

# ME185 Lab 2

TA: Ben Stabler <stabler@stanford.edu>

## 2.0.0 Summary

The goal of the second lab is to construct an analytical model for the electric go-kart you drove last week. The model should be flexible so that you can use it to predict the capabilities of a different electric vehicle. For example, within the lab you will adapt the model to quantify the change in performance you expect to observe with a new battery configuration, and you will be expected to build a similar model to predict the characteristics of your project and justify your choice of powertrain. An accurate model will help you optimize your choice of battery, motor and motor controller, which will help you get the best performance for your budget!

- You will work in assigned groups of 2 (and one group of 3).
- Your group will need at least one copy of MATLAB. If you don't have MATLAB, you might want to consider getting the student edition, but for the time being see the TA and he will show you how to get remote access to MATLAB on one of the on-campus clusters.
- Download the starter files from the ME185 website
- Your write up should contain:
  - A brief answer to each question
  - Graphs (when appropriate)
  - Your MATLAB code
- Please submit your results in a zip file containing code directory and your write-up as a PDF by emailing them to the TA!
- All of the members of your group should have a copy of the data they collected last week in the first lab. When you get to section 2.5.0 you should parse each data set and then use the cleanest set for the remainder of the lab.

## 2.0.1 The Go-Kart Powertrain

- Battery configuration
  - 16x GBS-LFMP 40Ah cells
  - [http://www.electricmotorsport.com/store/ems\\_ev\\_parts\\_batteries\\_lpf\\_gbs\\_kit48.php](http://www.electricmotorsport.com/store/ems_ev_parts_batteries_lpf_gbs_kit48.php)
  - [http://elitepowersolutions.com/products/product\\_info.php?products\\_id=91](http://elitepowersolutions.com/products/product_info.php?products_id=91)
- Motor
  - Motenergy ME0907
  - 3 Phase, Brushless, Axial Flux

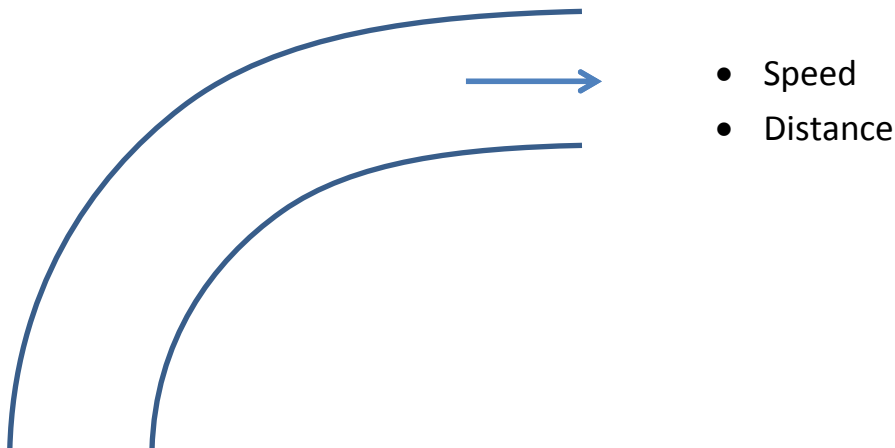
- [http://www.electricmotorsport.com/store/ems\\_ev\\_parts\\_motors\\_pmac.php](http://www.electricmotorsport.com/store/ems_ev_parts_motors_pmac.php)
- Motor Controller
  - Sevcon Gen4
  - [http://www.electricmotorsport.com/store/ems\\_ev\\_parts\\_controllers\\_sevcon\\_gen4\\_72-80\\_350.php](http://www.electricmotorsport.com/store/ems_ev_parts_controllers_sevcon_gen4_72-80_350.php)
  - 250A fuse
  - Maximum battery current limited in software to approximately 150A
- Drivetrain
  - Motor sprocket: 12 teeth
  - Drive axel sprocket: 40 teeth
  - Drive wheel diameter: 11 inches

## 2.1.0 The Model

The goal of the model is to inform you about how an EV powertrain will perform under real world conditions. The type of electric vehicle you are trying to characterize will determine the sort of conditions you should model. For the purpose of this lab we are modeling an electric go kart, so we describe a simple course consisting of *straights* and *corners*.

### 2.1.1 Corner

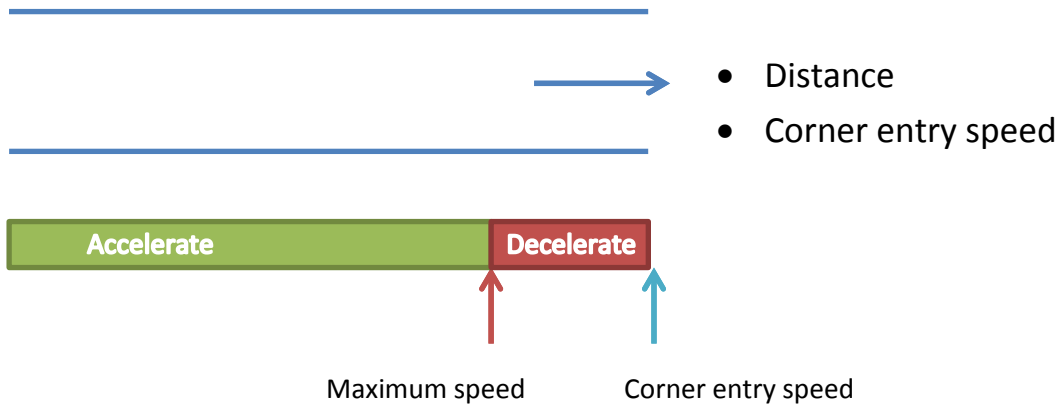
In our model we characterize the corner with a speed and a distance:



When the vehicle enters the corner we expect it to be travelling at a set velocity. We expect it to hold that velocity through the corner. In the context of a go-kart, we assume the *corner velocity* is the maximum velocity the go-kart can hold without sliding. The go-kart cannot accelerate in the corner without leaving the track, so the only power delivered by the motor during the corner is the power required to overcome losses, which are mostly air resistance, rolling resistance and cornering resistance. In the simple model we only consider air resistance.

## 2.1.2 Straight

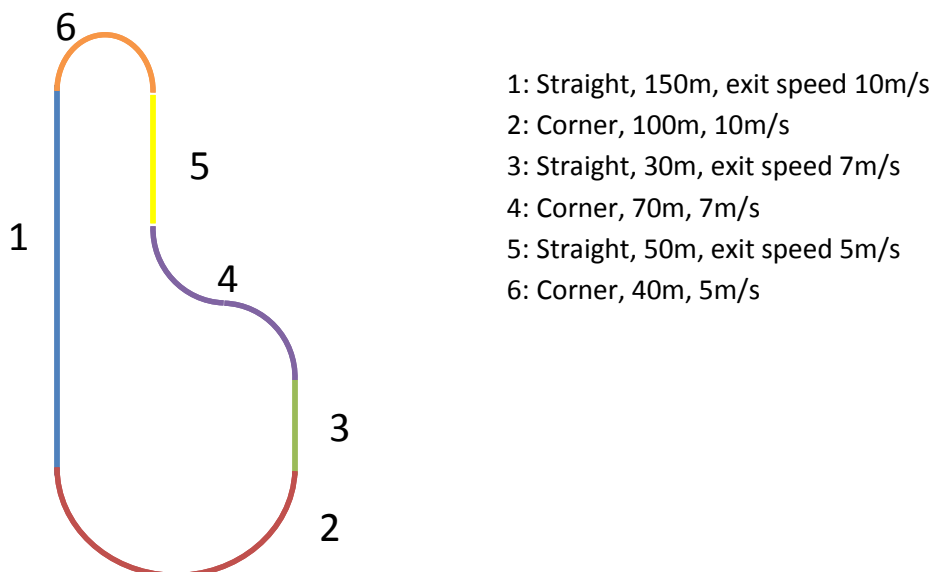
In our model a straight is characterized by distance and corner entry speed:



Within a straight we expect the vehicle to accelerate and pick up speed as fast as possible, and then brake as rapidly as possible so that the vehicle exits the straight (and enters the subsequent corner) at a given velocity. The acceleration and top speed of the vehicle are determined by the vehicle powertrain, mass, driving resistance, and friction between the tires and the road. The deceleration is determined mostly by the friction between the tires and the road. The vehicle must be travelling at or below the corner entry speed when it leaves the straight, or else it would leave the road.

## 2.1.3 Straights and Corners

The *straight* and *corner* primitives are combined to form a track:



Using this representation your model does not need to know the geometry of the track, just a few specific parameters for each segment. In your simulation the vehicle will travel around the track and at each point the 'driver' model will command the vehicle to accelerate or decelerate as appropriate, given the current speed, current segment and the vehicle's position within that segment.

### 2.1.4 Extensions to the model

The model is a simple representation of the type of course an electric go-kart might be expected to run. If you are designing a different type of electric vehicle, for example an electric skateboard, your use case will be very different and you may want to modify the model accordingly. For example you may want to introduce slope- a climb or fall would have negligible impact on a go-kart on a short course but it would have a large impact on a low speed light commuter vehicle!

### 2.2.0 Some EV powertrain concepts

There are three main components in the EV powertrain:



#### Battery Pack

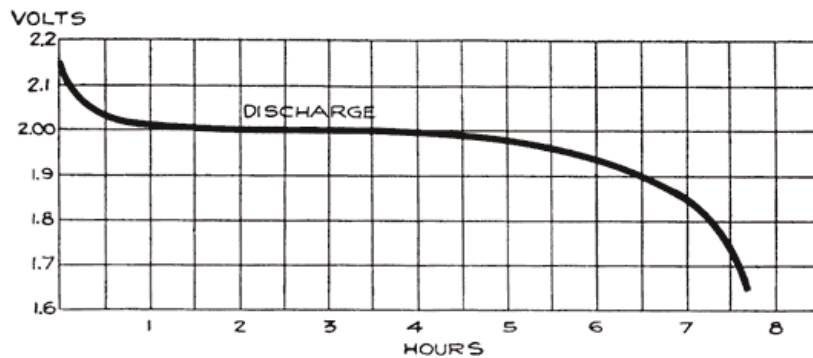
The purpose of the battery pack is to supply current at a fixed voltage. You will recall from your physics classes that power is the product of current through a component and voltage across that component ( $P = IV$ ), and you can calculate the power dissipated in a resistive element using the equation  $P = I^2R$ .

The ideal battery pack would have a high voltage, because high voltage means low current for the same amount of power, and low current means less power wasted as heat in the copper cables. Too high a voltage, however, results in increased stress on the powertrain components and can be dangerous. Most commercial electric vehicles have a battery voltage around 400V, and most vehicles built by hobbyists operate at or below 72V.

You may recall from chemistry that the voltage of a battery cell is an intrinsic property of the materials used to construct the cell. Lithium batteries typically have cell voltages around 3.3V, and a standard 12V car battery actually contains 6 individual cells at 2.1V each. The battery pack in an electric vehicle is constructed with many cells in series. On a lithium system, a Battery Management System (BMS) is typically employed to monitor each of the individual cells and ensure that they are within operating parameters. This isn't necessary on a lead acid system because lead acid batteries handle over- and under-charge situations gracefully (they don't catch on fire).

The battery cell voltage will vary as the battery is used. At full charge, for example, a 12V lead acid battery will sit close to 14.4V. As the battery is discharged it will fall to 12V and remain there for the majority of the cycle, and as reaches full discharge it will fall to 9V or below. The following figure from a

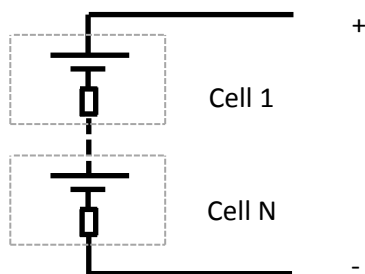
1922 text titled “The Automobile Storage Battery- It’s Care and Repair” is a battery curve for an individual cell within a 12V battery:



By measuring the voltage across the battery it is possible to infer the state of charge of the battery- however the flat region of the curve can make it difficult to estimate the state of charge accurately.

In this example the battery cell is being discharged at a rate of around  $\frac{1}{7.5}C$ . The C-Rate, or Hourly Rate, is a standard way to describe the rate of charge or discharge of a battery. It is measured relative to the total capacity of the battery. For example, a 40Ah battery being charged at a rate of 1C sees a constant current of 40A and will be fully charged in 1 hour. A 40Ah battery being charged at a rate of 2C will see 80A and will be fully charged in half an hour. When you are shopping for batteries you will find that they often specify a ‘charge rate’, a ‘continuous discharge rate’ and a ‘pulsed discharge rate’, which mandate the amount of current that can flow into or out of the battery in each of those conditions.

The ideal battery pack would be able to supply an infinite amount of current. In practice this is not the case- you are limited by the rate at which chemical reactions happen inside the battery as well as the resistive elements in the battery pack such as copper between batteries, battery contact points, and even the resistivity of the battery material itself. We model this as a series resistance in the battery pack:



*Each cell can be modeled as an ideal voltage source in series with a resistor we refer to as the ‘internal resistance’ of the cell*

Imagine that each cell in the battery pack has an internal resistance of  $1m\Omega$  and a voltage of  $3V$ , and that there are 40 cells in the battery pack. At zero current the total voltage of the battery pack is

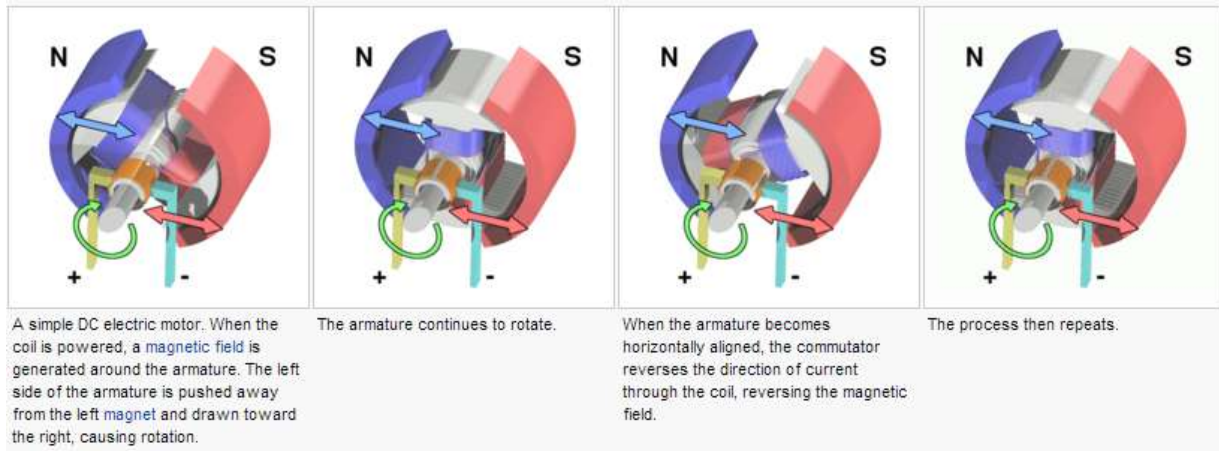
$3 \times 40 = 120V$ . As current is increased you start to see a voltage drop across the internal resistance of the cell, according to  $V = IR$ . At 10A the voltage drop across each virtual resistor is  $V = 10 \times 0.001 = 1mV$ , so each cell has an effective voltage of 2.99V and the total pack voltage is 119.6V. At 200A the drop across each resistor is 0.2V and total pack voltage you can observe is only 112V!

In addition to reducing the voltage observed by the motor controller, the internal resistance results in the dissipation of energy which heats the battery cells. For example, when 200A is flowing each  $1m\Omega$  cell dissipates  $P = I^2R = 200^2 \times 0.001 = 40W$  of heat- directly into the cell- or 1600W (2HP) across the entire battery pack. This is a huge amount of heat to dissipate- and the reason you must abide by the maximum current specification of the battery!

### Motor

Electric motors fall into two broad categories: brushed (commutated) motors, and brushless motors.

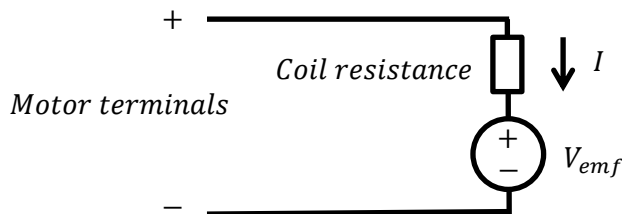
The following images from the Wikipedia article on brushed motors describe how a commutator works:



[http://en.wikipedia.org/wiki/Brushed\\_DC\\_electric\\_motor](http://en.wikipedia.org/wiki/Brushed_DC_electric_motor)

In short, the purpose of the commutator is to generate an alternating magnetic field from the fixed potential across the two terminals of the motor. An alternating magnetic field is required for continued motion of the armature.

We model the electric motor as follows:



When an electric motor is rotating it generates an opposing electromagnetic field which induces a potential, like a generator. The faster the motor is rotating, the greater the opposing field. The first motor equation describes this relation:

$$V_{emf} = K_e \omega$$

Where  $V_{emf}$  is the potential induced by the opposing EMF (commonly referred to as the back-EMF),  $\omega$  is the rate of rotation of the motor, and  $K_e$  is a motor constant. The reference material for the motor will provide a value for  $K_e$  or it's inverse  $K_v = 1/K_e$  (if your units are correct).

In a no-load scenario, you can calculate the rate of rotation of the motor if you know the motor constant  $K_v$  and the voltage applied across the motor terminals:

$$\omega = VK_v$$

The final motor constant,  $K_t$ , describes the relationship between current through the motor and torque at the motor shaft:

$$T = K_t I$$

Where  $T$  is torque and  $I$  is current through the motor. In a series circuit current is always the same so you can calculate current using the following formula:

$$I = \frac{V - V_{emf}}{R}$$

Where  $R$  is the coil resistance. You should be able to find  $K_t$  and  $R$  in the motor documentation.

You cannot observe  $V_{emf}$  so when you are controlling a motor you generally calculate  $V_{emf}$  based on the speed of rotation and then use  $V_{emf}$  to estimate the current, and therefore torque, that you can command with a given voltage  $V$  across the motor terminals.

When you are using metric units you should discover that  $K_e$  and  $K_t$  are equal in value- if not, check your equations!

Brushless motors do not use a commutator. They generally expose three power terminals, rather than two, and must be connected to a motor controller which drives it using a sinusoidal wave form. Rather than using a commutator to generate an alternating electric field from a constant voltage, a brushless motor is driven using an alternating voltage wired directly to the coils. The brushless motor and motor controller combination may be modeled in the same way as we have just modeled a brushed DC motor.

## Motor Controller

As we have established, the speed and torque generated by the motor can be controlled by varying the voltage applied across the motor. The role of the motor controller is to take the battery pack voltage, which is fixed, and convert it to a lower voltage which is applied across the motor. The motor controller will have some interface that you can use to control the voltage across the motor, and therefore its

speed and torque. All motor controllers will have a current limit which you cannot exceed. More advanced motor controllers will be able to measure the speed of the motor and provide a more abstract control interface- rather than controlling the voltage directly you control the speed and the controller compensates for irregularities in torque. Many controllers also contain more sophisticated protection circuits and digital serial interfaces for control and diagnostics.

## 2.2.2 Understanding the MATLAB model

Take some time to understand how the MATLAB model works.

The file is broken into three main sections:

### Initialization of constants and variables

The first section initializes constants and variables. There are many constants associated with the model, such as vehicle mass, traction limit, coefficient of aerodynamic drag, and even the track configuration:

```
29 - V_TRANSMISSION_RATIO = (V_MOTOR_SPROCKET / V_DRIVE_SPROCKET);
30     % Speed of vehicle is angular velocity of motor shaft (rad/s) *
31     % transmission ratio * drive wheel radius
32
33 - V_TRACTION_LIMIT = 0.8; % maximum accel/decel in g
34
35 - V_Cd = 0.3; % Drag coefficient
36 - V_Af = 1.2; % Frontal area in m^2
37
38 - V_MOTOR_SPROCKET = 100; % Motor sprocket radius in mm
39 - V_DRIVE_SPROCKET = 100; % Drive sprocket radius in mm
```

You should read through the constants and make sure you understand what they all mean.

```
3     %% Track
4 - track = { { 'Straight', 150, 10 }, ...
5             { 'Corner', 100, 10 }, ...
6             { 'Straight', 30, 7 }, ...
7             { 'Corner', 70, 7 }, ...
```

The track is stored in a multidimensional array. The notation `track{2}{3}` would retrieve the third parameter in the second segment of the track. The second segment is `{ 'Corner', 100, 10 }`, a corner length 100m with a maximum speed 10m/s, and therefore the third parameter is the speed 10m/s.

There are two classes of variables- variables which are recorded for later analysis, and variables which are not recorded and only have meaning within the simulation. Variables which are recorded are stored in a vector which corresponds with the 'time' vector:

```
75     % Variables that we are recording (to be plotted)
76
77 - time = zeros(N, 1);
78 - V_v = zeros(N, 1); % Vehicle velocity (m/s)
79 - V_dist = zeros(N, 1); % Distance travelled (m)
80 - V_trackpos = zeros(N, 1); % Distance travelled, reset at each lap
```

### The iterative loop

The main segment of the model is an iterative loop which updates the vehicle velocity according to the model.



## Track calculations

```
89     %% Track calculations
90     % Figure out our position around the track
91     trackpos = V_trackpos(i - 1);
92     % Figure out which section we're in
93     for j = 1:size(track,2)
94         if(trackpos < track{j}{2})
95             if(strcmp(track{j}{1}, 'Corner'))
96                 % It's a corner, maintain speed
```

The model uses `V_trackpos`, a variable maintaining the progress of the vehicle around the track in meters, to determine which segment of the track the vehicle is currently in (whether it's in a corner or a straight, and which particular corner or straight). At the end of the `%% Track calculations` section the variable `vtarget` is set with the instantaneous target velocity of the vehicle (the velocity that a driver would be trying to achieve at that particular time).

## Vehicle state

Once `vtarget` is known, the model calculates a voltage to apply to the motor.

Things get a bit complicated here. In the go-kart you (the driver) would adjust the throttle position according to your perception of speed and close a feedback loop. If you felt you were going too fast you would lift the throttle, and conversely if you started to slow down you would press the gas. The model simulates driver-throttle interaction using a simple PI loop, and treats acceleration and braking independently.

You shouldn't need to modify the controller in order to complete this lab, however you should understand how the drag force is calculated, how and why voltage and braking force are bounded, and how the net vehicle force is calculated.

Finally, the vehicle (longitudinal) acceleration is calculated (subject to the capability of the tires), and vehicle velocity, total distance travelled, and position around the track are calculated.

## Data Analysis

The final section of the script plots recorded variables for interpretation.

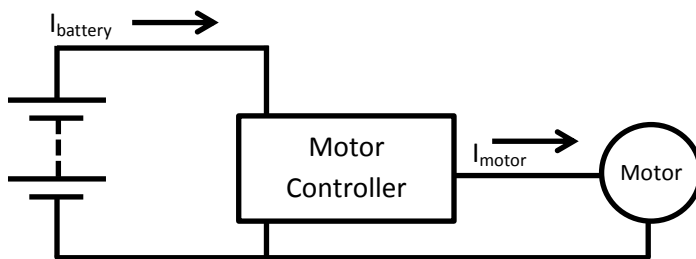
```
182 -     plota = subplot(2,1,1);
183 -     plot(time, V_v);
184 -     title(plota, 'Vehicle speed');
185 -     xlabel(plota, 'Time (s)');
186 -     ylabel(plota, 'Speed (m/s)');
187
```



## 2.4.0 Modifying the model

- 2.4.1: The motor constants,  $M_{Kt}$  and  $M_{Ke}$ , are incorrect (but still realistic). Use the information provided for the motor to determine the correct values. What are they? Hint: when expressed in the correct units,  $K_t$  and  $K_e$  are equal in value. Update the model with the correct values.
- 2.4.2: The motor is not 100% efficient. Take a look at the power curve for a similar motor: <http://www.robotmarketplace.com/products/images/ME0909-PowerCurve.pdf> Notice how a constant voltage is applied to the motor, however the rotational speed (RPM) of the motor declines as more torque is applied to the motor shaft. This is a symptom of motor non-ideality- the motor does not quite obey  $V_{emf} = K_e \omega$ . Add an inefficiency term to your model by multiplying the torque achieved for a given current by a factor 0.9. (This is a very simple approach that does not take the nonlinearities of the motor into account. As you develop more precise models for your own vehicle, you may want to model this differently.)
- 2.4.3: What is the top speed of the vehicle?
- 2.4.4: Modify the model so that it tracks the charge remaining in the battery pack. The total battery capacity, measured in amp-hours, is 40Ah (all the batteries are in series so the total capacity of the pack, in Ah, is the same as the total capacity of an individual cell, in Ah). You can calculate the charge remaining in the battery pack by 'coulomb counting'- measuring battery current in your model and multiplying by `timeStep` to calculate the amount of charge dissipated each time step.

Until now the model has been calculating only one current, the current through the motor. In practice the architecture looks like this:



Where  $I_{battery}$ , the current through the battery, is approximately

$$I_{battery} = \left( \frac{V_{motor}}{V_{battery}} \right) \times \left( \frac{1}{\text{motor controller efficiency}} \right)$$

For simplicity, assume the motor controller is 95% efficient across the entire operating range.

- 2.4.5: Starting from full charge, graph the remaining charge in the battery pack across three complete laps of the circuit. Approximately how much charge is dissipated? How many minutes would you expect the vehicle to run on this circuit?

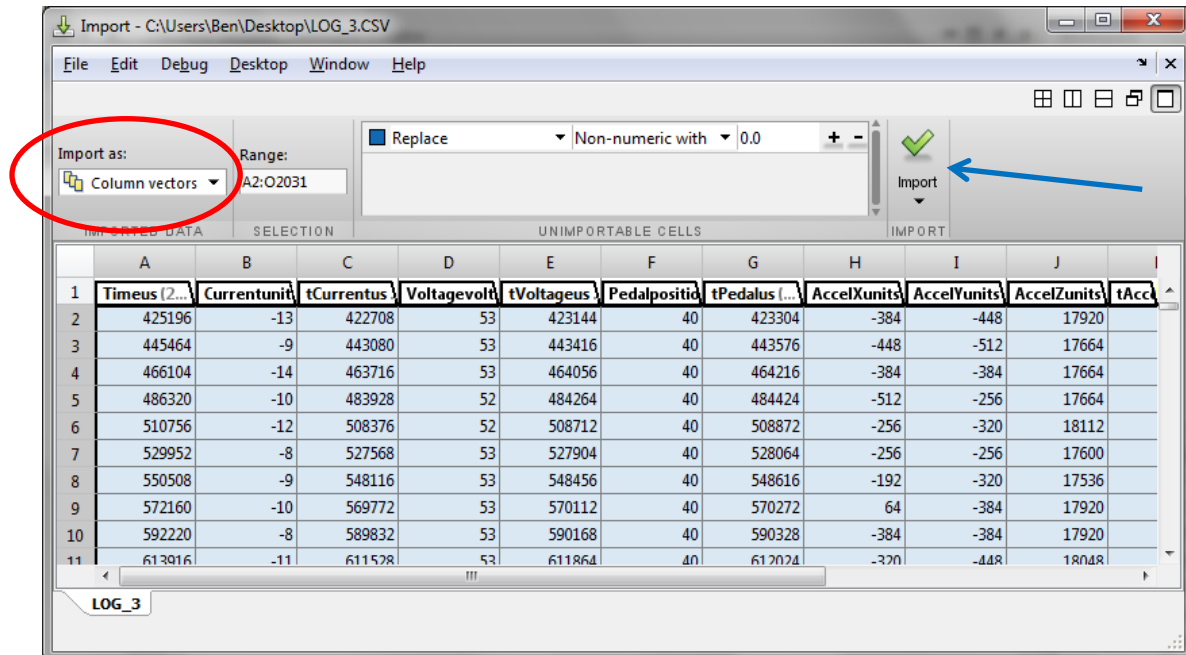
- 2.4.6: What is the maximum battery current? Is it within the maximum pulsed current rating of the battery? Is it within the maximum continuous current rating?**
- 2.4.7: Tweak parameters in your model which could be easily changed on the physical go-kart without swapping powertrain components- for example the gear ratio, drive wheel diameter and software motor controller current limit. Ensure that you do not exceed the limits of any of the components. What is the top speed when you set the maximum acceleration of the vehicle equal to the maximum acceleration the road surface allows? What is the highest top speed you can achieve while maintaining a maximum acceleration of at least  $5\text{m/s}^2$ ? What is the combination that you would pick, given your experience driving the go-kart?**

## 2.5.0 Analyzing Real-World Data

You have generated a CSV file using the Arduino based datalogger. The first step is to import the data into your Matlab workspace. Do this using the “Import Data” tool.

**NOTE: You need to import data into a separate instance of MATLAB, because in this section you will be comparing the model to the real world data, and some of the variable names (e.g. ‘time’) are the same and will overwrite each other.**

Make sure to import as “Column Vectors”!



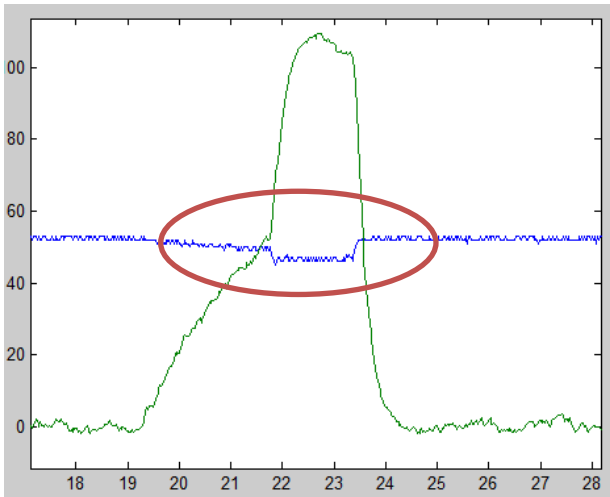
After you have imported the data, run `parse_data`. This script converts data entries to standard units and performs some filtering.

You will notice that each field in the data set has a ‘time’ column associated with it. This is because each measurement is taken sequentially, so accelerometer data in a given row isn’t quite matched up temporally with gyroscope data in the same row. You can decide whether this is significant for the analysis that you want to do.

Your data set contains the following fields:

- Time (in seconds)
- Vehicle speed (in m/s)
- 3 axis accelerometer data (in g)
- 3 axis gyroscopic data (in radians/second)
- Battery current (in amps)
- Battery voltage (in volts)

- 2.5.1: Plot battery current against time. (You will probably want to use `CurrentAv`, which is `Current` after a simple low pass filter.) What is the maximum current you achieve?
- 2.5.2: Integrate current over time using the `trapz` function. How many Ah do you draw from the battery pack over your run? Estimate the number of laps you should be able to run on a fully charged pack.
- 2.5.3: Plot battery power against time. What is the maximum power you achieve? What is this in horsepower?
- 2.5.4: Plot battery voltage and battery current against time on the same graph. You should observe the following dips in the battery voltage during periods of high current draw.



This is known as ‘sag’, and there is a corresponding behavior known as ‘lift’ when the battery is being charged. What might cause this?

- 2.5.5: Update your model with a similar current limit to what you observed on the actual go-kart (and change back any gearing ratio modifications you made). Update your model with the average battery voltage you observed in the actual go-kart. How much energy in Ah does the model consume per lap? How does this compare to the data you gathered?
- 2.5.6: Plot vehicle speed against time. What is the maximum speed you achieve? How does that compare to your model? Apply some averaging function to vehicle speed that allows you to calculate a realistic acceleration, and plot acceleration against time. What is the maximum acceleration you hit? What is the maximum deceleration you achieve?
- 2.5.8: EXTRA CREDIT: Analyze the accelerometer data. The accelerometer picks up acceleration AND gravity, and has an offset and scale error as well as random noise. Filter the data and plot a graph of longitudinal acceleration and current against time. What is the maximum acceleration you sustain? How does this compare to your model?
- 2.5.9: EXTRA EXTRA CREDIT: Construct a Kalman filter to combine acceleration and speed measurements. Plot filtered speed and acceleration curves against time. Include your MATLAB code. The Kalman filter

is an 'optimal observer' and is frequently used to combine several noisy estimates of a dynamic system.