Introduce yourselves!

We're going to tell you about a software algorithm that Team 1736 used in 2016 to prevent brownouts on the field.

Quote is not ours - it's from George Box, a statistician.

==========

Additional notes:

The quote - It's a playful reminder that you don't need to know everything about your robot, just the parts that matter.

Overview

- Background on Brownouts
- Physics Intro
- Modeling the Drivetrain
- Limiting the Battery Load
- Results from 2016

Here's what we'll cover in the next 50 minutes.

For those of you who aren't yet familiar, we'll discuss what a brownout on an FRC robot is, and why you don't want them.

We'll delve into some of the underlying physics, and use this to build up a mathematical model of how our drivetrain works electrically.

Finally, we'll present how to use this model to limit load on the battery, which is what prevents brownouts. Included in this will be our results from the 2016 season.

Warning: (some) Math Ahead.

- Minimal, Simplistic, Hand-wavey
- See the whitepaper for proofs

To not bore you all to death, we'll try to stay away from deriving equations on a powerpoint slide. We've both been through that in many college classes, we know it's not pleasant.

Instead, we'll try to share the key qualitative conclusions, and dribble in a few useful equations here and there. We'll keep the details of the math quite "hand-wavey" as we say in this business.

We do have a whitepaper we've published on Chief Delphi, and on our github. That doc will have the more complete derivations if you like math and want to see the full resource - it should have my contact info in it as well if you want to discuss.

==========

Image from http://strawstickstone.com/wp-content/uploads/2012/04/cautionmath.jpg

What is a brownout?

- Undesired component shutdown, due to **low system voltage.**

- System voltage gets low, sometimes.
  - Dead Battery?
  - Old Battery?
  - Big Load?

What are brownouts, and why do we care?

I'm willing to bet all FRC teams have seen this by now.

All components on the robot require some voltage to run. This System Voltage is provided by the battery's output. It goes lower when you are under load.

Fundamentally, Brownouts are caused when your system voltage gets too low, and various components cease to run..

Usually your system voltage is somewhere between 12 and 13 for a good battery and no load. Under some load, you might get down to 10 or 11. For more extreme loads or weak batteries, you go lower.

==========

Image from https://alexkourvo.files.wordpress.com/2016/09/006.jpg?w=450&h=253

Background on Brownouts

- New-ish on roboRIO
- Low system voltage causes problems
- Defined thresholds for certain events
  - Vsys < 6.8V
    - Motors & Servos turned off
  - Vsys < 6.3V
    - Power removed from SPI & I2C Devices
  - Vsys < ~4.5V
    - RIO reboot

To be specific, we define “System voltage” to be the voltage present at the input of the PDP (power distribution panel). It’s provided by the battery, and powers most of the devices on the robot.

The concept of the “brownout” on an FRC robot was officially introduced with the roboRIO, but the underlying low-voltage problem still existed in the cRIO and previous systems.

The Brownout triggers that the roboRIO added gave some better definition to how bad each level of system voltage drop was - predictability is good, but the bad effects are still present.

The 6.3V threshold has burned Casserole in the past - some of the I2C devices require an initialization sequence - if they accidentally restart during a match, and we don’t catch it in software, we can be reading wrong data from the sensor.

Radios, coprocessors, and other parts come in and out at different thresholds.

Reboots generally take a long time, and will likely cause you to be out of commission for a big chunk of the match.

All this put together, it’s a good idea to not let brownouts happen.

==========

Additional notes:

Some info from
http://wpilib.screenstepslive.com/s/4485/m/24166/l/289498-roborio-brownout-and-understanding-current-draw

Additional notes: the 6.8 threshold is controlled within the RIO - the processor and FPGA are still functioning, but have purposefully removed power from external devices. The 4.5V threshold is where internal components and/or the boost power supply for the 24V rail for the RIO begin to turn off. This threshold is approximate, since it's dependant on the tolerances of the components in the power supply, the resistance of the wires, etc.

Image from https://cdn.meme.am/cache/instances/folder504/26873504.jpg

Pre-Existing Solutions

- Mechanical design updates
  - Fewer motors
  - Less-aggressive gearing
  - Wheel modifications (more slippy)
  - Good answers, but compromises may not be acceptable
- Ramp motor commands in software
  - Also good, but requires some tuning...

As most of you are probably aware, there's lots of ways folks have solved this already.

Generally, mechanical solutions are first - using fewer motors or a less-aggressive gear ratio will do wonders. In a pinch, zip ties or duct tape on the wheels will make them more liable to slip… less performance in a pushing match, but less load on the battery. These are definitely good answers and should probably be pursued first. However, software can help you fine-tune the performance more.

One of the more common software solutions is to simply limit the max rate-of-change of the motor command. IE, if the driver commands full forward to full reverse really fast, the software will slowly ramp the motor command to the new value. This works because it gives the motor time to change speed, which minimizes time near stall. It does require some guess-and-check tuning to get right, and doesn't account for changing battery conditions….

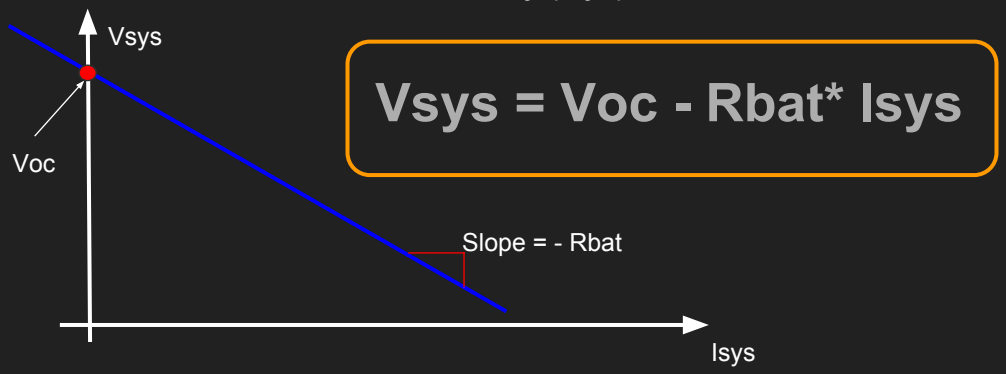
These definitely work well in many cases, there's a lot of team experiences which would show that. But, I believe we can do better.

Understanding Batteries

Let’s start to delve into some of the physics behind our drivetrain components. We’ll start at the source of energy, the battery.

Electrical Model for Battery

- Batteries convert chemical energy into electrical energy
- Chemical reaction provides a voltage (Voc), some internal resistance (Rbat)
- Battery provides an output voltage to the robot (Vsys).
- Robot exerts a load on the battery (Isys)



This is the model we will use to represent a battery. We found it very useful

The chemical reaction inside the battery provides electrical energy. We care about the properties of the voltage and current provided by that reaction.

The reaction creates a voltage - a push which can create electric current. Additionally, there is some resistance in the battery, from the physical construction of the battery and the byproducts of the reaction.

The combination of the voltage and resistance means that as we draw more current out of the battery, the system voltage goes down.

The more load we put on the battery, the more current will be drawn.

When little to no current is drawn, the battery will have a certain voltage at its output terminals. This is called the “Open Circuit” voltage. For a freshly charged battery used in FRC, this can be as high as ~13 volts.

For a decent battery with cables, 1736 found the resistance to be around 0.025 ohms. It’s small, but not negligible.

Now for some math.

This equation describes how the system voltage will decrease when current is drawn.

With no current, Vsys = Voc

When current is drawn, Vsys will be less than Voc

More current will cause more drop

A larger Rbat will cause more drop.

This linear relationship is illustrated by the graph.

Why This Matters:

- **Vsys = Voc - Rbat* Isys**
- Given Voc, Rbat, and Isys, we can calculate Vsys
- If we can calculate Vsys for a theoretical Isys, we can *predict* a system voltage drop before it happens
- If we foresee brownout-inducing Vsys levels, we can also *prevent* them.
- The 3-part quest:
  - *Characterize Battery (calculate **Voc** & **Rbat**)*
  - *Calculate theoretical **Isys** (motor physics)*
  - *Prevent bad behavior*

We have our first equation, which will be very important. We'll keep coming back to it. Again, this is the mathematical description of how our battery will work.

It is also a math equation. So, if we know certain values, we can figure out other values. This is algebra.

Given a given current draw Isys, and some info about the battery (Voc and Rbat), we can calculate the system voltage (Vsys)

Remember, we want to prevent brownouts. Brownouts are caused by Vsys getting too low.

Given Voc, Rbat, and a *predicted* Isys, we can *predict* Vsys. If we can *predict* Vsys, we can *prevent* brownouts.

So now, We will divide our quest for those numbers into three parts:

1) Characterize the parameters of the battery - this is the process of figuring out the values we don't know, from values we do know.
2) Use some more physics to predict motor current
3) Do something useful with that information - prevent brownouts.

==========

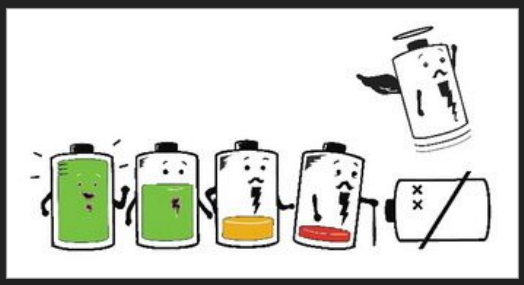
Image from https://cdn.meme.am/instances/500x/64866690/hobbit-going-on-an-adventure-lets-go-on-an-adventure.jpg

Part 1: Characterizing Batteries

Battery Discharge

- Open Circuit Voltage (**Voc**): *Down slightly*
- Internal Resistance (**Rbat**): *Increases*

CAUSES



- Bigger **Vsys** variation with changing load (**Isys**)
- When load gets bigger, **Vsys** will dip a *lot* lower.

Vsys = Voc - Rbat* Isys

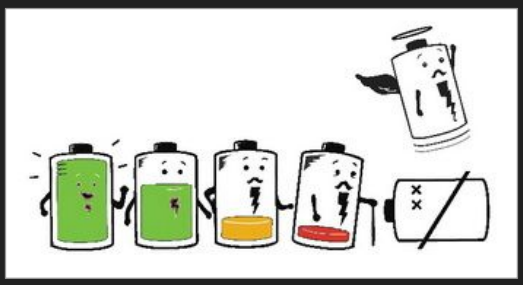
What happens when a battery discharges? Say, over the course of a match.

Voc will stay about the same, maybe drop slightly… depends on the battery.
Resistance will go up, an appreciable amount.
This increase causes Vsys to become more unstable. Under no load, it will probably stay around 12V. Under load, it will swing around way more

==========

Image from
http://rightyaleft.com/wp-content/uploads/2012/08/average-laptop-battery-life.jpg

Battery Discharge



- Open Circuit Voltage (**Voc**): *Down slightly*
- Internal Resistance (**Rbat**): *Increases*

CAUSES

- Bigger **Vsys** variation with changing load (**Isys**)
- When load gets bigger, **Vsys** will dip a *lot* lower.

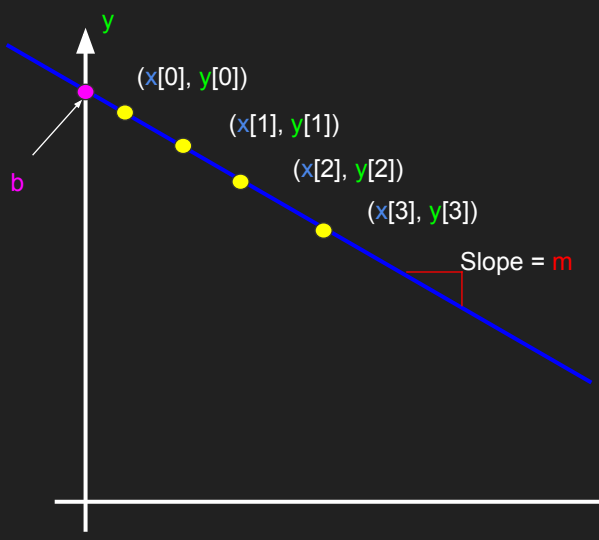
$$Vsys = Voc - Rbat^{*} Isys$$

Knowing these matters

The main takeaway - Knowing these two values will matter to our algorithm.

This means we should track them for the specific battery we are using on the robot.

Review - Slope-intercept form



$$y = mx + b$$

Given an input x, you can find an output y.

m and b are assumed constant.

Given multiple input x & y pairs, you can back-calculate values for m and b .

Before describing the algorithm, we'll review a crucial piece of math.

I suspect this is an algebra review for most folks.

Let's remember what slope-intercept form of an equation looks like.

Let Y be some function of X, which takes the following form, Y = MX+B.

On a cartesian plot of X & Y, the equation describes a line.

Specifically, The line described will have a slope of M, and intersect the Y axis at B.

A key point which we should note - Given a set of X and Y pairs, we can calculate what M and B should be.

Now, we'll use this crucial fact to help describe battery behavior.

Battery Characterization

- We *do* know Vsys and Isys - measured from the PDP
  - Slower, somewhat noisy, and delayed….
    - Not good for rapid decisions
    - But, battery parameters change slowly over time.
- We use measured Vsys and Isys pairs to characterize the battery
- Recall **Vsys = Voc - Rbat * Isys**
  - This is nearly in **y = mx + b** (slope intercept) form:
  - **Vsys = (- Rbat) * Isys + Voc**
- Algorithm:
  - Measure & remember a set of **Vsys**, **Isys** pairs over recent history
  - Find a best-fit line through those pairs
  - The **slope** and **y-intercept** of this line are your battery parameters

As some of you have probably already noticed, we do have access to Vsys and Isys - both measured from the PDP.

These come in over a datalink, there's some noise in their measurements… so, they're not super precise. I wouldn't want to use them for single-loop decisions of a current limit, they simply don't have the bandwidth, at least not in 2016… But...

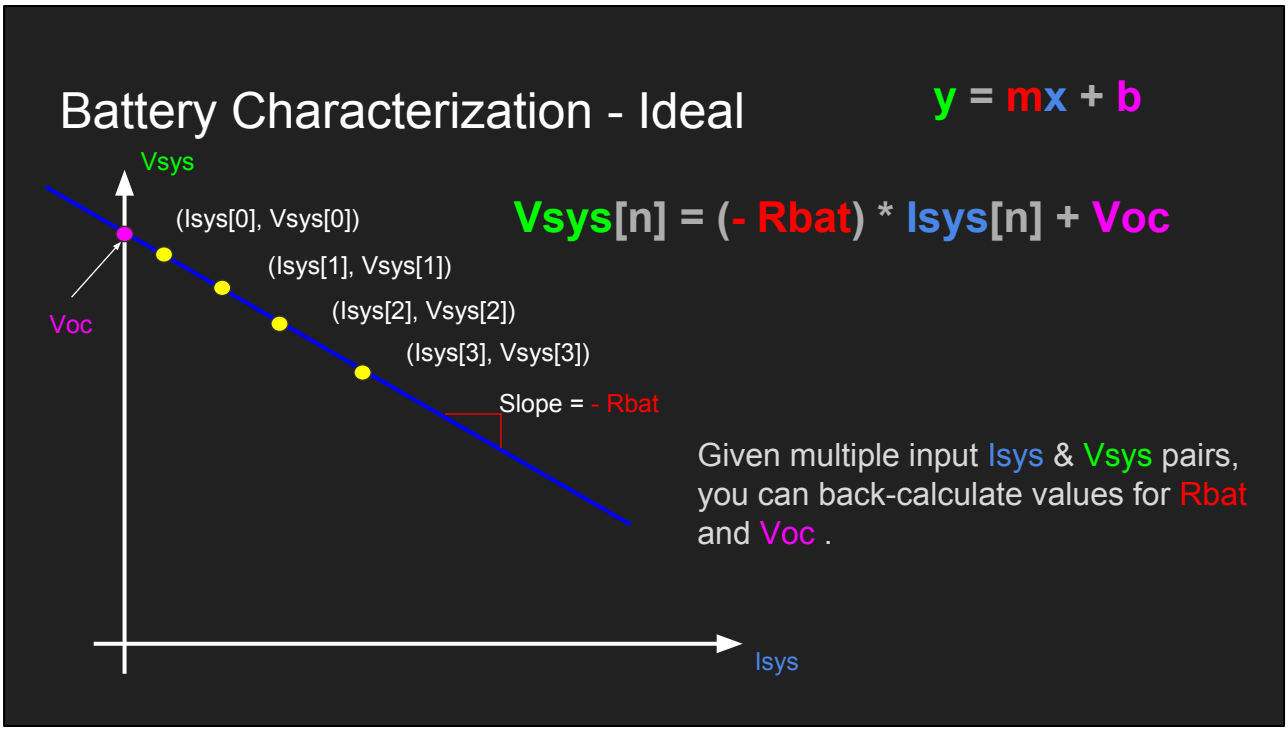
With some careful processing, we *can* use them for deriving other measurements which don't change much over time…. Such as our battery parameters!

Here's how we use our measured values (Vsys and Isys) to get the values we want (Rbat and Voc):

First, note that we can nicely re-arrange the formula into slope-intercept form. This makes it such that:

--On an X/Y plot of Isys on the X axis, and Vsys on the Y axis..
---The Y-intercept is Voc
---The negative of the slope is Rbat

So, if we simply remember a set of recent Vsys,Isys pairs, we can extract a slope and y-intercept which will yeild Voc and Rbat, the numbers we're looking for from part 1



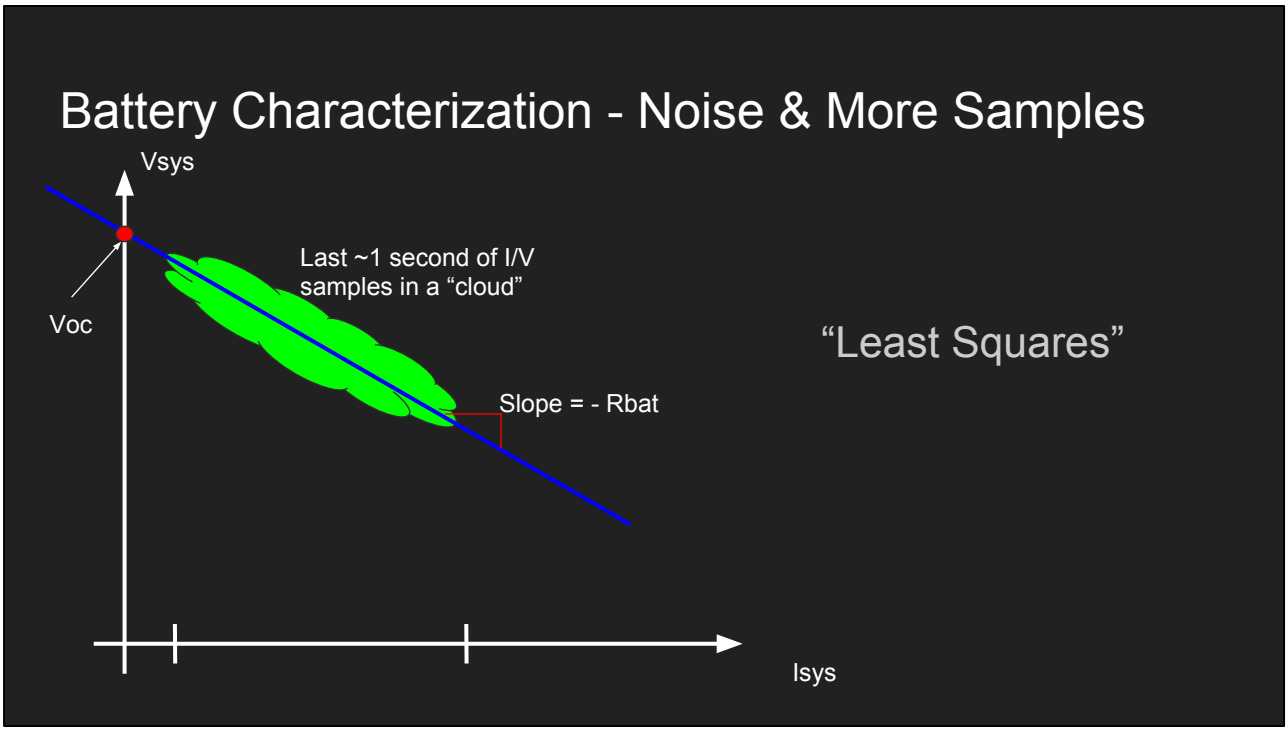
This is how the characterization works in the ideal case.

In this simple illustration, we've remembered the last four measurements. (that's the number in the bracket).

We then draw a line through those points.

The negative of the slope of the line gets you Rbat, and the y-intercept gets you Voc.

Remember again, this is the ideal case. In reality, the measurements will have noise, and you'll have more of them!



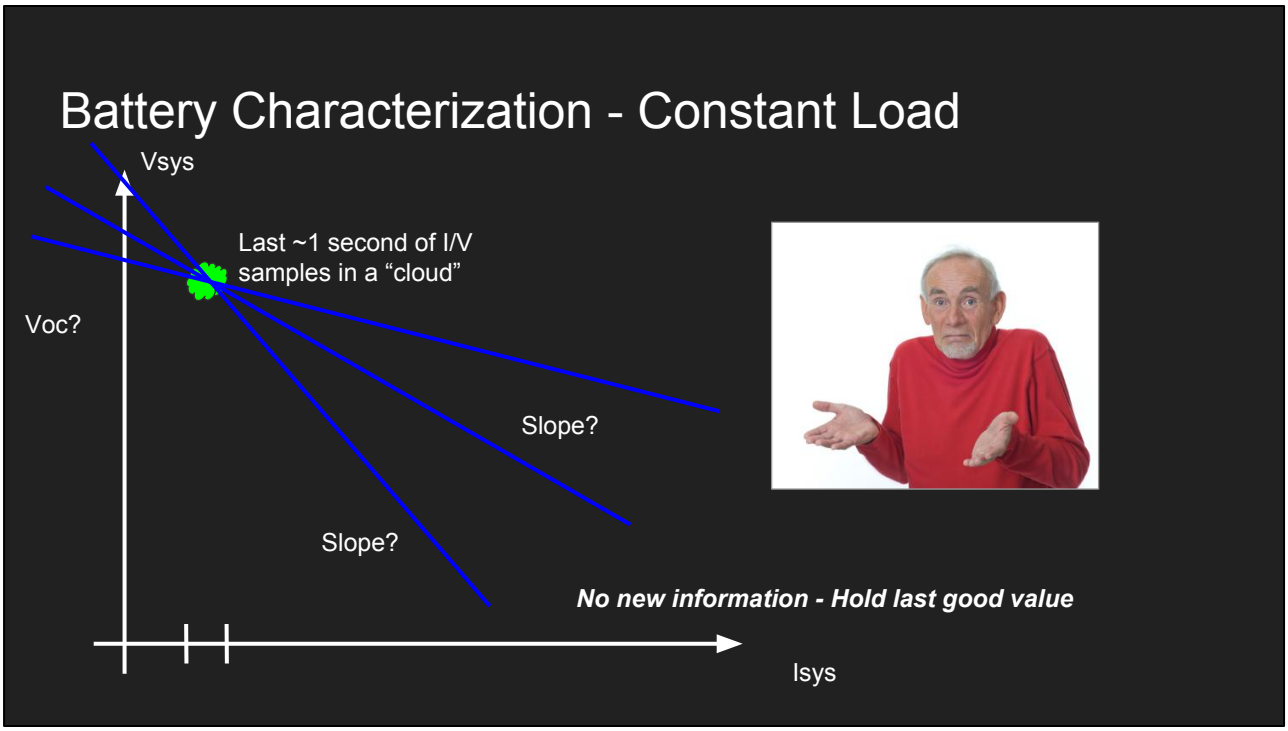
This is closer to the real picture. Although all these points should be on a line, timing skew and noise will perturb the points away from the line.

With a lot of inputs *near* the ideal line, you now have a point-cloud through which you are drawing a *best-fit* line.

There are a lot of existing and well-known algorithms for determining how to draw a best-fit line given a set of points - we borrowed one from a professor’s webpage, and you can see it in our 2016 codebase.

As long as this cloud of points is spread out enough, the best-fit line should do a good job of matching the parameters you desire.

Note that “spread-out enough” means there are many different Isys values over the past second. This happens when you’re accelerating a lot, pushing, climbing, etc. If you’re just sitting still, it won’t look like this. You’ll have all your Isys values bunched up…..



Like this!

Here’s a problem that arises when the load remains constant. The point cloud will be very compact.

This means that any best-fit line will be dominated by sensor noise, not by actual information about the battery.

For 1736, we get around this by simply holding the last good value when the spread gets too small. There were some other methods suggested, but this one worked just fine for us.

As a side note, We defined *spread* to mean “standard deviation” and “last good” to incorporate some concept of highest spread, but that was just an implementation that happened to work for us. The whitepaper explains it in more detail.

Battery Characterization - Conclusion

- By remembering measured Voltage and Current pairs from the PDP…
- We can do some math….
- Then, we can estimate battery parameters **Voc** and **Rbat**

Part 1 of quest… Completed.

Here are the conclusions of part 1 - Even if we lost you, remember these crucial facts:

We measure voltage and current at the PDP, and remember it for some period of history.

We do some math, involving best fit lines.

The estimates for Voc and Rbat parameters come from that best fit line.

These estimates are the two components we needed for part 1.

==========

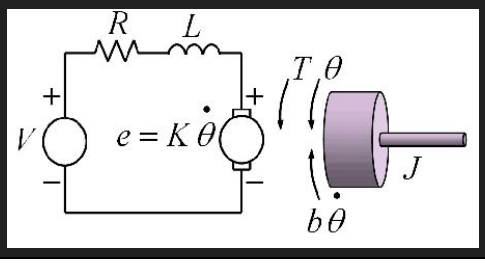
Image from https://openclipart.org/image/2400px/svg_to_png/202732/checkmark.png

Part 2: Calculating Motor Current Draw

Now that we know Voc and Rbat, we move on to estimating how much current we will be drawing from the battery - our theoretical Isys

Motor Physics

- Motors convert electrical energy into mechanical energy
- Motor Current → Torque (rotational pushing) on output shaft
- Motor Counter Electromotive Force (CEMF) → Opposing voltage generated within motor due to rotation
  - Motors == Generators because of this effect
  - Speed-to-Voltage Ratio is defined as the “motor velocity constant” (Kv)



Here’s some basic info about the motor model we’re using. We’re presuming brushed DC motors, like CIM’s used for FRC. The math applies to most FRC motors.

Like most electrical devices, you apply a voltage to a motor, and then some current flows through it.

The more current through the motor, the more torque it produces. This is why aggressive gear ratios cause lots of current draw.

For us, the more important part of a motor for us is the CEMF. CEMF stands for “Counter Electromotive Force” is a voltage which the motor generates while rotating.

Literally, the motor is acting as a generator. There’s power applied at the electrical inputs, which starts it rotating. But as it rotates faster and faster, it acts more and more like a generator, which *pushes back* against the input power. Eventually these two forces balance each other out, when the motor hits a steady-state constant speed

==========

Additional notes:

See https://en.wikipedia.org/wiki/Motor_constants for more info.

Image from http://www.maplesoft.com/support/help/content/1056/image44.png

Motor CEMF Effects

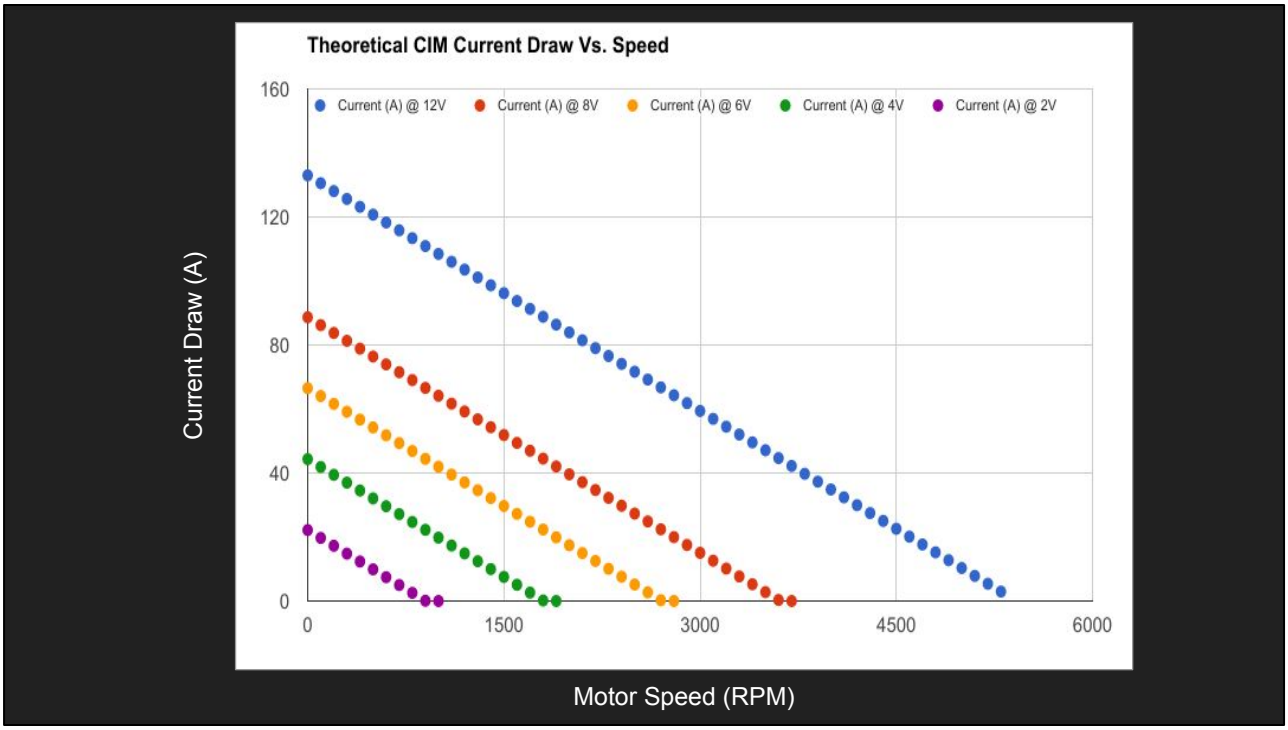
- CEMF Opposes the applied voltage to the motor
  - Voltage is applied by the speed controller
- The *bigger* the speed, the *bigger* the CEMF.
- The *bigger* the CEMF, the *lower* the current draw
- For the same CEMF, lowering the applied voltage will lower current draw

The more CEMF you have pushing against the applied voltage, the lower your current draw will be.

This means that if we want to keep current draw low, we can't apply big voltages until the speed is high enough.

==========
Additional notes:

See http://files.andymark.com/CIM-motor-curve.pdf for more data.



CIM motor current draw versus speed - just a quick example to show how this effect works.

Here's what to take away from this chart:
If you want less current draw, you have two options:

--More speed (in the correct direction)
--Less input voltage

Estimating Motor Current Draw

- Inputs needed:
  - Winding Resistance (calculated from motor's stall characteristics)
    - **Rm** = Vstall/Istall
  - Velocity Constant (calculated from motor's free-load characteristics)
    - **Kv** = (Vfree - Rm*Ifree)/ωfree
  - Motor Speed (Measured from encoder in RPM) (ωm)
- Use Formulas:

$$Im = \frac{Vm - \omega m * Kv}{Rm}$$

Here's the math for our model of the physics.

The last formula is most crucial:

The motor current (Im) depends on:

- Motor input voltage (Vm)
- Motor speed (ωm)
- Some constant values, from the motor's datasheet (Kv, Rm).

Given a speed (ωm) and an applied voltage (Vm), you can calculate the motor's current draw (Im)

The key takeaway is a slight modification to the above:

Given a speed (ωm) and a *commanded* voltage you are *about to apply* to the motor (Vm), you can *predict* what the current draw is about to be.

Estimate Total Current Draw - Conclusion

- Get driver commands
- Do some math to get the predicted **Im** for each motor
- Add up all **Im** terms.
- This is your predicted **Isys**

$$I_{sys} = \Sigma I_m + ...$$

Part 2 of quest…. Complete

Here's the surprisingly abrupt conclusion:

Intended applied voltage comes from what the driver's input commands are.

Motor speed can be measured with an encoder, maybe some scaling math for your gear ratios.

Use this to do math for all big motors (mostly the ones in your drivetrain), and add up the predicted current draws.

This is now your predicted Isys! We now have all components needed to *predict* a system voltage at any given time.

==========

Image from http://web.gps.caltech.edu/~jac/IDKArD/images/completed.png

Part 3: Preventing Bad Behavior

Pulling it all together

- Recall: **Vsys = Voc - Rbat * Isys**

Let’s come back to our initial formula for system voltage.

Note that everything on the right-hand side of the equation has been calculated for a certain set of driver inputs.

Should we actually use the driver’s commands exactly? Will Vsys get too low?

==========

Image from http://acfc7e9a085c910eb0418fc0.vgs0klcdfgmuiwahrfg2.netdna-cdn.com/wp-content/uploads/2012/10/puzzle-pieces-and-four-people.jpg

Pulling it all together

- Recall: **Vsys = Voc - Rbat * Isys**
  - ***Voc***, ***Rbat*** *were calculated in Part 1*
  - ***Isys*** *was calculated for the present driver commands in Part 2*
- By applying the above equation we now have an **estimate** for what **Vsys** will become if we execute the driver's commands exactly
- *IS IT OK???*

Let's come back to our initial formula for system voltage.

Note that everything on the right-hand side of the equation has been calculated for a certain set of driver inputs.

Should we actually use the driver's commands exactly? Will Vsys get too low?

==========

Image from http://acfc7e9a085c910eb0418fc0.vgs0klcdfgmuiwahrfg2.netdna-cdn.com/wp-content/uploads/2012/10/puzzle-pieces-and-four-people.jpg

Limiting

- If your estimated Vsys is too low, we shouldn't do exactly what the driver commanded.
- Required mitigation: Reduce battery load.
  - Scale back drivetrain commands?
  - Power/Energy budget for all components?
- Complete answer is implementation dependent.
  - 1736: Try smaller and smaller scaling factors on drivetrain until we find one that works

$$I_{sys} = \sum I_m + \ldots$$

Part 3 of quest…. completed!

If you estimate that your driver commands is causing Vsys to be too low, you can now do something!

For 1736, in 2016, reducing motor command to the drivetrain was good enough to pull back the battery load.

We used a simple iterative solver to find a scaling factor that met our system voltage minimum criteria.

The whitepaper does a quick example of inverting the equations for our drivetrain system to exactly calculate the scaling factor at each loop. However, due to the absolute value signs involved from the speed controllers, the reverse solution is non-trivial

==========
Additional notes:

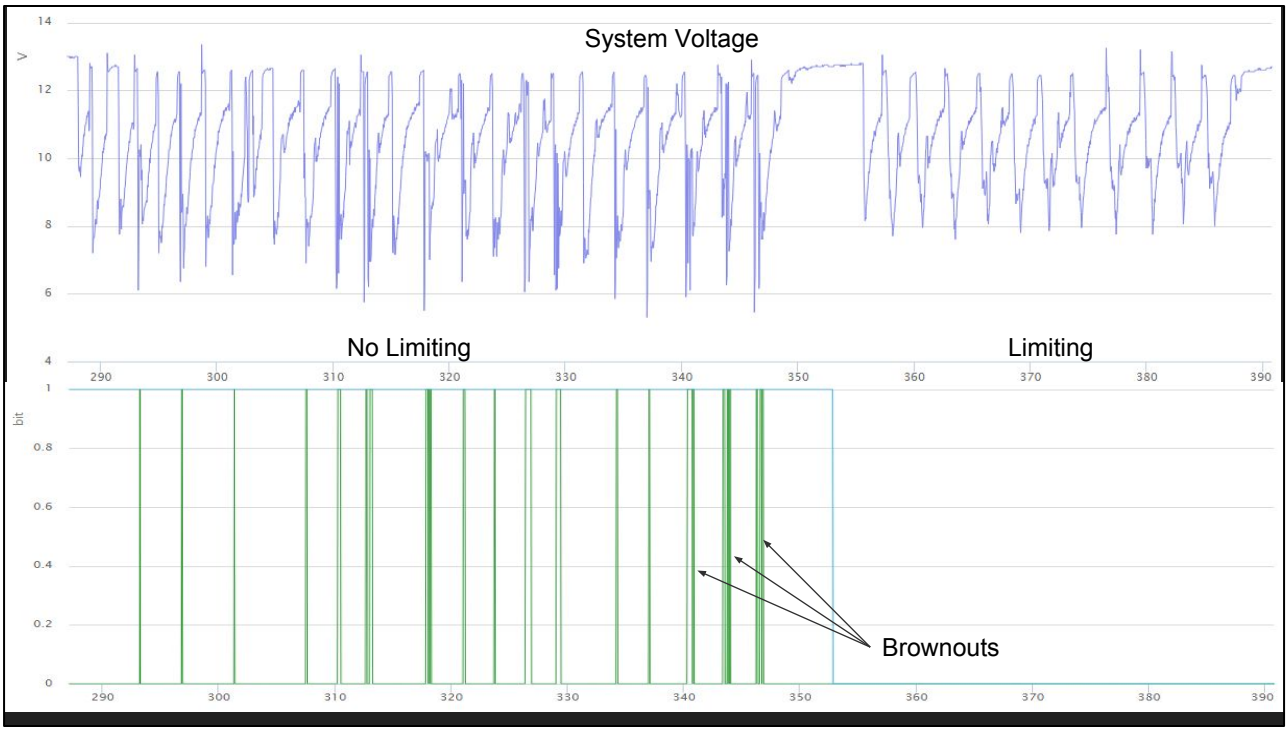
Iterative solver = Guess a limiting factor in the range [0.9, 0.1]. Re-calcualte Vsys using the limiting factor. If Vsys is still to low,k try a lower limiting factor. If it's ok, use that limiting factor.

Real-world implementation

- Many resistances unaccounted for
  - Wiring, Speed Controllers, circuit breaker, connectors, etc.
- Capacitance/Inductance not modeled
- Some math presumes slowly-changing signals
- Battery parameters assumed independent of present load… mostly true?
- Not all current draw sources accounted for
- Filtering on measured values required
- But…. Good enough?
  - *All models are wrong….*

1736 did a decent amount of tweaking to get the drivetrain current transient prediction to match the actual measure current pretty closely (accounting for delay). However, the steady-state actual/estimates had a decent amount of mismatch. Still not sure why - we have much to learn…. But, we got the usefulness out, so how much is it worth to go back and analyze more?

Results



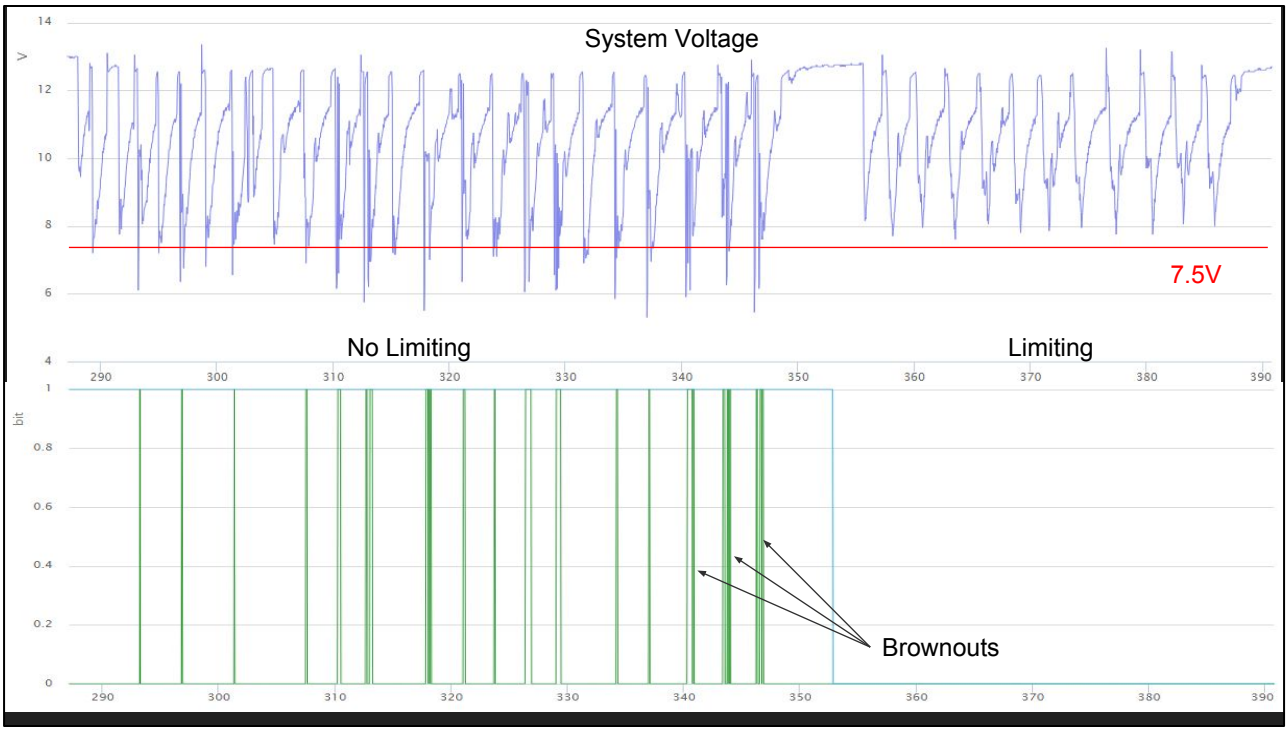
The “so what”.

Top is system voltage, bottom is brownouts (green) and limiter disable (blue)

Test Procedure:

- Charge up an old battery
- Run high-gear fwd / rev cycles without limiting until brownouts start to happen regularly
- Enable limiting
- Run cycles again, observing no brownouts.

Notice the limiter keeps the system voltage (mostly) above software-defined threshold of **7.5V.**



The “so what”.

Top is system voltage, bottom is brownouts (green) and limiter disable (blue)

Test Procedure:

- Charge up an old battery
- Run high-gear fwd / rev cycles without limiting until brownouts start to happen regularly
- Enable limiting
- Run cycles again, observing no brownouts.

Notice the limiter keeps the system voltage (mostly) above software-defined threshold of **7.5V.**

Questions?

Special Thanks To:
Sponsors: Caterpillar Inc., PTC, Peoria Police Benevolent Association
Key Content Reviewers: Jeremey Lee, Larry Schmidt, Ether

Backup Slides

Electrical Primer

Voltage & Current

- Electricity = movement of electrons
- Voltage = how much force we're pushing on those electrons with
  - Always measured between two points
  - Measured in units of Volts (V)
  - Variable is usually V
- Current = how many electrons are moving
  - Always measured at a single point
  - Measured in units of Amperes (A)
    - Aka Amps
  - Variable is usually "I"

Super simple explanations of electricity, for the uninitiated. This is greatly over-simplifying what i'm sure many of you have degrees in, but hopefully it will serve to make our point.

"Electricity," as we use it in FRC, can be thought of as simply the movement of electrons from one point to another

There's got to be a force to get anything to move. The force that causes electrons to move is called "Voltage"

Lots of volts means we're pushing hard on the electrons to move. Lots of push generally means you'll get lots of electrons moving.

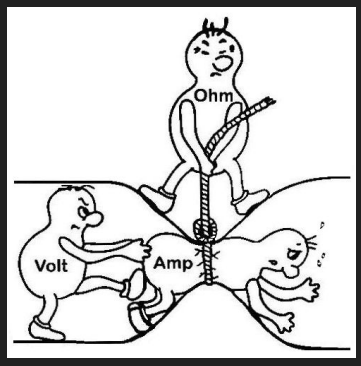
"Current" is a measurement of how many electrons passed a given point in one second. Similar to flow-rate, for those of you who like liquids.

Note that in most equations, current is indicated by the letter "I". This is because current was first described by a French guy, and they have different words for stuff in French. I would have thought the same would have happened with voltage (which was first described by an Italian dude), but I guess we got lucky.

It's important to remember that voltage and current will definitely be related, but not always in a simple way. We'll try to make it simple though.

Resistance

- Opposes the flow of current
  - “Opposes” means it generates a push in the opposite direction
  - Recall that voltage is an electrical “push”
- Everything* has some resistance
- “Nice” resistors have the property:
  - **V = I * R**
  - V = pushback voltage
  - I = current going through the resistor
  - R = some ratio (the “Resistance” of the material, measured in ohms)



Resistance is the property of a material that causes it to oppose the flow of electricity.

Whenever you have current flowing through a material, the material will push back. This “push-back” is measurable as a voltage.

The more current flowing, the higher that pushback voltage will be.

The exact ratio of current to pushback voltage is a constant, known as the “resistance” of the material, measured in units of Ohms

As weird as the classic anthropomorphic-peanut diagram of resistance is, it does to a good job of illustrating the relationship: The voltage peanut tries to push the Amp (current) peanut along the conductor, but the Ohm (resistance) peanut fights back… with a rope, I guess….

Everything = everything you’d use in FRC. Some superconductors have weird quantum properties which allow them to hit exact zero resistance, but that’s well beyond the scope of this discussion.

The “Lumped Circuit” Model

- We group the various properties of a device into “lumps”
- We connect those lumps to show relationship
- The Resistance lump:

- The Voltage lump:

The Lumped Circuit Model is one of the most basic of any ways to understand electrical circuits and systems

It’s taught in probably every electronics course you’ll take in college, and at least once in high school physics courses. You may have never heard its name, but you’ve probably seen it.

The basic idea is simple enough - take any of the electrical properties of a device, and smush them into various “lumps”. Then interconnect those lumps to show the relaltionships between the properties. Do it right, and you’ll have a diagram which you can use to help do math.

This math can be used to derive useful info - how much voltage your battery is putting out, for example.

Battery Load

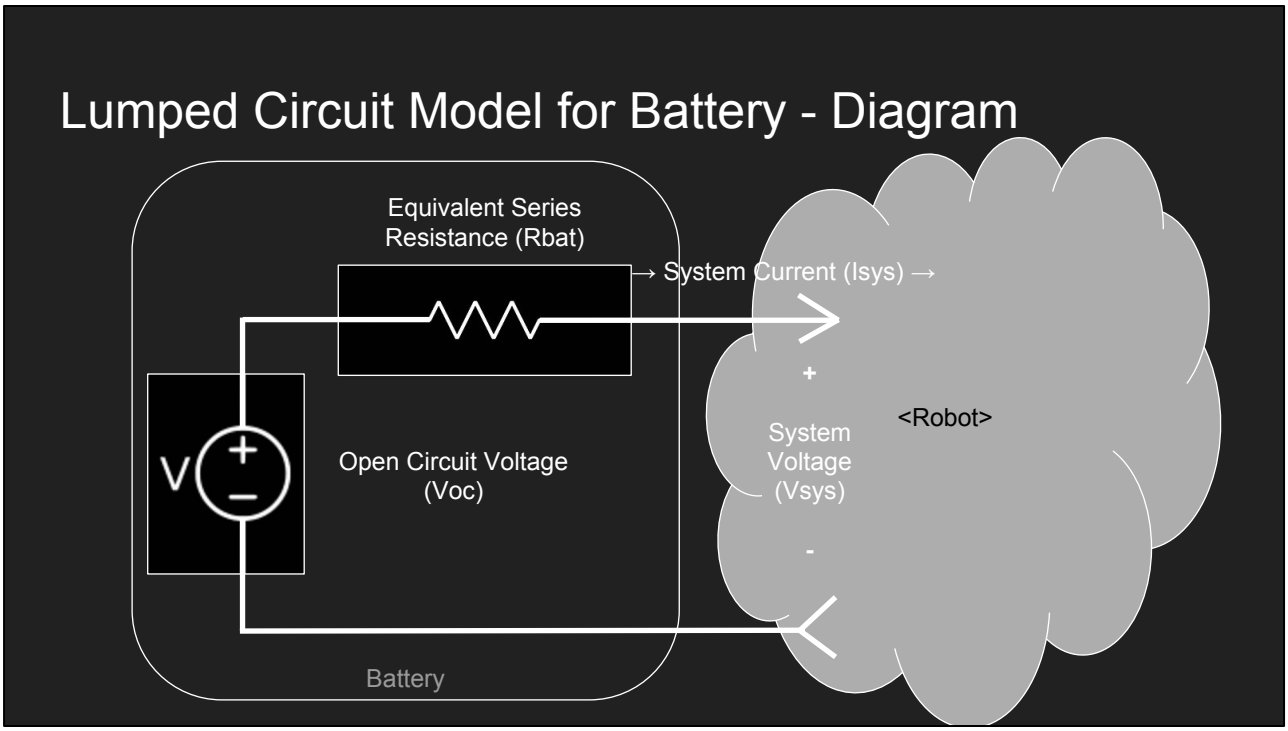
- For this presentation, we define “Load” to be the amount of current pulled from the battery
  - Isys
  - Can be measured at the PDP
  - Big load = lots of current (~150A or more)

“Load” can mean different things, depending on the context.

We use it a lot in this presentation, so we’ll explicitly define it. When we say a battery is “under load”, it means we’re drawing lots of current from it.

Isys is the variable we’ll use in equations on it. This variable indicates “The total current drawn out of the battery”

Battery Lumped Circuit Model



This is the model we will use to represent a battery. We found it very useful

The chemical reaction inside the battery provides electrical energy. We care about the properties of the voltage and current provided by that reaction.

The voltage source represents the energy input into the system. The voltage is physically provided by the chemical reaction inside the battery.

The resistor represents the fact that the battery is non-idea. The materials and construction are such that current flowing through the battery (and therefor through the robot) will decrease the system voltage. We will lump all this into the “Equivilant Series Resistance”.

The more load we put on the battery, the more current will be drawn.

When little to no current is drawn, the battery will have a certain voltage at its output terminals. This is called the “Open Circuit” voltage. For a freshly charged battery used in FRC, this can be as high as 13.5 volts.

For a decent battery with cables, 1736 found this to be around 0.025 ohms. It’s small, but not negligible.

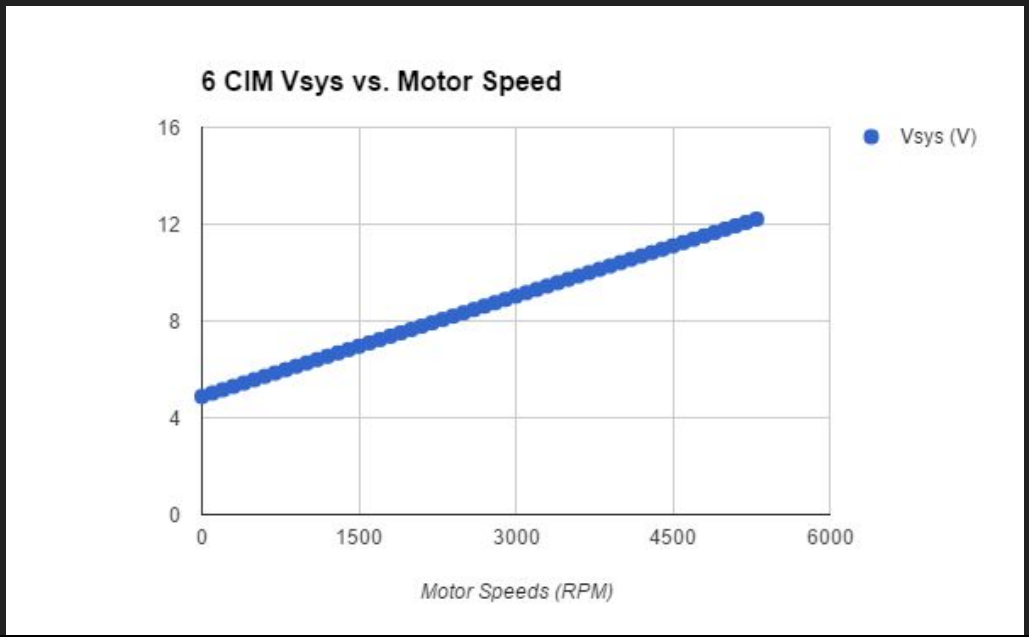
==========

Additional notes:

Backup slides have a primer on electronics

System Voltage vs. Motor Speed Calculations

Steady-State System Voltage Vs. Motor Speed @12V



Interestingly, it's linear.

This is steady state current draw - Inductance or other system voltage effects might come into the picture….

It also presumes certain battery parameters are constant. See calculations spreadsheet for more info