

*Practical Controls  
for Robot Geeks*

Ted Larson  
Winter 2008

# Objectives of this Presentation

---

- To get your mind thinking about applications of controls in your robots.
- Give you a brief introduction to a number of different types of controls that are out there.
- To give you enough technical jargon, that you can do your own investigation on controls, and control theory, so you aren't overwhelmed right out of the gate.
- Show you how to implement a control in a robot, that you can begin using now.
- Immediately solve the hardest problem in hobby robotics. HOW DO I MAKE MY ROBOT DRIVE STRAIGHT?!?!?
- Give you tons of references, and book titles to look at, to keep your brain busy well after I am done.

# Overview

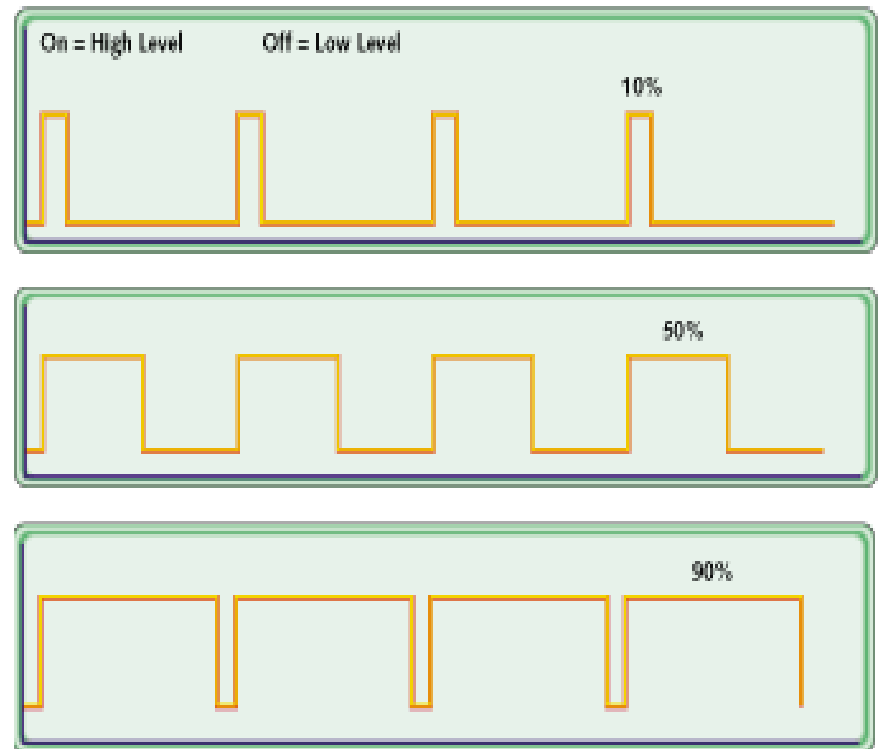
---

- Motor Driver Basics & Feedback
- Control System Types
- Deciphering Control Papers
- Proportional Integral Derivative (PID) Control
- Examples in Robots
- How to drive straight!
- References to learn more
- Going Broad on most things – Lets try not to get hung up on a million details.

# Motor Driver Basics

---

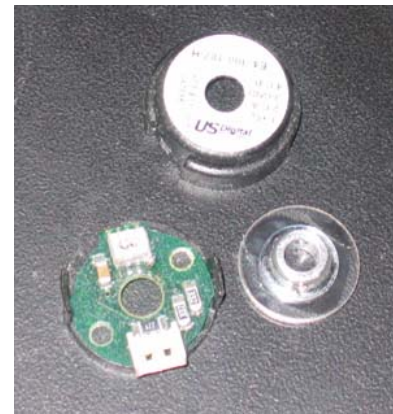
- Pulse Width Modulation (PWM)
- Sign Magnitude vs. Locked Anti-phase
- Easy to use chips are the L298, L293, LMD18200 and the SN754410.



# Motor Feedback

---

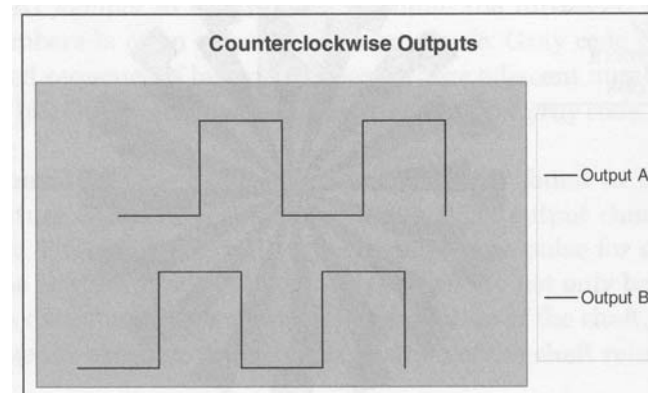
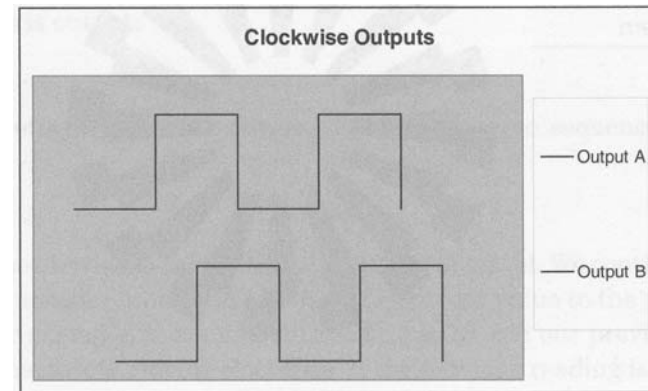
- Many ways to do it such as potentiometers, hall-effect sensors, or optical encoders.
- Speed and Direction
- The more accurate, the better results.



# Optical Encoders (My favorite)

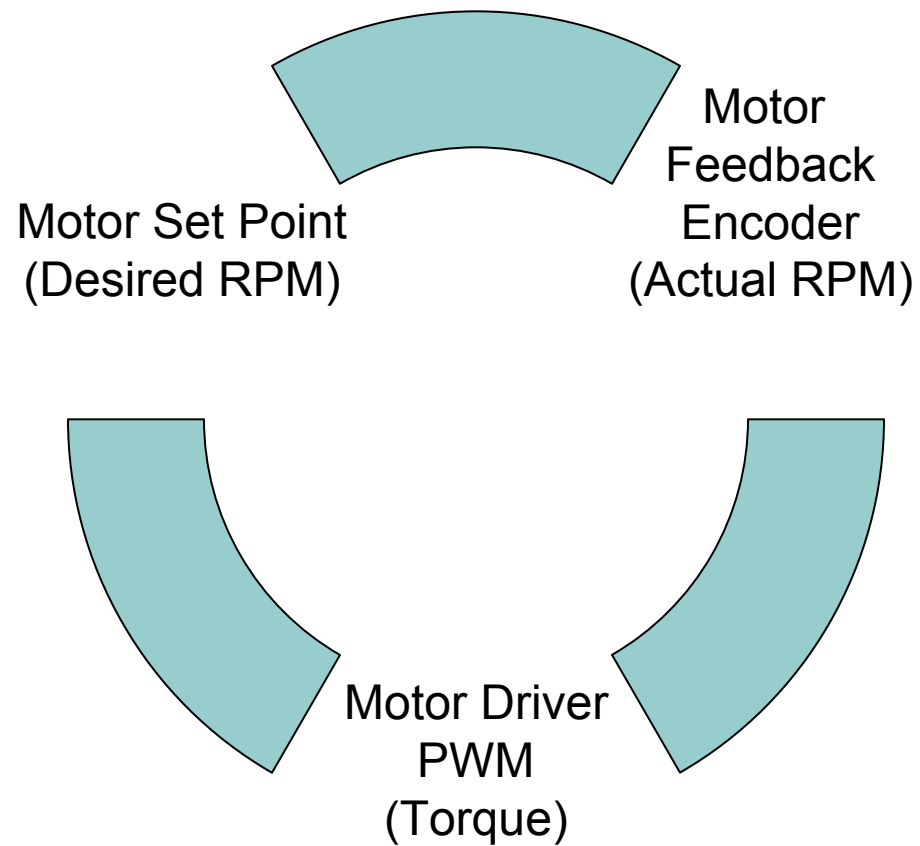
---

- Quadrature
- Motor mount versus wheel mount trade-off.
- Reducing the problem to speed and direction.
- Better for velocity controls, not so good for position controls, without a reference sensor, or index pulse.



# Closed Loop Motor Control

---

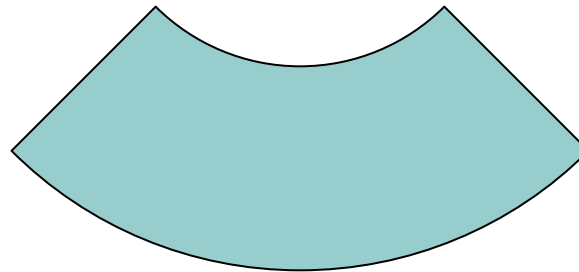
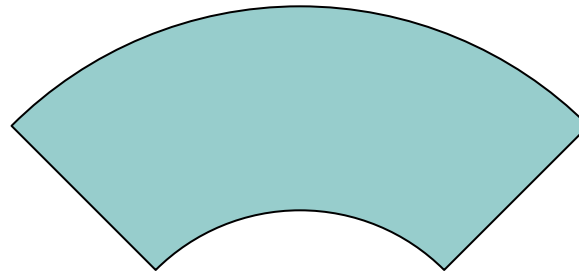


# Open Loop Motor Control

---

Motor Set Point  
(Desired RPM)

Motor Driver  
PWM  
(Torque)



# Controls are Your Robots New Best Friend!

---

- Motor Control – Position & Velocity
- Precision Turning and Odometry
- Navigation
- Distance to objects and approach
- Wall following & Line Following
- Pan-Tilt Heads
- Object Tracking & Target Pursuit
- Balancing Robots
- Many more applications! Use your imagination!

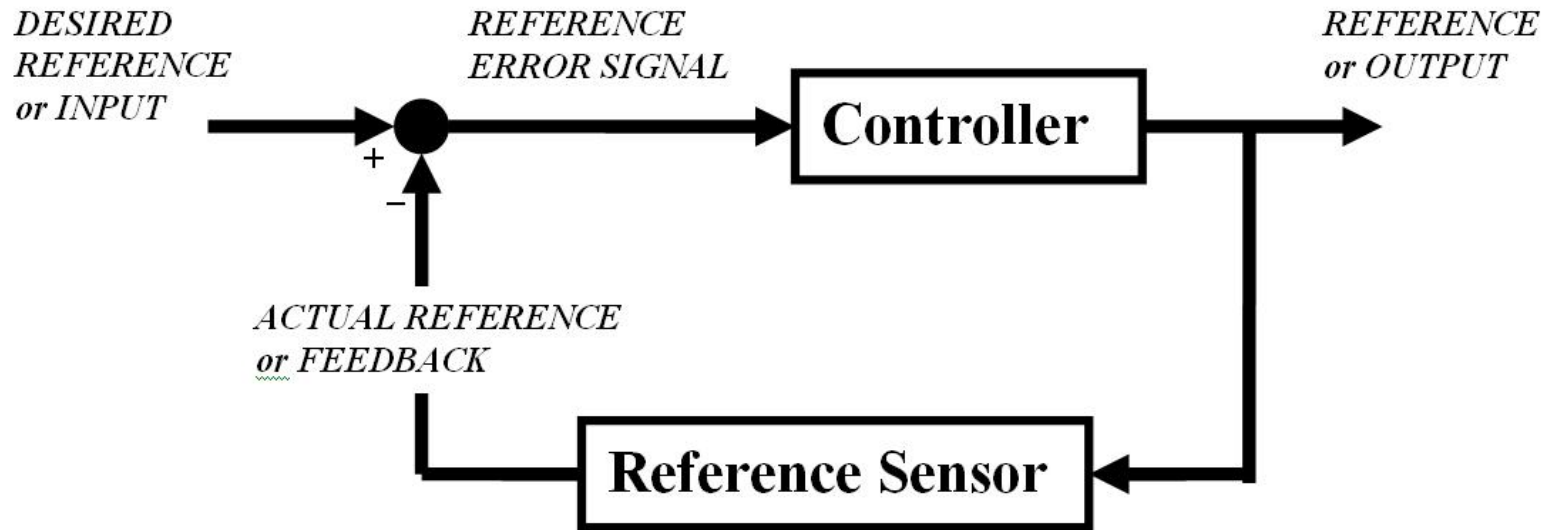
# Control Choices

---

- The Modeling dilemma / Slippery Slope
  - How do you model this stuff?
  - Easy to get in over your head fast
  - Hmm...Signal Processing and Controls all seem to blend together into the same stuff.
- Closed vs. Open Loop Controls
- P, PI, PD, or PID Controls - SISO
- MIMO LTI, Optimal Controls – LQR, Minimax, LQG, and  $H^\infty$  controls.

# Control Diagrams

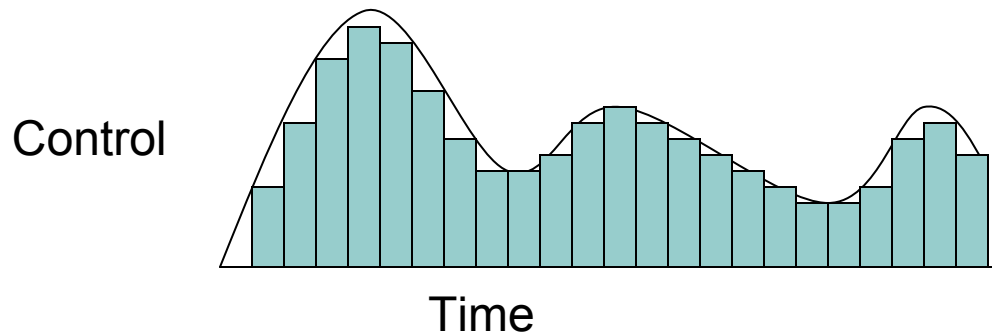
---



# Control Loop Timing

---

- Timing is critical for loop stability
- Discrete vs. Continuous time (theory)
- Bandwidth of sensor feedback will most likely be the determine scaling.



# Deciphering Control Papers

---

- Modeling vs. Real World
  - Transfer Functions
- Continuous vs. Discrete Time
  - Laplace Transform
  - Z-Transform
- Theoretical Design Methods
  - Nyquist, Bode & Nichols Plots
  - Root Locus
  - Lead/Lag Compensation
  - LTI Representation
  - State Space Representation
  - Sample Rate Selection
  - Quantization Effects in Digital Systems (Round-off..etc)
  - Agghhh! Is there an insane asylum I can check into nearby?

# Deciphering Control Papers

---

- Considerations when choosing a method.
  - Non-Linearity
  - Parametric Models
  - Process Variations
  - Disturbances
  - Setpoint Changes
  - Measurement Noise

## PID Control – Overview

---

- (P) Proportional – Control action is proportional to the control error.
- (I) Integral – Makes sure the process output agrees with the setpoint in steady state.
- (D) Derivative – Improves closed-loop stability, and reaction to setpoint changes.

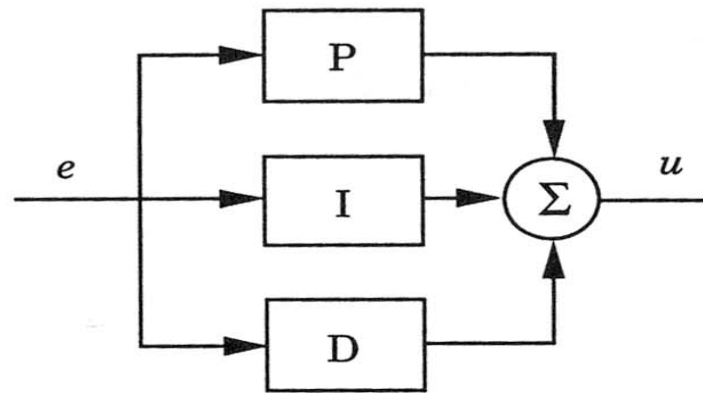
# PID Control Usage

---

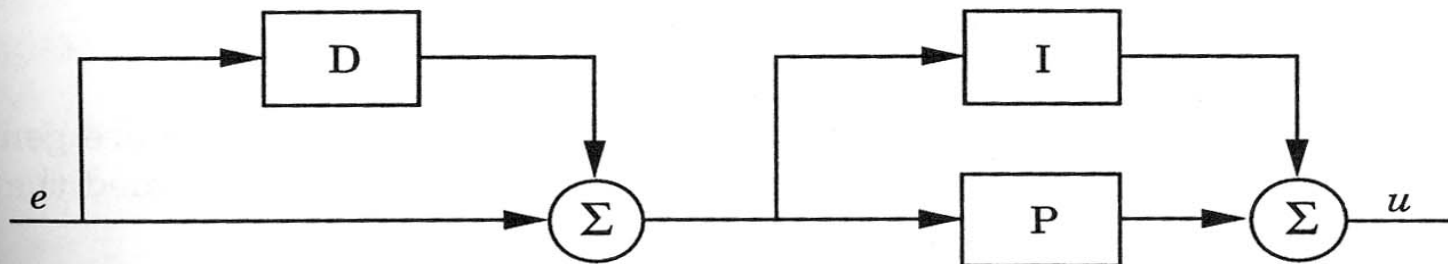
- System to be controlled needs to have a linear transfer function.
- Applications for PID
  - Temperature Regulation
  - Pressure Regulation
  - Flow Rate Regulation
  - Chemical mixing or composition
  - Velocity or Speed Regulation

# Two Most Common PID Forms

---



Non-interacting form



Interacting form

# PID Control Mathematically

---

$$f(t) = K_p e(t) + K_i \int e(t) dt + K_d \dot{e}(t)$$

- Math Cheat Sheet:
  - Integral of Velocity is Position.
  - Derivative of Position is Velocity.
  - Integral of Acceleration is Velocity
  - Derivative of Velocity is Acceleration
- Velocity vs. Position Control –What is the difference anyway?

## Simple Proportional Control

---

- $\text{Error} = \text{Desired RPM} - \text{Current RPM}$
- $C_v = \text{Error} * K_{\text{prop}}$
- $\text{New PWM} = \text{Last PWM} + C_v$
- Do this loop over and over again, until the error reaches zero.

# Proportional Integral Control

---

- $I_{sum} = I_{sum} + Error$
- $Error = Desired\ RPM - Current\ RPM$
- $Cv = (Error * K_{prop}) + (I_{sum} * K_{int})$
- $New\ PWM = Last\ PWM + Cv$
- Do this loop over and over again, until the error reaches zero.

# Full PID Control (For a motor)

---

```
if(!speed)
    mint=0
rpm = ReadRPM();
err = speed - rpm;
de = err - perr; //Change in error
macc+=((err/pid_k_prop)+(de/pid_k_dev)+(mint/pid_k_int));
perr = err;

if(macc > MAX_OUTPUT)
    macc = MAX_OUTPUT;
if(macc < MIN_OUTPUT)
    macc = MIN_OUTPUT;
if(macc!=MAX_OUTPUT && macc!=MIN_OUTPUT)
    mint+=err;

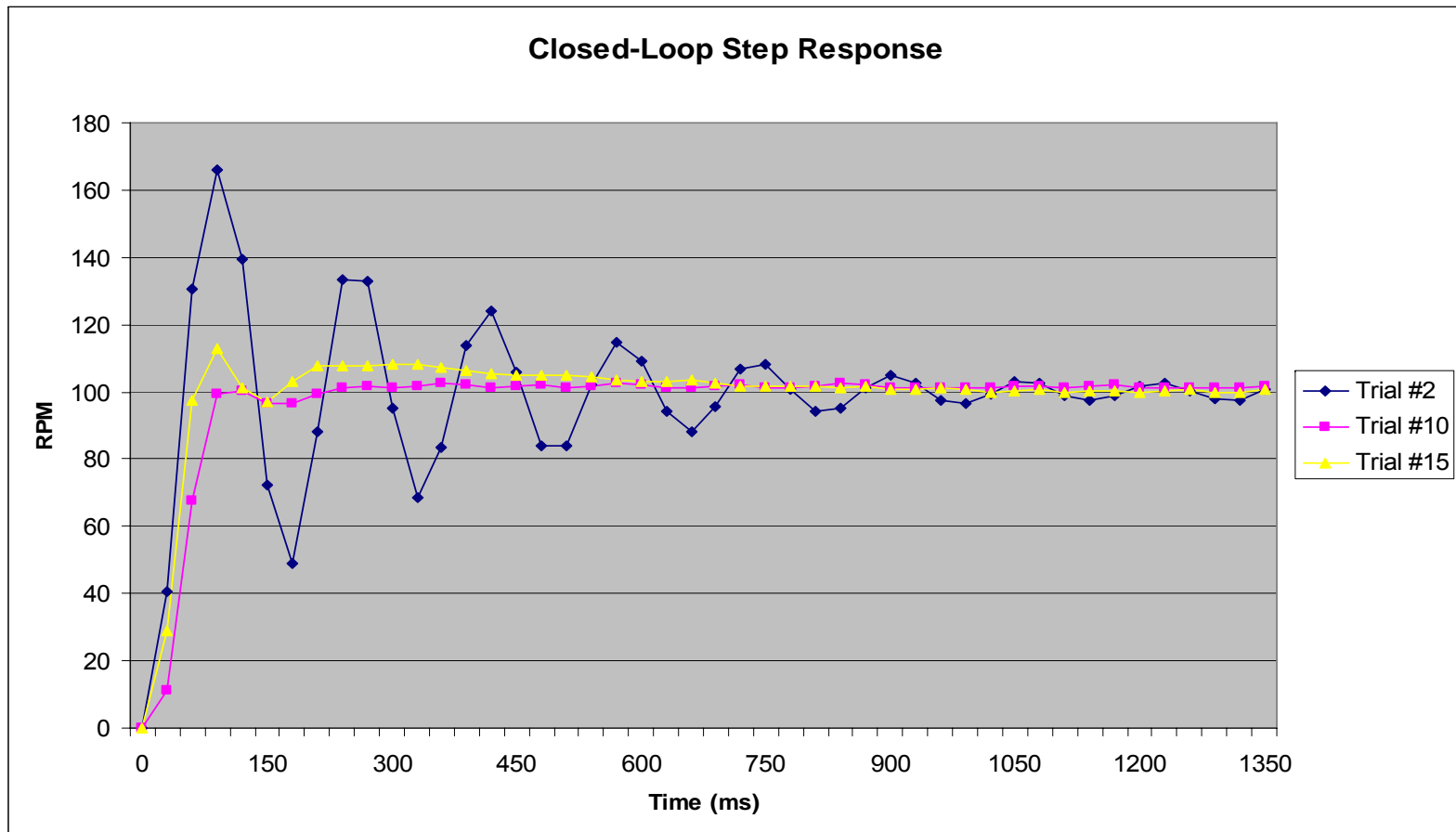
SetPWM(macc);
```

# Tuning a Closed Loop Control

---

- Selecting initial values for  $K_p$ ,  $K_i$ , and  $K_d$ 
  - Ziegler-Nichols Tuning
  - Open & Closed Loop Step Response
- Tuning by hand (trial-and-error)
  - Boost  $K_p$  until it oscillates
  - Boost  $K_d$  to stop oscillation, back off  $K_p$  by 10%
  - Dial in  $K_i$  to Hold position or velocity smooth
- Automatic tuning more theory than practical -Good luck!
- Realistic testing environment is key.
- Try different environments, and find the best set of variables that works for all. You might find there isn't one! Uh-oh...now I need to go back to all that theory.

# Step-Response Optimization



# Examples

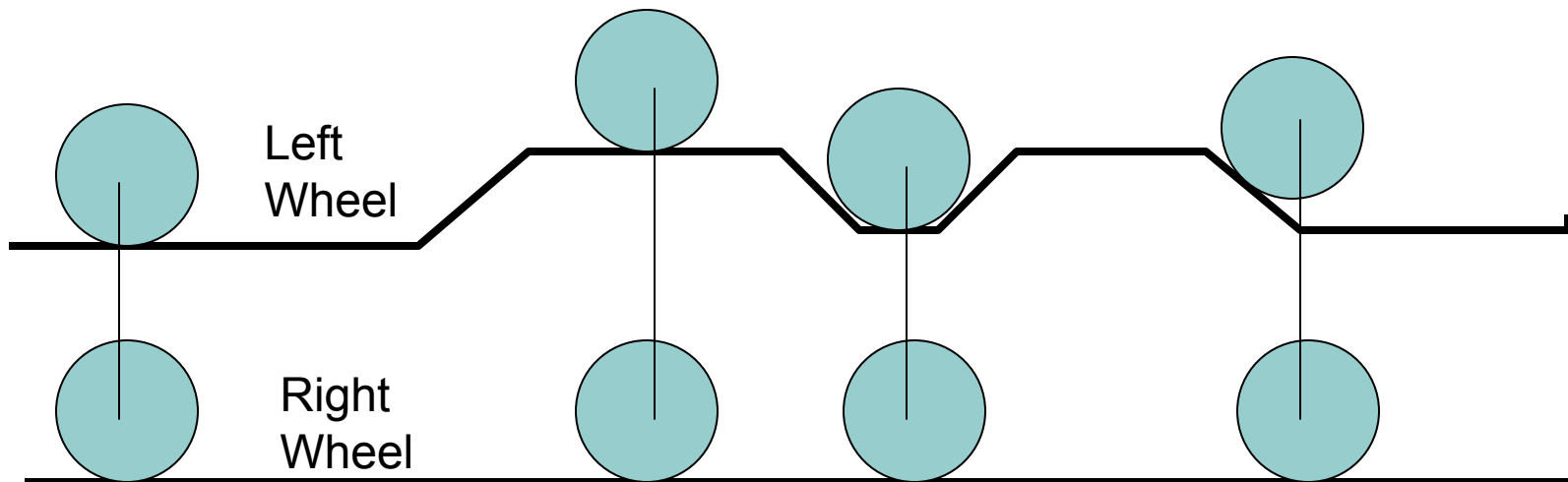
---

- PID Control for the tabletop Challenge
- Pan-Tilt Head experiments – Notice how the control is over-tuned.



# How to Drive Straight & Steer

---



- What to do when one wheel travels different distance than the other?
- How do you keep two wheels traveling at the same speed?

# Algorithm to Drive Straight

---

Loop:

PVLeft = ReadLeftSpeed()

PVRight = ReadRightSpeed()

Isum = Isum + (PVLeft – PVRight) + Steering

LeftErr = SP – PVLeft

RightErr = SP – PVRight

CVLeft = Kprop \* LeftErr – Kint \* Isum

CVRight = Kprop \* RightErr + Kint \* Isum

SetRightPWM(CVRight)

SetLeftPWM(CVLeft)

Goto Loop

\* Taken from back of Building Robot Drive Trains – Robot DNA Book.

## Beginning PID Books

---

- Robot DNA: Building Robot Drive Trains by Dennis Clark and Michael Owings
- Mobile Robots Inspiration to Implementation by Joseph-Flynn-Seiger
- PID Controllers: Theory, Design and Tuning by K. Astrom and T. Hagglund

## Advanced PID Books

---

- Control System Design Guide – Using your computer to understand and diagnose feedback controllers – George Ellis – Third Edition
- Advanced PID Control – Karl J. Astrom and Tore Hagglund
- Controller Tuning and Control Loop Performance – “PID without the math” – Second Edition – David W. St. Clair

## Advanced Controls Books

---

- Applied Optimal Estimation by the Technical Staff, The Analytic Sciences Corporation – Edited by Arthur Gelb
- Optimal State Estimation, Kalman,  $H^\infty$ , and Nonlinear Approaches by Dan Simon
- The Control Handbook – William S. Levine, CRC Press & IEEE Press

## Math Books

---

- Numerical Linear Algebra – Lloyd N. Trefethen & David Bau.
- MATRIX Computations – Gene Golub & Charles Van Loan – Third Edition
- Schaum's Outlines – Signals and Systems – Hwei P. Hsu

# Papers and Datasheets

---

- **Larry Barelo's Motion Control Paper**  
[http://www.barello.net/Papers/Motion\\_Control/index.htm](http://www.barello.net/Papers/Motion_Control/index.htm)
- **Designing a PID Motor Controller:**  
<http://www.seattlerobotics.org/encoder/200205/PIDmc.html>
- **Using a PID-based Technique for Competitive Odometry and Dead-Reckoning**  
[http://www.seattlerobotics.org/encoder/200108/using\\_a\\_pid.html](http://www.seattlerobotics.org/encoder/200108/using_a_pid.html)
- **LM628 Datasheet**  
<http://www.national.com/pf/LM/LM628.html>
- **Wikipedia – Control Theory**