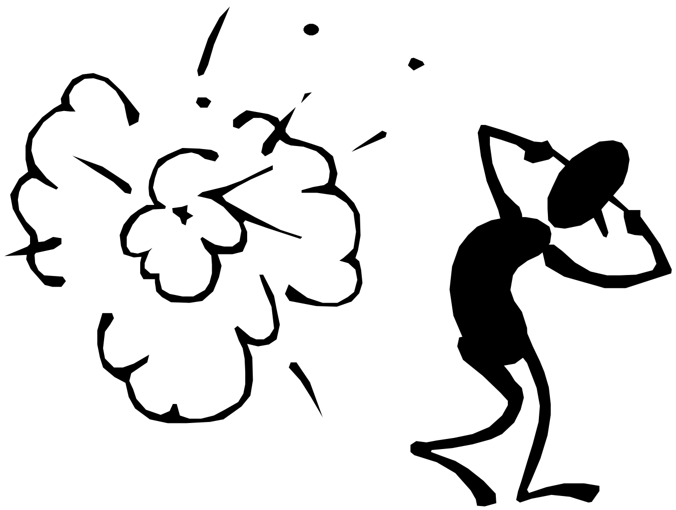
# *EE281 Fall 2002*

## ~~EECS192 Spring 2000~~

# ~~Project~~ Presentation



**Erik Olson**

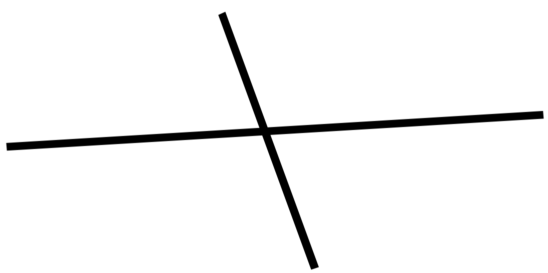
**Toru Kuzuhara**

# What is Natcar?

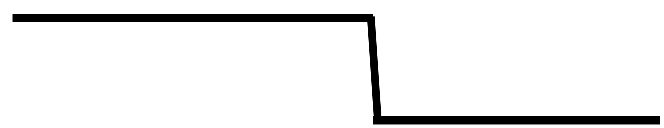
- Small annual competition sponsored by National Semiconductor
- Goal is to design an autonomous car to navigate a racecourse specified with a current carrying wire and tape
- Participating Universities:
  - UC Berkeley
  - UC Davis
  - San Jose State
  - Stanford, Oklahoma State, Sacramento State as well

# Natcar Rules

- 1/10 scale car powered with a single 7.2V NiCd battery
- The motor and the steering servo may be purchased commercially, but all sensing and control electronics must be designed and built
- Course marked by white tape on dark carpet
- The tape is put down on top of a wire which is formed in a complete circuit and driven by a 75 kHz sinusoidal signal. The loop is driven with 100 mA RMS +/- 10%
- The course may cross itself at an angle of greater than 60 degrees
- May be a 6 inch straight section followed by a 3 inch “jog”



Crossing



Jog

# Components of a Workable Solution

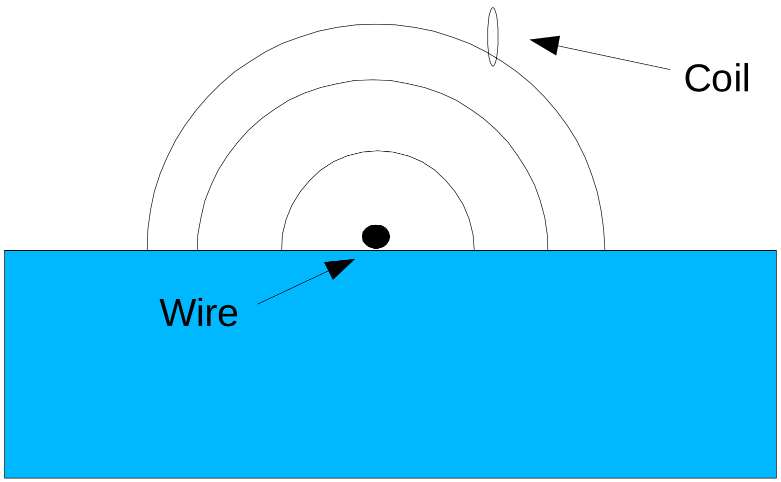
- Sensor Design
- Steering/Motor
- DC/DC Converters
- Feedback Control

# Sensor Design

- Optical solution
  - Emit light onto the floor, and try to detect if white tape is there
  - Digital signal, basically on the tape, or off
  - Relatively noise free
  - Could use multiple sensors to get a relative position

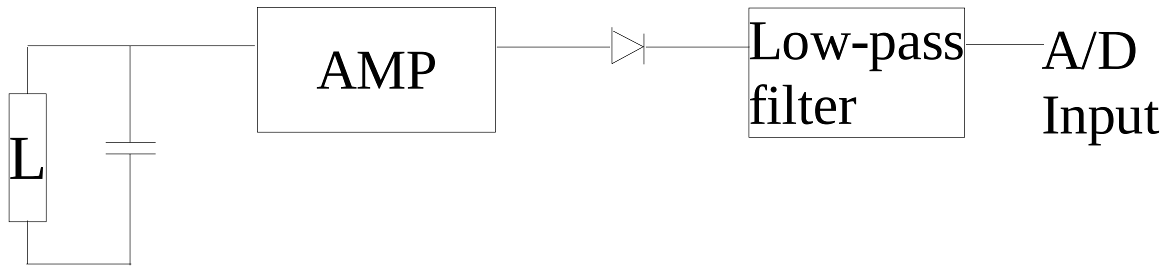
# Better Sensor Design

- Detect the magnetic field of the wire under the tape using coils (inductors)
  - Analog signal, dependent on distance from wire, angle of B field
  - Could be noisy



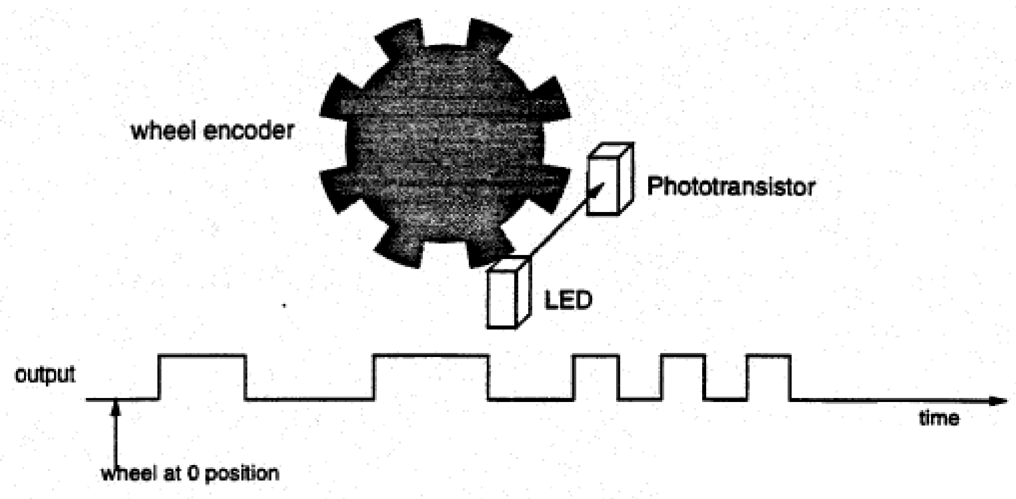
# Sensor Amplification

- The current picked up by the inductor is very small, and it is AC
  - First need to bandpass
  - Then amplify
  - Then rectify
  - Then low pass
  - Feed to A/D converter



# Speed Sensor

- LED + Phototransistor can be used to detect speed (Hamamatsu S6846)

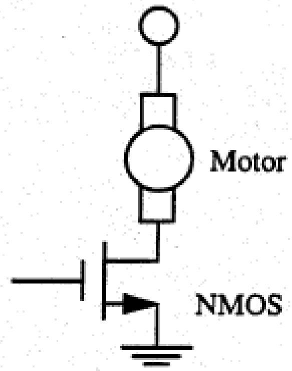


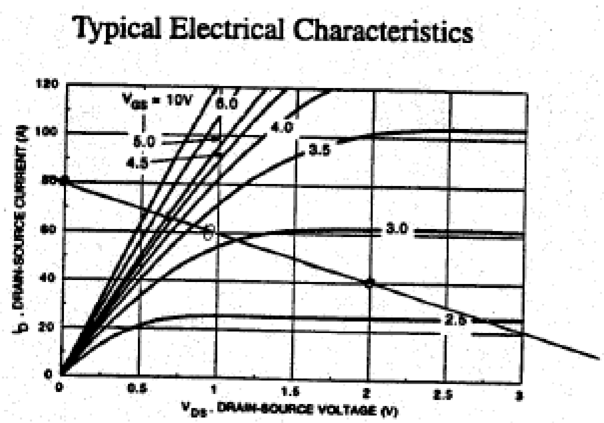
# Steering

- High torque digital servos (>$50)
  - Highly accurate control
  - Fast response time
  - Simple to control (PWM input)

# Motor Control

- Motor controlled by sending PWM to power MOSFET (NDP7060L)
  - Need high Vgs to prevent MOSFET failure
  - Easy to get linear speed control



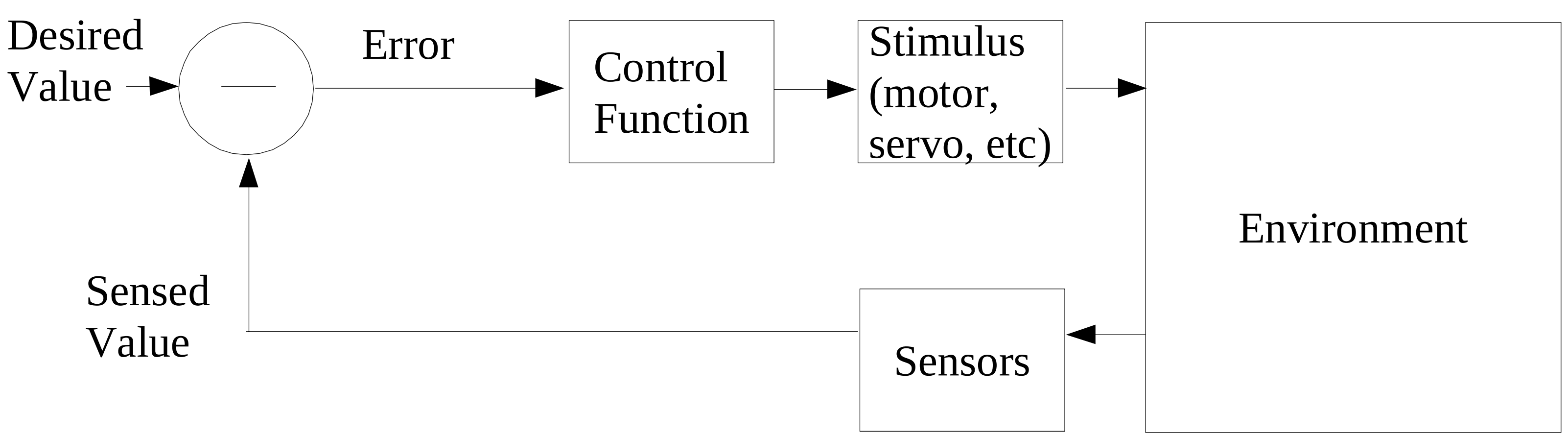



# DC/DC Converters

- Only allowed a 7.2V battery
- Need to generate 5V for electronics
  - Regulate 7.2V (LM2940)
- >9V for power MOSFET
  - Step-up DC/DC converter
- May need redundant regulators to deal with noise from motor/servo

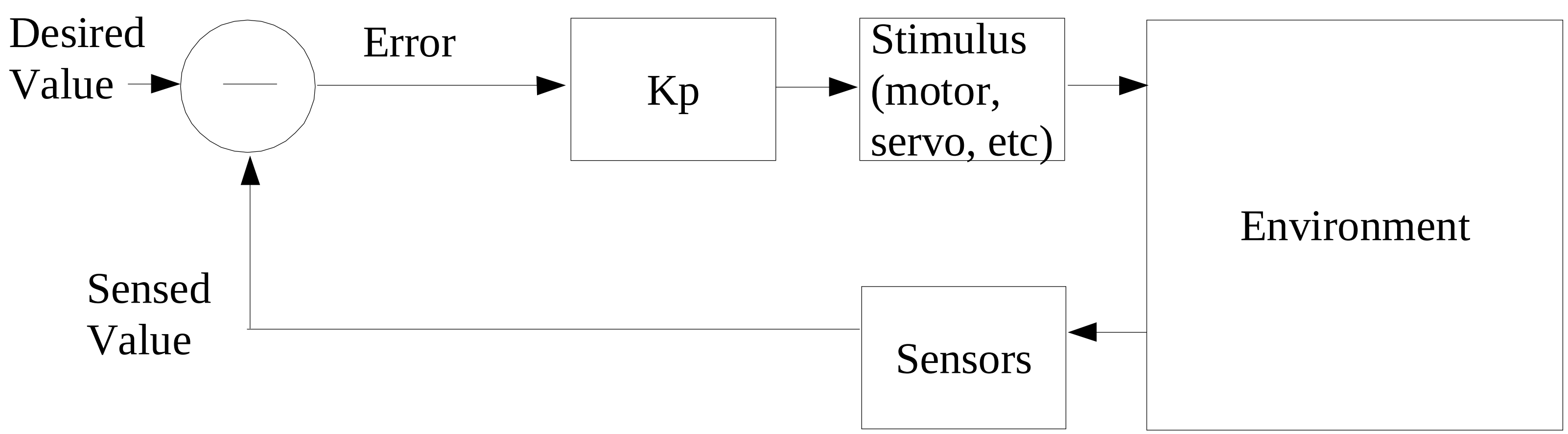
# Feedback Control (Simplified)

- Feedback control is a great way to control the car!



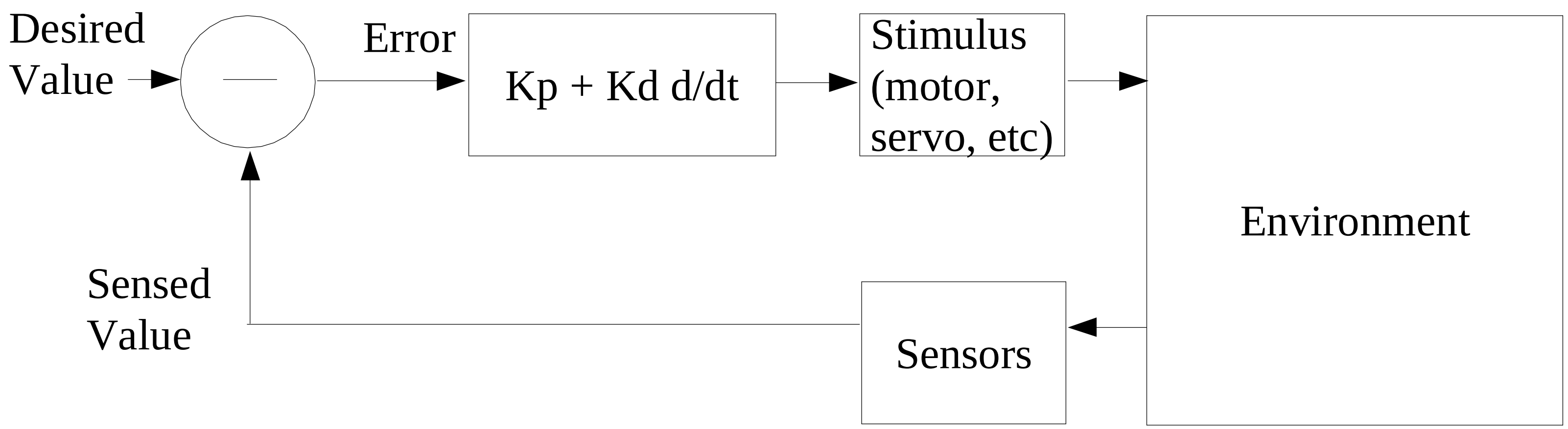
# Proportional Control

- Compensate with the measured error times constant Kp
  - Easy to implement
  - Likely to overshoot



# Proportional + Derivative Control

- Compensate with the measured error times constant Kp + derivative of error times Kd
  - Harder to implement, need to tune Kp and Kd very carefully
  - With the right values, you can get very good damping, and little or no overshoot



# Natcar 2000

## Our Implementation

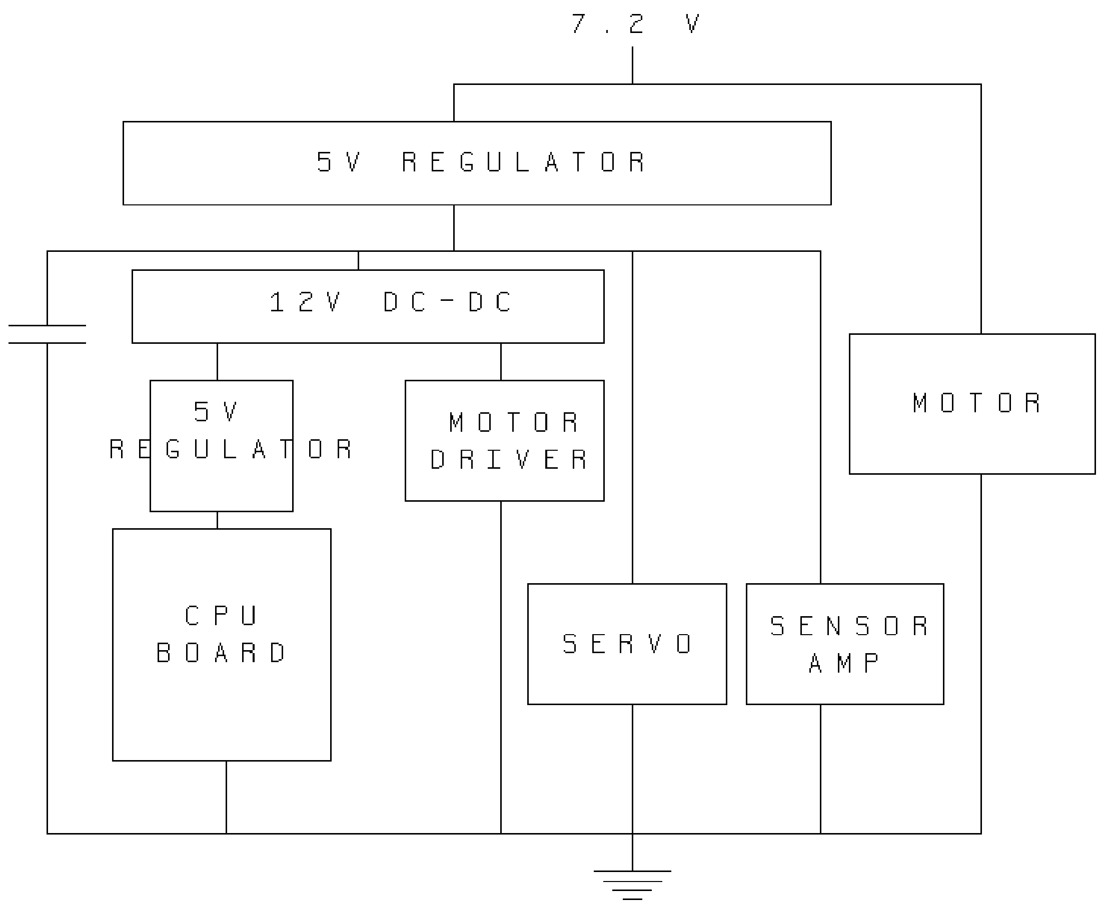
# Objectives

- Understand how to model real world behavior using theoretical calculations
- Gain practical experience with circuit design and implementation

# Strategy

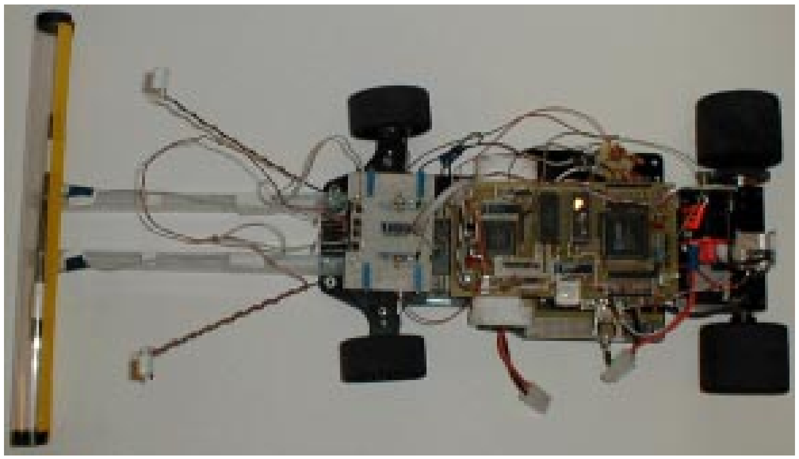
- KISS - Keep It Simple Stupid
- Simple, robust hardware
- Simple, reliable software

# Power Configuration




- 2200uF cap to smooth voltage spikes caused by motor
- inefficient regulator
$\rightarrow$ DC-DC $\rightarrow$ regulator path
- Simple
- Reliable

# Sensor Configuration



- Four sensors to provide greater stability
- Different heights for variable range
- POTs to adjust for optimal gain

# Software

- Simplicity
  - Easy to Code
  - Easy to Debug
  - Fast Execution
- Depends on linearity of delta

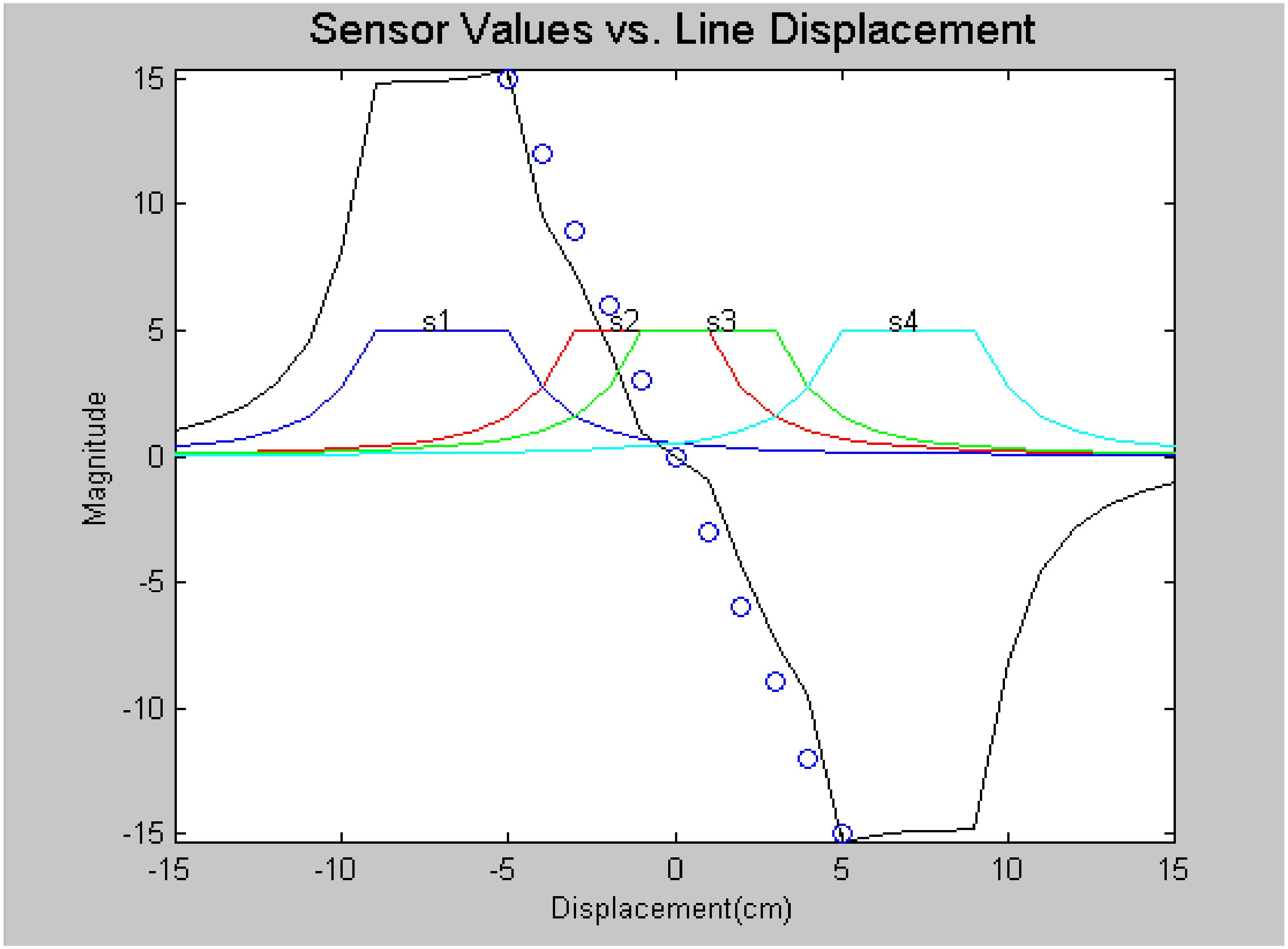
<u>Control Code</u>

```
if (sum >= 60){
    delta = (ch1 + ch2 - ch3 - ch4)/2;
    ServoPWM = 165 + delta;
} else if (ServoPWM > TURN_CENTER){
    ServoPWM = TURN_HARD_LEFT;
    MotorPWM = 15;
} else {
    ServoPWM = TURN_HARD_RIGHT;
    MotorPWM = 15;
}
/* Delta ranges from 16 to -16 */
if (delta > 0){
    MotorPWM = 25 - (delta/2);
} else {
    MotorPWM = 25 + (delta/2);
}
```

# Control

- Proportional Control
- Key Issues for Performance
  - Sensor Placement
  - Sensor Range
  - Software Digital Filtering
- Problems
  - Some instability due to narrow sensor range
  - Confusion at crossings

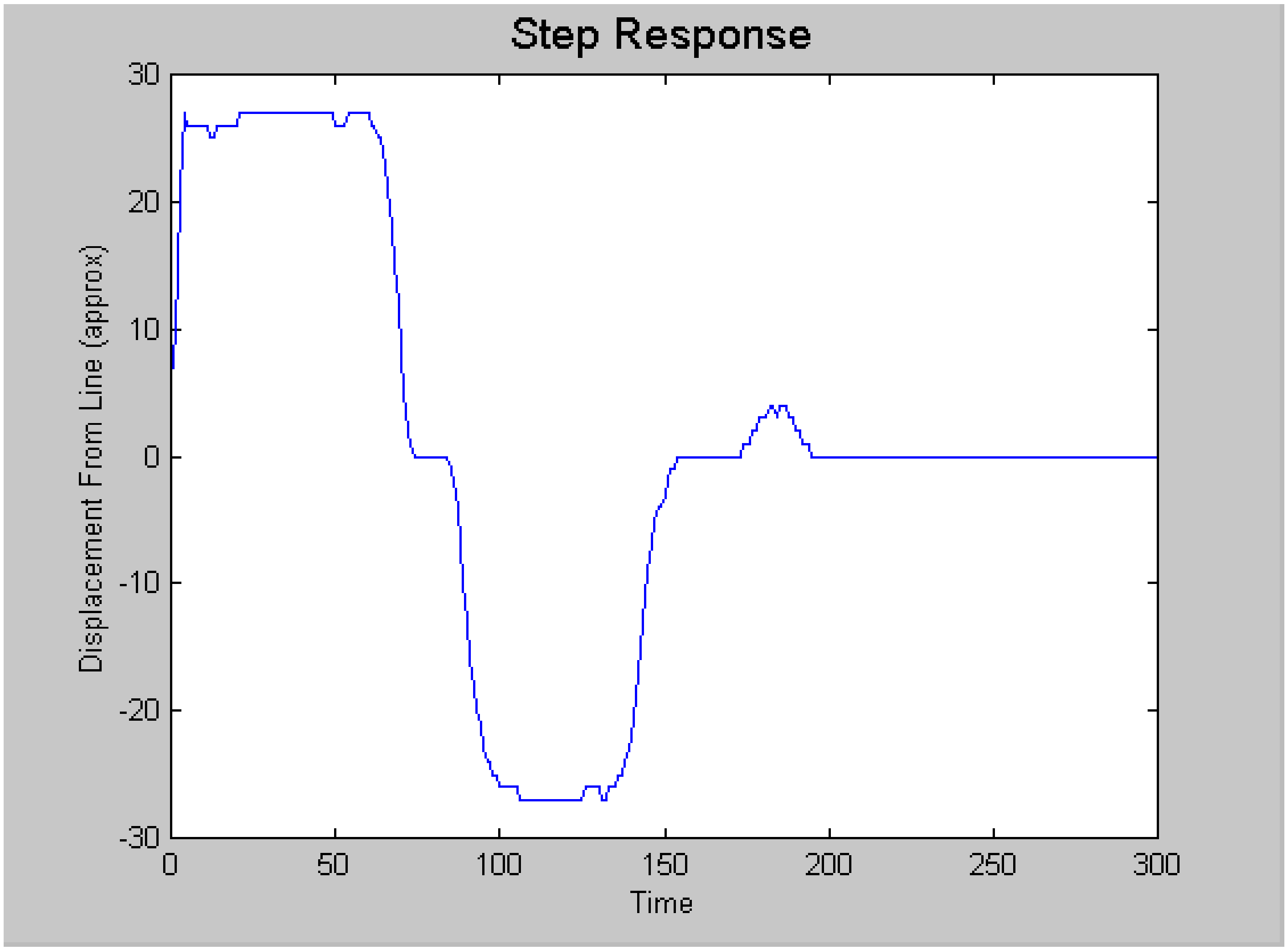
# Simulated Sensor Values



# Observed Performance

- Not the same as simulated performance!
  - Difficult to simulate sensor noise
- Real sensors, servos, wheel alignment, motor noise, etc. can cause problems
- Track variations are unpredictable

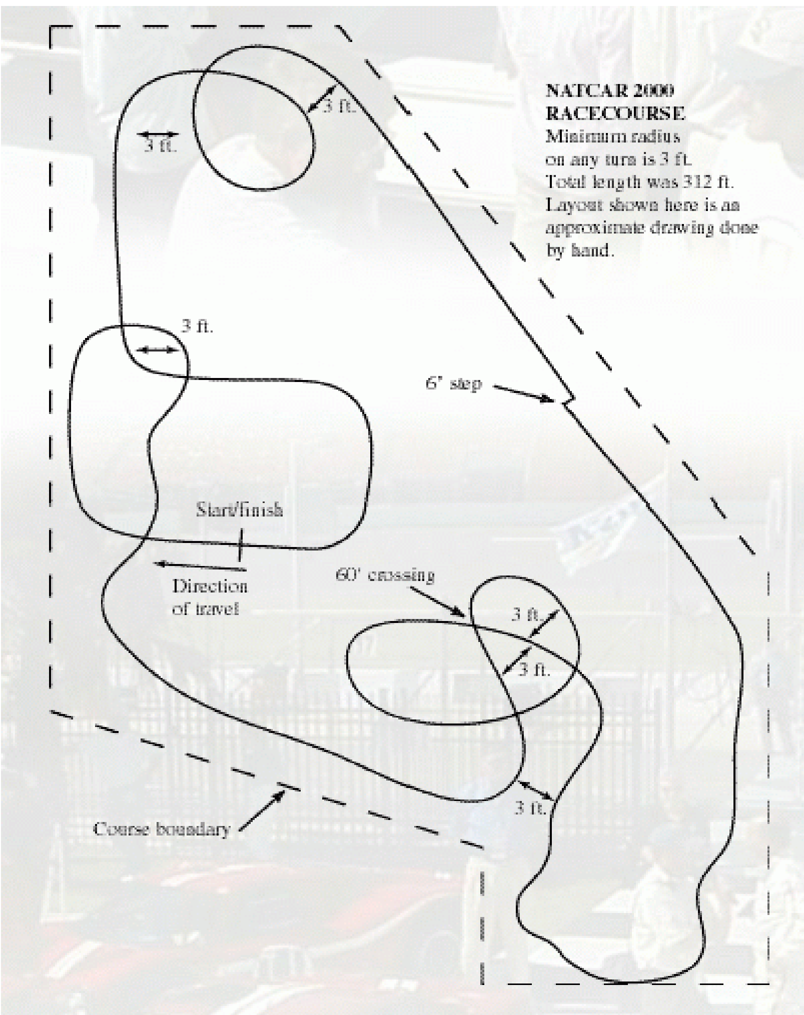
# Step Response



# Additional Features to be Implemented

- Derivative control
- Hamamatsu - velocity control
- Special jog and line crossing algorithms
  - weak on large jogs
  - consistently has trouble with 60 degree crossings
- Improvement of sensor placement
- Course memorization (maybe)

# Results




| Place | Team | Course Time (s) | Average Speed (ft/sec) | Racers |
|---|---|---|---|---|
| 1st | UCB4 | 31.76 | 9.82 | Max, Weng, & Ben-Artzi |
| 2nd | UCD3 | 40.95 | 7.62 | Evan Scarisbrick & Adrien Hagen |
| 3rd | UCD4 | 51.74 | 6.03 | Brett Bodine & Steve Maldonado |
| 4th | UCB3 | 52.45 | 5.95 | Kuzuhara & Olson |
| 5th | SJSU2 | 56.00 | 5.57 | Kwan, Chi, & Xie |
| 6th | SJSU4 | 67.44 | 4.63 | Chung & Ho |
| 7th | SJSU1 | 70.50 | 4.43 | Durrin, Haastrup, Bradbury, & Duell |
| 8th | UCB5 | 72.58 | 4.30 | Hori |
| 9th | UCD1 | 78.13 | 3.99 | Fang & Z. Chen |

# More information

- Natcar
  - http://www.ece.ucdavis.edu/natcar/
- Stanford
  - http://www.stanford.edu/group/natcar/
- Berkeley (a great source for detail on everything in this presentation)
  - http://www-inst.eecs.berkeley.edu/~ee192/