

EECS192 Lecture 11

Apr. 4, 2017

Notes:

1. Progress Report due Tues 4/4 at beginning class
2. Check off 4/7: practice course, 5 min
3. Mon. 4/10: (430-530 pm) round 1
 1. 6.5 makes first turn
 2. 7 half track in < 5 minutes
 3. 9 track in less than 2 minutes
4. CalDay Sat. April 22 @ UCB,
5. Quiz 5 on 4/11 on steering control
6. Brushless motor snubbing

Topics

- Backup brushed driver
- Notes from HW2
- SW notes
- Discrete time control conclusion (with sampling rate)
- Step response and P.I.D. intuition
- Feedforward steering control

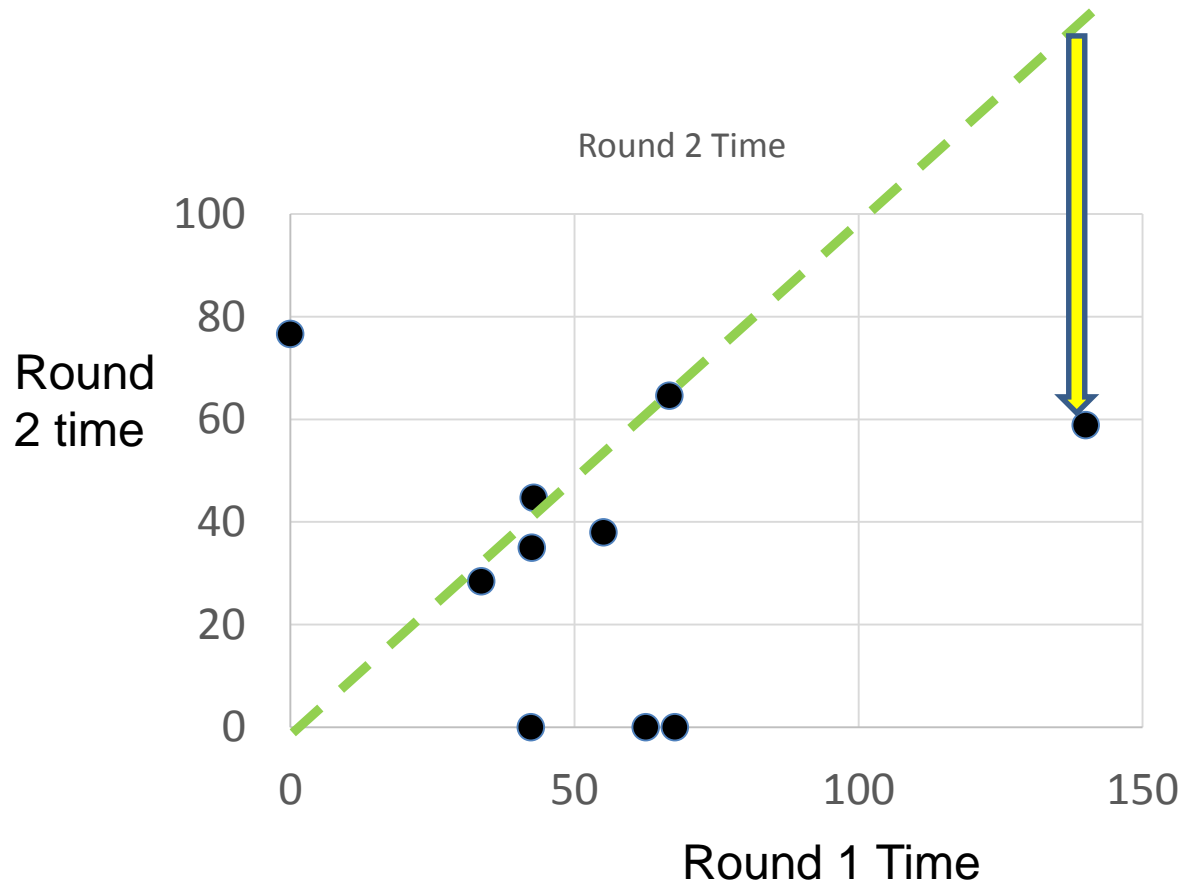
Better is the enemy of the good.

Optimism:

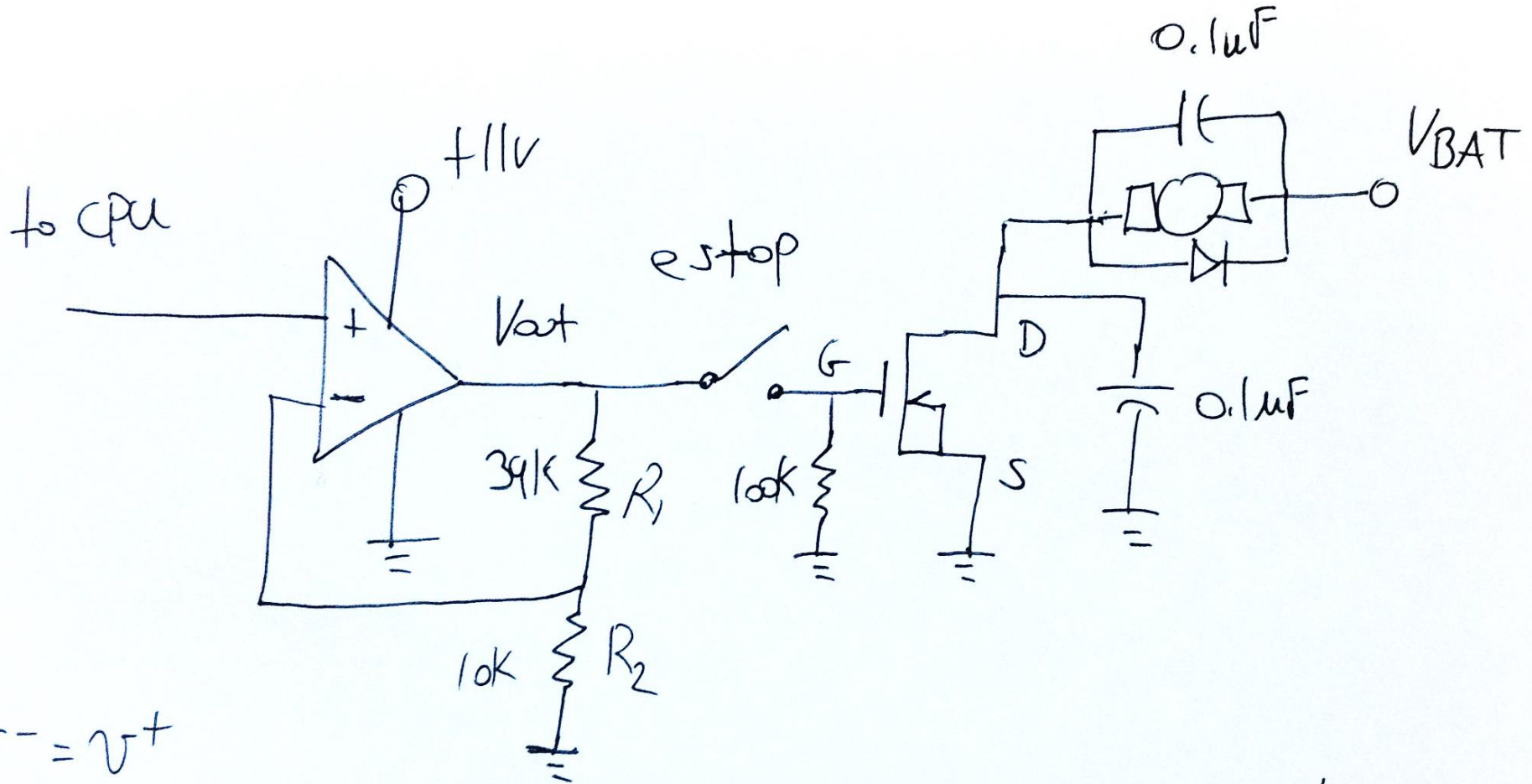
underestimate complexity+ overestimate ability

Previous Round 2 Results

- **2015:** 36.11, 36.29, 37.19, 38.83, 40.77, 44.44, 44.95, 47.95, 48.3, 49.77, 50.49, DNF
- **2016:** 28.45, 34.97, 37.99, 44.71, 58.87, 64.58, 76.6, DNF, DNF, DNF

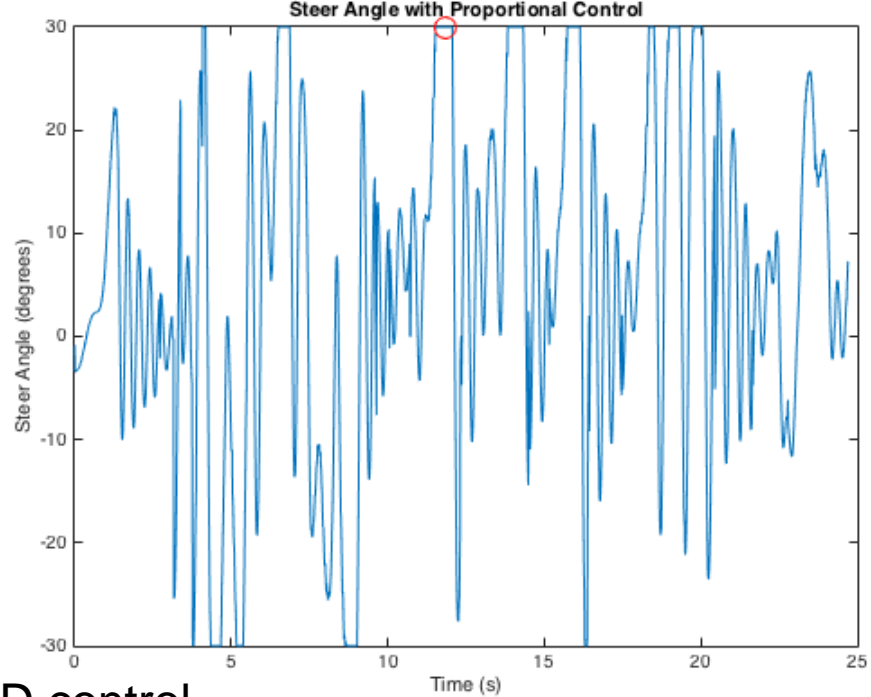
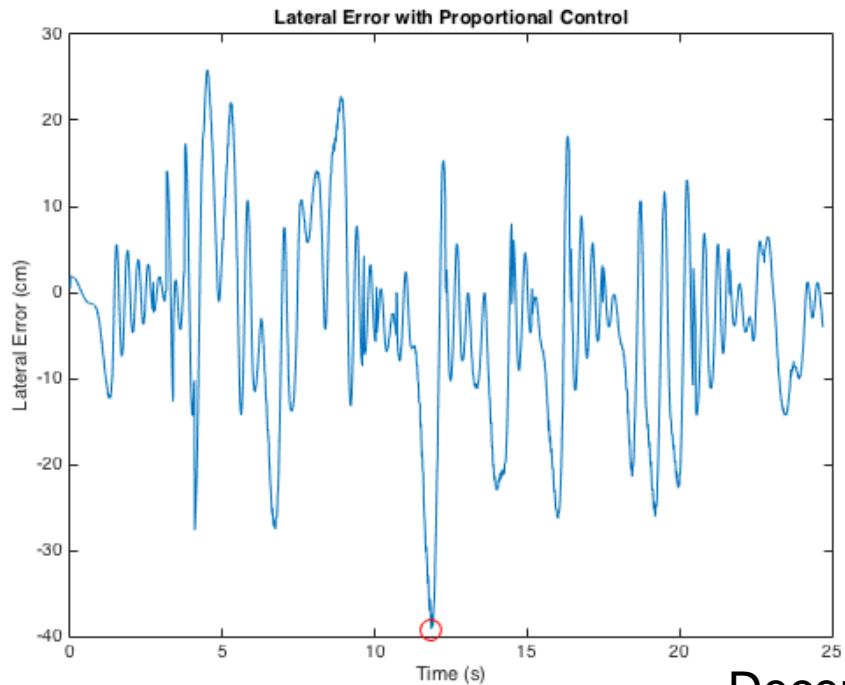


Backup brushed motor drive

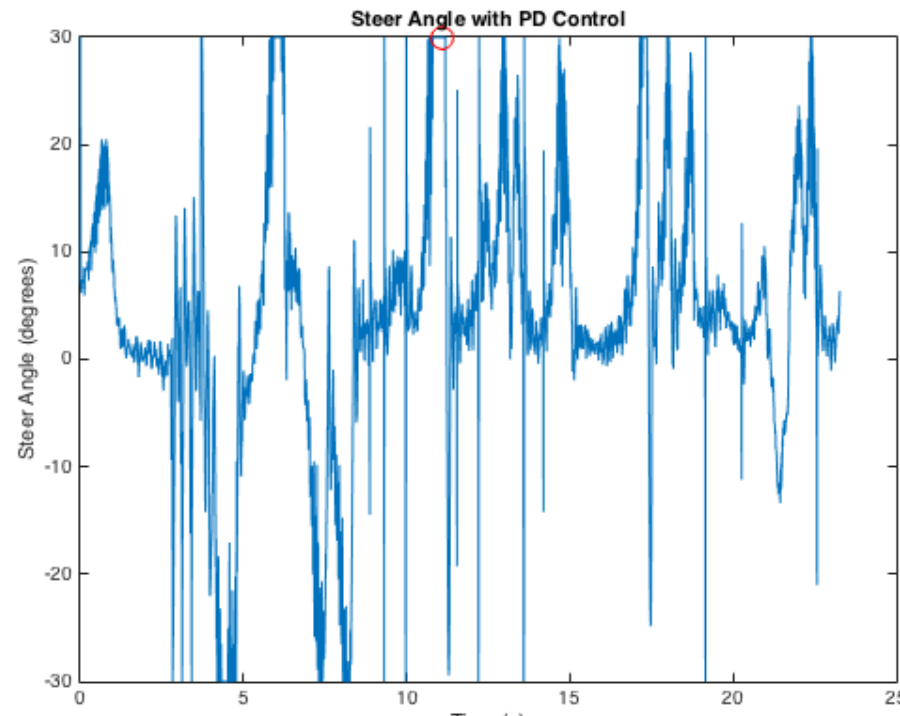
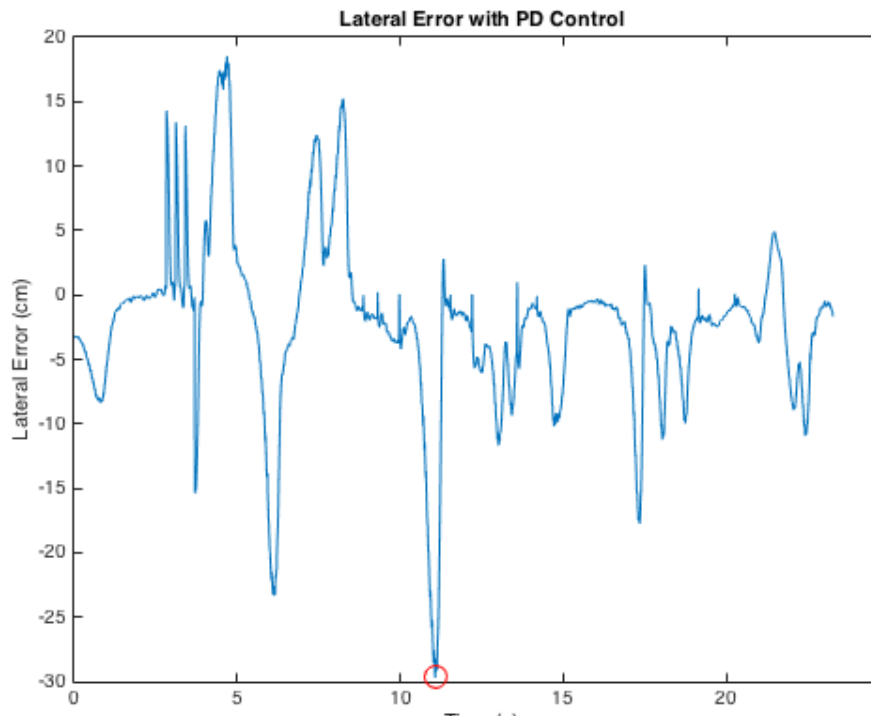


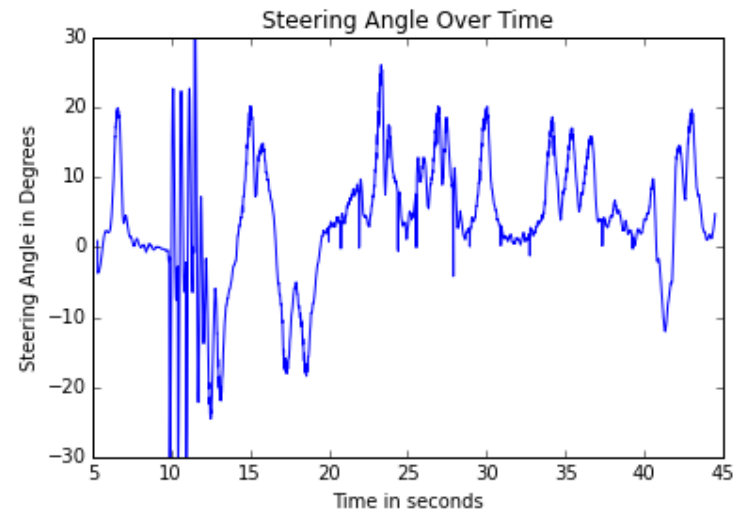
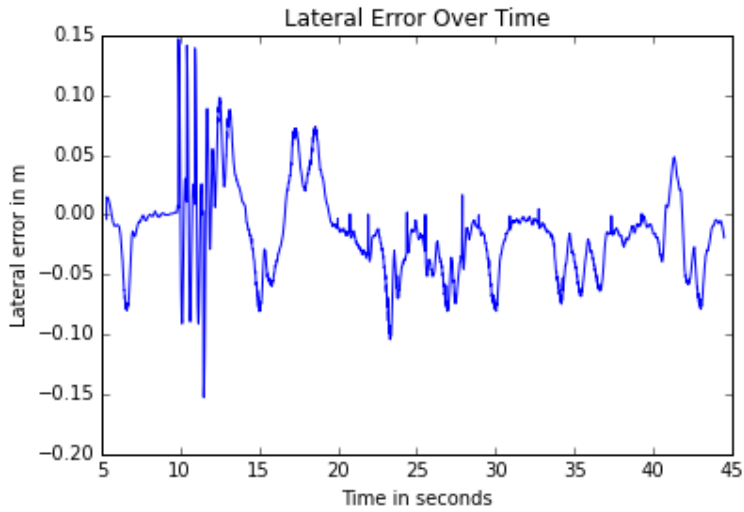
$$v^- = v^+$$

$$= V_{out} \cdot \frac{R_2}{R_1 + R_2} \Rightarrow V_{out} = \frac{R_1 + R_2}{R_2} v^+, \quad \frac{39k + 10k}{10k} \approx 4.9x$$

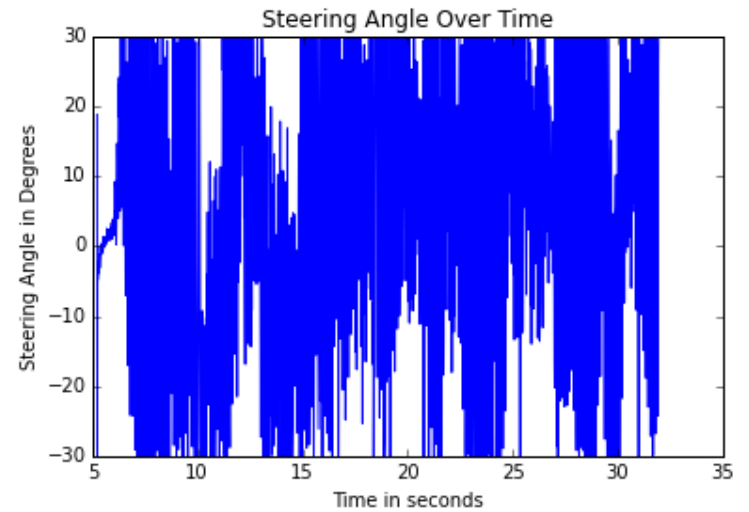
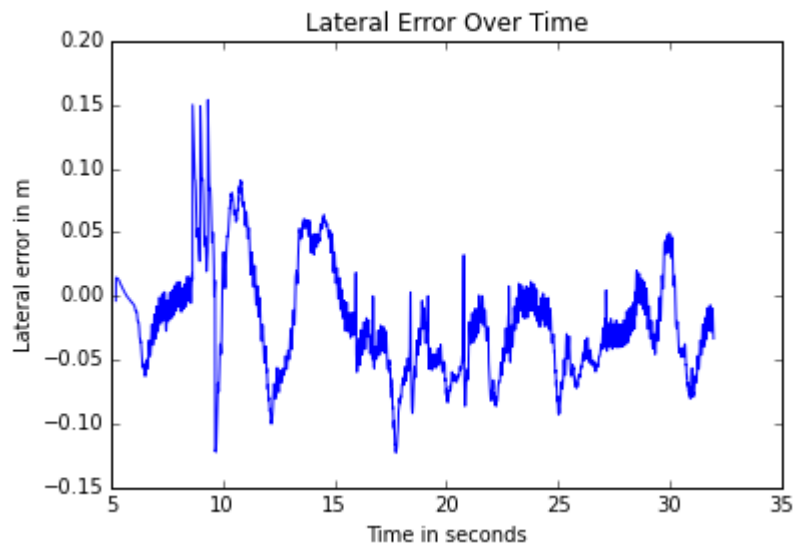


Decent PD control

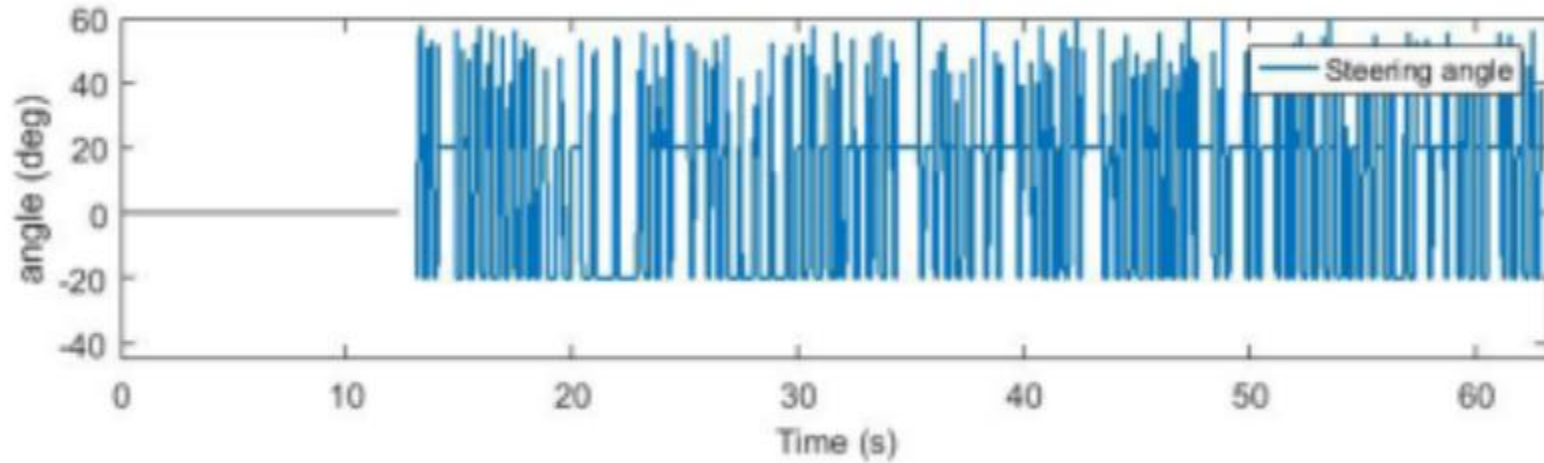




Too aggressive PD control



Angle Limit?



Best lap time 21.8 sec

Simulation notes

- What are other line tracking errors in addition to 128 pixel quantization?
- What are some practical limits on steering control?

10 Questions to Consider when Reviewing Code

Jacob Beningo

Embedded Systems Conference -2017

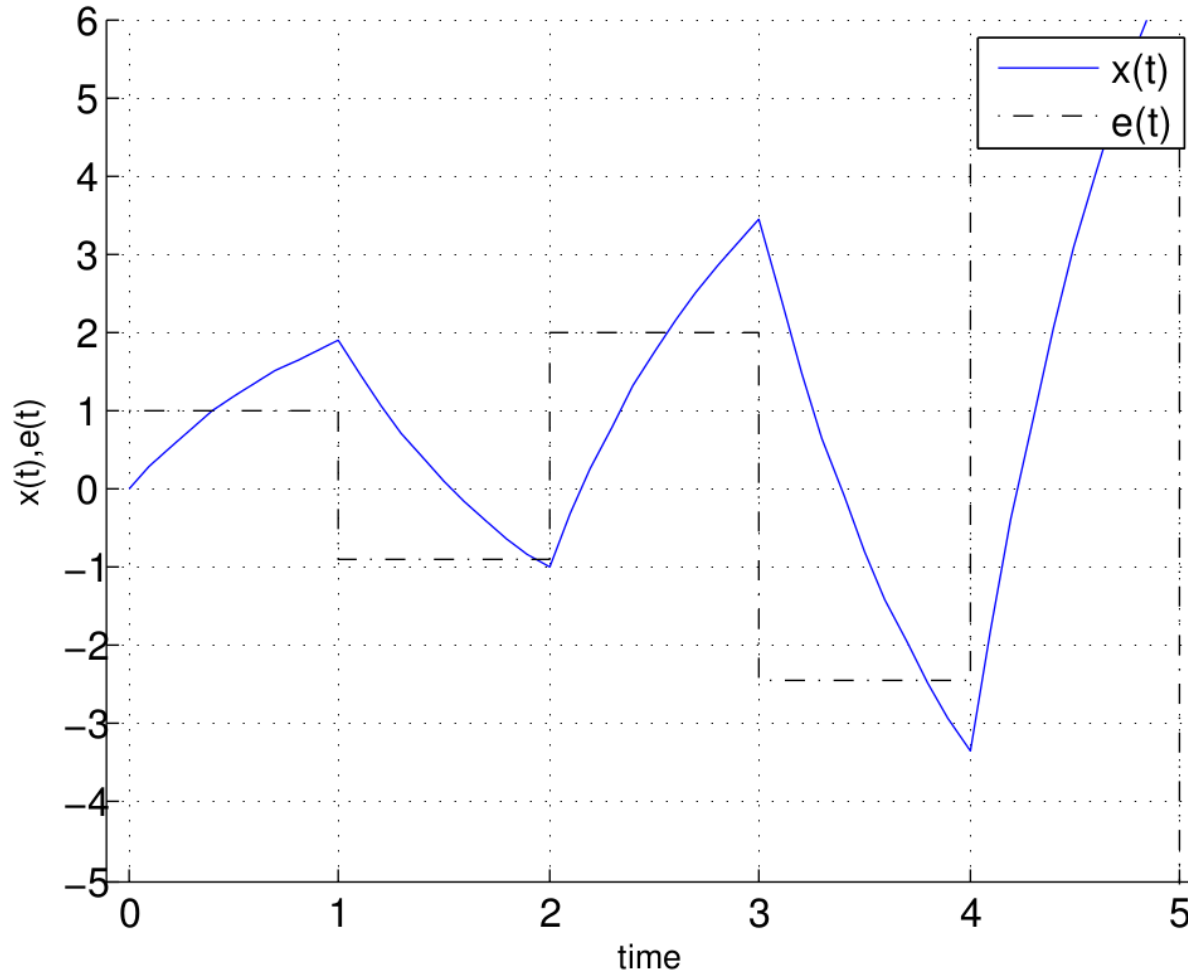
<https://www.designnews.com/electronics-test/10-questions-consider-when-reviewing-code/143583201956491?cid=nl.x.dn14.edt.aud.dn.20170329>

1. Does the program build without warnings?
2. Are there any blocking functions?
3. Are there any potential infinite loops?
4. Should this function parameter be const?
6. Has extern been limited with a liberal use of static?
7. Do all if ... else if ... conditionals end with an else?
8. Are assertions and/or input/output checks present?
9. Are header guards present? The guard prevents double inclusion of the #include directives.
10. Is floating point mathematics being used?

Discrete Time Control

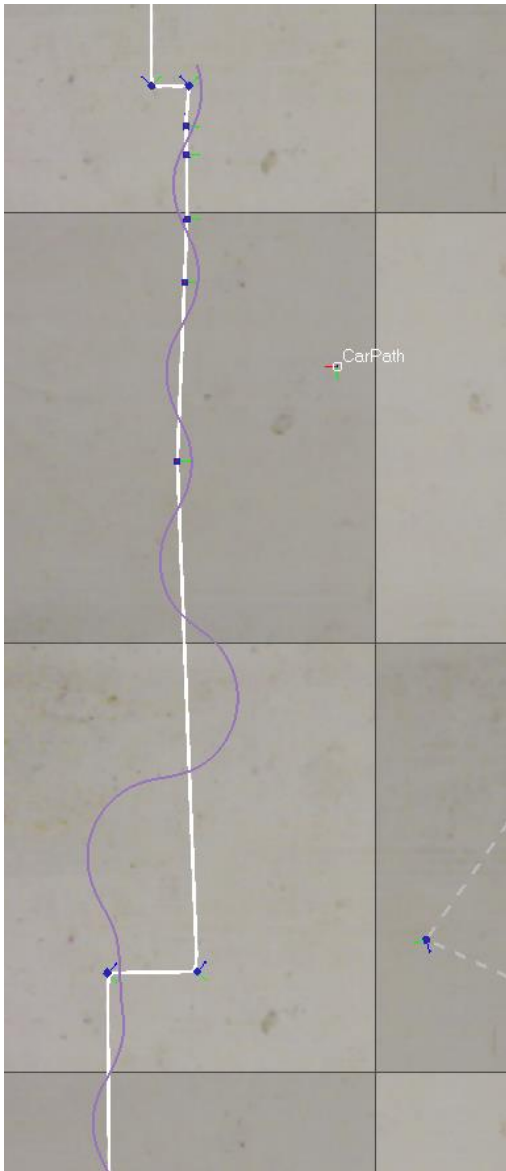
$$u[n] = k_p(r[n] - y[n])$$

Time Series Plot:unnamed

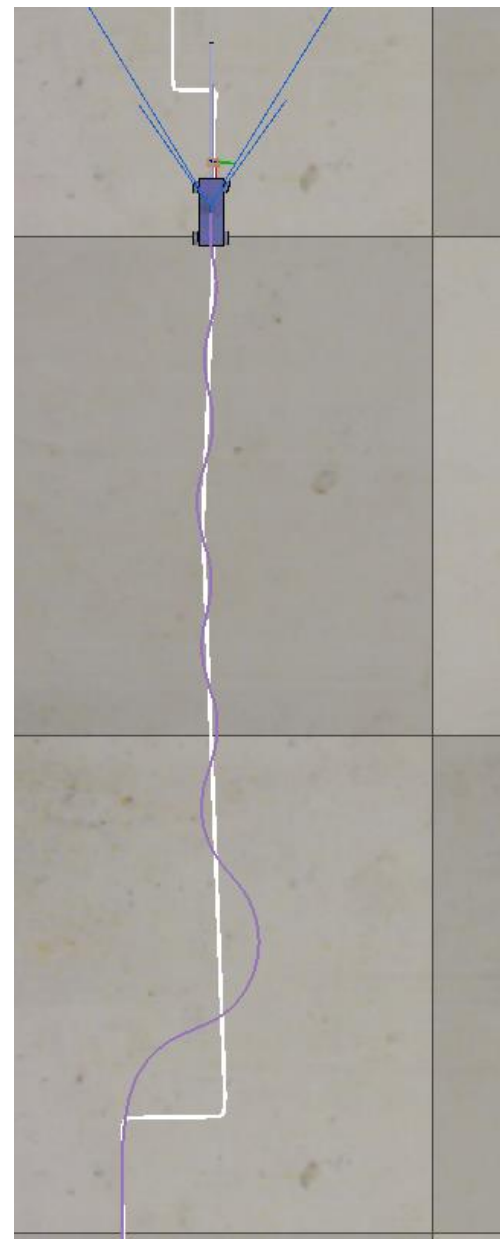


On board

Step response example

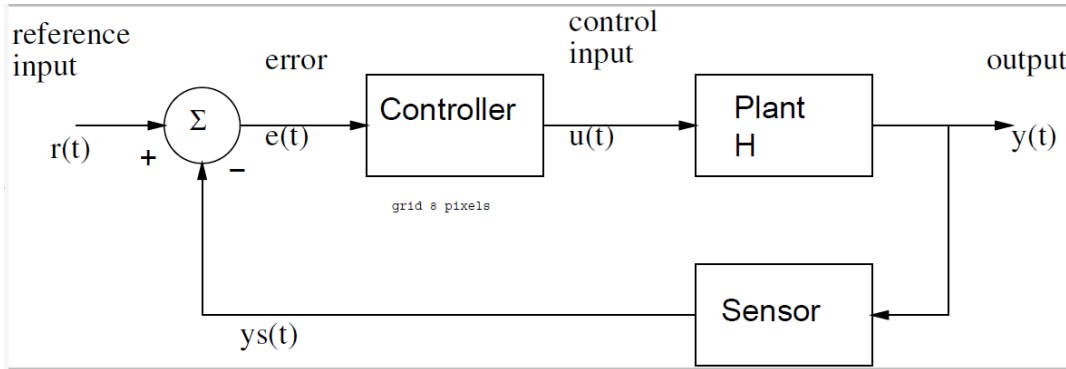


2 ms/s, boom 0.1 m, $k_p=1500$ deg/m

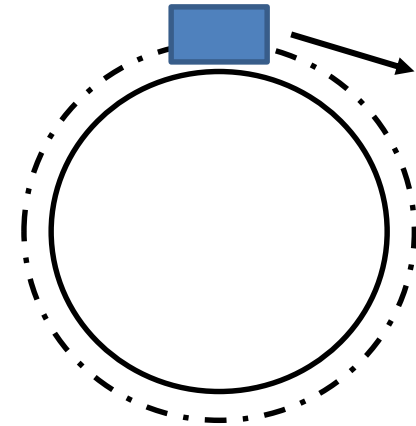


2 ms/s, boom 0.1 m, $k_p=500$ deg/m

Steering Control- feedforward



Note steady state error:
car follows larger radius



Proportional control:
 $r = 0$ (to be on straight track)
 $\delta = u = k_p * e$

Proportional+derivative

P+I+D

On board

