Lab 2

ECEN 2270
University of Colorado Boulder

February 25, 2022

Luke Hanley
Nicholas Haratsaris
Ginn Sato

Introduction

In this lab report we begin to power our DC wheel motor, as well as analyze our encoder. As has been the procedure in previous labs, we first simulate and analyze our circuits in LTSpice, then once we attain the correct resistor and capacitor values, we build the circuit on our breadboard. Overall, the goal of this lab is to power the motors, and interpret the encoder signal that is sent back to us by the motor.

The first part of the lab is focused on powering our motor with a safe, stable DC supply voltage and current. Our objective is to supply the motor with a safe, stable ~7V DC supply given our 8V power supply. The circuit is designed in such a way that it allows us to solve for our mechanical motor parameters using simple oscilloscope measurements. These measurements are used to later simulate and verify the workings of our motor.

The second part of the lab looks at our motor encoder chip, which outputs a frequency relative to the speed of the wheel shaft. Our objective is to output a voltage proportional to our wheel speed, given the input encoder frequency. We use LTSpice to design, simulate, and debug the circuit. Then, we add this circuit to our breadboard to operate in conjunction with the motor supply circuit from the first part.

Equipment List

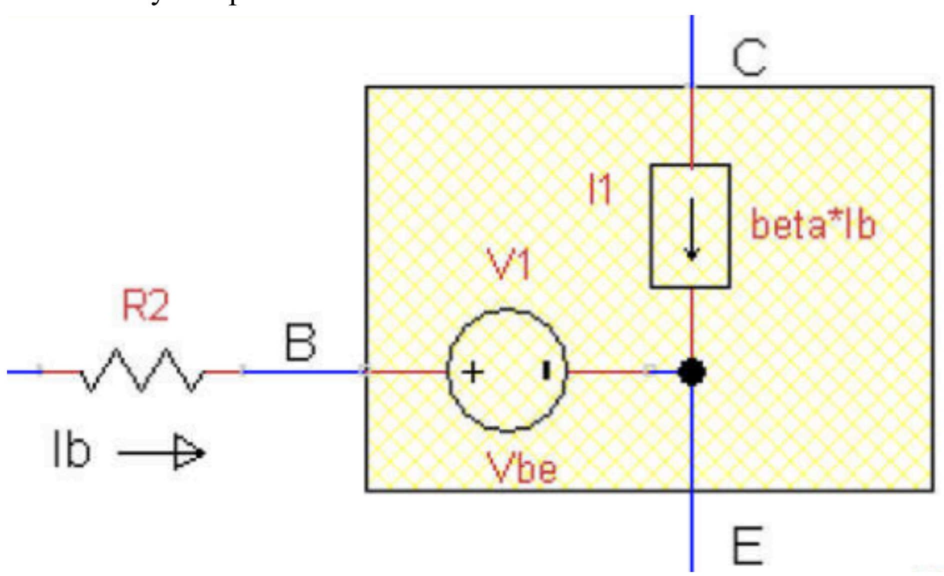
- AD2 Scope w/ BNC Probes and Flywire Assembly
- LTSpice Software
- MATLAB Software
- DC Motor Model 4 LTSpice File
- Fully Assembled ROB 0025 Robot
- Jumper Wires and Breadboard
- 8V DC Power Supply
- DC Motor with Encoder
- 10k to 100k Potentiometer
- MJE200G BJT Transistor
- TLV272 OpAmp
- LM555 Timer
- 1N4148 Diodes
- Assorted Resistor Booklet
- Sparkfun Capacitor Kit

2.A Lab Exploration Topic

1) BJTs are made from silicon, a type of semiconductor
2) There are two types of BJTs, NPN transistors and PNP transistors. The difference between the two is the polarity of the current that controls the transistor, and whether the current goes into the emitter or the collector.
3) A BJT has three nodes, the emitter, base, and collector. Depending on whether we have an NPN or a PNP transistor, a voltage difference between the base and the emitter, or collector, controls if current flows. An NPN has the collector at a positive voltage and the emitter grounded, whereas the PNP has an inverse setup. Thus for NPN, if the base node is more positive than the emitter, current flows from collector to emitter. For a PNP, if the base node is more positive than the collector, current flows from emitter to collector. Additionally, in an NPN, the base node current flows into the emitter, and for a PNP, the

base node current flows out of the base node. This is important to understand for circuit design and BJT integration.

4) The key electrical characteristics of a BJT are the base-emitter voltage or base-collector voltage, depending on NPN or PNP, in which the difference determines when current flows. The other key parameter is the DC current gain from collector to emitter, or vice versa, which is a gain factor unique to each BJT.
5) A BJT is either off, on, or saturated. These different modes are determined by the base-emitter/collector voltage. If this voltage difference is below a certain threshold, the transistor is off and no current flows. If it is above that threshold, but not too high, it is on, and current flows. The last mode is saturation, which is when the difference voltage is so high that the output current does not keep increasing. It is not good practice to operate in this mode, as the additional voltage creates energy that cannot go anywhere, and the BJT will likely produce a lot of heat.
6) BJTs can be modeled with an independent voltage source and a current controlled current source. The voltage source is on the base node, and a resistor in series with this source determines the base current. The base current is the current that controls the collector/emitter output current, in which this current times a constant (DC gain factor) produces the final output current through the emitter/collector. Thus a simple NPN BJT can be modeled by this picture:



7) The common collector circuit connects the collector node to the input voltage, and the output node is the emitter, thus this circuit amplifies the current through the base and outputs that current at the emitter.
8) The common emitter circuit connects the emitter at ground, and the output node is the collector, thus the circuit amplifies the current going from emitter to collector, by a factor

of the current coming out of the base, which is determined by the difference voltage between the base and the collector.

2.A.2 Variable Motor Supply Circuit

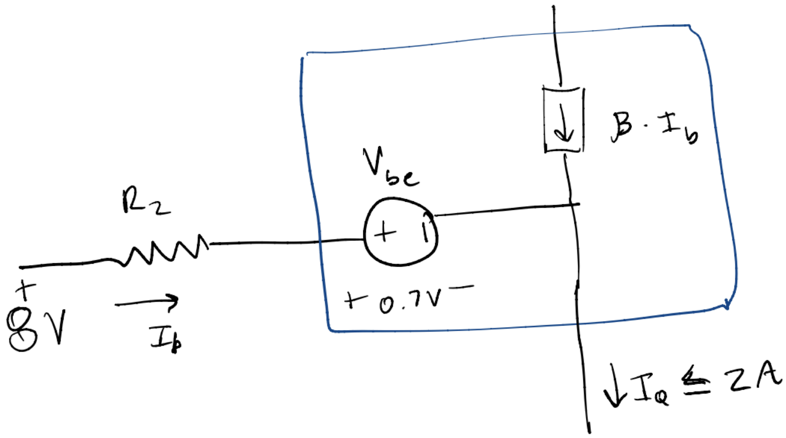
This circuit's overall purpose is to provide a safe, stable signal to our motor(s). First we have an 8V battery which provides power to the circuit. The purpose of the opamp is to output a stable voltage that we can manually adjust via our potentiometer, which varies from 10 $k\Omega$ to 100 $k\Omega$. Because of the potentiometer and the voltage drop it induces, we avoid saturation in our opamp.

Following our opamp, we need to make sure the current supplied to our motor is a safe amount, specifically we calculated 2A for the maximum current. The purpose of the transistor itself is to stabilize the current being supplied to the motor, and also allow us to regulate it with our R_2 resistance. We calculated this resistance to be about 330 Ω, so we used the 330 Ω resistor in our kit. We expect the maximum current through our motor, I_M, to be approximately 2A. Using this information we can calculate V_I such that $V_I = R_3 * I_M = 0.75$ V. We then use the fact that $V_M = V_P - V_I = 8 - 0.75 = 7.25\,V$.

Connected directly from our 8V power supply to ground is a 0.1 uF capacitor that acts as a decoupling capacitor to prevent large drops in our supply voltage. A capacitor stores voltage potential that fills in rapid drops from our power supply in order to maintain a stable current flowing through our motor.

R_3 is used to measure the motor current, thus we do not want to have too large a resistance or we will affect the rest of our circuit. We also need to be careful of dissipating too much power in these ¼ W rating resistors, so we have to put enough of these 1.5 Ω resistors in parallel to do so. Calculations for both R_2 and R_3 are shown below.

2



$$I_b + 100 I_b \le 2A$$

$$101 I_b \le 2A$$

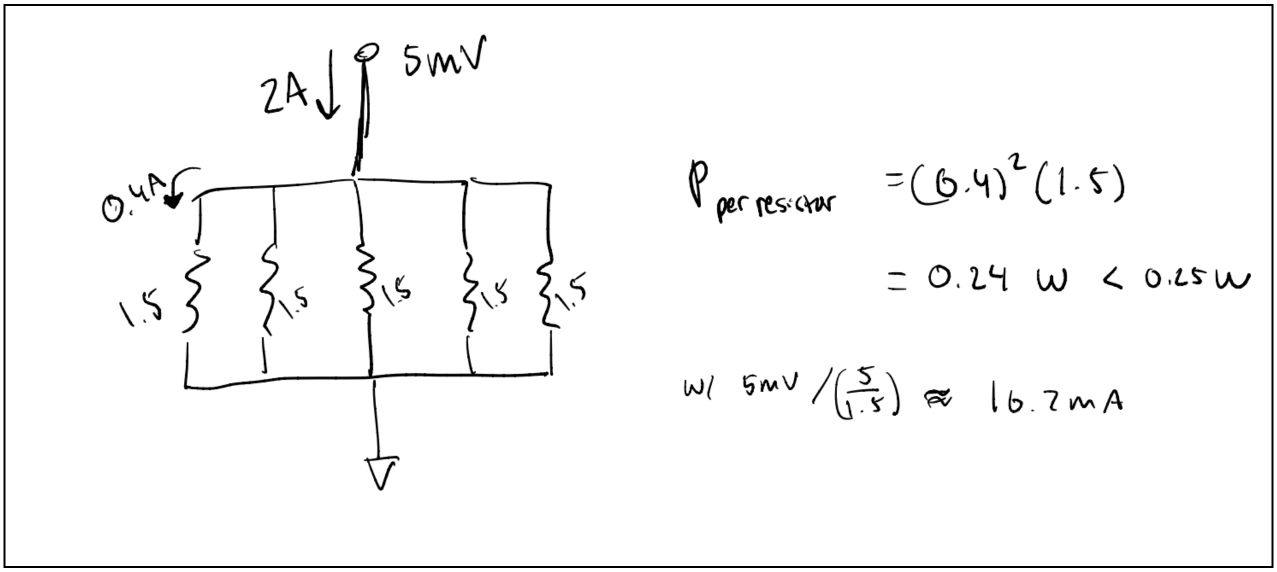
$$I_b \le 0.0198 \text{ A}$$

$$8 - 0.7 = I_b \cdot R_2$$

$$\frac{7.3}{0.0198} = R_2 = 368$$

$R_2 \approx 330\Omega$ & what we have in kit

Hand Calculations for R_2

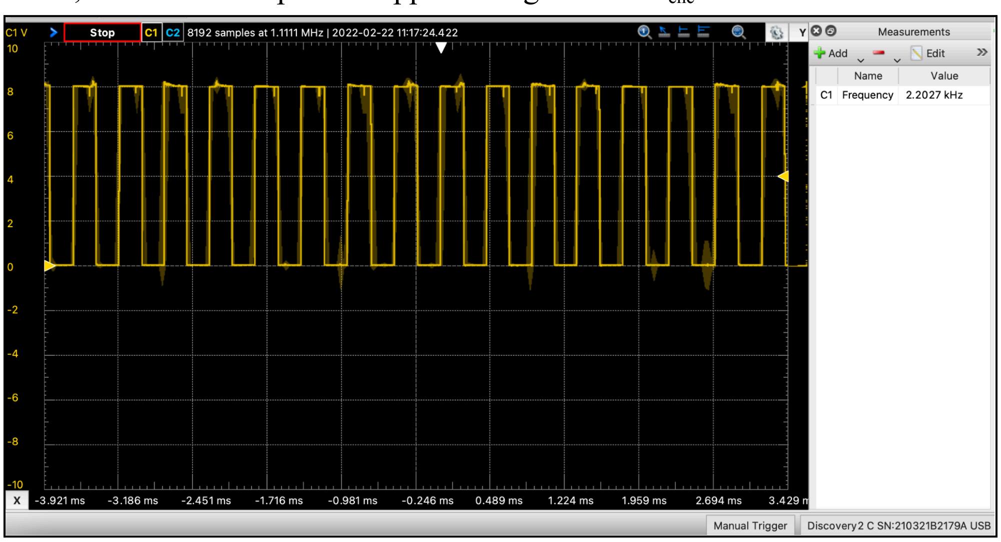


Hand Calculations for R_3

2.A.3 Connect Motor and Encoder

Throughout this lab, some of the most important measurements come from the Motor -V_{DC} and more +V_{DC}, which are represented by V_P and V_I respectively. We also connect our encoder wires to circuit power and circuit ground. We can measure the encoder frequency anywhere on our board that is separated from the rest of the circuit. We measured V_P to be around 6.6 V and V_I to be around 150 mV.

The figure belows shows the pulse frequency, f_{enc}. We measured this by connecting our scope probe to circuit ground and the node connected to the encoder output. This frequency varied a lot depending on which power supply we used. In this example the max frequency we measured was around 2.2 kHz, but with other power supplies we got a max f_{enc} of around 2.5 kHz.



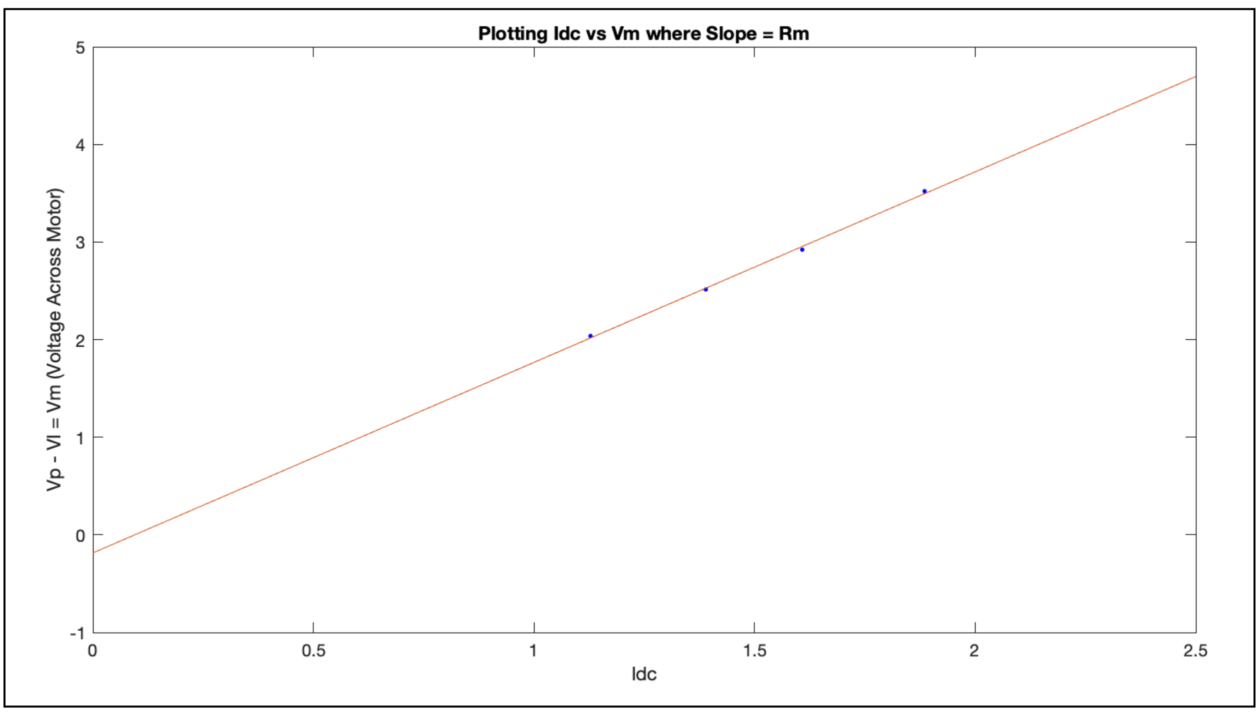
Measuring the f_{enc} frequency with scope probes

2.A.4 Determining Motor Parameter R_M

In order to determine R_M we took four different measurements of the voltage at V_P and the voltage at V_I. We know that the voltage across the motor, $V_M = V_P - V_I$. We also know that the voltage at V_I is the voltage across the resistor, R_3, with respect to ground. Our R_3 value is four 1.5 Ω resistors in parallel which produces an $R_{EQ} = R_3 = 0.375$. We take $V_I/R_3 = I_{DC}$. We then use this value for I_{DC} along with the voltage across the motor, V_M, to find the resistance of the motor. We measured $V_M = V_{DC}$ and I_{DC} at various voltages to attain an accurate measurement of R_M. A table of our data points is shown below along with a plot where the slope of the line of best fit is used to represent our R_M value. We found our $R_M = 1.95$ Ω.

V_{DC} (V)	3.520	2.921	2.514	2.037
I_{DC} (A)	1.885	1.608	1.389	1.128

Data points for V_{DC} and I_{DC} to determine R_M



Plot of I_{DC} vs V_M in order to find R_M

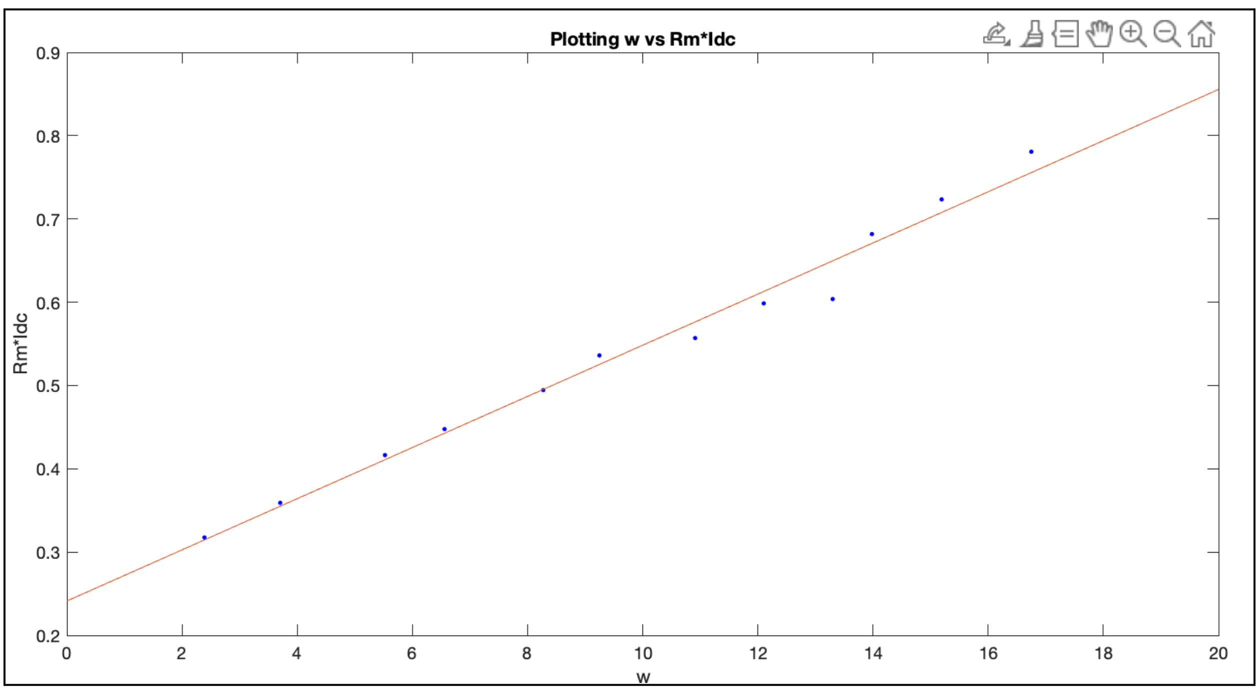
2.A.5 Determine Motor Parameters k, B, and T_{INT}

We performed essentially the same calculations from our Lab 0. First we measured the voltage at V_P, V_I, and V_{ENC}. We know that our $V_{DC} = V_P - V_I$, and when we plot our waveform on wavegens we can use the measurement tool to find the frequency at different V_{DC} values. We stepped our V_{DC} down in increments of around 0.5 V, attaining a total of 12 different data points for each frequency. These data points are shown below.

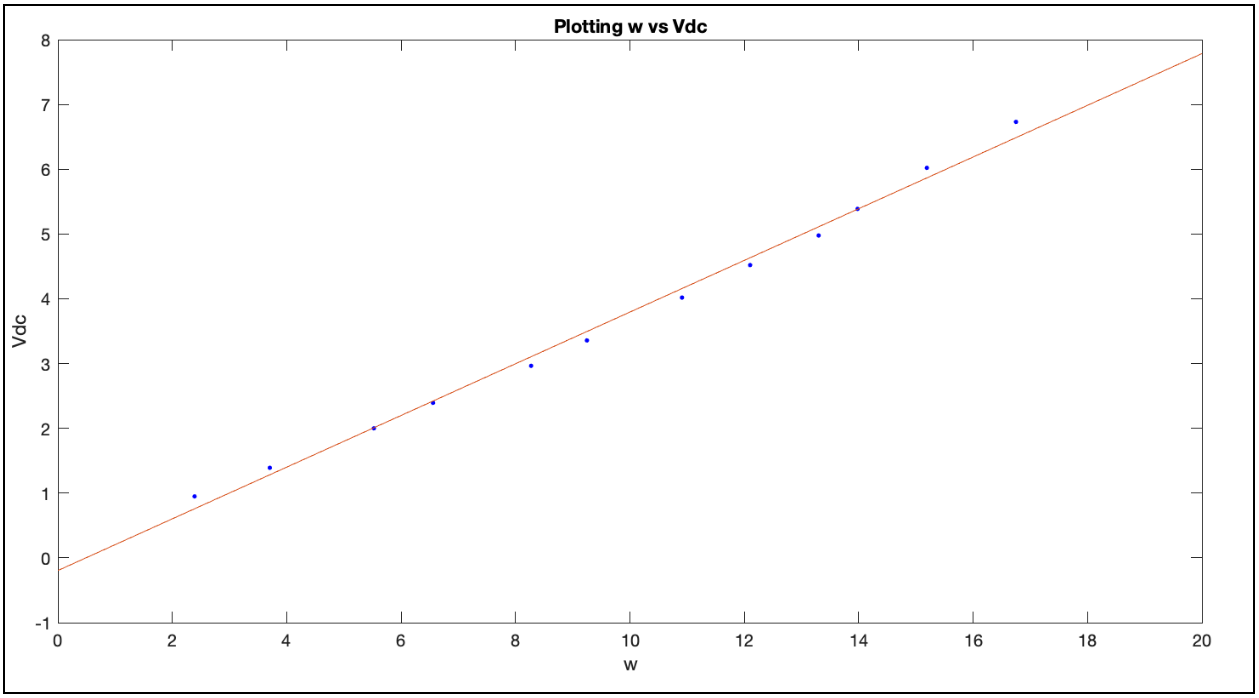
V_P [v]	6.88	6.16	5.52	5.09	4.64	4.13	3.46	3.06	2.48	2.08	1.46	1.01
V_I [v]	0.15	.139	.131	.116	.115	.107	.103	.095	.086	.08	.069	.061
f_{enc}[hz]	2559	2321	2136	2032	1849	1667	1413	1264	1002	844	566	365

Table of measurements for V_P, V_I and f_{enc}

Using the equation for Vdc we can rearrange so that we have $k = (V_{DC} - R_M I_{DC})/w$. We can split this up into two terms, $k2 = V_{DC} / w$ and $k1 = R_M I_{DC}/w$. Finding the slopes of both plots gives the values k1 and k2, which we can use to solve for k where k = k2 - k1. To find the slopes we used the polyfit function in matlab. Plots are shown below. We found that **k = 0.3683**.

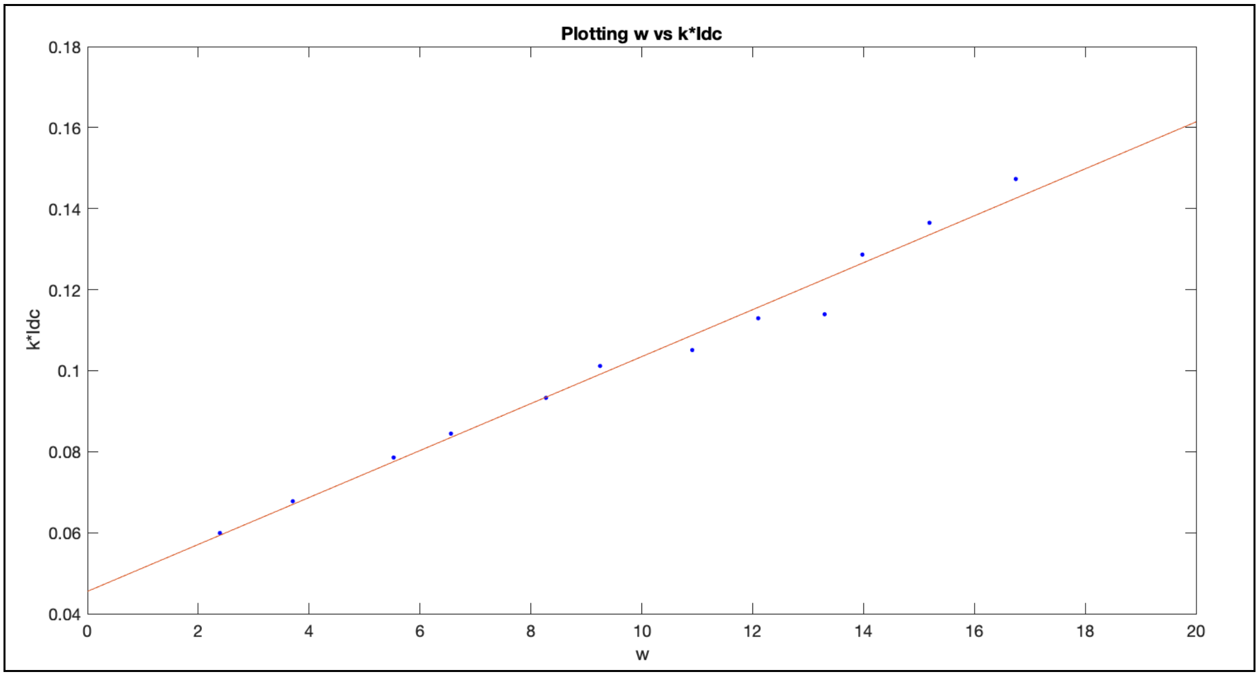


Plot of w vs R_M*I_{DC} to Find Slope Equivalent to k1



Plot of w vs V_{DC} to Find Slope Equivalent to k2

Now that we have k, we can solve for both B and T_{INT}. We know that $T_{EXT} = 0$ so T_{LOAD} is equivalent to T_{INT}. We can solve for T using $T = k*I_{DC}$ and substitute this into the equation at steady state giving us the following equation: $k*I_{DC} = B*w + T_{INT}$. We have points for both $k*I_{DC}$ and for w, so we just need to find the slope and intercept of this simple linear equation. Polyfit is perfect for this. We use polyfit to graph a line of best fit across the given data where the slope of the line represents B and the y-intercept represents T_{INT}. Calculations are shown below as well as the plot which results in **B = 0.0059** and **T_{INT} = 0.0455.**



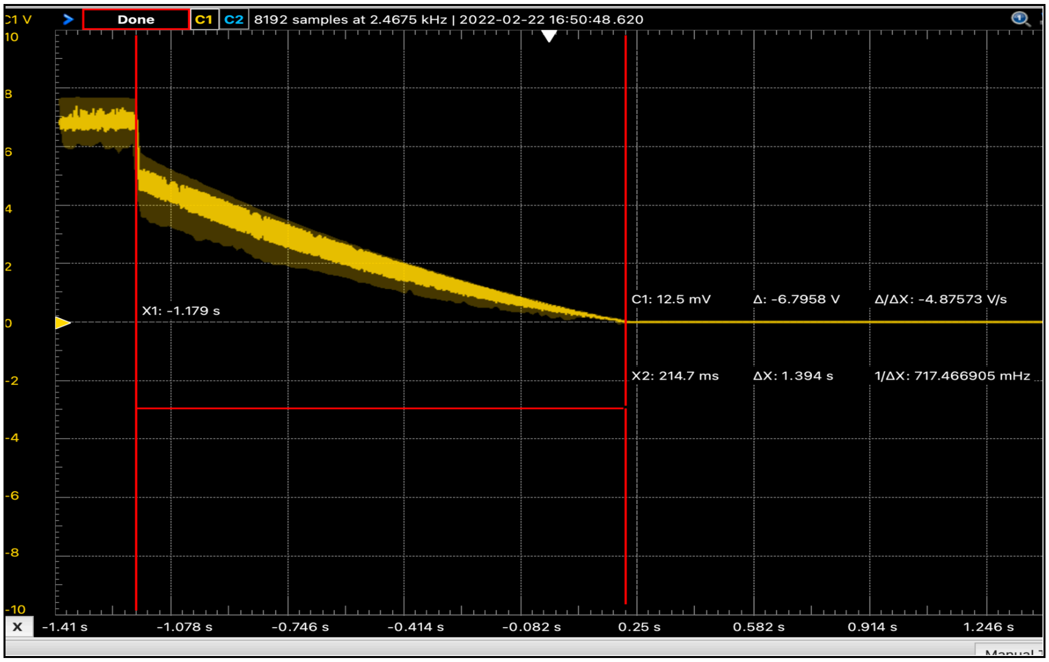
Graph of k*I_{DC} vs w

2.A.6 Determine Motor Parameter J

In order to find J, we must first derive the Differential Equation that represents the circuit. After a few calculations we are left with the equation:

$$\omega(t) = (\omega_0 + \frac{T_{INT}}{B})e^{-Bt/J} - \frac{T_{INT}}{B} \quad \text{for} \quad t > 0.$$

We know that the initial frequency is the frequency at the max voltage, with our potentiometer completely turned. This $f_{enc} = 2559$, with this we can calculate our ω_0 value using the equation $\omega_0 = 2\pi f_{enc}/960$. We find that $\omega_0 \approx 16.75\ rad/s$. Now we need to find the time taken for V_{DC} to equal 0. We use the trigger feature on the AD2 scope to achieve this. A plot is shown below of our results.

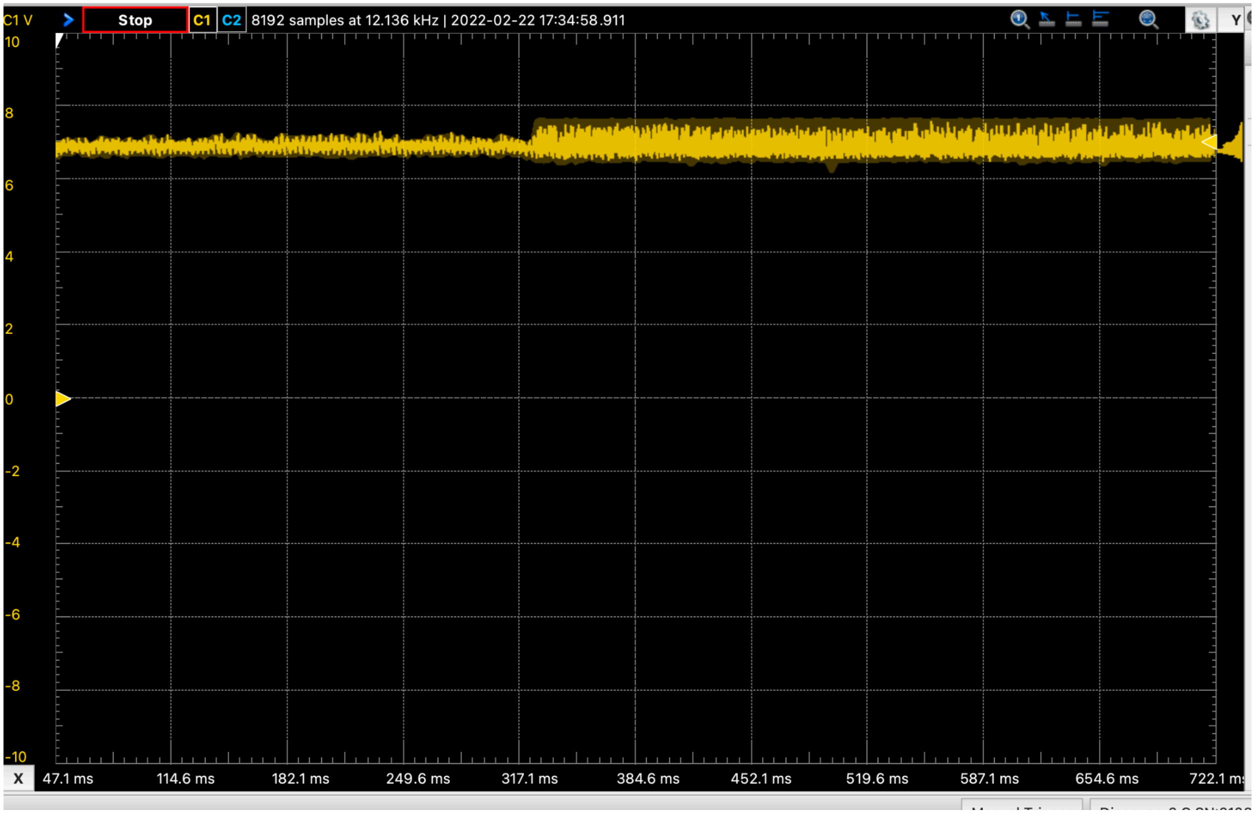


Transient Response of Motor Turn-Off at w output

By looking at the plot of V_{DC} vs time we see that it takes about 1.394s for $V_{DC} = 0$. We let $\tau = 1.394s$ which represents the time at which $\omega(t) = 0$. We can plug all of these values including B and T_{INT} that were calculated in the previous section to attain a numerical value for J.

$$\omega(\tau) = 0 = (\omega_0 + \frac{T_{INT}}{B})e^{-B\tau/J} - \frac{T_{INT}}{B} \quad \rightarrow \quad J \approx 0.00794\, kg \cdot m^2$$

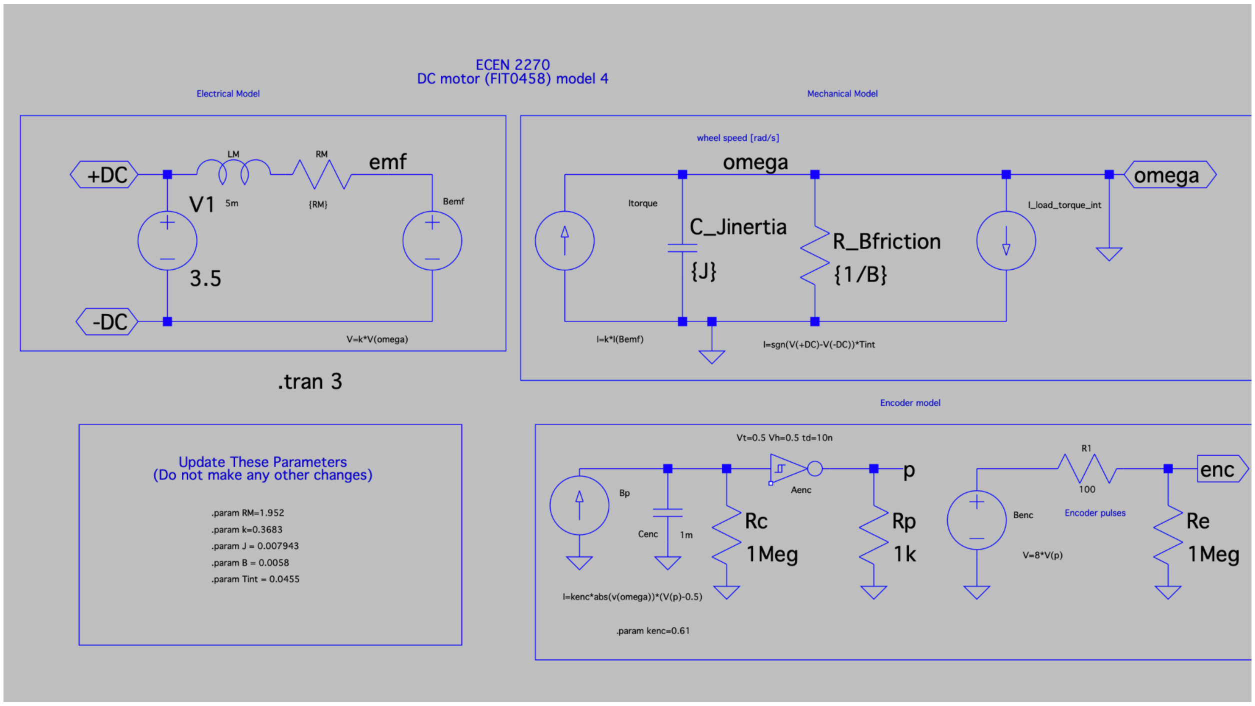
It is important to note that we ran this transient response of the motor power being turned off many times. We saw the decay time or τ values range anywhere from 1.30s to 1.54s. This is most likely due to inconsistencies in the wheel's temperature and friction affecting how fast the wheel slows down and comes to a stop. We decided to choose the most common value which was about 1.39s. The motor supply voltage noise could be reduced by placing a decoupling capacitor from the V_P terminal directly to ground. The graph below shows the decoupling capacitor included vs excluded.



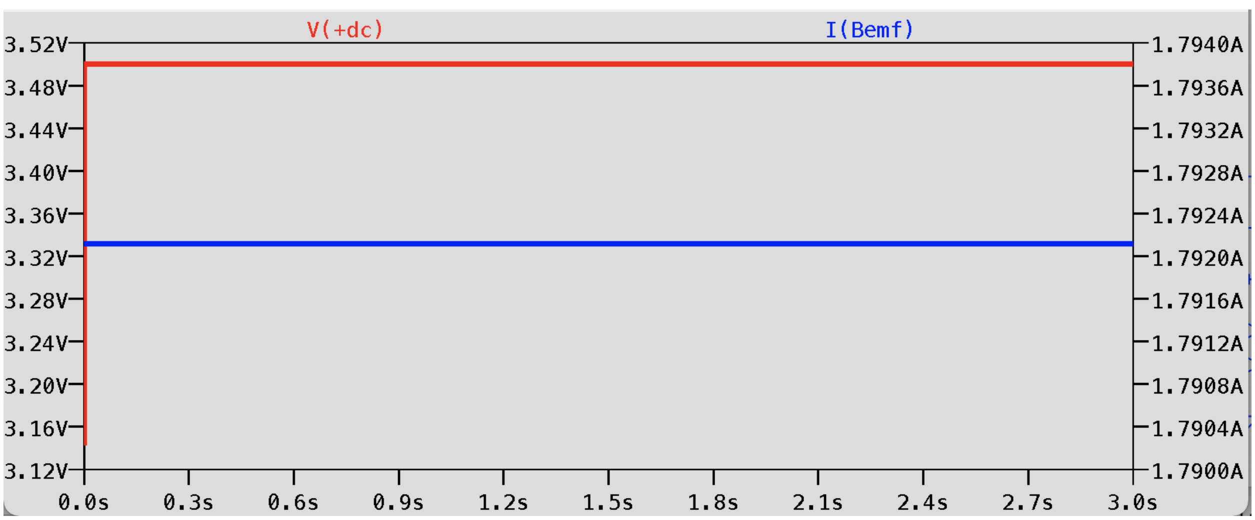
V_{DC} Noise with decoupling Capacitor (left noise) and without decouple capacitor (right noise)

2.A.7 Validate Motor Model

We can use the DC motor model to simulate the different experiments we did in real life and confirm the results we found. First by connecting a DC power supply across our V_{DC} terminal and then setting our omega node to ground, which is the equivalent of testing V_{DC} and I_{DC} with the locked motor. We can apply different voltages across V_{DC} and test the corresponding I_{DC}.
The LTSpice updated model and corresponding time domain response is shown below.



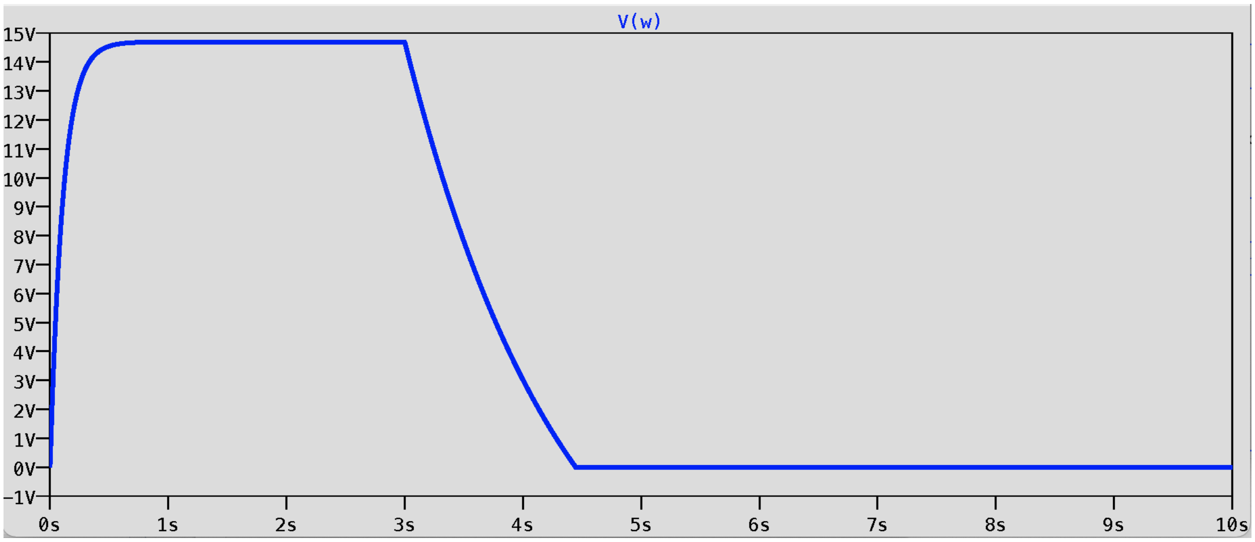
LTSpice Model for Locker Motor (omega = 0V)



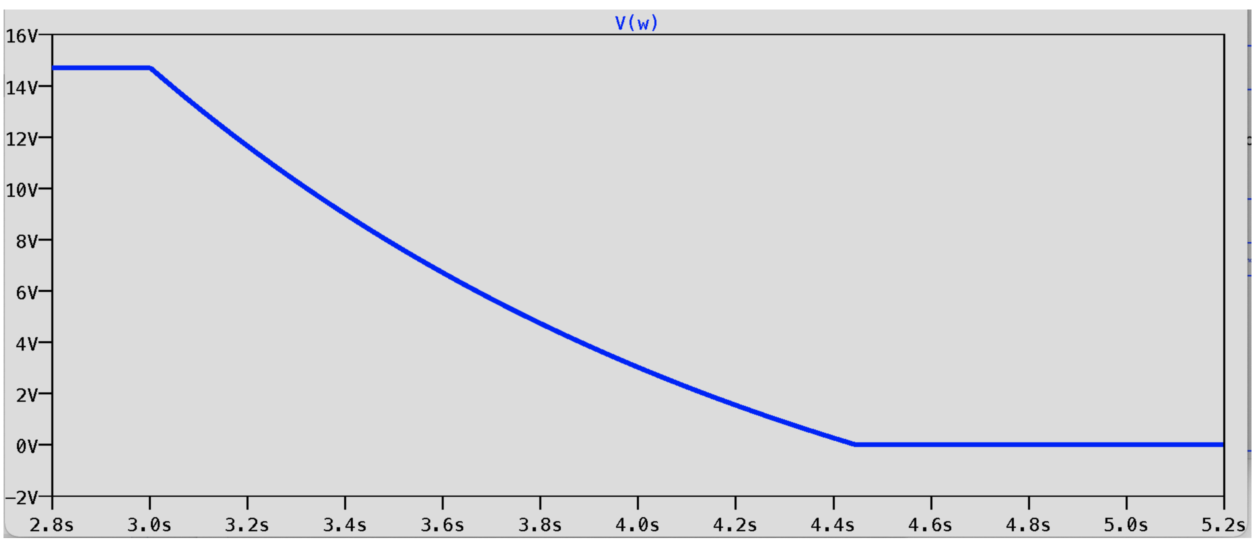
Output of Locked Motor Test

From this test we can see that with a V_{DC} of 3.5V, we get an I_{DC} of about 1.8 A. In our tests we measured and calculated an I_{DC} of about 1.88 A, so this is what we expect.

We can also confirm our results for J, B, and T_{INT} by simulating the decay time for our motor when we turn off the power supply. Below is a simulation of this LTSpice simulation where the motor supply is turned off at time = 3 seconds. We see that the decay time is around 1.45 seconds which is roughly what we calculated in our actual motor. This value is most likely higher because it is a simulated value whereas in our real motor we were getting decay times ranging from 1.3 to 1.54. This is right in this range.



Simulated Response of Motor speed after turning off power supply



Zoomed in Simulated Response of Motor speed after turning off power supply

2.B Lab Exploration Topic

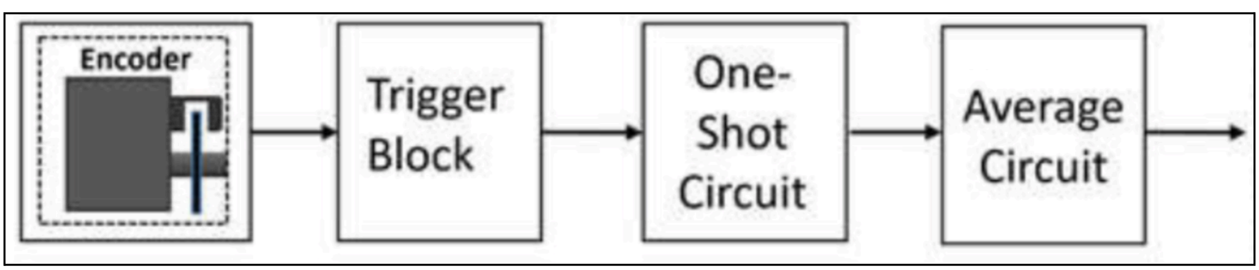
1) R1 should be chosen to be much greater than RT, because if R1 was a comparable value to RT it would create a voltage divider, and we would lose part of the signal voltage across RT. We want the entire signal to be passed through the RC filter so R1 >> RT. The same principle applies to Rin. We want to pass the entire voltage signal to our load, so we

choose a value of R1 << Rin, so that a high enough voltage is passed across the load resistor and capacitor.

2) C1 should be much greater than Cin. Because parallel capacitors add like series resistors, our time constant τ essentially equals R1*(C1 + Cin). Thus, if C1 is not much greater than Cin, our RC filter will be affected.
3) The input to our trigger circuit is the f_{enc} frequency passed to our system from the motor chip. This signal is a PWM signal with 8V amplitude and a 50% duty cycle. Depending on our potentiometer and how much voltage is supplied to the motor, the frequency of f_{enc} will vary- higher supply voltage means a higher frequency, and a lower supply voltage produces a lower frequency. The load of this signal is the LM555 timer, and more specifically, this signal is passed through the trigger pin and into a comparator inside the component. The comparator has an ideal input resistance of infinity, so we can say that the load of our signal is essentially an open circuit.
4) A greater value for R1 will decrease the overall current drawn by the load. Thus R1 should be chosen with ideal load supply current in mind. If we have a leaky capacitor with a high impedance resistor in parallel, then R1 should be much smaller than the non-ideal 1MΩ resistor, otherwise we will see a voltage drop across R1 that will affect the signal passed to our load.

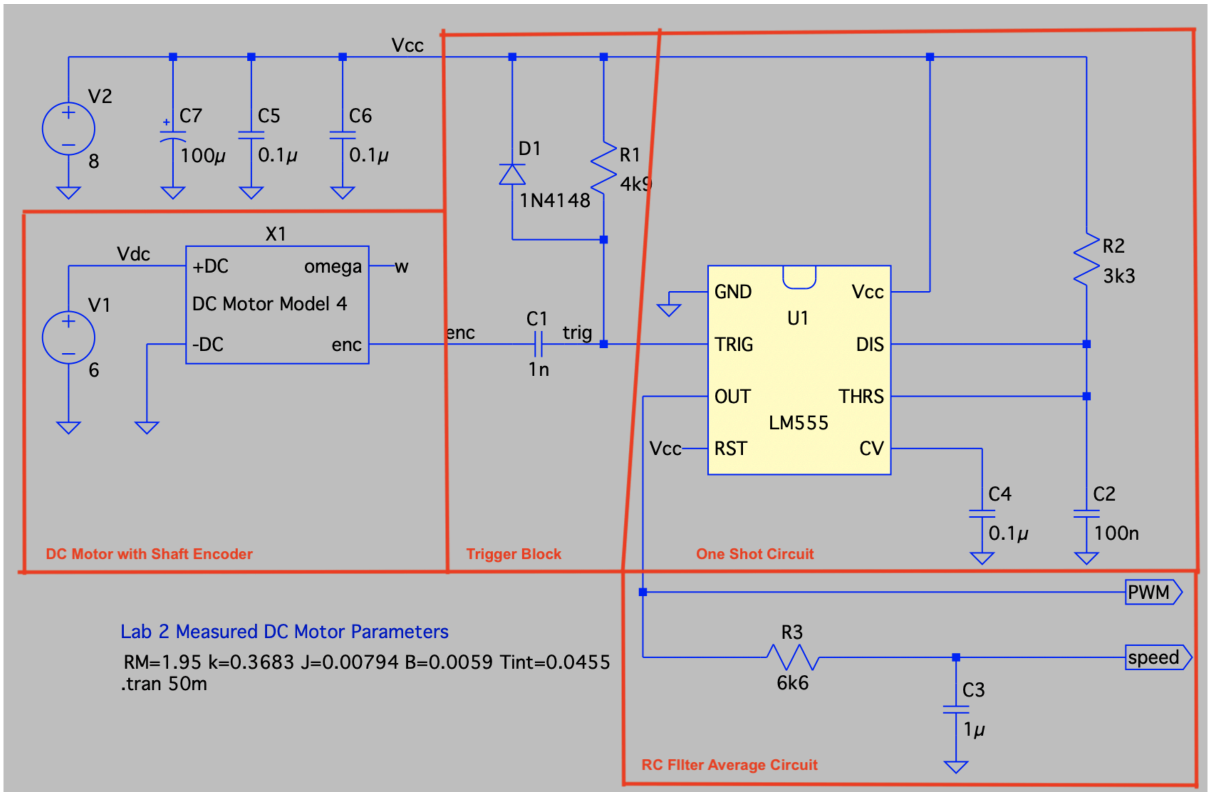
2.B.2 Speed Sensor Circuit Design and Simulation

Once the DC motor parameters have been found and verified, the issue of making a speed sensor for the motor can be approached. In order to achieve this goal, the encoder frequency coming out of the motor must be converted to an analog voltage output with little to no noise. In order to do this, a circuit following this block diagram is used to get a linear relationship between encoder frequency and speed sensor output voltage:



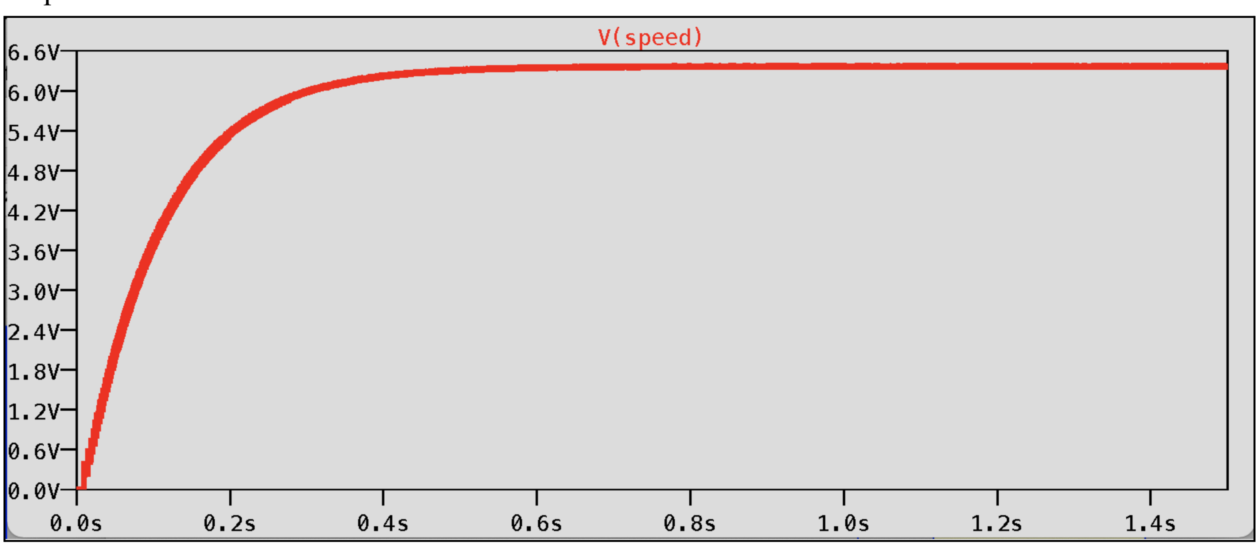
Block Diagram of Speed Sensor Circuit

We build the following LTSpice model that was derived from the block diagram above. Discussions of how we chose each R and C value is described in later sections of this report.

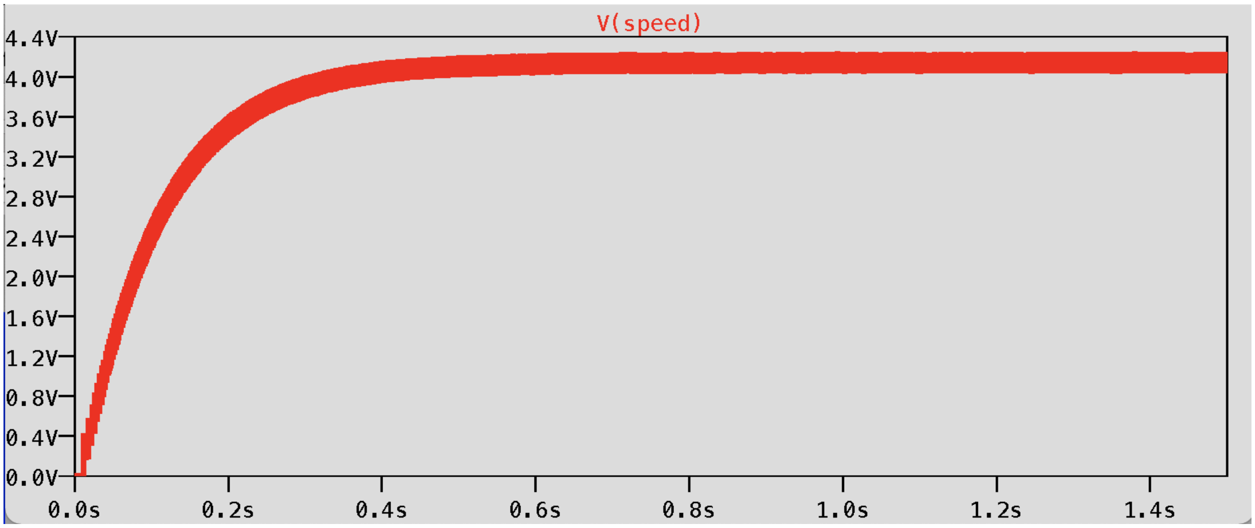


Speed Sensor Model in LTSpice with Corresponding Blocks

We show two different transient responses below that depict the linear relationship between our motor voltage and the corresponding speed sensor output voltage. It is clear that as we decrease our motor voltage we get a lower speed sensor output as well as more noise due to the fact that our encoder frequency is lower and the RC LPF will allow more voltage through it at lower frequencies.



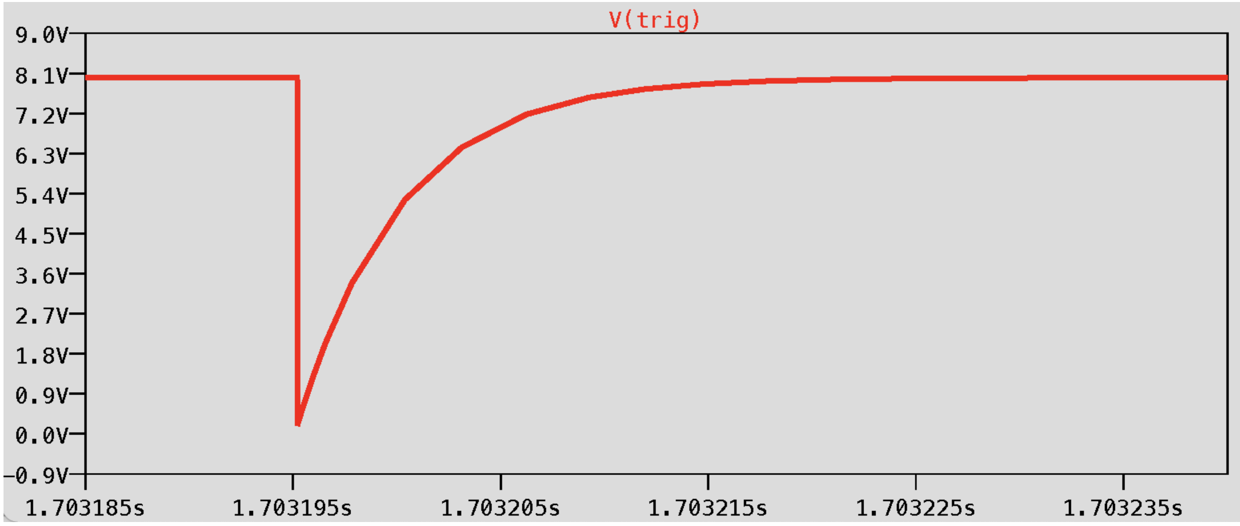
V_{DC} = 6V and the Corresponding Speed Sensor Output Voltage



V_{DC} = 4V and the Corresponding Speed Sensor Output Voltage

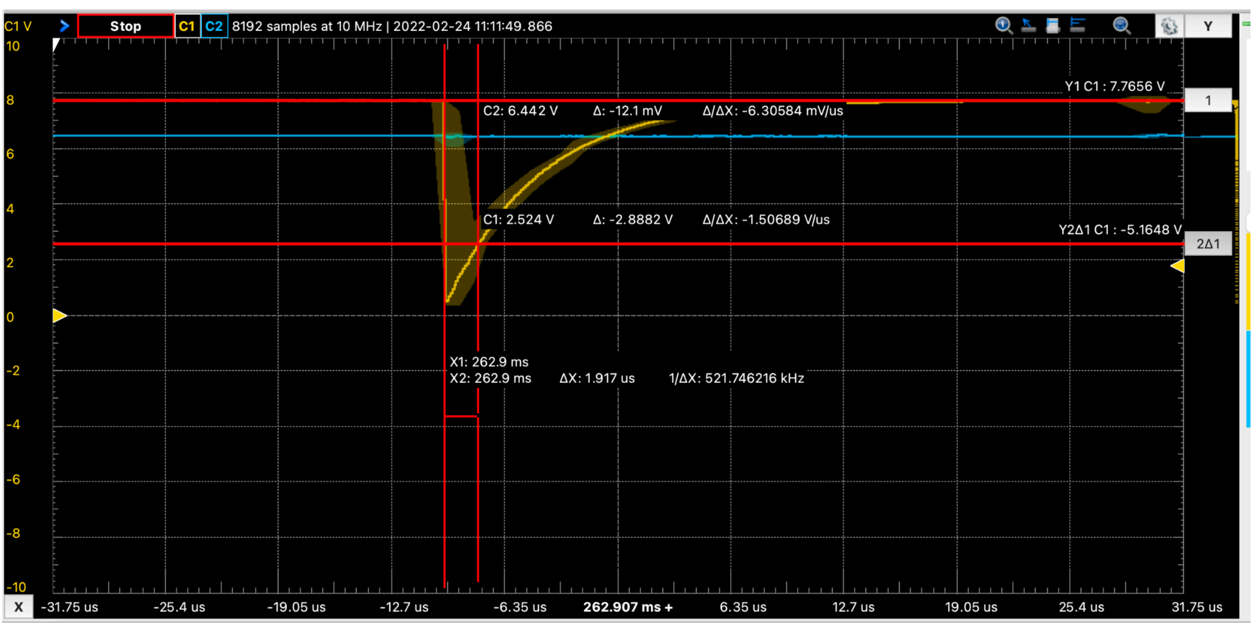
2.B.3 Build Trigger and "One Shot" Circuits

After building our trigger and "one shot" circuits, we want to measure our value for t_{trig} and compare it to the value found in spice simulations. To find t_{trig} we measure the voltage at the node labeled trig. We measure from the fall time to the rise time of ⅓ total voltage. In our spice plot, this value was ~1.94µs, as seen below.



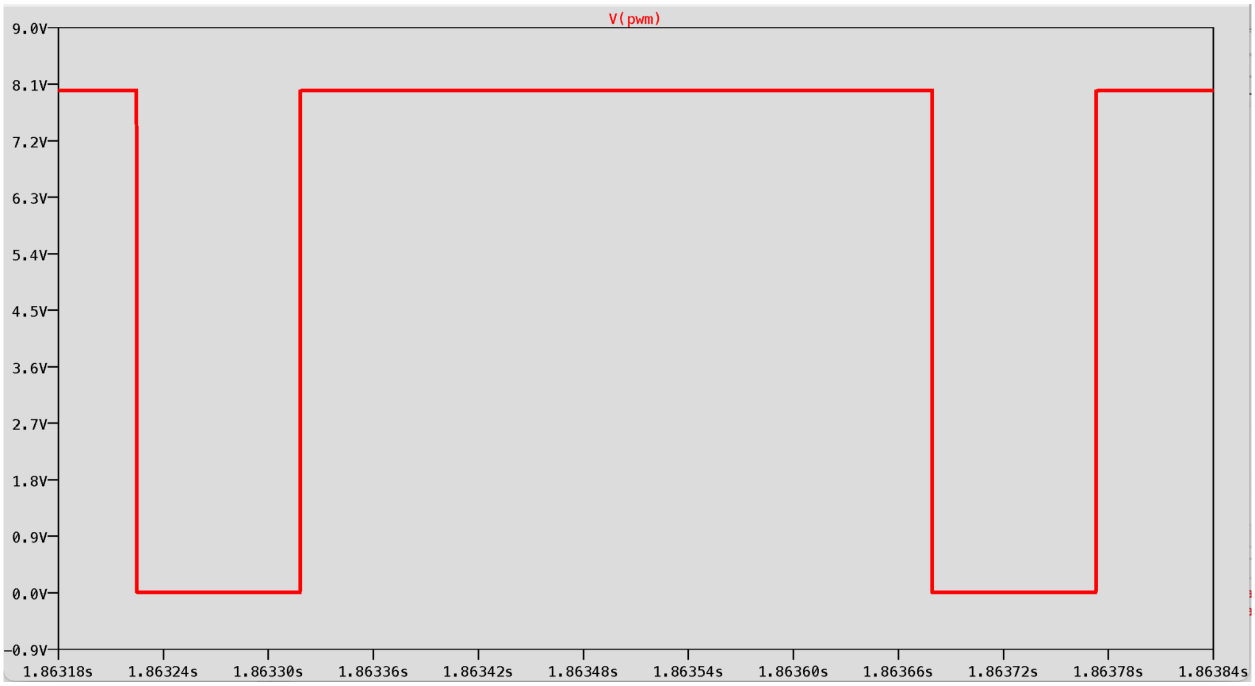
Simulated t_{trig} found to be ~1.94 µs

Calculating for values of R_1 and C_1, we use the equation $2.0\mu s = ln(3/2) * R1 * C1$, and if we choose C_1 = 1nF, then R_1 is roughly equal to 4.7kΩ. We choose this value of t_{trig} because we want it to be bigger than 1µs, but much less than t_{on}. Building and testing this circuit with our AD2 yields the scope screenshot shown below. Using our x and y cursors, we can see t_{trig} = 1.917µs, almost exactly what we simulated, thus we did not have to adjust our values for R_1 and C_1.



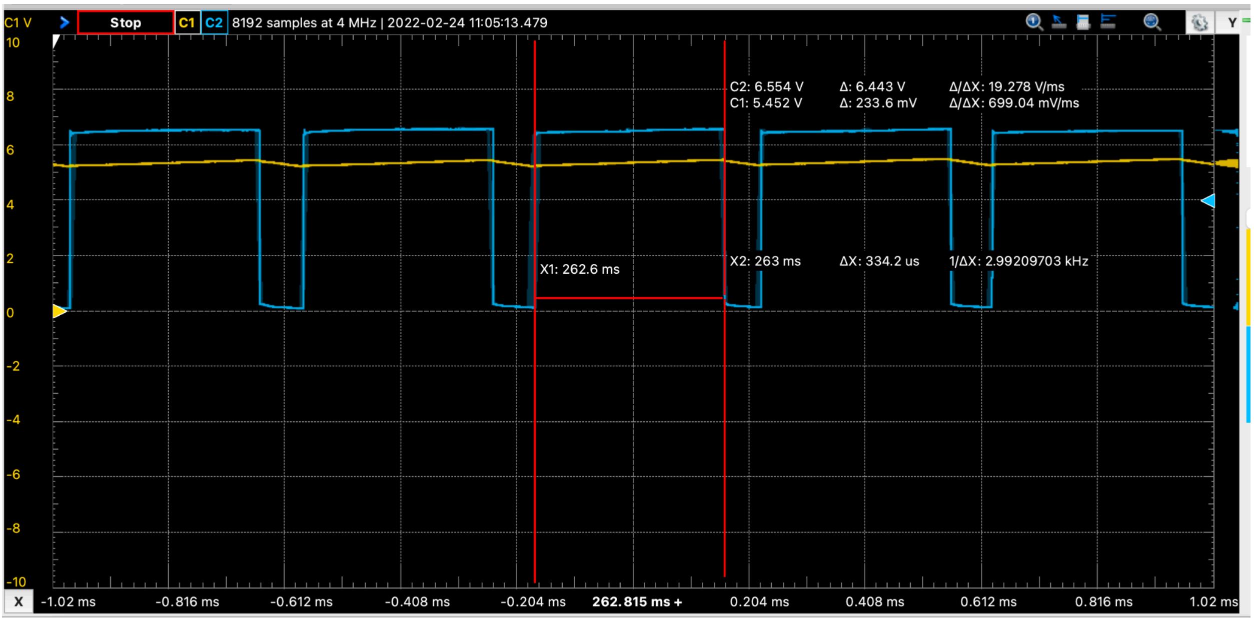
Measured t_{trig} found to be ~1.92 µs

Next we simulated our PWM node waveform. The value t_{on} is the time that the waveform is high for a given period. As seen below, this value is simulated to be ~360µs.



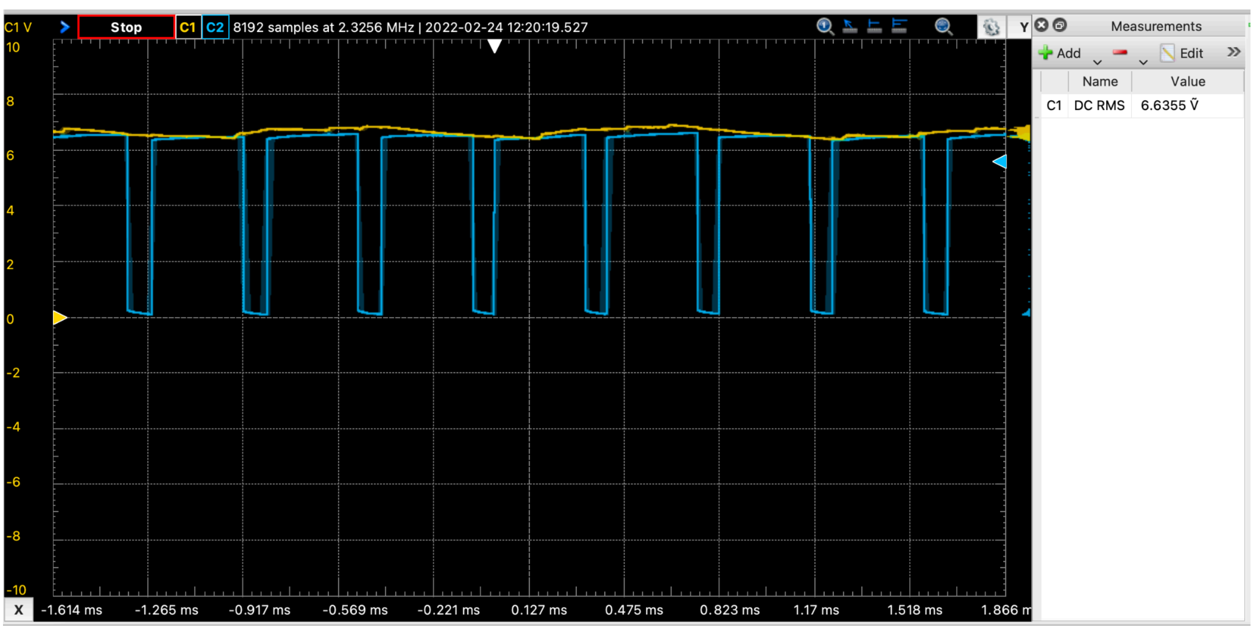
Simulated t_{on} found to be ~360µs

To choose values of R_2 and C_2, we use our max encoder frequency derived in Lab 2.A, which we measured to be ~2.5kHz. Thus we determine that t_{on} must be less than 1/(2.5k). We then use the equation $ton = R2 * C2 * ln(3)$, to solve for values of R_2 and C_2 that satisfy the above inequality. Choosing C_2 to be 100nF, we solve for an R_2 value of ~3.3kΩ. Building and probing this circuit with our AD2, we determine that our real value of t_{on} is 334.22µs, as seen below. This is less than our encoder period, and also much bigger than our t_{trig} value, so settle on these values of R_2 and C_2.



Measured t_{on} found to be ~334µs

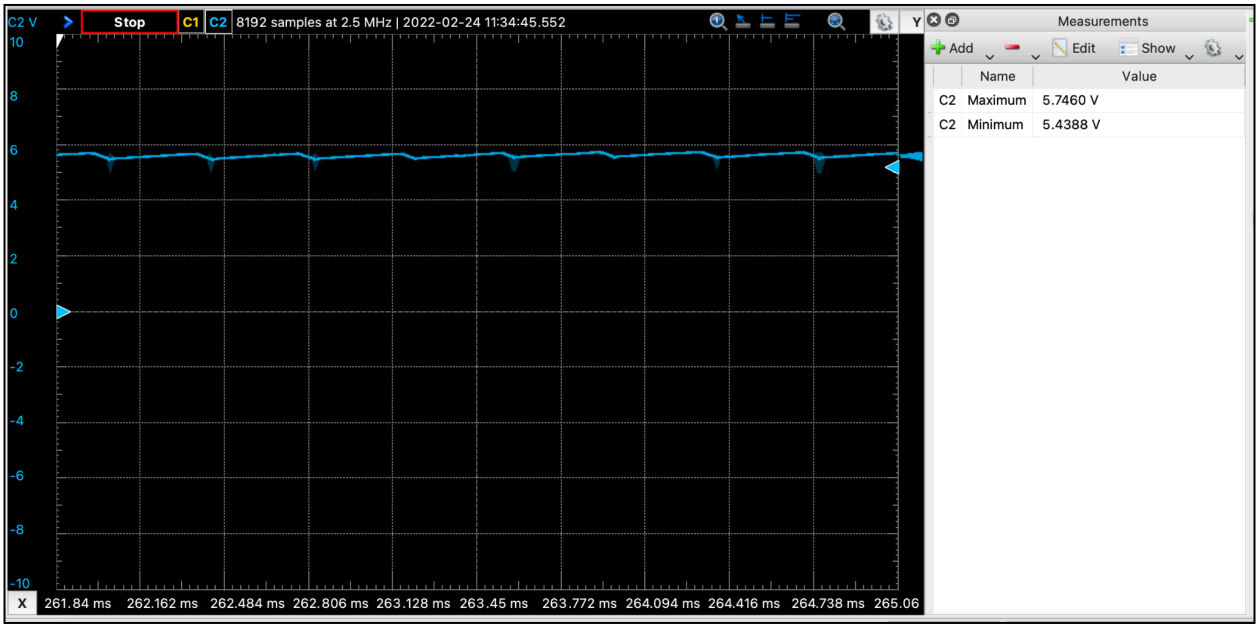
To validate our t_{on} measurement, below is a screenshot showing our maximum V_{DC} motor voltage to be ~6.64V on channel one, and our PWM waveform on channel two.



Maximum V_{DC} plotted with PWM Output Voltage

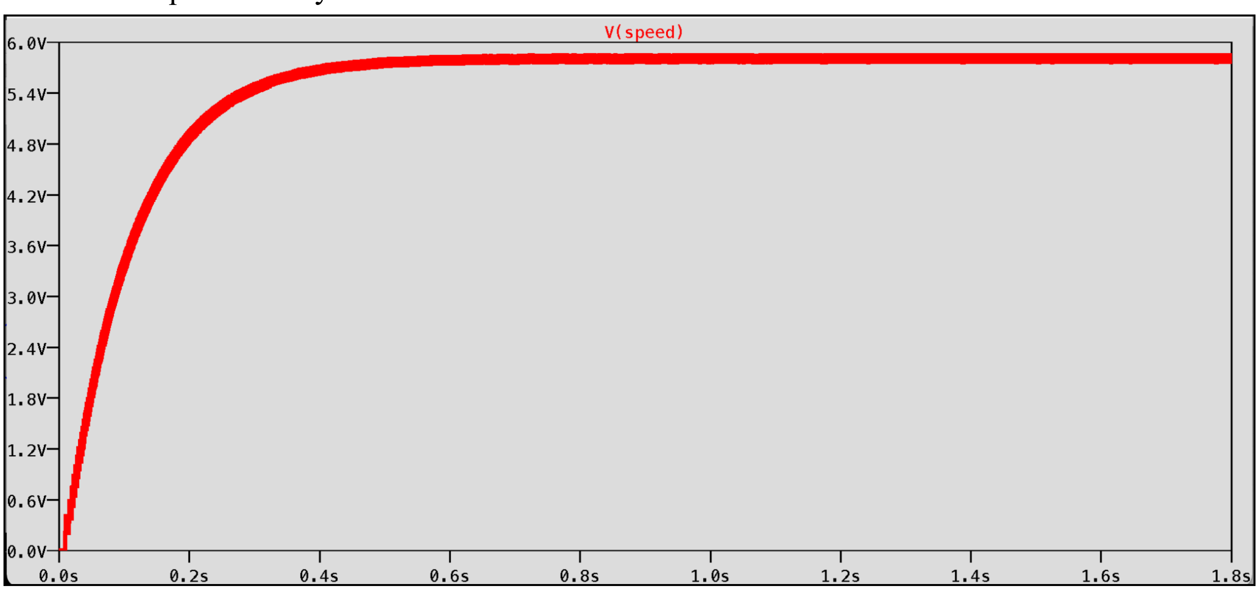
2.B.4 RC Filter and Final Speed Sensor Circuit

In order to find the R_3 and C_3 values for a low pass filter, we want to attenuate our max encoder frequency of 2.5 kHz by -40dB. To achieve this we chose an resistance value of $6.6\ k\Omega$ which we accomplished using two $3.3\ k\Omega$ resistors in series, and a capacitance of $1\mu F$. Below is a graph showing the output voltage of our speed sensor on channel 2. Notice that the V_{PP} is just around 300 mV, which is what we want. The average voltage is just around 5.5V at maximum encoder frequency.



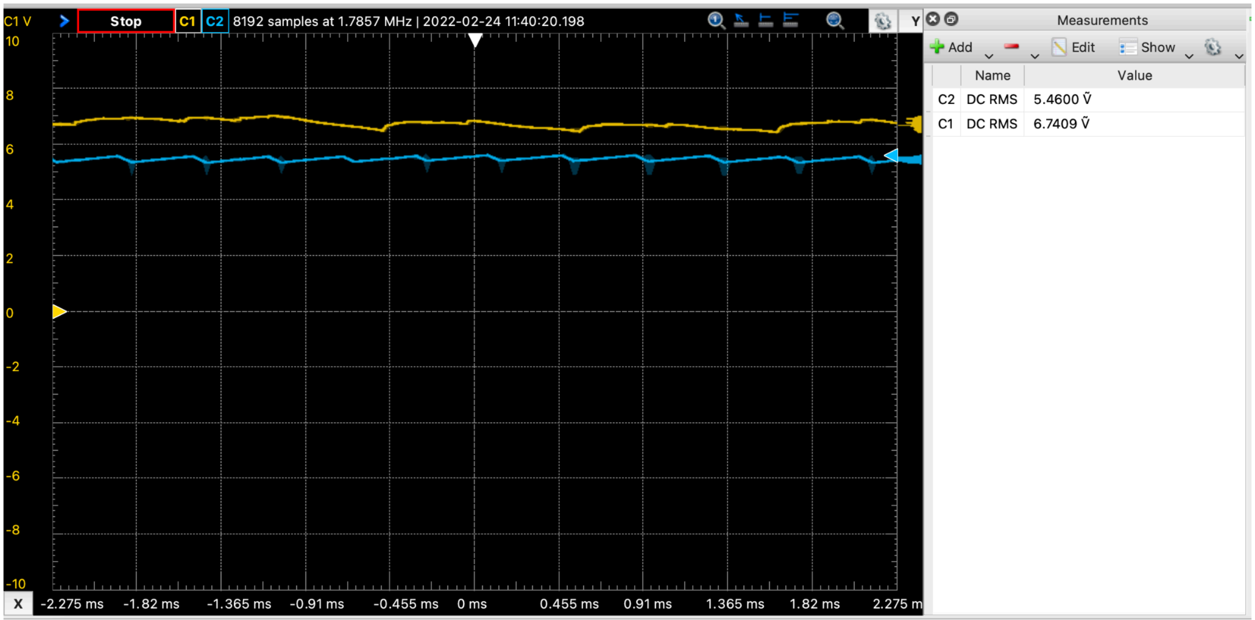
Speed Sensor Output to Verify Functionality of RC-LPF

We can also simulate our speed sensor output voltage using LTSpice to verify our results. We get the resulting graph shown below when we adjust the V_{DC} voltage to be closer to the true value of our motor voltage. We can see that the speed sensor voltage caps out at around the same voltage as we find experimentally.

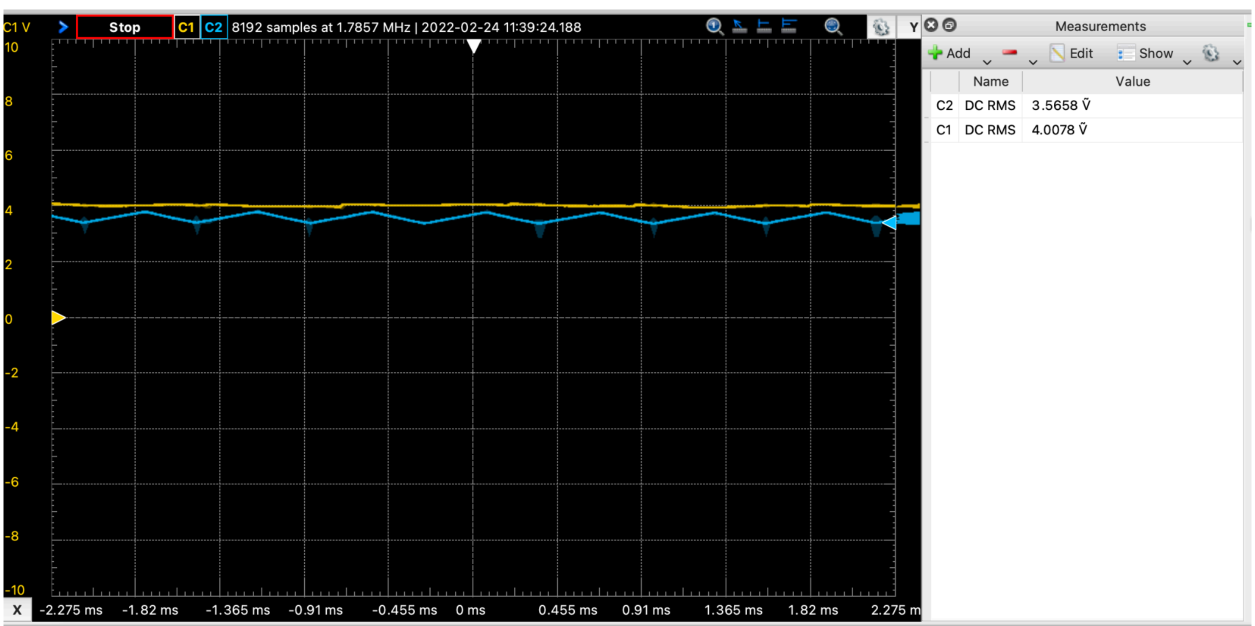


Simulated Speed Sensor Output To Verify Functionality of RC-LPF

Next we want to compare the relationship between the motor voltage V_{DC} and the speed sensor output voltage. Plots of this relationship are shown below at two different encoder frequencies.



Measuring V_{DC} (Channel 1) and Speed Sensor Output (Channel 2) at Max Speed

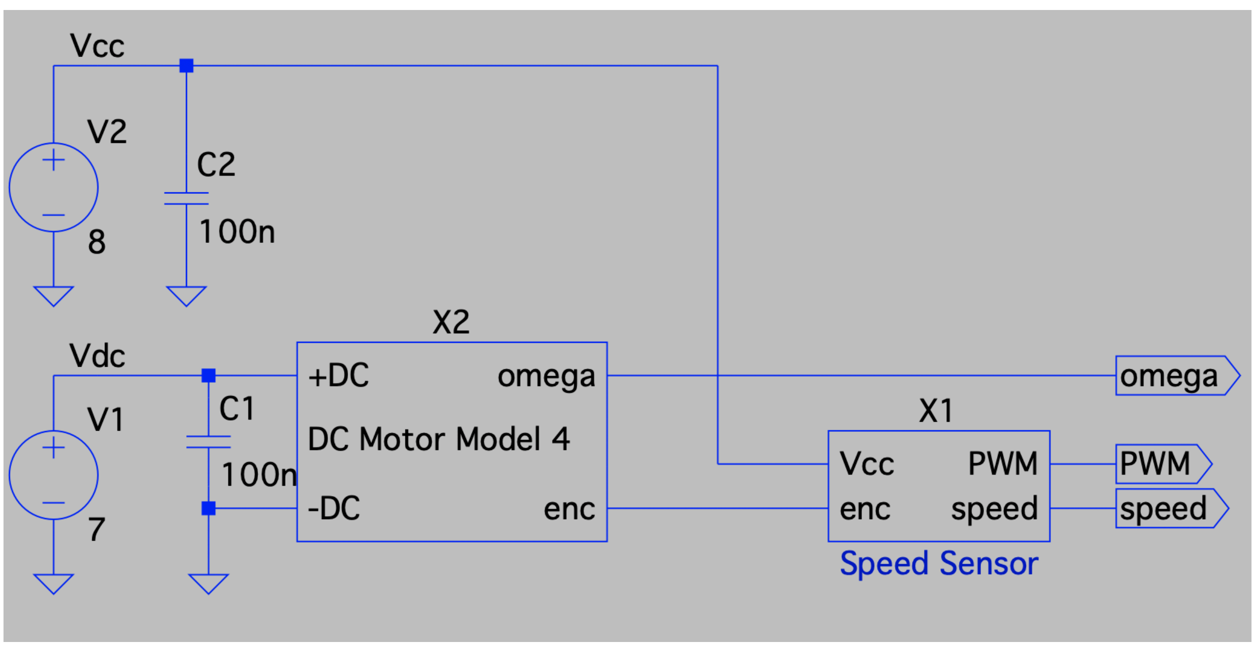


Measuring V_{DC} (Channel 1) and Speed Sensor Output (Channel 2) at ~ Half Speed

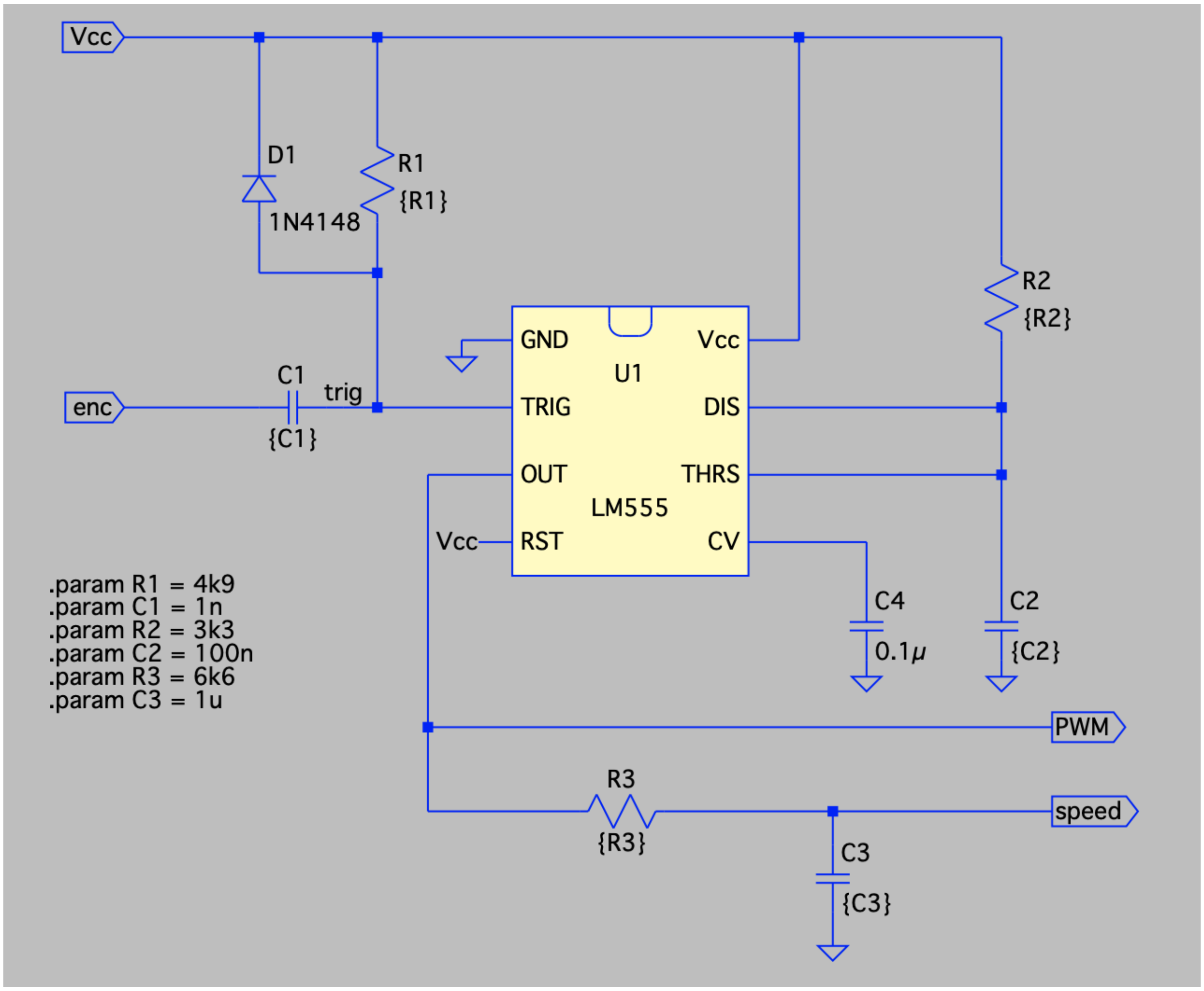
In these waveforms it is evident that as we decrease the motor voltage by turning our potentiometer, we see a corresponding decrease in our speed sensor output voltage. There is a clear linear relationship between these values. The speed sensor output voltage tends to be slightly less than our V_{DC} for all values from 0 to 8 V. We can also notice that the slight noise visible in C2 has a slightly smaller frequency and a slightly more noise due to the functionality of a low pass filter.

2.B.5 Speed Sensor Sub-Circuit, Model Validation

By turning the trigger block, one-shot circuit, and RC low pass filter sections into its own LTSpice model, the speed sensor circuit can be represented in a simple and easy to read model.

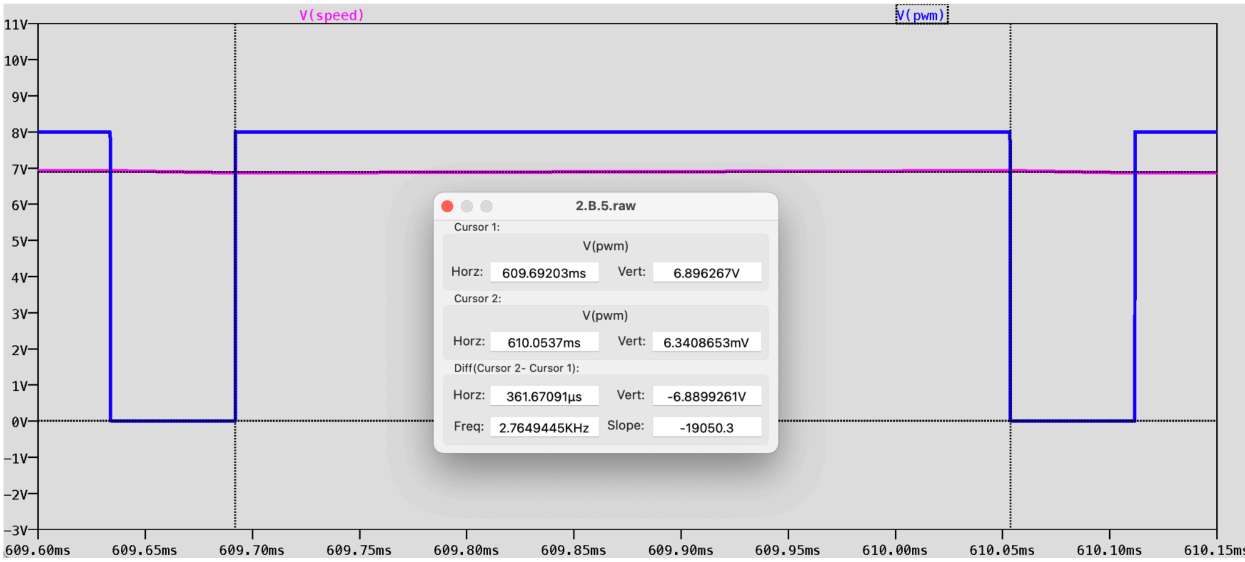


Final Circuit with Speed Sensor Block

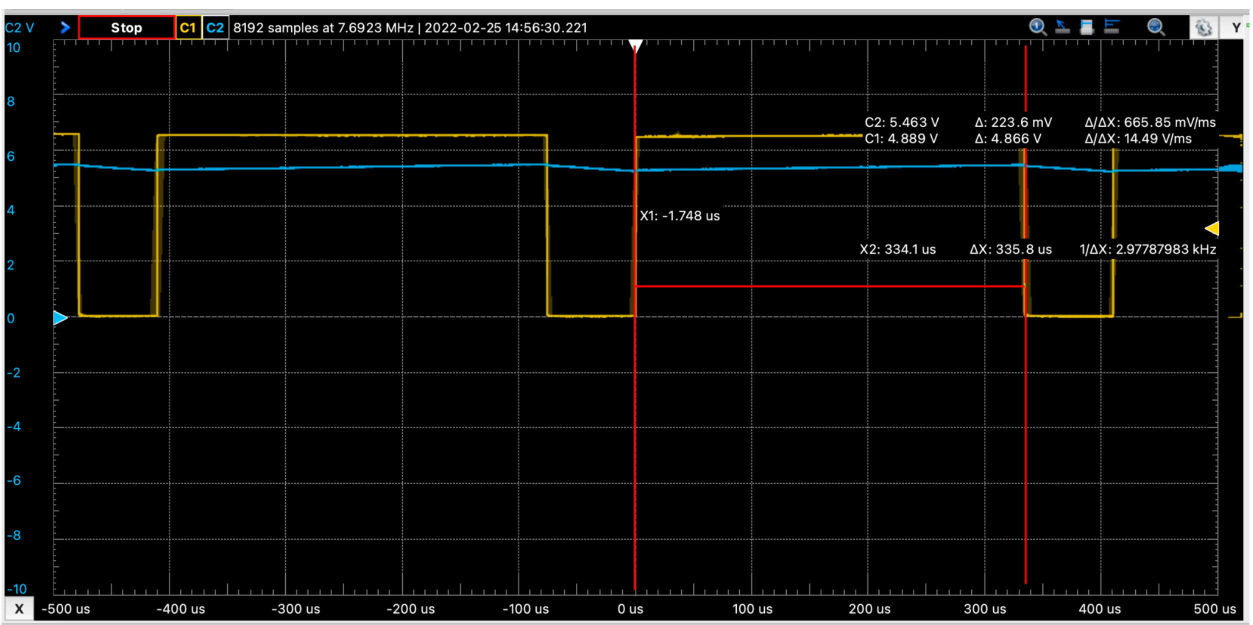


Full Model of Speed Sensor Block with R and C Parameter Values Shown

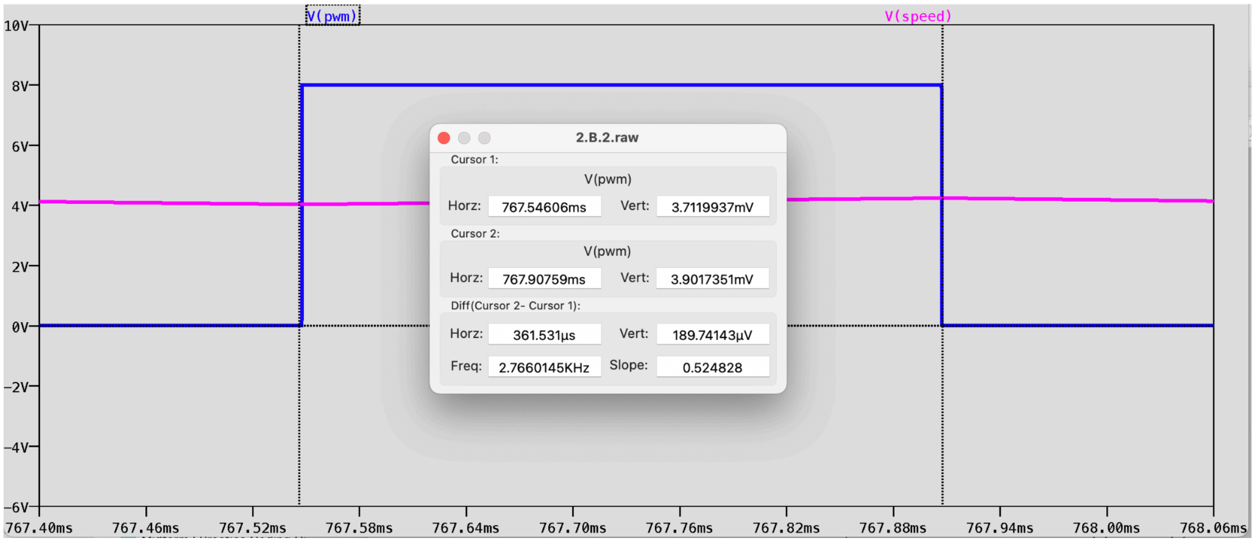
In order to verify the correctness of our final simplified LTSpice model, we run a couple different simulations with this model. First we verify the correctness of our PWM and Speed outputs with a V_{DC} of 7 V. We see below that the t_{ON} is approximately 360 µ*s* which is exactly what we expect at any V_{DC} value. The t_{on} stays the same, but the period increases with a decrease in V_{DC}.



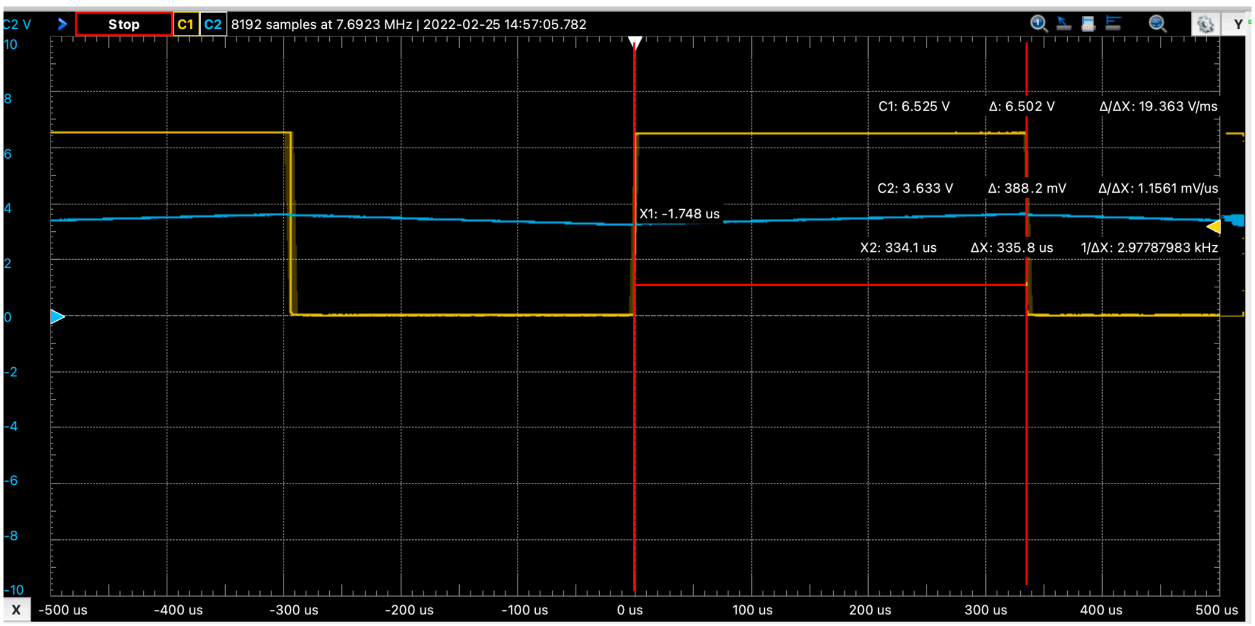
Simulated Speed Sensor Output and PWM output with V_{DC} at 7V



Actual Speed Sensor output (Channel 2) and PWM (channel 1) with V_{DC} ~ 7 V



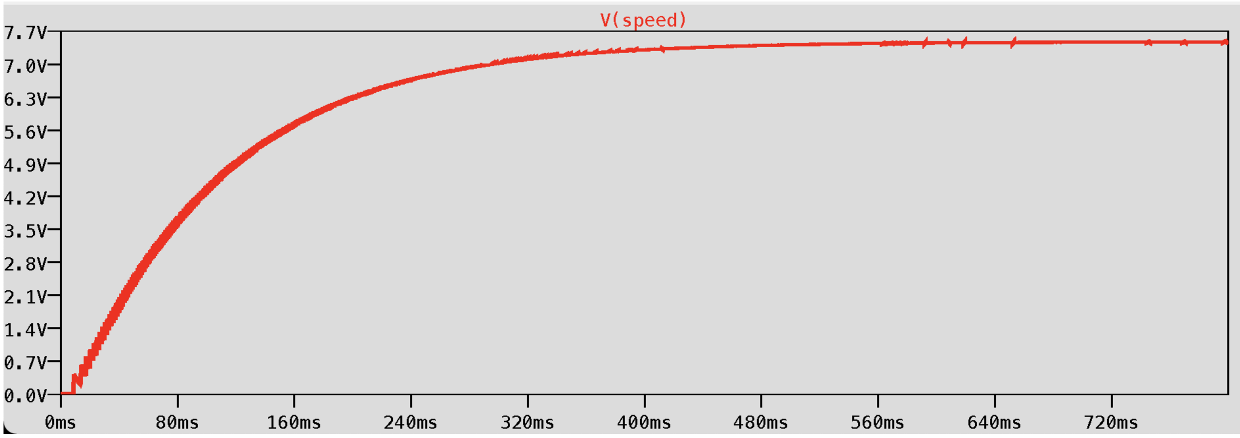
Simulated Speed Sensor Output and PWM output with V_{DC} at 4V



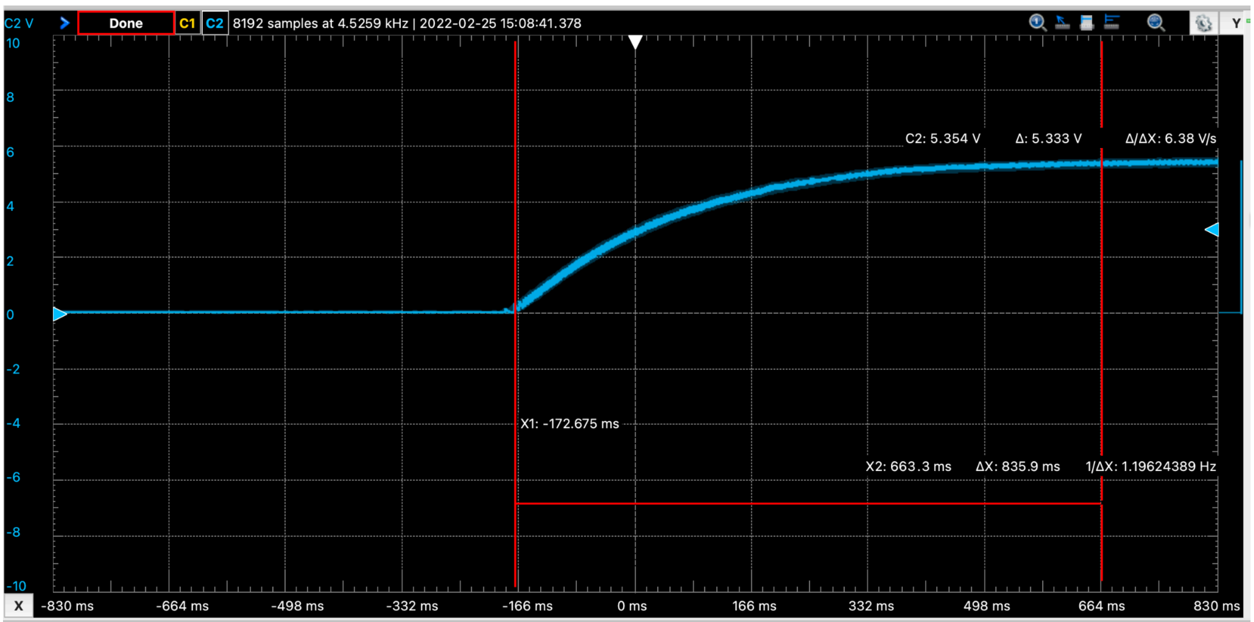
Actual Speed Sensor output (Channel 2) and PWM (channel 1) with V_{DC} ~ 4 V

From these results we can once again conclude that our motor parameters are fairly accurate. The only major discrepancy in our V_{speed} at higher V_{dc} values. Our hardware circuit measured speed sensor output is about 1V less than what we simulate. This is most likely due to some error in our R_M measurement. If we change our R_M parameter to about 4Ω we get something closer to what we are actually experiencing in the hardware circuit. However at lower V_{DC}'s our speed sensor voltage is right where we expect it to be.

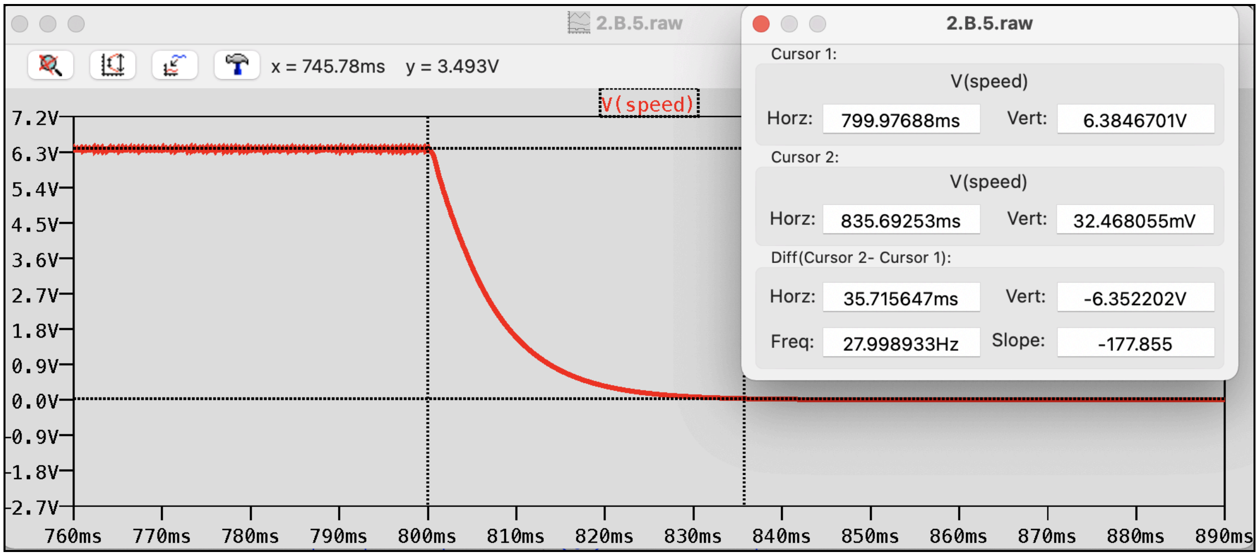
We can also run a transient analysis of both the motor start-up and motor turn-off in simulation and on our circuit to further confirm our motor model and parameters.



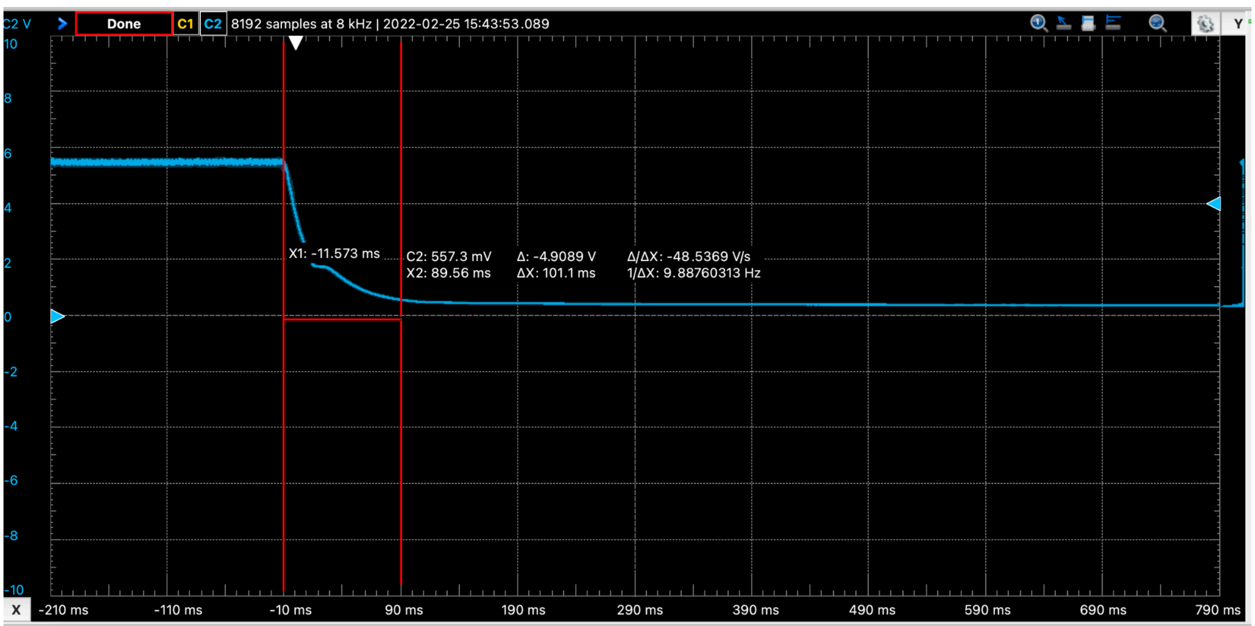
Simulated Motor Startup with V_{DC} = 7V



Actual Motor Startup at Max V_{DC} ~ 7V



Simulated Transient Response of Motor Turnoff for Speed Sensor Output



Actual Transient Response of Motor Turnoff for Speed Sensor Output

After adding in switches to our LTSpice circuit we can model the behavior of turning off our power supply and simulating the corresponding result. We see that the transient response of the motor turn on is just about what we expect where it takes approximately 800 ms for the motor to reach maximum voltage and thus speed.

We also run a transient response for the motor turnoff. This is less comparable to our simulation. The simulation turn off time is only about 35 ms, whereas our real circuit takes about 90 ms. This is most likely due to some errors in our measurements or the non-idealities being ignored in the simulation. We obviously try to model the behavior of a motor to the best of our ability using real electrical components, but they may not always behave the same way that an actual motor does. We can however expect the turn off time for the voltage at V_{DC} to be very similar to what we simulate. This was confirmed in part 2.A of the lab.

Overall, the simulation behaves very similar to our actual hardware circuit. In the future we can reliably use this model to simulate certain functionalities of our circuit like speed and supply voltage relationships. We can less reliably use it for other measurements such as the decay time of our speed sensor output in response to turning off the supply.

Conclusion

We achieved all of our objectives throughout this lab, which entailed simulating and building our DC motor supply circuit, as well as our frequency encoder circuit. Additionally, through the implementation and testing of our circuits we were able to measure and calculate our motor parameters, and affirm that they were within a reasonable range of the ideal values. Through our encoder circuit and the use of the LM555 timer we converted our encoder signal to a DC voltage signal proportional to the speed of the wheel shaft. Looking ahead, we think this signal could be passed in to a microcontroller, such as the Arduino Nano Every, to perform digital operations and vary the DC motor supply voltage.

One limitation we encountered in this lab was the sampling rate of our AD2 oscilloscope. When measuring our t_{trig} value we found that the AD2 did not have a high enough sampling rate to accurately measure our 1.9μs t_{trig} value, and instead we had to use the high end lab oscilloscopes, which we never explicitly instructed on usage. We also ran into several discrepancies in our measurements due to an inconsistent battery powered source. To solve this problem we used the DC power supply’s in the lab that provide more reliable sources.

Throughout the lab we followed a process of simulating and analyzing each subcircuit in LTSpice, and then implementing it on the breadboard. This process taught us the importance of “Rule #9” in engineering, that is, when we implement our real circuits in hardware, we know why we are doing something and what we expect to achieve. When we built each subcircuit, we were consistently testing our inputs and outputs, and comparing our measured values to our ideal simulated values, to affirm we were completing each step before moving on to the next.

Additionally, through the idea of a circuit network being a sum of smaller subcircuits, we learned the importance of space optimization on our breadboard. As we added each subsequent subcircuit, we saw our overall space on the board lessen.

An improvement we would make to the lab would be a portion of the lab dedicated solely to the understanding, design, and implementation of the LM555 timer. We feel that a section devoted solely to the timer would help us understand its role, before we actually implement it into our overall circuit. Another aspect of the lab that we felt was somewhat lacking was the decoupling capacitor, which we were told “reduces noise”. The circuits that we were instructed to build had a plethora of decoupling capacitors across various nodes and power rails, and we think it would help to explain specifically what each one is doing. Although we have an electrical model for our motor and an understanding of where each parameter comes from, it would also be useful to explain the implications of each parameter to the physical properties of the motor. For example, we were told that we had a “good motor” based on the parameters we experimentally determined, but where is this correlation derived from?