Lab 3

ECEN 2270
University of Colorado Boulder

March 18, 2022

Luke Hanley
Nicholas Haratsaris
Ginn Sato

Introduction

In this lab, we build upon previous labs in that we will be using the output of our speed sensor circuit. Thus, we must first construct a circuit that drives both motors (one one side), and that also can be implemented to drive forward and reverse motion. Additionally, we construct a feedback control system that takes one input, and uses the encoded output to maintain the speed of the motors. Overall, the goal of this lab is to create a safe speed control circuit.

The first part of the lab focused on creating the motor driver circuit. The objectives here were to simulate this circuit effectively, build and test it on our breadboard, and finally to solder it on a perf board and affirm its functionality. Utilizing the perf board allows us to store this circuit “inside” the robot. Additionally, it is important to test the circuit on the breadboard before soldering the perf board, because desoldering and debugging on a soldered per board is very difficult and tedious.

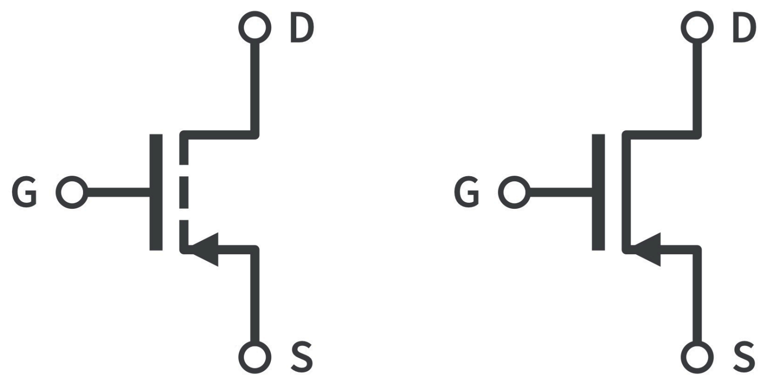
The second part of the lab is focused on creating the feedback control system, given that we already have our speed sensor circuit and encoded output Vs. The objectives here are to simulate this circuit in both open loop and closed loop modes, build and test open and closed loop modes, and finally to add a circuit that implements direction control and feeds its outputs to the motor driver circuit from part A.

Equipment List

- Texas Instruments TLV272 Operational Amplifier
- Fairchild Semiconductor MJE200 NPN Transistor
- Fairchild Semiconductor MJE210 PNP Transistor
- Diodes Incorporated ZVN2106A N-Channel MOSFET
- Diodes Incorporated ZVP2106A P-Channel MOSFET
- Sparkfun Resistor Kit
- Sparkfun Capacitor Kit
- Digilent Analog Discovery 2
- DC Power Supply
- Fully Assembled ROB 0025 Robot
- Jumper Wires and Solderless Breadboard
- Perfboard
- Soldering Equipment
- MATLAB Software
- LTSpice Software

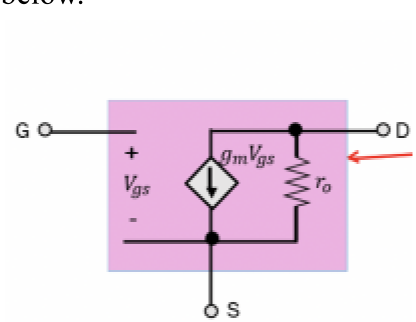
3.A Lab Exploration Topic

1) MOSFETS come in two types, depletion type and enhancement type. For these, both N-channel and P-channel transistors can be used.
2) Here is a common schematic for the MOSFET. Depletion type is on the right, and enhancement type on the left.



When the V_{GS} of the EMOSFET is above the threshold voltage the EMOSFET is activated. This turns it into a constant voltage source. Otherwise the drain current is zero and there is no induced channel.

3) These Mosfets are made from semiconductor material, primarily and most commonly silicon. The properties of these materials are typically non conductive, however when introducing an impurity their conductivity increases significantly. The physical structure of these is an either n-type or p-type body that is determined by the charge on it, with drain, gate, and source electrodes on top. The structure is strategically made to take advantage of the properties of semiconductors.
4) There are three operating principle of mosfets which are determined by the subthreshold, cutoff, and weak-inversion modes. These determine the functionality of the mosfet at different voltage levels.
5) A simple equivalent circuit would just include an input voltage Vgs, and a voltage controlled current source, controlled by Vgs. The current source would be connected between the nodes Vd and Vs in parallel with a resistor. A circuit schematic is shown below.



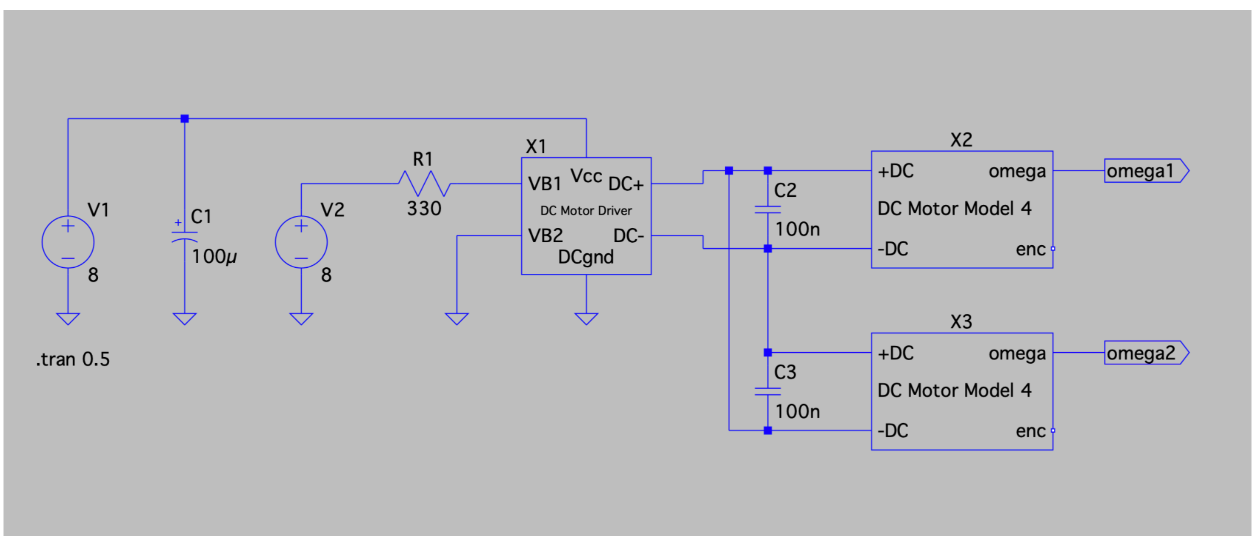
6) Essentially, MOSFETS are voltage controlled current sources, whereas BJTs are current controlled current sources.

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

	R_M	V_{B1}	V_{B2}	DC+	DC-	I_{RM}	P_{Q1}	P_{Q2}	P_{Q3}	P_{Q4}
Forwards	1.5 Ω	3.53	0	2.77	0.764	1.337	6.93	0	0	1.02
	10 Ω	6.37	0	5.65	0.720	0.492	1.15	0	0	0.355
Backwards	1.5 Ω	0	3.83	0.96	2.87	1.27	0	6.47	1.21	0
	10 Ω	0	6.48	0.93	5.54	0.45	0	1.10	0.43	0

3.A.3 Simulate the Motor Driver Circuit

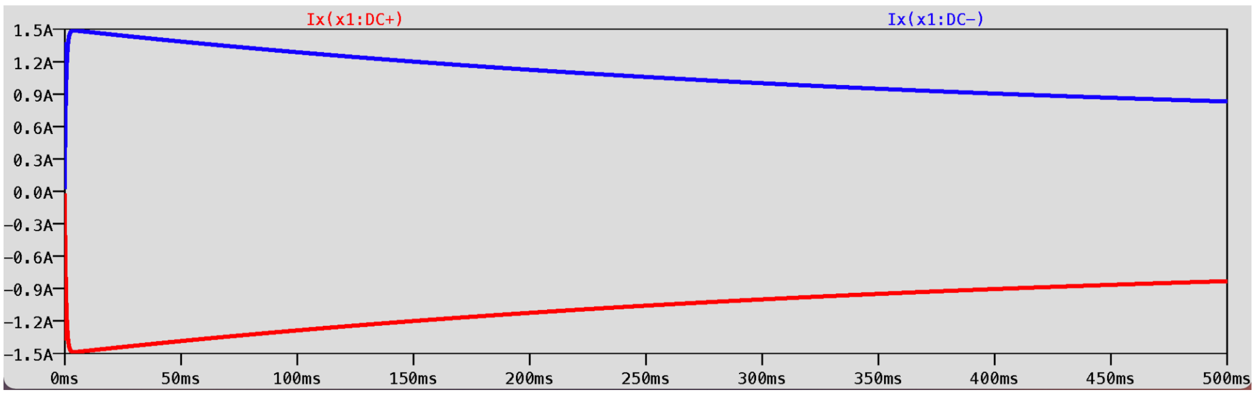
After building our DC Motor Driver schematic, we use the DC Motor Model 4 schematic from a past lab to simulate the entire circuit in LTSpice. The physical reason X2 runs forward and X2 runs backward is that inside our robot chassis, the motors are mirrored from the left side to the right side, so if a forward direction for the left side is CCW then a CCW rotation on the right wheel correlates with the backward direction. Thus, we have to invert the wiring between the two schematics to produce a uniform direction, despite our LTSpice voltage indicating otherwise.



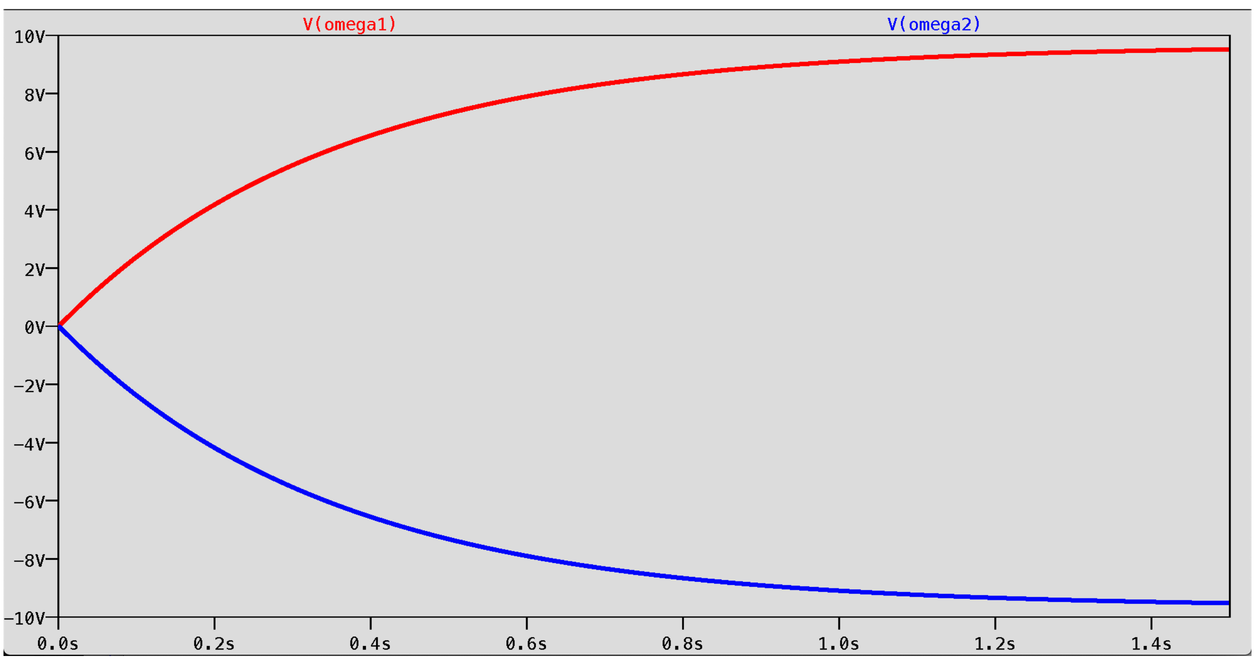
Motor Driver Circuit in LTSpice

We also simulated the speed of each motor by looking at the omega1,2 outputs of our DC Motor Model 4 schematic. We can see both the transient domain as well as the maximum speed each motor approaches. As we explained earlier, the X3 speed correlated voltage is negative, but we know that because of how the motors are oriented, this is in fact a positive, forward speed.

Case I) No Motors Stalled

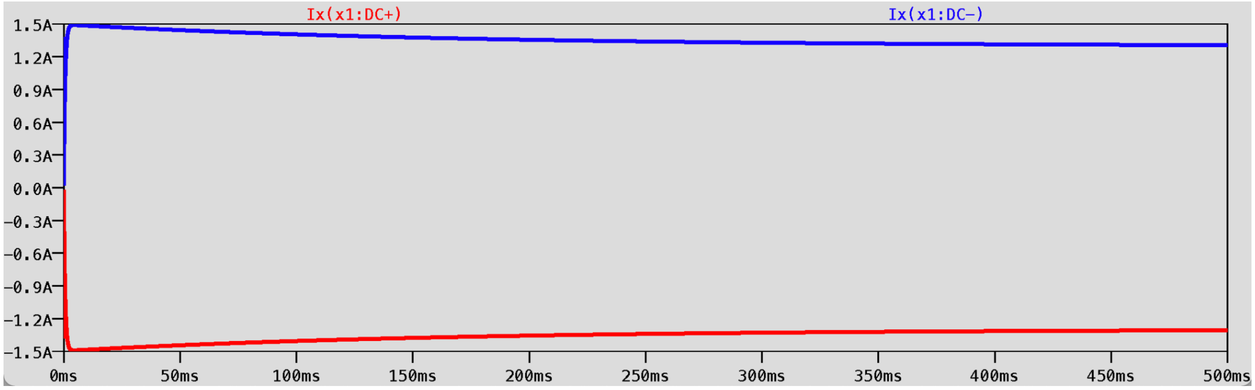


Transient Motor Current Response with both Motors running

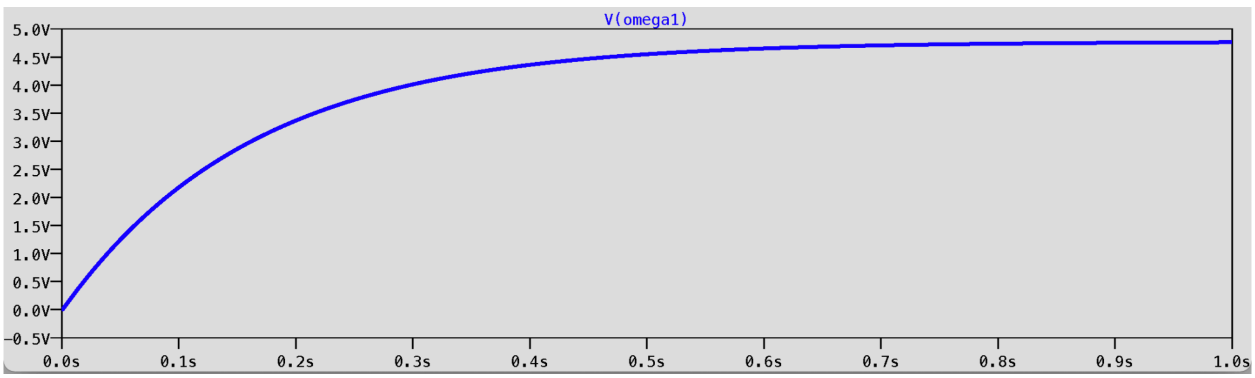


Omega 1 and Omega 2 with both Motors running

Case II) One Motor is stalled

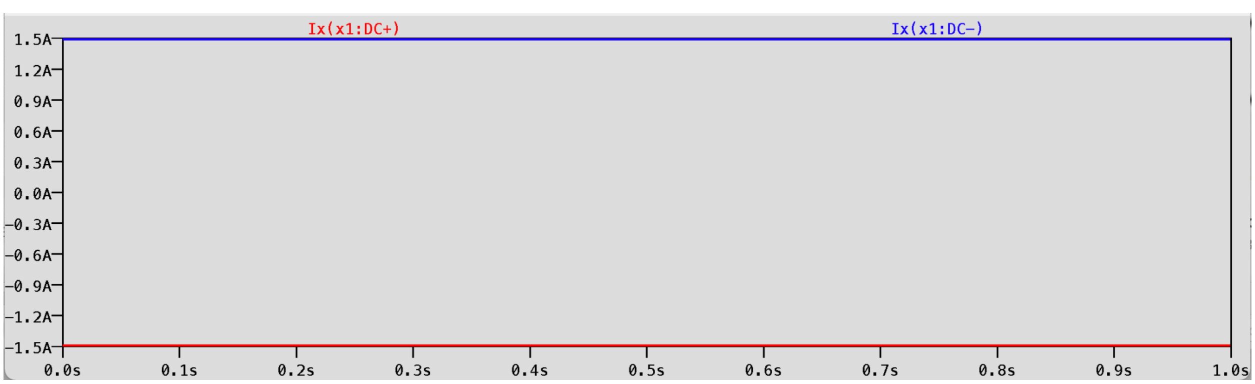


Transient Motor Current Response with one Motor Stalled



Omega 1 with Omega 2 Stalled

Case III) Both Motors stalled



Transient Motor Current Response with both Motors stalled

It is somewhat redundant to show a plot of V(omega1) and V(omega2) here because they are both zero. We stall the motor by forcing the wheel speed to be zero, which we do in spice by grounding one or both nodes.

3.A.4 Testing Motor Driver Circuit on Breadboard

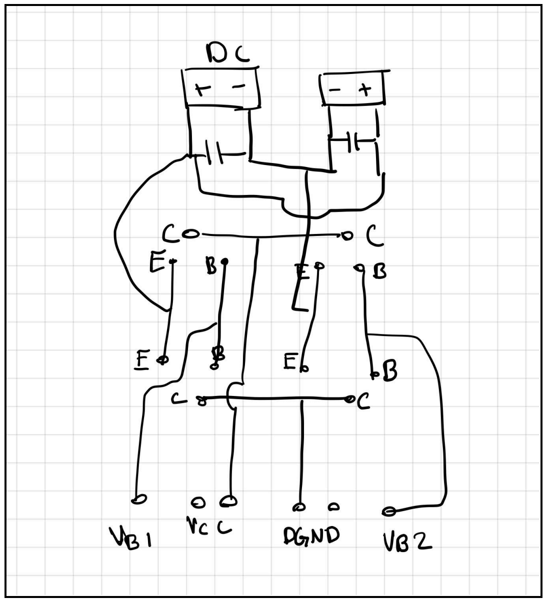
After building the half H-Bridge for our motor driver circuit using the values R_B =330Ω and and R_1=0.375 Ω which comes from three 1.5 Ω resistors in parallel. We measure the V_I of our motor driver circuit and use Ohm's Law to derive the current running through our motors. A table of our measured values is shown below with the corresponding averages from each measurement.

Group Member	One	Two	Three	Average
I_{DC} w/ No Wheels Stalled	0.3920	0.3947	0.4477	0.4115
I_{DC} w/ One Wheel Stalled	1.1680	1.2240	1.2267	1.2062
I_{DC} w/ Both Wheels Stalled	1.4133	1.4933	1.4933	1.4667

3.A.5 Construct the Motor Driver Circuit

After testing half of our H-Bridge circuit on our breadboard, we were confident in our chosen component values, so we built and soldered the full H-Bridge circuit on a perf board. A picture of one of our circuits as well as our circuit diagram is shown below. The soldering itself was relatively difficult, but through a combination of jumper wires and flux we were able to make it work. Throughout the soldering process, we used our digital multimeters to verify our connections as we went, which we knew would make the debugging process much less of a headache.

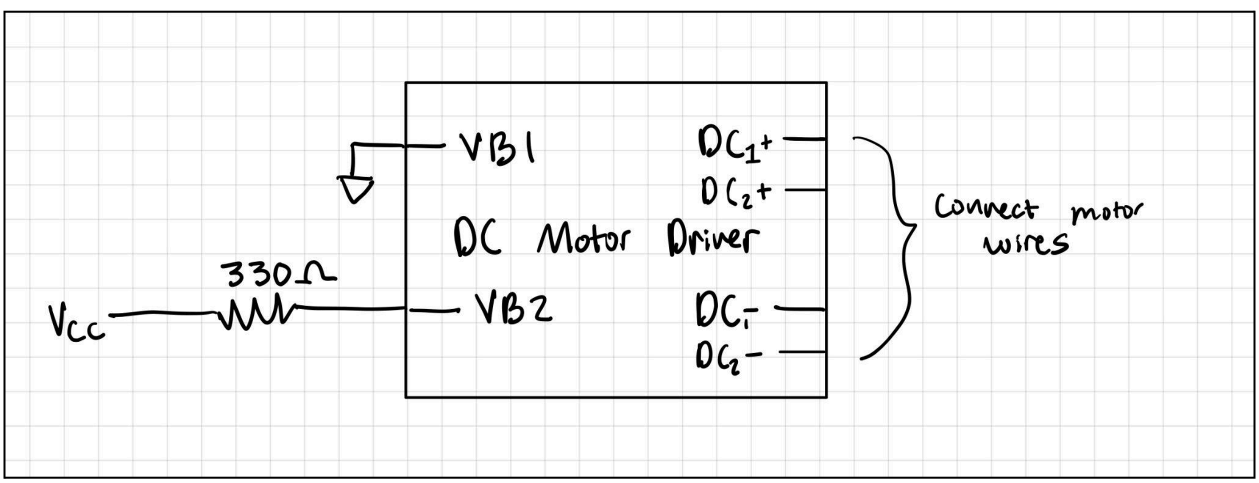
It's not the prettiest circuit, but the advantage here is that it frees up breadboard space and allows us to keep this circuit "below deck".



Circuit Diagram for Motor Driver

3.A.6 Test the Motor Driver Circuit.

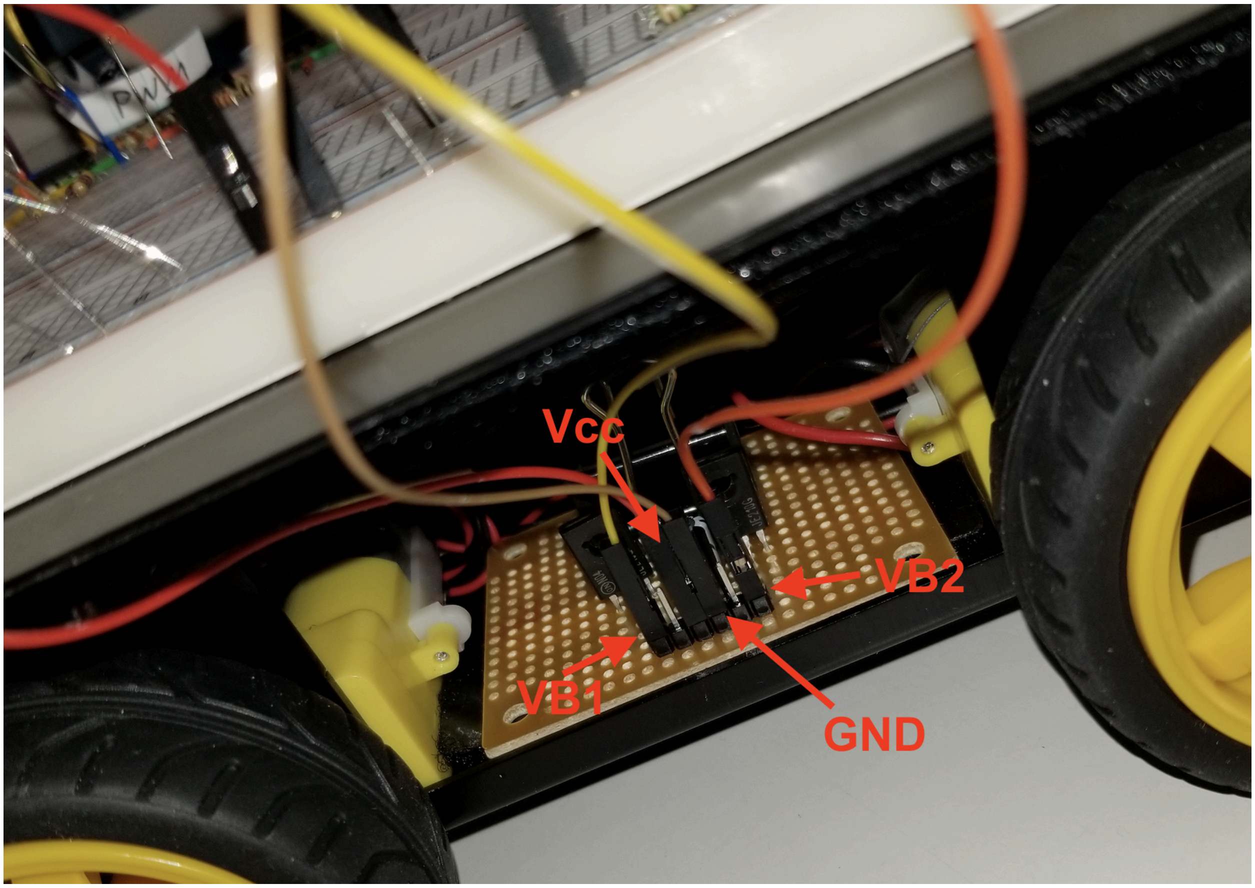
Surprisingly, all of our motor driver circuits worked from the get go, so we didn't have to debug anything. We also verified that the overall circuit was drawing roughly 390 mA, exactly what we expected from the previous test. We also verified both forward and reverse directions for the motor, simply by switching the VB1 and VB2 connections.



Test Setup Circuit Diagram

3.A.7

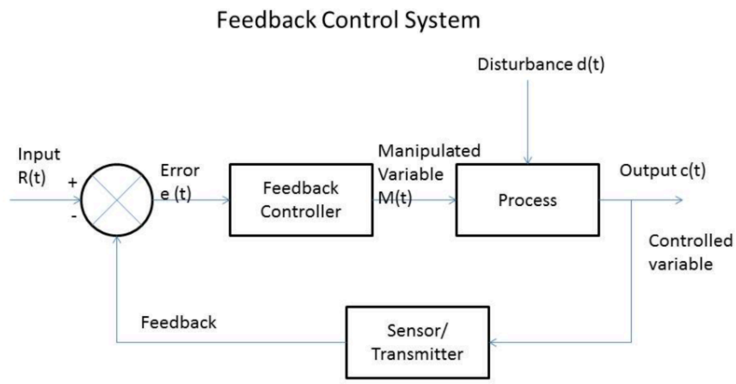
A picture of the mounted circuit is shown below.



3.B Exploration Topics

1) A common feedback control system we thought of would be the heating of a house. This system would read in a temperature, adjust its heat output accordingly, and then repeat the cycle. Another example, perhaps more applicable here, would be cruise control: the system wants to be at a specific speed, so it reads the current speed, either speeds up or slows down, and then repeats the loop till it reads the desired speed.
2) In positive feedback control systems we sum the setpoint and output values vs a negative feedback control system which takes the difference. In general it has been found that negative feedback systems are more stable than positive due to the fact that negative feedback systems are able to attain an accurate measurement of the error and thus tell the system how to fix such an error. This also makes negative control systems less susceptible to random variations in circuit components.
3) A proportional control system responds in a way that is proportional to the difference of the setpoint and output value. An integral control eliminates error than is introduced through a proportional control system at steady-state. The combination of proportional control and integral control is an extremely widespread algorithm in process control due to the balance it provides. Furthermore, a derivative control system also keeps track of

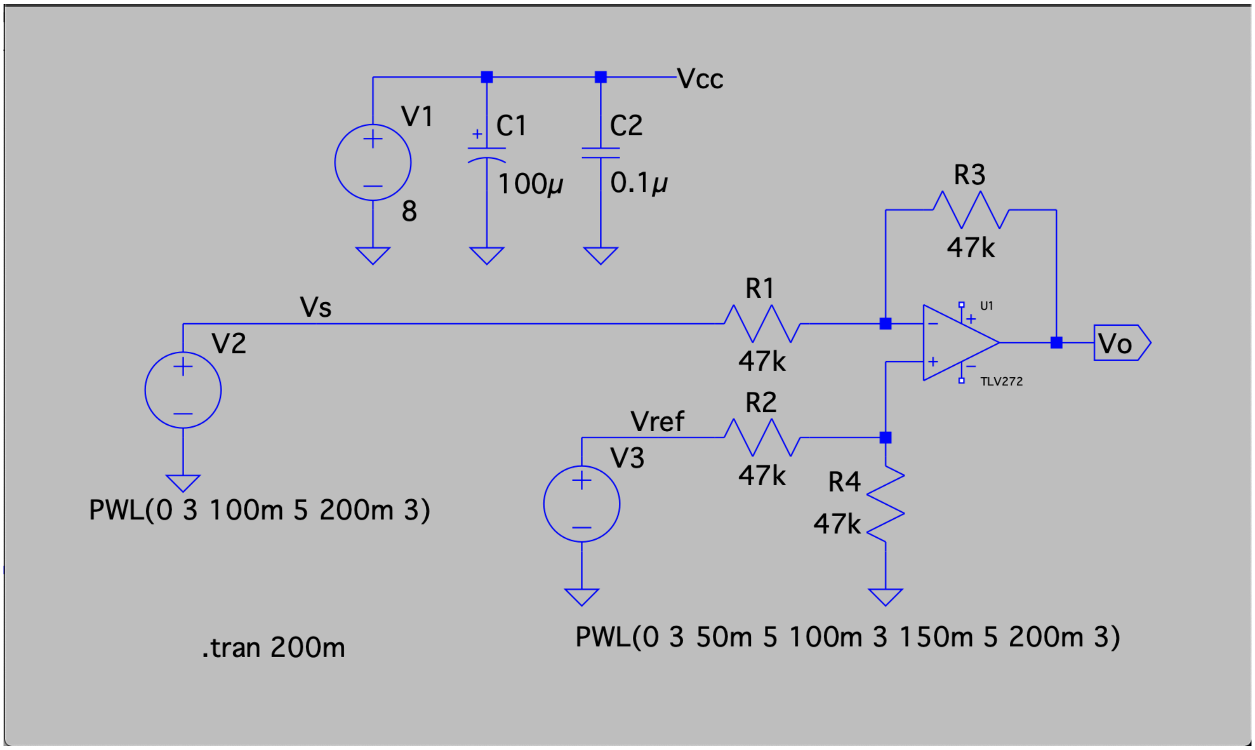
the rate of change of the process variable and can make changes to the process accordingly. The combination of all three of these control processes is a very reliable and widespread method.



4) We liked this diagram in particular because it includes external variables, listed as "disturbance", that would affect our feedback. In our case this would be a stalled wheel. It also describes the difference between the input and the feedback as the error, which makes sense because it's how far off the output is from our desired state.
5) Some of the main advantages of feedback control systems are that it compares the feedback with the desired state to correct itself, non ideal components do not affect the system as much as they normally would, and having a closed loop system keeps out a lot of unnecessary noise and disturbances.
6) Some of the main disadvantages are that the system has more components and becomes more complicated, errors can cascade themselves and cause instability, and the gain decreases across the loop, forcing us to include gain components in the already complicated system.

3.B.2 Investigating the Use of Virtual Ground

In order to construct the circuit and close the feedback loop with an I-Compensator, we first need to look at the reasons to need a virtual ground. First we build and simulate this subtractor circuit shown below in LTSpice. This circuit forms a subtracting amplifier, so it should subtract Vs from Vref. We expect to see Vo as the subtraction of Vref from Vs, and because of our two signals we know that once Vref is lower than Vs we should get a negative output voltage Vo.



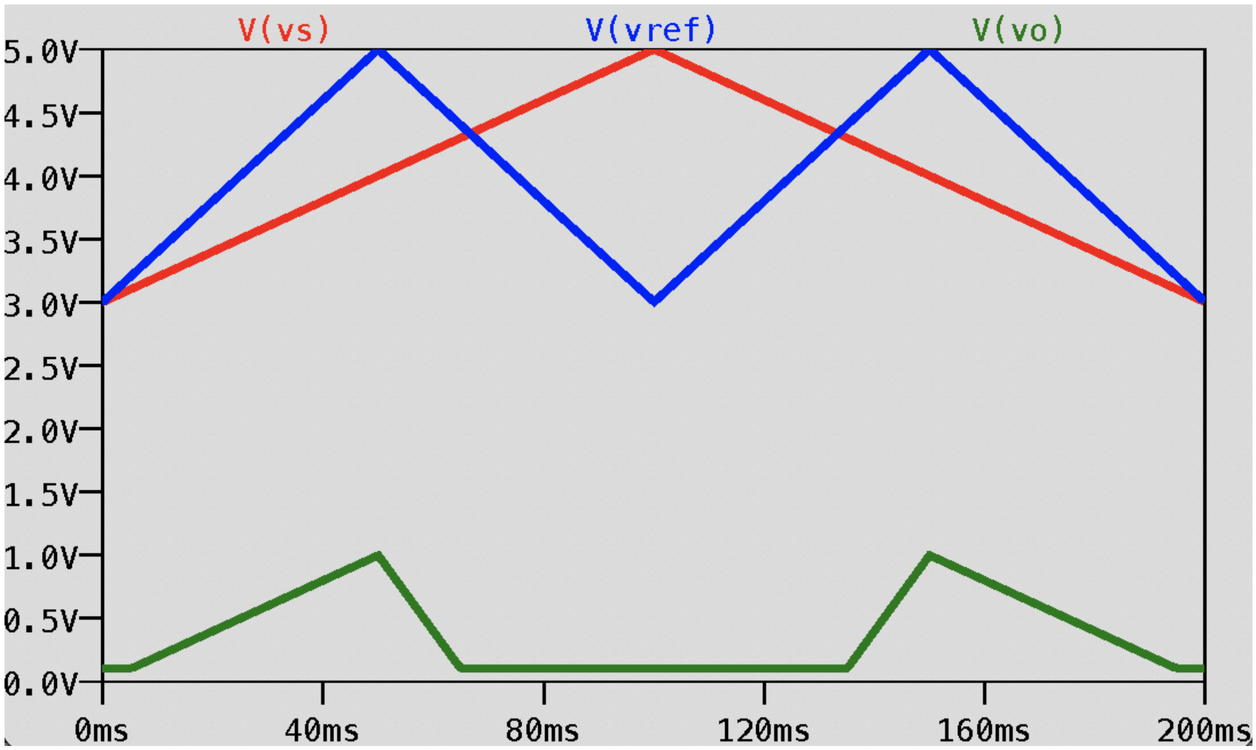
Difference Amplifier using Regular Ground

The gain of this type of op amp circuit should make Vo be equal to Vref minus Vs, scaled by an amount relative to the ratio of the resistances used. The equation is as follows under the case when R1 = R2 and R3 = R4.

$$V_O = \frac{R3}{R1}(V_{ref} - V_s)$$

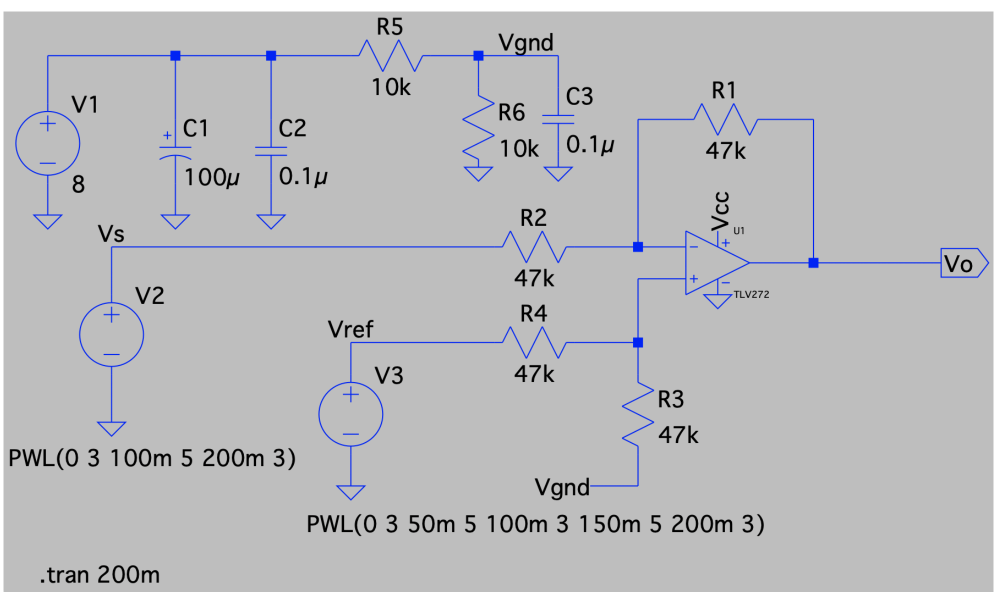
All four resistors have values of 47kΩ, making the coefficient equal to one. This means that the output would be expected to be exactly relate to the op amp inputs with the following equation:

$$V_O = V_{ref} - V_s$$

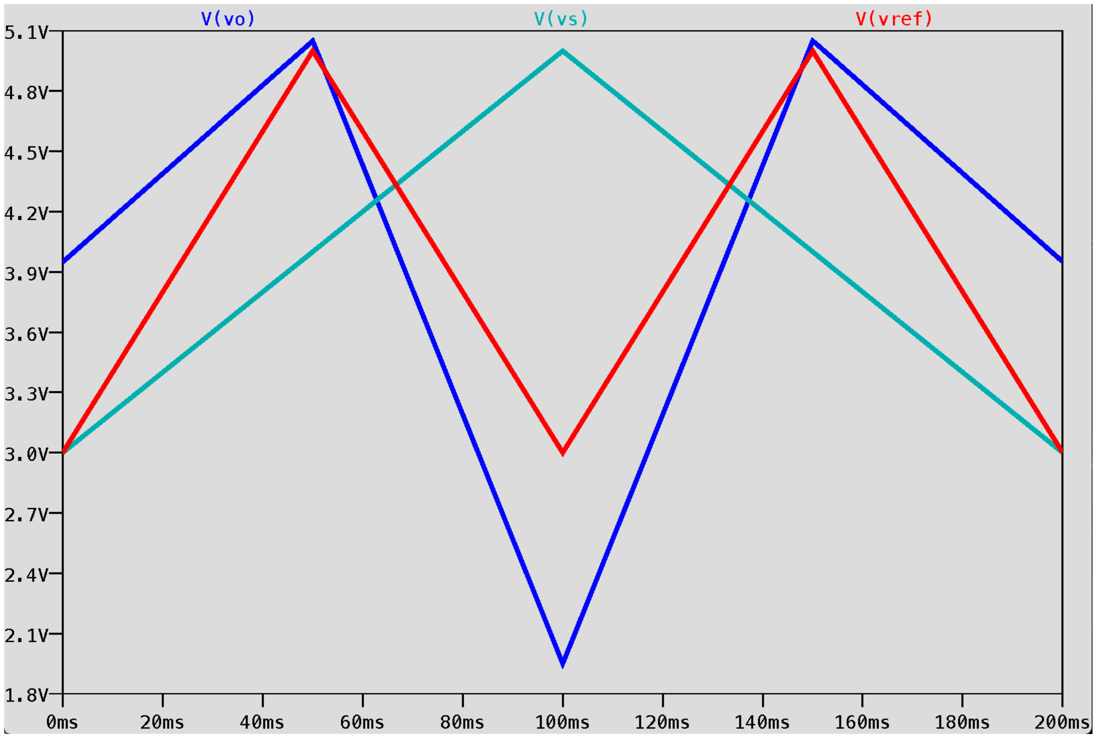


Voltages of Inputs and output of Regular GND Difference Amplifier

Clearly there is a limitation to this circuit, as Vo simply stays at zero where it should be negative. This is because of the power ports on our opamp, and the fact that we have the same ground throughout the circuit. Thus, we will use a voltage divider on our 8V signal, and reconstruct the circuit using a virtual ground for the noninverting input. The op amp will continue to use regular ground. The following circuit shows the changes made from the addition of virtual ground.



Difference Amplifier using Virtual Ground

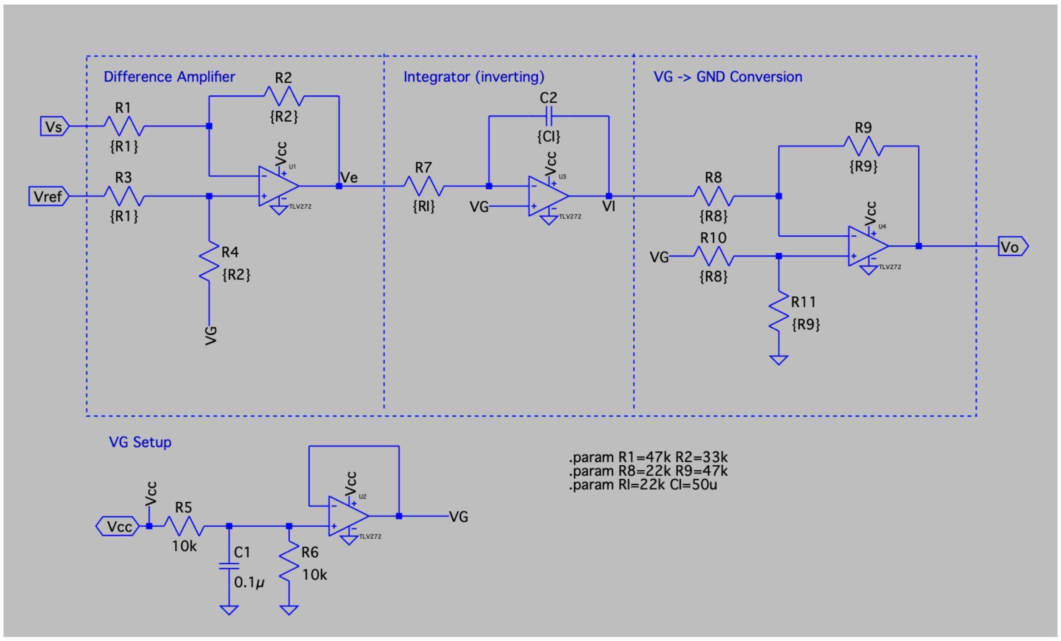


Voltages of Inputs and output of Virtual Ground Difference Amplifier

As we can see, this change to a virtual ground improves the output of the differential amplifier. The addition of the virtual ground offsets the output by 4V, allowing Vo to dip below its initial value, helping it work effectively as our V_{REF} and V_S values change. Because the robot for the class is only driven by a single DC supply, this virtual ground creates the voltage offset that allows for the differential amplifier output to attain the differencing effect that we require.

3.B.3 I-Compensator Circuit in LTspice

We are going to build a closed loop system with an I-Compensator for our robot. Thus we must first simulate these components. Closed loop systems are difficult to debug, so we start by making an open loop circuit, and supplying our input voltages from a waveform generator. We choose RI = $10k\Omega$ and CI = $10\mu F$ to attain a time constant of about 100ms.

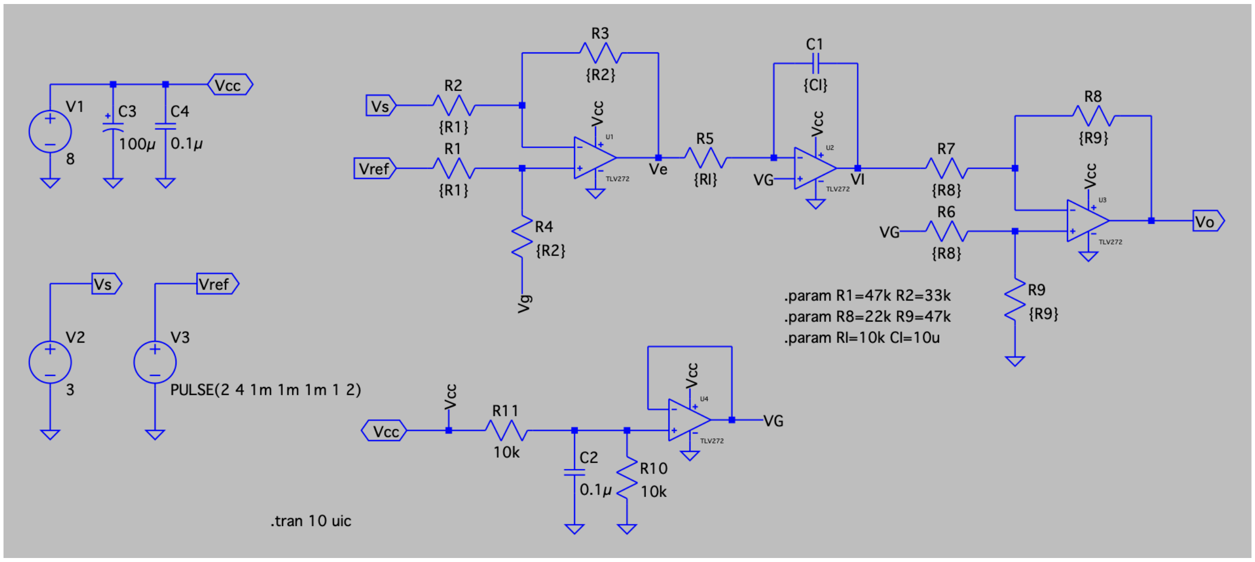


Compensator Circuit in LTSpice

This circuit is then generated into a symbol (.asy file) for later use.

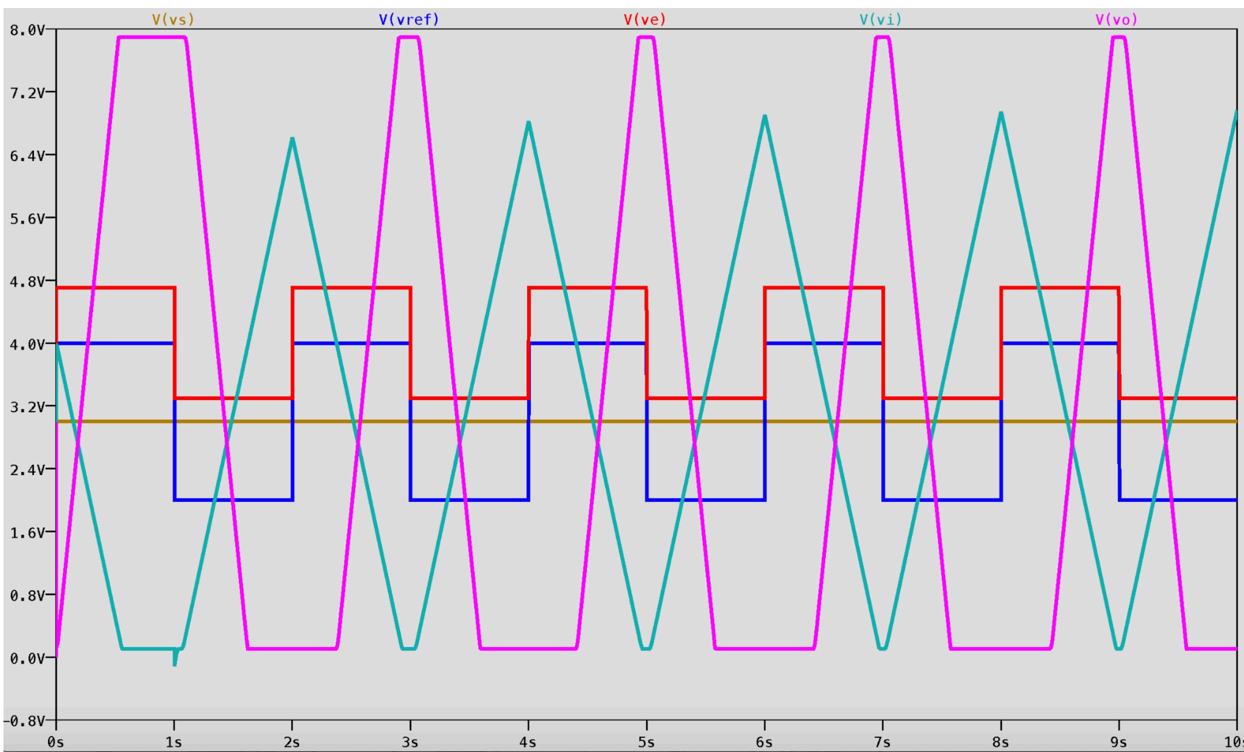
3.B.4 Compensator Circuit Test (open-loop) in LTSpice

Using the general schematic we built in the previous section, we will now run simulations, using a DC Vs signal of 3V, and a squarewave (2,4) signal for Vref. We are also incorporating a virtual ground into the circuit, for reasons explained earlier in the report.



Open Loop Compensator Circuit

Next we will run this circuit and plot, Vs, Vref, Ve, Vi, and Vo.

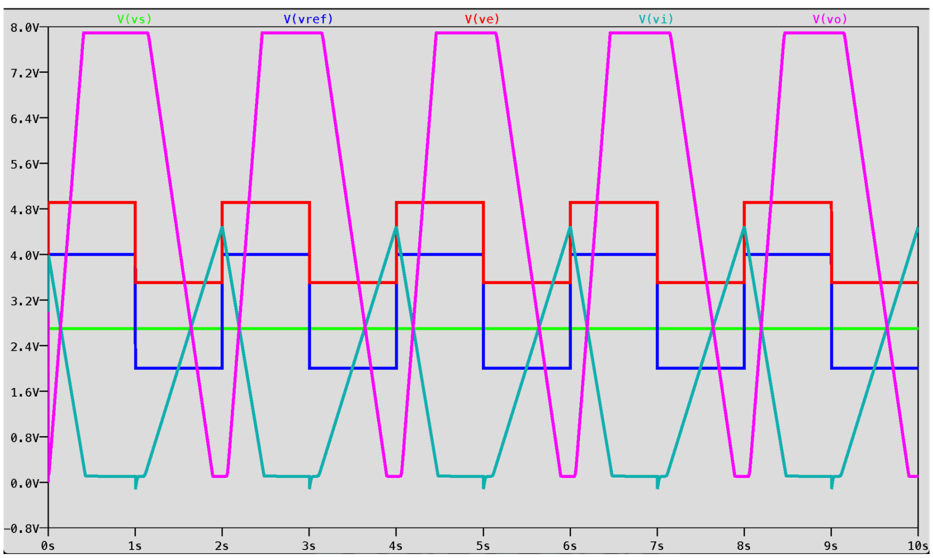


Open Loop Compensator Circuit Response of Different key Voltages

Vs and Vref are passed in to the difference amplifier, which subtracts and inverts the two signals while also adding some gain that is equal to ratio of the resistors. The output signal, Ve, is with respect to virtual ground, which is why we see a square waveform with an average value of 4V.

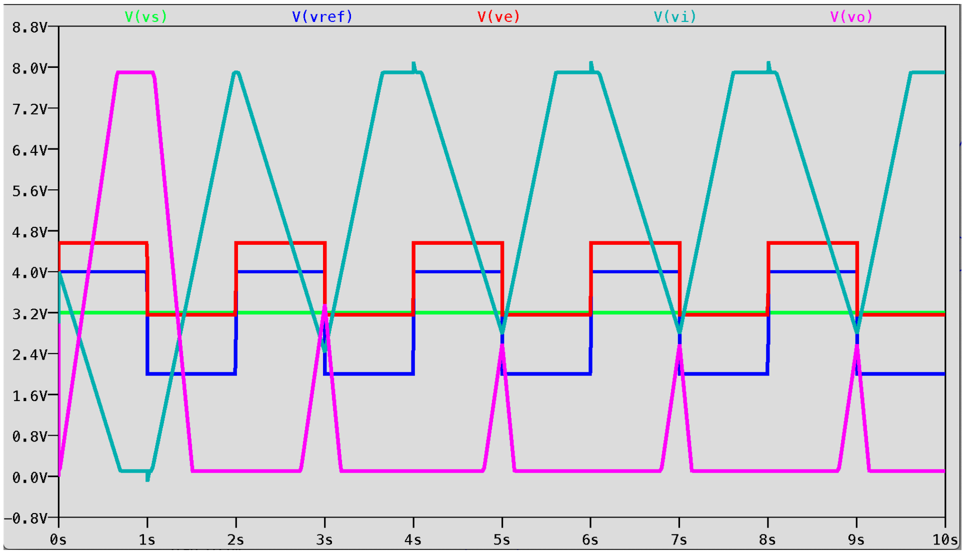
This makes sense because one of our input voltages is a square wave while the other is DC. Ve is then passed into the integrator opamp, which literally inverts and integrates the squarewave with reference to virtual ground, and outputs the signal as Vi. This triangular signal makes sense, because when the squarewave is high Vi is decreasing, and when it is low (with reference to VG) Vi is increasing. Lastly Vi is passed into an opamp circuit that inverts the signal, and references it to circuit ground instead of virtual ground, which is why we see saturation.

If we lower Vs to 2.7V, then Vs - Vref becomes more negative, so Ve becomes more positive due to the inverting relationship. Thus when we integrate Ve into Vi, Ve is more high than low with respect to Virtual Ground, so Vi decreases to a lower value thus saturating more at 0V. This causes the graph to lose the triangle waveform we saw before, which then makes the converted signal Vo saturate more on the high edges.



Open Loop Compensator Circuit Response with Vs = 2.7 V

If we raise Vs to 3.2V, then Vs- Vref becomes more positive, so the inverted Ve becomes less positive. Thus when we integrate and invert this Ve signal into Vi, we see that the signal stays relatively high and saturates at the peaks. It also has sharper edges that don't allow it to reach the low values that it would ideally. When we invert this signal through a difference amplifier into Vo, it's a signal that saturates at zero instead of 8 and has a small peak every period.



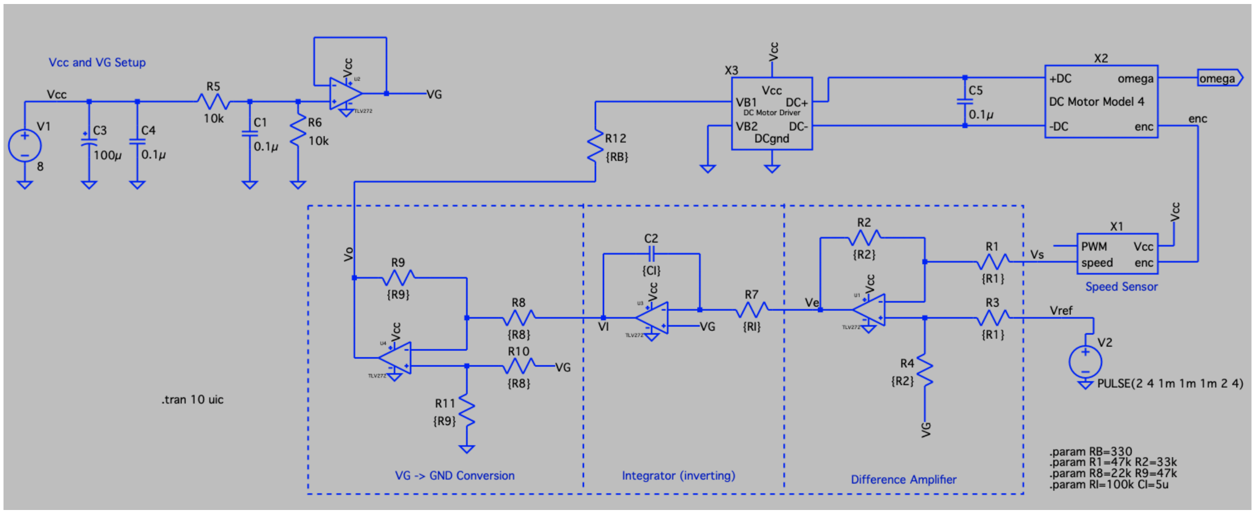
Open Loop Compensator Circuit Response with Vs = 3.2 V

Overall, we can conclude that increasing Vs decreases Vo, and decreasing Vs increases Vo. This information will be useful for when we close the feedback loop. This makes sense because we pass Vs through three different inverters before reaching Vo.

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

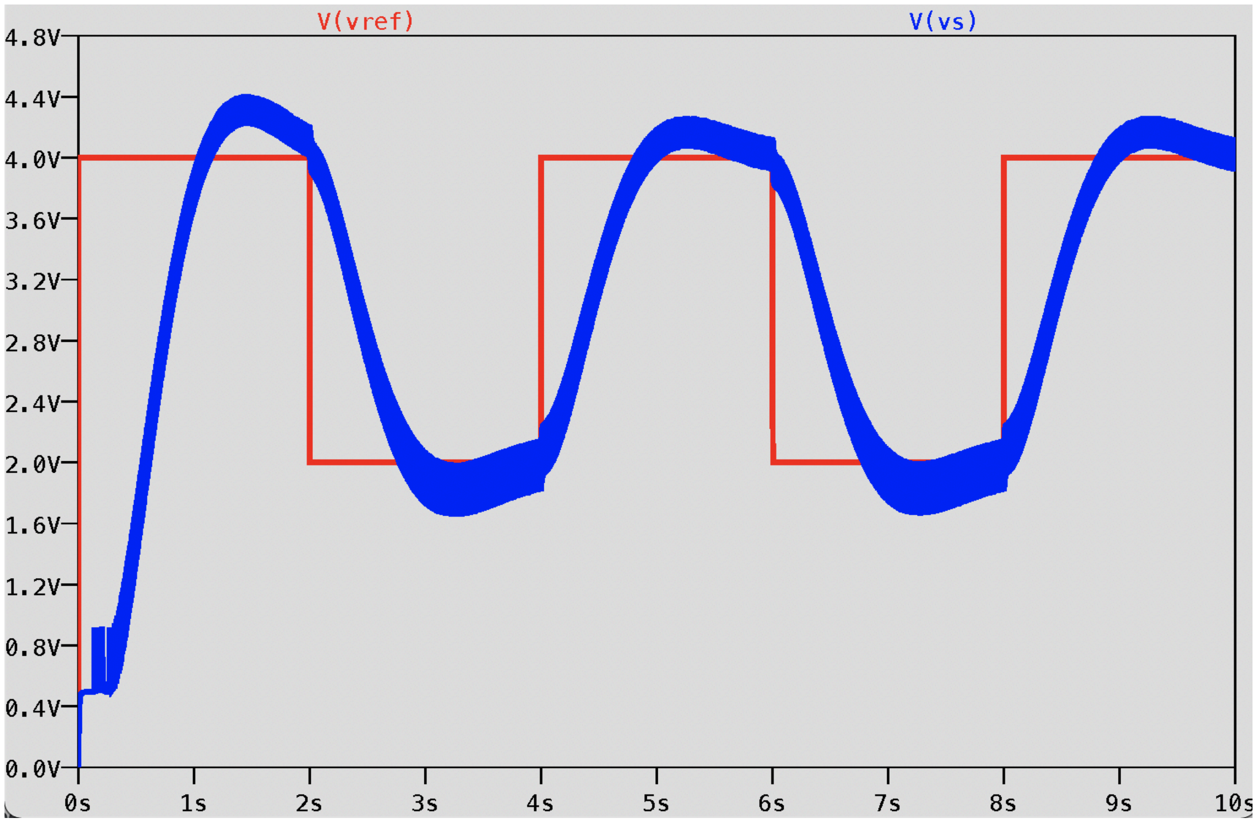
Now that we have simulated and interpreted the open loop circuit, we are ready to test the closed loop circuit. A picture of the LTSpice schematic is shown below.

After some intense calculations we determined our $\zeta = 1$ is achieved with an RC time constant of about 0.5. We manipulated these values a bit until we got our best results. This was an effort of trial and error of adjusting RI and RC, comparing the Vs and Vref output. The goal was to make Vs follow as closely as possible, because this happens with critical damping. Our finalized values were C = 5μF and R = 100 $k\Omega$.



Closed Loop Compensator in LTspice, CI=5μF, RI=100kΩ

Running a transient simulation on this closed loop compensator will allow us to compare the output of the speed circuit, Vs, with the provided input of Vref. This important step verifies that we will be able to have the motors closely follow a toggled voltage provided from an Arduino pin.

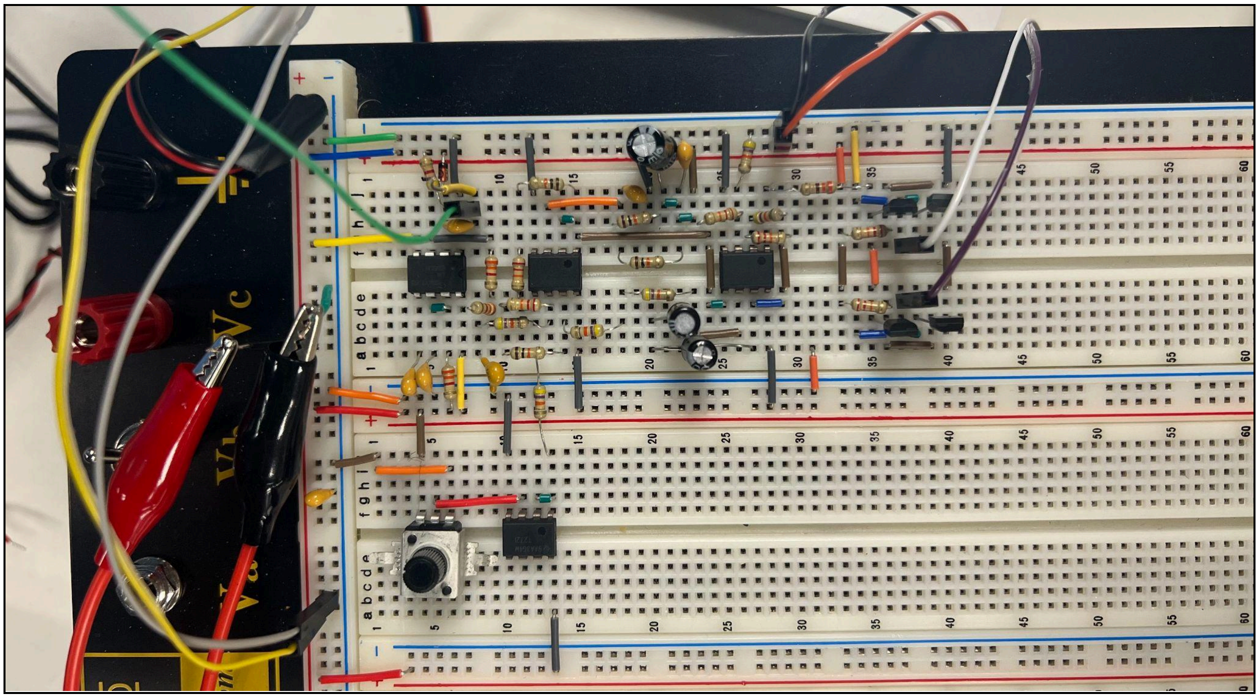


Comparison between Vref and Vs in Closed Loop Compensator

As seen by the graph, the compensator does a good job having Vs follow Vref. There is an inevitable rise time, but the system is a close to critically damped as possible with the available capacitor and resistor values.

3.B.6 Build Compensator Circuit

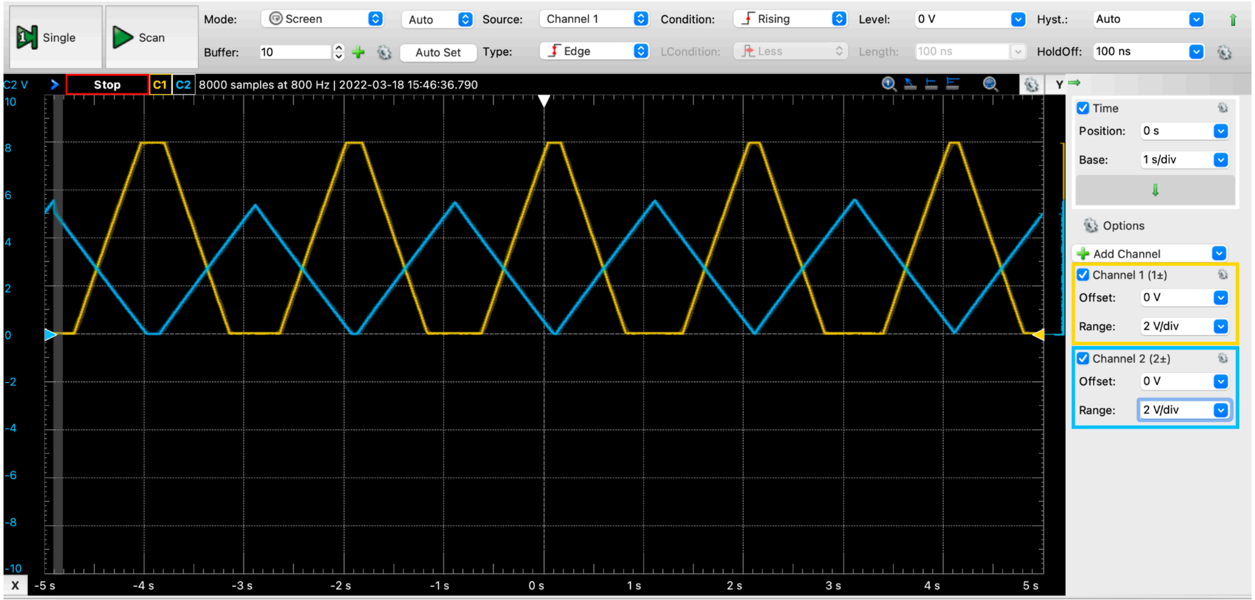
After running open and closed loop simulations in spice, we are ready to build and test these circuits on our breadboard(s). This circuit was somewhat tricky to build, but given that each TLV272 has two integrated op amps, we are able to build the circuit efficiently. A picture of one of our circuits is shown below.



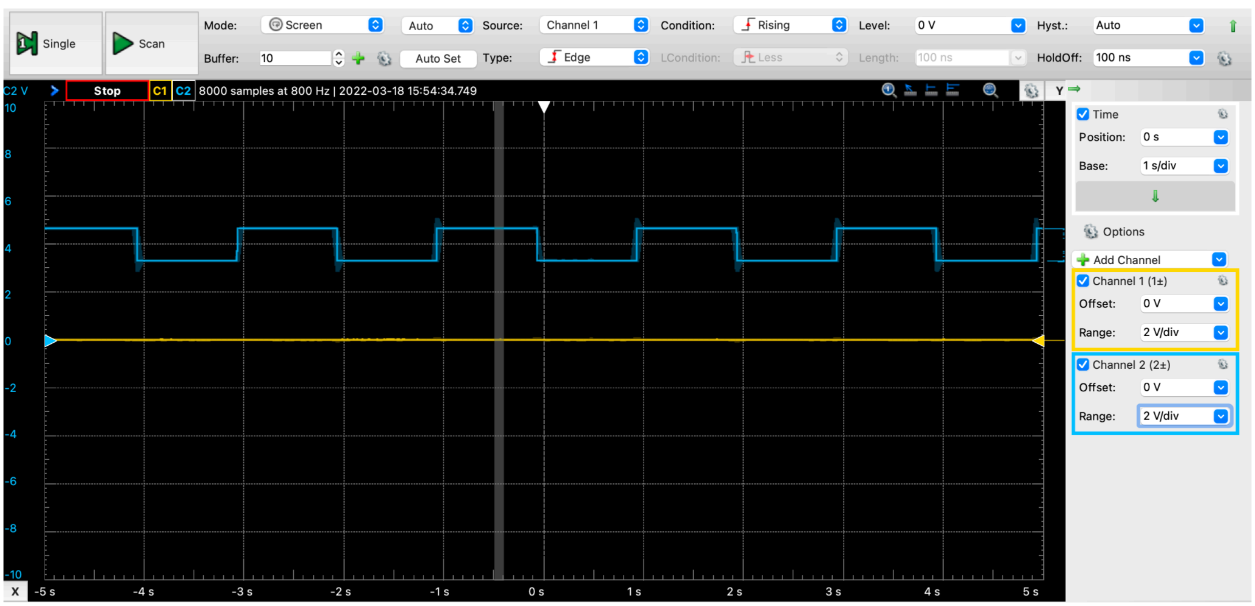
Fully Functioning Circuit with Compensator Implementation

3.B.7 Compensator Implementation Test (open-loop)

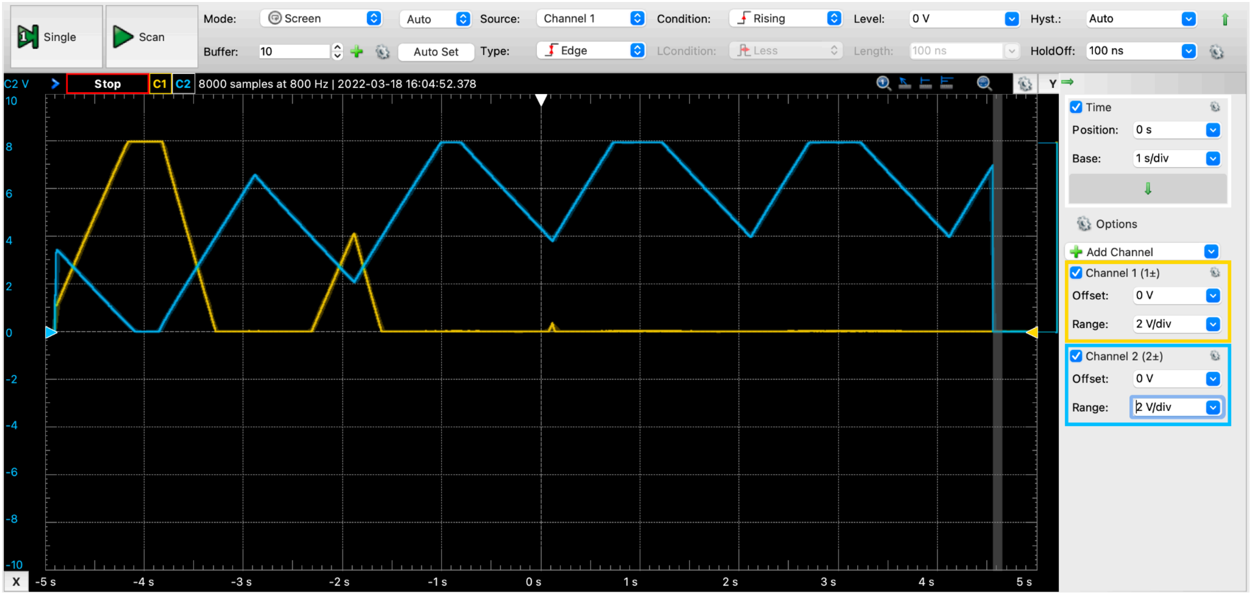
Now that we have built our circuit, we will test it in the open-loop configuration. We do this by using our AD2 wavegen to supply Vs and Vref. We ran the circuit with 3V for Vs and a 500mHz squarewave (2s period) between 2V and 4V for Vref. We then probed Ve, Vi, and Vo to affirm our signals with what we simulated. For these outputs we used a similar time constant of about 100ms, using a C = $5\mu F$ and R = $22k\Omega$ After a lot of debugging and frustration, we were able to achieve correct outputs for all three circuits that match very closely what we simulated in LTSpice. This confirms that our circuit is working correctly and we can move forward in the lab. The waveforms for Vi, Ve, and Vo are shown below at different Vs levels.



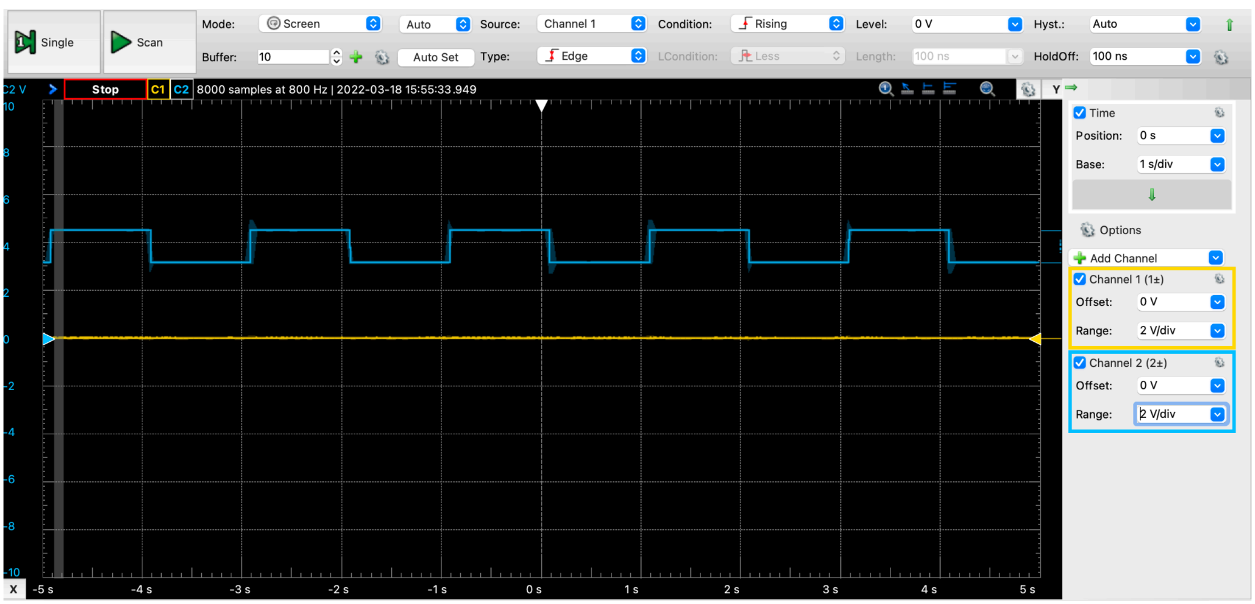
VI (Channel 2) and Vo (Channel 1) Open-Loop Test with Vs = 3V



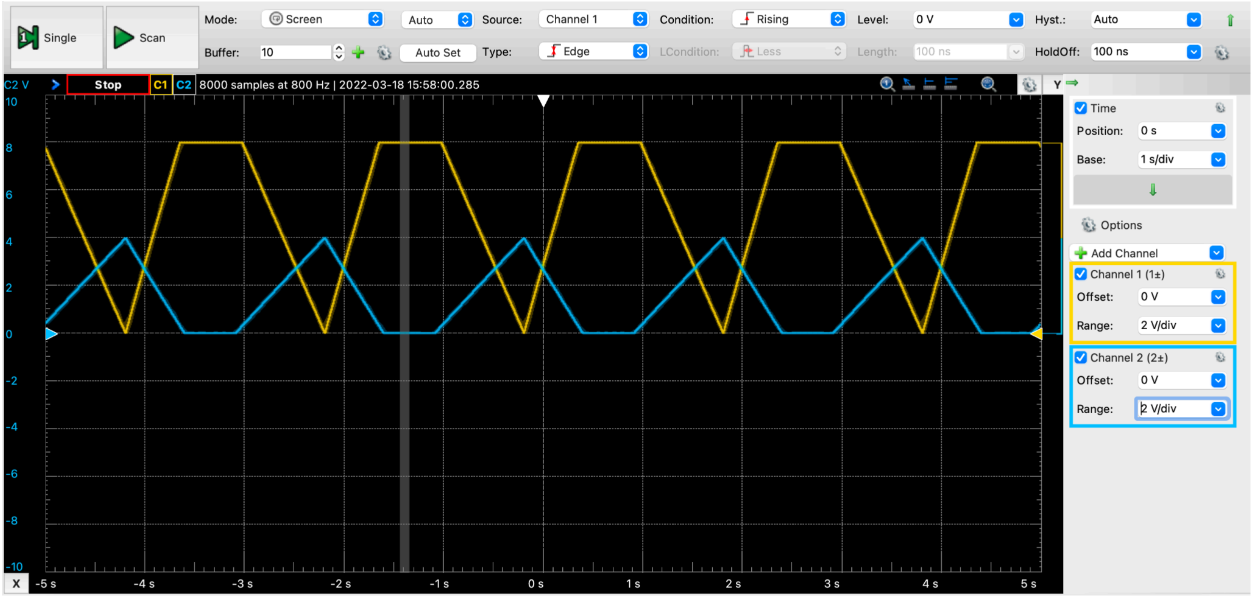
Ve (Channel 2) Open-Loop Test with Vs = 3V



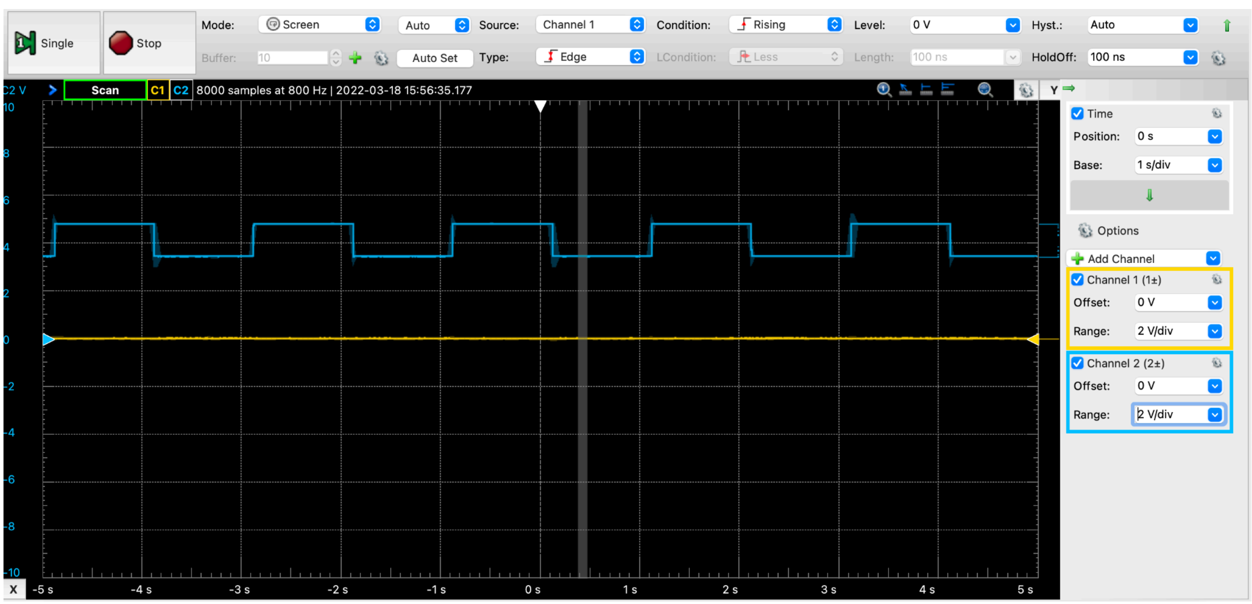
VI (Channel 2) and Vo (Channel 1) Open-Loop Test with Vs = 3.2V



Ve (Channel 2) Open-Loop Test with Vs = 3.2V



VI (Channel 2) and Vo (Channel 1) Open-Loop Test with Vs = 2.8V



Ve (Channel 2) Open-Loop Test with Vs = 2.8V

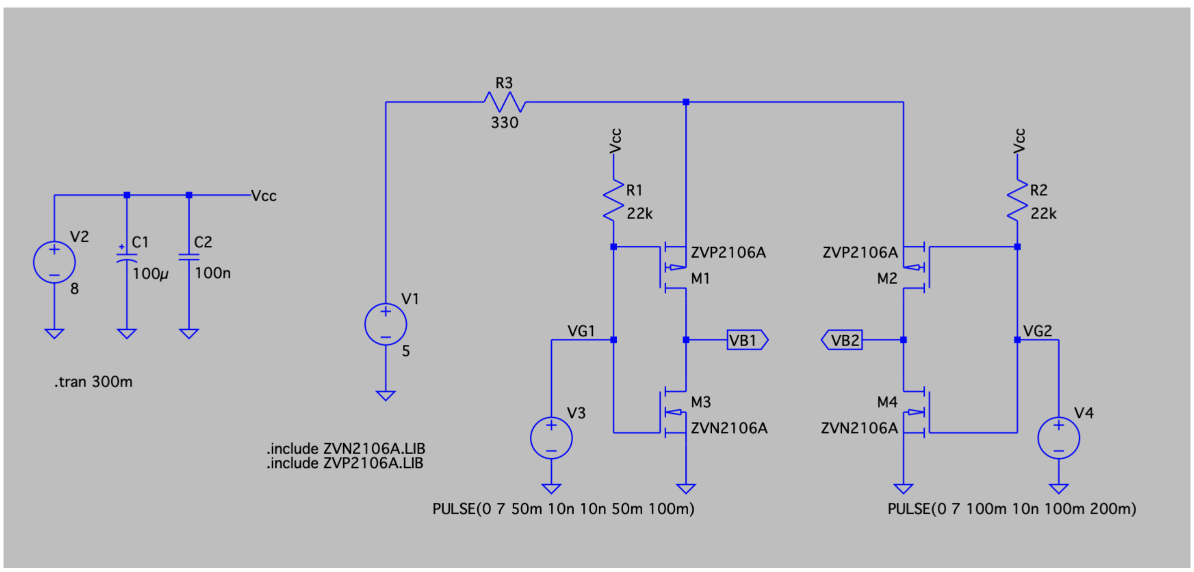
3.B.8 Test Closed-loop with Compensator Implementation

Now that we have confirmed our I-Compensator implementation in open-loop, it is time to close the loop. We connect the output of our speed sensor circuit to Vs, and the existing potentiometer/opamp circuit output to Vref. We also connect the motor driver circuit from Part A by connecting a 330Ω resistor (Rb) in series with our compensator output Vo, sending that signal into VB 1 or 2, and grounding the other VB pin. The motor driver circuit also has Vcc and DC GND pins that are connected to our power rails.

The only test we really do here is adjusting the potentiometer, and confirming that our wheels speed up or slow down. We test both directions of operation by switching the wiring of VB1 and VB2. We also observe a small delay between adjusting Vref and the actual wheels spinning, which we know is related to our damping factor ζ. We will do tests regarding this damping and delay later on in the report.

3.B.9 Direction Control in LTSpice

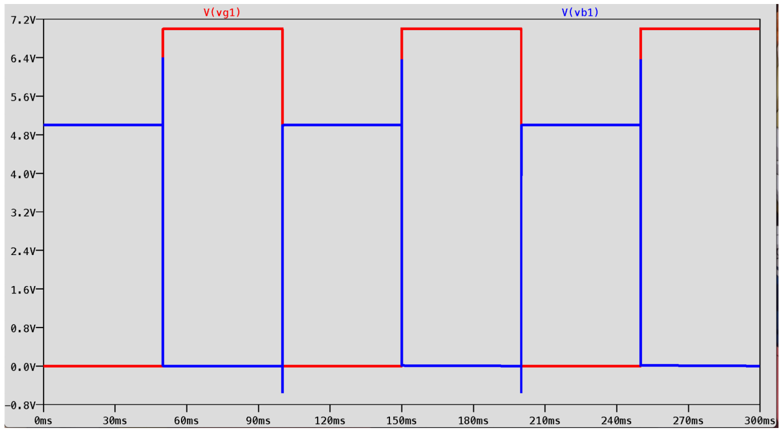
Now that we have our feedback loop working correctly, we want to add a direction control subcircuit, which will relate to our VB1 and VB2 pins, since the polarity of their wiring drives forward or reverse motion. Keeping with the theme, we will build and simulate this subcircuit in spice first. A circuit schematic is shown below.



LTSpice Direction Control Schematic

Our voltage sources V3 and V4 are set up so that when one is high, the other is low. Each source is connected to one of the mosfet gate nodes. If the gate-source voltage is high enough, then current will flow through the source to drain, or vice versa depending on whether we have a ZVP or ZVN mosfet. Therefore, when the mosfets are active, there is no voltage drop across them, meaning that when they are active their respective VB is grounded.

A plot of our waveforms is shown below, which exhibits the behavior previously described. We can see that when VG1 is low, VB1 is high, and vice versa. The second plot displays the VG2 VB2 relationship, with a slightly different VG2 waveform, but we can see that this plot is still what we expect.



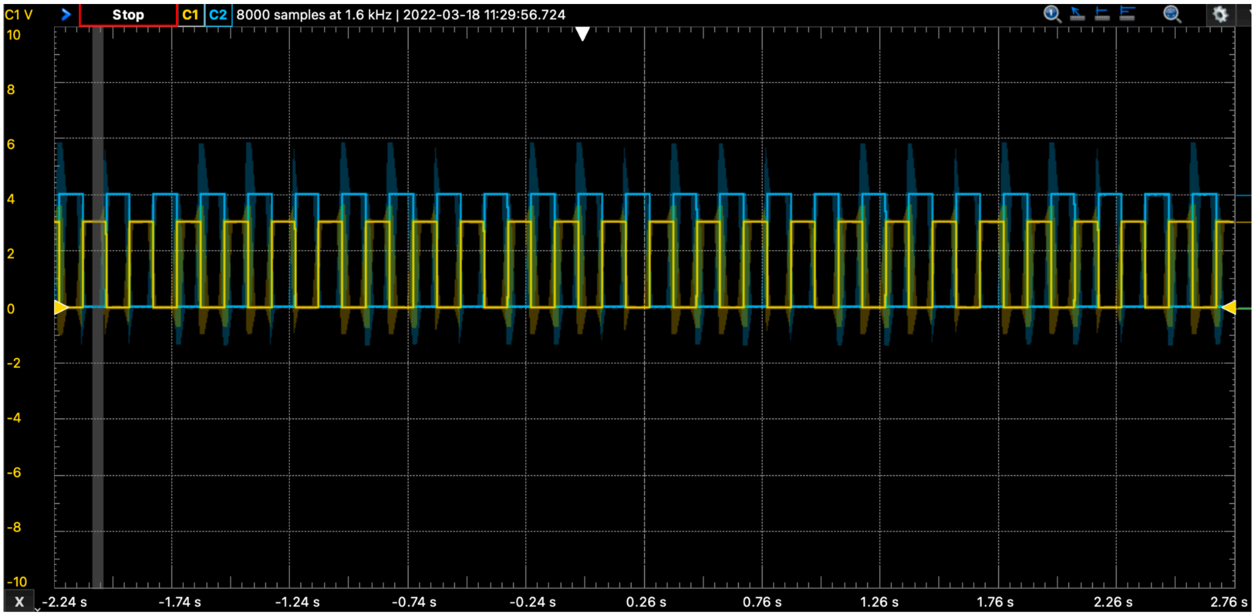
Comparison between VB1 and VG1

We also need to develop a method for testing this circuit separate from our other circuits. An important practice of electrical engineering and circuit design is building and testing each subcircuit one at a time, and then adding it to the entire circuit.

We decided that we would test half of the circuit at a time. We would supply 5V DC from the lab DC power supply to VB, and 8V DC to Vcc, whilst also grounding the NMOS sources. Then we would send a squarewave into VG, alternating from 0 to some high value, say, 3V. We would then probe VB, and hope to observe that when VG is high, VB is zero, and when VG is zero, VB is high. Because of our circuit schematic, we can test each half of the circuit at a time.

3.B.10 Adding Direction Control to Closed Circuit Implementation

Now that we simulated our direction control circuit, we are ready to test it. We built it on our breadboard, but refrained from passing in our I-Compensator output Vo. There are two scenarios to test here, the relationship between VB1 and VG1, and the relationship between VB2 and VG2. In both these cases, V1 will be set at a DC voltage of 5V through a voltage generator, while the square waveforms of VB1 and VB2 will be provided by the Analog Discovery waveform generator.

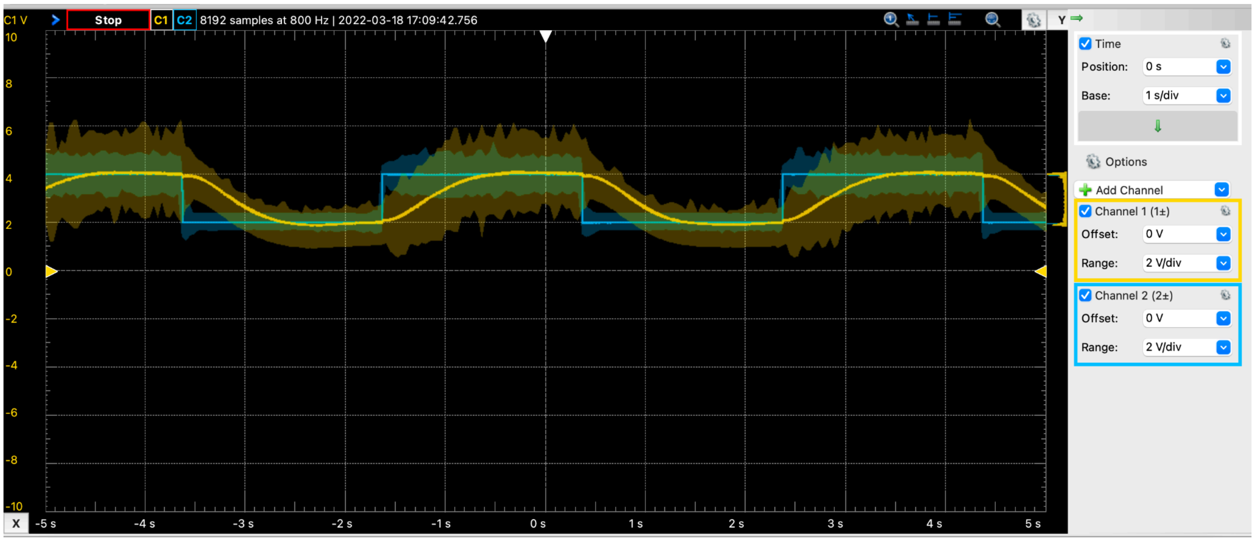


Voltages from Direction Control Circuit, VG 5 Hz Square Waveform (Blue), VB (Yellow)

As is evident from the graph, when VG is high, VB is low, and vice versa. We performed this test on both halves of the circuit, however we felt only one graph was necessary.

3.B.11 Test Overall Speed Control Loop Implementation

Finally, it's time to put everything together. We integrate our direction control subcircuit into the overall speed control loop. To make sure that everything is working correctly we first use the potentiometer circuit as our Vref. In this case our Vi essentially follows our potentiometer. When our potentiometer is turned all the way down our Vi is at roughly 0V and our wheels are spinning at max speed. The Ve in the case of max speed is roughly 6V, and 4V at minimum speed which makes sense because Ve is in reference to our virtual ground. In this static test we ensure that our wheel speed follows from the resistance of our potentiometer.



Dynamic Test using Rectangular Waveform of Vs (Channel 1) and Vref (Channel 2)

From this graph we can approximate that ζ is slightly less than 1. We don't see an overshoot in the Vs waveform so we know this is not overdamped, however it does take a little less than a second to reach the steady state value. After testing of many different RC time constants this was the best output that we were able to achieve. In this graph we are using a $C = 5\mu F$ and $R = 94k\Omega$ to get a time constant of approximately 0.5.

Conclusion

We achieved all of our objectives throughout this lab, which entailed simulating, testing, and soldering the motor driver circuit, as well as simulating and building the speed control loop circuits. In doing so, we have created a feedback control system that takes one input as a reference voltage and drives the speed of the motors to reach that parameter, by reading the encoded frequency through the speed sensor circuit built in a previous lab. In this lab, we vary the input voltage with a potentiometer, but going forward we will likely use a microcontroller to supply this voltage, allowing us to create code that drives our motors.

Naturally, we encountered limitations in this lab. The main limitation is space constraints on our breadboard. We had to construct these circuits as efficiently as possible, which made for an extremely complicated breadboard. This complexity made debugging difficult, as well as finding specific nodes of the circuit difficult. Another limitation was the inability to achieve ideal resistance and capacitance values. We had to construct combinations of resistors and capacitors to achieve the values we solved for in simulation, and in the case of the electrolytic capacitor, we had to put two in series, which was difficult to implement given their size and the limited space we have.

The two most important things we learned in this lab would have to be the general construction and implementation of a feedback control system, as well as how to use combinations of transistors to supply a polarized current. In regards to the feedback loop, an important principle we learned was to test the feedback controllers in an open loop before closing the loop, which made our debugging process much easier. Additionally, the overall value of transistors was something that we took away from this lab, specifically their ability to switch signals on and off, without needing any code or digital information to do so.

Something we think could be improved in this lab would be more communication regarding RI and CI values. Specifically, the fact that we use two different RC values in two different sections. The ordering of the lab sections goes open loop simulation, closed loop simulation, open loop hardware, closed loop hardware. Because in the closed loop simulation we solved for RC values in regards to zeta, we implemented these RC values in hardware in the next section. This made debugging our open loop circuit extremely frustrating, because we were unaware that they should've been chosen in regards to a time constant value. None of the TAs seemed to know this either, at least when we were debugging our outputs Ve, Vi, and Vo, none of them mentioned the fact that our RC values should've been chosen for a time constant of 100ms.

Overall, we are satisfied with our work on this lab, and are excited for future lab reports utilizing the circuits we have built.