EECS192 Lecture 6
Motor Modelling and Steering Introduction
Feb. 23, 2021

Topics

- Checkpoint 5:
- Available space survey- circle/figure 8
- Checkpoint 6:
- Motor model (for velocity control)
- Steering Introduction
- Simulator (preview)

CP5- velocity sensing

The car should be upside down, or lifted off the ground so it does not move

C5.1 Demonstrate that you have velocity sensing working and the output in terms of some physical units (m/s, mm/s, etc). Turning the wheels by hand should show a low velocity.

C5.2 Set a low constant motor PWM. Show that the estimated velocity is relatively constant.

C5.3 Show that with the constant PWM from C5.2 that the velocity sensor estimated velocity drops if the wheels are loaded or stopped.

C5.4 Show velocity control. The recommended target setpoint is 3 m/s, which should provide enough encoder counts for a somewhat stable control loop. It's fine if the applied PWM is noticeably jittering or if the actual velocity is inaccurate. However, if you load the wheels (with, say, a book), the controller should compensate by applying a higher drive strength. (Print PWM and sensed velocity as load is applied to wheel.)

C5.5 Show velocity control working with the basic line sensing from C4.4. (Printing PWM, sensed velocity, and line center is sufficient, as load is applied to wheel and car is positioned by hand)

C5.6 All members must fill out the checkpoint survey before the checkoff close. Completion is individually graded.

CP6- Closed Loop Track with Velocity Control

Set up a figure 8* track. Use 1 meter string with chalk attached to make circles, and connect with tangent lines, and 60 degree crossing. Use white masking tape for figure 8 if on light background, or black tape if on light background.

C6.1 Show car driving the figure 8*, at speed of 1 m/sec or better.

(You may use a wireless command to tell the car to start or stop running, but no other commands may be sent to the car.)

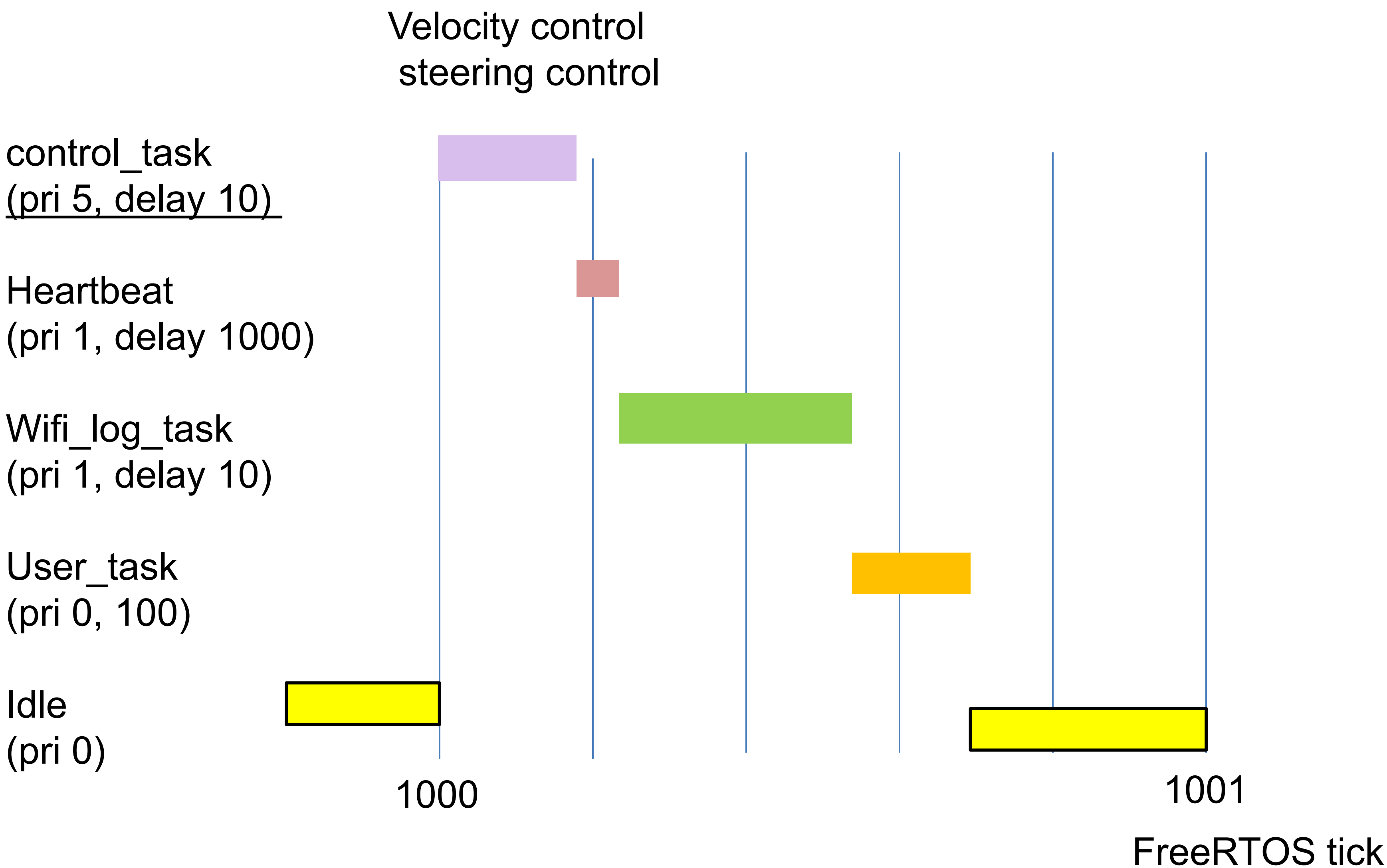
C6.2 Submit plots on one graph: steering angle command (degrees or radians), track error (cm), ESC command (% full speed), sensed velocity, all versus time axis in seconds.

C6.3 All members must fill out the checkpoint survey before the checkoff close. Completion is individually graded.

**** If you do not have space for a full size figure 8, use smaller than 1 m radius to fit. If you do not have room for a figure 8, use a circle of up to 1 m radius.***

(example Amazon tape): https://www.amazon.com/Removable-Painters-Painting-Labeling-Stationery/dp/B082R27TP6/ref=sr_1_7?dchild=1&keywords=1+inch+white+masking+tape&qid=1613947385&s=industrial&sr=1-7

Skeleton Tasks



vTaskDelay(delay / portTICK_PERIOD_MS);

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?

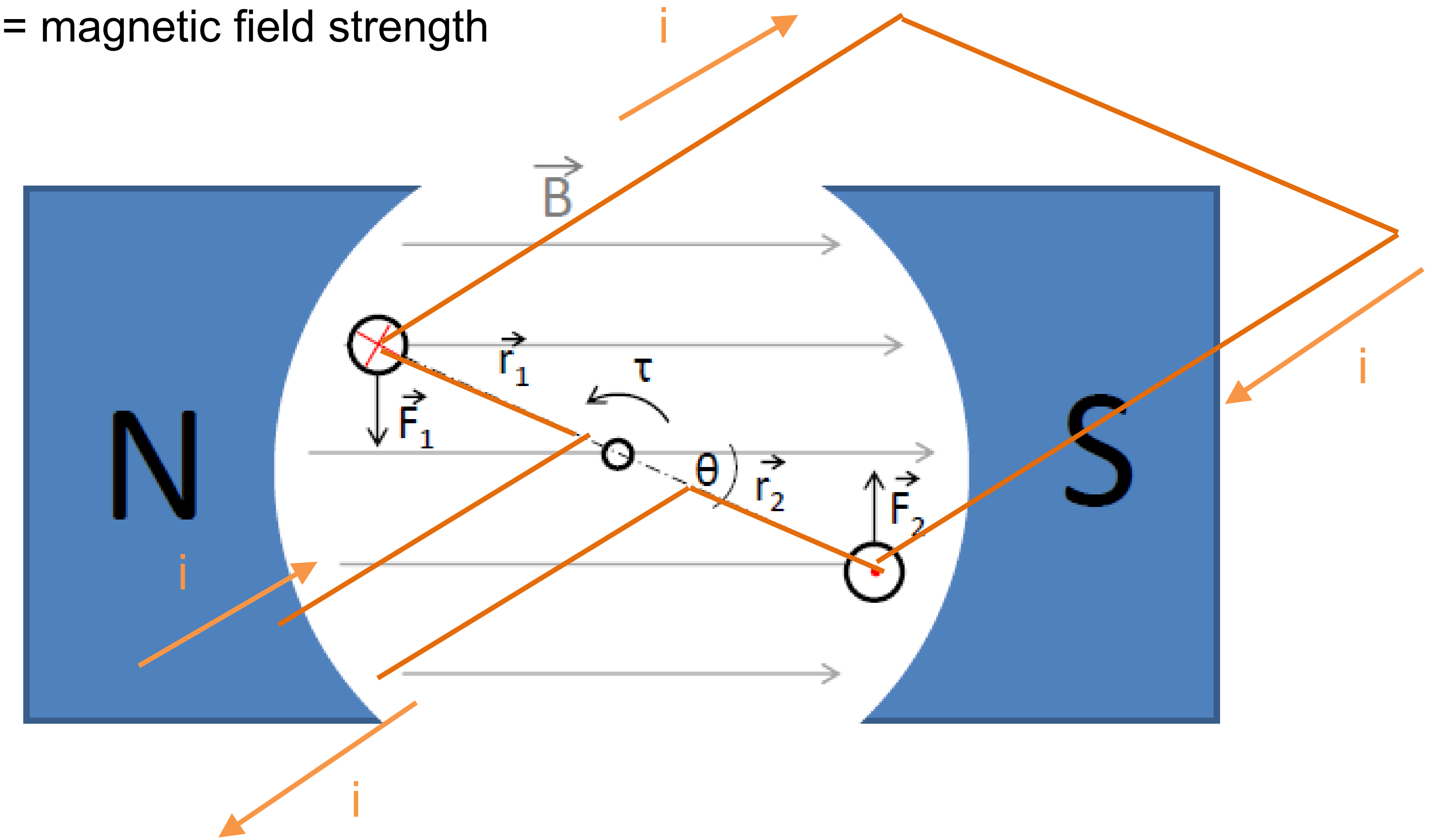
EECS192 Lecture 6
Motor Modelling and Steering Introduction
Feb. 23, 2021

Topics

- Checkpoint 5:
- Available space survey- circle/figure 8
- Checkpoint 6:
- Motor model (for velocity control)
- Steering Introduction
- Simulator (preview)

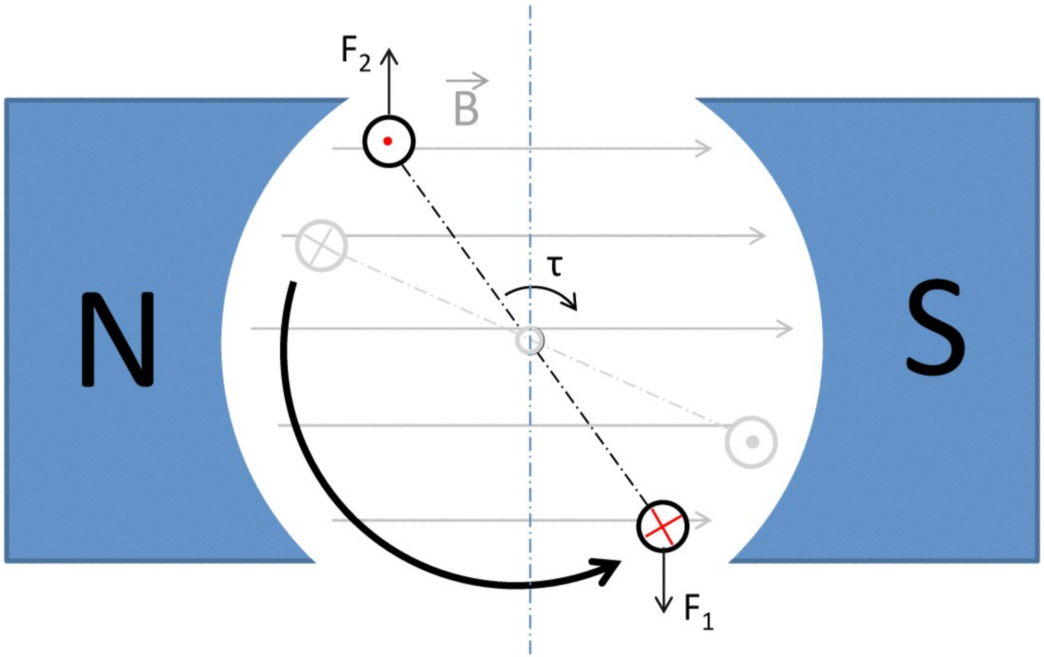
DC Motor Physical Model

i=current
B = magnetic field strength

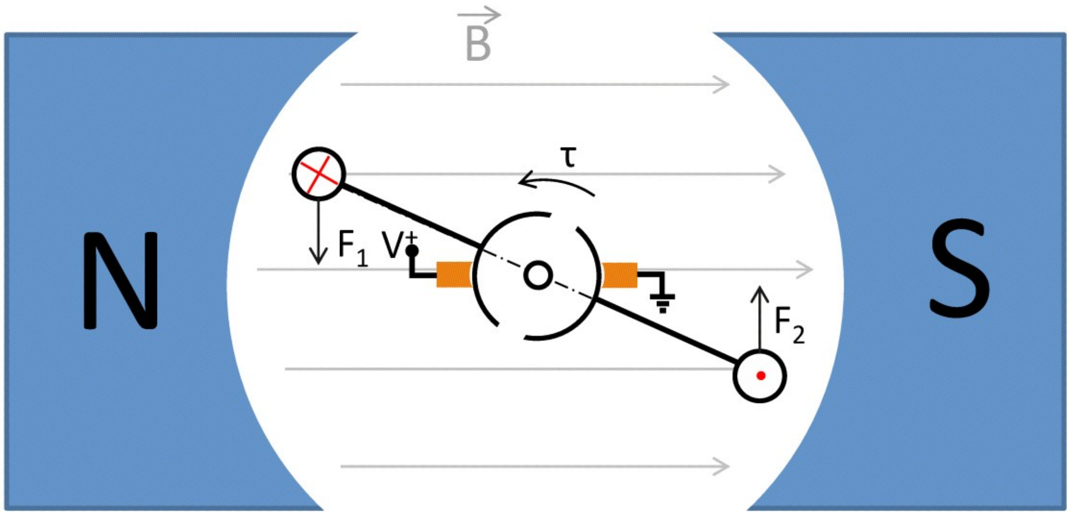


$$\vec{F} = i\vec{l} \times \vec{B}$$

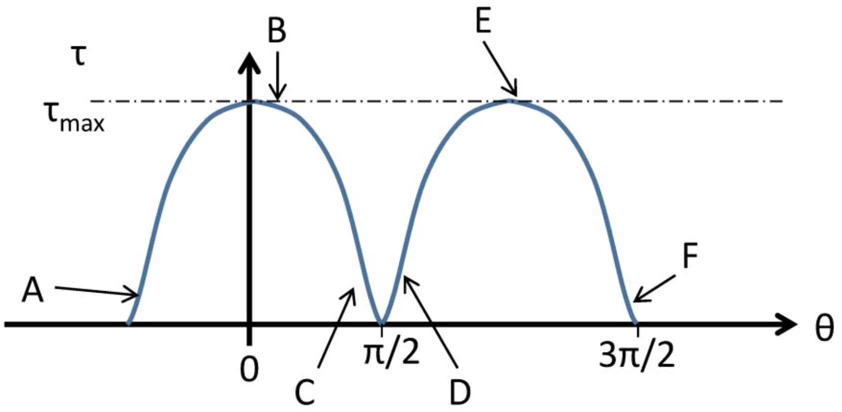
$$\tau = \vec{r}_1 \times \vec{F}_1 + \vec{r}_2 \times \vec{F}_2$$



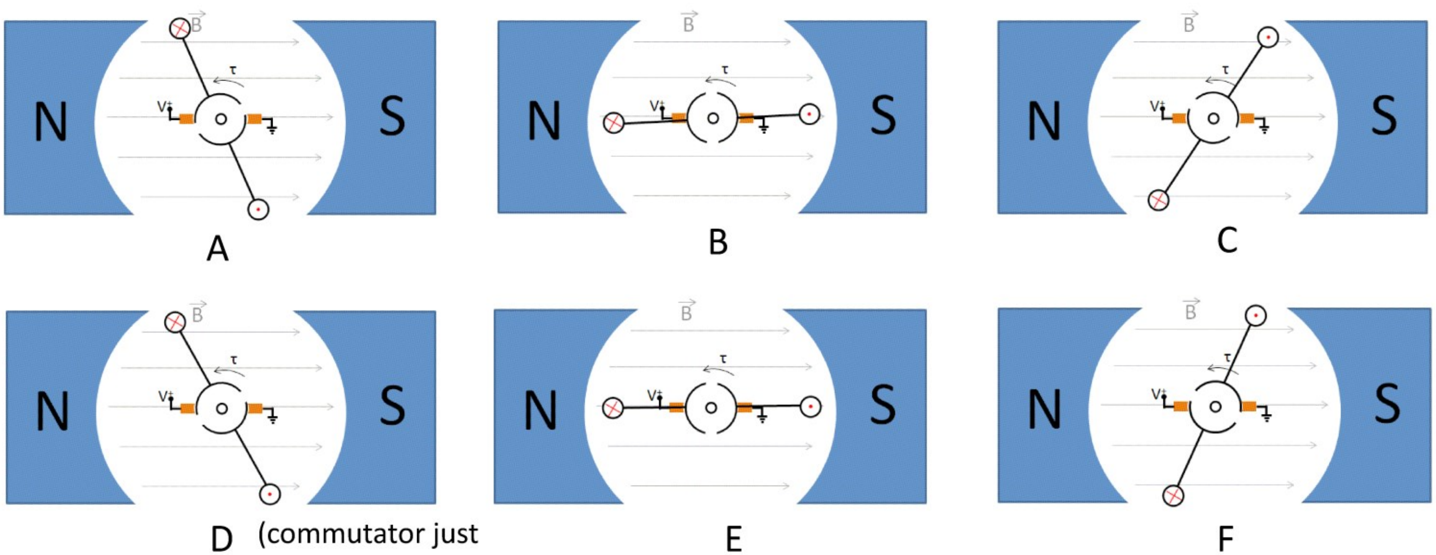
Rotating past the θ = π/2 line (blue) is a problem if current in the loop is in the same direction, because it would cause the reversal of the torques.



So, we add a commutator to *reverse* the current through the loop when the coils turn past $\theta = \pi/2$

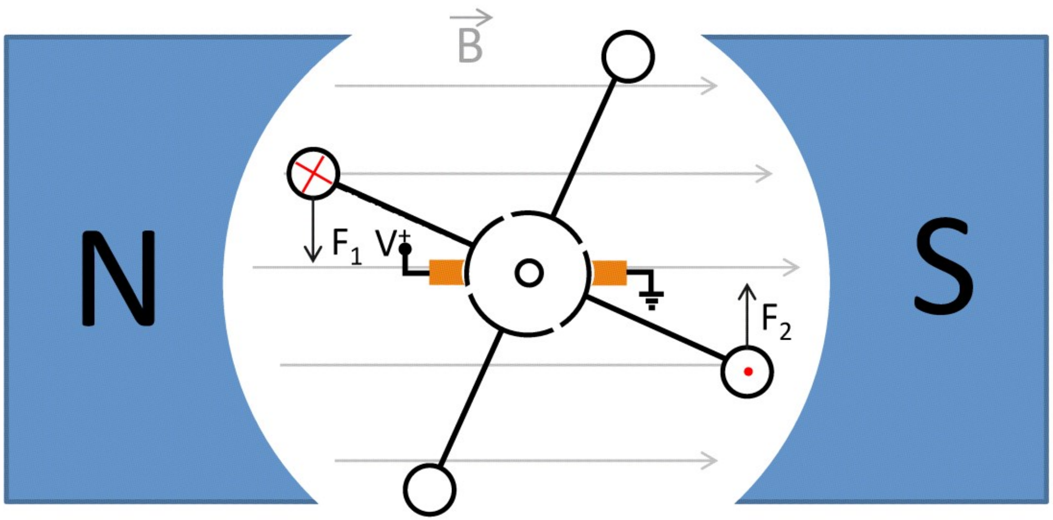


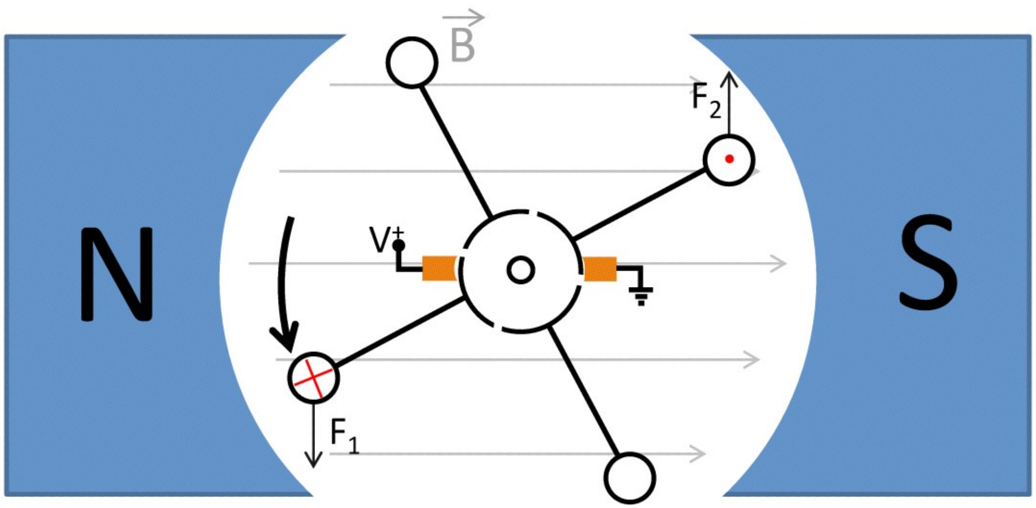
It works, but big torque ripple with only two segment commutator.

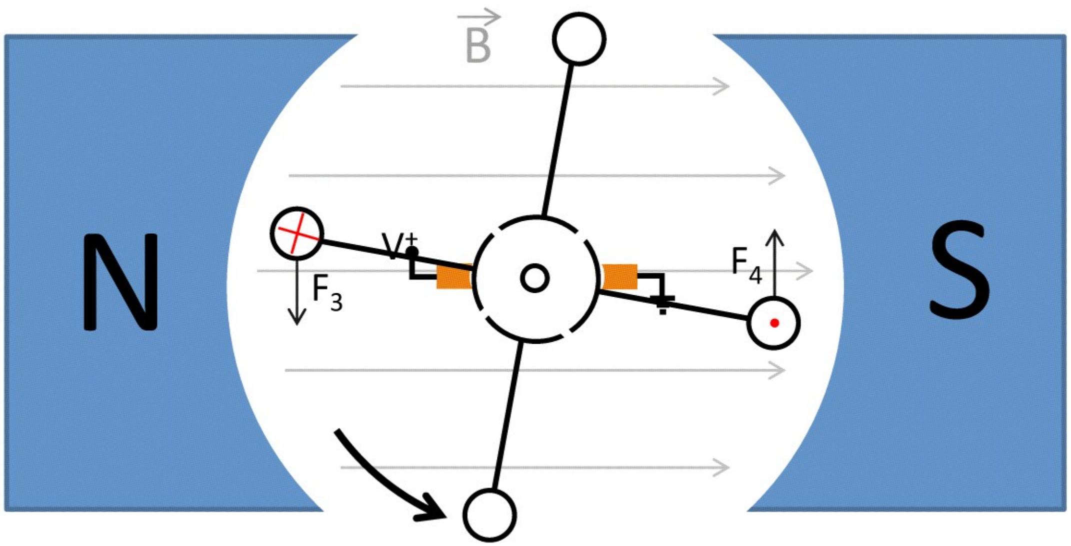


(commutator just reversed the current directions)

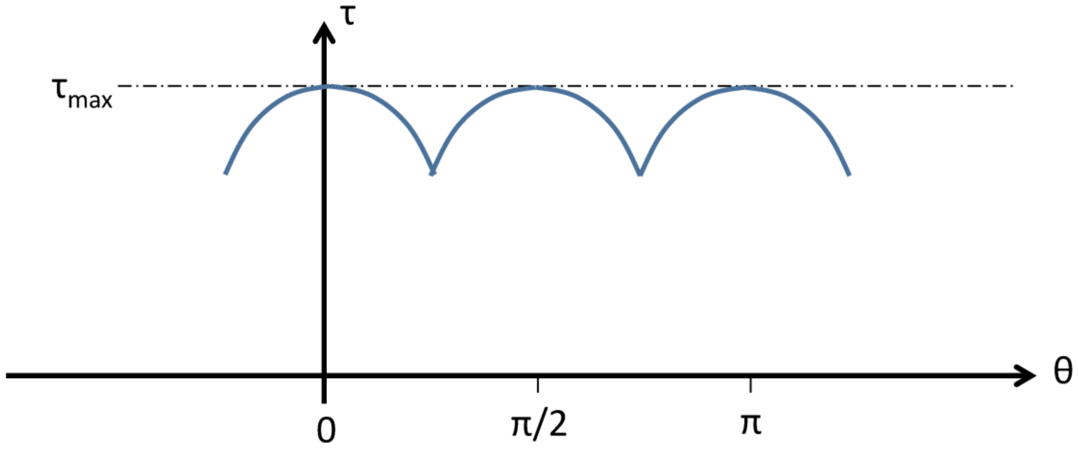
Four segment commutator → reduced torque ripple.
(Current passing only through one winding at a time)




B
F2
V+
N
S
F1



B
N
S
V+
F3
F4



Now, less torque ripple.

Same principle applies to a 6-segment commutator design, like we discussed in class

Motor Model

http://inst.eecs.berkeley.edu/~ee192/sp18/files/NiseAppendixI.pdf

http://inst.eecs.berkeley.edu/~ee192/sp13/pdf/motor_modeling.pdf

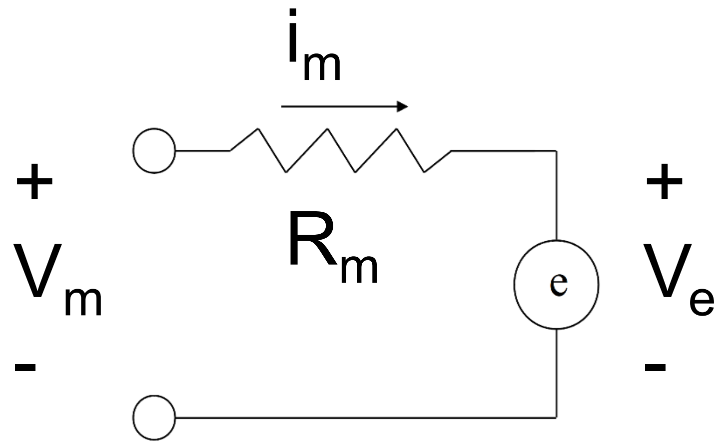
Torque equation: $\tau = k_\tau \, i_m$

Faraday's Law: $-d\Phi/dt$, where Φ is magnetic flux in loop

Back EMF equation: $V_e = k_e \, \dot{\theta}_m$

Motor Electrical Model

Motor Electrical Model
Back EMF
Motor electromechanical behavior



Also- see motor worksheet……

$$i_m = \frac{V_{BAT} - k_e \dot{\theta}_m}{R_m}$$

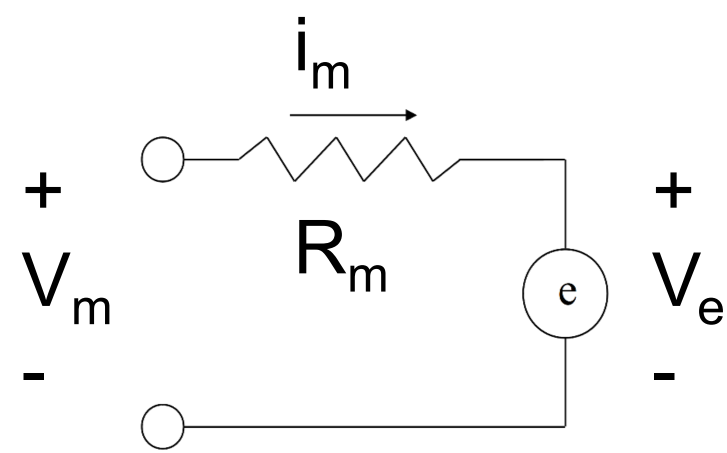
Torque equation: $\tau = k_\tau \, i_m$

Back EMF equation: $V_e = k_e \dot{\theta}_m$

Conclusion:
$\langle i_m \rangle = ?$

Motor Resistance?
Peak current?

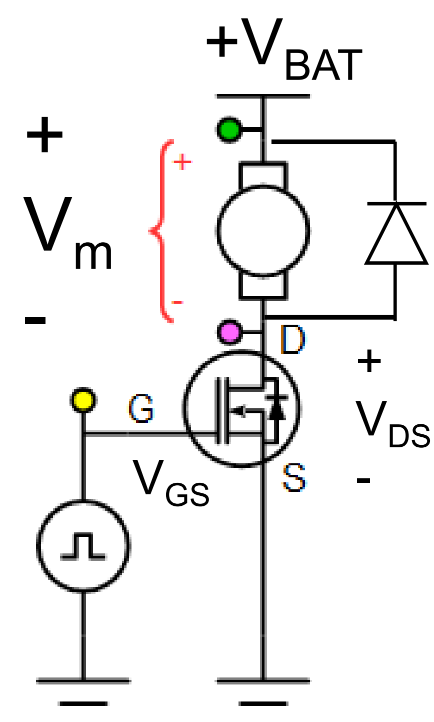
Motor Electrical Model



Motor Electrical Model
Back EMF
Motor electromechanical behavior

Also- see motor worksheet……

$$i_m = \frac{V_{BAT} - k_e \dot{\theta}_m}{R_m}$$



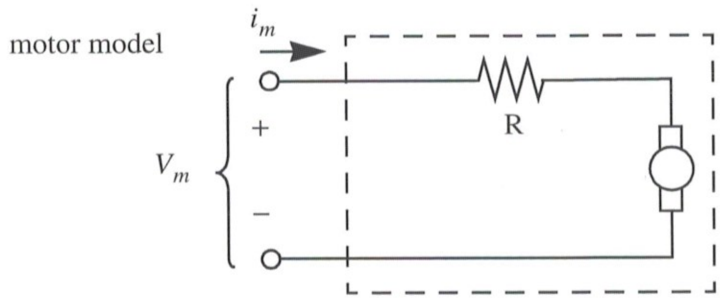
Motor Resistance?
Peak current?

Conclusion:
$<i_m>=?$

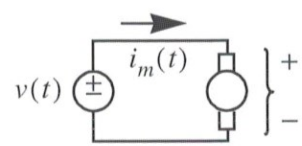
Motor model

For this problem, consider a DC permanent magnet motor (as used in your car). The car is on a carpet and moves in a straight line with no slip between the wheels and the carpet. The car is initially moving at a speed of 2 meters per second.

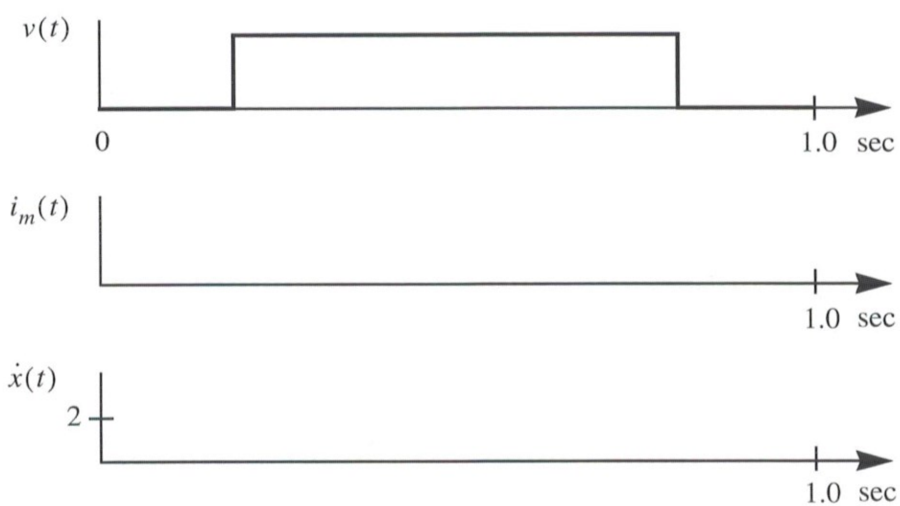
You can assume a motor model as shown below. The qualitative shape of the curves is more important than magnitudes.



[4 pts.] **a)** Consider the motor driven from a voltage source with voltage $v(t)$, as shown. Sketch car velocity $\dot{x}(t)$ and motor terminal current for the time indicated.

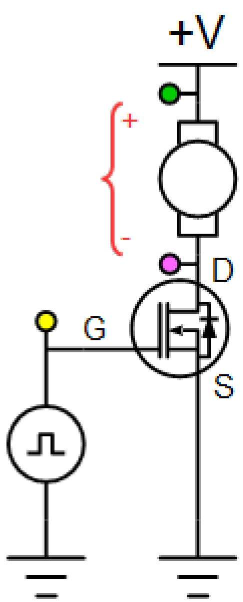


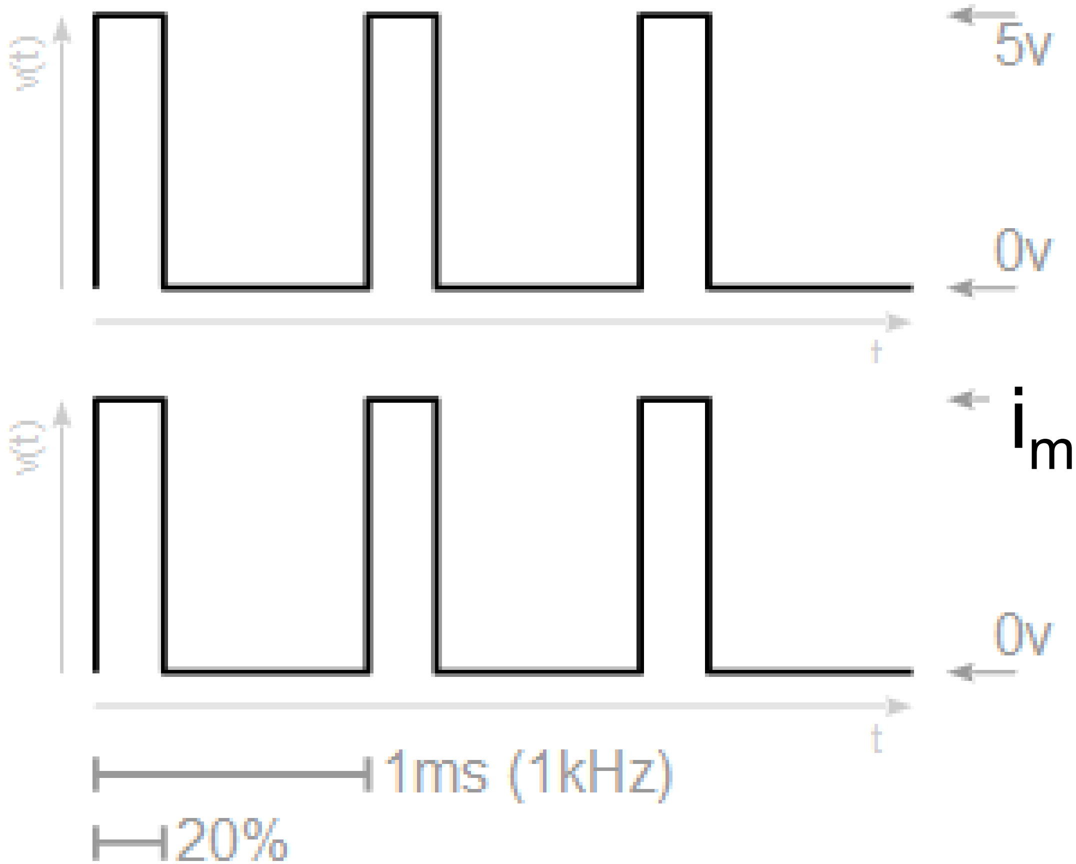
Let peak speed = 5 m/sec
Accel = 5 m/s^2
k_e = 1 v/(m/sec)
On board



(for answer
see sp99 final solution)

PWM for Main Motor control

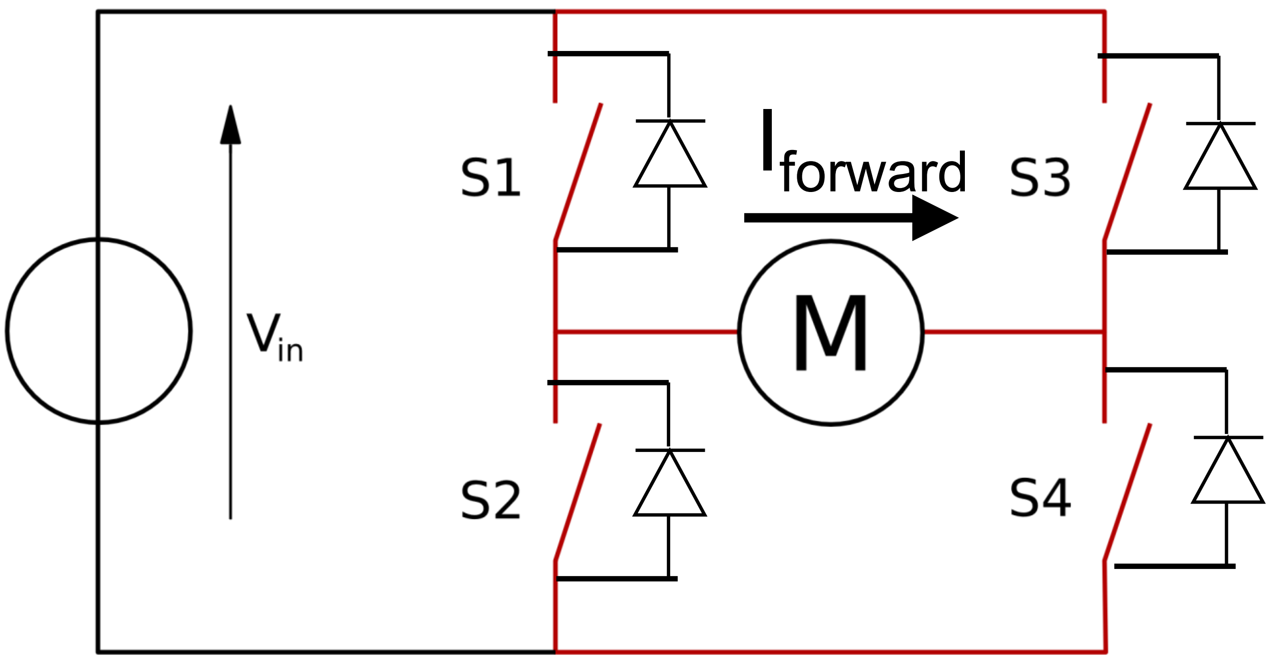




$<i_m> = (T/T_o)\, i_{max}$

Is i_{max} constant?

H Bridge Concept



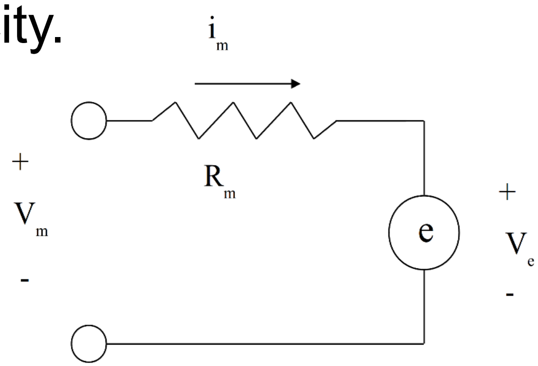
S1	S2	S3	S3	Function?
Off	Off	Off	Off	
On	Off	Off	On	
Off	On	On	Off	
On	On	Off	Off	
On	Off	On	off	
Off	On	Off	on	

Practice Q2

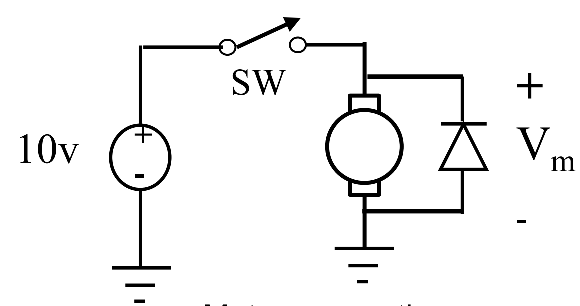
Consider a DC permanent magnet motor (as used in your car).
The car is initially at rest. The motor is connected as shown below.
Neglect battery and switch resistance. Neglect motor inductance. Assume diode is ideal.
Assume motor resistance = 0.2 ohm, and that the car accelerates to 4 m/s in 2 seconds.
Assume back EMF constant is 1V/(m/s).
Assume time constant for deceleration is 1 second.

Switch turns on at 0 sec, off at 2 sec.

Complete the sketches below for motor current i_m, motor voltage V_m, and car velocity.



Motor model

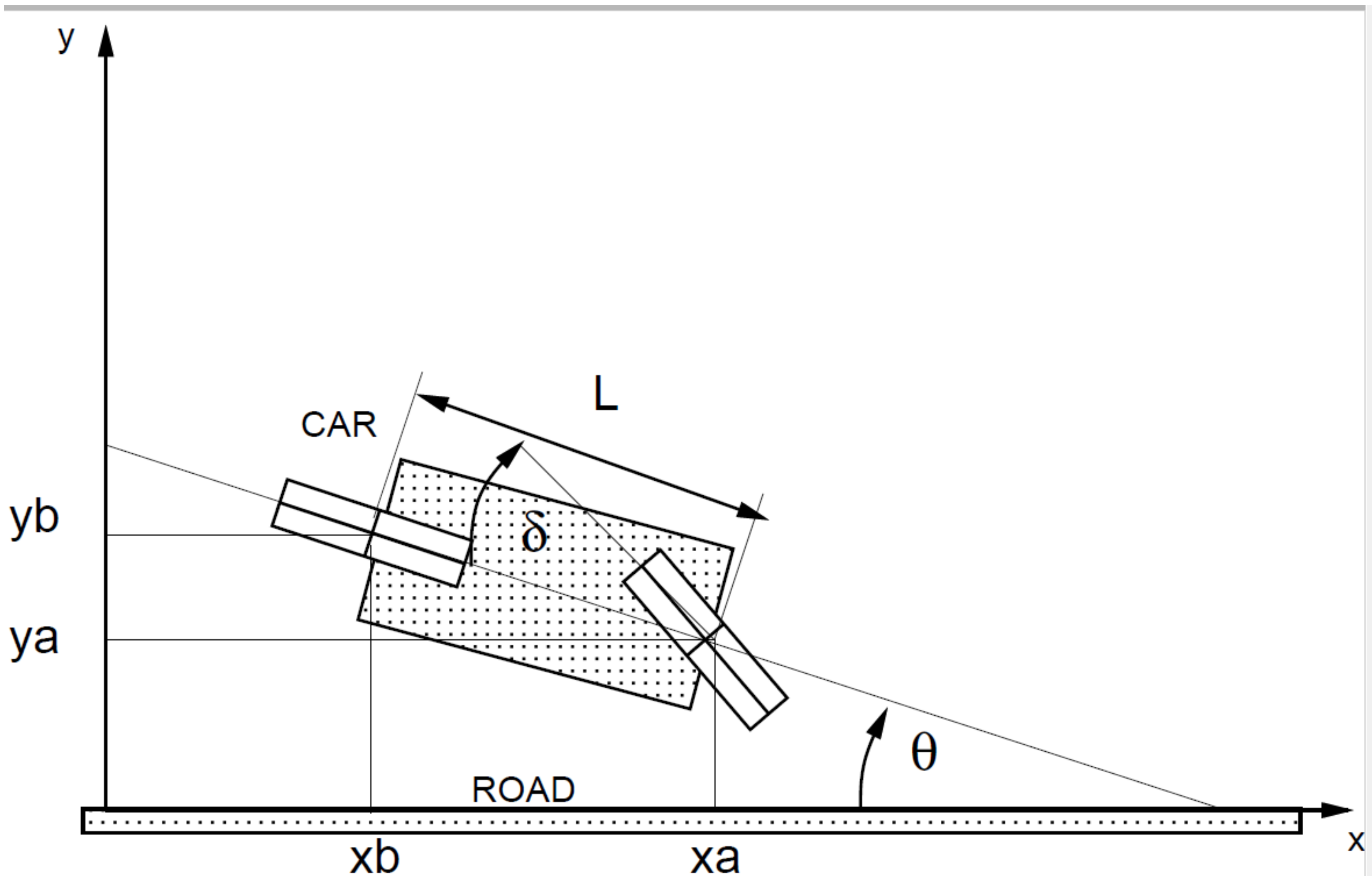


Motor connection

EECS192 Lecture 6
Motor Modelling and Steering Introduction
Feb. 23, 2021

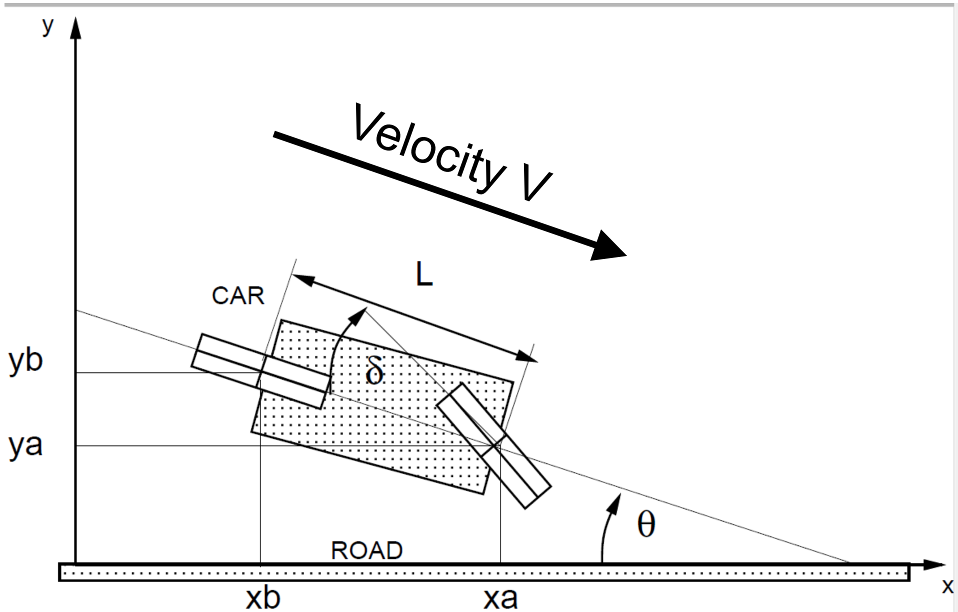
Topics
- Checkpoint 5:
- Available space survey- circle/figure 8
- Checkpoint 6:
- Motor model (for velocity control)
- Steering Introduction
- Simulator (preview)

Bicycle Steering Model



Bicycle Steering Model

More detailed models see: https://inst.eecs.berkeley.edu/~ee192/sp15/refSteer.html



$$\dot{x}_b = V cos(\theta(t)) \quad (1)$$

$$\dot{y}_b = -V sin(\theta(t)) \quad (2)$$

$$\dot{\theta} = \frac{V}{L} tan(\delta(t)) \quad (3)$$

$$y_a = y_b - L sin(\theta(t)) \quad (4)$$

Bicycle Steering Model-linearized

$$\dot{x}_b = V cos(\theta(t))$$

$$\dot{y}_b = -V sin(\theta(t))$$

$$\dot{\theta} = \frac{V}{L} tan(\delta(t))$$

$$y_a = y_b - L sin(\theta(t))$$

Original non-linear equations

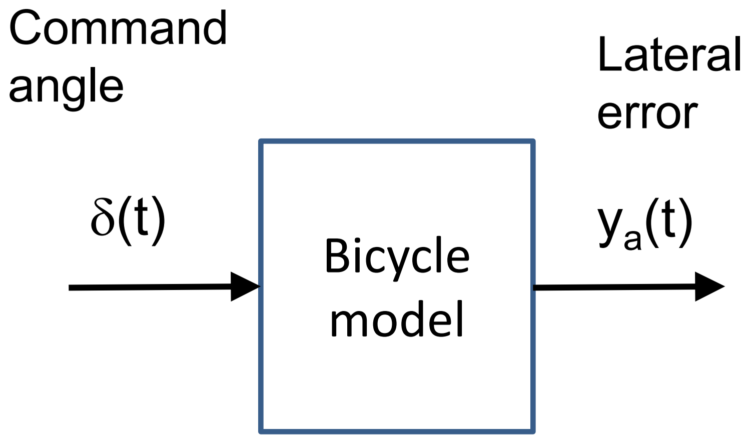
<u>Assume small angle, constant V:</u>

$$\dot{y}_b \approx -V\theta$$

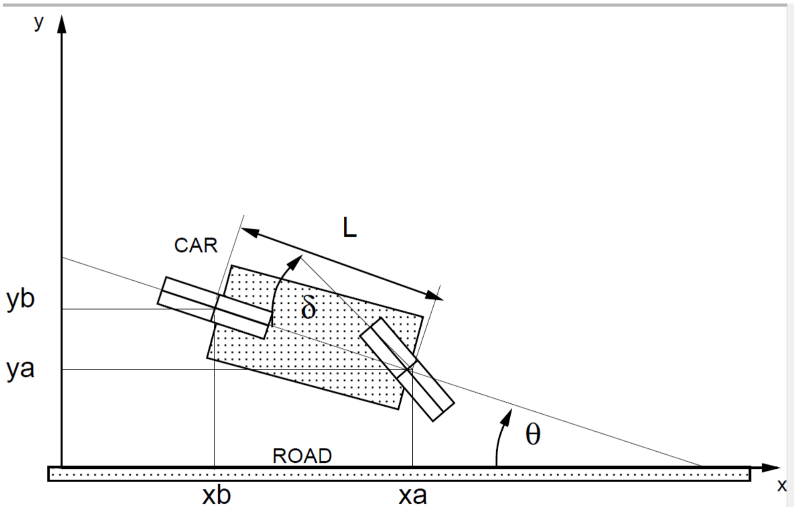
$$\dot{\theta} \approx \frac{V}{L}\delta(t)$$

$$\dot{y}_a \approx \dot{y}_b - L\dot{\theta} = -V\theta - L\dot{\theta}$$

$$\ddot{y}_a = \frac{-V^2}{L}\delta(t) - V\dot{\delta}(t).$$

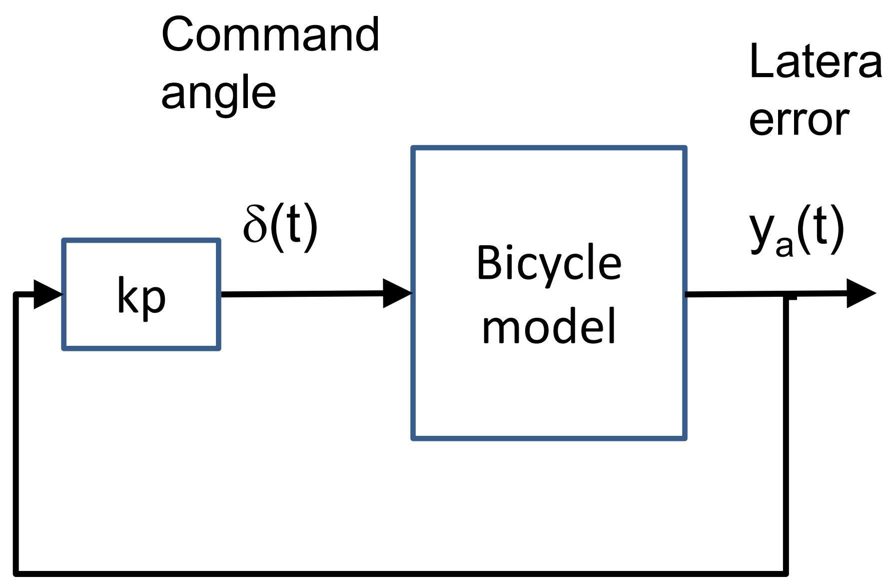


Bicycle Steering Model



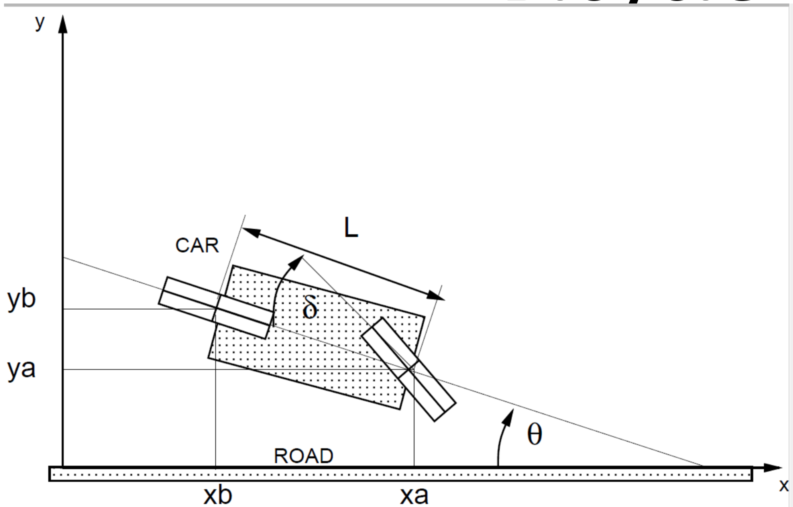
Proportional control:

$$\delta(t) = k_p y_a(t)$$



Check angle in your car, check sign of kp…

Bicycle Steering Model



Proportional control:

$$\delta(t) = k_p y_a(t)$$

$$\ddot{y}_a + V k_p \dot{y}_a(t) + \frac{V^2}{L} k_p y_a(t) = 0.$$

Laplace transform:

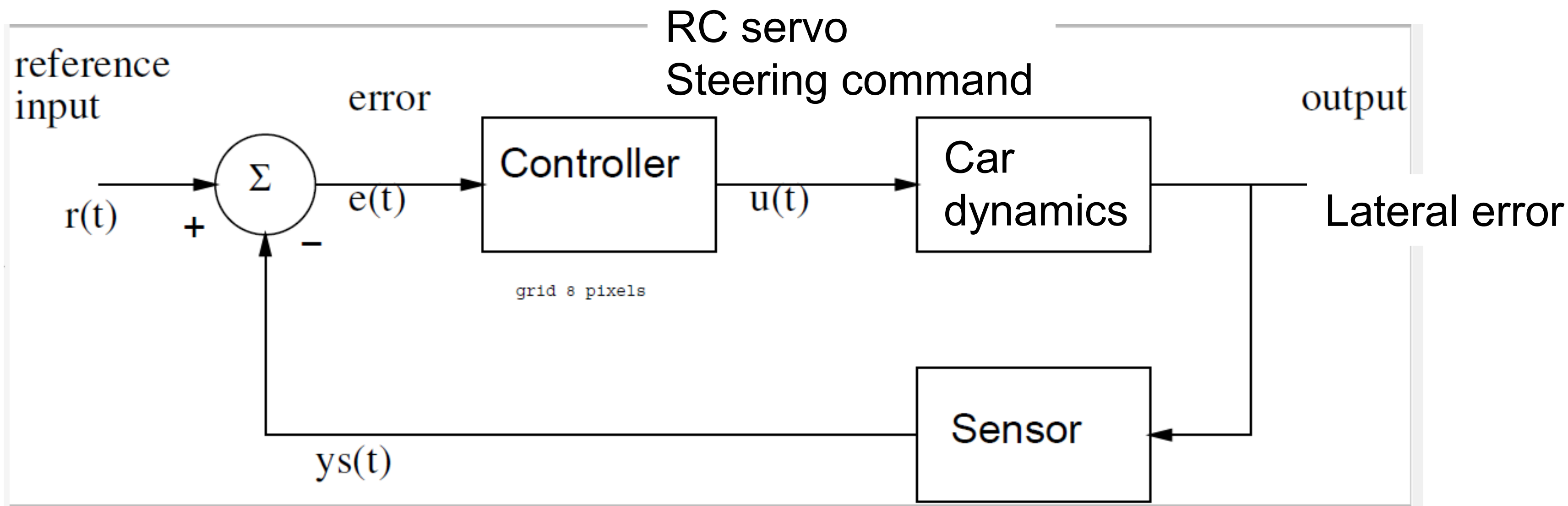
$s^2Y(s) + V\, k_p\, s\, Y(s) + (V^2/L)\, k_p\, Y(s) =$

$+s\, y(0^-) + y'(0^-) + V\, k_p\, y(0^-)$ (initial conditions)

Eigenvalues:

$$\lambda_{1,2} = \frac{V}{2}\left(-k_p \pm \sqrt{k_p^2 - \frac{4k_p}{L}}\right)$$

Steering Control overview



Offset from track
r(t) =0 (mostly)
Where might offset be useful?

Check sign for kp….

Proportional control:
u = kp*e = kp* (r-y);

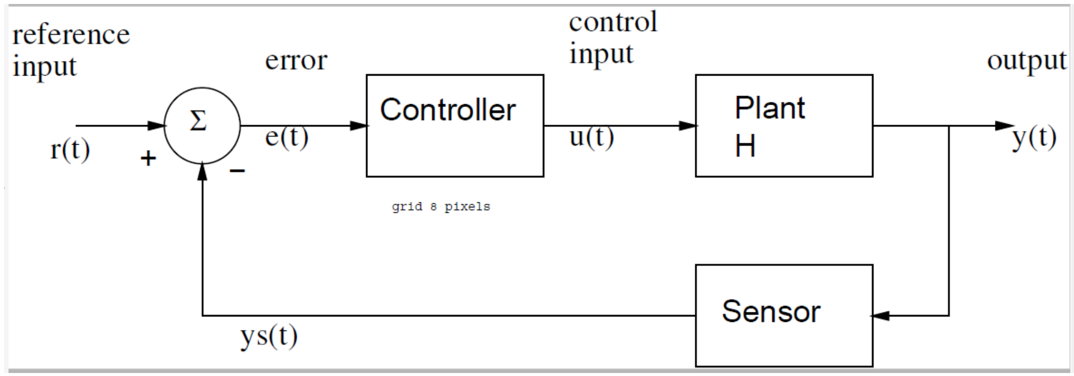
Proportional + derivative control:

```
u = kp*e + kd * y_sum;
y_dot = (y - y_old)/T;
```

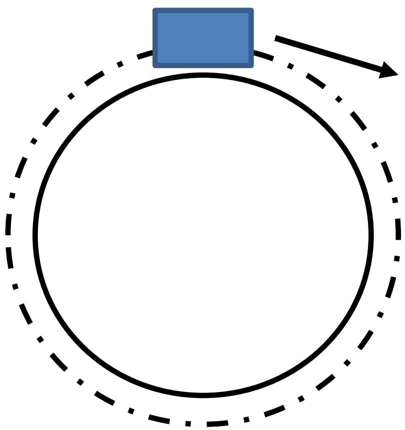
Proportional + integral control

```
u = kp*e + ki * e_sum;
e_sum = e_sum + e;
```

Bicycle Steering Control



Note steady state error: car follows larger radius

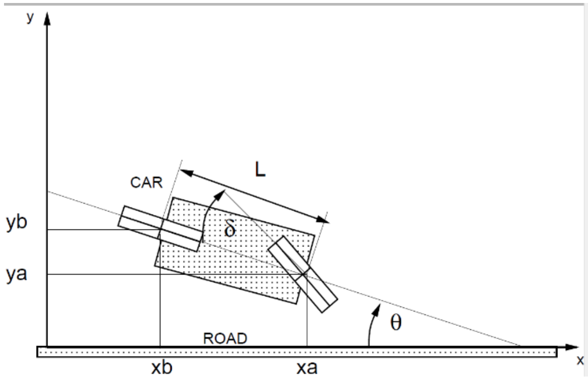


Proportional control:
r = 0 (to be on straight track)
δ=u = kp*e

Proportional+derivative

P+I+D

Bicycle Steering Model- Proportional control



Proportional control: $\delta(t) = k_p y_a(t)$

$$\ddot{y}_a + V k_p \dot{y}_a(t) + \frac{V^2}{L} k_p y_a(t) = 0.$$

Eigenvalues: $$\lambda_{1,2} = \frac{V}{2}\left(-k_p \pm \sqrt{k_p^2 - \frac{4k_p}{L}}\right)$$

Critical damping:

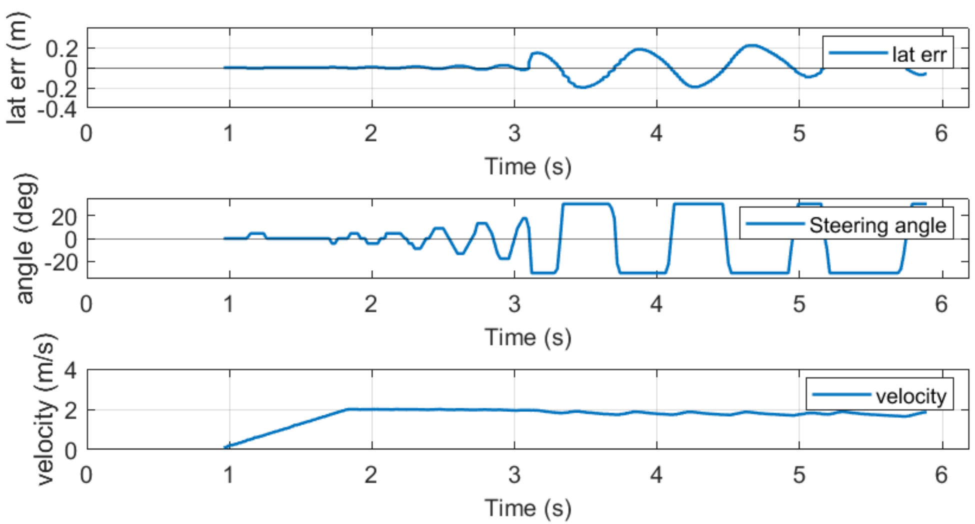
$\lambda_1 = \lambda_2$ ➔ $k_p^2 = 4\, k_p/L$ or

$k_p = 4/L = 4/0.3$ m = 13 rad/m = 760 deg/m

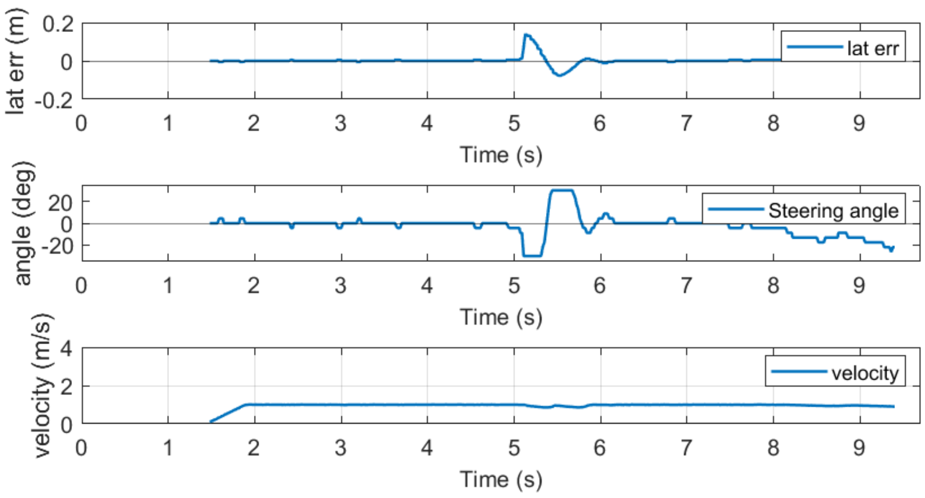
At 2 m/s, doesn't work well- servo saturates, also other non-linear dynamics…

Bicycle Steering Model- Proportional control

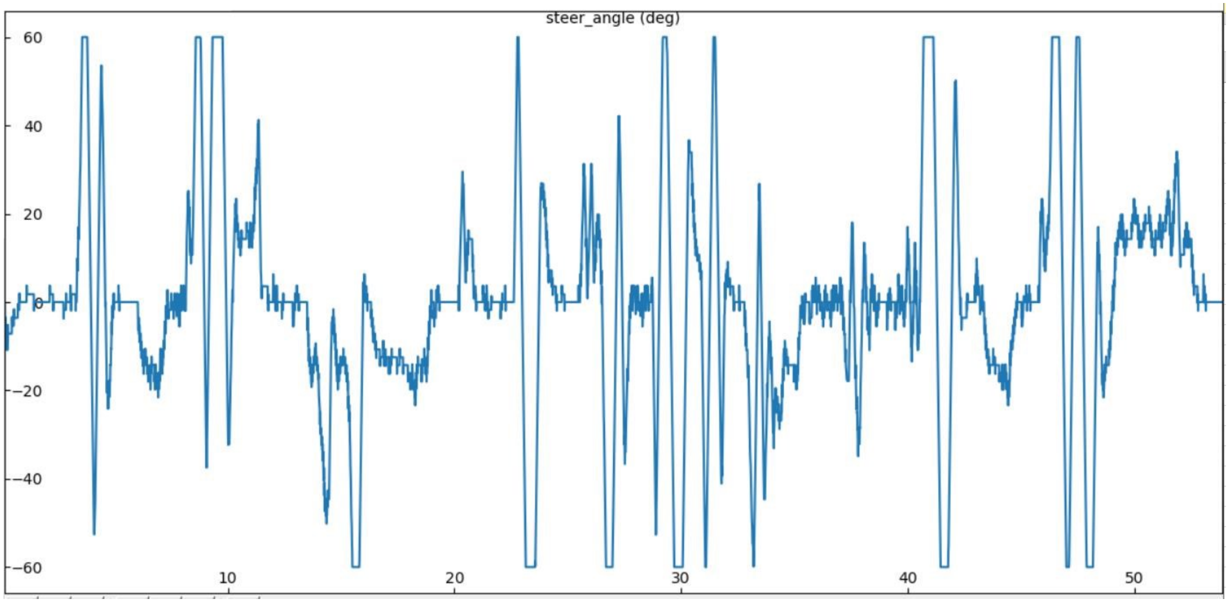
2 m/s k_p =800 deg/m Kd = 0

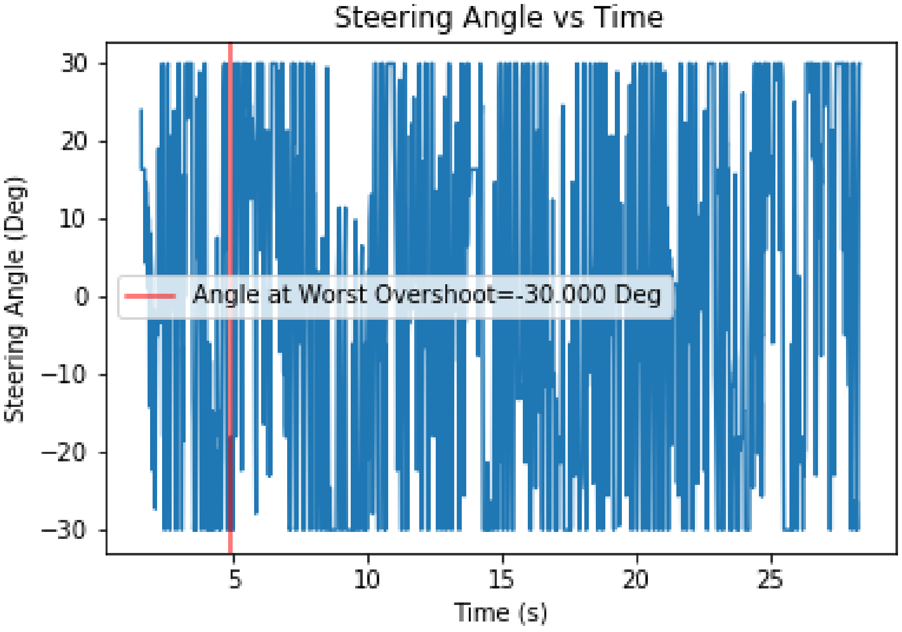


1 m/s k_p =800 deg/m Kd = 0



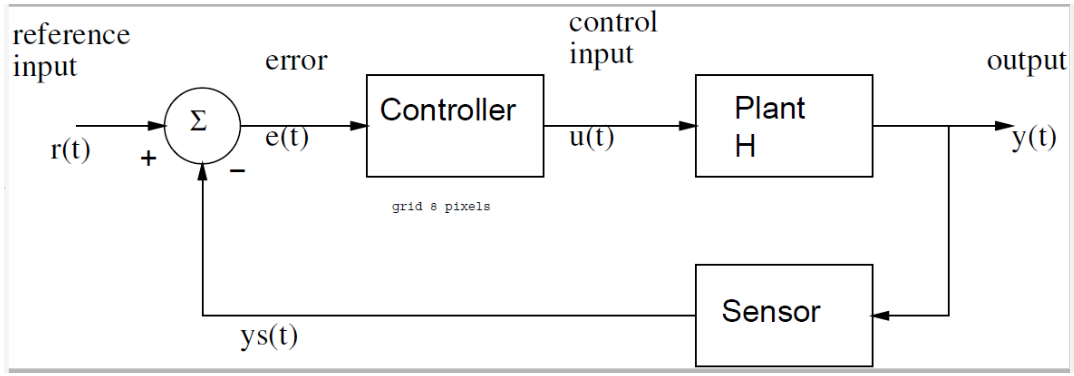
Steering saturation



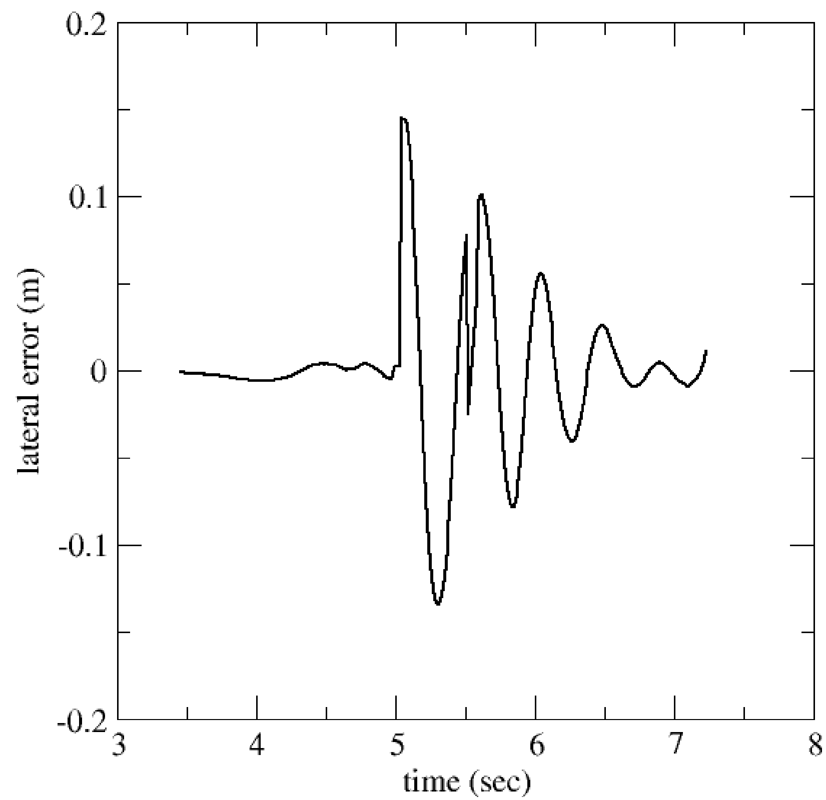


Lesson: if tracking is good, steering angle change is small

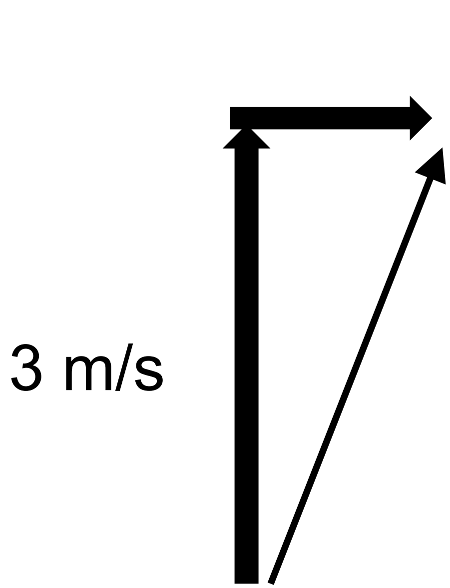
Steering Control- PD

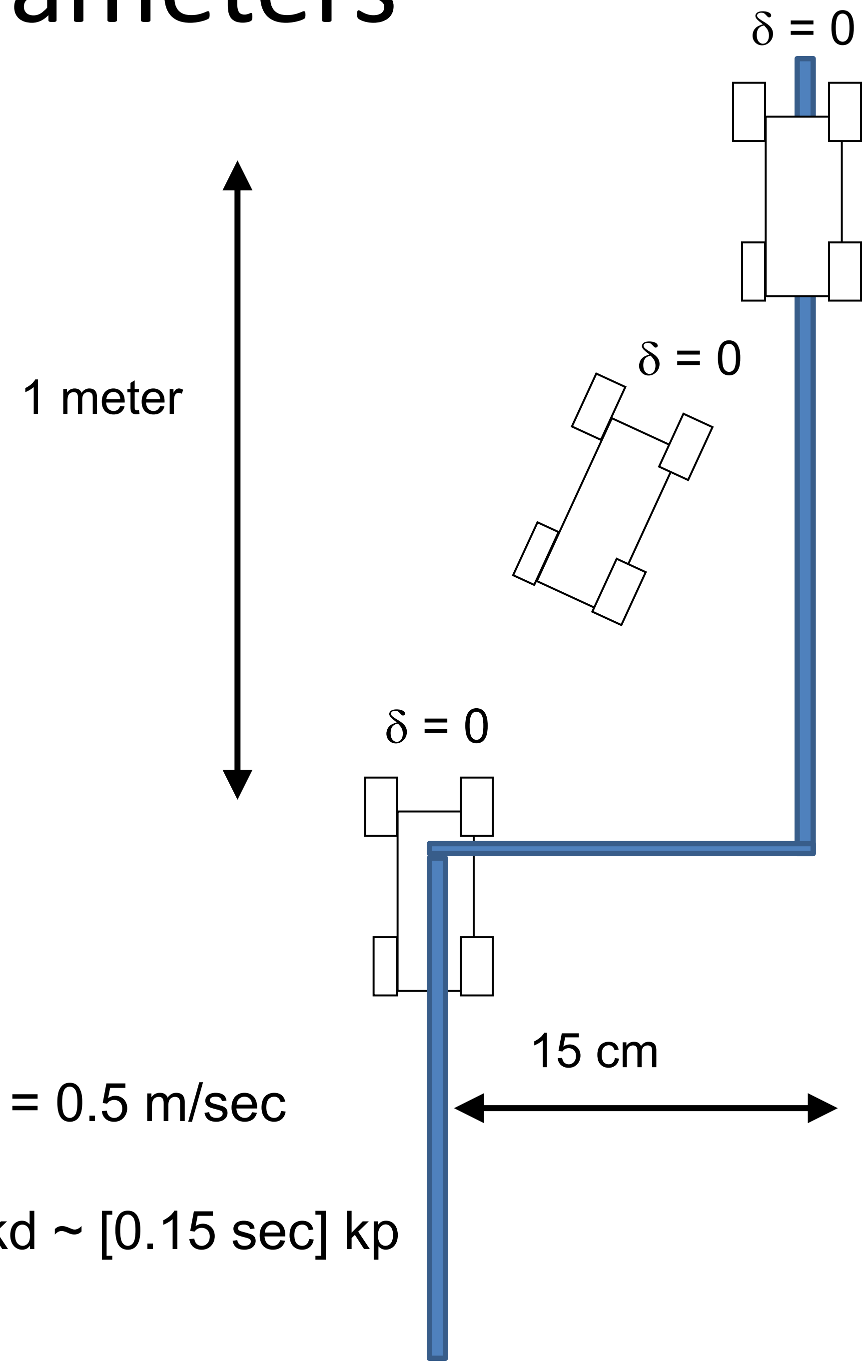


Example under-damped steering:



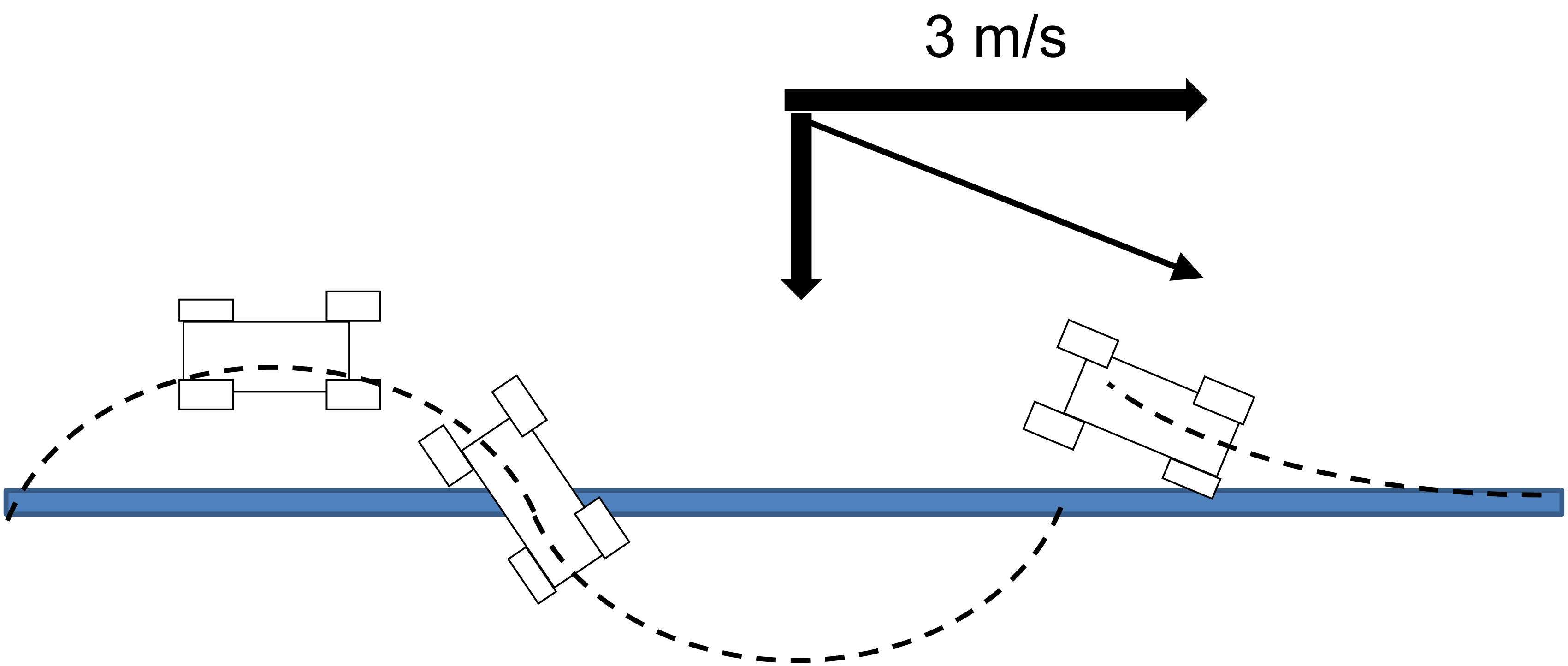
PD parameters





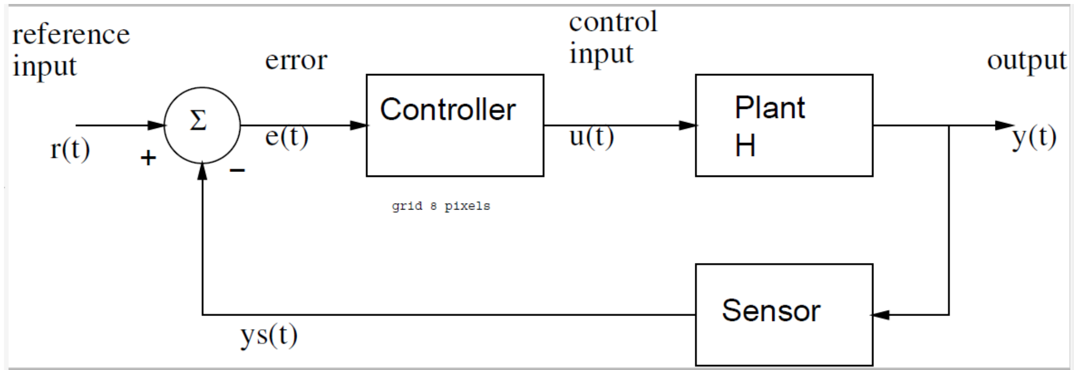
Step: 15 cm
Choose step response 1m = 300 ms
Then lateral velocity = 15 cm/300 ms = 0.5 m/sec
At mid point:
$\delta = 0 = kp\ 7.5\ cm + kd\ 0.5\ m/sec$ ➔ kd ~ [0.15 sec] kp

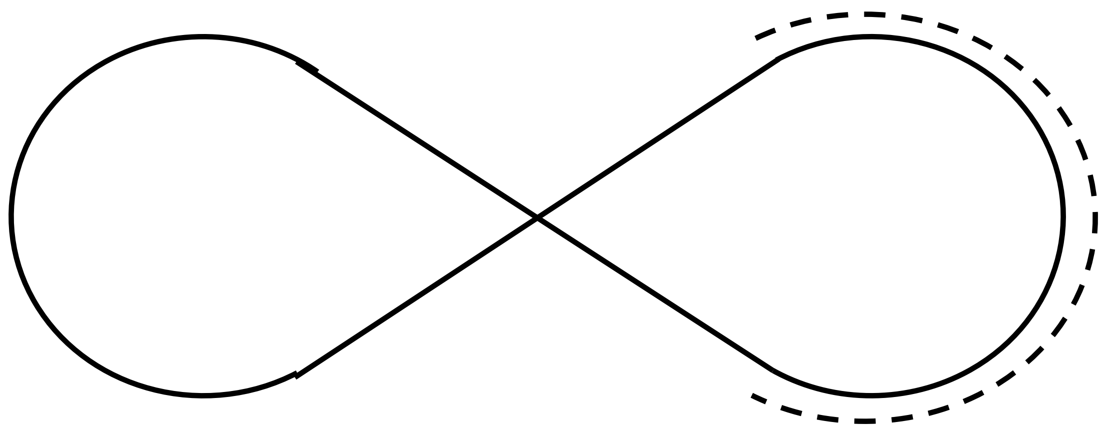
PD parameters



Step: 15 cm
Choose step response 1m = 300 ms
Then lateral velocity = 15 cm/300 ms = 0.5 m/sec

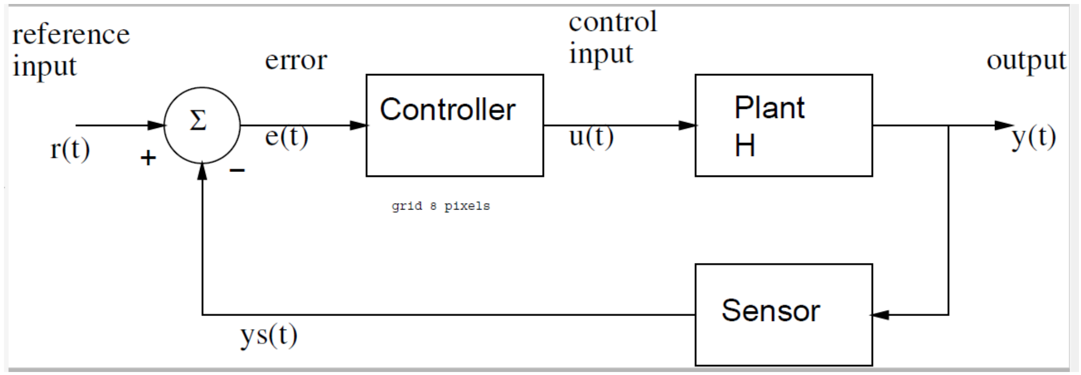
Proportional + Integral

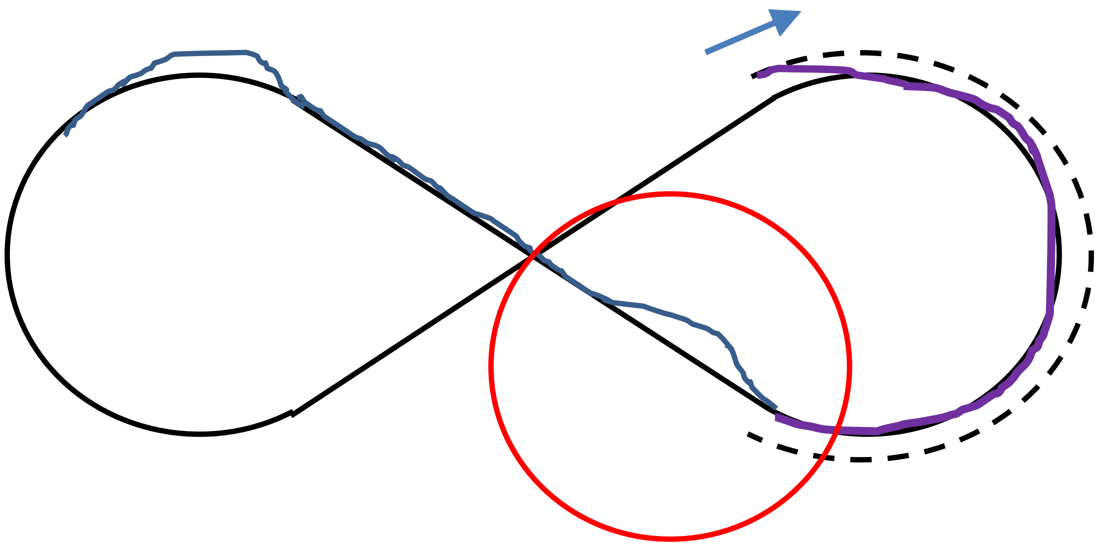




Anti-windup

Proportional + Integral





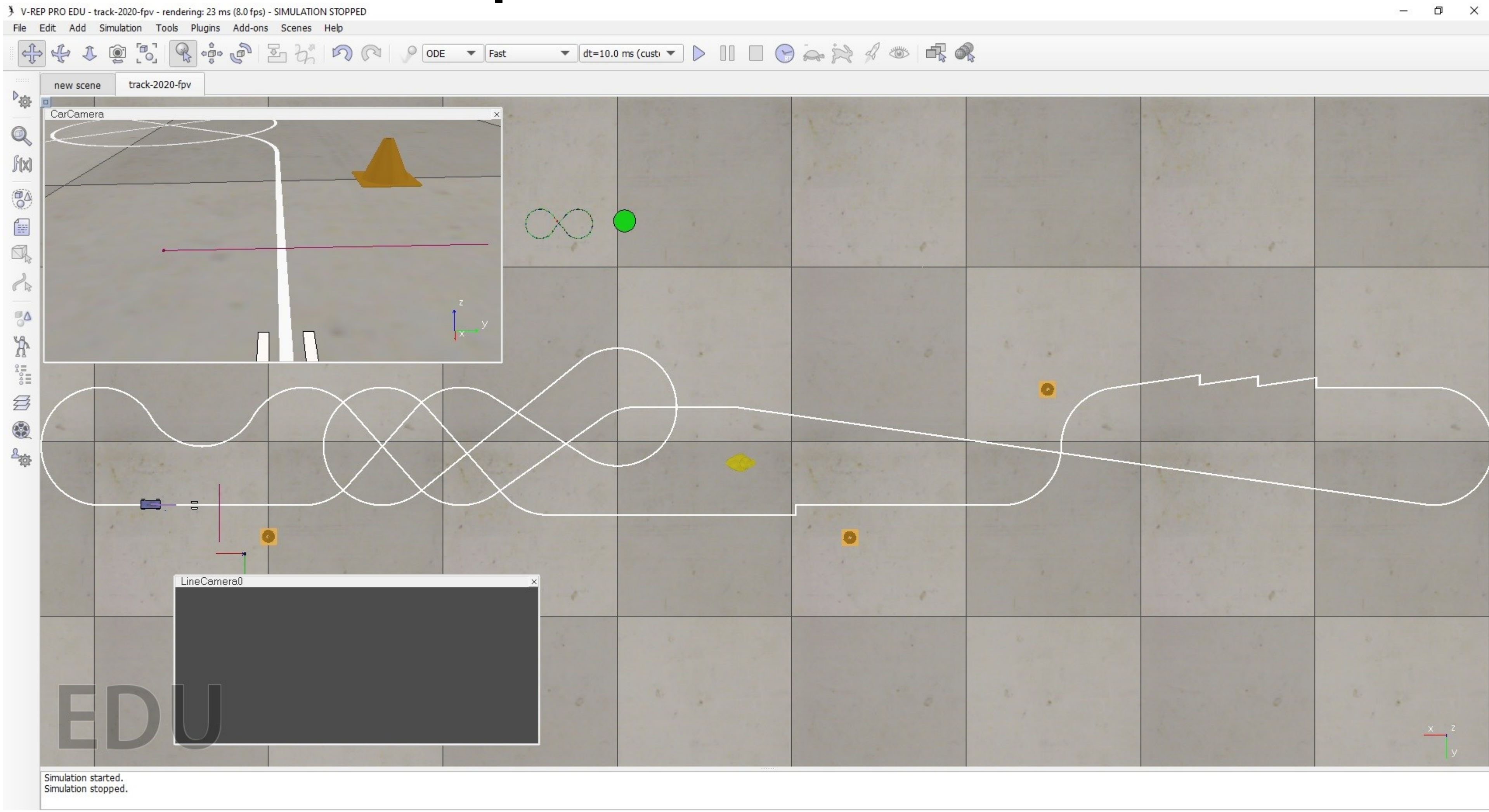
P+I control: delta =kp e + ki (integral e)
P control: delta = kp e

Anti-windup

EECS192 Lecture 6
Motor Modelling and Steering Introduction
Feb. 23, 2021

Topics
- Checkpoint 5:
- Available space survey- circle/figure 8
- Checkpoint 6:
- Motor model (for velocity control)
- Steering Introduction
- Simulator (preview)

V-rep simulation - FPV

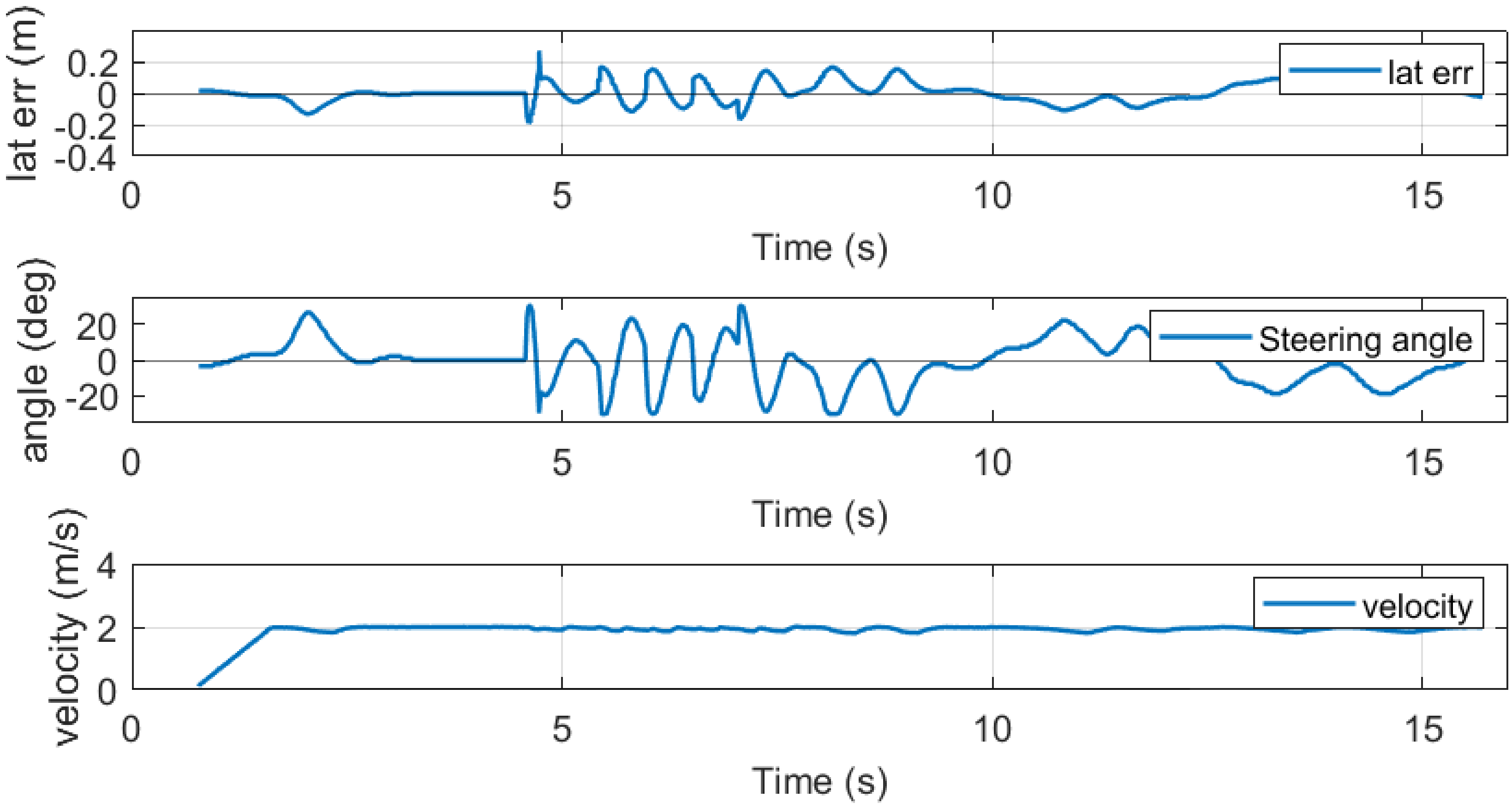


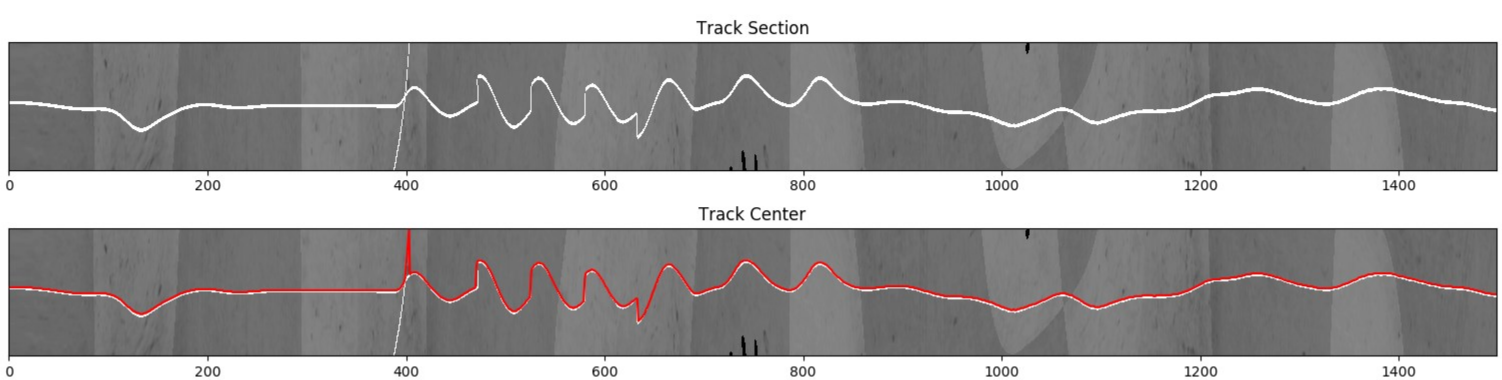
demo

Round2-fastvcar-v2 + simulator-pub/controller.py

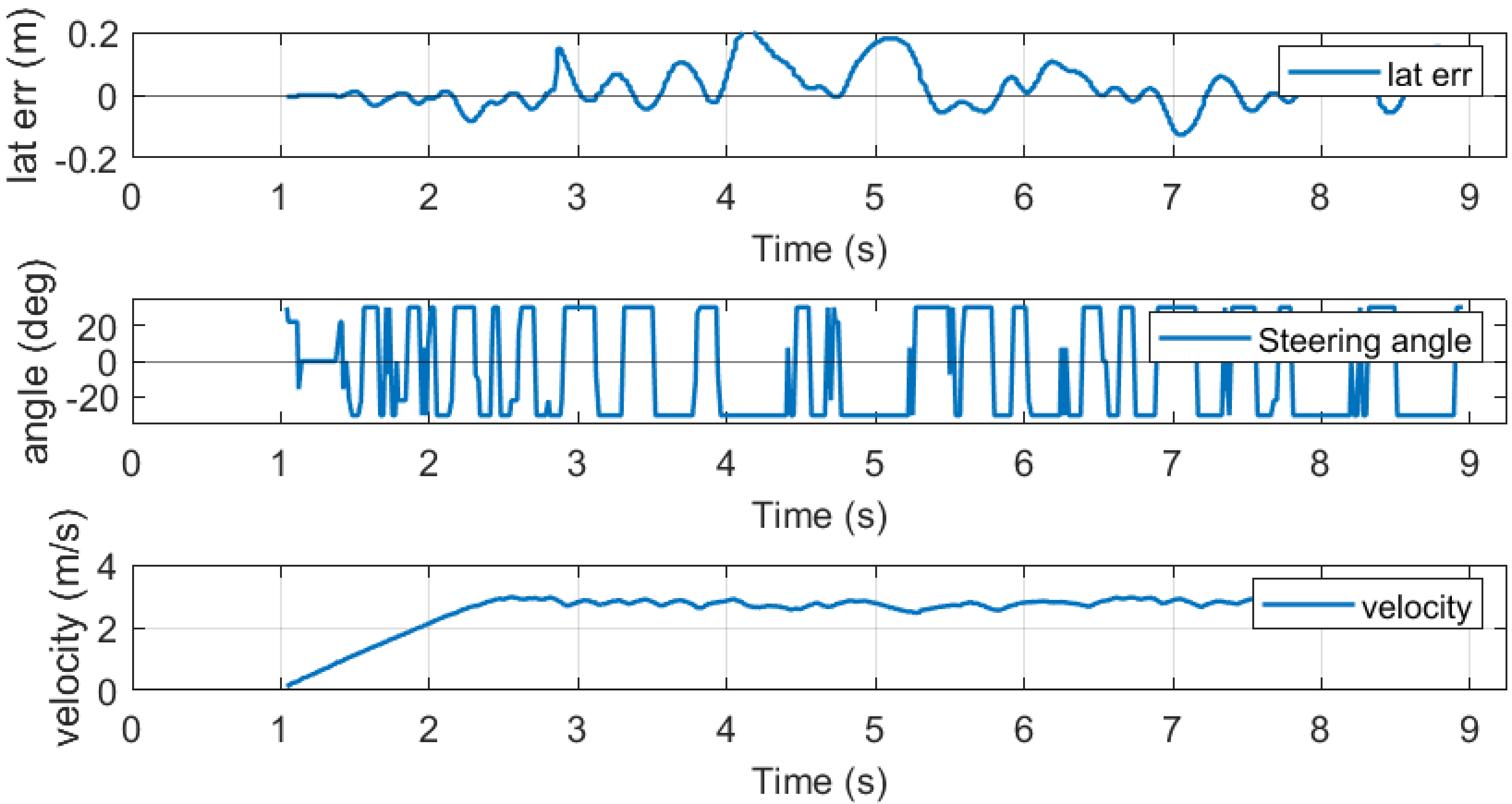
Extra Slides

V-rep simulation

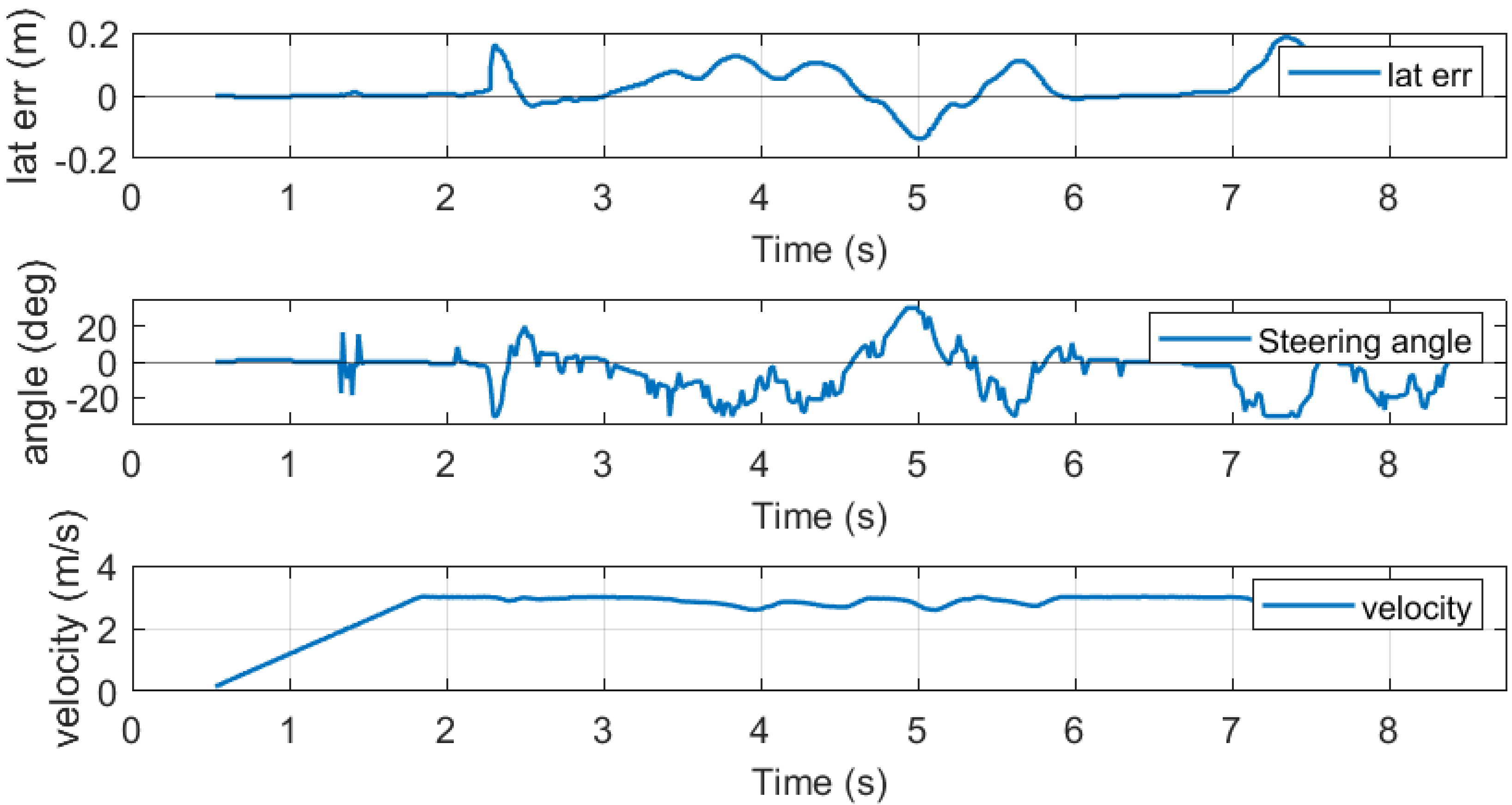




Proportional + derivative control.
Kp = 40 deg/cm, 70 rad/m
Kd = 1000 deg/(m/sec)
V=3 m/s, slew rate 600 deg/0.16 sec
NOTE: = bang-bang!
What is problem with bang bang?
Break servo, nonlinear (unstable)

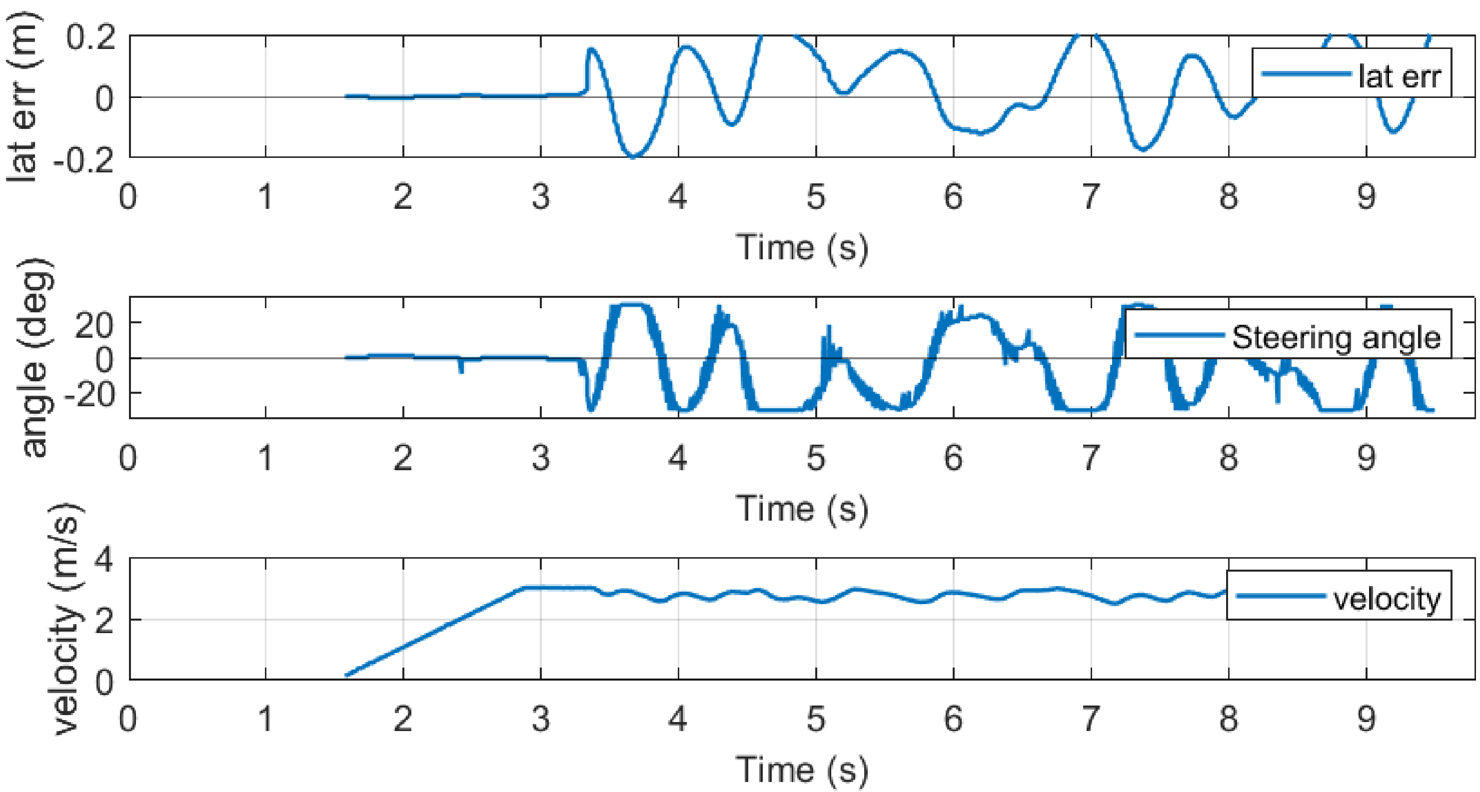


Proportional + derivative control.
Kp = 200 deg/m,
Kd = 30 deg/(m/sec) = (0.15 sec) Kp
V=3 m/s, slew rate 600 deg/0.16 sec
NOTE: = not bang-bang

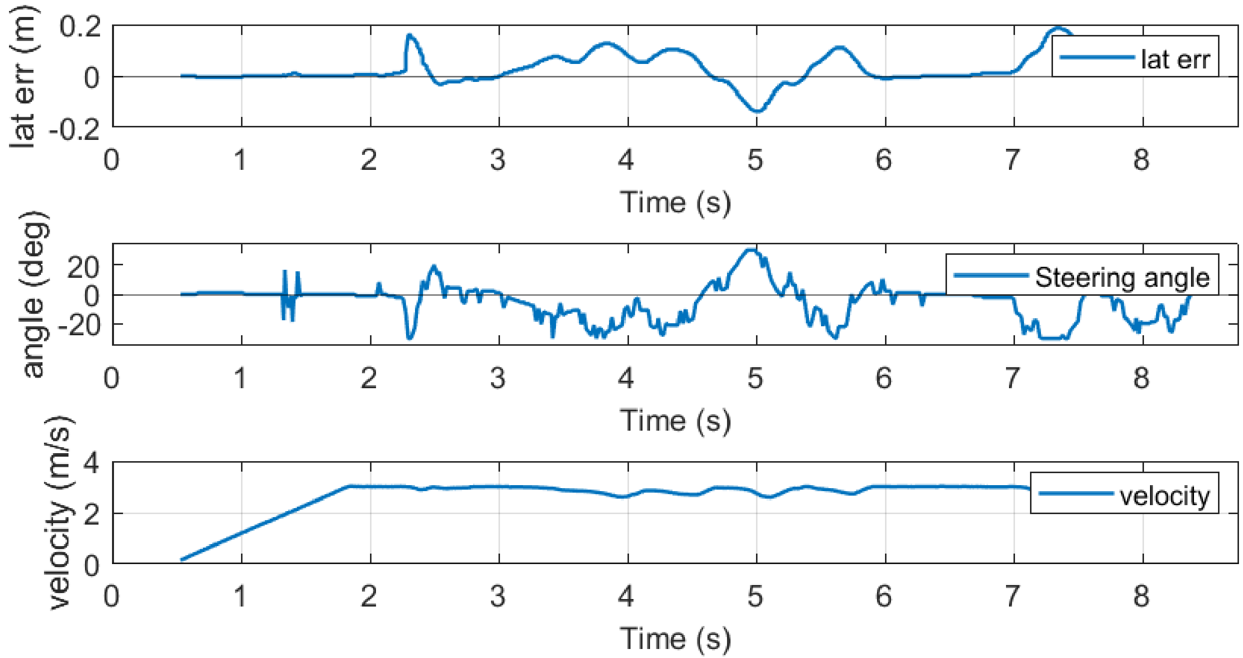


Proportional + derivative control.
Kp = 200 deg/m, Kd = 30 deg/(m/sec)
V=3 m/s NOTE: NO STEERING DELAY, no deadband

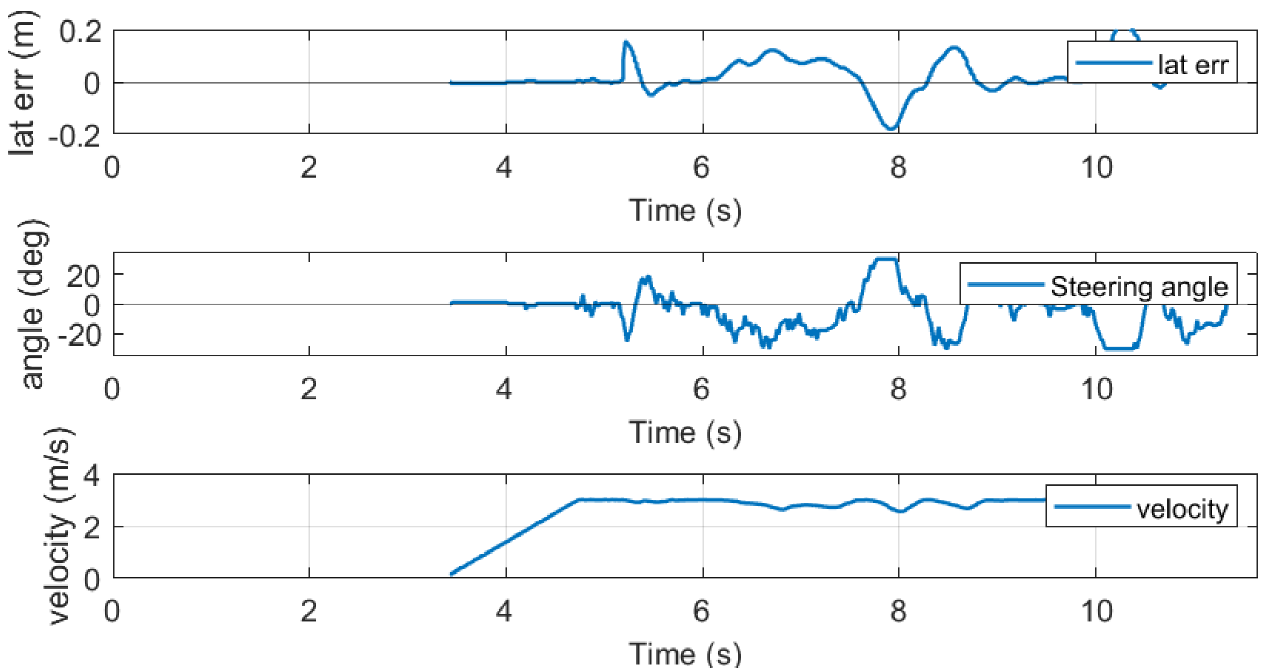
```
def set_steering_fast(self, angle_cmd, dt):
      self.steering_state = angle_cmd # update state
      self.vr.simxSetFloatSignal('steerAngle',
         angle_cmd*(math.pi/180.0), vrep.simx_opmode_oneshot)
      return(angle_cmd)
```



Kp = 200 deg/m, Kd = 30 deg/(m/sec). V=3 m/s

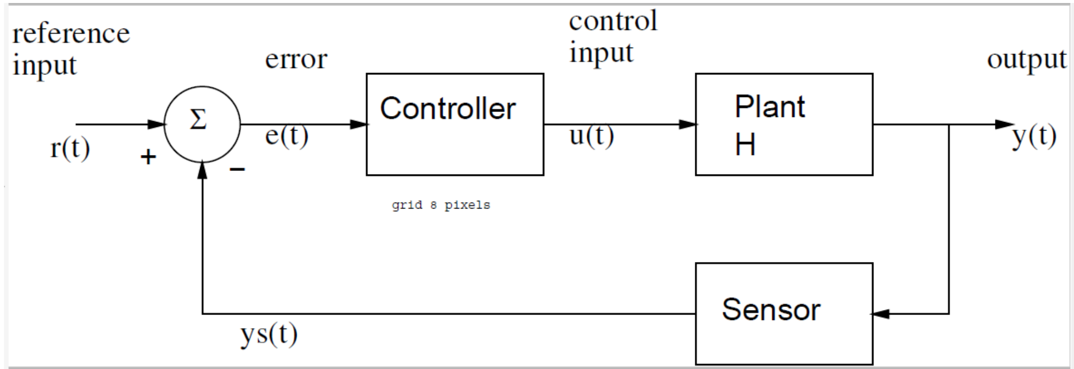


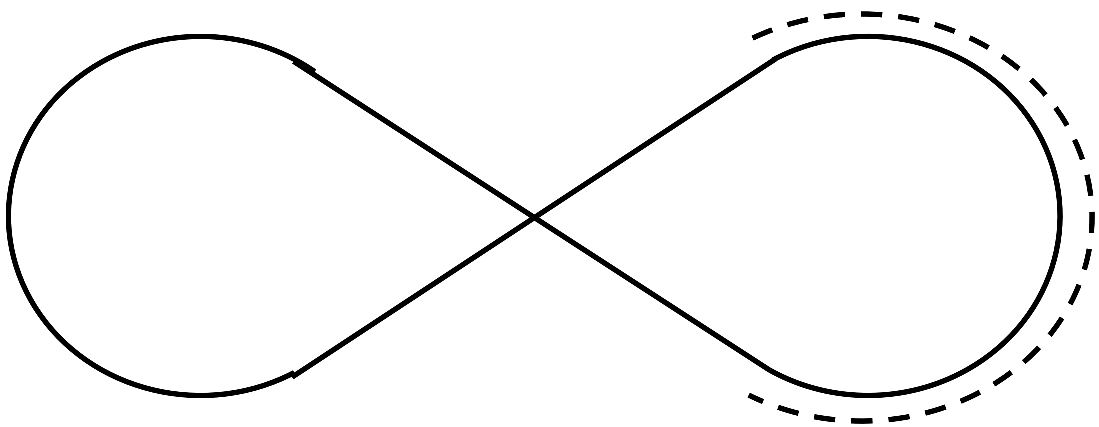
Slew 600 deg/160 ms



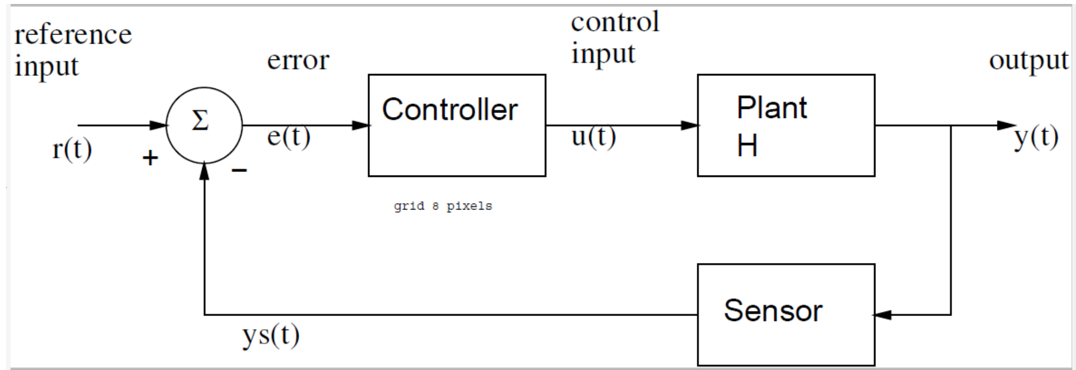
Slew 60 deg/160 ms

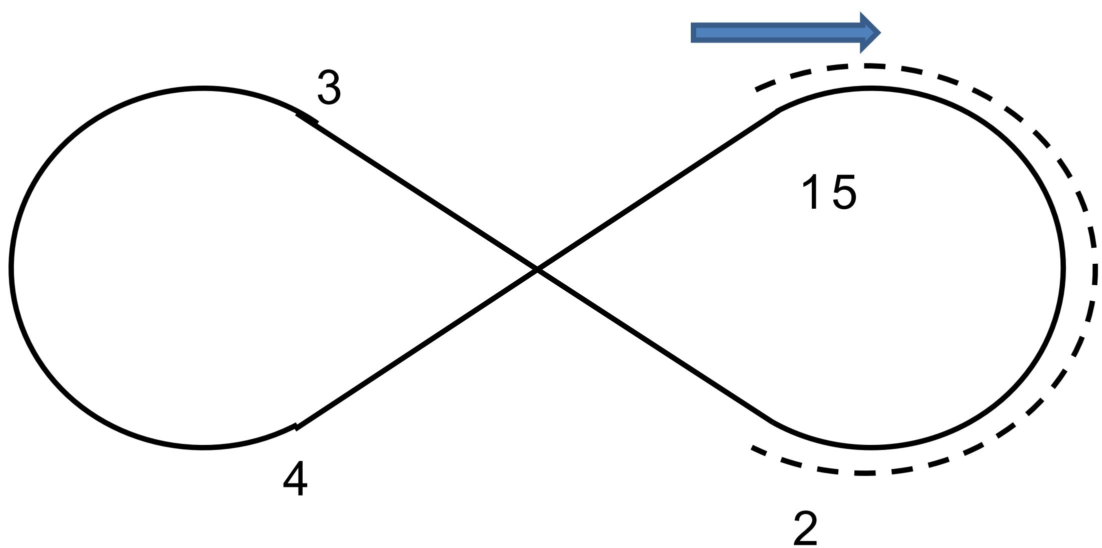
Feedforward





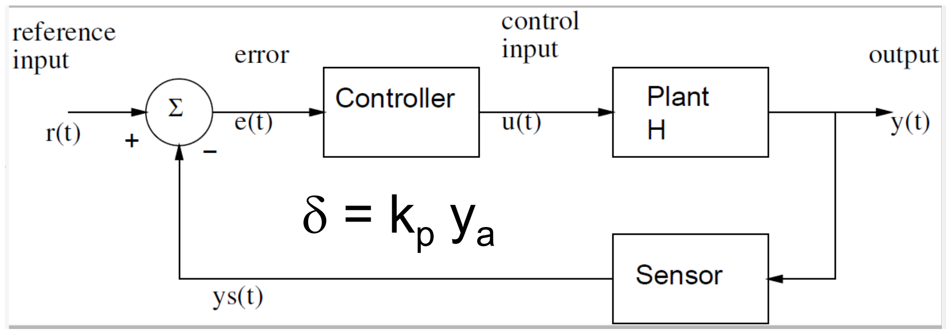
Feedforward

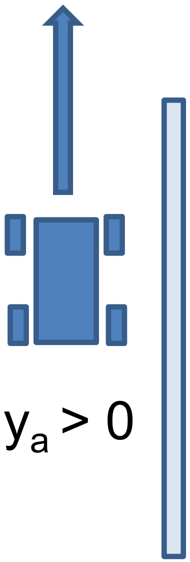


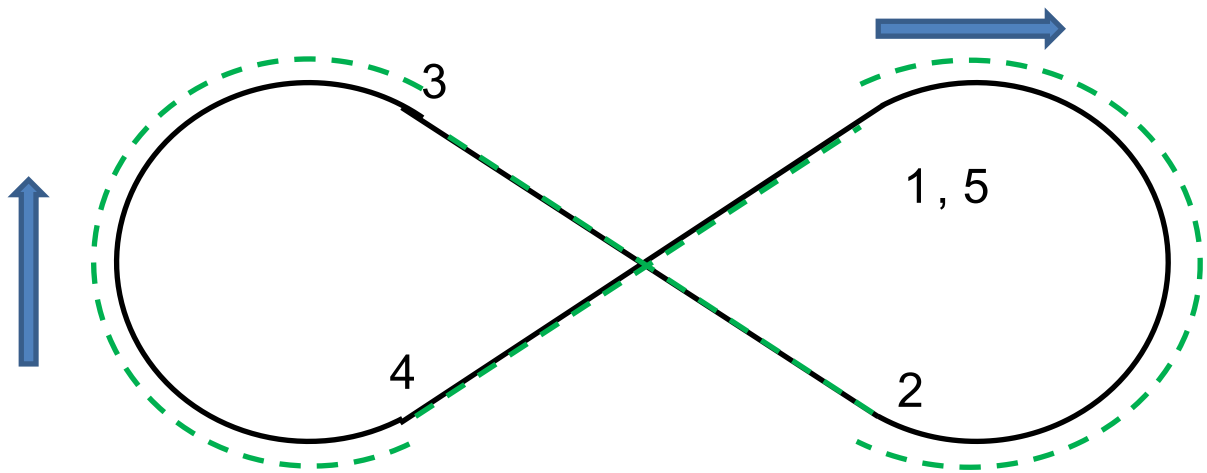


r(x) =

Feedforward using track memorization



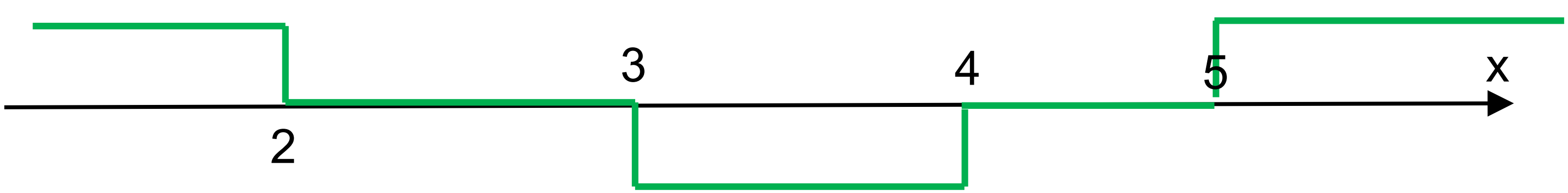




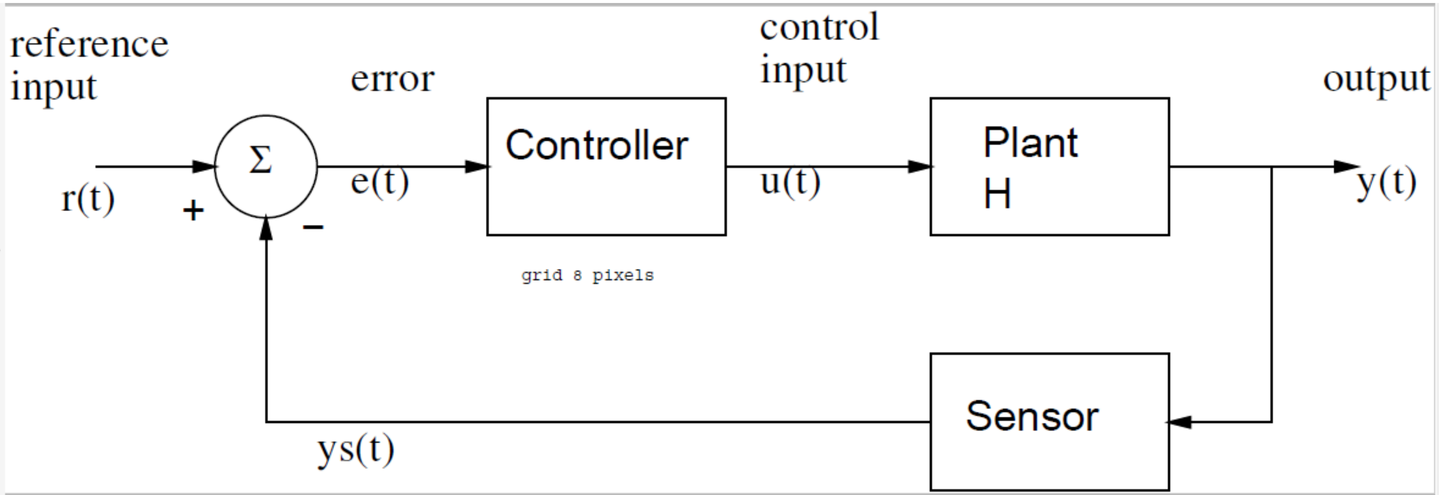
Check signs … $r(x) = - e(x + v\,\Delta t)$ preview of turn

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

r(x) =



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:

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

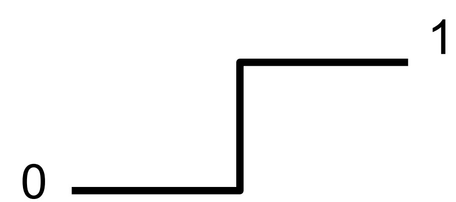
Zero Input Response (non-zero init condx, r(t)=0):

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

a'=a – b k_p b' = b k_p

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

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

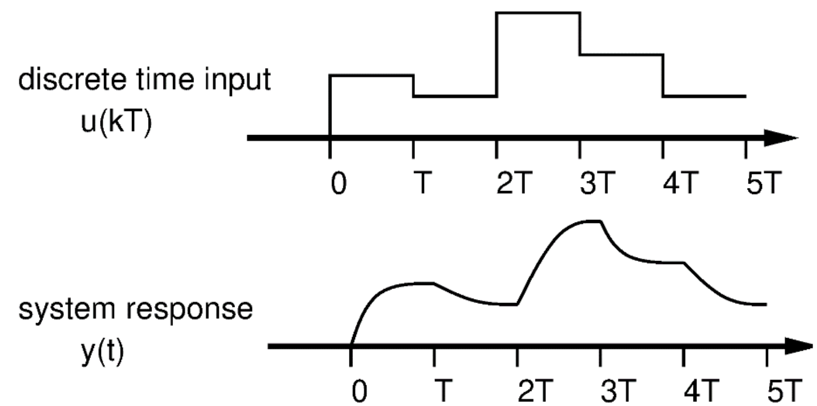


Step Response (zero init condx) by convolution:

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

Control Synopsis- Discrete Time

Superposition of Step Responses



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

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

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

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:

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

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

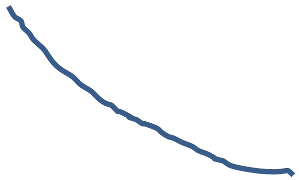
For stability:

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

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

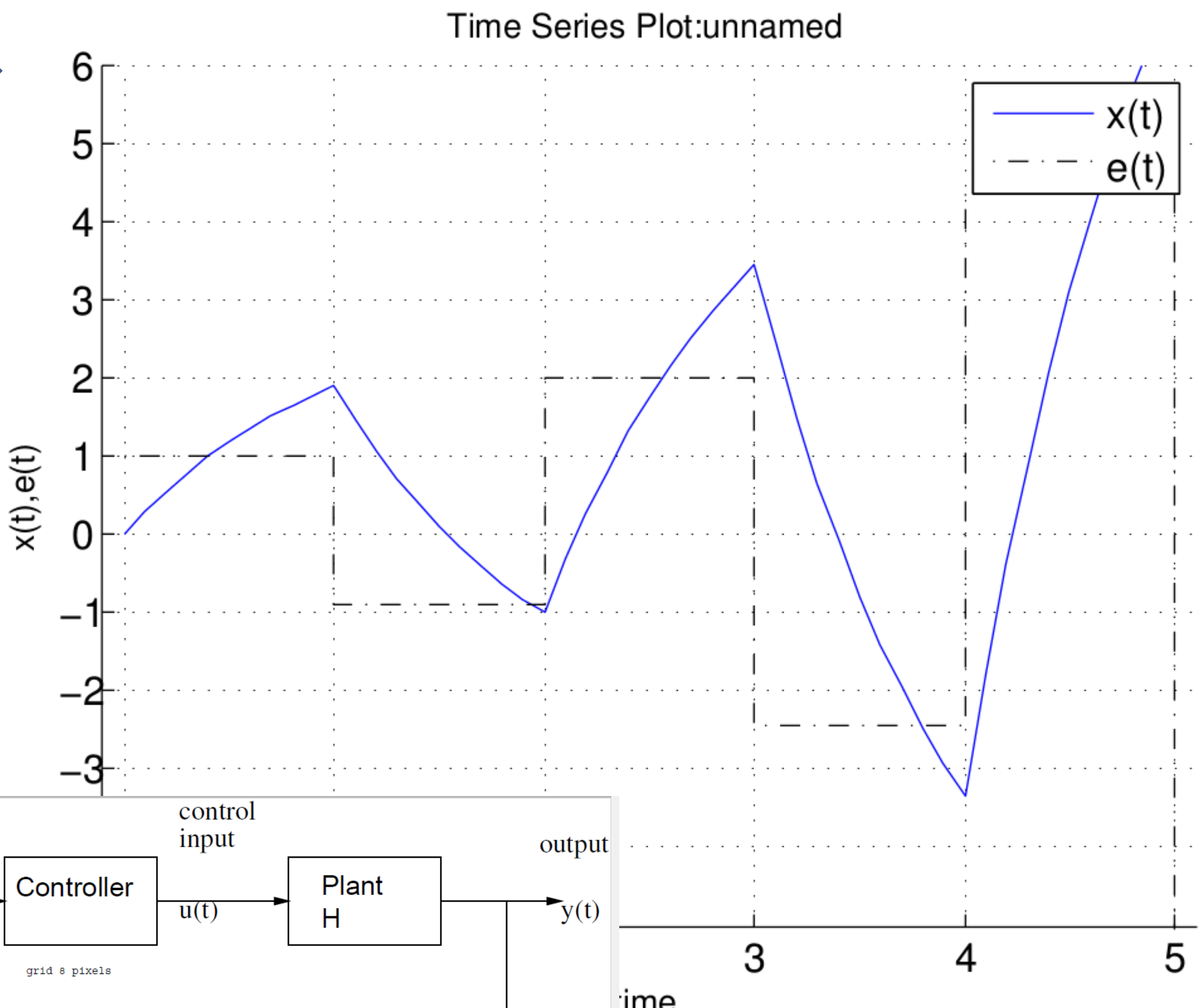
Discrete Time Control

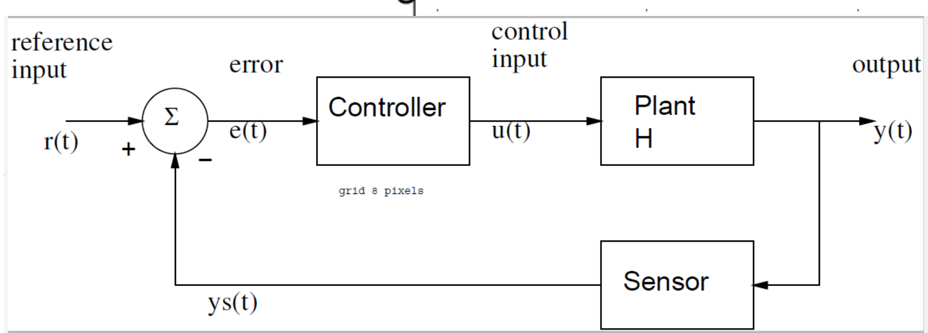
e(t), u=kp e(t)



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

Let x[k] = y[k]





On board

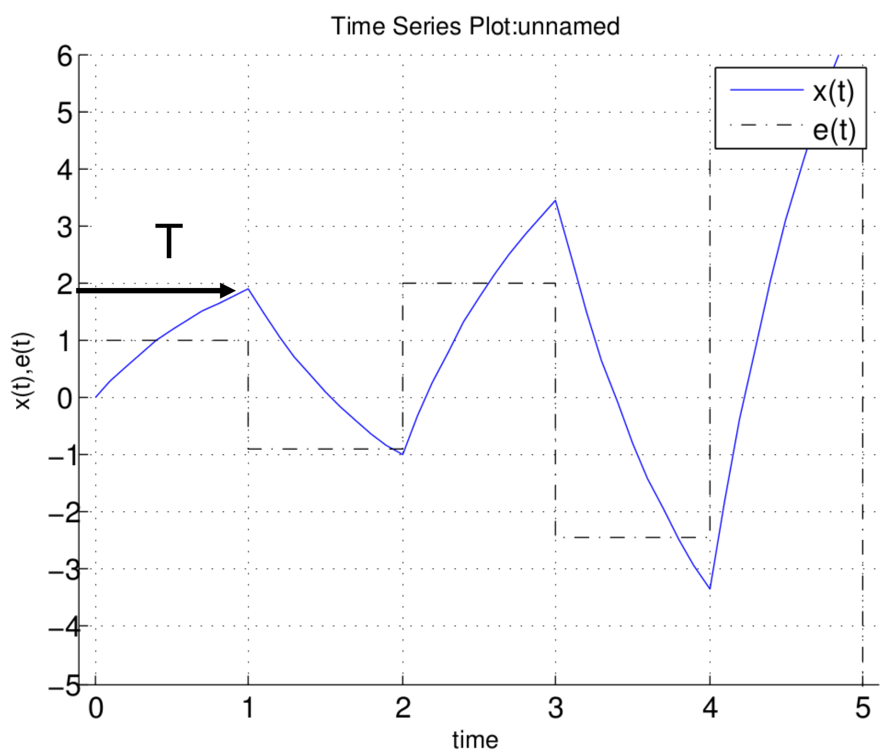
Example control- discrete time

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

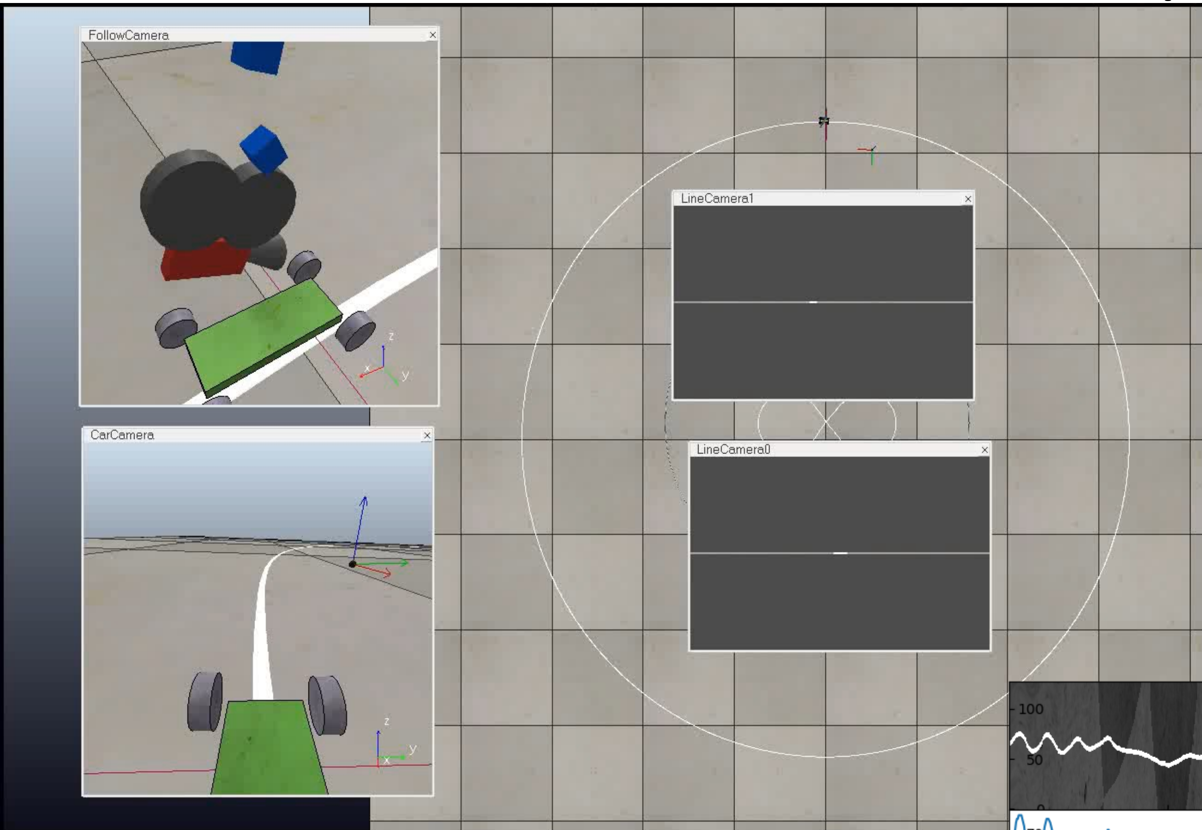
Let x = car velocity
Reference r=1 m/s unit step, k=3
e(t) = r(t) – x(t)
Let control input u[n]=3(r[n]-x[n]) = 3e[n],

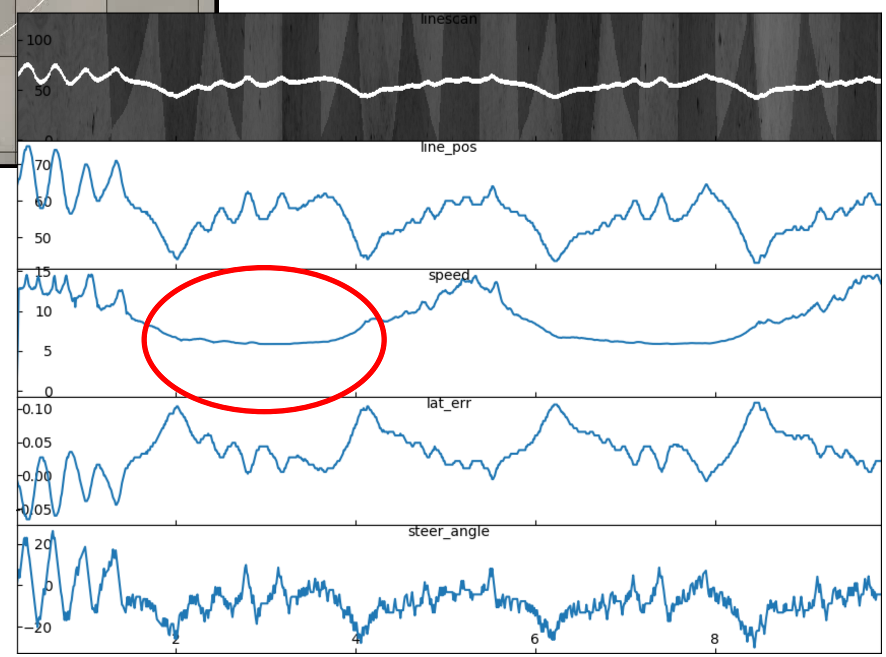
Watch out for delay!
Watch out for excess gain!

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



Circle at 10 m/s





Slow down due to steering sliding