Stryke Force Talon SRX Motor Training Course

December 9, 2017

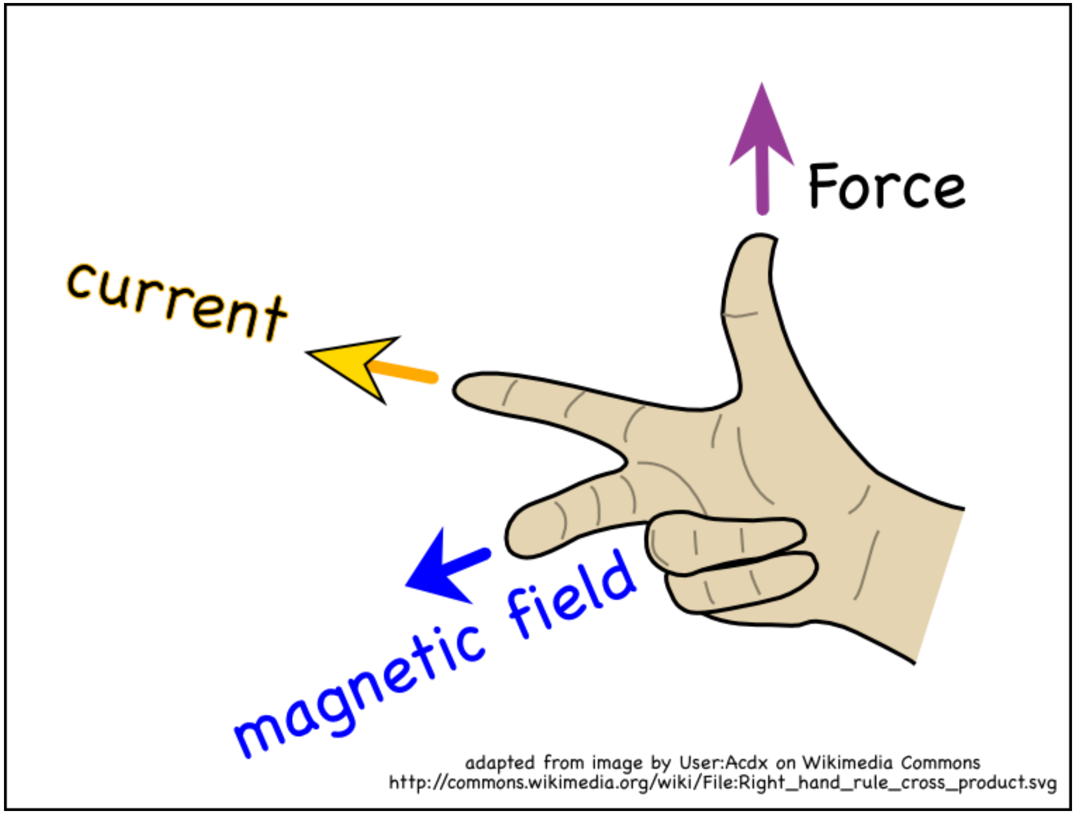
Jerry Culp

Agenda

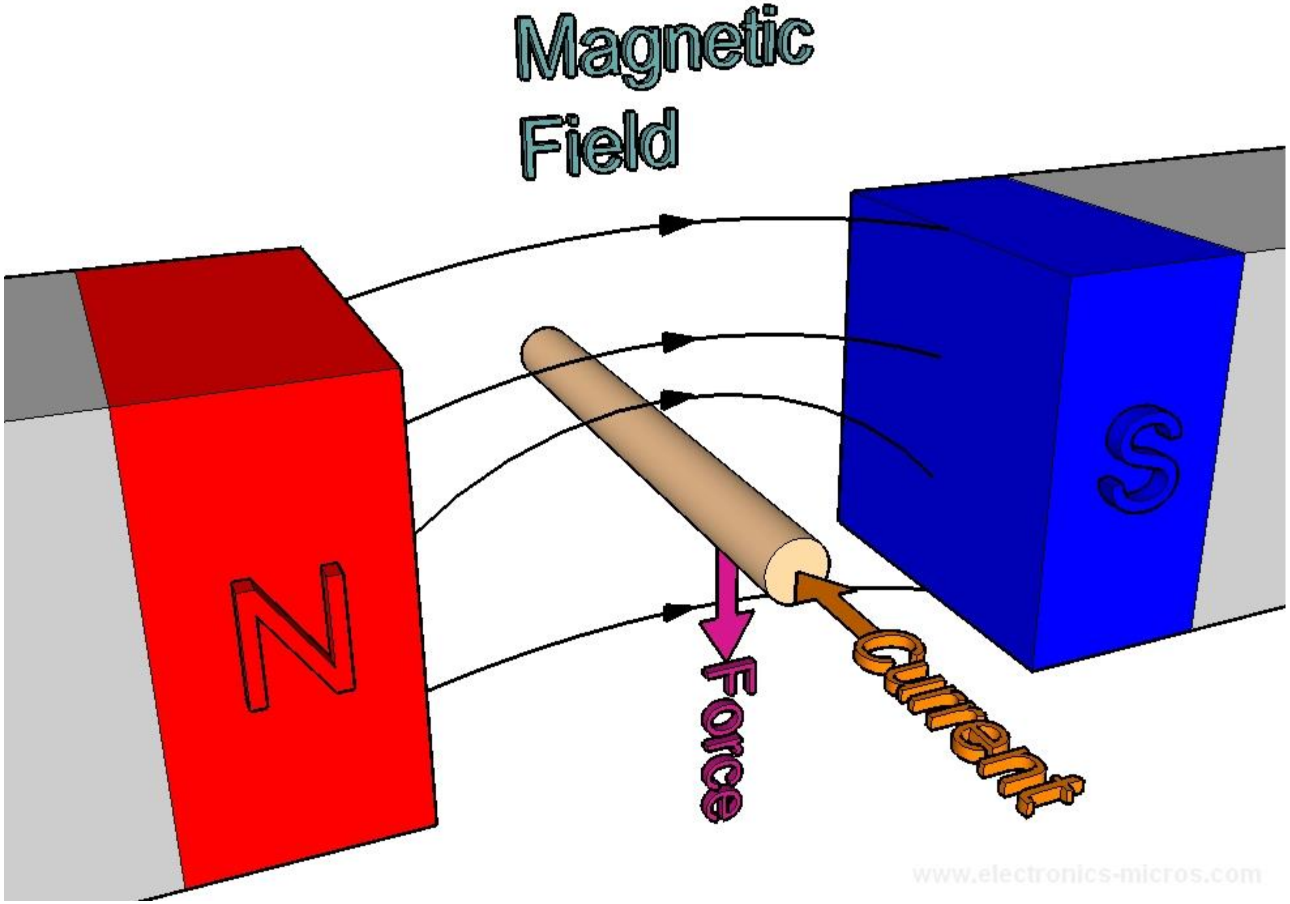
- How does a motor work (turning current into rotary motion)
- Motor performance (speed/torque curves and what they mean)
- Talon SRX overview
- Talon control modes
- Encoders
- Configuring an entire axis (Talon SRX + sensor + motor)
  - Open loop
  - Closed loop - PID
  - Tuning a Closed loop
    - Velocity
    - Position
    - Motion Magic

How does a motor work

The Right Hand Rule



How does a motor work



How does a motor work

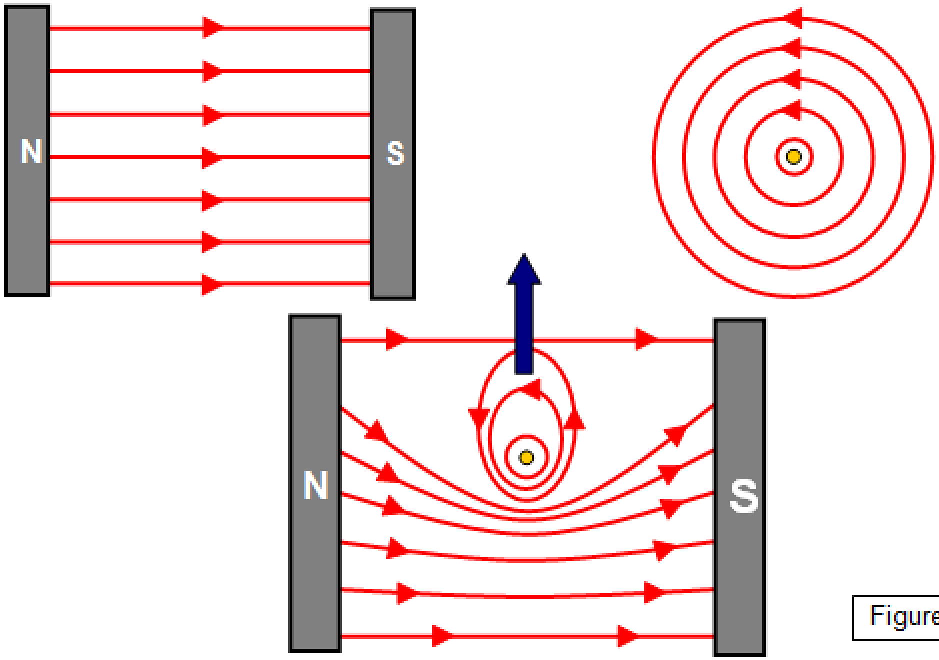
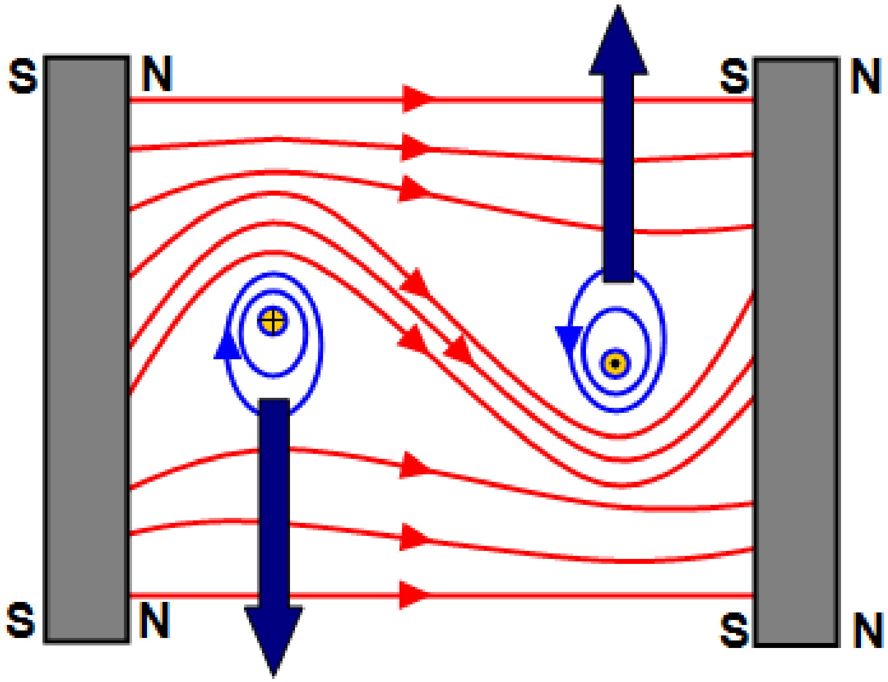
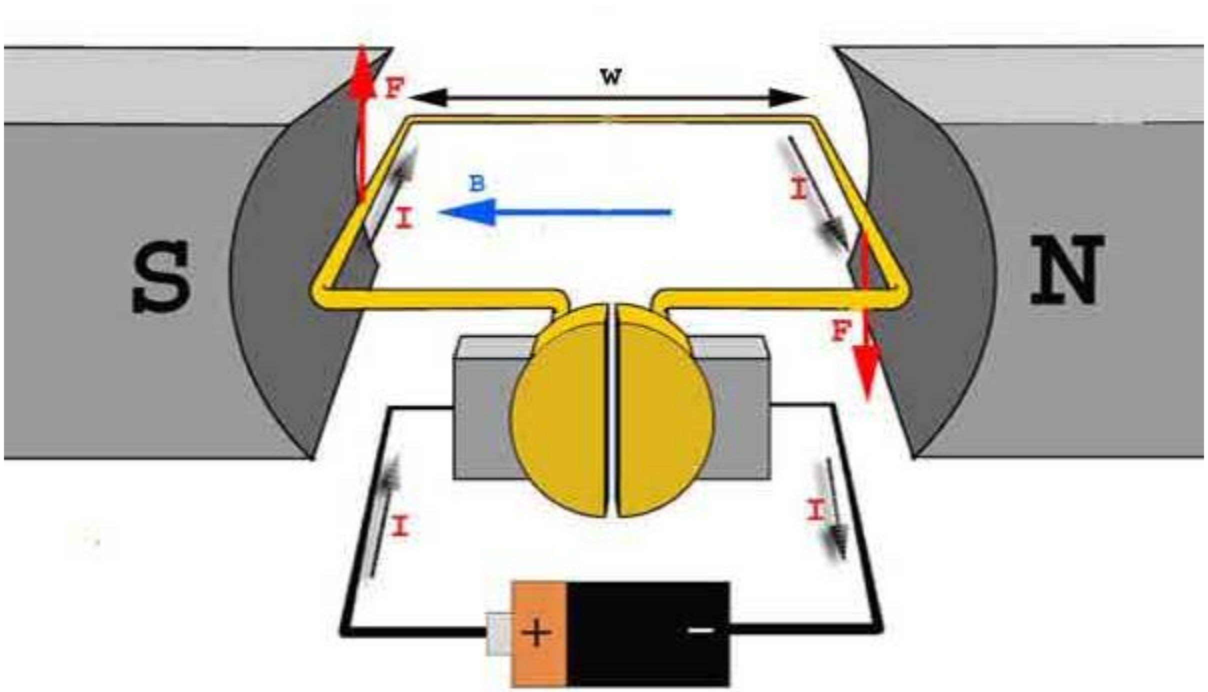


Figure 4

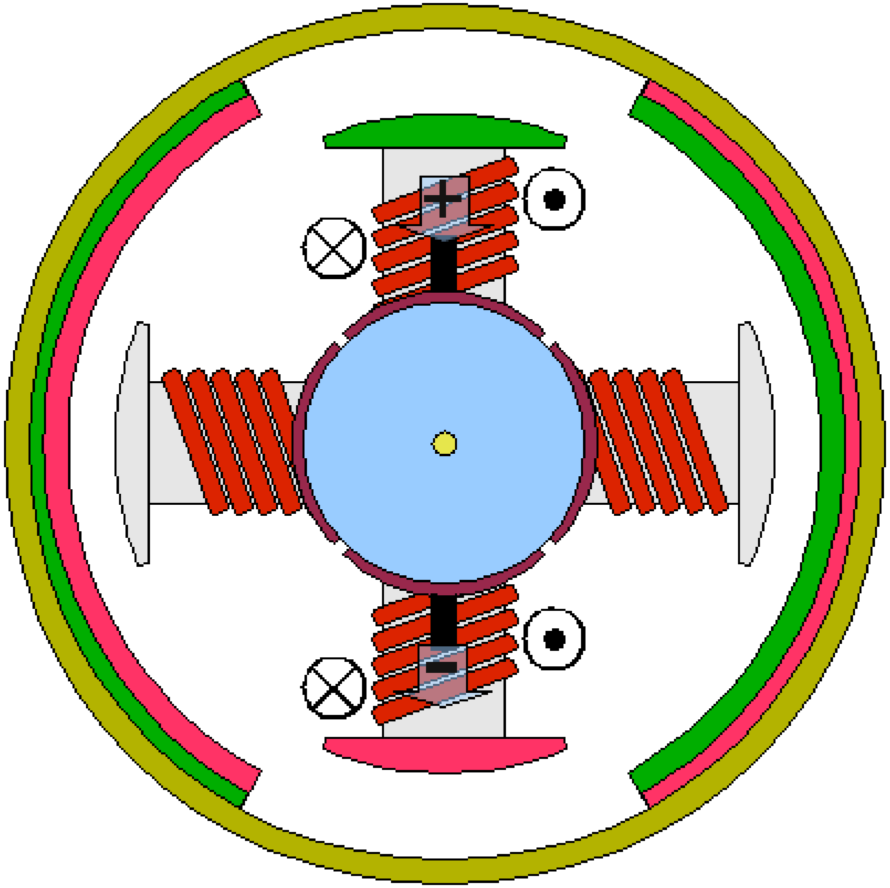
How does a motor work



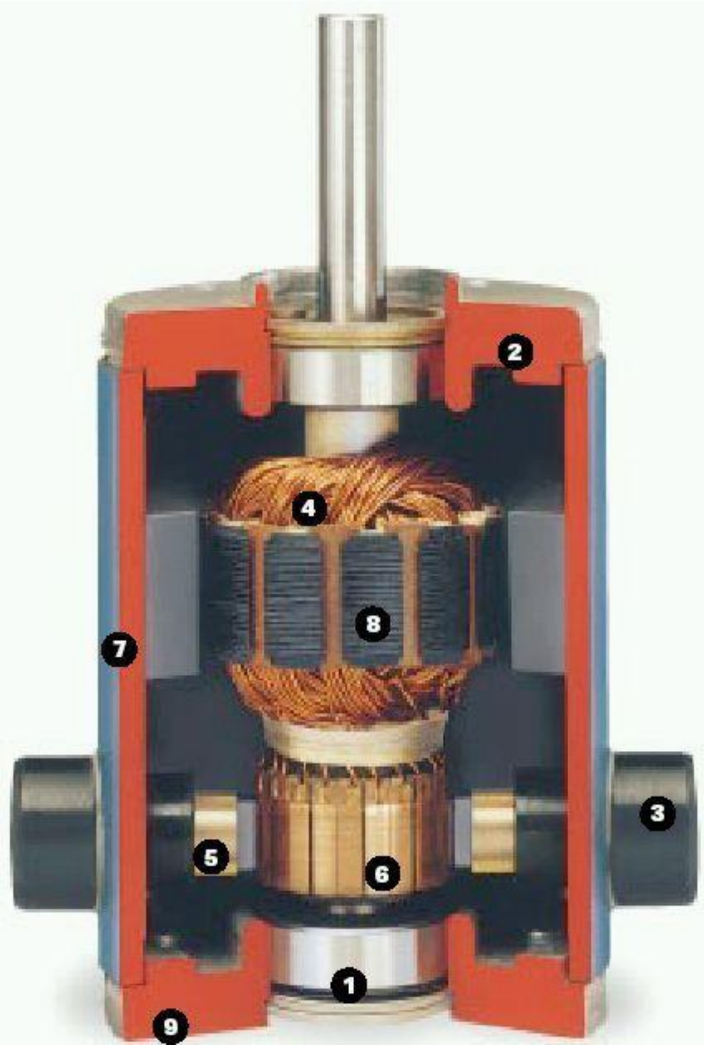
How does a motor work



How does a motor work

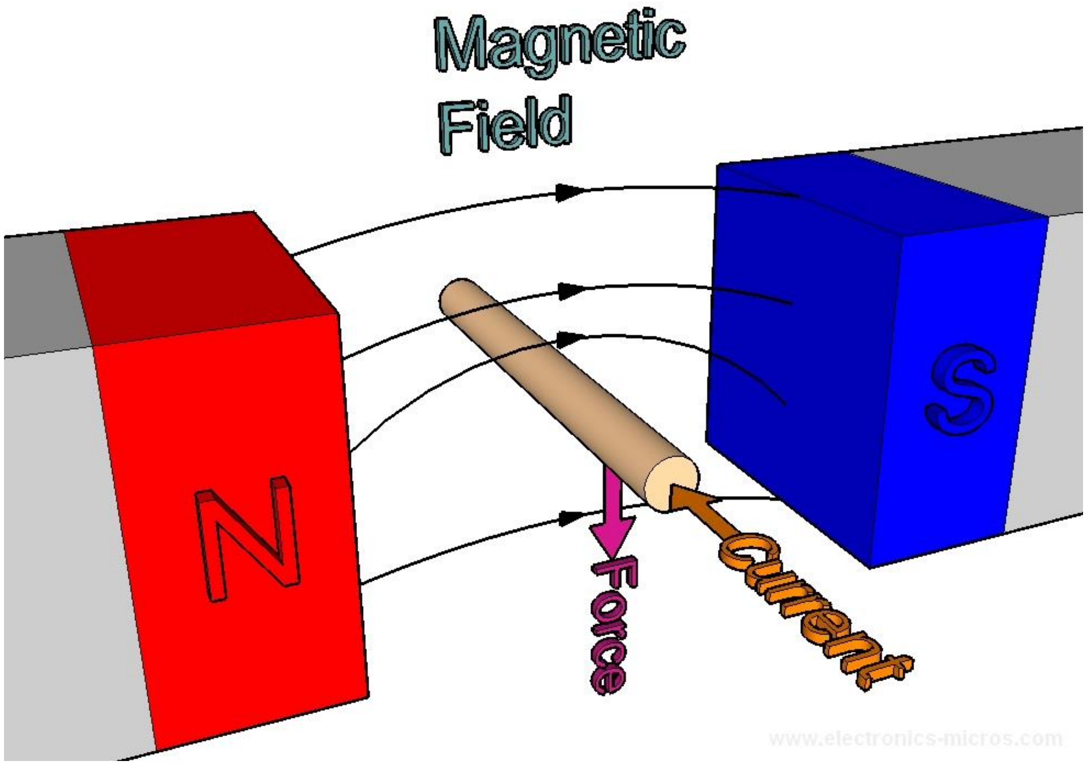


How does a motor work



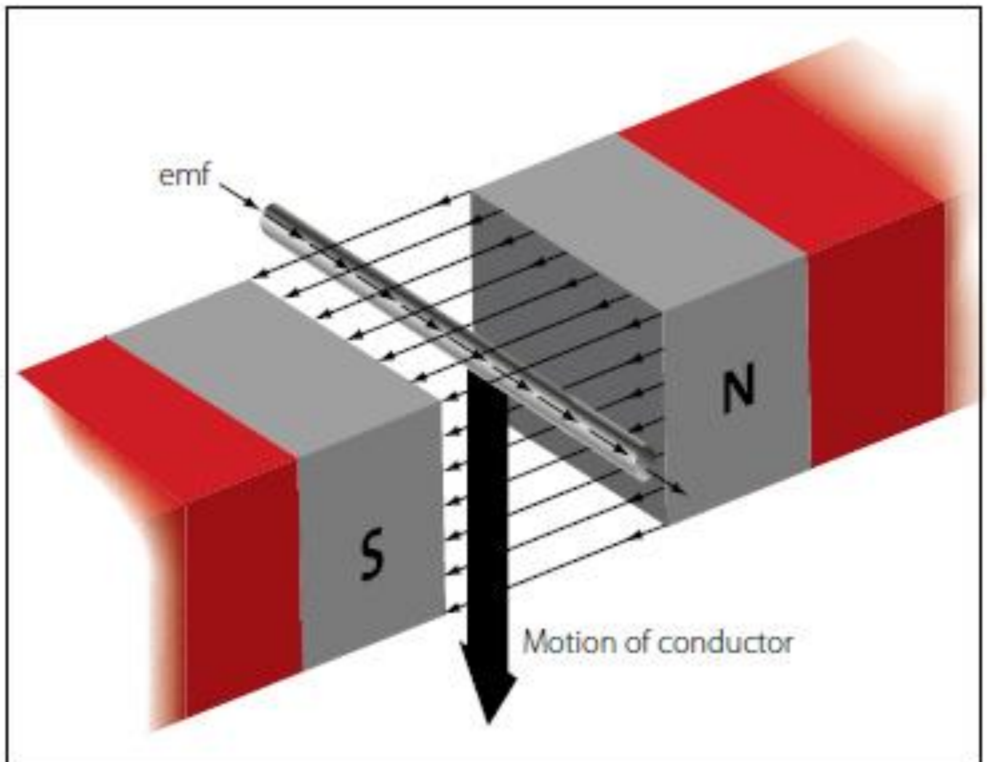
How does a motor work

This is how we make torque



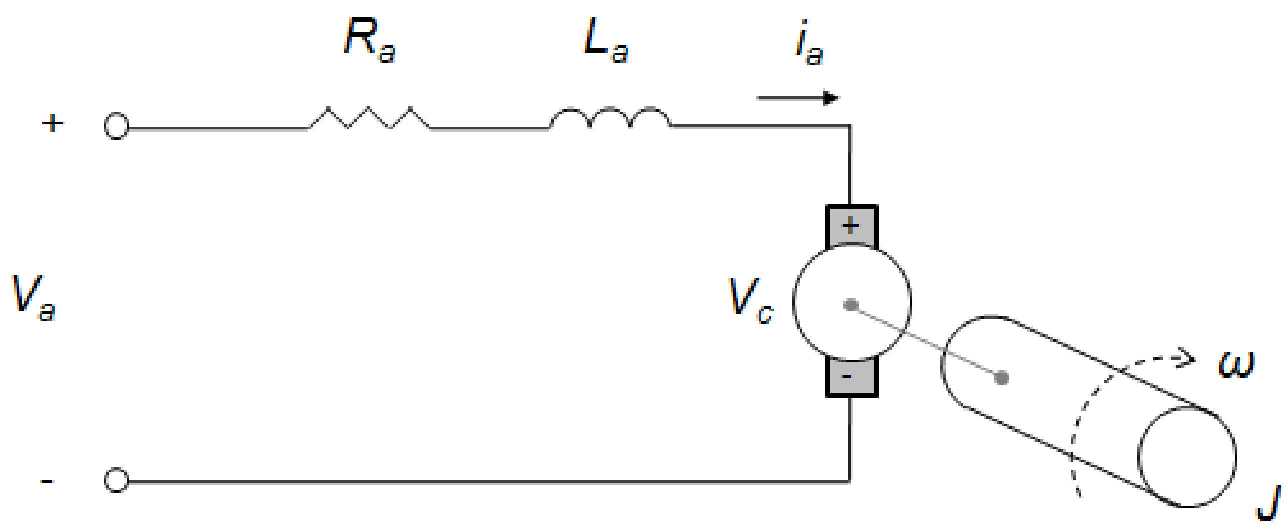
How does a generator work

What is “Back EMF”?



How does a motor work

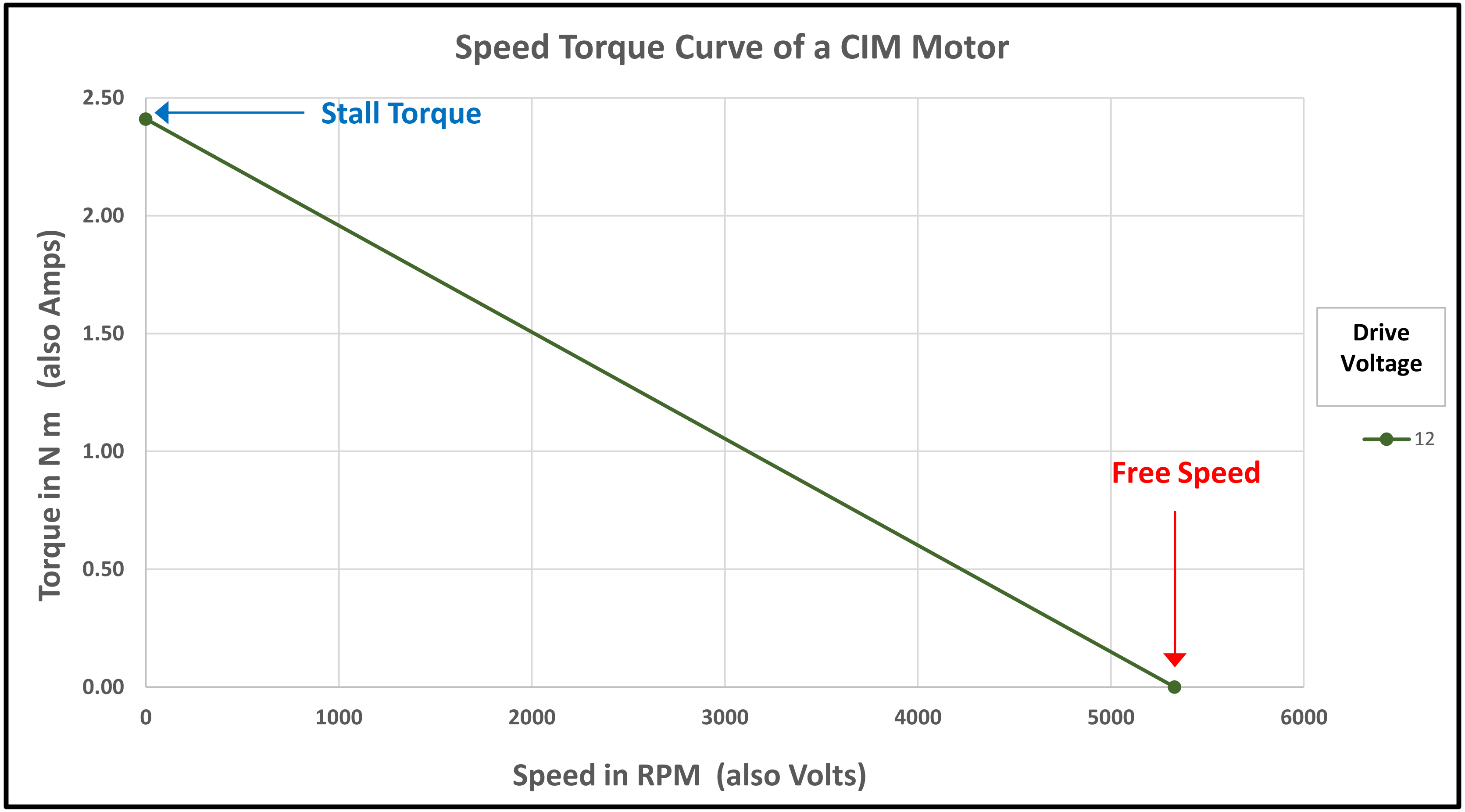
Electrical Schematic of a Motor



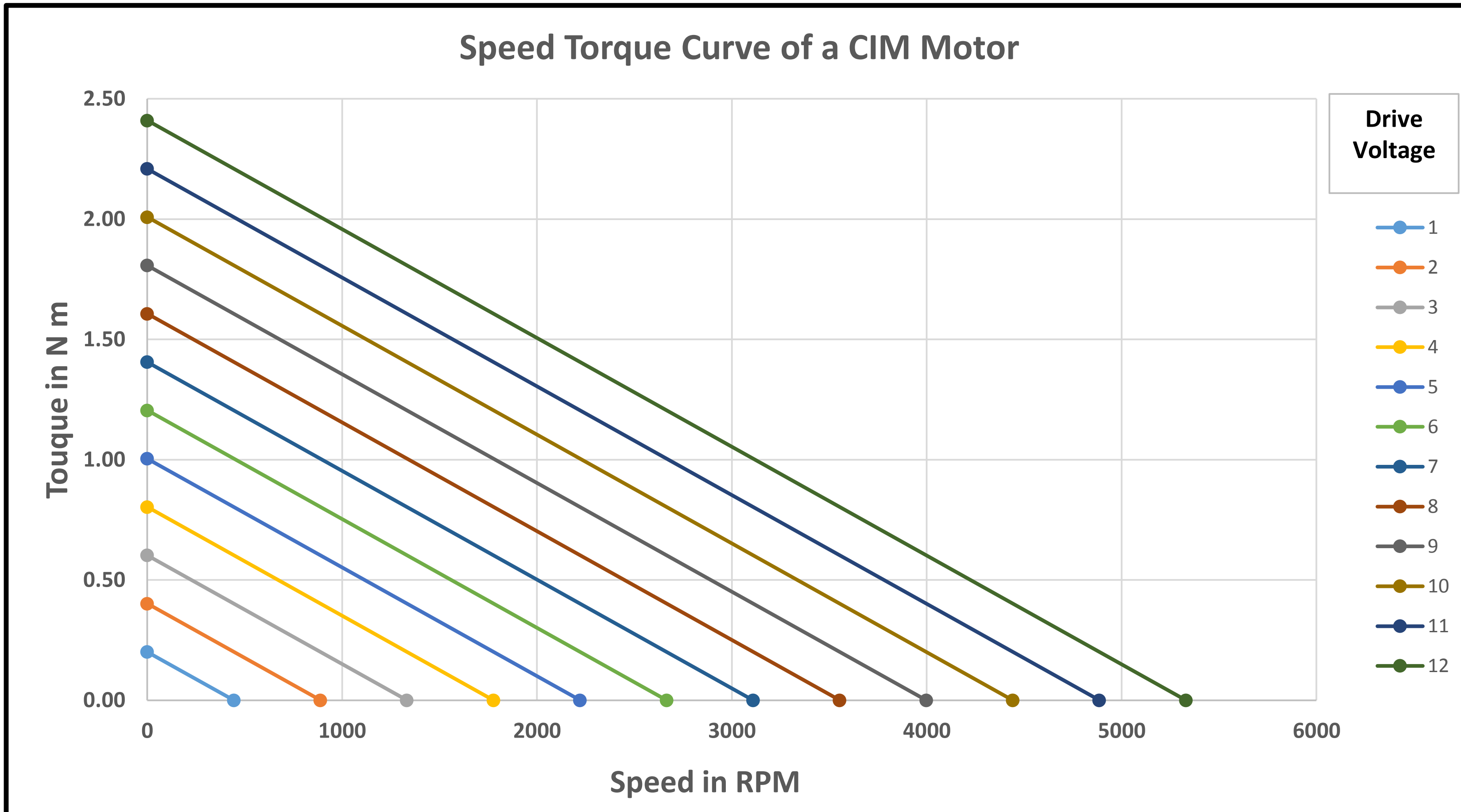
Volts = Speed

Amps = Torque

Motor Performance

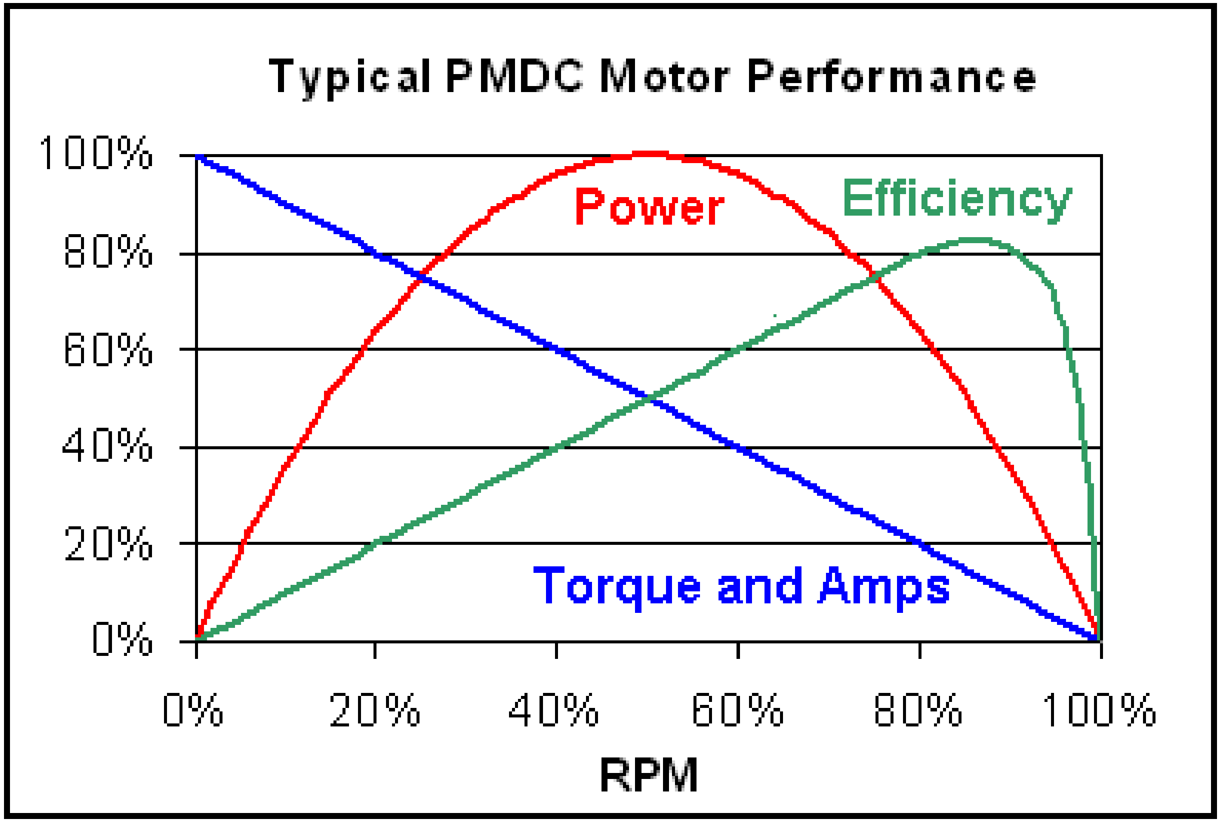


Motor Performance



Motor Performance

Speed Torque Curve



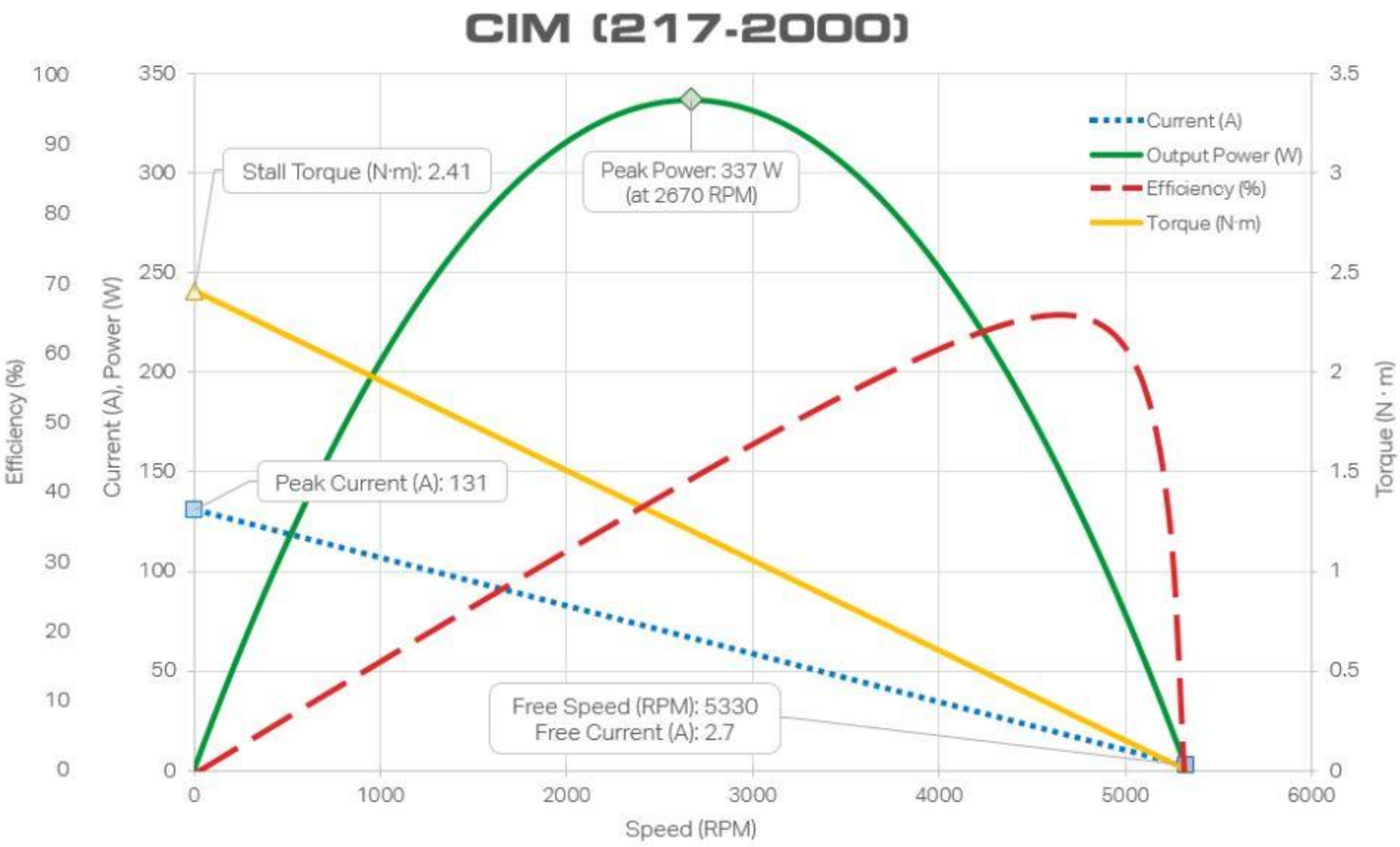
Motor Performance

CIM
Lots of copper
Low speed
Heavy
No fan

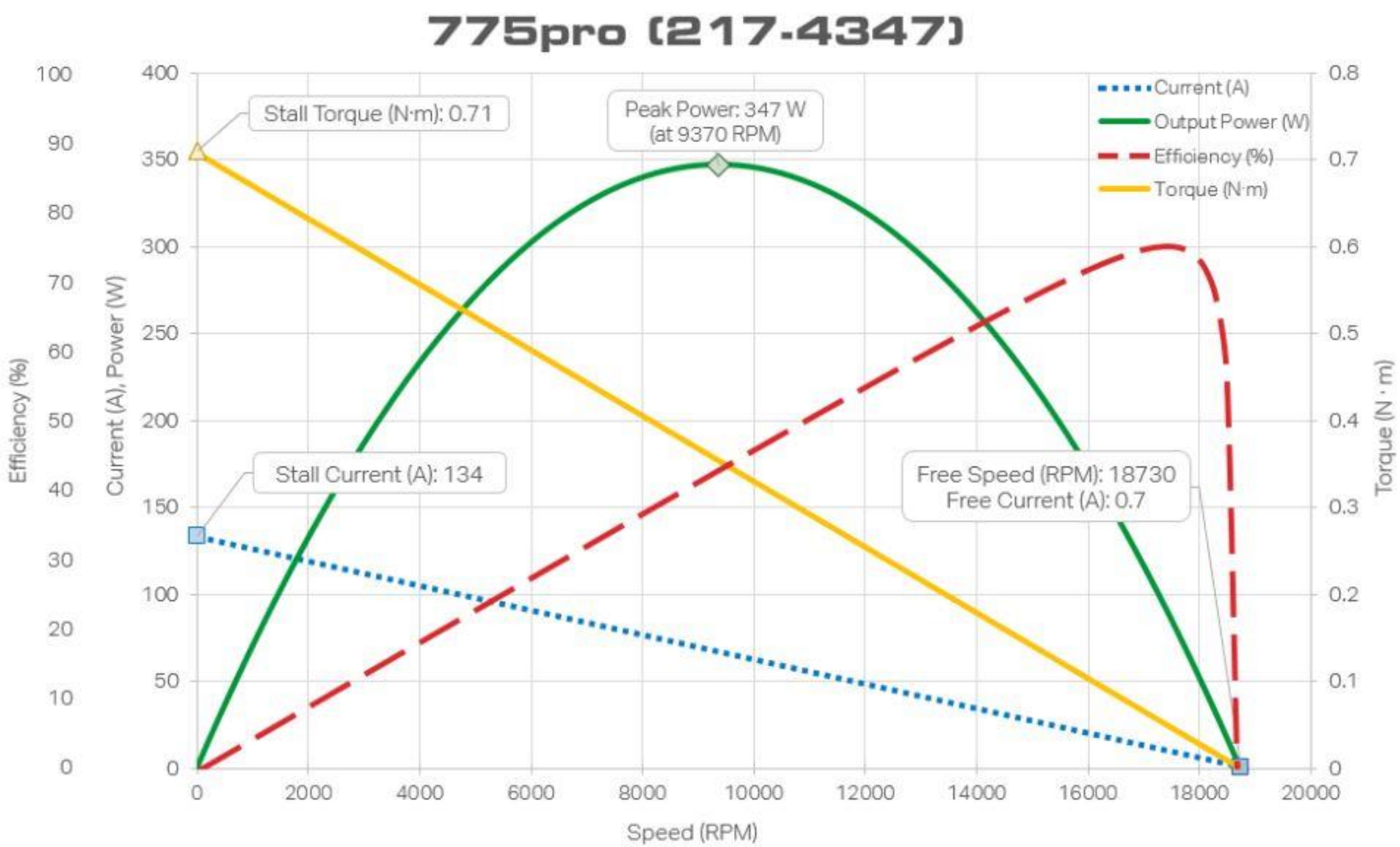
Bag
Less copper
Medium speed
Light
No fan

775
Less copper
High speed
Light
Fan

Motor Performance - CIM



Motor Performance - 775



Motor Performance

Motor	Price	Max Power	Free Speed	Stall Torque	Weight
CIM	$28	337 W	5,310 rpm	2.4 N-m	2.82 lbs
Mini-CIM	$25	230 W	6,200 rpm	1.4 N-m	2.16 lbs
BAG	$25	147 W	13,100 rpm	.43 N-m	.71 lb
RS-775pro	$18	347 W	18,700 rpm	.71 N-m	.74 lb
RS-550	$7	253 W	19,300 rpm	.48 N-m	.48 lb
AM-9015	$14	179 W	16,000 rpm	.43 N-m	.5 lb
Snow Blower Motor	$39	20 W	100 rpm	7.9 N-m	1.1 lbs

Motor Performance

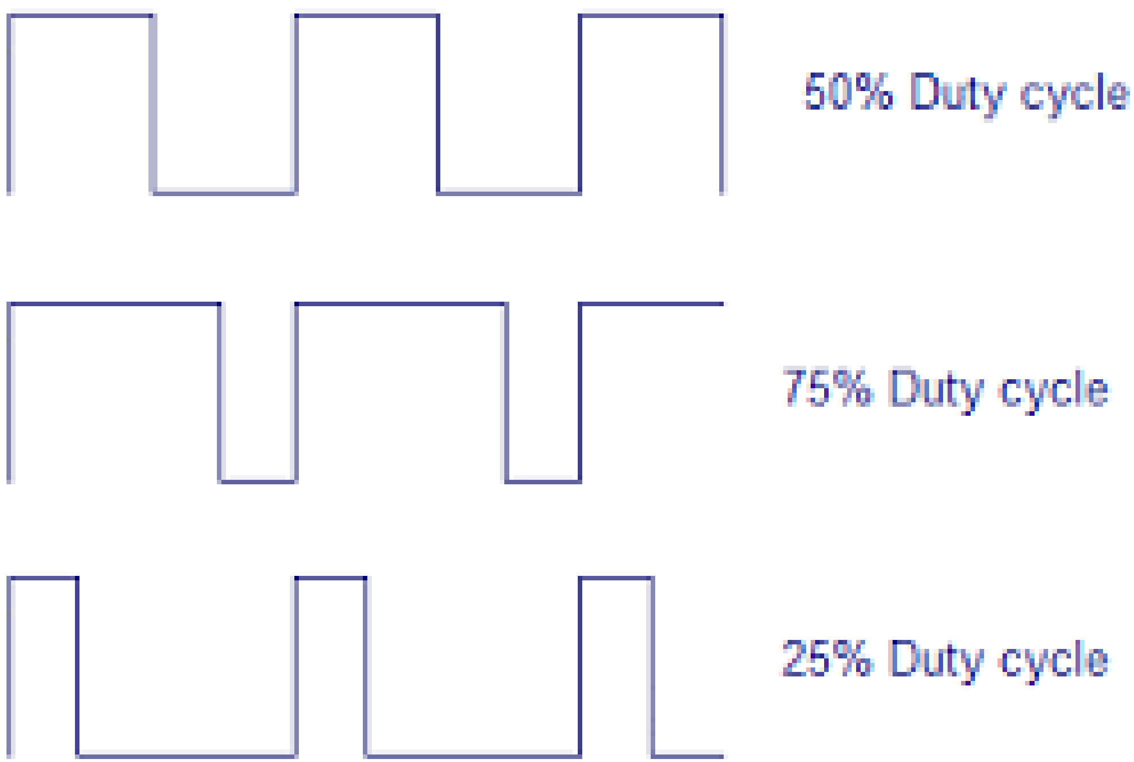
Gearbox Options (change the speed / torque relationship)

Motor Performance

Heat!

Motor Performance - PWM

ON = 12 volts **OFF** = 0 volts



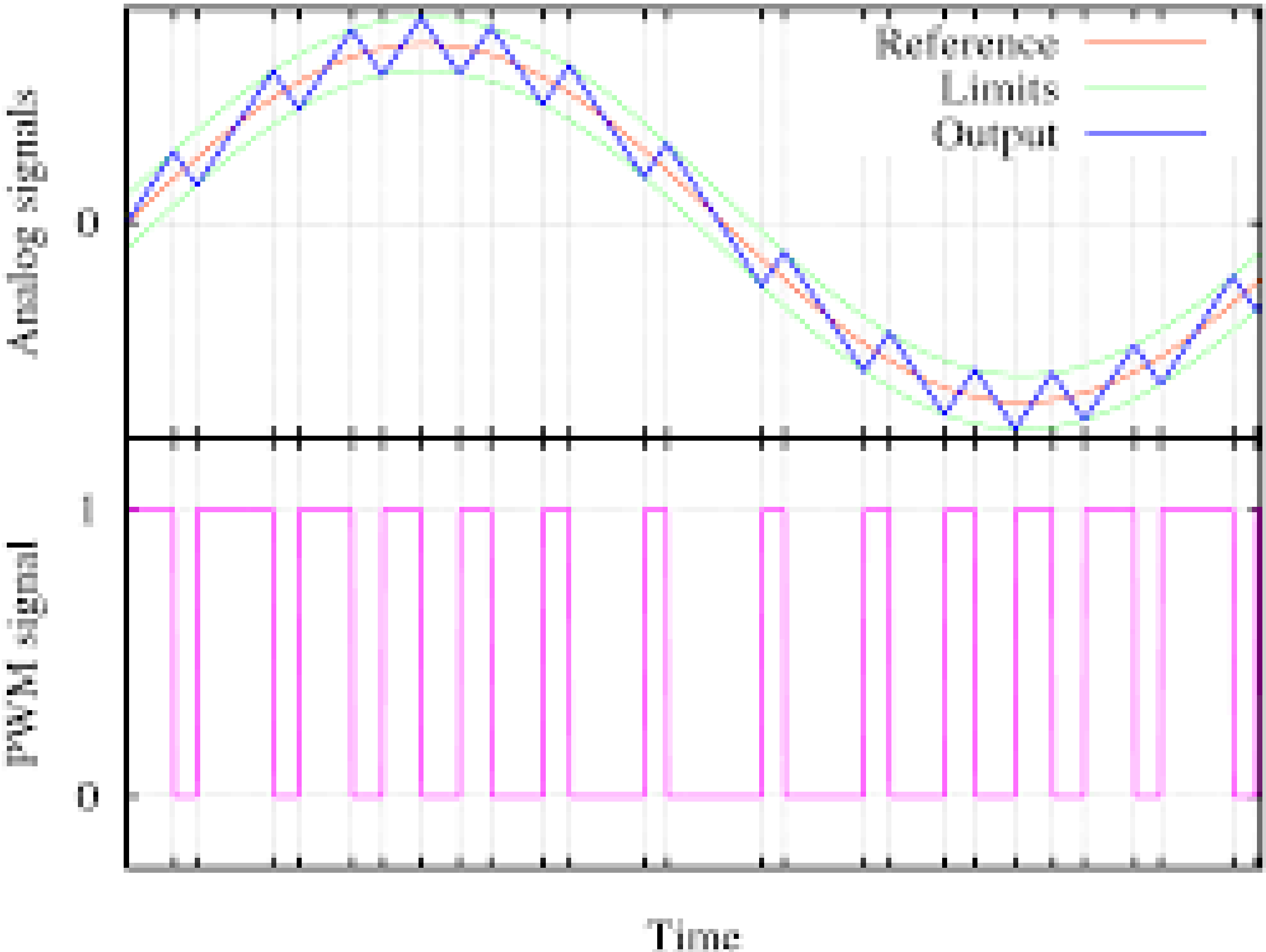
= 6 volts

= 9 volts

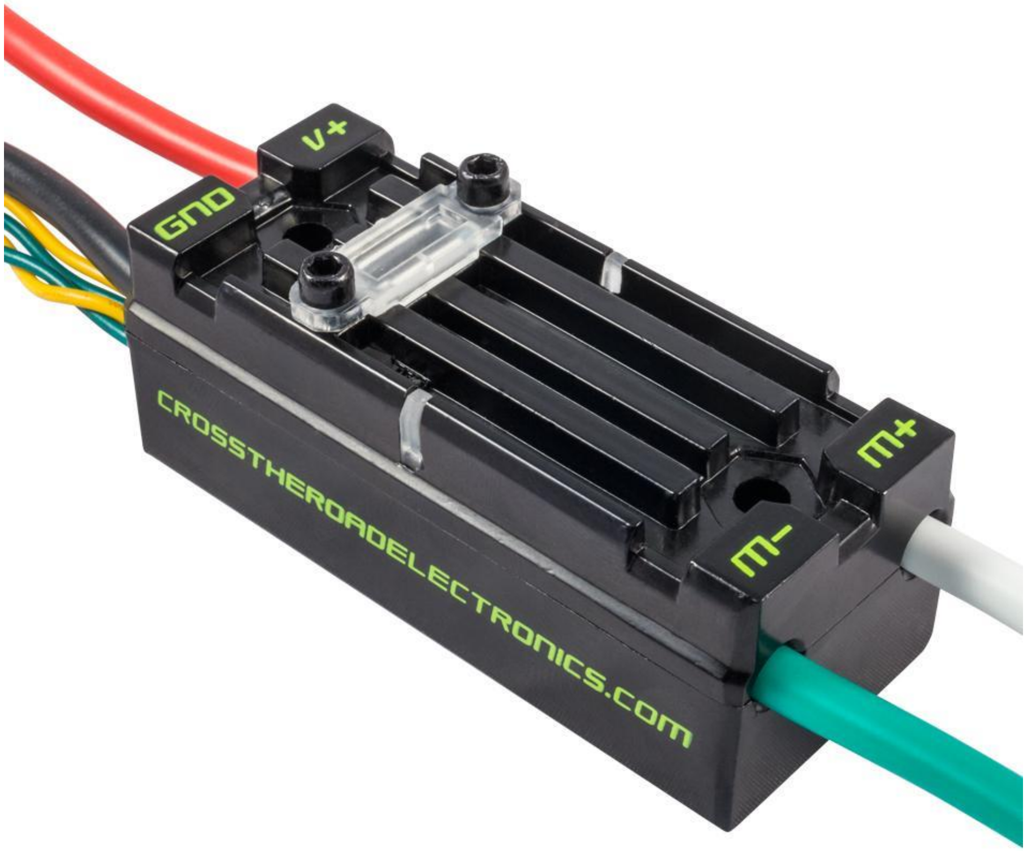
= 3 volts

Pulse **W**idth **M**odulation

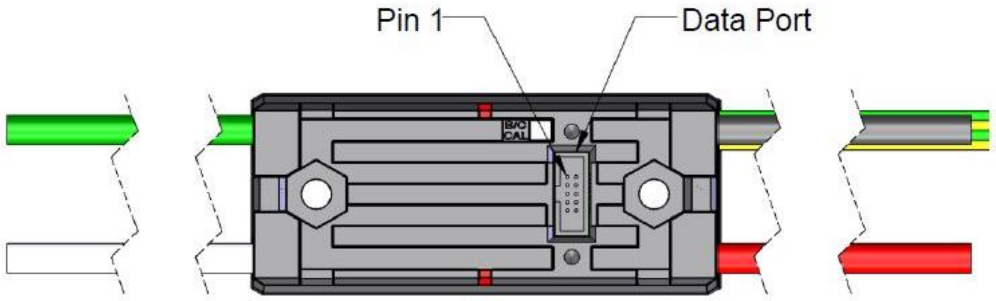
Motor Performance - PWM

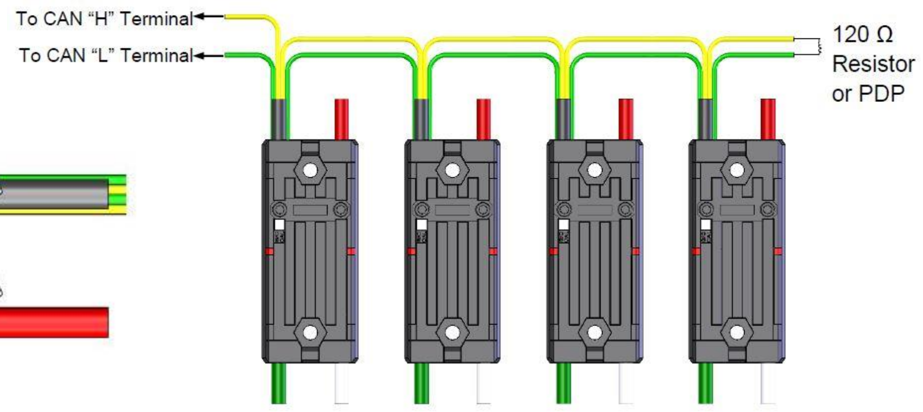


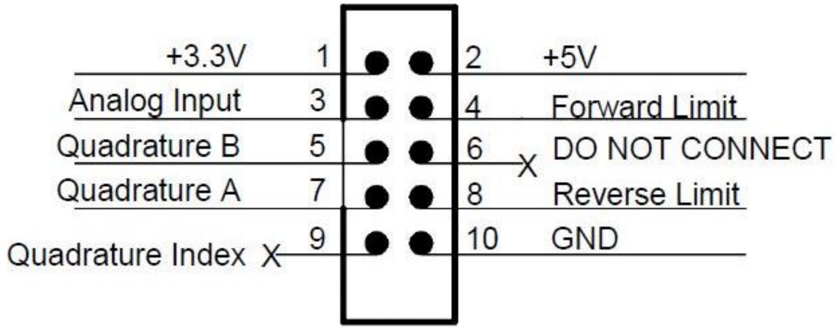
Talon SRX



Talon SRX Wiring





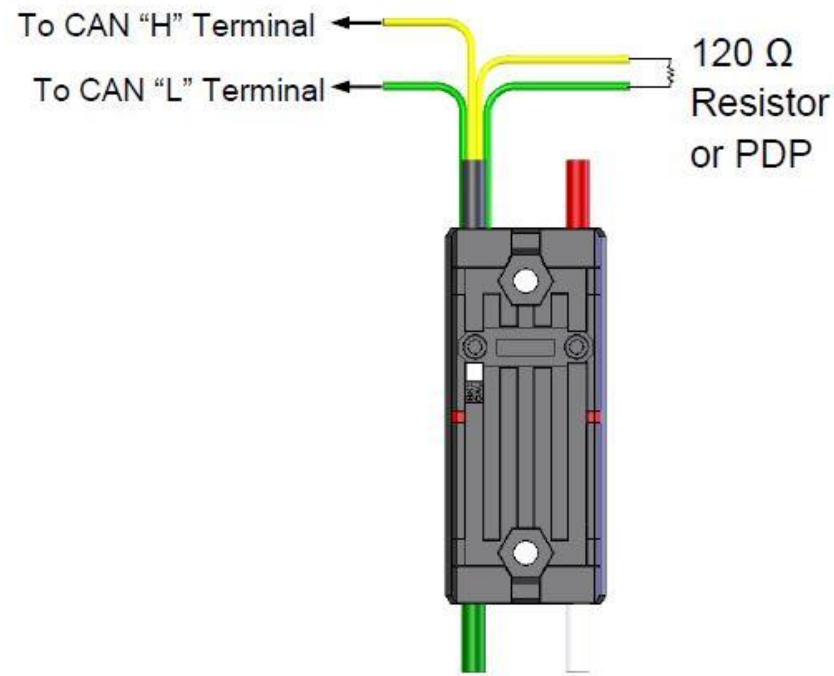


Data Port Pinout

Talon SRX Communications

- PWM
- CAN Bus
    - Frames
        - Command – 10mS
        - General – 20mS
        - Feedback – 20mS
        - Quadrature – 100mS
        - Analog – 100mS

Frame rates are programable
Watch out for CAN bus overload



Talon SRX Blink Codes

Blink Codes During Normal Operation		
LEDs	**Colors**	**Talon SRX State**
Both	Blinking Green	Forward throttle is applied. Blink rate is proportional to Duty Cycle
Both	Blinking Red	Reverse throttle is applied. Blink rate is proportional to Duty Cycle
None	None	No Power is being applied to Talon SRX
LEDs Alternate[1]	Off/Orange	CAN bus detected, robot disabled
LEDs Alternate[1]	Off/Red	CAN bus/PWM is not detected
LEDs Alternate[1]	Red/Orange	Damaged Hardware
LEDs Strobe "towards" (M+)[2]	Off/Red	Forward Limit Switch or Forward Soft Limit
LEDs Strobe "towards" (M-)[2]	Off/Red	Reverse Limit Switch or Reverse Soft Limit
LED1 Only "closest" to M+/V+	Green/Orange	In Boot-loader
Both	Solid Orange	Neutral throttle is applied. Throttle is zero or is within dead band.

Talon SRX Versions

- Hardware
  - 1.4   3.3 vs 5.0 volt Quadrature pins (Qb & Qi are limited to 3.3)
  - 1.7

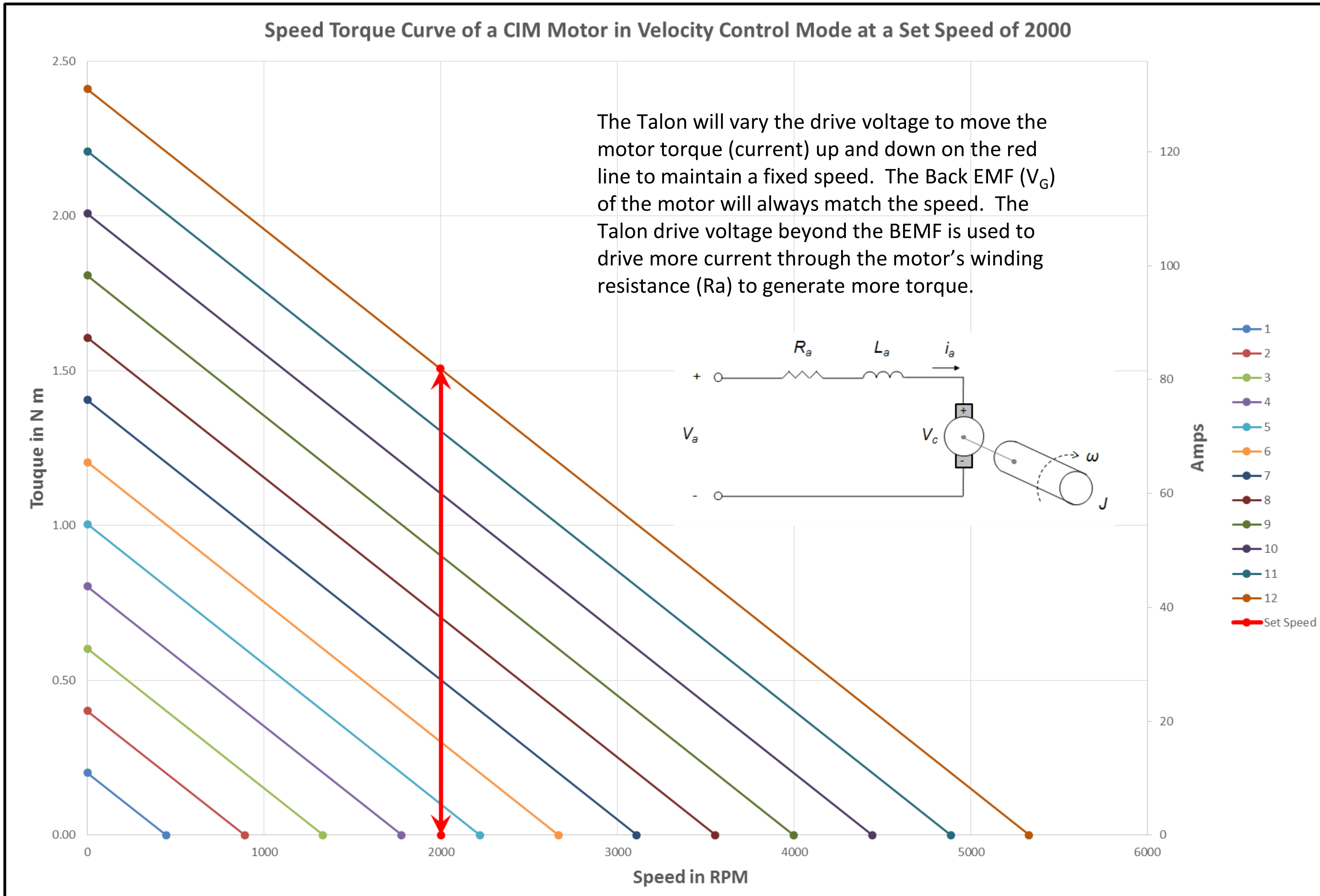
- Firmware
  - X.XX (2018)
    - Remote sensor CAN connectivity
    - 4 PID slots
  - 2.34 (2017)
  - 2.00 (2016)
  - older

Talon SRX Control Modes

- Duty Cycle (-1 to 0 to +1)
- Voltage (-12 to 0 to +12)
- Velocity
- Position
- Current Control
- Slave

- Motion Control
- Motion Magic

Control – Velocity Mode



Talon SRX Parameters

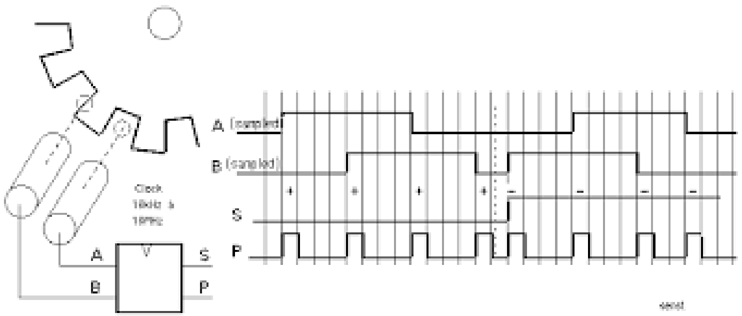
- P
- I
- D
- FF
- Izone
- Max voltage out (+&-)
- Nominal voltage (+&-)
- Min Acceptable Error
- Voltage Ramp
- Voltage Ramp Closed loop
- Current limit
- Hard limits
  - NO / NC
- Soft limits
  - Value
- Reverse Encoder
- Reverse Output
- Encoder Scaling
- Velocity Filtering

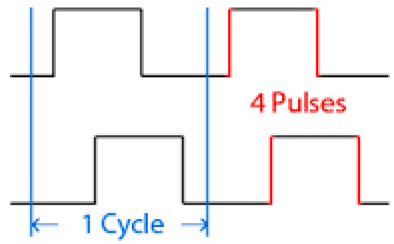
Talon SRX Sensors

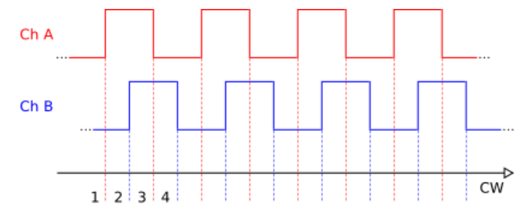
- Quadrature Encoder
  - US Digital
  - CIMcoder
  - AM Mag
  - CTR magcoder
- Analog Encoder
  - Rotary potentiometer
  - String potentiometer
- Limit Switch
  - Forward
  - Reverse
  - Soft limits
- Other
  - DC Voltage
  - Frequency
  - Period
  - Duty Cycle

Encoders - Quadrature

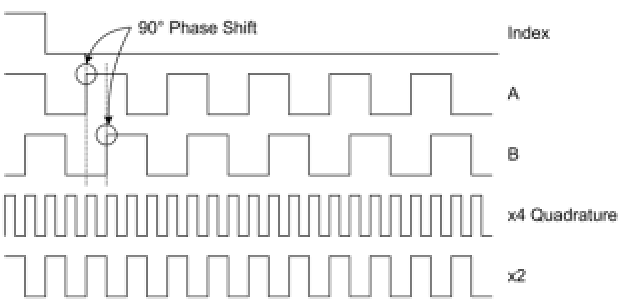
How do they work?

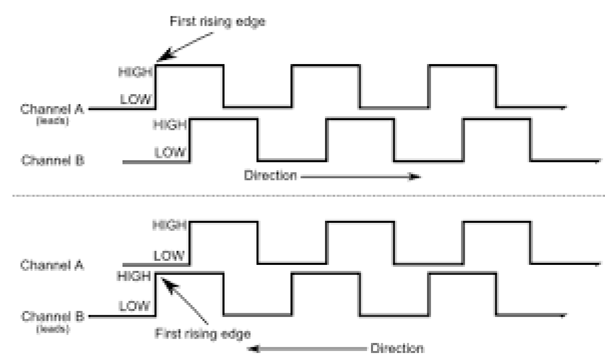




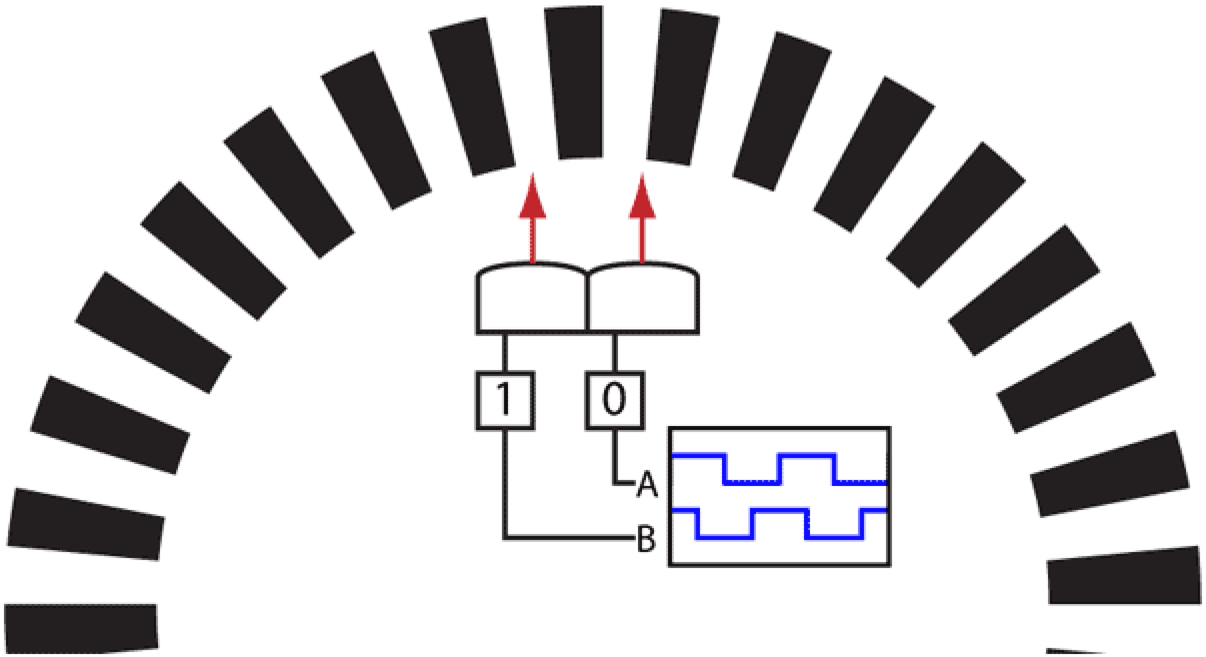


Old A	Old B	New A	New B
0	0	1	0
1	0	1	1
1	1	0	1
0	1	0	0





Encoders - Quadrature

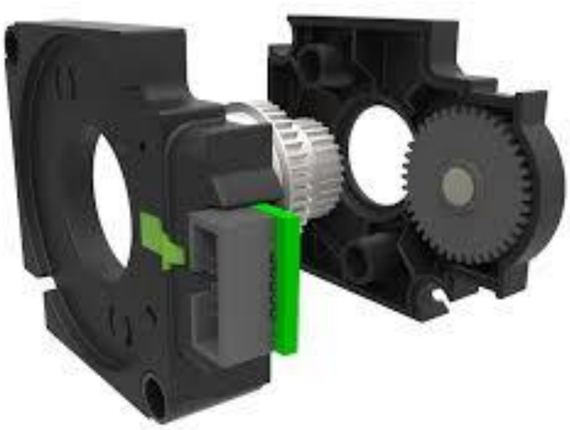


Encoders - Quadrature

- CIMcoder
  - Magnetic
  - 20 cycles per rev
  - 80 counts per rev

Encoders - Quadrature

- CTR Magcoder
  - Magnetic
  - 1024 cycles per rev
  - 4096 counts per rev
  - Includes Absolute!



Encoders - Quadrature

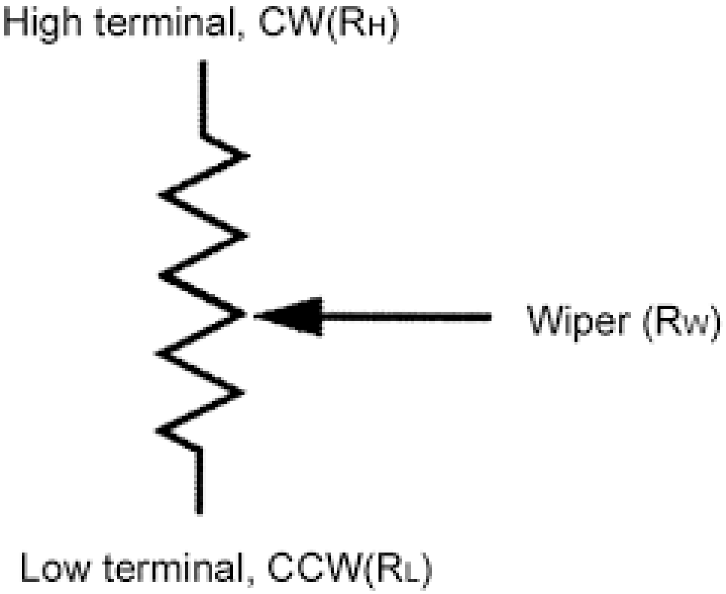
- US Digital E4T
  - Optical
  - 360 cycles per rev
  - 1440 counts per rev

Encoders - Quadrature

- Andy Mark
  - Magnetic
  - 7 cycles per rev
  - 28 counts per rev

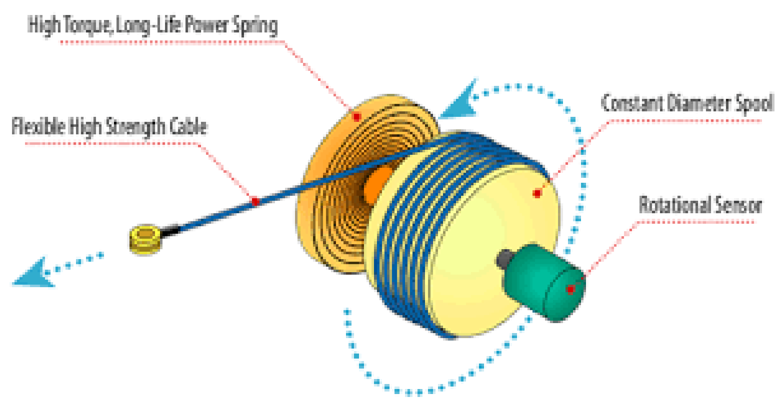
Encoders – Rotary Pots

- Several Manufactures
  - Analog
  - Need an ATD
  - Absolute



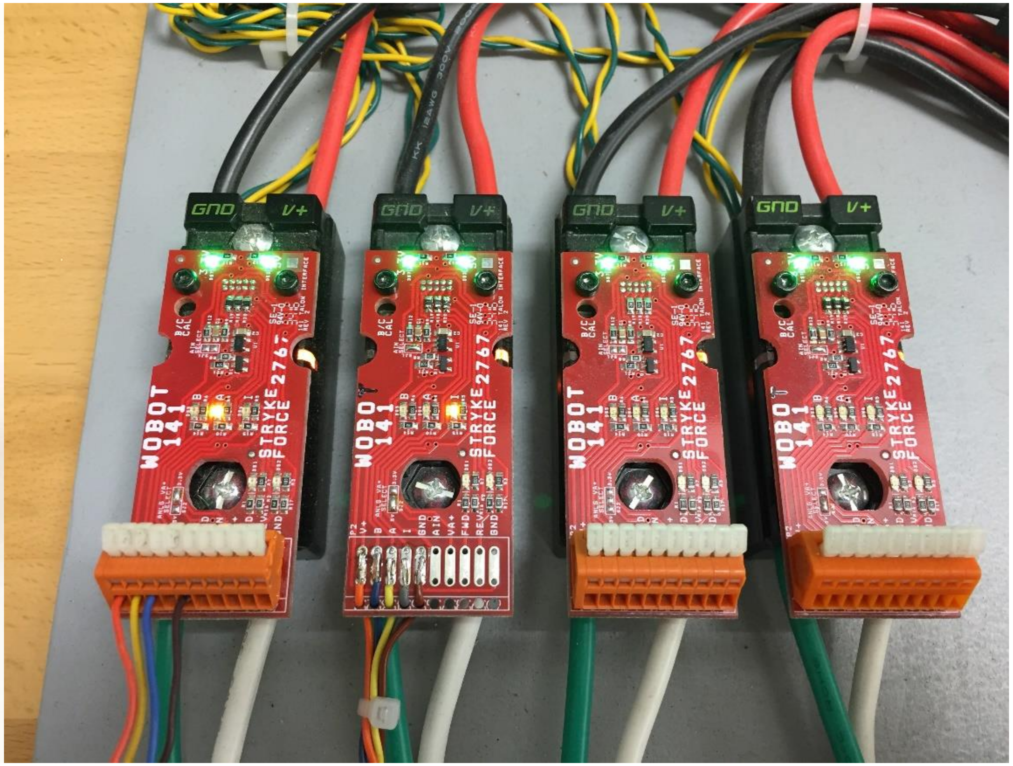
Encoders – String Pots

- Several Manufactures
  - Analog
  - Need an ATD
  - Absolute



Encoders – Sentinel Interface

The Sentinel puts an end to any and all interface issues

***Shameless plug** … $18 and in stock at **Andy Mark**!*

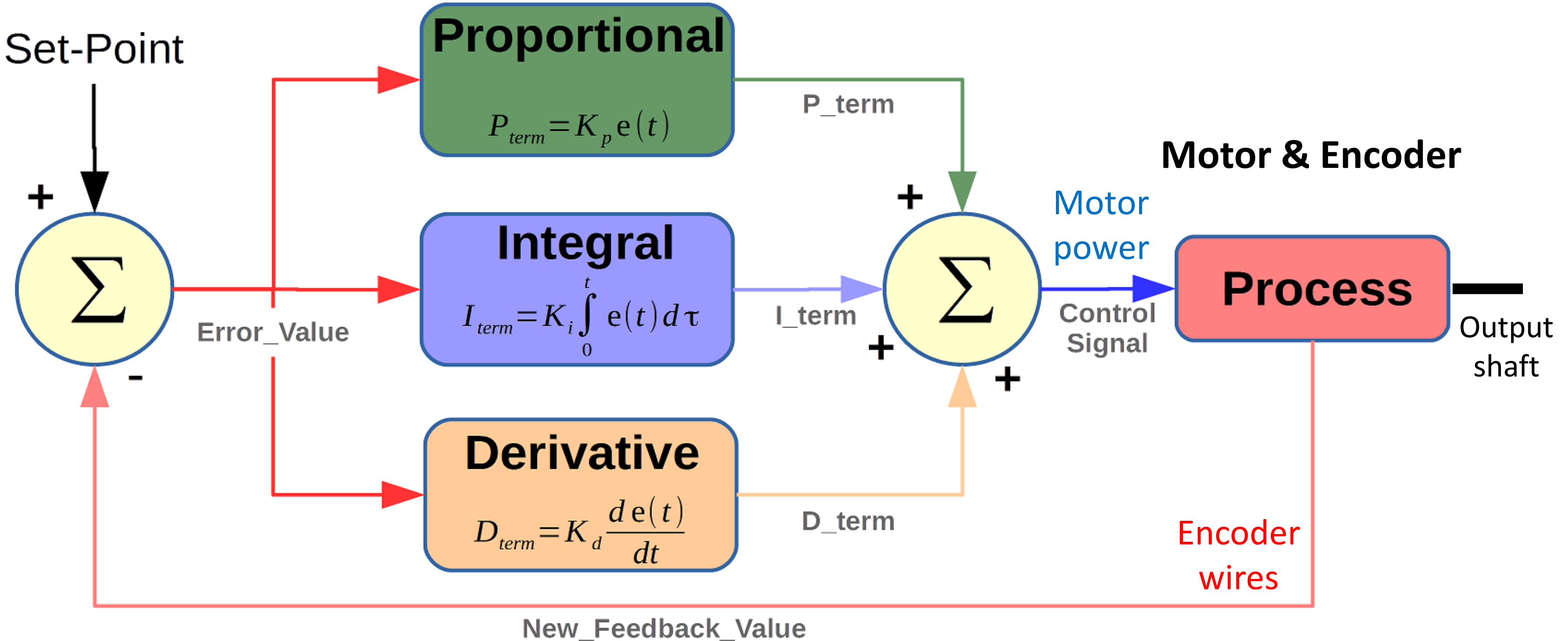
Control

- Open Loop
  - Fixed voltage – Speed changes with load
    - Duty Cycle (-1 to 0 to +1)
    - Voltage (-12 to 0 to +12)

- Closed Loop (add an encoder)
  - Variable voltage – Speed stays constant with load
    - Speed
  - Variable voltage – Position stays constant with load
    - Position & Motion Magic
  - Variable voltage – Torque stays constant with load (somewhat)
    - Constant Current

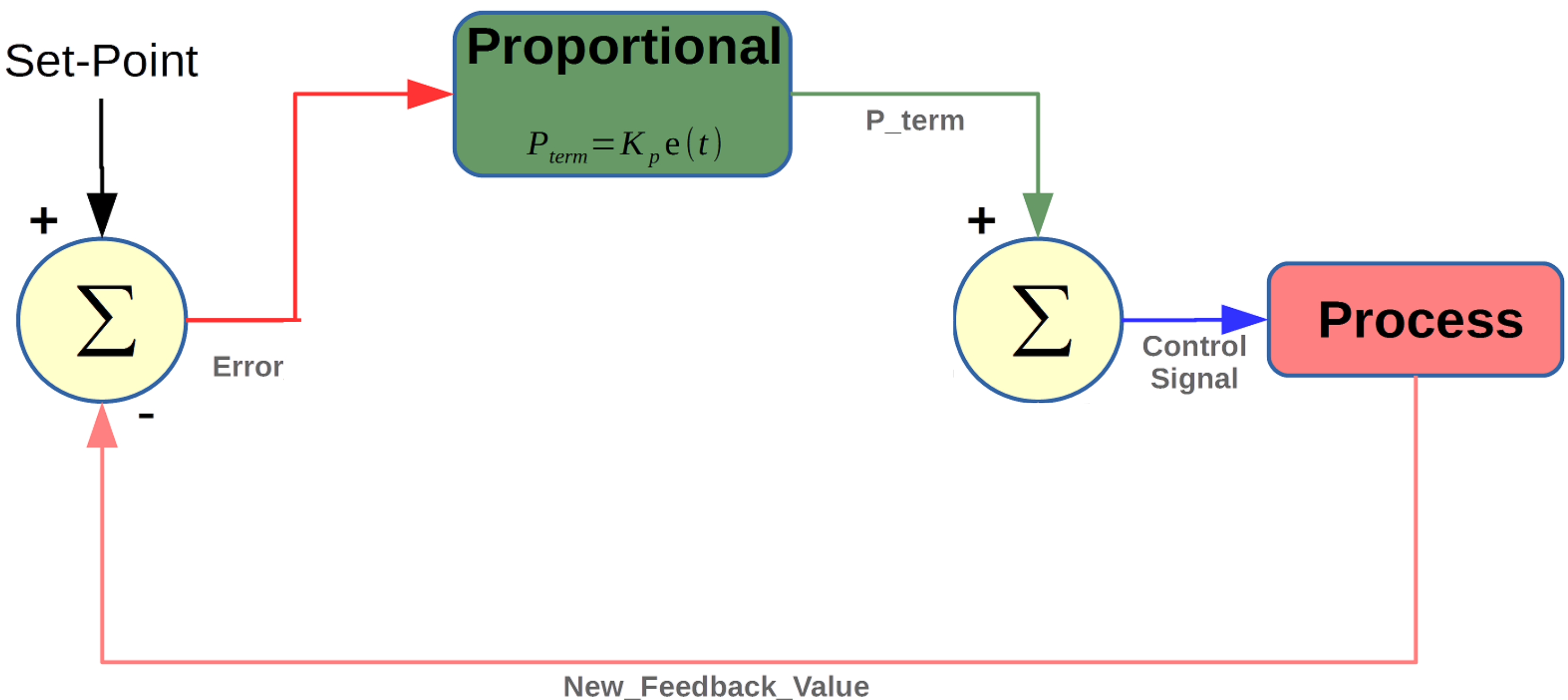
Control – PID

What on earth is PID?

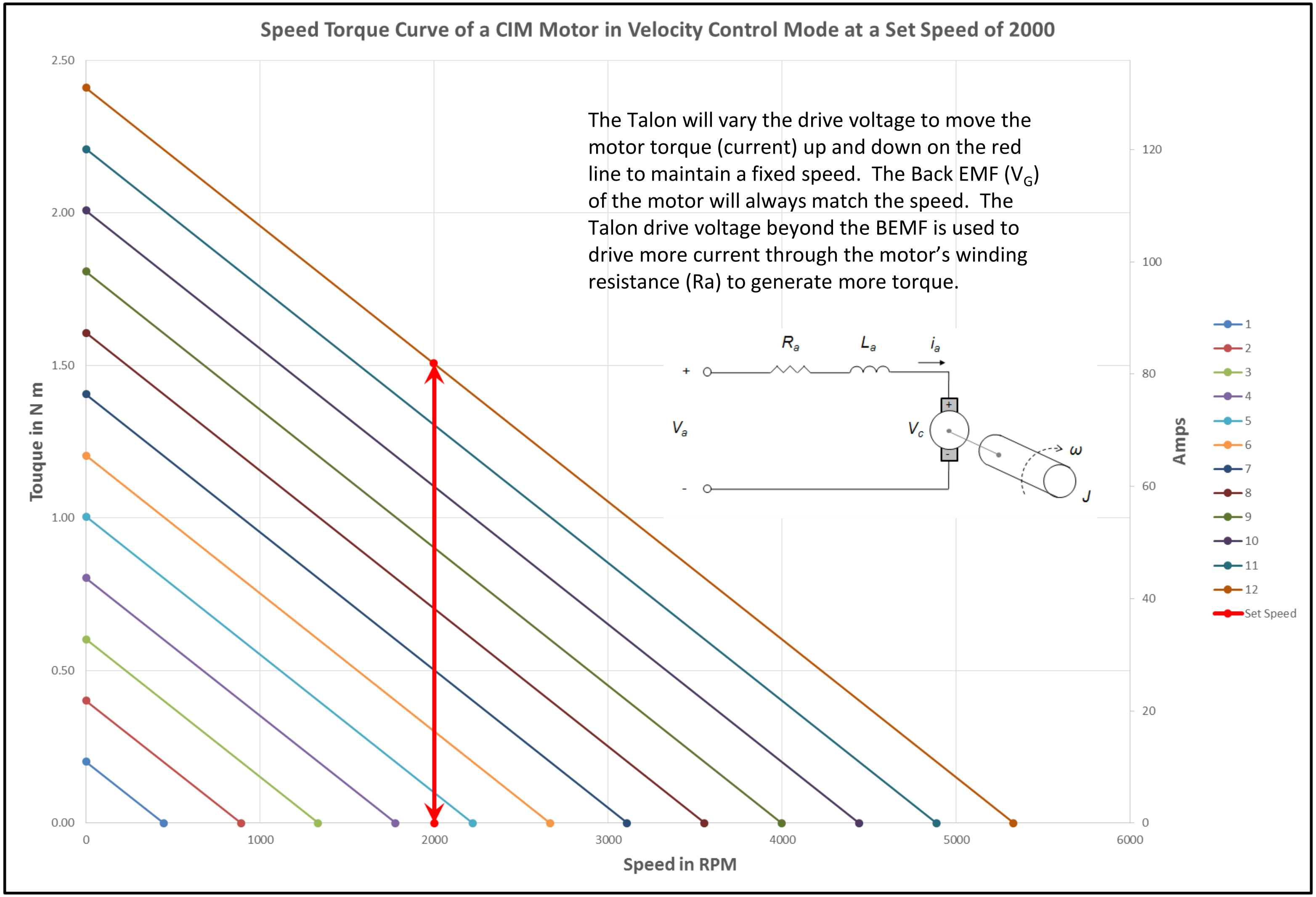


Control – P

Lets start with P

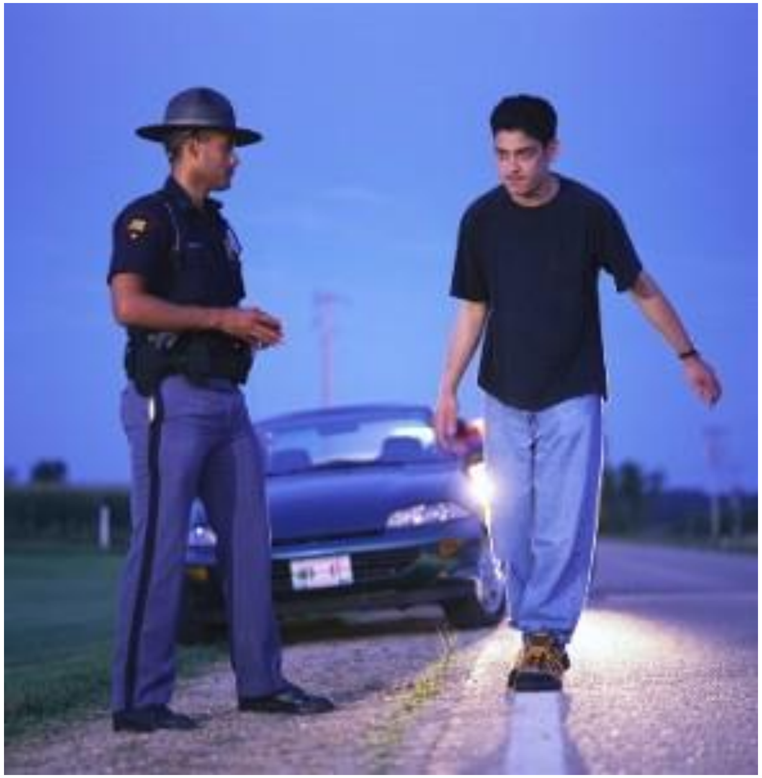


Control – Velocity Mode



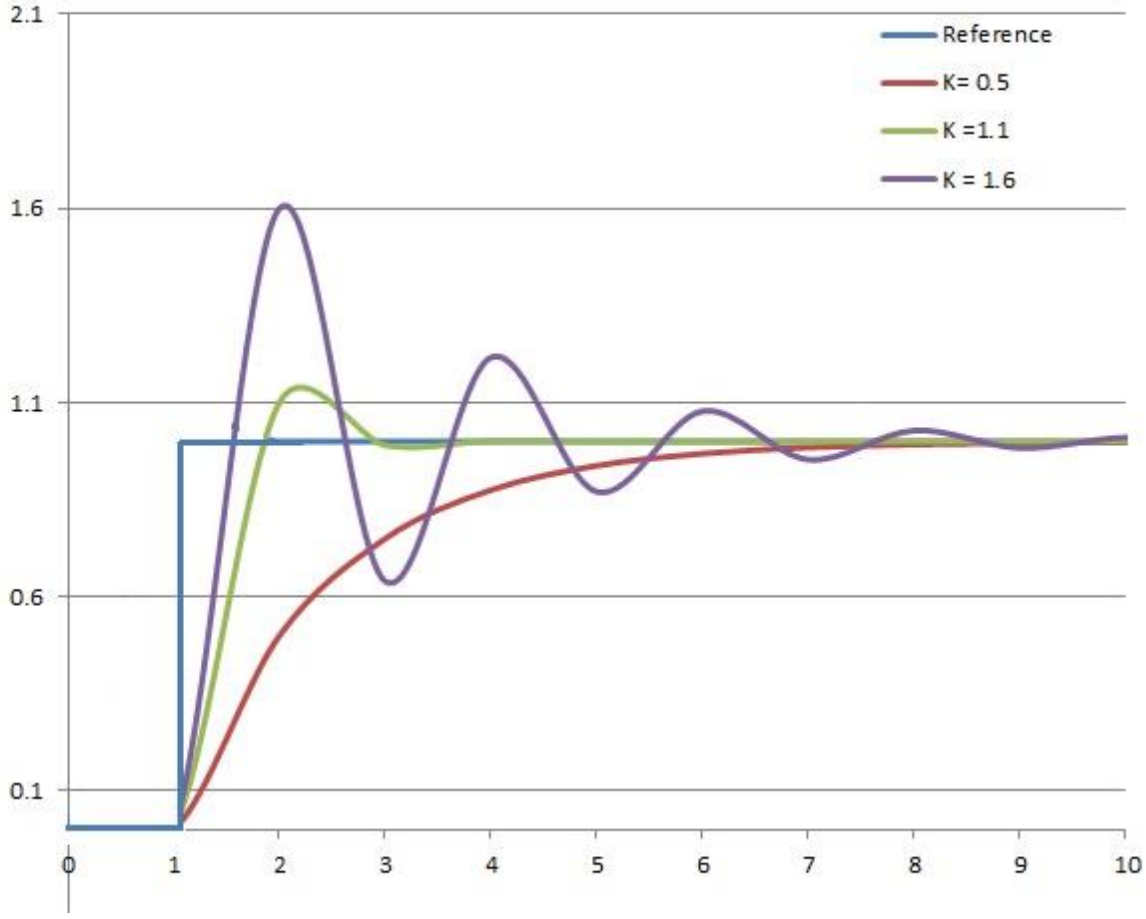
Control – P only

Walk the line … a real live control loop!



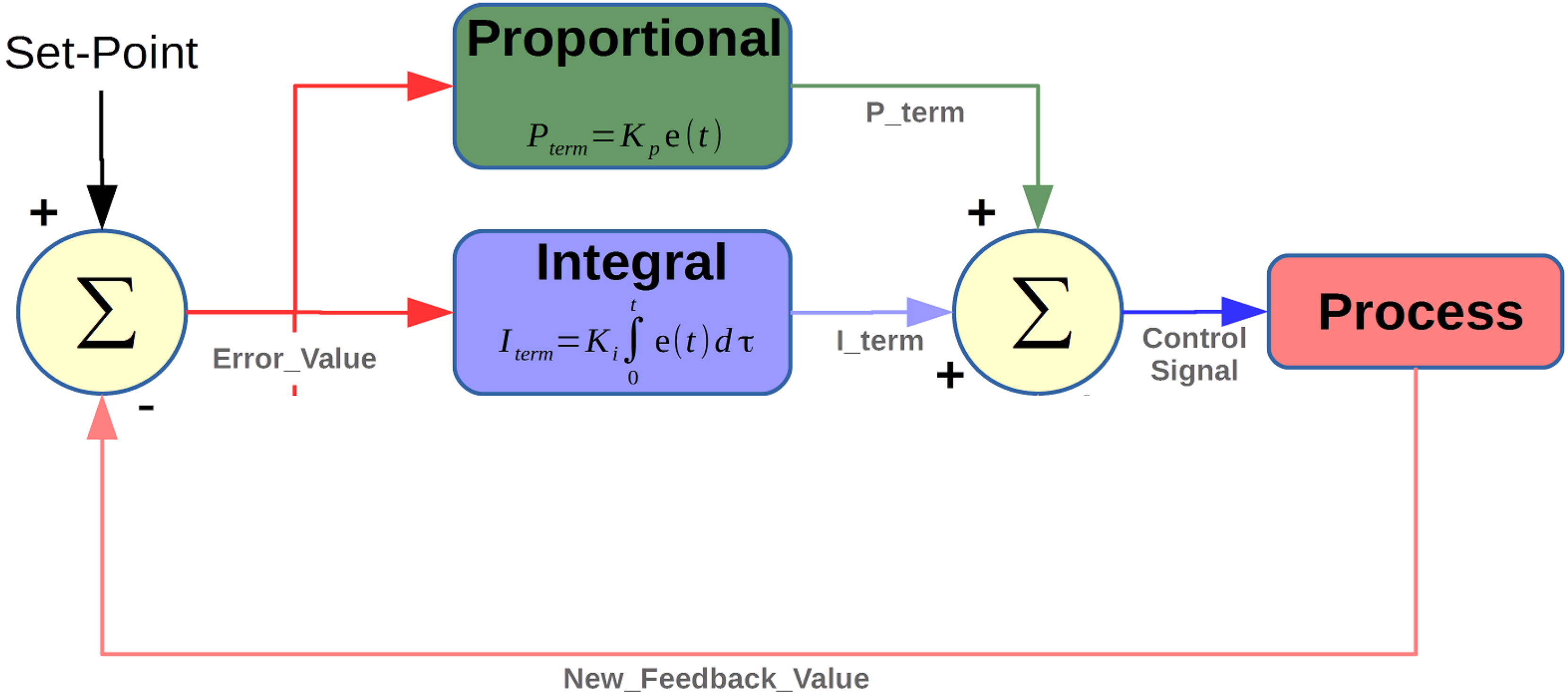
Control – Tuning

You can tune a piano, but you can't tuna a fish



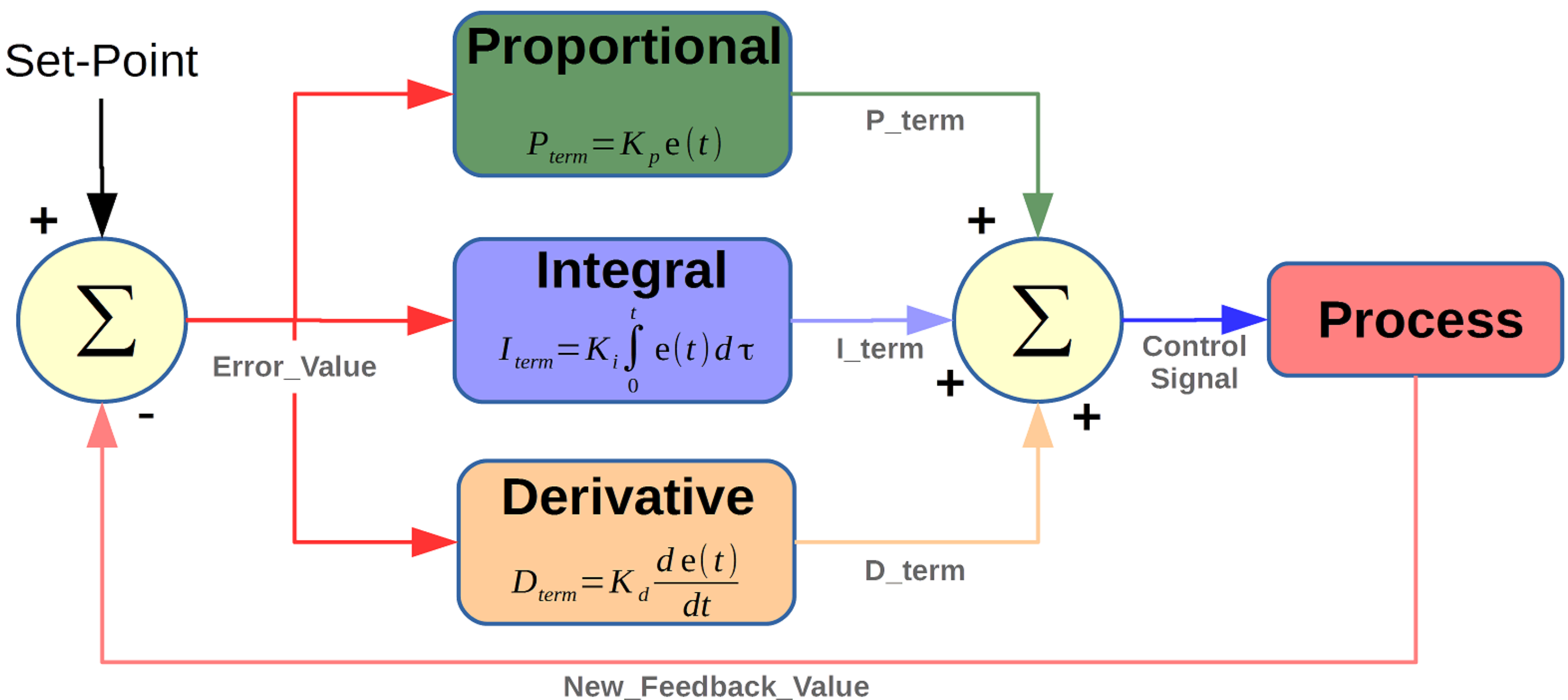
Control – PI

Add in I

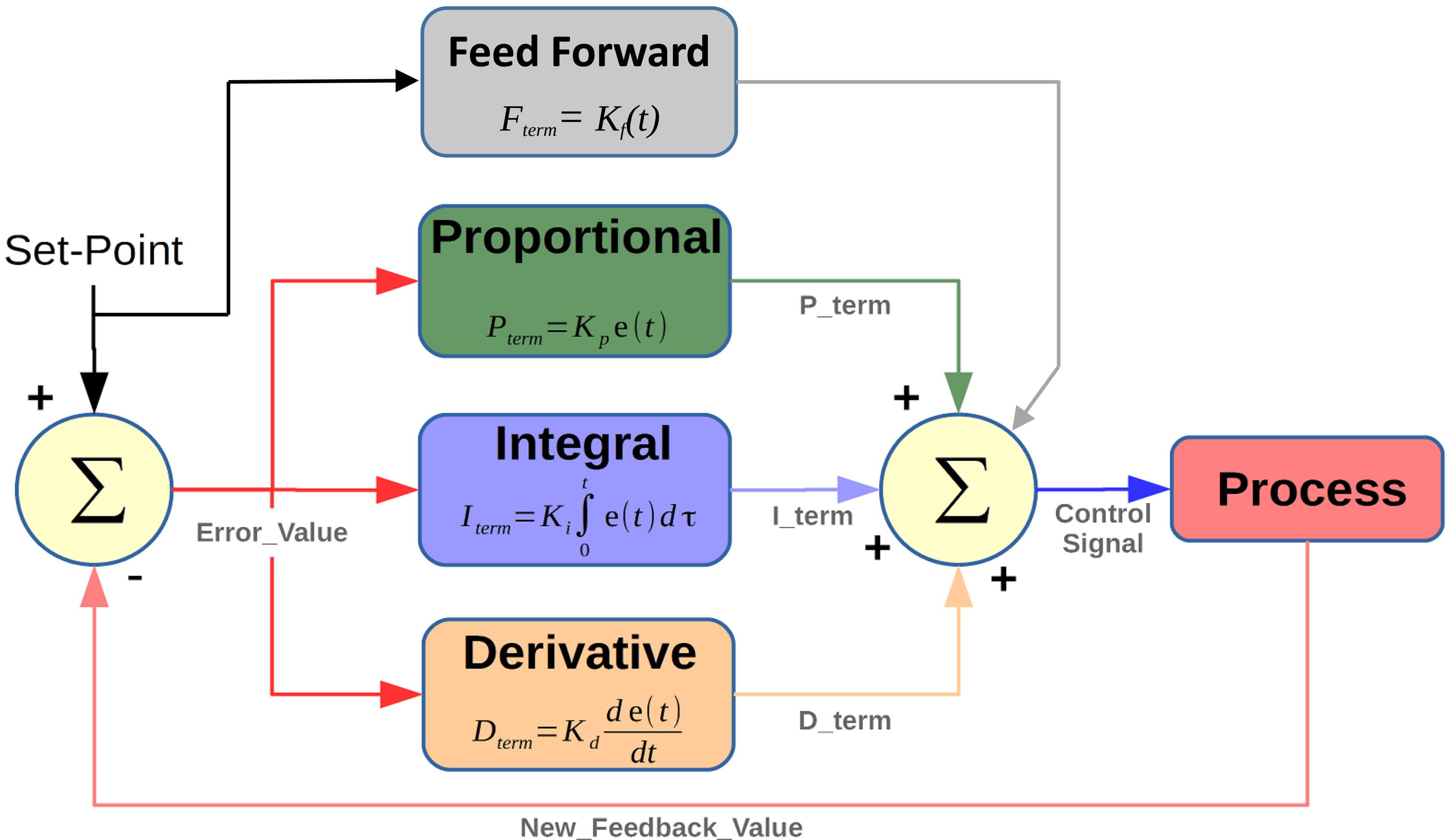


Control – PID

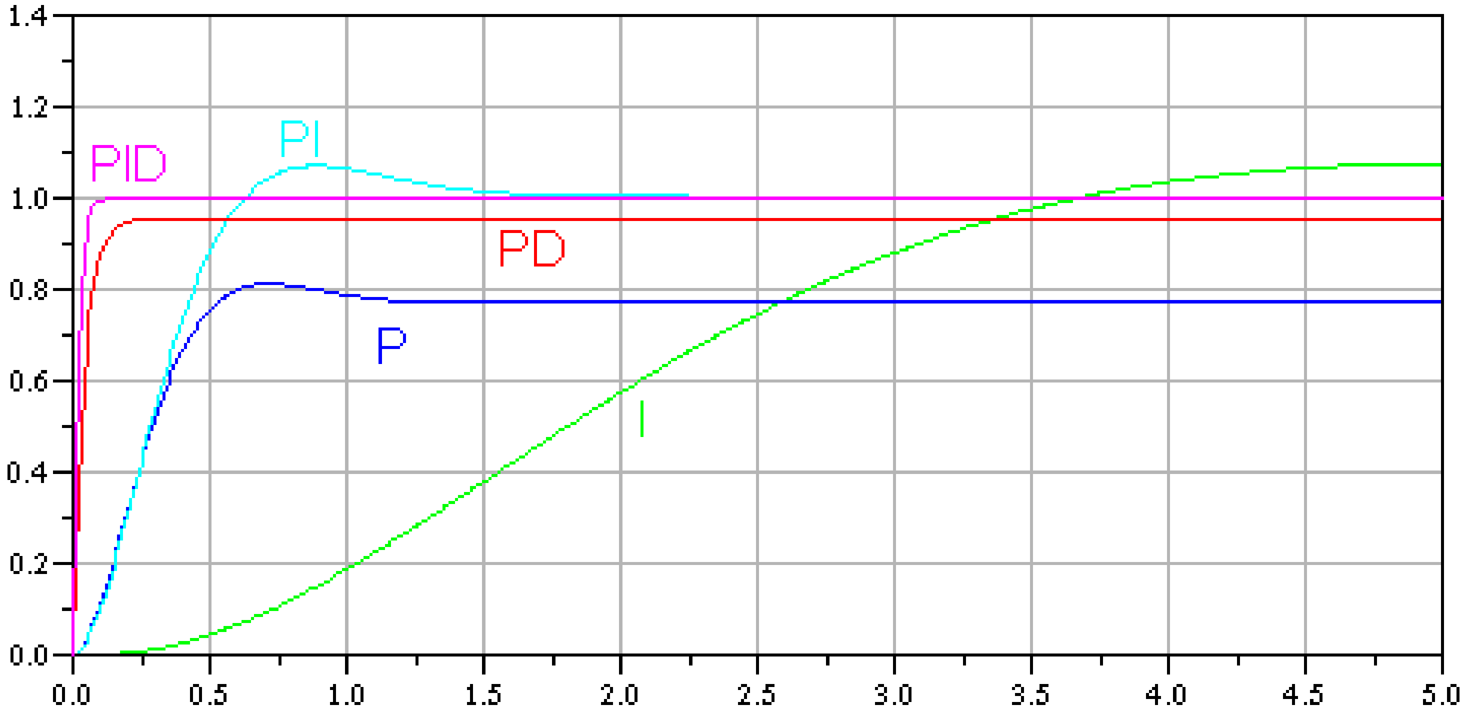
Add in D



Control – PID *with* FF



Control – PID Effects



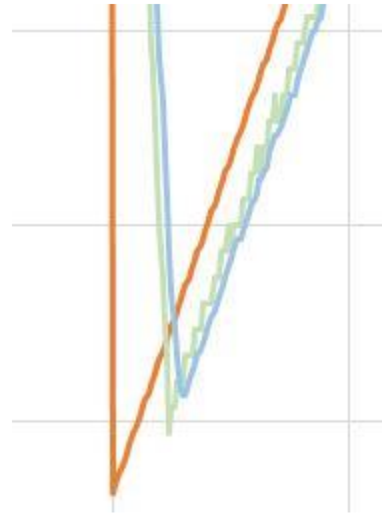
Control – The Real Thing

Lets do some tuning!

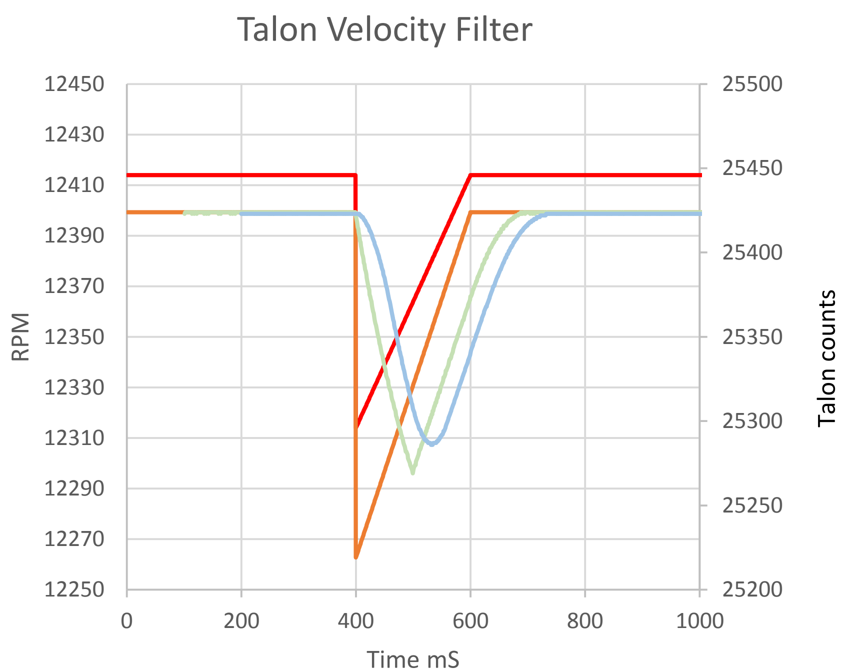
Tuning - Velocity

- CIM & CIMcoder with a Pulley (no load)
  - I limit
  - F
  - P
  - I
  - Izone
  - D
  - Velocity Filter
  - Voltage Ramp

Tuning - Velocity



- Velocity Filter – Period
- Velocity Filter - Samples



Shaft
Talon input
After Period
After Samples

Period – 100mS
Samples - 64

Period – 25mS
Samples - 8

Tuning - Velocity

- CIM & CIMcoder with a Pulley (no load)
- with Drag
- with Inertia
- with Drag and Inertia

Tuning - Position

- 9015 with Vex 100:1
    - I limit
    - P
    - D
    - Voltage Ramp
    - Voltage Max Limit

Tuning – Motion Magic

- A **Position** loop running on top of a **Velocity** loop
    - I limit
    - F
    - Cruise
    - Acceleration
    - P
    - D
    - Nominal Error (min acceptable error)

Quick Guide to Simple Tuning

- <u>Disclaimer!</u>

    1. This is a quick start on how to get something up and running with a “seat of the pants” approach.
    2. This is not a comprehensive tuning process.

Quick Guide to Simple Tuning

- <u>For a standard velocity loop with some amount of static load / drag</u>
  1. Ensure FF, P, I, D, Izone are all set to zero. Ensure Max FWD and Max REV volt are set to 12 and -12.
  2. Ensure your encoder is in phase with your motor. Run the motor in voltage mode and note that a positive voltage creates a positive velocity. If not, reverse the encoder.
  3. Select Velocity mode
  4. Increase FF until the actual velocity (process variable) is close to, but not higher than the requested velocity (set point) throughout the desired dynamic range.
  5. Increase P until the loop goes unstable indicated by continual oscillations or a long time to stop oscillating. Then cut that P value in half. Check it throughout the entire dynamic range.
  6. Run through the range of desired set points and note the largest error at any single speed.
  7. Set Izone to 1.5x of the largest error observed in step 6
  8. Increase I until the loop goes unstable. Reduce I until the settling time is optimized. You will end up trading off some overshoot vs a longer settling time. (I will likely be one tenth of P)
  9. If required, increase D to reduce the overshoot incurred in step 8. (D will likely be 10 to 100x of P)
  10. Go back and adjust P, then I, then D to observe the effects. They all interact so after the first pass, there may be some tweaks that will improve the performance.
  11. You may find that limiting the Max FWD and Max REV will help stabilize the loop if you don't need the full speed.

Quick Guide to Simple Tuning

- <u>For a standard position loop with some amount of static load</u>
  1. Ensure FF, P, I, D, Izone are all set to zero. Ensure Max FWD and Max REV volt are set to 12 and -12.
  2. Ensure your encoder is in phase with your motor. Run the motor in voltage mode and note that a positive voltage creates a positive position change. If not, reverse the encoder.
  3. Select positon mode
  4. IF there is a static load on the axis that ALWAYS loads the system in the SAME direction, Increase FF until position (process variable) is close to, but not higher than the requested position (set point) throughout the desired dynamic range. If there is no static load leave FF at zero.
  5. Increase P until the loop goes unstable. It will continue to oscillate or takes a long time to settle. Then cut that P value in half. Check it throughout the entire dynamic range.
  6. Increase D to manage the position overshoot that results from increasing P. With some D, you will likely be able to increase P beyond what was stable with P alone.
  7. A PD approach may be sufficient to drive the position close enough to the desired setpoint.
  8. You may find that limiting the Max FWD and Max REV will help stabilize the loop if you don't need the full speed.
  9. If you need more precision … continue

Quick Guide to Simple Tuning

- For a standard position loop with some amount of static load (continued)
  10. Run through the range of desired set points and note the largest error at any position.
  11. Set Izone to 1.5x of the largest error observed in step 3
  12. Increase I until the loop goes unstable. Reduce I until the settling time is optimized. You will end up trading off some overshoot vs a longer settling time.
  13. Go back and adjust P, then I, then D to observe the effects. They all interact so after the first pass, there may be some tweaks that will improve the performance.
  14. If you have a high resolution encoder (lots of ticks per distance) you may find the increasing the Minimum Error to the max you can tolerate will help stabilize the loop.
  15. You may also find that increasing the nominal FWD and REV voltages to the minimum amount that will actually make the axis move (in voltage mode) will also help stabilize the loop, especially if you are using some I.
  16. If your axis goes over center where gravity or some other force loads the system in both directions, do no use any FF. Just start with P and go from there.

REO Speedwagon 1978

You can Tune a piano, but you can’t Tuna a fish

Questions?


stryker
Midlink
Business Park
CLASS OF CHAMPIONS
20
17
FIRST
27
67