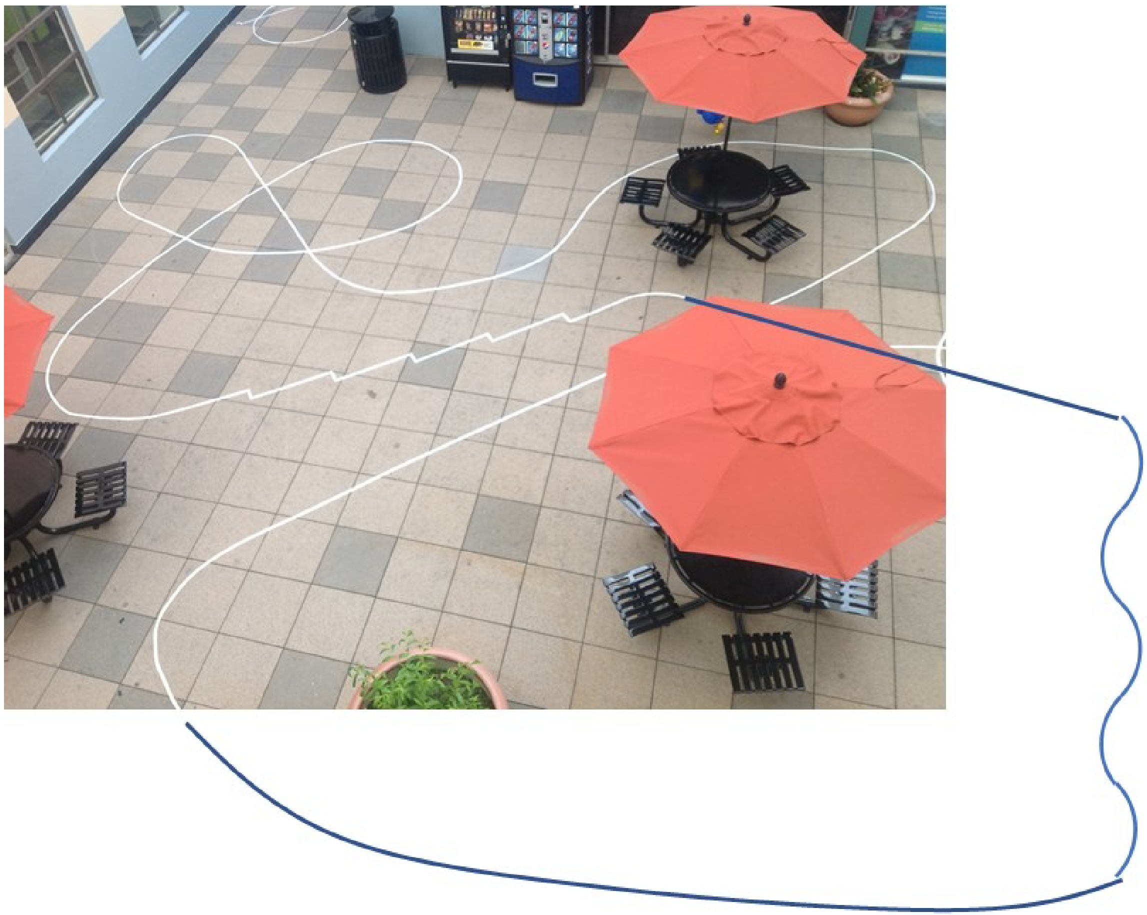
EECS192 Lecture 11
Control
April 5, 2021

Topics

- Checkpoint 8: Step+ telemetry
- Progress report (due Fri 4/9 8 pm)
- Round 1 (simulation) in class 4/20 (was 4/13)
  - Submit Python 3.7 code, we run.
- Round 2 (live/hardware) in class 4/27
- ESP32 A/D Issues
- HW 2 Simulation notes
- Quiz 4 on steering 4/13
- Control Fundamentals
- Robustness (revisited)

Planned Cory Courtyard Track

Thu and Friday 2-4 pm, supervised by staff member David Au.
All Covid protocols to be followed.
Details on Piazza when set up. Track avail no later than April 15 (see signup for possible earlier times with Andrew)



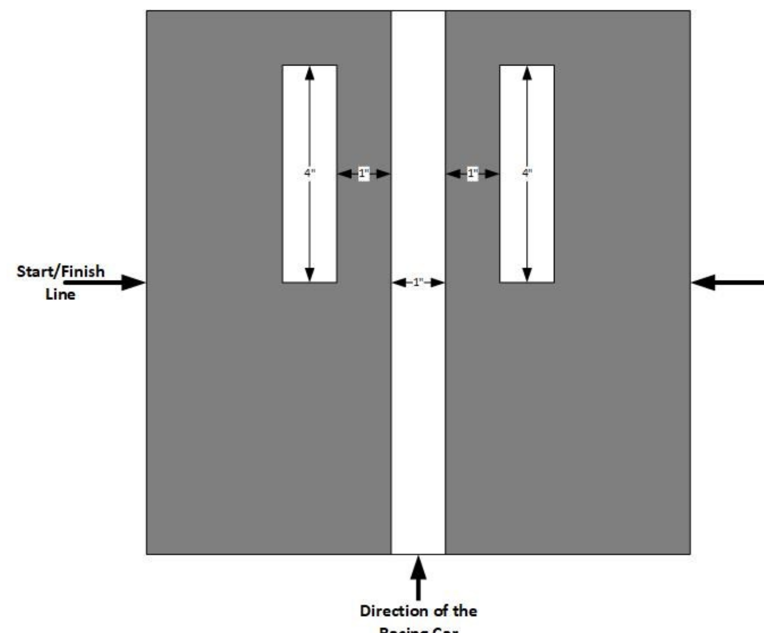
All curves minimum radius 3 feet (bigger radius is fine)
(use 3 foot string and mark circle with chalk)

Checkpoint 8: Stopping (Fri 4/9)

Set up a straight or curved track with length sufficient for car to accelerate to (ideally) 2 m/sec or better and stop when it sees the stop pattern.

Checkoff Procedure

- C8.1 Show car driving on track and then doing emergency stop from remote command.
- C8.2 Show car driving on track and doing emergency stop when it crosses NATCAR stop marking of parallel lines. Natcar Finish Line
- C8.3 Show telemetry plot of speed and other relevant parameters.
- C8.4 All members must fill out the checkpoint survey before the checkoff close. Completion is individually graded.



Progress Report Due Fri 4/9 (8 pm)

https://inst.eecs.berkeley.edu/~ee192/sp21/docs/progrpt.pdf

EECS192 Lecture 11
Control
April 5, 2021

Topics

- Checkpoint 8: Step+ telemetry
- Progress report (due Fri 4/9 8 pm)
- Round 1 (simulation) in class 4/20 (was 4/13)
  - Submit Python 3.7 code, we run.
- Round 2 (live/hardware) in class 4/27
- ESP32 A/D Issues
- HW 2 Simulation notes
- Quiz 4 on steering 4/13
- Control Fundamentals
- Robustness (revisited)

Virtual Race 4/20 (more details to follow)

Goal: every team ``competes'' with control alg. on same track.

- Pre-submit your code on bCourses. **Only submit your updated controller.py**
- Your code must work on Python 3.x.
- You must be using the provided carInterface.py.
- You must not modify the call signature to SimulationAssignment.__init__, setup_car, and control_loop. We will be using a different __main__ block to run your code.
- You have 3 minutes of simulation time, with requests for re-runs handled in round robin order.
- Your code must run as a self-contained unit:
- Your code may not do disk operations (outside the csvfile.writerow) or network operations.
- If using memorization, your code must automatically (and without being restarted, or with command-line arguments, or any manual intervention in general) detect the end of a run and continue the next lap with whatever memorization-based algorithms you're using.
- As per NATCAR rules, you may not pre-program in data about the track.

DRAFT Round 1 Deductions
details to be added

-0.3 points	Twitchy steering
-0.3 points	Goes wide on curves
-0.3 points	Oscillatory step response
-0.3 points	Slower than 2 m/s
-0.5 points	Hits cone(s)

Draft Spec. for Physical Race 4/27 in class

- Local tracks (depending on what team has available) (maybe Cory Courtyard for students in Berkeley?)
- Metrics:
  - Accel-decel (straight line- drag race with stop line)
  - Step speed with overshoot penalty
  - Speed and error on S –curve
  - Speed and error on Fig. 8

Better is the enemy of the good.

Optimism:

underestimate complexity+

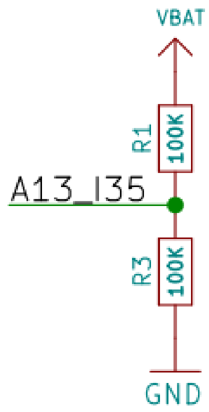
overestimate ability

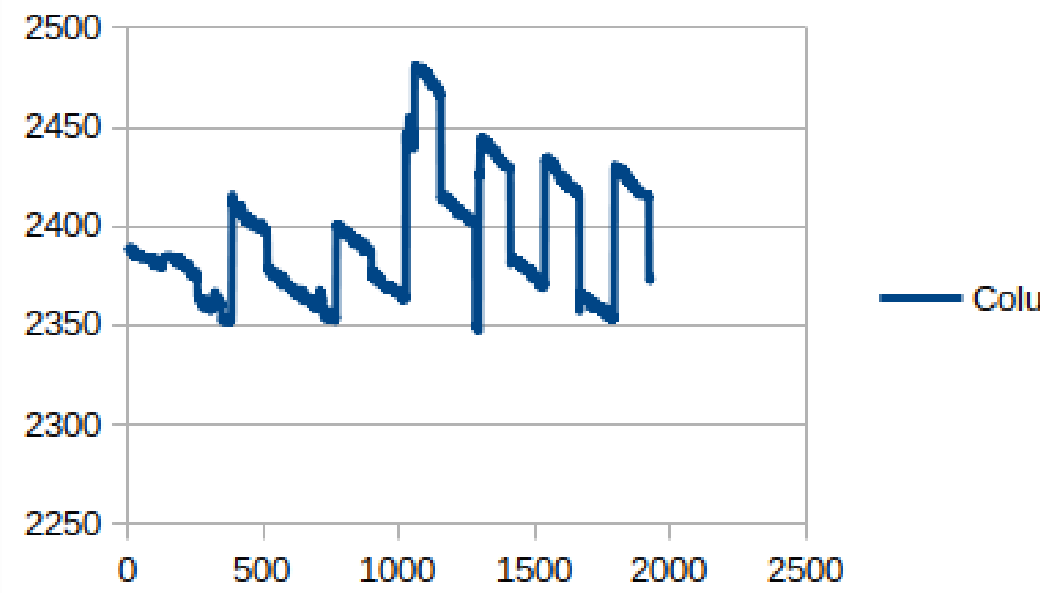
EECS192 Lecture 11
Control
April 5, 2021

Topics

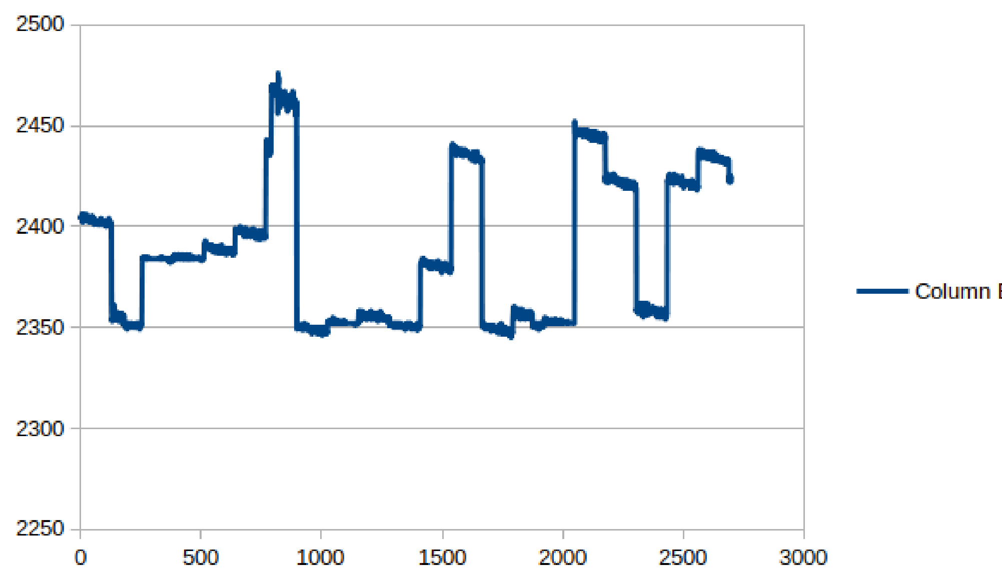
- Checkpoint 8: Step+ telemetry
- Progress report (due Fri 4/9 8 pm)
- Round 1 (simulation) in class 4/20 (was 4/13)
  - Submit Python 3.7 code, we run.
- Round 2 (live/hardware) in class 4/27
- ESP32 A/D Issues
- HW 2 Simulation notes
- Quiz 4 on steering 4/13
- Control Fundamentals
- Robustness (revisited)

ESP32 A/D Issues (from Lec 10)

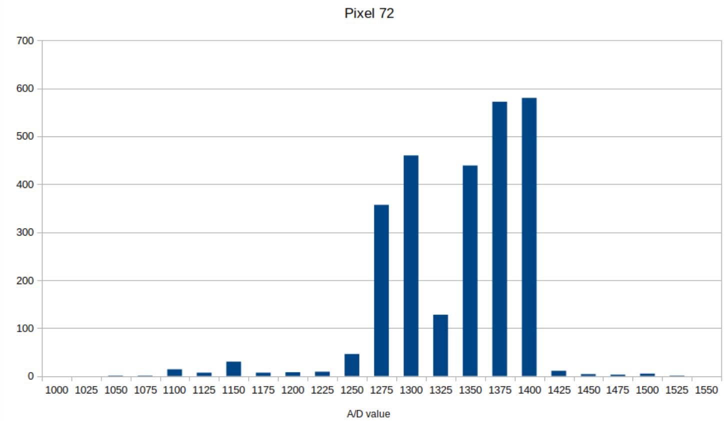


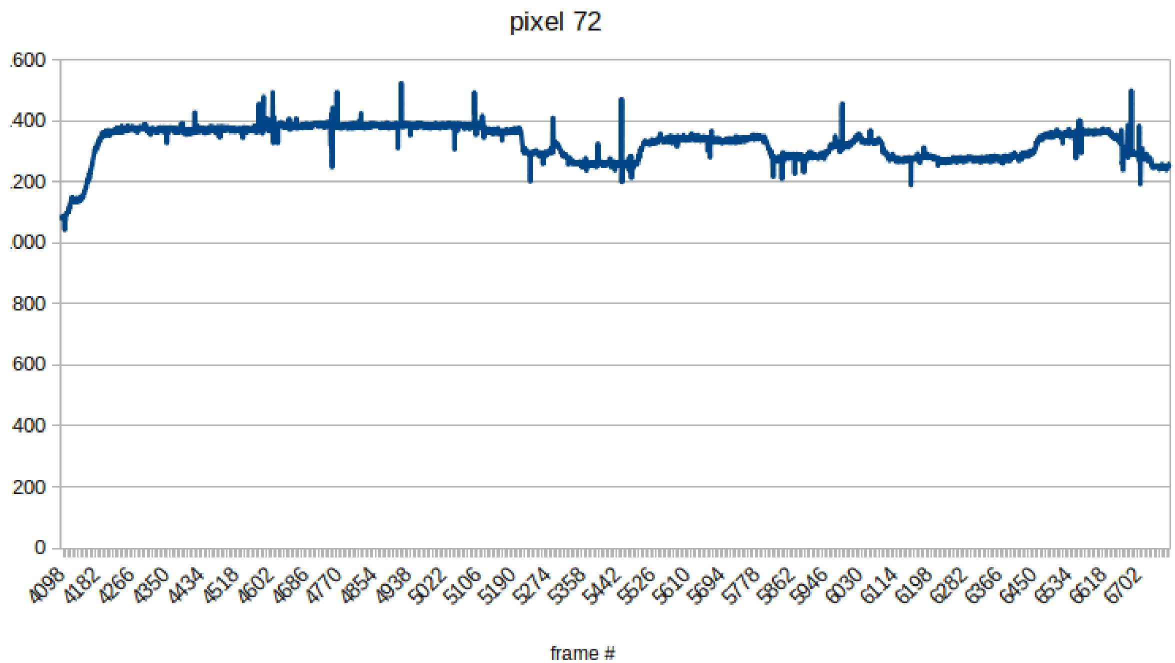


static const adc_channel_t channel = ADC_CHANNEL_7;
128 reads are done at 51 us for
`adc1_get_raw()`
and
128 reads are done at 12.2 us for
`local_adc1_read()`
Then there is a 5 second delay before reading the A/D again. A lot of drift is noticeable.



A/D Issues- camera





Frame read every 30 ms
Read ~2500 frames (~75 sec)

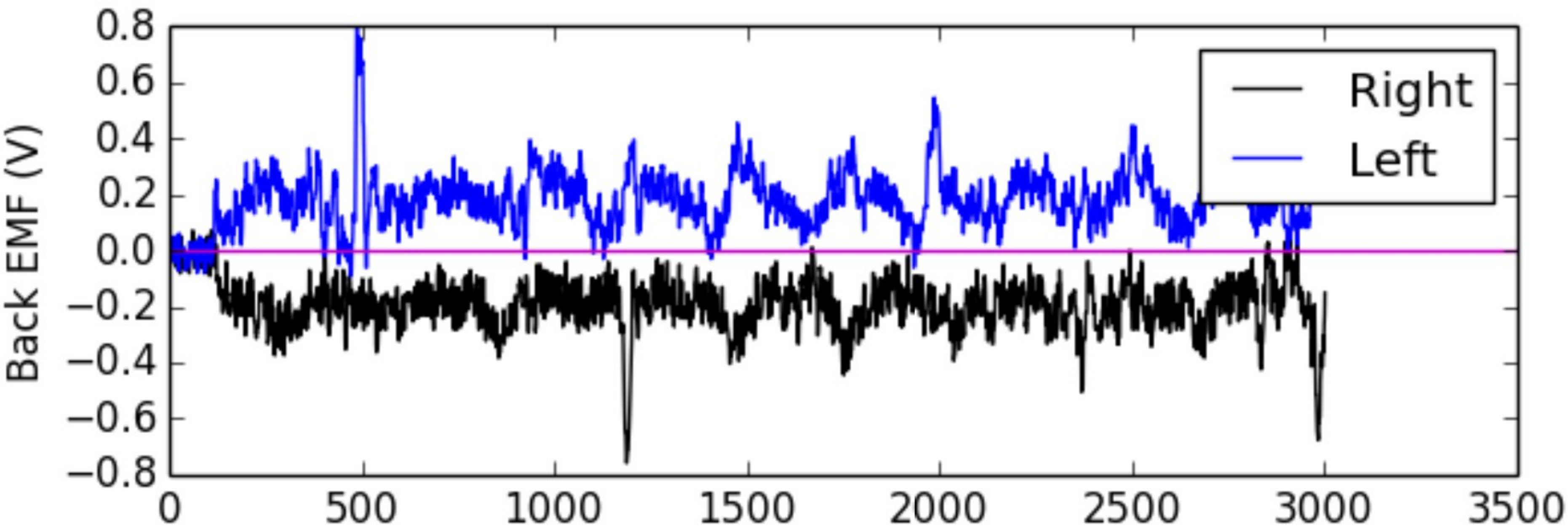
128 reads are done in 2.5 ms using `local_adc1_read()` inside interrupt (about 20 us per read)
Problems:
Problems with signal:
1) Signal has two levels ~1400 and ~1300 and shifts center on order of 10 seconds
(could be handled by moving average if using frame subtraction)
2) Signal has ``impulsive'' spikes with range of +-100 (median filter helps here)
3) Signal drifts for first 200 frames until stabilizing near 1400 (warm up time???)

Digital Filtering

- Moving average
  - $y1[n] = (y[n-2]+y[n-1]+y[n])/3$
- Median filter (outlier rejection)
  - median(7,10,11,12,16,200,205)=?
- Notch filter (mechanical vibration)
  - $y[n] = (x[n-2]+2x[n-1]+x[n])/4$
- Model based filtering (or Kalman filter)

Moving Average vs. Median Filter

Example: motor brush noise, back EMF measurement



Time (ms)

{0,2,-1,4,0,2,1,1,20,1,0,2} ➔

{0,2,-1,2,0,1,1,1,1,1} 3 element median filter

{0,2, 0.3,1.7, 2, 1,1.3, 7.3, 7.3, 7,1…} 3 elem MA

EECS192 Lecture 11
Control
April 5, 2021

Topics
- Checkpoint 8: Step+ telemetry
- Progress report (due Fri 4/9 8 pm)
- Round 1 (simulation) in class 4/20 (was 4/13)
  - Submit Python 3.7 code, we run.
- Round 2 (live/hardware) in class 4/27
- ESP32 A/D Issues
- HW 2 Simulation notes
- Quiz 4 on steering 4/13
- Control Fundamentals
- Robustness (revisited)

HW2 Notes- slew rate limit 2021

Team	**Speed (m/s)**	**Error**
1	3.22	35 cm
2	3.0	~25 cm
3	3.0	22 cm
4	3.0	20 cm
6	3.0	25 cm
7	3.0	20 cm

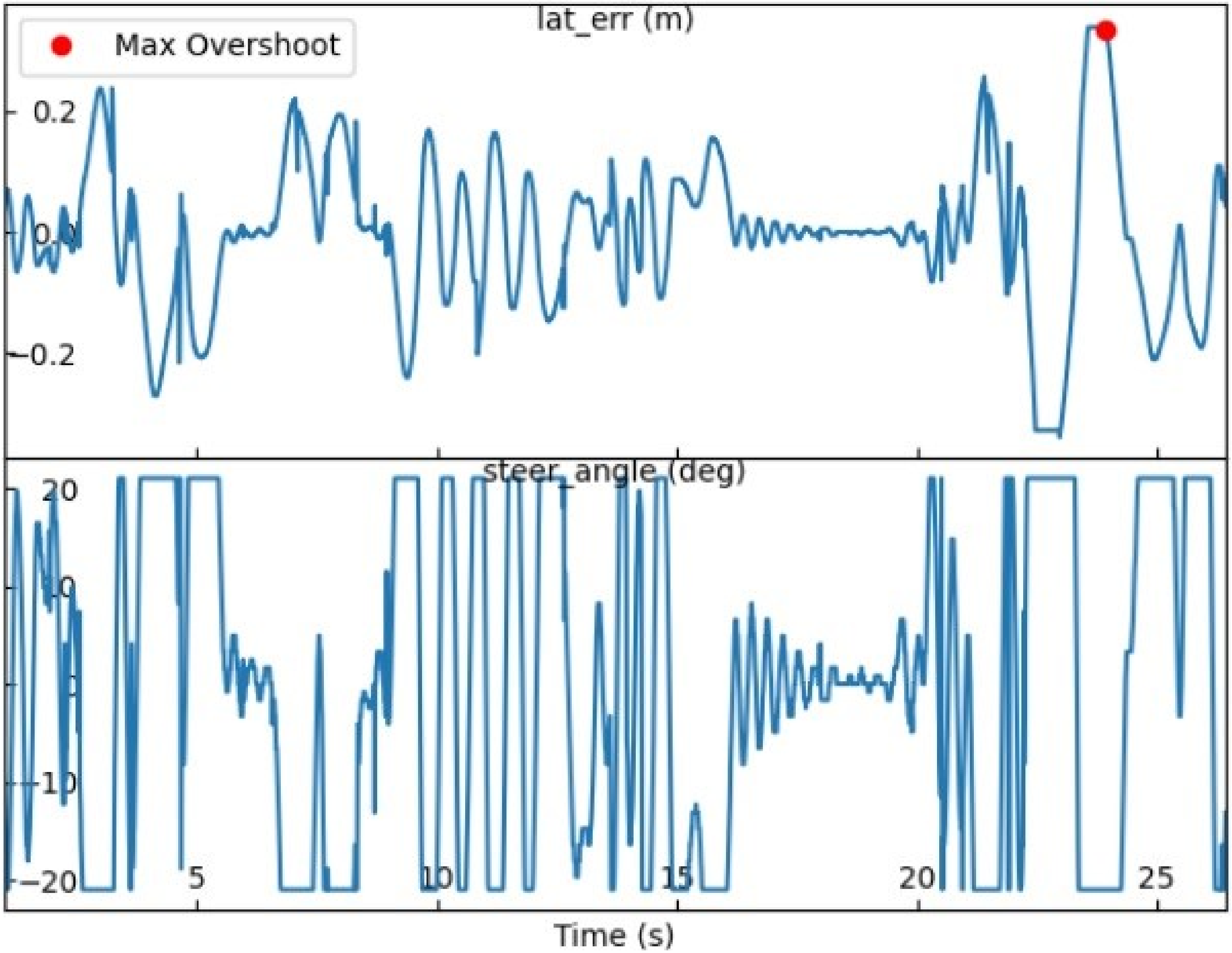
Note cone is at: 15” = 38 cm

HW2 Notes- slew rate limit 2020

Team	Speed (m/s)	Error
1	3	29 cm
2	2.7	32 cm
3	3.5	30 cm
4	3.3	30 cm
5	3	24 cm
6	2.7	27 cm
7	2.5	23 cm
8	2.7	27 cm
10	2.5	32 cm
11	2.6	13 cm

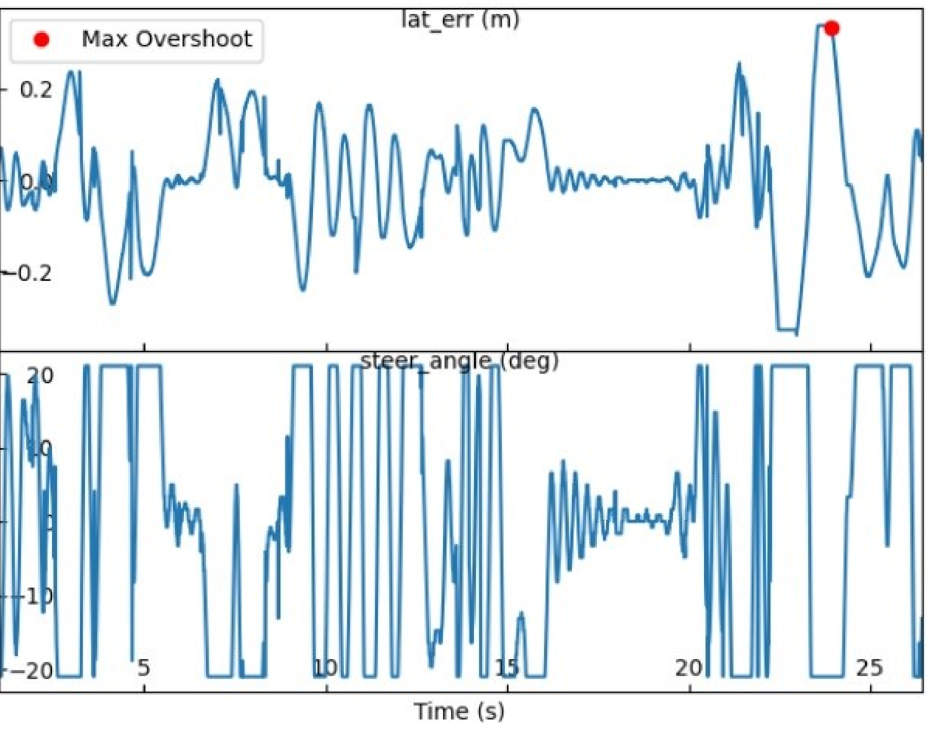
Note cone is at: 15” = 38 cm

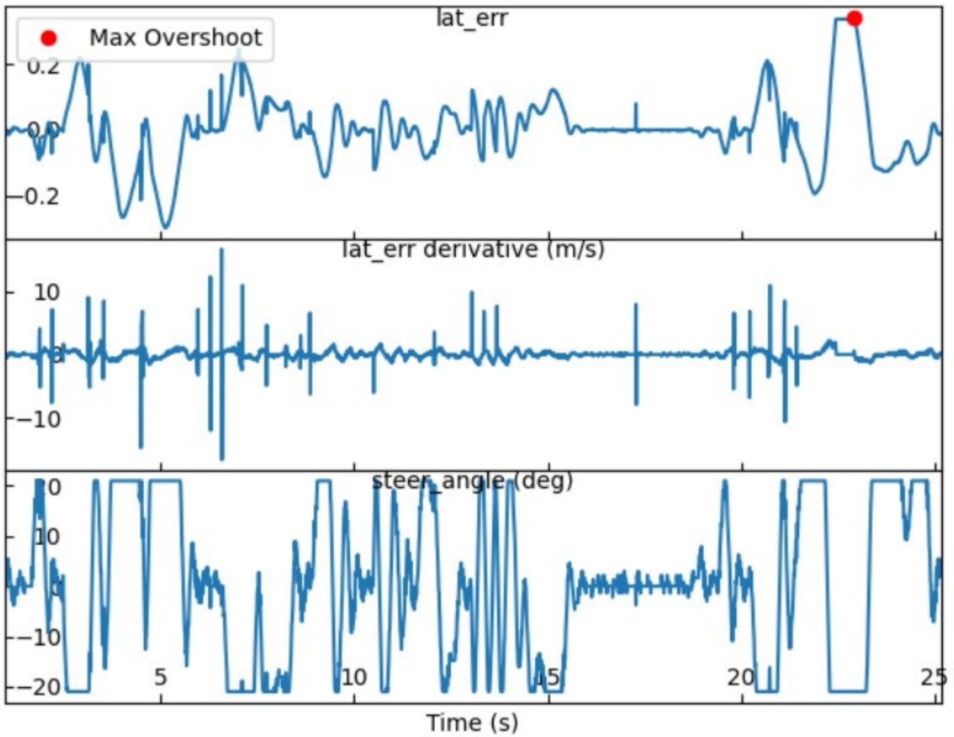
HW2 Sim P control



Issues:
Steering control is ``bang-bang'' due to saturation,
outside well-behaved linear approximation

HW2 Sim PD+slew limit control

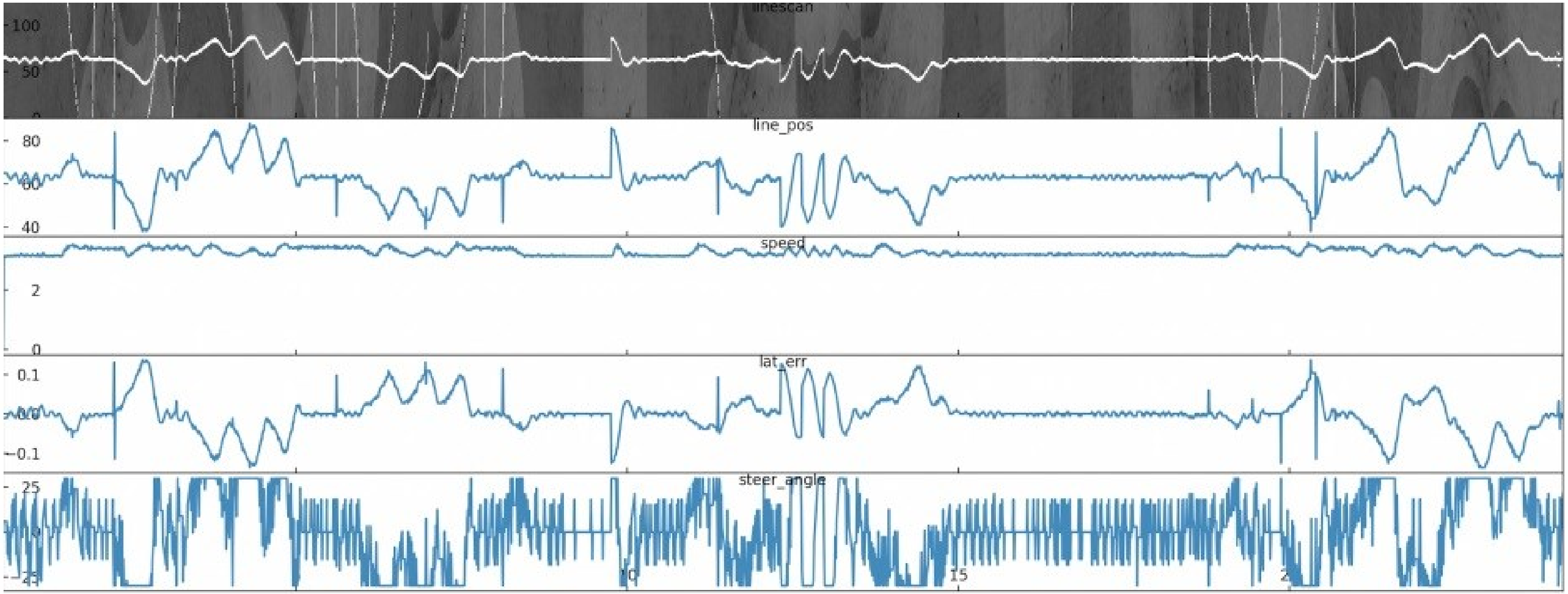




(kp too large)

Bang-bang can cause extra sliding (think driving on ice), which requires extra corrections, …

PD control



Not how wiggles in lateral error get magnified to rapid steering angle changes
(kd too large, may need smoothing on lateral velocity)
Twitch on steering leads to ripple on lateral error, leads to larger ripple on lateral velocity

Simulation notes

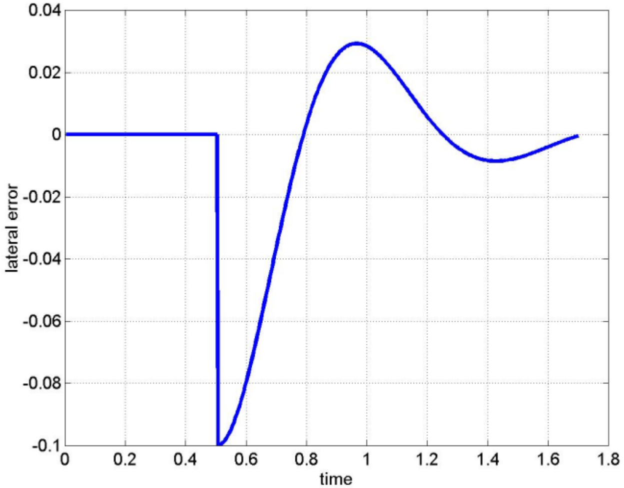
- What are other line tracking errors in addition to 128 pixel quantization?
- What are some practical limits on steering control?

EECS192 Lecture 11
Control
April 5, 2021

Topics

- Checkpoint 8: Step+ telemetry
- Progress report (due Fri 4/9 8 pm)
- Round 1 (simulation) in class 4/20 (was 4/13)
  - Submit Python 3.7 code, we run.
- Round 2 (live/hardware) in class 4/27
- ESP32 A/D Issues
- HW 2 Simulation notes
- Quiz 4 on steering 4/13
- Control Fundamentals
- Robustness (revisited)

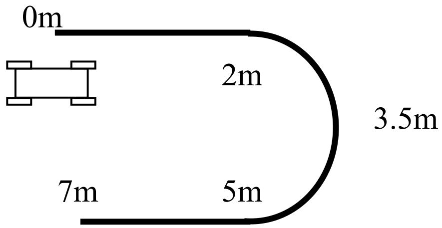
Quiz 4 steering preview

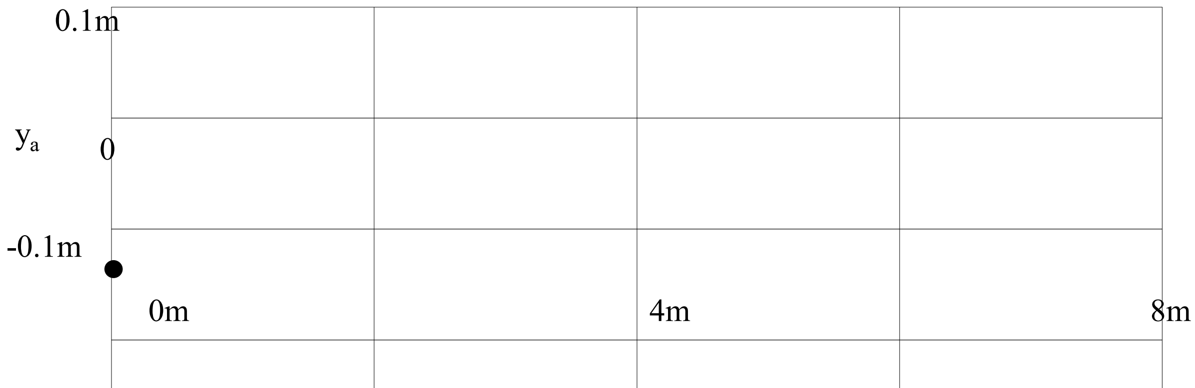


Measured step response (as function of time) on a 10 cm step at 2 m/sec using P+D controller is given as shown above.

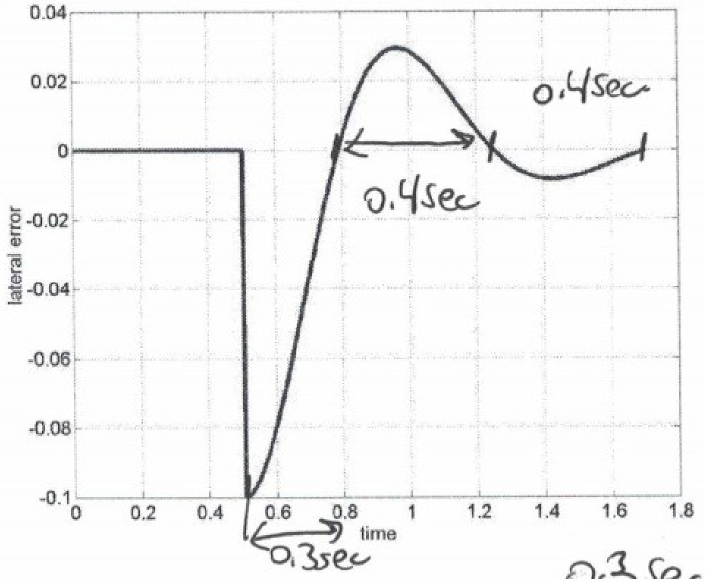
Note per cent overshoot is 30%, and it takes 0.3 sec after crossing the line to recross.

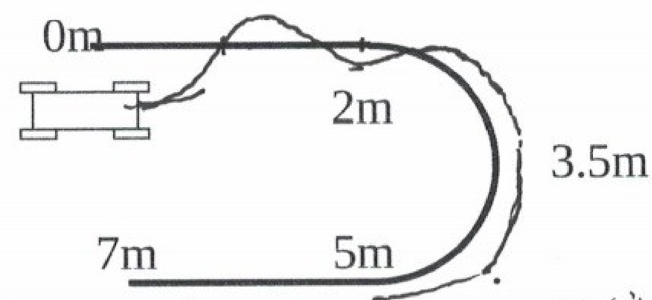
For initial car position as shown on path, car speed 2 m/sec, sketch lateral error y_a as function of longitudinal **distance**. Numbers are path length in m (min radius turn).

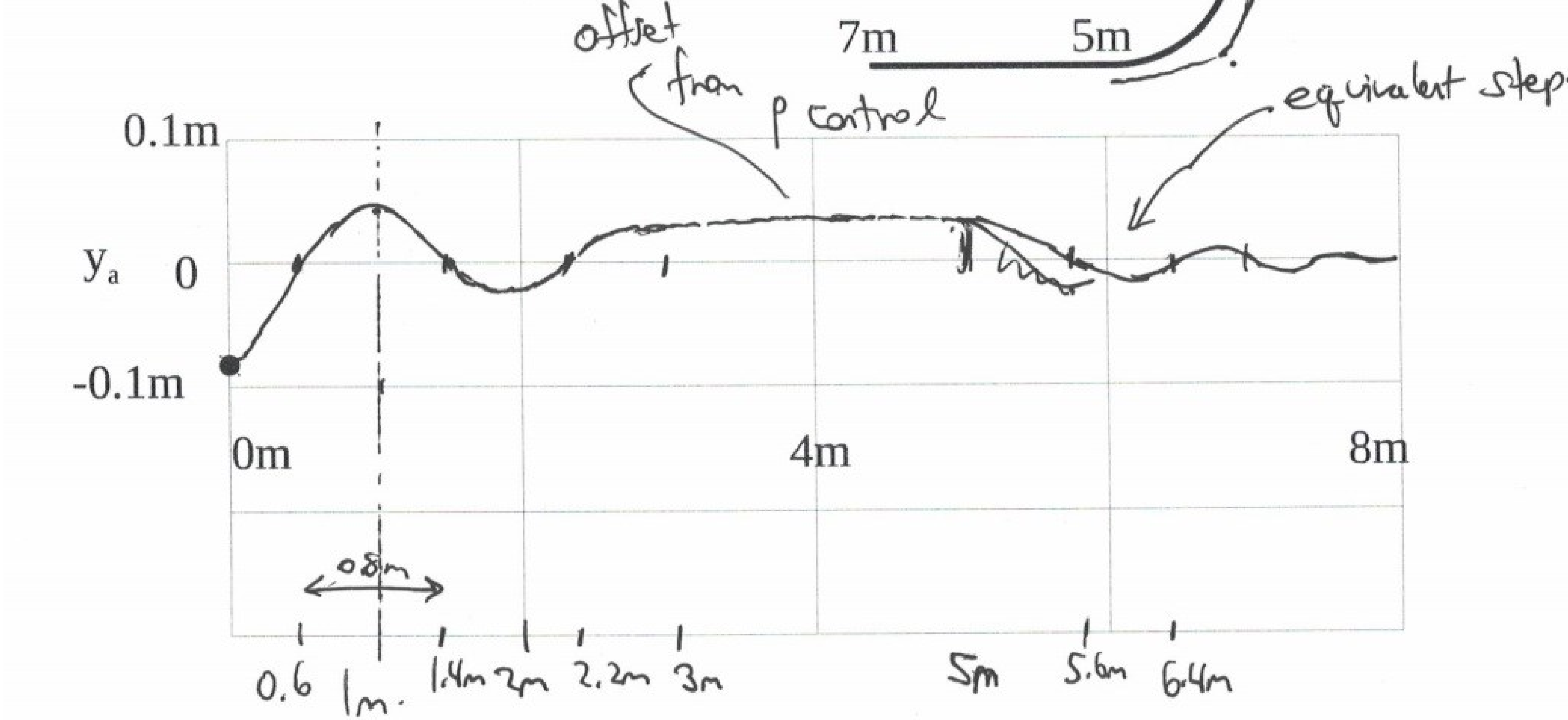




Quiz 4 steering preview





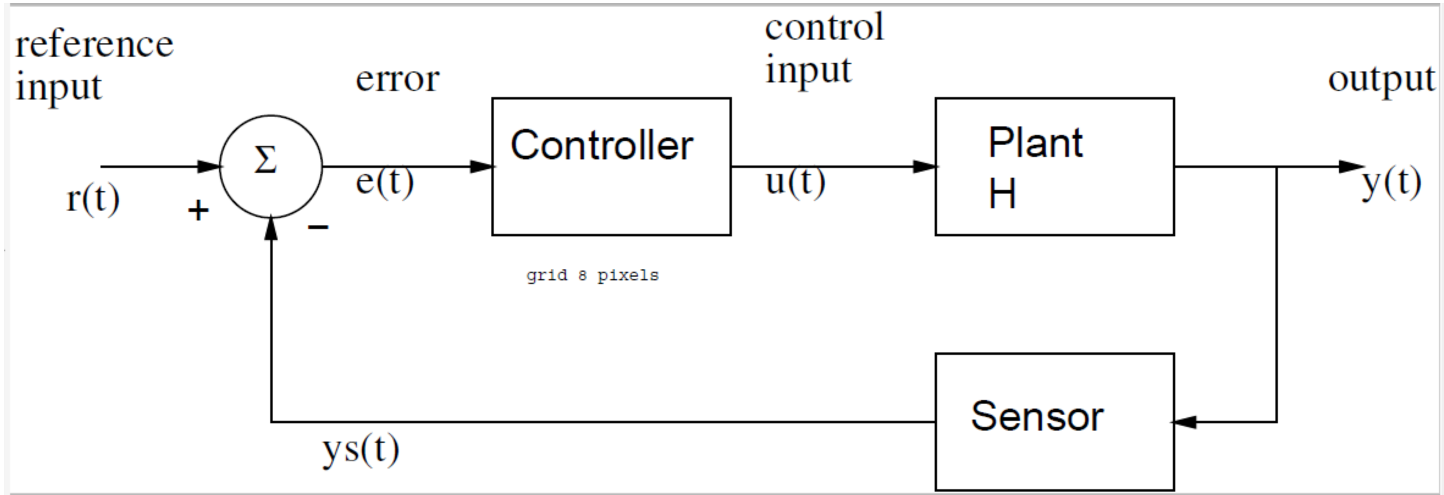


EECS192 Lecture 11
Control
April 5, 2021

Topics

- Checkpoint 8: Step+ telemetry
- Progress report (due Fri 4/9 8 pm)
- Round 1 (simulation) in class 4/20 (was 4/13)
  - Submit Python 3.7 code, we run.
- Round 2 (live/hardware) in class 4/27
- ESP32 A/D Issues
- HW 2 Simulation notes
- Quiz 4 on steering 4/13
- Control Fundamentals
- Robustness (revisited)

Control Synopsis



State equations: $\dot{x}(t) = ax(t) + bu(t)$

Output equations: $y(t) = cx(t) + du(t)$

Control Law (P): $u(t) = k_p e(t) = k_p(r(t) \;-\; y(t)).$

Control Synopsis

Control Law (P): $u(t) = k_p e(t) = k_p(r(t) \; - \; y(t)).$

New state equations:

$$\dot{x} = ax + bk_p e(t) = ax + bk_p(r - x) = (a - bk_p)x + bk_p r.$$

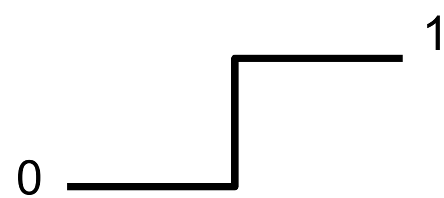
Zero Input Response (non-zero init condx, r(t)=0):

$$x(t) = x(0)e^{(a-bk_p)t} \quad for \quad t \geq 0.$$

a'=a – b k_p b' = b k_p

Total Response (non-zero init condx) by convolution:

$$x(t_o) = e^{a' t_o} x(0) + \int_0^{t_o} e^{a'(t_o - \tau)} b' r(\tau) d\tau \; . \qquad (10)$$

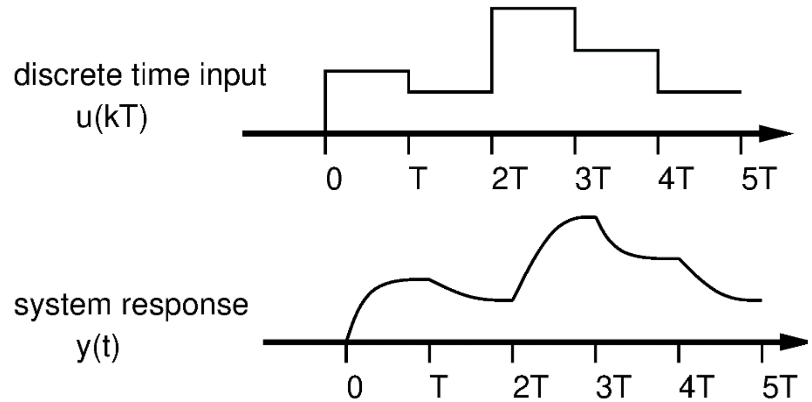


Step Response (zero init condx) by convolution:

$$x(t_o) = b' \int_0^{t_o} e^{a' t_o} e^{-a' \tau} d\tau \;=\; \frac{-b' e^{a' t_o}}{a'} e^{-a' \tau}|_0^{t_o} \;=\; \frac{b'}{a'}(1 - e^{-a' t_o}) \; . \qquad (11)$$

Control Synopsis- Discrete Time

Superposition of Step Responses



$$x((k+1)T) = e^{a(k+1)T}x(0) + e^{a(k+1)T}\int_0^{(k+1)T} e^{-a\tau}bu(\tau)d\tau \ . \qquad (15)$$

$$x(kT) = e^{akT}x(0) + e^{akT}\int_0^{kT} e^{-a\tau}bu(\tau)d\tau \ . \qquad (14)$$

$$x((k+1)T) = e^{aT}x(kT) + e^{a(k+1)T}\int_{kT}^{(k+1)T} e^{-a\tau}bu(\tau)d\tau \ = e^{aT}x(kT) + \int_0^{T} e^{a\lambda}bu(kT)d\lambda \ , \ (16)$$

Control Synopsis- Discrete Time

$$G(T) \equiv e^{aT} \quad \text{and} \quad H(T) \equiv b \int_0^T e^{a\lambda} d\lambda \ . \tag{17}$$

State equations:

$$x((k+1)T) = G(T)x(kT) + H(T)u(kT) \tag{18}$$

Output equations:

$$y(kT) = Cx(kT) + Du(kT) \ . \tag{19}$$

Total Response (non-zero init condx) by convolution:

$$x(k) = G^k x(0) + \sum_{j=0}^{k-1} G^{k-j-1} H u(j) \ . \tag{23}$$

Control Synopsis- Discrete Time

Control Law (P): $U(kT) = k_p\,[r(kT) - x(kT)]$

New state equations:

$$x((k+1)T) = G(T)x(kT) + H(T)k_p(r(kT) - x(kT)) = [G - Hk_p]x(kT) + Hk_p r(kT)\ . \quad (24)$$

$$x((k+1)T) = [e^{aT} + \frac{k_p}{a}(1 - e^{aT})]x(kT) + Hk_p r(kT) \ = \ G'x(kT) + Hk_p r(kT) \ . \quad (25)$$

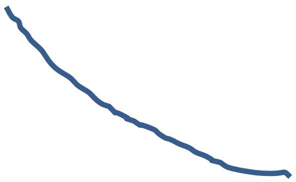
For stability:

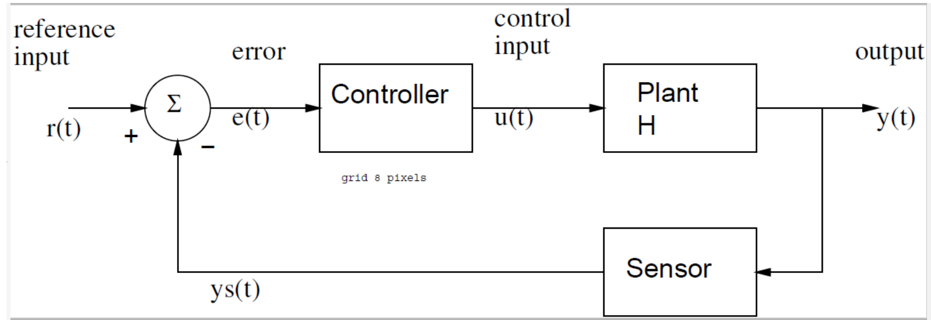
$$|e^{aT} - \frac{k_p}{a}(e^{aT} - 1)| < 1. \quad (26)$$

Notes: stability depends on gain **and** T!

Discrete Time Control

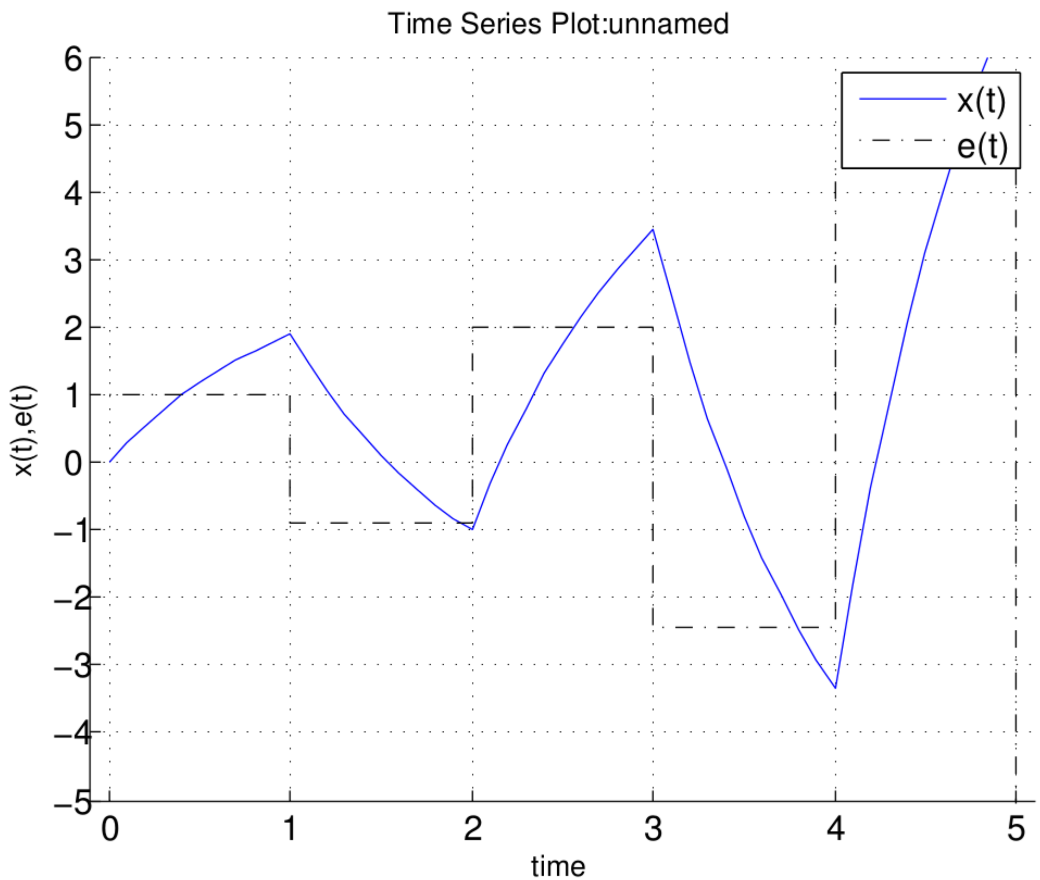
e(t), u=kp e(t)





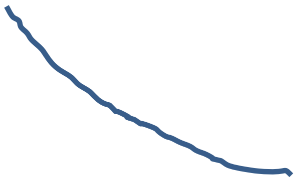
u[k] = kp*(r[k]-x[k])

Let x[k] = y[k]



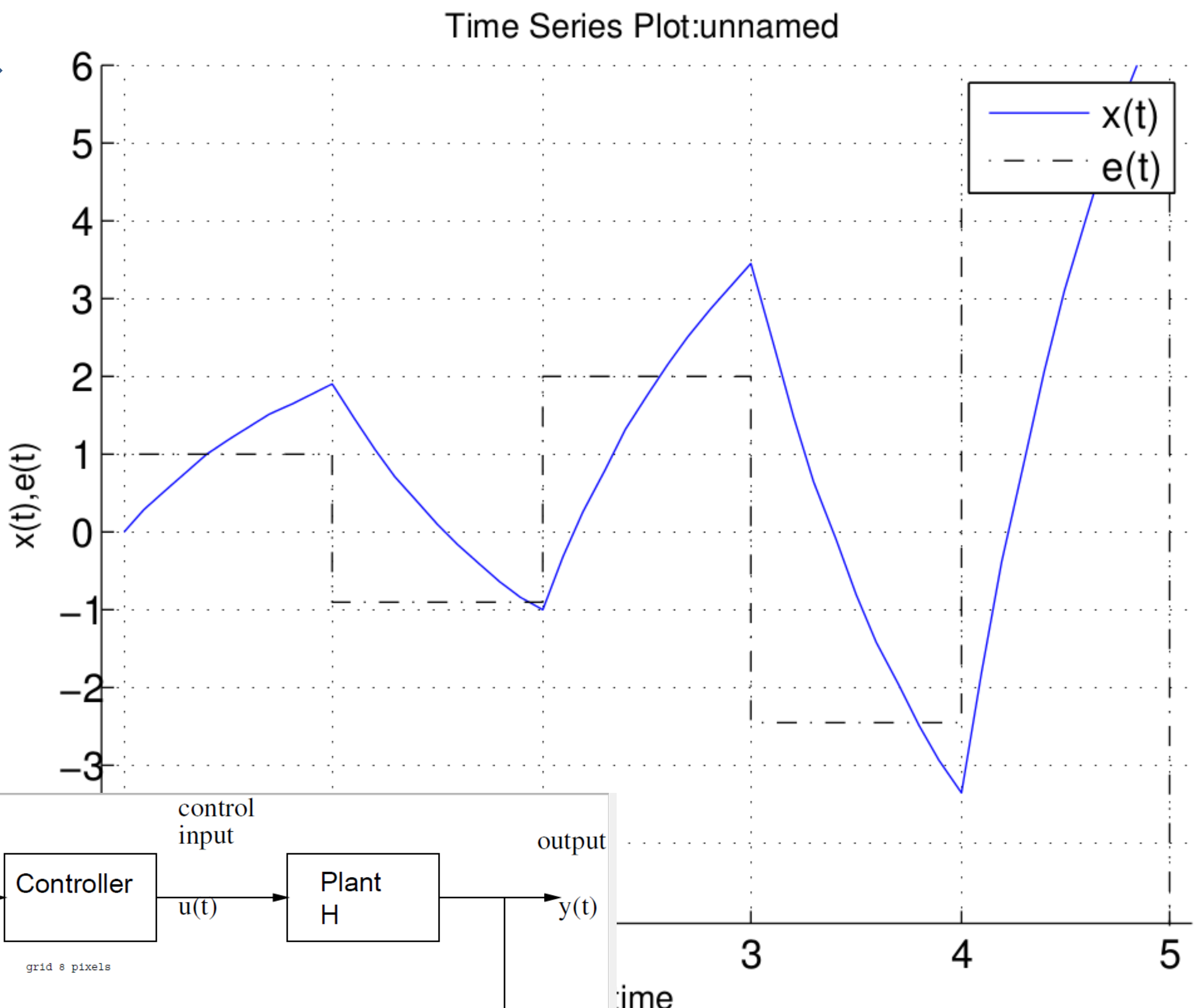
Discrete Time Control

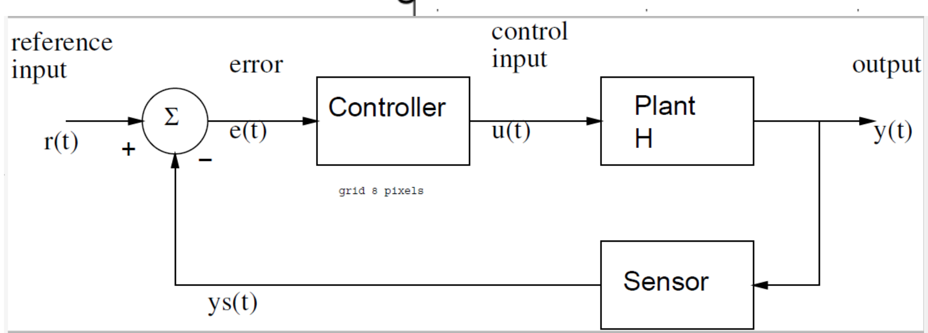
e(t), u=kp e(t)



u[k] = kp*(r[k]-x[k])

Let x[k] = y[k]





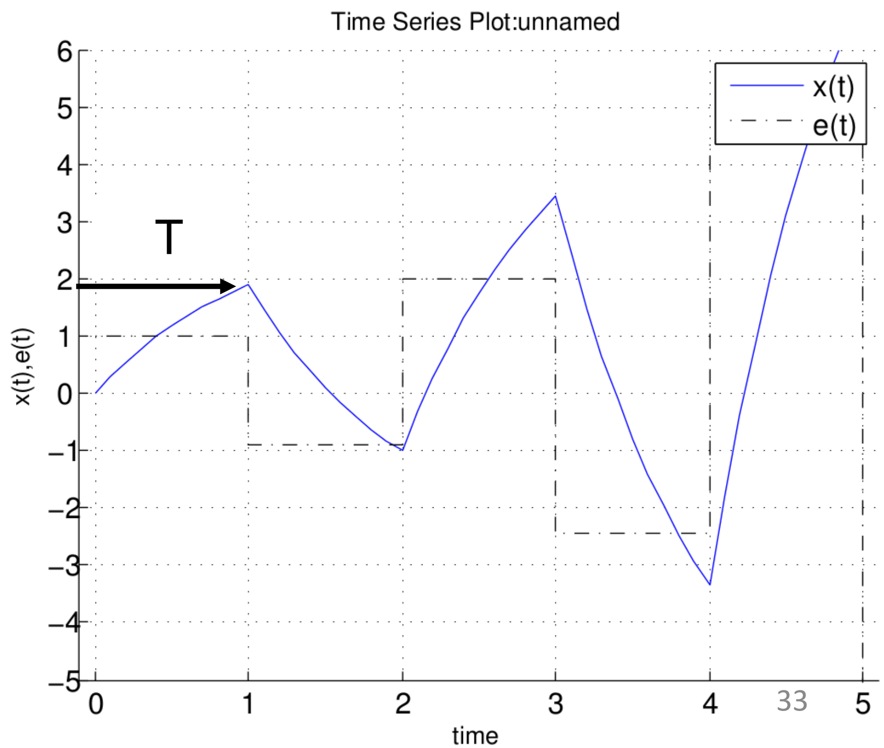
Example control- discrete time

First order CT system $\dot{x} = -x + u$

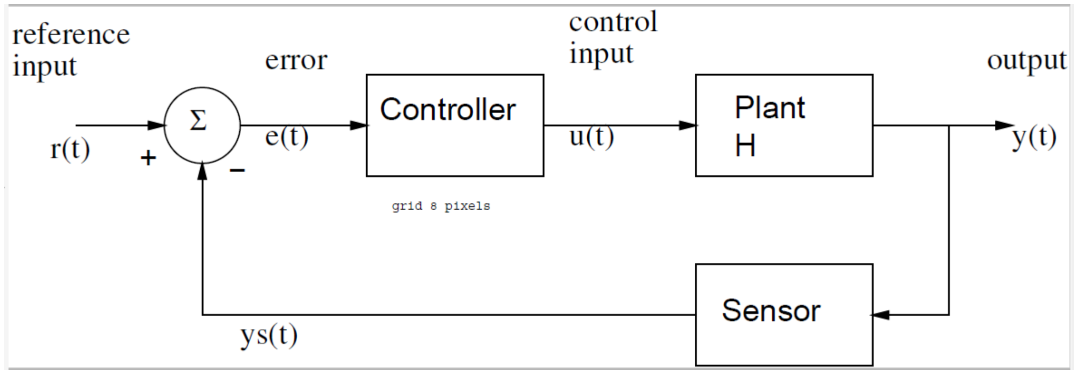
Let x = car velocity
Reference r=1 m/s unit step, k=3
e(t) = r(t) – x(t)
Let control input u[n]=3(r[n]-x[n]) = 3e[n],

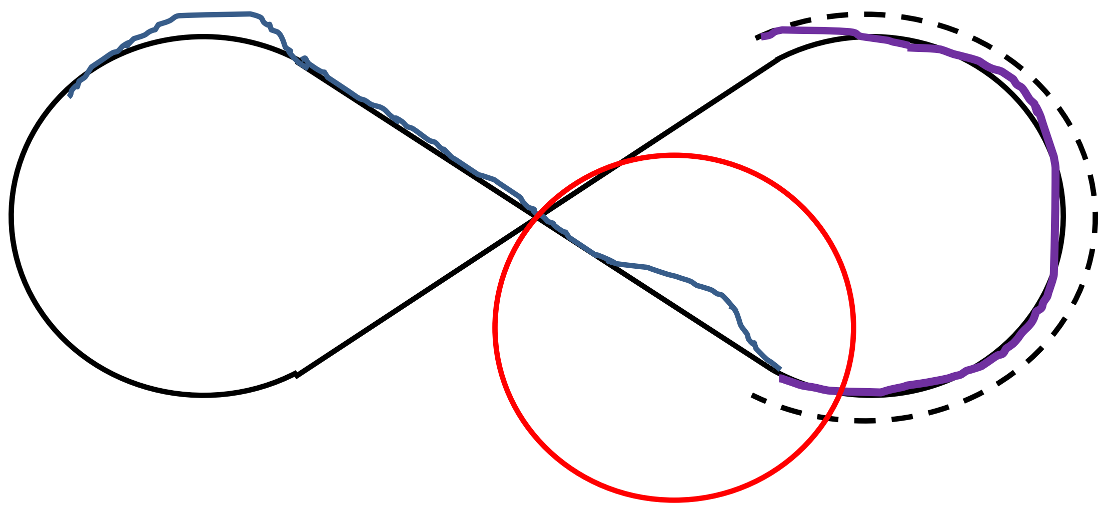
Watch out for delay!
Watch out for excess gain!

t (sec)	x(t)	e(t) = r(t) - x(t)	u(t)
0^-	0	0	0
0	0	1	3
1	2	-1	-3
2	-1	2	6
3	3.5	-2.5	-7.5
4	-3.5	4.5	13.5



Proportional + Integral

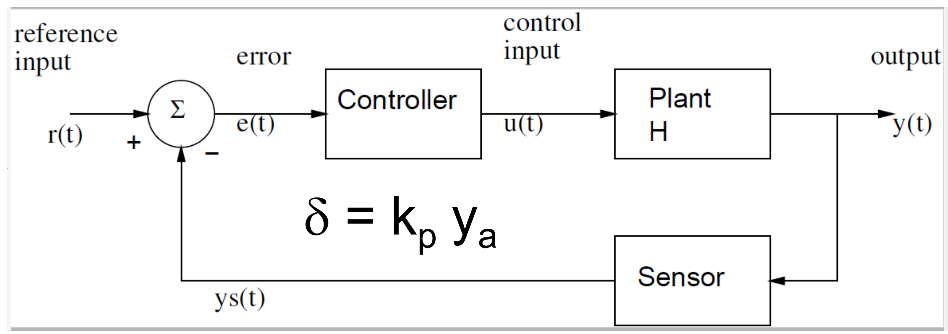


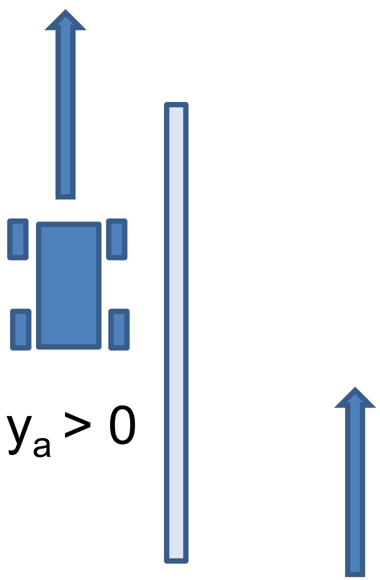


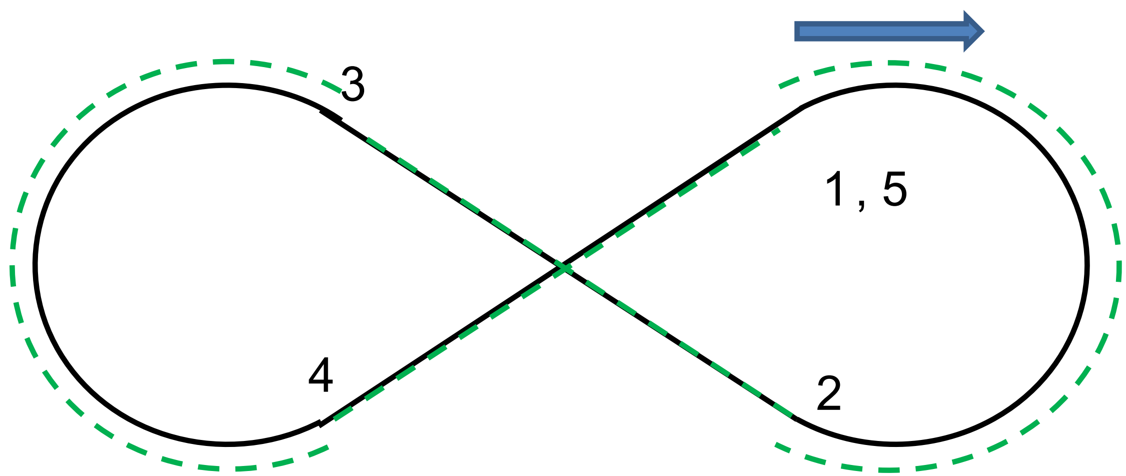
P+I control: delta =kp e + ki (integral e)

P control: delta = kp e

Feedforward using track memorization

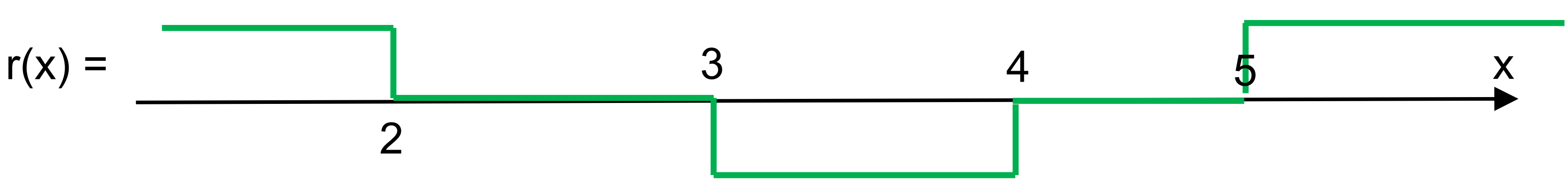






Check signs … $r(x) = -\, e(x + v\,\Delta t)$ preview of turn

or $\delta = k_p\, y_a + (1 - a)\, \delta_{old}$



EECS192 Lecture 11
Control
April 5, 2021

Topics

- Checkpoint 8: Step+ telemetry
- Progress report (due Fri 4/9 8 pm)
- Round 1 (simulation) in class 4/20 (was 4/13)
  - Submit Python 3.7 code, we run.
- Round 2 (live/hardware) in class 4/27
- ESP32 A/D Issues
- HW 2 Simulation notes
- Quiz 4 on steering 4/13
- Control Fundamentals
- Robustness (revisited)

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?

Software Robustness

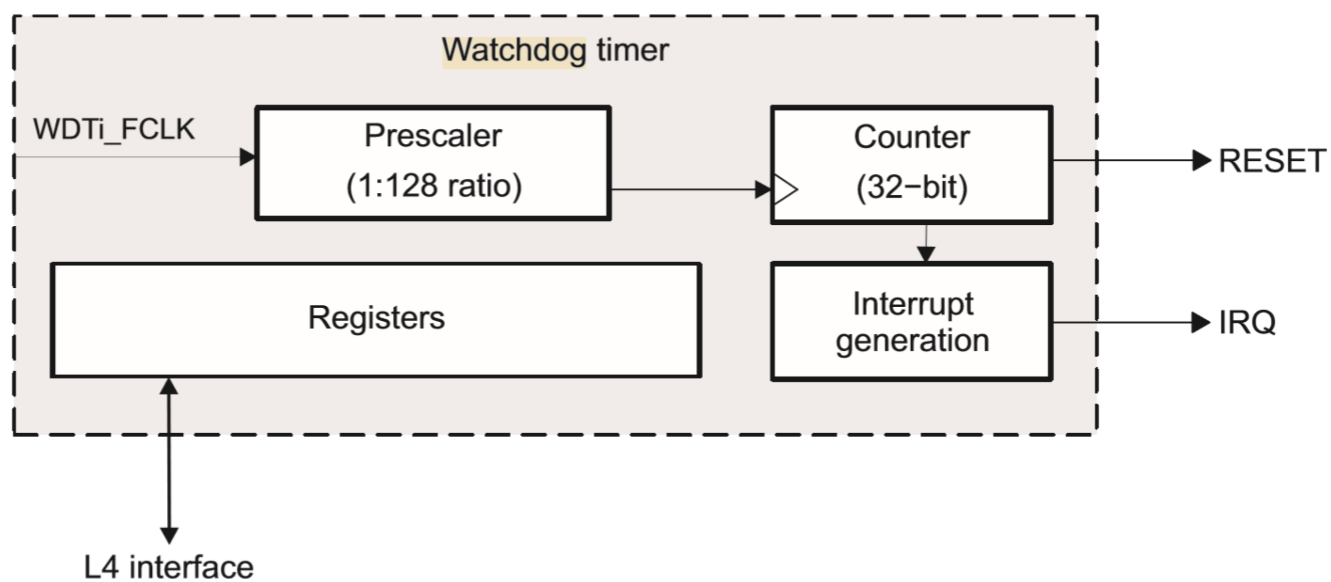
- Checksums for bit rot
- Lost track detection
- Autocalibration at startup
  - (sanity check for steering angle vs line error)
  - AGC
- State Observer/estimator
- Discrete State observer
- Watch dog timer/computer operating properly COP

C.O.P. Watchdog timer

- Despite extensive software and hardware testing, faults will still occur in real devices. Even momentary noise spikes on a power supply can lock up a processor occasionally. Such events will occur on the power grid several times a year. Watchdog timers provide a last line of defense to prevent system failure with minimal hardware cost.
- https://developer.mbed.org/cookbook/WatchDog-Timer

Watch Dog Timer (not ESP32 specific)

Figure 20-94. 32-Bit Watchdog Timer Functional Block Diagram



Watchdog reset

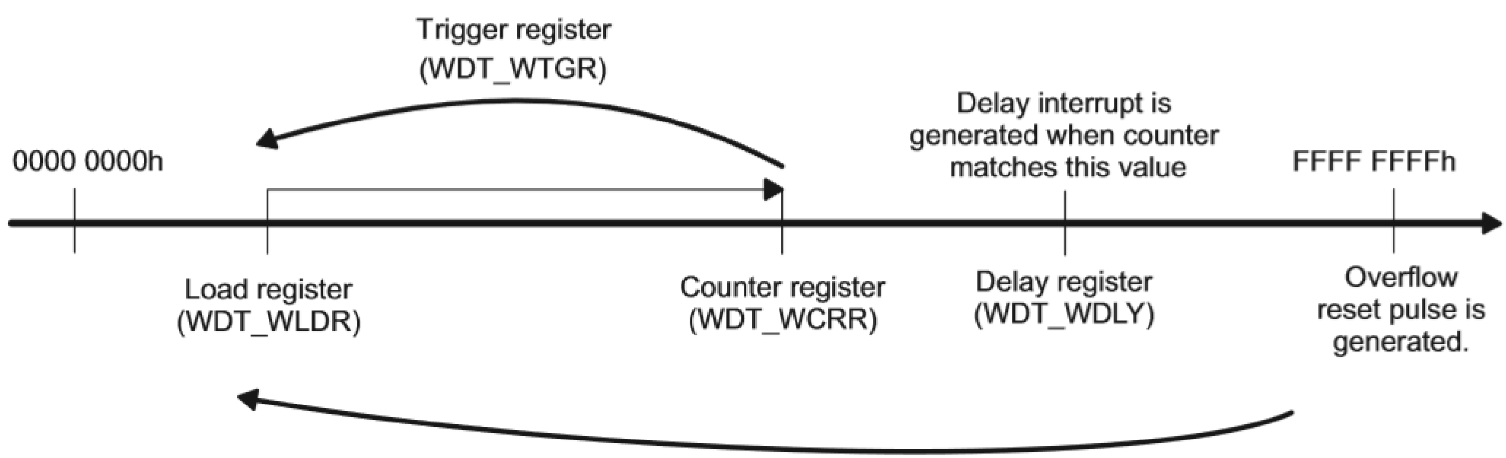
20.4.3.5 Overflow/Reset Generation

When the watchdog timer counter register (WDT_WCRR) overflows, an active-low reset pulse is generated to the PRCM module. This RESET pulse causes the PRCM module to generate global WARM reset of the device, which causes the nRESETIN_OUT pin to be driven out of the device. This pulse is one prescaled timer clock cycle wide and occurs at the same time as the timer counter overflow.

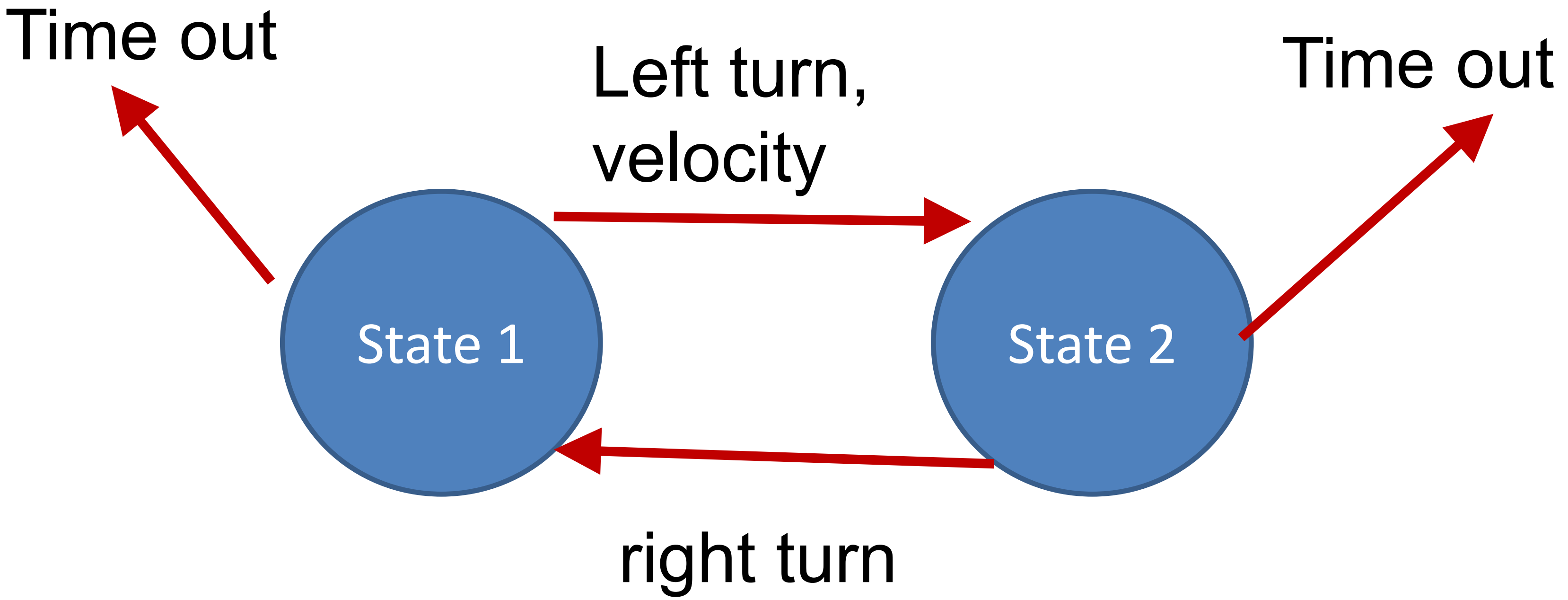
After reset generation, the counter is automatically reloaded with the value stored in the watchdog load register (WDT_WLDR) and the prescaler is reset (the prescaler ratio remains unchanged). When the reset pulse output is generated, the timer counter begins incrementing again.

Figure 20-95 shows a general functional view of the watchdog timers.

Figure 20-95. Watchdog Timers General Functional View



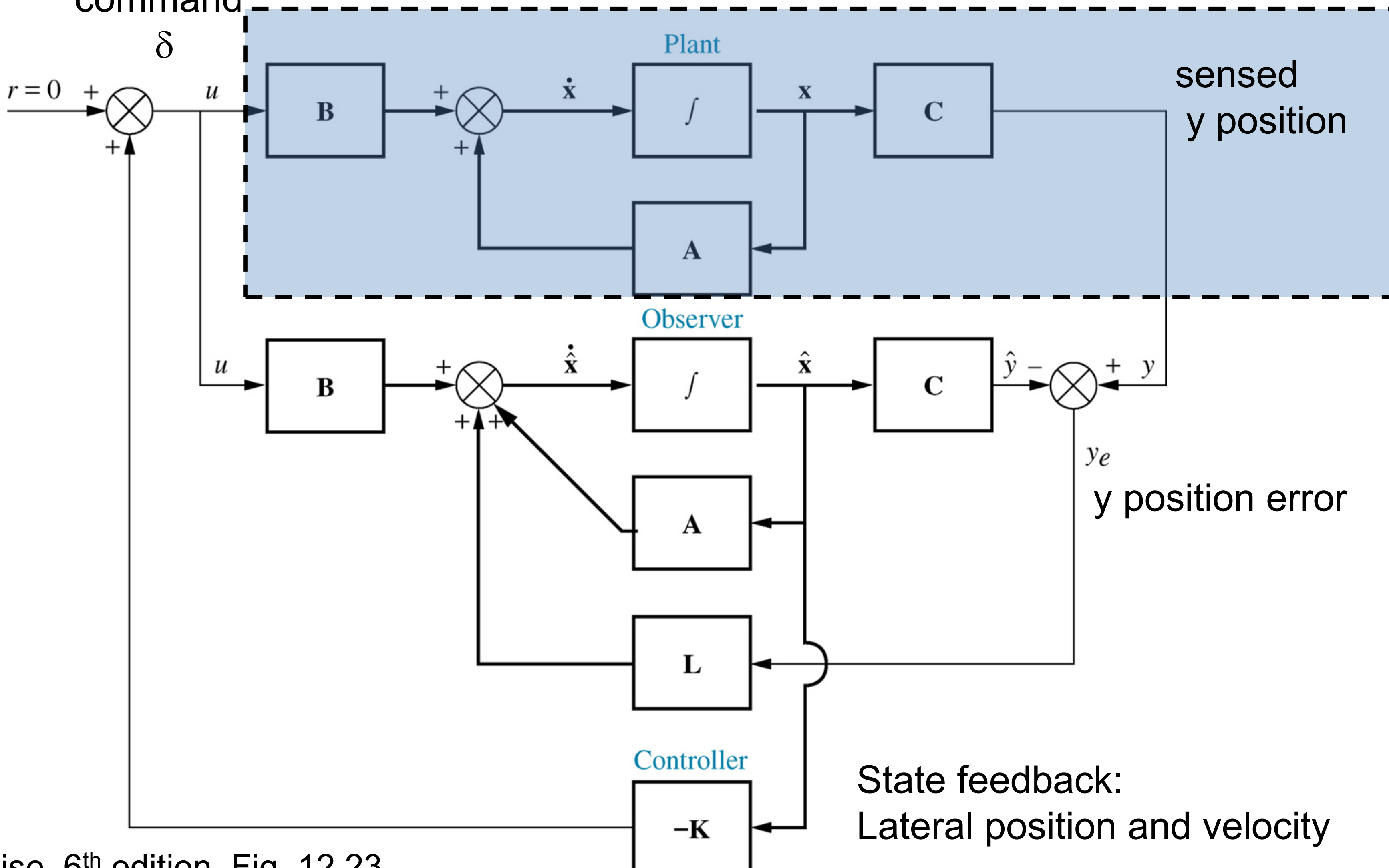
FSM Recognizer (generalized WDT)



Software Robustness: Observer

(on board)

Steering command δ



sensed y position

y position error

State feedback:
Lateral position and velocity

N. Nise, 6th edition, Fig. 12.23

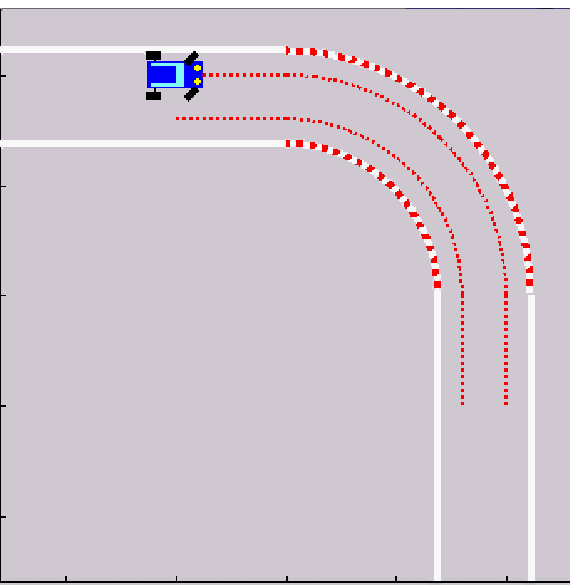
Extra slides

Steering References (on web page)

- Vehicle Dynamics and Control During Abnormal Driving

http://soliton.ae.gatech.edu/people/dcsl/research-abnormal.html

Prof. Panagiotis Tsiotras, Georgia Tech



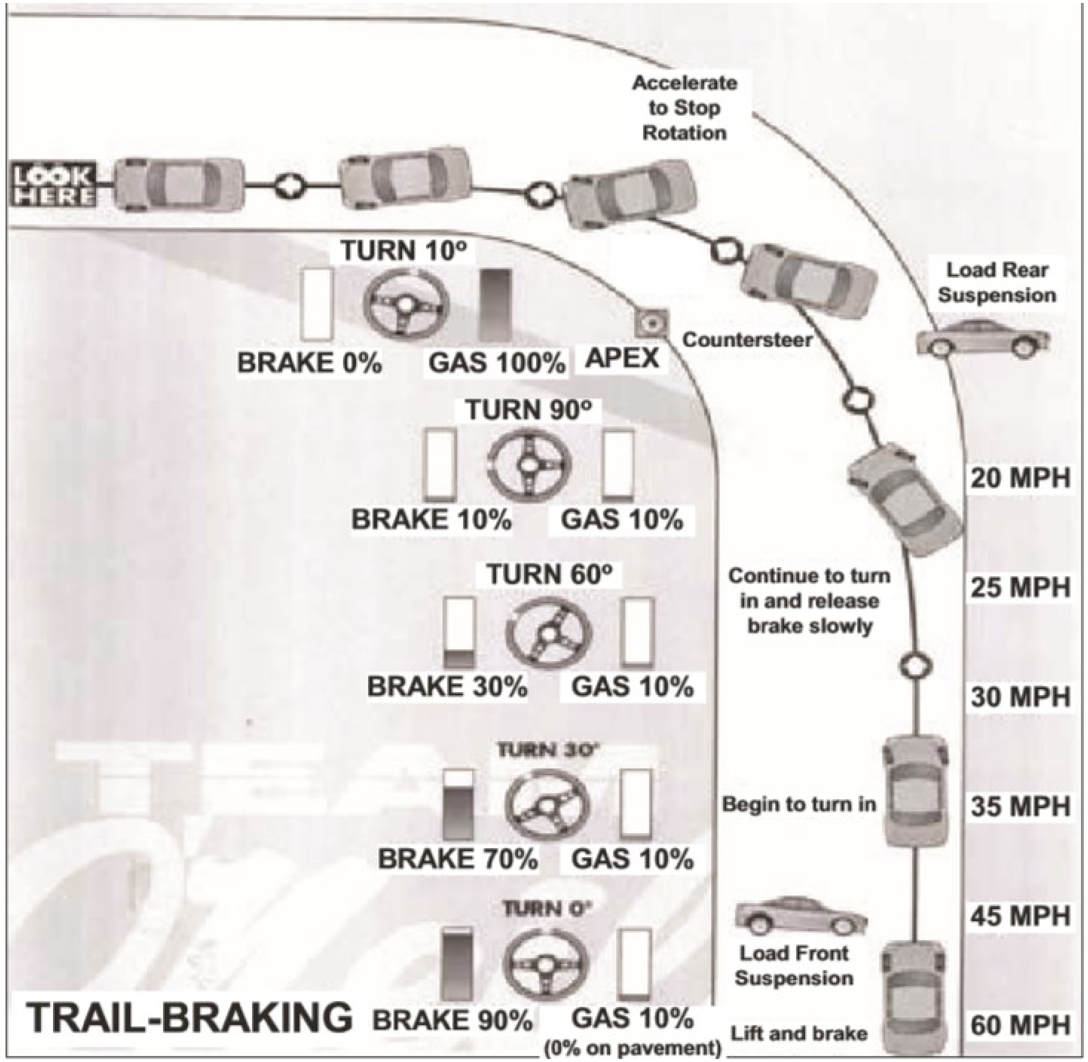
http://soliton.ae.gatech.edu/people/dcsl/movies/skidding.avi

http://soliton.ae.gatech.edu/people/dcsl/movies/TrailBraking.avi

Steering References (on web page)

- Vehicle Dynamics and Control During Abnormal Driving (Georgia Tech)
- Velenis, E., Tsiotras, P., and Lu, J., "Aggressive Maneuvers on Loose Surfaces: Data Analysis and Input Parameterization,'' 15th IEEE Mediterranean Control Conference, June 26-29, Athens, Greece.
- Velenis, E., Tsiotras, P., and Lu, J., "Modeling Aggressive Maneuvers on Loose Surfaces: The Cases of Trail-Braking and Pendulum-Turn,'' European Control Conference, Kos, Greece, July 2-5, 2007.
- Some nice turning simulation (Georgia Tech): (video 1) (video 2)
- Baffet, G. Charara, A. Dherbomez, G. ''An Observer of Tire Road Forces and Friction for Active Security Vehicle Systems'' Mechatronics, IEEE/ASME Transactions on Publication Date: Dec. 2007 Volume: 12, Issue: 6 On page(s): 651-661
- Tseng, H.E. Ashrafi, B. Madau, D. Allen Brown, T. Recker, D. ''The development of vehicle stability control at Ford'' Mechatronics, IEEE/ASME Transactions on Publication Date: Sep 1999 Volume: 4, Issue: 3 On page(s): 223-234
- T. Pilutti, G. Ulsoy, and D. Hrovat, ''Vehicle steering intervention through differential braking,'' Proc. American Control Conf. Seattle, Wash. June 1995.
- Brennan, S. Alleyne, A. ''Using a scale testbed: Controller design and evaluation'' Control Systems Magazine, IEEE Publication Date: Jun 2001 Volume: 21, Issue: 3 On page(s): 15-26
- Brennan, S. Alleyne, A. ''The Illinois Roadway Simulator: a mechatronic testbed for vehicle dynamics and control,'' Mechatronics, IEEE/ASME Transactions on Publication Date: Dec 2000 Volume: 5, Issue: 4 On page(s): 349-359
- Chankyu Lee K. Hedrick Kyongsu Yi , ''Real-time slip-based estimation of maximum tire-road friction coefficient,'' Mechatronics, IEEE/ASME Transactions on Publication Date: June 2004
- Han-Shue Tan; Guldner, J.; Patwardhan, S.; Chieh Chen; and others. Development of an automated steering vehicle based on roadway magnets-a case study of mechatronic system design. IEEE/ASME Transactions on Mechatronics, Sept. 1999, vol.4, (no.3):258-72.
- Guldner, J.; Sienel, W.; Han-Shue Tan; Ackermann, J.; and others. Robust automatic steering control for look-down reference systems with front and rear sensors. IEEE Transactions on Control Systems Technology, Jan. 1999, vol.7, (no.1):2-11.
- Patwardhan, S.; Han-Shue Tan; Guldner, J. A general framework for automatic steering control: system analysis. Proceedings of 16th American CONTROL Conference, Albuquerque, NM, USA, 4-6 June 1997). Evanston, IL, USA: American Autom. Control Council, 1997. p. 1598-602 vol.3.
- Patwardhan, S.; Han-Shue Tan; Guldner, J.; Tomizuka, M. Lane following during backward driving for front wheel steered vehicles. Proceedings of 16th American CONTROL Conference, Albuquerque, NM, USA, 4-6 June 1997). Evanston, IL, USA: American Autom. Control Council, 1997. p. 3348-53 vol.5.
- Guldner, J.; Han-Shue Tan; Patwardhan, S. Study of design directions for lateral vehicle control. Proceedings of the 36th IEEE Conference on Decision and Control, San Diego, CA, USA, 10-12 Dec. 1997). New York, NY, USA: IEEE, 1997. p. 4732-7 vol.5.
- Analysis of automatic steering control for highway vehicles with look-down lateral reference systems. Vehicle System Dynamics, Oct. 1996, vol.26, (no.4):243-69.

Steering: Trail Braking Maneuver (Rally car)



1. Brake hard, drive straight (increased load on front wheels)
2. Increase steering command, reduce braking (oversteering)
3. Decrease steering, counter steers, apply throttle to stabilize

Velenis, E., Tsiotras, P., and Lu, J., "Modelling Aggressive Maneuvers on Loose Surfaces'' European *Control Conference*, 2007.

Steering: Trail Braking Maneuver

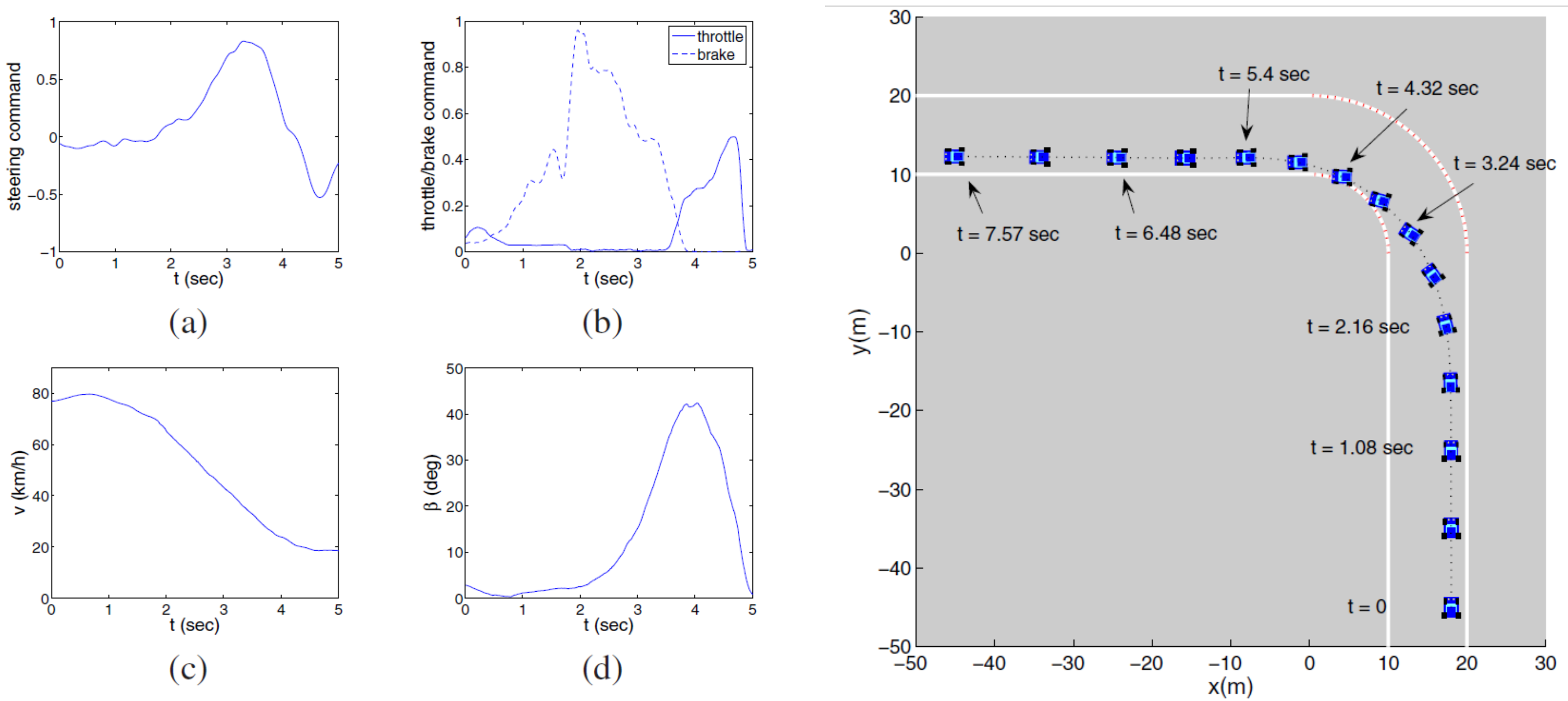


Fig. 3. Trail-Braking maneuver experimental data: (a) Normalized steering command; (b) Normalized throttle and braking commands; (c) Vehicle speed; (d) Vehicle slip angle.

1. Brake hard, drive straight (increased load on front wheels)
2. Increase steering command, reduce braking (oversteering)
3. Decrease steering, counter steer, apply throttle to stabilize

Velenis, E., Tsiotras, P., and Lu, J., "Aggressive Maneuvers on Loose Surfaces: Data Analysis and Input Parameterization," *15th IEEE Mediterranean Control Conference*, June 26-29, 2007 Athens, Greece.

Steering: Trail Braking Maneuver

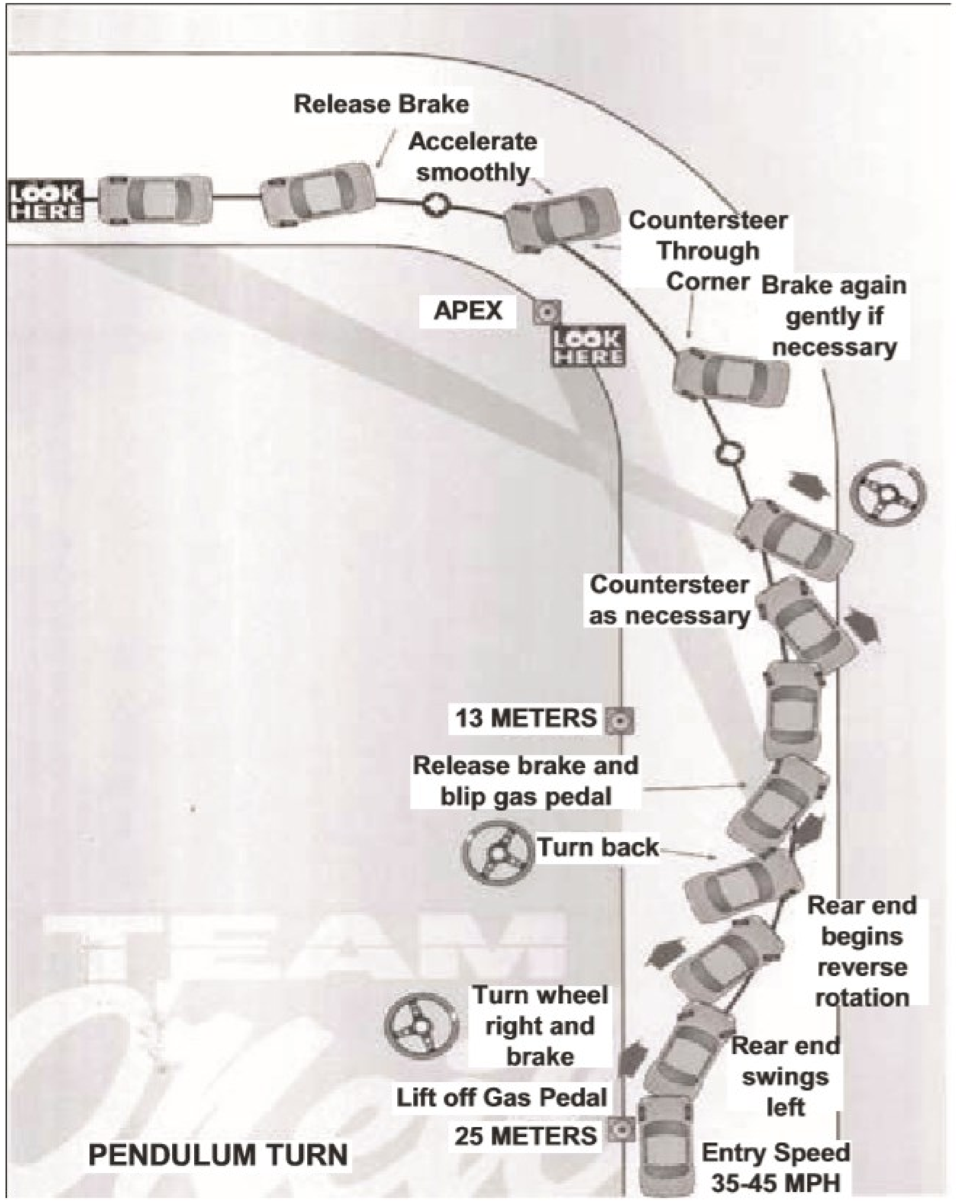
- Vehicle Dynamics and Control During Abnormal Driving

http://soliton.ae.gatech.edu/people/dcsl/research-abnormal.html

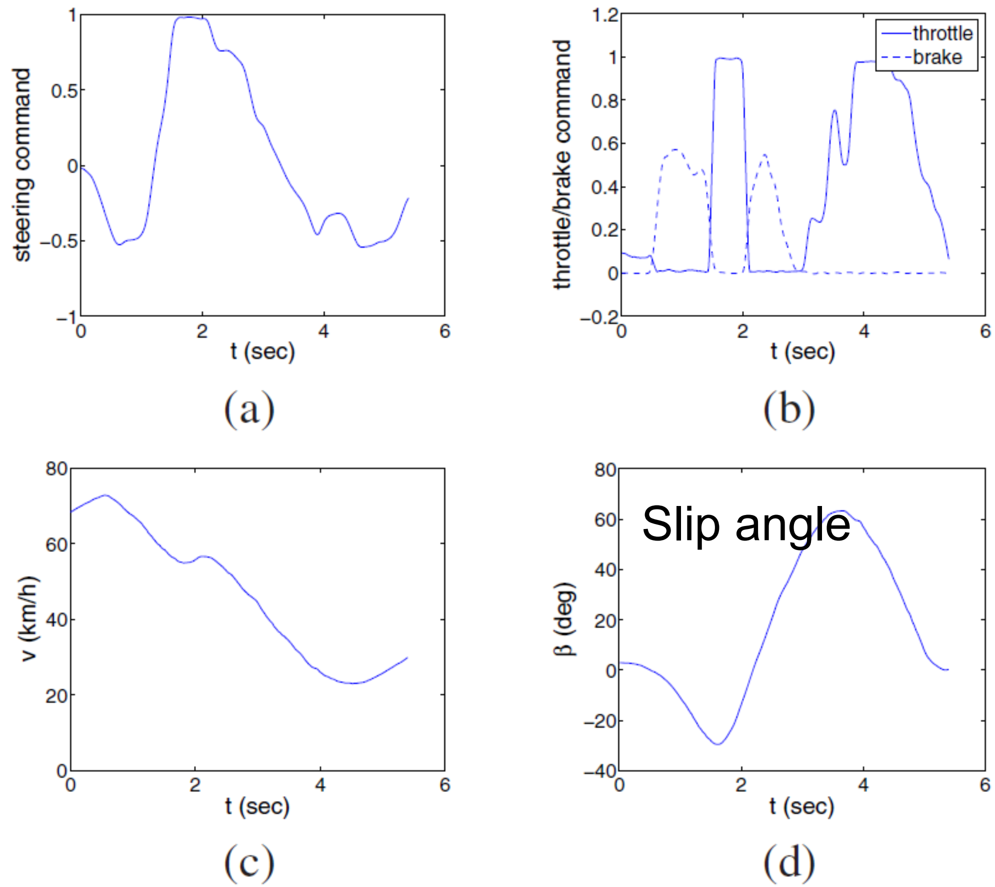
Prof. Panagiotis Tsiotras, Georgia Tech

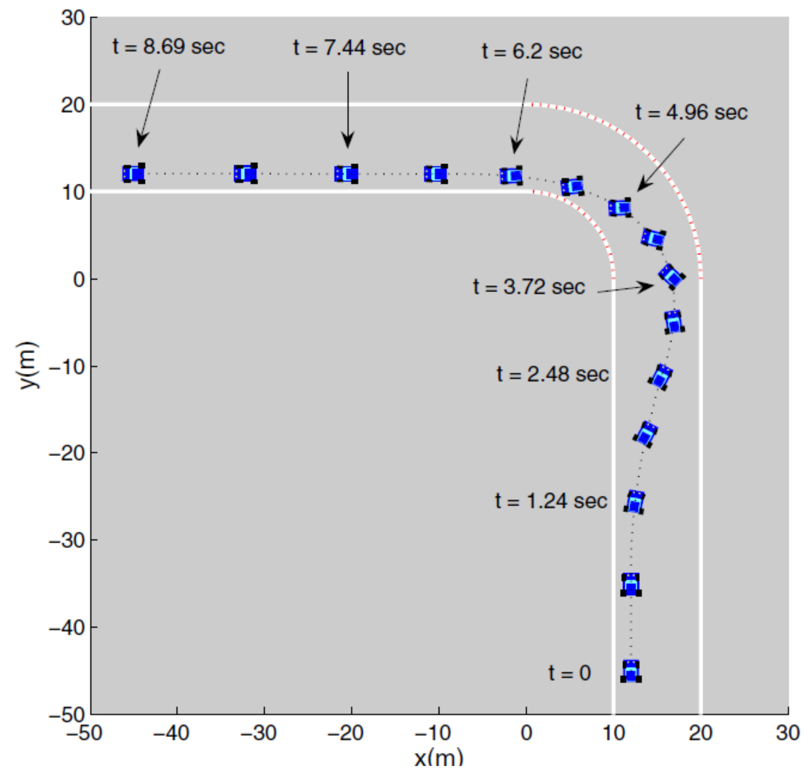
http://soliton.ae.gatech.edu/people/dcsl/movies/TrailBraking.avi

Steering: Pendulum Turn Maneuver (Sim)



Steering: Pendulum Turn Maneuver (Sim)





1. Turn opposite while applying brakes (increased load on front wheels, oversteering)
2. Throttle blip to damp rotation
3. steer in direction of turn and apply brakes to rotate fast
4. Decrease steering command, counter-steers, applies throttle to stabilize

Velenis, E., Tsiotras, P., and Lu, J., "Aggressive Maneuvers on Loose Surfaces: Data Analysis and Input Parameterization," *15th IEEE Mediterranean Control Conference*, June 26-29, 2007 Athens, Greece.

http://soliton.ae.gatech.edu/people/dcsl/movies/PendulumTurn.avi

Vehicle Stability through Differential Braking

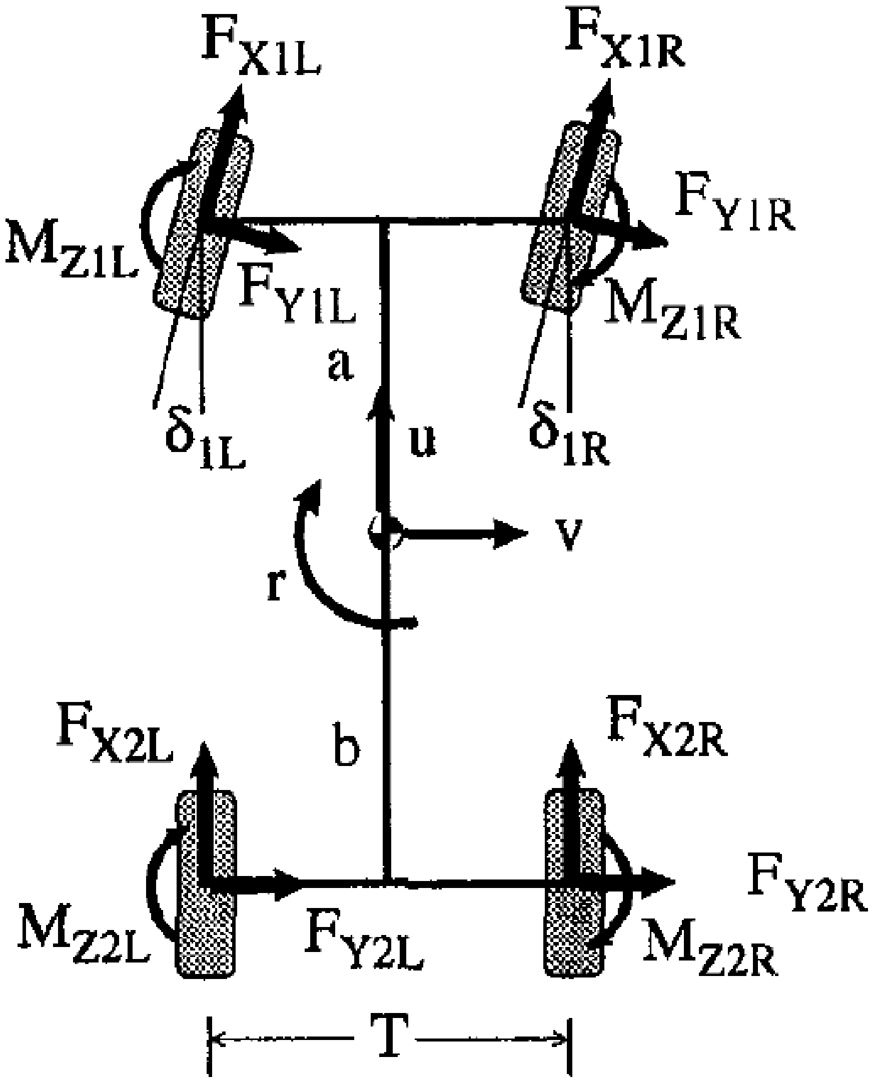


Fig. 1 Seven DoF vehicle model

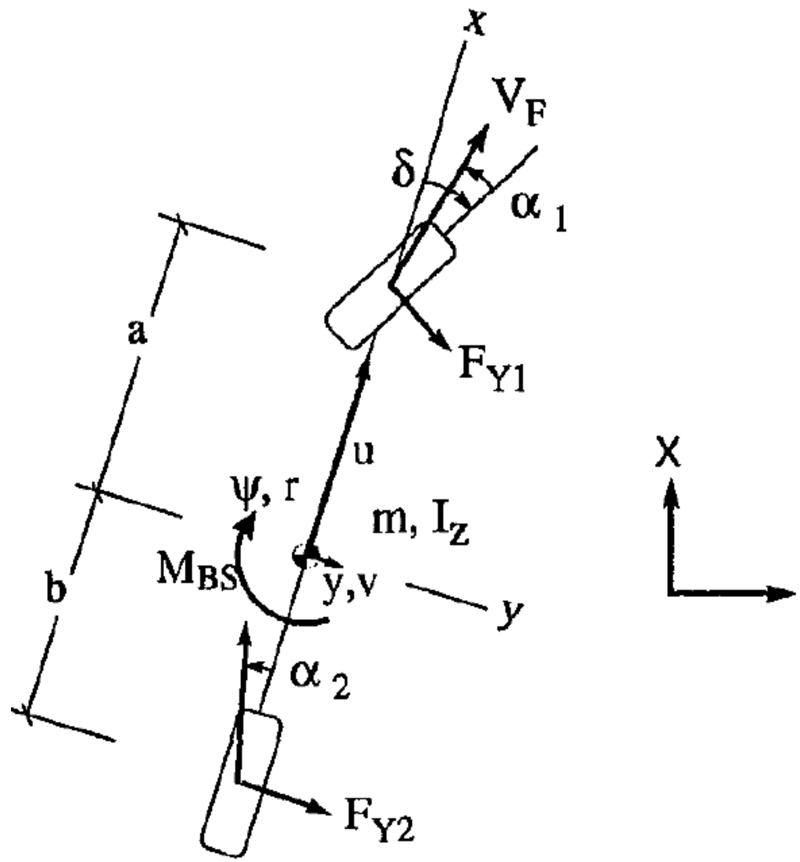
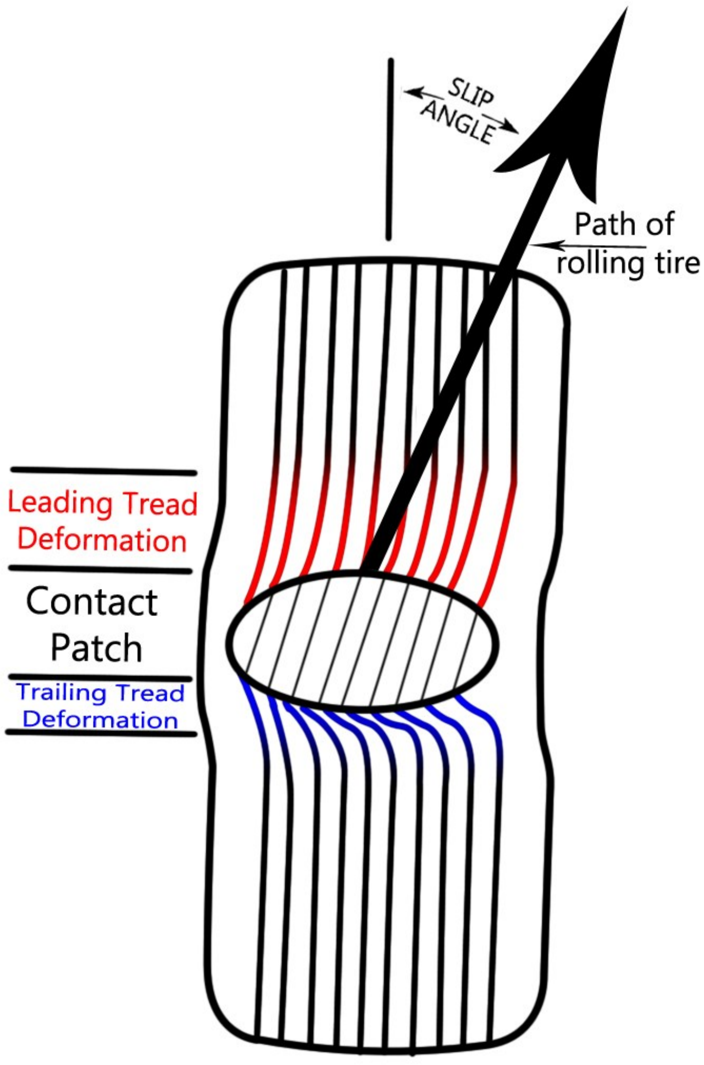


Fig. 2 Two DoF model

- T. Pilutti, G. Ulsoy, and D. Hrovat, "Vehicle steering intervention through differential braking," Proc. American Control Conf. Seattle, Wash. June 1995.

Tire Slip Angle



http://technicalf1explained.blogspot.com/2012/10/f1-tirespart-2.html