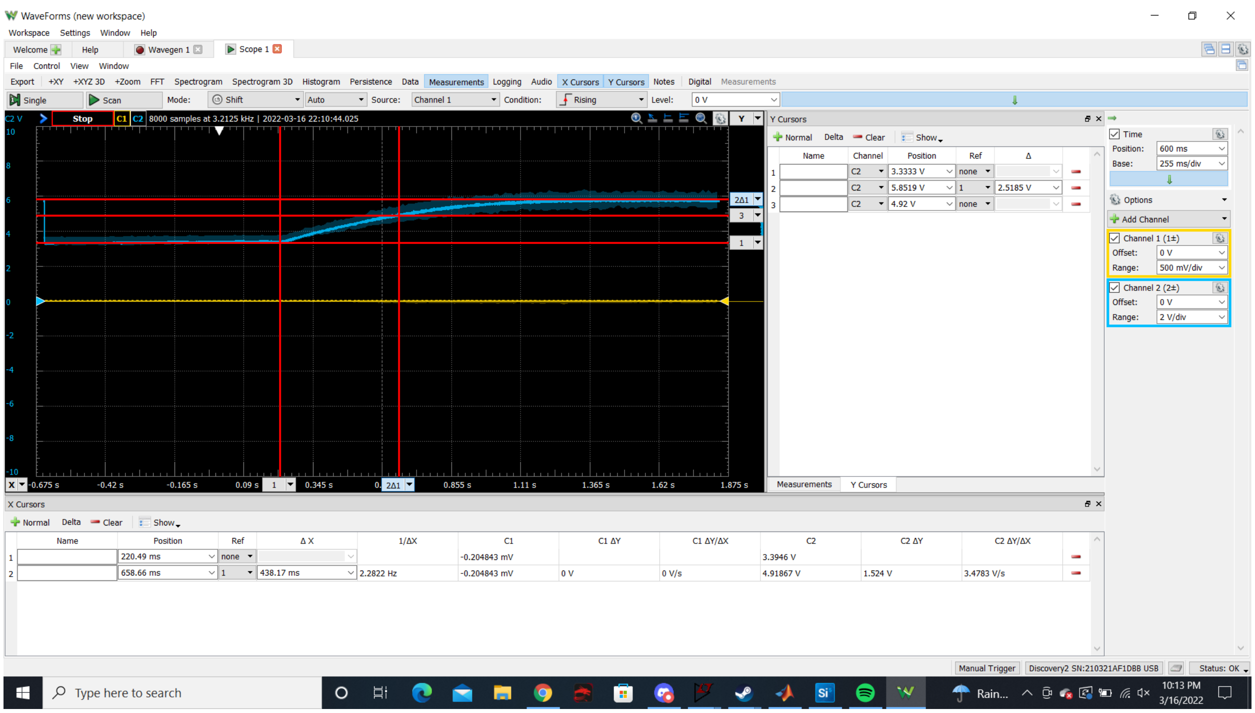


LTspice For Dynamic Test With Pulse Between 2 and 4V for Vref



Our motor responds well to this varying input, indicating that our circuit is working well and not significantly over or underdamped. From this waveform, we are able to measure a time constant, allowing us to estimate ζ. Below are our measurements for time constant:

Time Constant Measurement



The time constant is found by measuring from the start of the rising wave to about 63% of the peak voltage. So, we took the difference between our max and min voltage in this wave, then placed an x cursor at the start and an x cursor at the equivalent y value of 63% of this difference. The difference between the two x cursors gave us 438 ms, which is relatively close to our intended time constant of 608 ms. Using this measured time constant, we can use our previous equation for finding R_I and C_I to instead calculate the actual ζ value.

Measured ζ

$$\zeta = \frac{\omega_m}{2\sqrt{\frac{K_{sense}}{R_I C_I} G_o \omega_m}}$$

$R_I C_I = \frac{\zeta^2 1.28RJ}{k^3}$, $R_I C_I$ = 438 ms

$0.438 = 0.608\zeta^2$

$\zeta^2 = 0.721$

$\zeta = 0.849$

Our estimated ζ value is slightly under the target value of 1, however we have decided that it is close enough and will keep it unless we run into issues in the future. These tests all indicate that the circuit is performing reasonably well compared to our simulations.

Discussion and Conclusion

Overall, the lab objectives were met. We were able to build all components of the robot and match them with the LTSpice simulations that we created, so we can say that the lab objectives that were laid out in the introduction were met.

The main conclusions that can be drawn are that an H-bridge circuit can be used to control the direction of a DC motor, and the feedback control system is very useful for reducing any error in a circuit that comes from an external source.

The only limitation I can think of is the construction of the H-bridge, it seemed simple at first but was a lot more time consuming and soldering onto a protoboard was not a very intuitive way to go about it, but other than that there was not much else that could have been seen as a limitation.

One important thing that I learned from this lab was the function of the H-bridge and how it acts as motor control. Studying this also helped further solidify my understanding of the functionings of transistors. Another important thing was how to construct a feedback control system and how they function. Although I would like to further understand the design process and the exact workings of the compensator circuit, I did learn a lot of important things, one of them being how to select resistor and capacitor values to keep the damping ratio of the circuit within a reasonable range.