Characterizing FTC-Legal Motors

Dynamometer Project

2017-2018

Samantha Hordyk
Madeline Nguyen
Dr. John Fraser

Table of Contents

Description of Project

We will be using various methods to measure electrical, mechanical, and electromechanical properties of commonly used legal motors in FTC. These parameters will be used in PID algorithms, to allow the robot to function better and reach the values desired in a reasonable amount of time. The goal of this project is to not only obtain this information, but also to describe the process we followed so that others in the FTC community know that they can do the same. Thus, others can understand these parameters and how they are used, and can then use the software to suit their needs.

Motor Parameters:

- Moment of inertia of the armatures (represented by J)
- Motor viscous friction constant (represented by b)
- Electromotive force constant (represented by Ke)
- Motor torque constant (represented by Kt)
- Electric resistance (represented by R)
- Electric inductance (represented by L)

Equations:

$$\tau = Kt \cdot i \tag{1}$$

$$e = K_e \cdot \frac{d\theta}{dt} \tag{2}$$

$$J \cdot \frac{d^2\theta}{dt^2} + b \cdot \frac{d\theta}{dt} = Ke \cdot i \tag{3}$$

$$(L \cdot \frac{di}{dt}) + R \cdot i = V - (Ke \cdot \frac{d\theta}{dt}) \tag{4}$$

where τ represents torque delivered by the motor shaft, *i* is motor input current, *e* is voltage across the motor terminals, and θ is the motor shaft angle in radians, so $\frac{d\theta}{dt}$ is the shaft angular velocity. We can refer to that as $\boldsymbol{\omega}$ below. We will also refer to $\frac{d^2\theta}{dt^2}$ as $\boldsymbol{\alpha}$.

If all quantities are represented in SI units, Ke and Kt are identical. We'll just refer to them as K. Interestingly, this means that K has units of $kg \cdot m^2/sec^2/A$ or $V \cdot sec$! It's a good exercise to prove those are the same.

What's a Dynamometer?

A dynamometer is a measuring instrument that can control a motor and measure its properties. In our case, we're only interested in DC electric motors with encoders, and a limited range of types, FTC legal motors. A basic electric motor dynamometer can apply a current to a motor and load the shaft to a run at a specific angular velocity with a brake, allowing measurement of the torque delivered and the voltage across the terminals. The torque measurement can be made by a lever arm and a force sensor, either by allowing the brake to rotate about the shaft axis, or by allowing the motor to do the same. Because we may want to run our brake through an external gear train, we choose to measure the torque at the motor side.

What is PID?

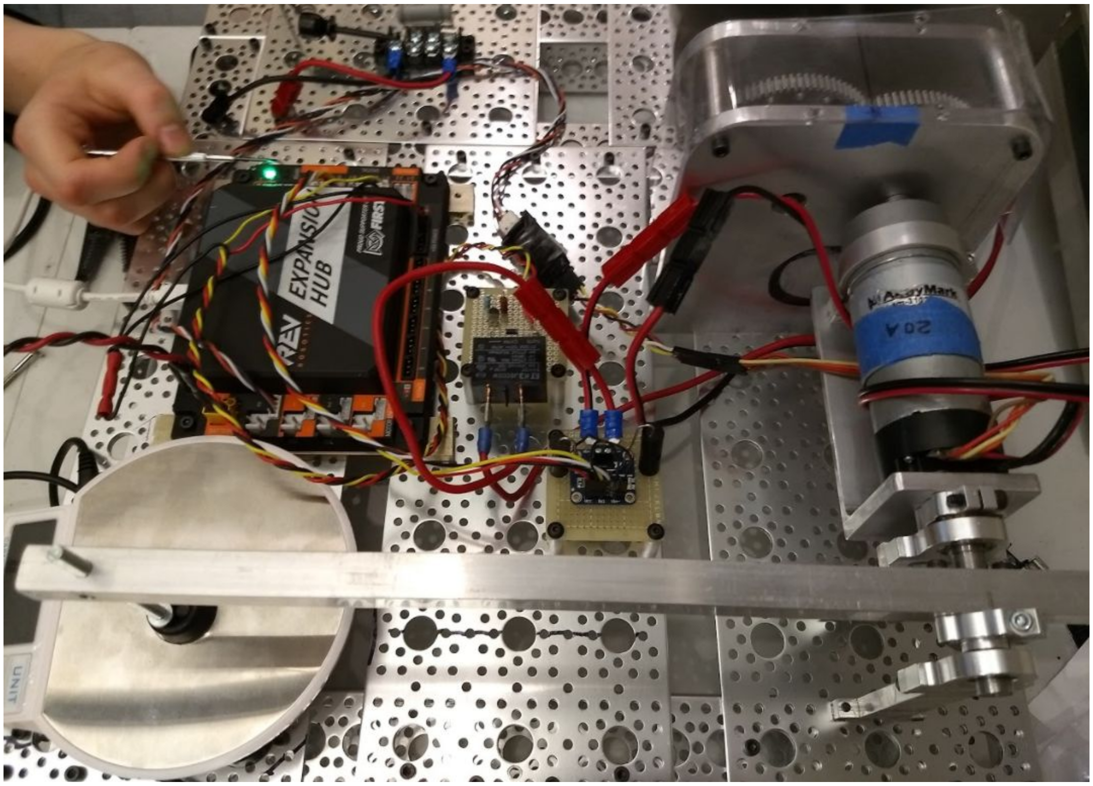
PID stands for proportional, integral, and derivative. It is a control loop feedback mechanism that applies a correction value based on an error value calculated by the three letters, or terms of PID. The *setpoint*, represented by r(t), is the desired value. The PID control loop takes the setpoint and compares it to *process variable*, or PV, which is value in reality. To put in context, if PV and r(t) are equal to each other, then the control doesn't have to do anything, and the control output value will be zero.

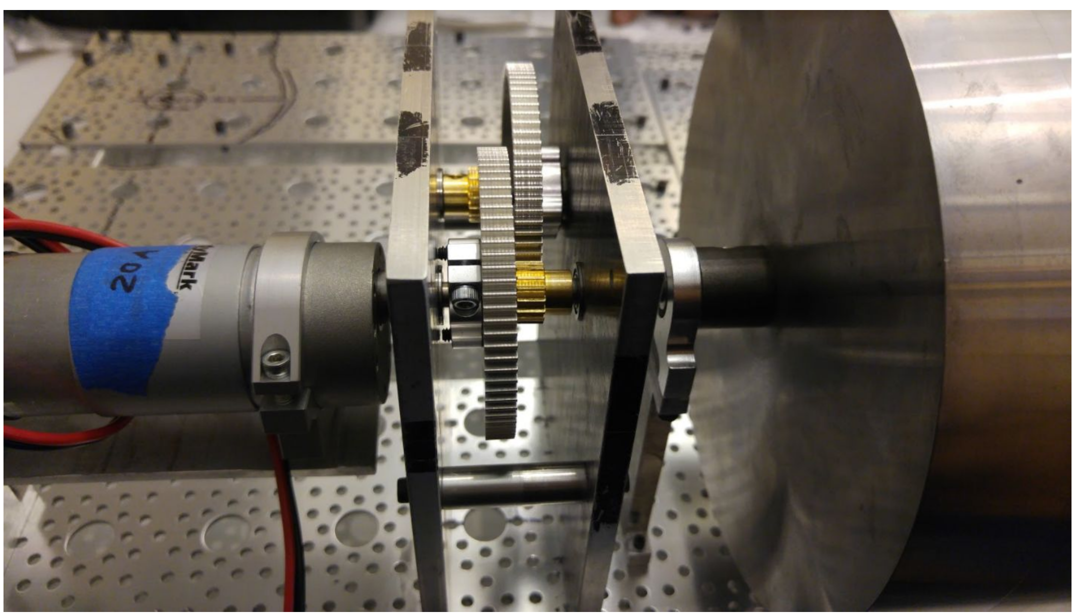
"P" for Proportional - This term calculates the *present* error. It allows the control output to move *proportionally* to the error. If the error is great, the control output will be great as well.
"I" for Integral - This term accounts for the *past* values in error.
"D" for Derivative - This term accounts for the *future* error based on the rate of change in the current error.

The entire PID control loop is a sum of all three of its terms, but each term can be used on its own. For example, it is common for a control loop to only have a proportional element.

Benefits to FIRST

Our Dynamometer





Machining Process

List of Machined Parts:

- Gearbox
  - Plates, spacers, washers
- Flywheel axle
- Standoffs
- Cradle
  - Lever arm
  - Cradle Motor spacer to fit hub

Gearbox:

- *Plates:* The gearbox we made we made up of two identical plates, spacers, and washers. This gearbox is a 25:1 gearbox, with the fastest shaft on the flywheel. The plates on the gearboxes were made on a manual milling machine and each of the four sides line up perfectly with each other. In addition, the holes for the shafts are placed so that the gears (from Actobotics) mesh with each other perfectly.
- *Washers:* The washers are placed between the gears and the plates of the gearbox to ensure the gears don't rub on the bearings.

Standoffs:

These standoffs were machined to make sure every part of the dynamometer was concentric. They were designed so that an Actobotics bearing mount would fit on top, and so that threaded holes on the standoffs match the spacing with those the bearing mount. This way, the bearing mounts can be secured onto the standoffs. These were manufactured using a milling machine.


Experiments

First Experiment:

Starting from equation (1), we can see that $\mathbf{K}$ is equal to $\frac{\tau}{I}$, so one experiment that is immediately suggested is to apply a constant current i to the motor electrical terminals and see what torque τ is delivered. This can be done with a dynamometer, but while we are building that, we can quickly set up a test with a known moment of inertia J attached to the motor shaft, driving the motor from a constant current power supply, and using the relation

$$\tau = J \cdot \boldsymbol{\alpha}. \tag{5}$$

We don't know the internal motor armature moment of inertia J_m, but we can attach a flywheel with a much larger and well known moment of inertia J_f. If the angular acceleration time is much longer that what of an unloaded motor, we know the flywheel moment of inertia is much larger than the motor moment of inertia.

We can calculate τ from a measured value of $\boldsymbol{\alpha}$. We can easily write a program to start the power by driving a relay connected between the power supply and the motor, and collect an array of encoder counts and their occurrence times. This data can be reduced to directly determine $\boldsymbol{\alpha}$, and thus τ from equation (5), and finally $\mathbf{K}$ from equation (1).

We think that we can collect encoder count data every 30 milliseconds or so, and want to get lots of data in the regime where the motor is running slowly so we can clearly resolve the acceleration with data of that granularity. Our worst case is the AndyMark 60, because it has the gear ratio and thus the highest torque and the highest encoder count rate relative to the output shaft angular velocity.

We chose to seek a moment of inertia that would force an AndyMark 60 to take 5 seconds to reach half of maximum angular velocity when driven at half the published stall torque.

We have fairly complete data from an AndyMark 40 motor, provided by AndyMark.
http://files.andymark.com/am-2964_testing.pdf

The no-load speed is quoted at 130 rpm, the stall current at 11.5 Amps, and the stall torque at 400 in · oz. Units, ugh!

130 rpm = 130 $\frac{rev}{min} \cdot \frac{1\,min}{60\,sec} \cdot \frac{2\pi\,rad}{rev}$, and

400 in · oz = 400 in $\cdot \left(\frac{1\,m}{39.32\,in}\right) \cdot \left(\frac{1\,lb}{16\,oz}\right) \cdot \left(0.454\text{kg} \cdot 9.81\,\frac{m}{sec^2} \cdot \frac{1}{1\,lb}\right)$

$$\omega_{max} = 13.6 \ \frac{rad}{sec}$$
$$i_{max} = 11.5 \text{ A}$$
$$\tau_{max} = 2.83 \ \frac{kg \cdot m^2}{sec^2} = 2.83 \text{ N} \cdot \text{m} \ \textbf{(AndyMark 40)}$$

For the AndyMark 60, with a gear ratio 1.5 times higher, angular velocity will be approximately 1.5 times lower, and torque will be approximately 1.5 times higher, since the different gear boxes will have different internal drags. At very low angular velocities, the internal drag should not be a factor, so stall torque should be exactly multiplied by the gear ratio.

$$\omega_{max} \cong 9.1 \ \frac{rad}{sec}$$
$$i_{max} = 11.5 \text{ A}$$
$$\tau_{max} = 4.2 \ \frac{kg \cdot m^2}{sec^2} = 4.2 \text{ N} \cdot \text{m} \ \textbf{(AndyMark 60)}$$

If we ask ourselves what moment of inertia J_f will cause an AndyMark 60 to take 5 seconds to reach $\omega_{max} \div 2 \cong 4.5 \frac{rad}{sec}$ in 5 seconds when driven by $i_{max} \div 2 = 5.75$ A and thus delivering $\tau_{max} \div 2 = 2.1 \ \frac{kg \cdot m^2}{sec^2} = 2.1 \text{ N} \cdot \text{m}$, from equation (5) we get

$$J_f = \tau \ / \boldsymbol{\alpha}$$

and since

$$\boldsymbol{\omega}(t) = \boldsymbol{\alpha} \cdot t$$

for constant angular acceleration from zero to 4.5 $\frac{rad}{sec}$ in 5 sec,

$$\boldsymbol{\alpha} = 4.5 \ \frac{rad}{sec} \div 5 \text{ sec} = 0.90 \ \frac{rad}{sec^2}$$

So

$$J_f = 2.1 \ \frac{kg \cdot m^2}{sec^2} \div 0.90 \ \frac{rad}{sec^2}$$

$$= 2.33 \text{ kg} \cdot \text{m}^2$$

It turns out that this is an impractically large moment of inertia to get in a fabricated object. If we think for a moment of our robots, this is not an unexpected result. A 15 kg robot can accelerate to full speed in about 3 seconds driven by four motors. From a kinetic energy standpoint, a moment of inertia that would have a kinetic energy comparable to a robot after being driven at full power for 5 seconds would have to be fairly large and heavy.

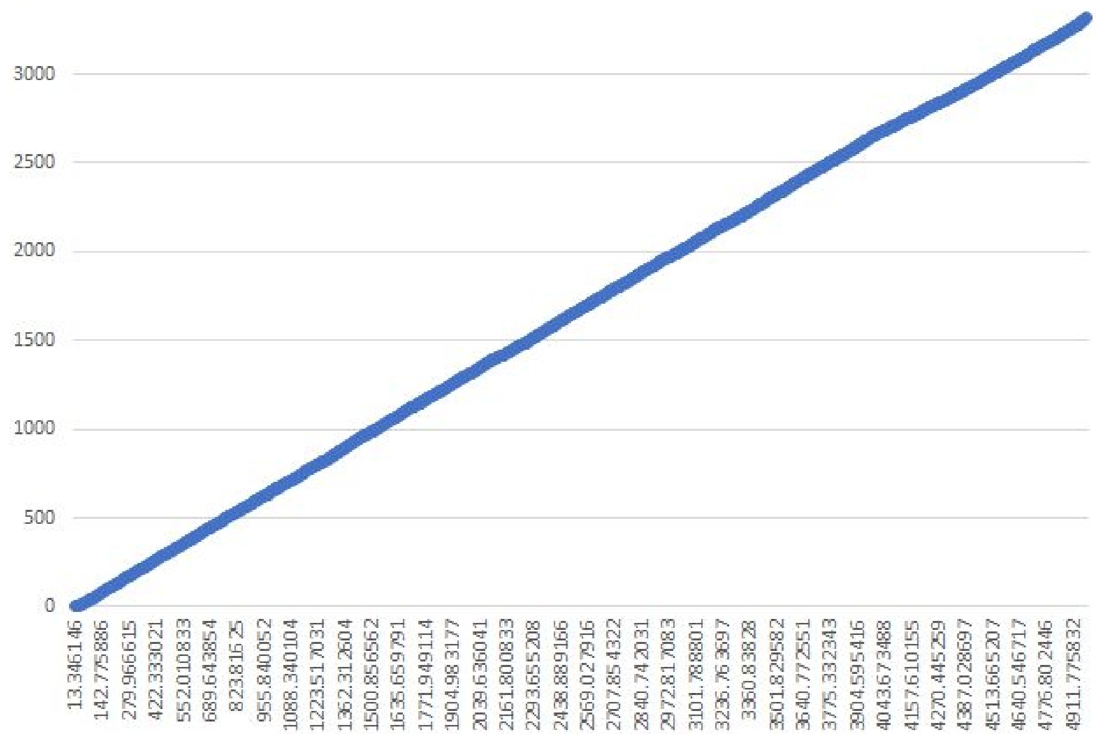
For a disc of uniform mass density rotated about its principal axis,

$$J = \tfrac{1}{2}mr^2 = \tfrac{1}{2}\pi r^4 \rho h,$$

Where m is the mass, r is the radius, ρ is the mass density, and h is the thickness. If we assume we use steel for the material, ρ is about 7850 $\frac{kg}{m^3}$, so a practical sized disc of r = 0.075 m and h = 0.05 m has a moment of inertia of 0.0124 $kg \cdot m^2$, 188 times less than desired.

Our First Motor Test Counting Encoder Ticks:

Our first test was conducted with a REV Core Hex motor. Simply running the motor at full power, (1.0) in code, for every encoder count incremented, we logged the runtime. The motor ran for 5000 milliseconds. The encoder count and the corresponding runtimes were recorded into one dimensional arrays called “motorPos[]” and “time[]”. These two arrays are parallel arrays because they share the same index, referenced as “numSamples” (number of samples). Concluding the test, the values inside the arrays are stored into a text file using the FileWriter class that prints each time, and motor position with a space in between. We found that this format is convenient to import into Excel and make graphs. Here is our first graph of the REV Core Hex motor.



The x axis displays the runtime in milliseconds, while the y axis displays the encoder counts. As shown on the x axis, the motor was running for 5000 milliseconds, but the time value is slightly smaller than 5000 because the last read encoder count was before 5000 milliseconds. The graph is mostly linear, as no moment of inertia was attached to the motor; it was simply hand-held. It is only at the beginning where it appears to be accelerating, and moves at a constant speed for the rest of the time running.

Our First Motor on a Gearbox Test:

We ran an AndyMark NeverRest 20 motor hooked up to the flywheel, and for the first time, the completed gearbox. We plugged the motor directly into a 12 volt battery and ran it for 40 seconds. The motor was simply handheld, as the cradle for the motor was not done at the time, and the battery was unplugged as soon as the motor became too warm to hold. After the motor was unplugged, the flywheel took 40 more seconds to come to a complete stop. At this point, we are unsure if the motor reached its full speed because the battery was unplugged when the motor became unsuitable to hold. This test was purely to test out our new gearbox to see if major changes were needed. At the end of the test, it is safe to conclude that the gearbox is in great condition, and we are ready to move onto the next step.

Ramp Motor Test:

After the previous test, we decided to ramp up the motor and record the final speed before slowing down to see if the motor was capable of reaching and staying at a power limit, since we had just finished our encoder cable and motor cable. This test was also done on the same AndyMark NeverRest 20 motor. The software is very simple and is set up so that while the current motor power is less than the set motor power, the motor power is increased by 1% (0.01 in the software) and waits 100 milliseconds before the comparison between motor power and set motor power occurs again. This process runs for 5000 milliseconds, giving time for the motor to ramp up to its full set power of 10% (0.1) in just 1000 milliseconds. After running this test, the motor took a while to get the flywheel moving, and we changed the set speed to 40% (0.4) instead of 10% (0.1). The motor had no problem accelerating to this speed, and the final speed displayed on our robot phone screen was 0.4000000000001, staying true to the set max speed in software.

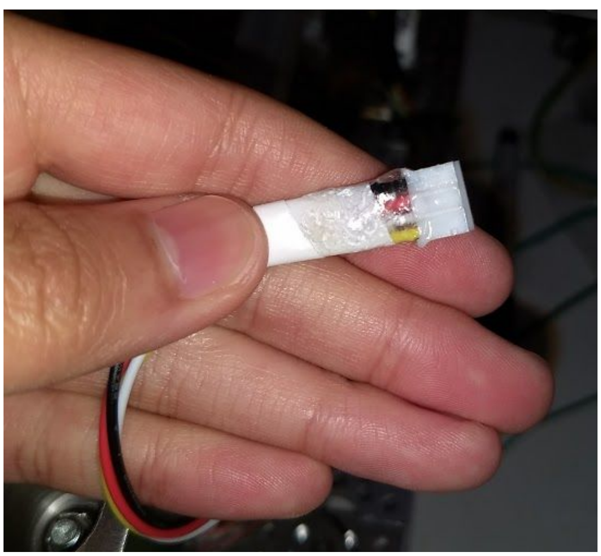
Reduced Drag Test:

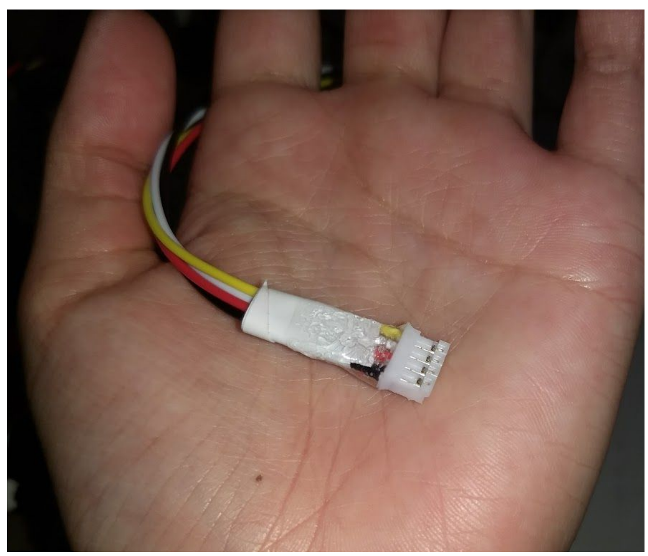
The purpose of this test was to see how much the maximum speed of the flywheel would change if light oil was applied to the gearbox. After putting light oil on the gears and washers, we plugged a battery directly into our AndyMark 20 motor and unplugged it after one minute and nine seconds. At this point, the motor was getting very warm. The maximum speed reached during this test was 1200 RPM but the motor was still accelerating. At four minutes and ten seconds the speed was 600 RPM. The flywheel came to a stop at six minutes and 26 seconds.

REV Hub Connection Problem

The type of connector that is used on the REV Hub is not exactly ideal because it provides a very fragile and unreliable connection. We have faced many problems in the initialization process because of these connectors, and it took up a while to debug. When we had narrowed the problem down to the connectors, we immediately purchased brand new cables and began to make cables guaranteed to keep a connection. In other words, we wouldn't put them in or pull them out or do anything else without protection.

Firstly, we gently wrapped a triangular piece of electrical tape around the wire, about 2.5 mm away from the start of the housing. We wrapped it around so that it formed layers that we could apply super glue onto, the thinnest layer of tape near the housing. We then generously piled up super glue around where the wire meets the housing, careful not to let too much drip and cure far inside the housing, and spread the rest of the glue thinly to the second to last electrical tape layer. Here are a few pictures of the finished product for our Adafruit INA219 high-side current and bus voltage sensor:

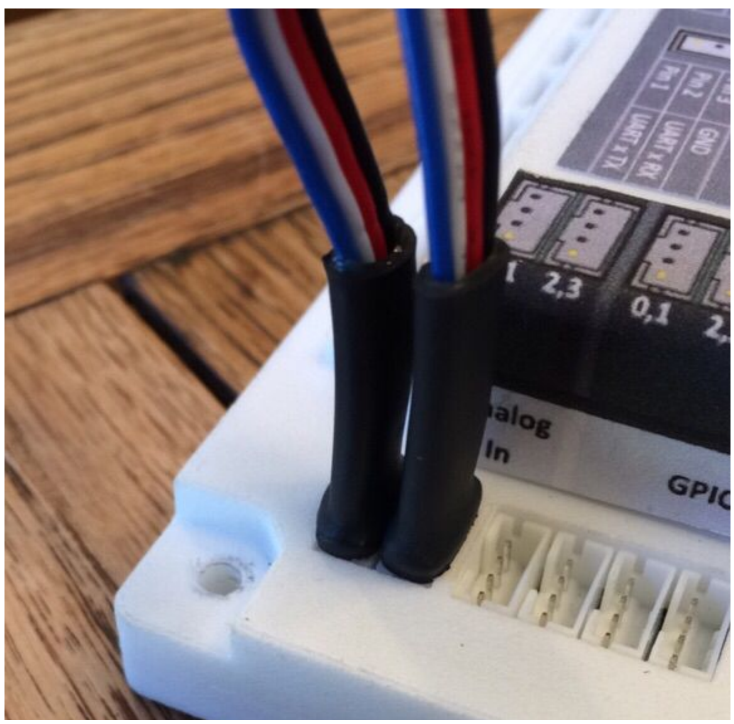




As you can see, we have provided a solid grip on the wire so it is easy to pull out without damaging. After we made this wire, we haven't used another one and we no longer have any problems initializing our OpModes. The reason why we had problems initializing before was because the INA219 sensor was never really fully connected to the hub. We often receive errors like "Problem with 'ina'". The other end has BLS connectors, but we never had a problem with that side, just the side connected to the Hub was problematic.

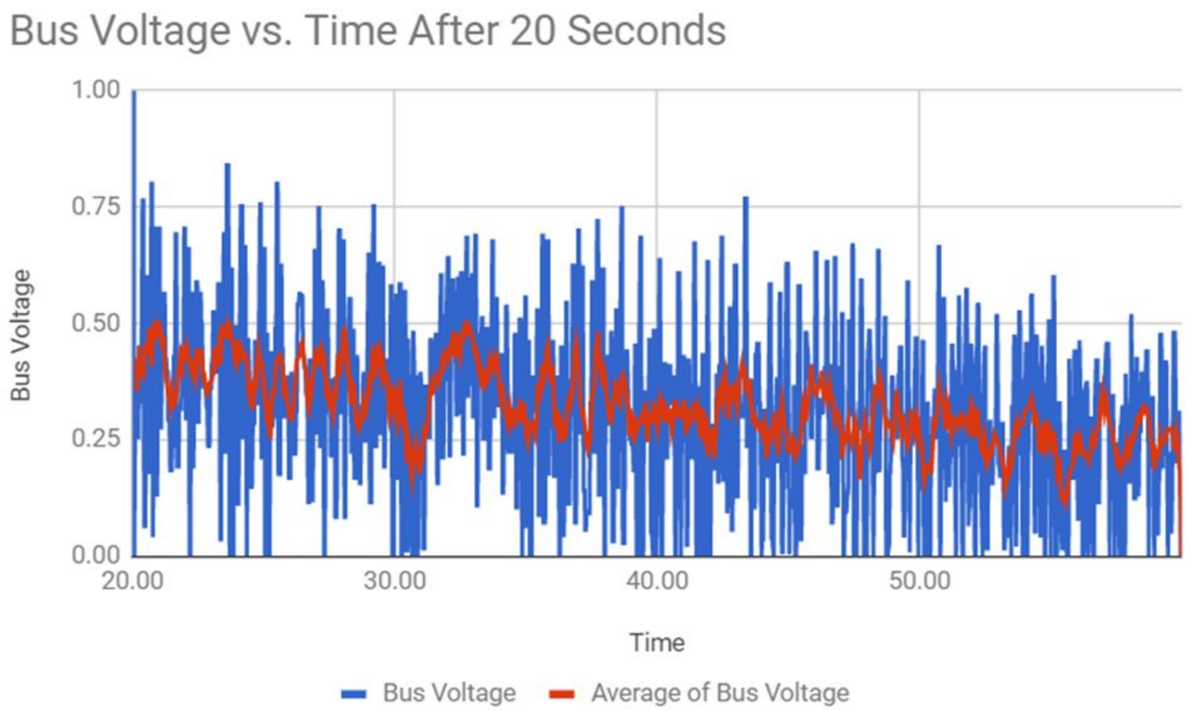
But although the electrical tape and superglue worked well, it was a rather time-consuming job just to make one cable. Luckily, Samantha Schatz, one of our club members, has come up with a very promising solution. She has found some heavy duty self-sealing heat shrink to put over the wires. Here are a few pictures:

The heat shrink wraps around the housing. It appears to seal to both the housing and the wires together and sticks to the housing well.



If you wanted to put two protected cables next to each other, it could still be done, although the fit is a bit tight.

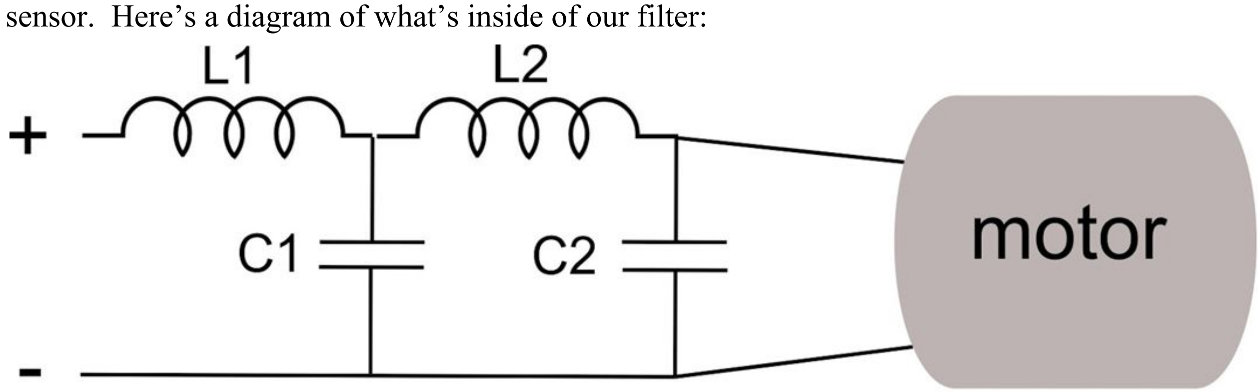
Noisy Bus Voltage Data and Filter



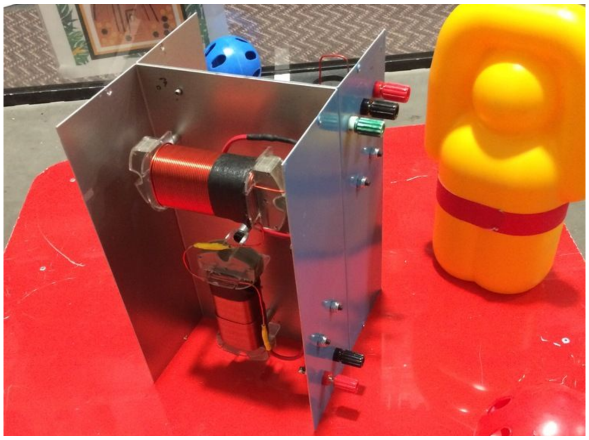
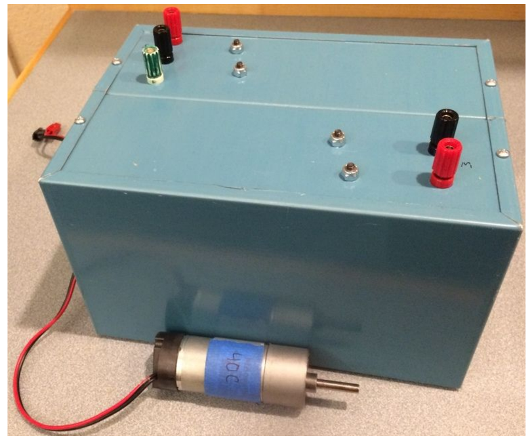
The graph above displays data from an experiment where we ran motor AM20B for 60 seconds, reading bus voltage after 20 seconds. The voltage was read from our INA219 sensor, but the bus voltage measurements were extremely noisy, as shown in the blue graph above. We then took the blue graph and did a 15 point average on it, which is displayed with the red graph. However, this data is still extremely noisy, so we made an electronic filter for bus voltage data.

Low Pass Filter:

The purpose of this filter was to keep the bus voltage noise low enough to measure, and smooth out the bus voltage for the power supply. The filter is located between the relay and the INA219 sensor. Here's a diagram of what's inside of our filter:



Where L1 = 33.5 mH, L2 = 23mH, C1 = 8.3 μF , C2 = 2 μF .

Experimental Notes:

Our Kt and Ke measurements don't quite match, although they are theoretically the same constant. Kt is calculated with the motor's torque measurements, which involves internal drag and internal gearboxes. The gearboxes are not completely efficient, thus the Ke measurement is much more accurate. However, we will be investigating this issue further during the summer of 2018.

Procedures

Flywheel and INA219 Experiment

In this experiment, encoder counts will be logged to measure the position, which the angular velocity and angular acceleration of the motor can be calculated from. In addition to logging encoder counts, an INA219 current sensor will be used to log bus voltage and shunt current, while a scale will be used to calculate the torque. Using this data, you can calculate the value of **K** (the electromotive force constant, and the motor torque constant).

NOTE: For all tests involving the INA219 current sensor, please do NOT set the voltage on the power supply to anything higher than 16 volts. We found that a high voltage will fry the sensor.

Set Up:

1. Make sure all bearings and the gearbox are well lubricated with a light oil.
2. Plug the robot phone into the REV Hub.
3. Connect the motor (through the filter) and the relay to the power supply. Make sure the motor encoder cable is plugged into the motor you're testing.
4. Set the power supply to a current limit of 4 amps and a voltage limit of 18 volts for an AndyMark 20 motor. For a AndyMark 60 motor use 2.5 amps and 18 volts. For an AndyMark 40 motor, use 4 amps, and 18 volts. For REV Core Hex motor, set the power supply to 2 amps and 16 volts. (but then turn the power supply off, because if it is on it interferes with the program).
5. Select the "DynamometerTests" OpMode on the Driver Station phone and initialize the OpMode. Once initialized, you will see a message at the bottom of the Driver Station phone screen reading "INA done initializing".
6. Set up an external timer.
7. Make sure the guard cable is plugged into the negative port of the power supply and the other side is attached to the ground bar.
8. Turn on the power supply.
9. Tier the scale so that while the arm is resting on it, it reads zero.

18 volts
Spin up 90 seconds, spin down and record for 90 seconds take data every 100 milliseconds

During Experiment:

1. Start the OpMode by pressing play, and activate the external timer at the same time.
2. Record the scale readings at 1 second, 10 seconds, 20 seconds, and 29 seconds.

3. As the timer approaches 30 seconds, keep an eye on the bottom of the driver station phone screen for any error messages concerning the INA219 current sensor.
    a. *The message should display as "problem with 'ina'". If this should happen, make sure that the power supply is set to 4 amps and 18 volts. Then, restart the test.*
4. Assuming that the INA219 current sensor does not throw any errors, continue the test by waiting until the timer reaches 1 minute. By then the OpMode should automatically stop because of a timer inside of the OpMode. The robot controller phone may be safely unplugged from the REV hub at this point.

Retrieving the Data from the Phones:

1. After unplugging the robot controller phone from the REV hub, plug the phone into a computer.
2. On the robot controller phone, pull the top menu bar down and select "USB for Charging". If this option is not visible then make sure the robot controller phone is plugged into the computer.
3. If the radio button is on "Charging only", select "Transfer Files (MTP)" instead.
4. On your computer, open File Explorer and select "This PC", "MotoG3", and "Internal Storage". Open "testFile.txt" and testFile2.txt".
5. Delete the empty lines on the top of both text documents. On "testFile.txt", Delete all placeholders after the 30 seconds. Copy all of the "testFile2.txt" and paste it to the end of "testFile.txt". Save the file with the motor type, voltage, and current, adding "Log" (to the end to indicate it is a text document) as the title and put the file into a folder with the date that the test took place. Here is an example of a file title: "AM20A5amps20voltsLog".
6. Insert the data into a Google Spreadsheet by selecting an empty cell and selecting the "File" menu. From the "File" menu, click on "Import" and this will open a mini window titled "Import file". Select the "Upload" option and select the file from your computer.
    a. Under "Import Action", select the "Replace data starting at the selected cell" radio button. For the separator character, select "Custom" and type a single space " " into the textbox. Select the blue "Import" button at the bottom of the screen.
    b. On a blank Excel Spreadsheet, leave about 6 empty rows at the top. Starting at Column A, write out the four data arrays received from the log like so:

5				
6	**Time**	**Encoder Counts**	**Bus Voltage**	**Shunt Current**
7	12.592448	0	0.96	-0.001

Data Analysis:

1. You should have four columns of data: Time, Encoder Counts, Bus Voltage, and Shunt Current. First, convert the time (in milliseconds) to seconds.
2. Take the counts per revolution of whichever motor is being tested (Ex. 560 for AndyMark20) and divide it by 2π. This will give you the counts per radian of that motor.
3. To calculate the position, subtract the encoder counts from the previous data point, then divide by counts per radian.

AM20 counts/rad=		89.13					
Time	Encoder Counts	Bus Voltage	Shunt Current		Time (sec)	Position (radians)	Fitted Position
11.6312	0	0.024	-0.001		0.011631	=(C8-$C$8)/$D$4	

4. Use the method Solver in Microsoft Excel to fit an equation to the position data, but only to the data taken after the first thirty seconds (or after the power supply is shut off) [see step 5 to understand why]. The angular velocity will be decaying at an exponential rate, so the equation to use is P(t)=P(∞)-B*e^(-γ*(t-30)).
   a. Estimate values for P(∞), B, and γ. It does not matter if your estimates are incorrect, because Solver will change the estimate to a different value.
   b. Apply the equation referred to above (including estimated values) as a formula to create a column of calculated position data.

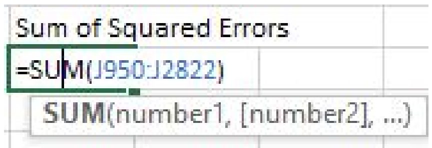
P(t)=P(∞)-A*e^(-G*(t-30))			
P(∞)	A	G	Sum of Squared Errors
312.415088	279.919	0.008698	664.6275

Time (sec)	Position (radians)	Fitted Position	Squared Errors	Derived Angular Velocity	Fitted Bus Voltage	Squared Errors (voltage)
30.26107	34.37675	=$O$951-$P$951*EXP(-$Q$951*(G955-$G$949))				

   c. Create another column for the squared error of the calculated position data. This is done by creating a formula that subtracts a calculated position from an actual position, and then squares it. Apply the formula to all the calculated position and actual position data.

Position (radians)	Fitted Position	Squared Errors
34.37675	32.97993	=(H955-I955)^2

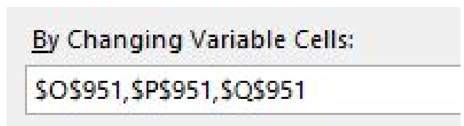
   d. Using the SUM function, add up all of the squared error data.



e. Then, use the cell that contains the value of all the squared error data added together as the "Set Objective" in the Solver parameters.

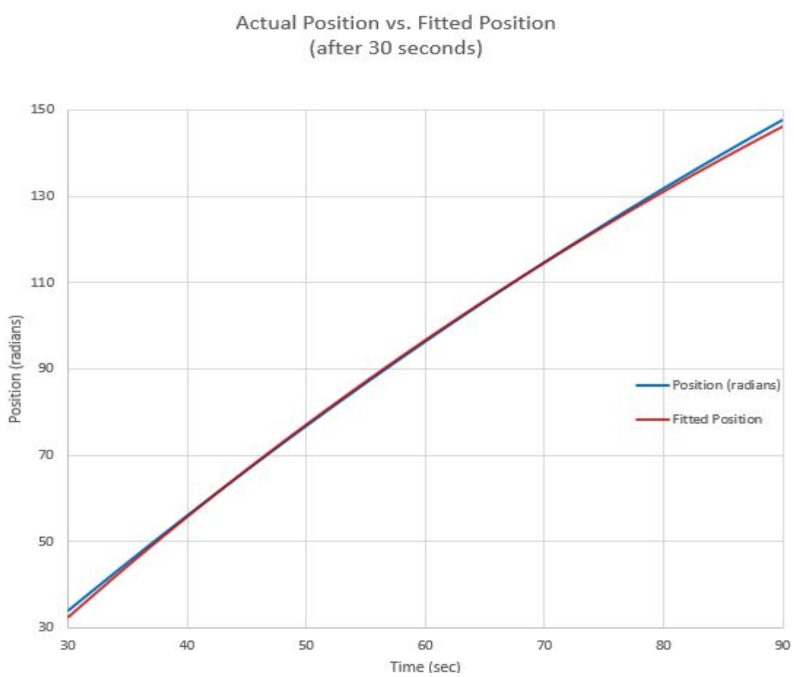
f. Set Solver to minimize the "Set Objective".

g. Select the cells that contain the estimated values for P(∞), B, and γ and use them as the variable cells in Solver.



h. Use the GRG Nonlinear solving method.

i. Then solve. It is best to do this twice, just to make sure you have the correct values. The sum of squared errors is likely to still be high, as there are three variables. (Hint: To verify that you did not make any errors, graph the calculated position data alongside the actual position data.) Here is an example:



5. Then, in order to calculate the **K** value, you will need to have velocity data which will be created by taking the derivative of the position equation you calculate.
    a. After you have velocity data, you will use the equation V=Kω to calculate K. Because this equation is only true when there is no current, you must use position data from only ***after*** the first thirty seconds, or after the power supply is turned off.
6. Take the derivative of the equation you created (ω(t)=B*γ*e^(-γ(t-30))) and apply it as a formula to create a “Derived Angular Velocity” column.

Derived Angular Velocity						
=$P$951*$Q$951*EXP(-$Q$951*(G950-$G$949))					B	γ
2.433321	0.852838	0.727333		312.415088	279.919	0.008698

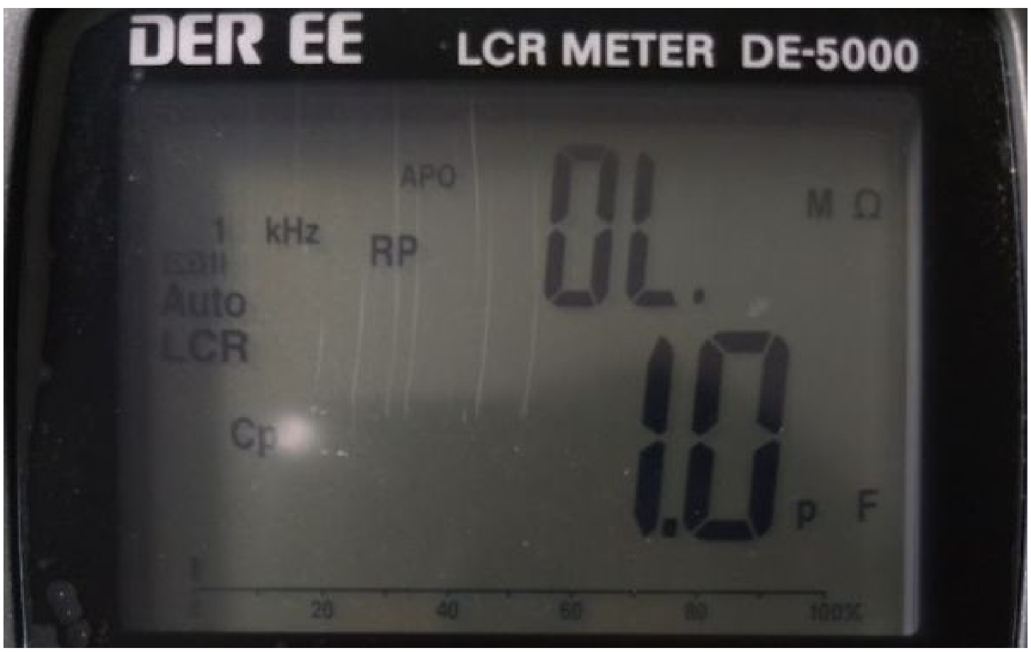
7. Then, using Solver again, create another equation for calculated bus voltage. This will be the linear equation V = K · ω, with **K** being the slope.
    a. Estimate the value for **K**.
    b. Apply the formula referred to above, including the estimated value of **K**, to the bus voltage data.
    c. Create a new column of calculated bus voltage data.
    d. Create a new column of squared error, this time subtracting a calculated bus voltage from an actual bus voltage and squaring it.
    e. Use the SUM function again and add all of the squared errors.
    f. Select the cell that contains all of the squared errors as the “Set Objective” in Solver.
    g. Set Solver to minimize the “Set Objective”.
    h. Select the cell containing the estimated value of **K** as the variable cell in Solver.
    i. Use the GRG Nonlinear solving method, and solve. The number generated for **K** is the value of the electromotive force constant and motor torque constant.

LCR Meter Experiment

Using a DER EE LCR Meter DE-5000, while the motor is NOT connected to the flywheel or the gearbox, this experiment directly measures **L** (the inductance of the motor) at different positions, and **Q**, a quality factor. It is important to measure the inductance at different motor shaft positions because it changes depending on the commutator. This will allow you to calculate **R** (the resistance of the motor winding).

Set Up:

1. Unplug the motor from the filter and put the two Anderson PowerPole plugs into a vice. Place the plugs upright and clamp the plugs gently.
2. Plug the guard cable into the LCR meter, with the other side connected to the ground bar.
3. Put the LCR meter into Auto LCR Mode and check to make sure that the meter is reading at 1 kiloHertz with Q and microHenrys. Q and microHenrys will only be displayed when the meter is connected to the motor. When the meter is not attached to the motor, these units will not display. You will receive the following image instead.



4. Put the two probes from the LCR meter into the motor's plugs and record the **Q** and **L** values into a spreadsheet, with **L** in one column and **Q** in another.
5. Turn the shaft to a different position. If the shaft is not connected to the flywheel, we would recommend that you use pliers or attach a coupler to the shaft for easier turning.
6. Repeat Steps 4-5 nine more times for ten data points.
7. Put the two probes from the LCR meter into the motor's plugs and record the DC resistance in another column of the spreadsheet.
8. Repeat steps 7 and 5 nine more times for ten data points.

Data Analysis:

1. With the **Q** and **L** values recorded into a spreadsheet like shown below, calculate the resistance **R** by multiplying 2π by **L** inductance and 1000 to convert kHz and divide by **Q** and 1000000 to convert microHenrys to Ohms. Your formula for each resistance cell should look similar to the example that follows (with F7 as the parallel inductance index and G7 as the parallel **Q** index: `=2*PI()*1000*(F7/G7)/1000000`.
2. Average the inductance and resistance values. Put the averaged values below the original ten measurements. The winding resistance in the example pictured below is a value of 1.9 ohms.

Inductance (L)	**Q**	**Resistance**	R = (2πFL)/Q		**DC Resistance (**DCR mode)
uH		Ohms			
697.2	1.03	4.253			3.65
643	2	2.020			1.46
645.2	1.38	2.938			2.35
731.4	1.94	2.369			1.75
732.8	2.17	2.122			1.49
646.4	1.92	2.115			1.57
695.9	1.77	2.470			1.83
645.6	1.79	2.266			1.69
656.4	1.7	2.426			1.81
641.2	2.08	1.937			1.38
674		**2.5**			
			(measured at 1 kHz with motor not turning, and at random positions)		

Free Motor Spin-Up Experiment

To measure the voltage with more accuracy, we used the Rigol DS1054Z 4 Channel Oscilloscope. Here is the link to the user's manual: https://telonicinstruments.co.uk/rigol-uk/MSO-DS/MSO-DS1000Z%20User%20Guide.pdf. During this test, no flywheel is attached to the motor. The motor under test is simply spun alone while an oscilloscope measures times of transitions (voltage vs. time) in one of the encoder channels. The **J** variable (motor's moment of inertia without the gearbox attached) is calculated from this experiment.

Set Up:

1. Attach the channel 2 probe onto the signal side of the filter, on the red and black banana jack. Attach channel 1 probe to motor encoder port 1, on ground and Channel B.
2. Turn the oscilloscope on. This will take a moment.
    a. Set channel 1 to DC coupled, using a 1x probe with a 20M Bandwidth limit.
    b. Set the horizontal scale to 200 milliseconds per division and the vertical scale to 500 millivolts per division. For an AM3.7 motor, set the horizontal scale to 20 milliseconds per division. Channel 1's vertical offset should be -1.65 volts. The measurements should be made on a vertical cursor B, and set one small division off from the center of the screen, and turn the intensity of the trace up to around 86% for a better picture.
    c. Set the horizontal position to the left.
    d. Set the trigger on channel 1 to 1.4 volts.
    e. Switch the mode to "Single" mode.
    f. For an AM3.7 motor, set the cursor mode to "manual" mode.
3. Turn the power supply on and set it to 0.5 amps and 18 volts. For a Matrix motor, set the power supply to 0.35 amps and 23 volts. Turn the power supply off.
4. Turn the REV expansion Hub on. (Plug the battery into the REV hub so that it can read encoder counts.)
5. Put the clip leads onto the relay.
6. Turn the power supply on.
7. Because this current is too low to start the motor, give the motor a quick turn by hand.
8. Let the motor run so that you can see a signal on the oscilloscope.
9. Turn the power supply off.

Collecting the Data:

1. If the first transition goes down, start counting cycles at the first up transition. If the first transition goes up, start counting cycles at the second up transition.
2. Select cursor B and move it to the first counting transition.

3. For each up transition, find the time in milliseconds by lining up your cursor with the line of transition. Record the time displayed on the top right corner of the screen for each up transition.
4. Repeat Step 2 nineteen more times.

Data Analysis:

The times we recorded in this experiment represent every fourth encoder count. So, when we assign positions to the encoder count values, we decrease the counts per radian by a factor of four. The parameter we want to calculate form this test is the motor armature moment of inertia, called **J** in our equations. The relevant equation is $\tau = J \cdot \alpha$, where we know the torque τ is equal to $\mathbf{K} \cdot I$ where I is the current and is controlled during the experiment, and **K** has already been determined.

We calculate positions in radians for each of the first twenty encoder ticks, and then use our usual least squares fitting technique with the Excel Solver nonlinear option to fit the following equation to the position and time data:

$$P = P_0 + \frac{K \cdot I \cdot (t - t_0)^2}{J}$$

Where P_0 is an arbitrary position offset and t_0 is an arbitrary time offset. **J** is the desired moment of inertia.

Terminal Velocity Experiment

This experiment involves running the motor until it reaches its highest speed without the flywheel attached. With the Rigol DS1054 Z 4 Channel Oscilloscope used in the previous test "Free Motor Spin-Up Experiment", you can measure the frequency of the motor encoder signal. The frequency will give the velocity (encoder counts per time divided by 4). A DC power supply (we used TEKPower's TP3005T DC Regulated Power Supply) will also be used to measure the voltage supplied to the motor. Both voltage and frequency will be measured and recorded when both seem to have stopped increasing. We will calculate variable **b**, the motor viscous friction constant from this test.

Set Up for AndyMark Neverest 20 Motor:

1. Turn on the power supply. Set the DC power supply to 0.4 amps and 18 volts. Turn the power supply off.
2. Make sure the motor is cool to the touch and switch the oscilloscope from single mode to normal mode. Also make sure that the encoder cable is plugged in.
3. Short the relay.
4. Turn the power supply on.
5. Let the motor spin up for a while and lower the current to 0.25 amps. If this current doesn't work, change it to a higher value and record this new value.
6. Put the oscilloscope into the same settings as the Free Motor Spin Up Experiment.
7. Let the motor spin up while watching the current on the power supply and the frequency on the oscilloscope until both of these values appear to have reached a maximum value, and are steady at these values.
8. Record the frequency from the oscilloscope and current from the power supply.

Set Up for AndyMark Neverest 60 Motor:

1. Connect the power supply to the relay, then the relay to the AndyMark Neverest 3.7 motor. Make sure that the AndyMark Neverest 3.7 motor is attached to the flywheel, and the motor under test is attached to the gearbox as well.
2. Attach channel 1 of the oscilloscope to motor encoder port one, on the ground side and on the clock signal. Put the oscilloscope in Auto mode, so that the "RUN/STOP" button is green.
3. Turn on the power supply. Set the power supply to 18.00 volts and 0.8 amps (this value can vary, but always record this value). Then turn the power supply off.
4. Turn the scale on. Then tare the scale with the arm on. Record the initial value in kilograms.
5. Short the relay using a clip lead. Then turn the power supply on, and help the flywheel start moving as needed; the starting current may not be sufficient to get the flywheel moving.

6. Once the flywheel has reached a stable velocity, and the frequency (on the oscilloscope) and scale reading are stable as well, record the following values: current (from the power supply), the scale reading, and the frequency (from the oscilloscope). These numbers will be used to calculate the variable **b** later in data analysis.
7. Turn the power supply off, and take the clip leads off of the relay.

Set Up for AndyMark Neverest 3.7 Motor:

1. Have a running time of 5 minutes at 18 volts, then rest to cool before running the actual test. This gets the motor grease out of the way.
2. Run the motor at 18 volts and 0.34 amps for 30 seconds.
3. Lower to 0.24 and run for 30 seconds

Data Analysis:

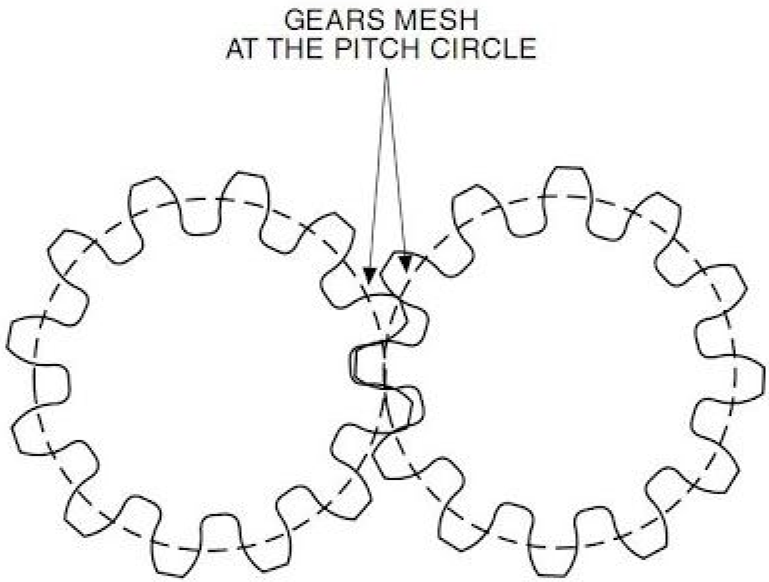
1. We cannot measure torque because it is such a low value and very inaccurate. Thus, we use the equation $\omega = \frac{Ke \cdot J}{b}$ **,** so the terminal angular velocity is equal to the **K** constant, which we have already calculated, times the recorded current, divided by the motor viscous friction constant.
2. Multiply the recorded frequency value (in Hertz) by 4 to convert it from cycles per second to encoder counts per second. Dividing this number by counts per revolution (which for an AndyMark Neverest 60 Motor can be found on this link: https://www.andymark.com/NeveRest-60-Gearmotor-p/am-3103.htm), converts the number into revolutions per second. Then multiply this value by 2π to convert to radians per second. This gives you the value of ω.
3. To calculate **b**, multiply *Ke* and *J*, and then divide by ω.

Final Calculated Parameters

Motors	**J** (kgm^2)	**b** ($\frac{kg\,m^2}{sec}$)	**Ke** (spin down)	**R** (Ohm)	**L** (uH)
AM 20 A	9.01×10^{-6}	0.0022	0.351	2.3	691
AM 20 B	9.01×10^{-6}	0.0025	0.389	1.9	684
AM 20 C	8.93×10^{-6}	0.0028	0.385	5.1	717
AM 40 A	2.22×10^{-5}	0.2269	0.753	2.5	674
AM 40 B	1.74×10^{-5}	0.5600	0.705	3.8	705
AM 40 C	2.47×10^{-5}	0.0180	0.763	2.1	716
AM 60 A	1.04×10^{-5}	0.0330	1.066	3.3	694
AM 60 B	8.42×10^{-6}	0.0200	1.076	5.1	696
AM 3.7 A	2.79×10^{-5}	0.00014	0.099	8.9	679
AM 3.7 B	3.15×10^{-5}	0.000176	0.108	2.6	797
AM 3.7 C	3.09×10^{-5}	0.00017	0.105	8.7	880
Matrix A	9.43×10^{-6}	0.00151	0.340	3.8	718
Matrix B	7.76×10^{-6}	0.00191	0.363	7.8	777
Matrix C	7.23×10^{-6}	0.00186	0.338	20.6	658
CoreHex A	7.33×10^{-4}	0.0112	0.822	3.6	1356
CoreHex B	6.55×10^{-4}	0.0080	0.858	11.3	1352
CoreHex C	4.54×10^{-4}	0.0078	0.711	5.6	1342

Data Sheet for LN324N: http://www.ti.com/lit/ds/snosc16d/snosc16d.pdf

Spur Gear Mesh:
http://lcamtuf.coredump.cx/gcnc/38-meshing.gif


GEARS MESH
AT THE PITCH CIRCLE