EECS192 Lecture 5
Motor Control and Modelling
Feb. 16, 2021

Notes:
- 2/16 Quiz 1 (10 minutes) line camera timing issues

Topics
- Checkpoint 4: line camera hints and kinks
- Checkpoint 5,6
- HW1 Line finding (Python)
- Quiz 1 (bcourses)
- Velocity control (intro)
- Project Proposal Feedback
- Battery safety
- Motor model

CP4- Open Loop Figure 8 and Line Camera

C4.1 Show car driving an open-loop topological figure 8, at any speed with at least one full CW and at least one full CCW circle. Car should be going as slow as practical.

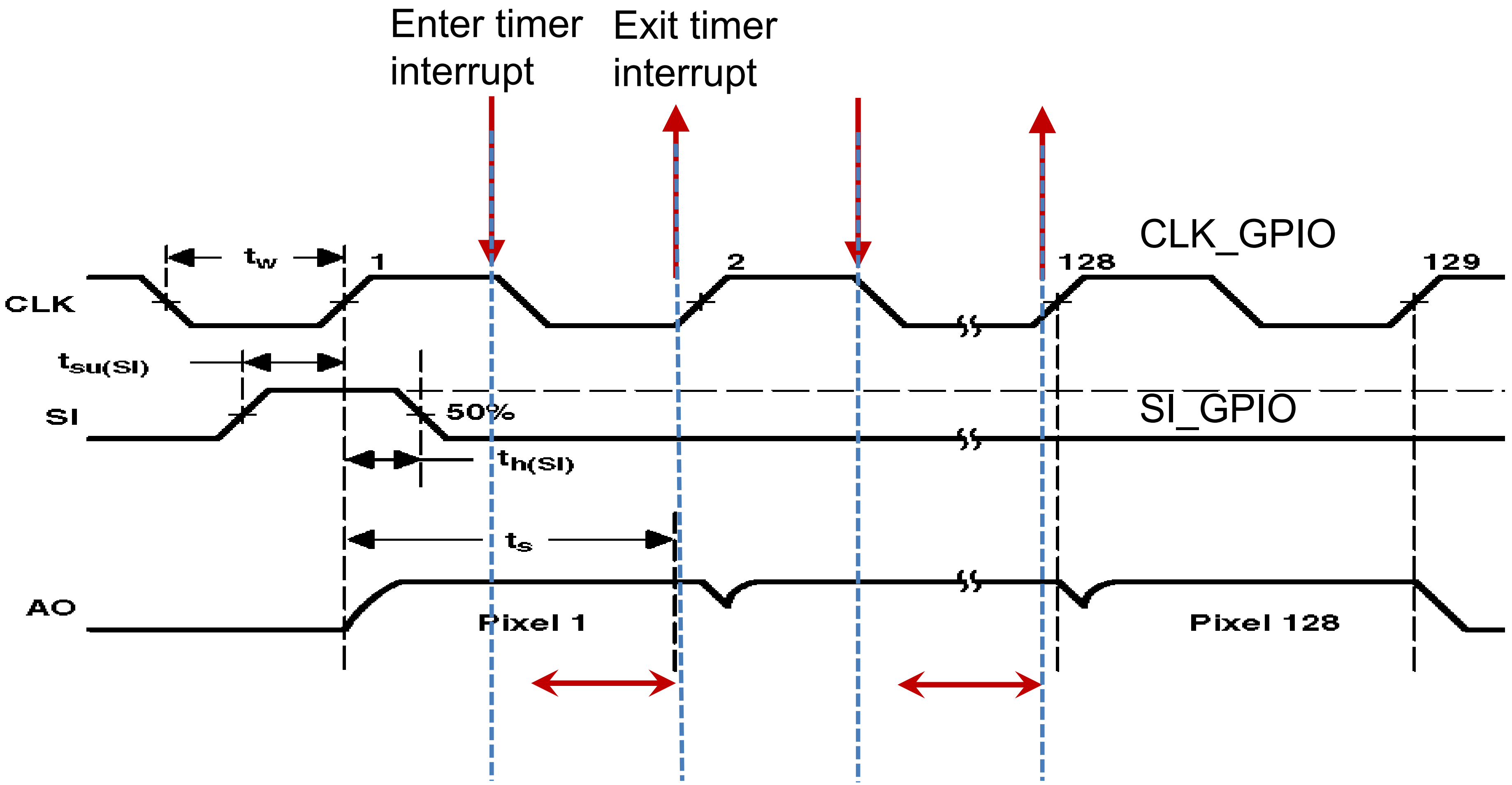
C4.2 Show ability to stop car in middle of figure 8 using ‘‘Emergency stop command’’

C4.3 Show that you are able to read the line camera data and discriminate the line. Possible ways to do this include printing data to the serial console or using the UDP logging framework (preferred). You must be able to explain the output format quickly during the checkpoint.

C4.4 Use a dark surface and a white stripe approximately 2.5 cm wide. When the camera is moved to the left or right of the track, show that the steering servo/car front wheels will respond appropriately. You are not expected to have a nice sensing algorithm (`findmax` is sufficient) or well tuned steering control loop for this checkpoint.

C4.5 All members must fill out the checkpoint survey before the checkoff close. Completion is individually graded.

TSL 1401 line camera



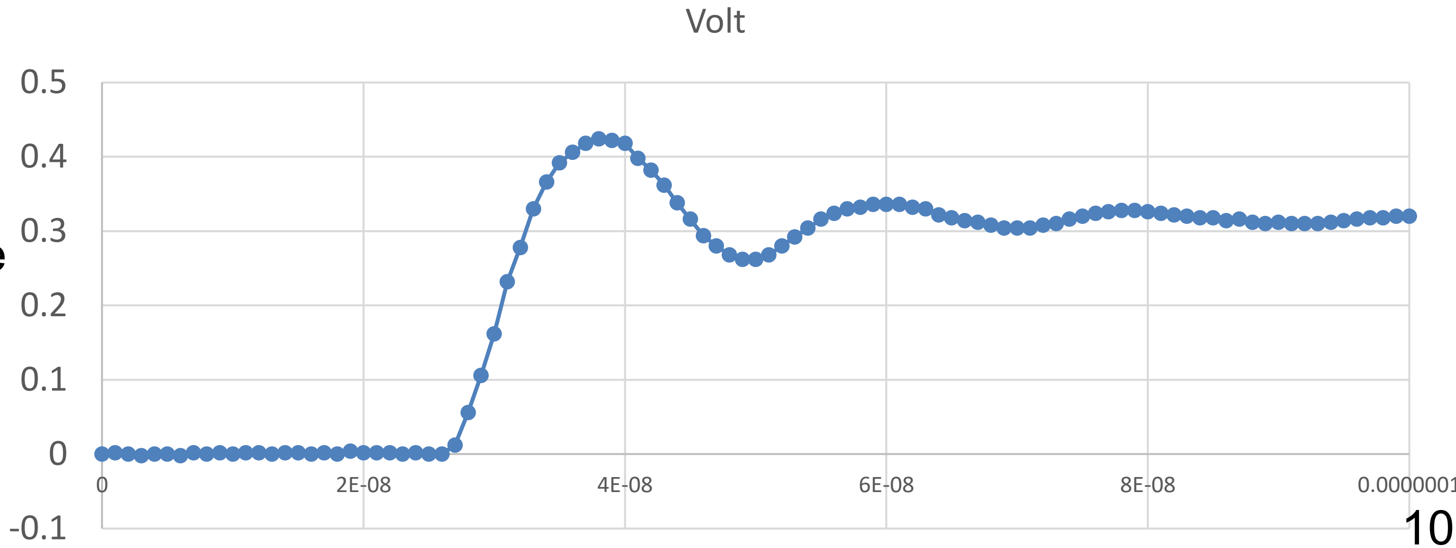
local_adc1_read()

xTaskCreatePinnedToCore(cameraTask, "cameraTask", 2048, NULL, tskIDLE_PRIORITY + 1, &camera_handle, **1**); // core #1

Ringing due to fast rise/fall time on GPIO

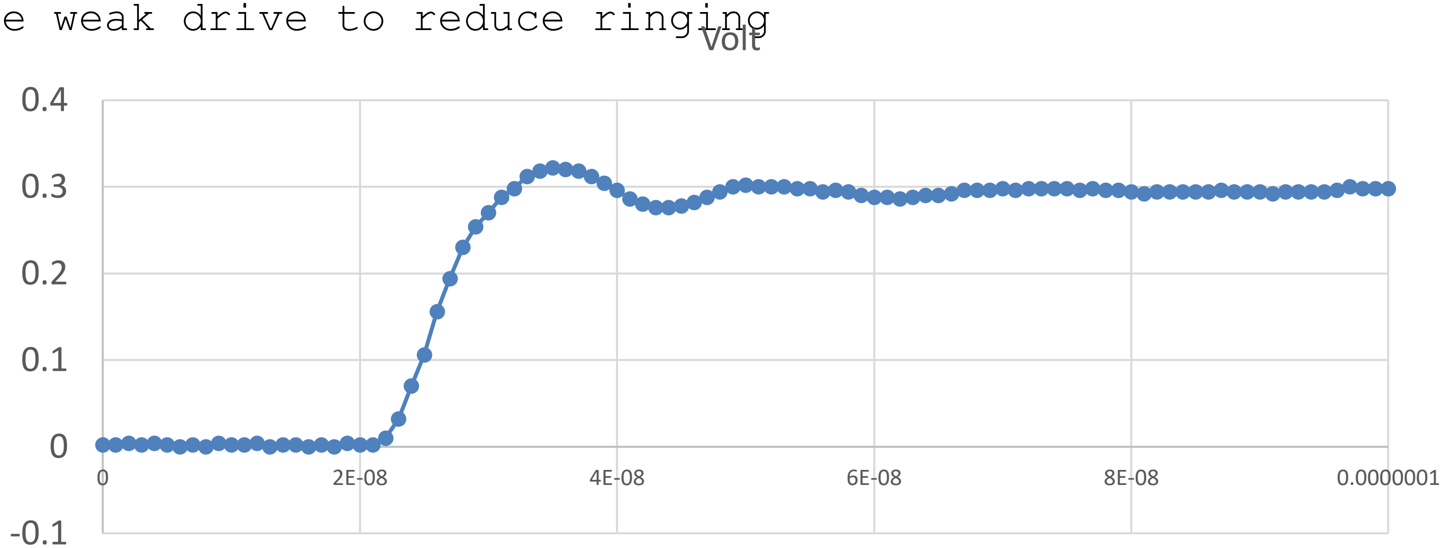
```
gpio_set_drive_capability((gpio_num_t) 18, (gpio_drive_cap_t) 4);
```

X10
probe



100 ns

```
gpio_set_drive_capability((gpio_num_t) 18, (gpio_drive_cap_t) 0);
// make weak drive to reduce ringing
```



ADC Converter (Ch. 30)

adc1_config_width(ADC_WIDTH_BIT_12);
adc1_config_channel_atten(ADC1_CHANNEL_0,ADC_ATTEN_DB_0);
~~int val = adc1_get_raw(ADC1_CHANNEL_0);~~ 45 us!!!! Too slow

ATTEN_DB_0: 0-1 V, ATTEN_DB_11, 0-2.5V

Instead use `int IRAM_ATTR local_adc1_read(int channel)` 🡐 12 us !

(but call adc1_get_raw() once to get ADC properly initialized)

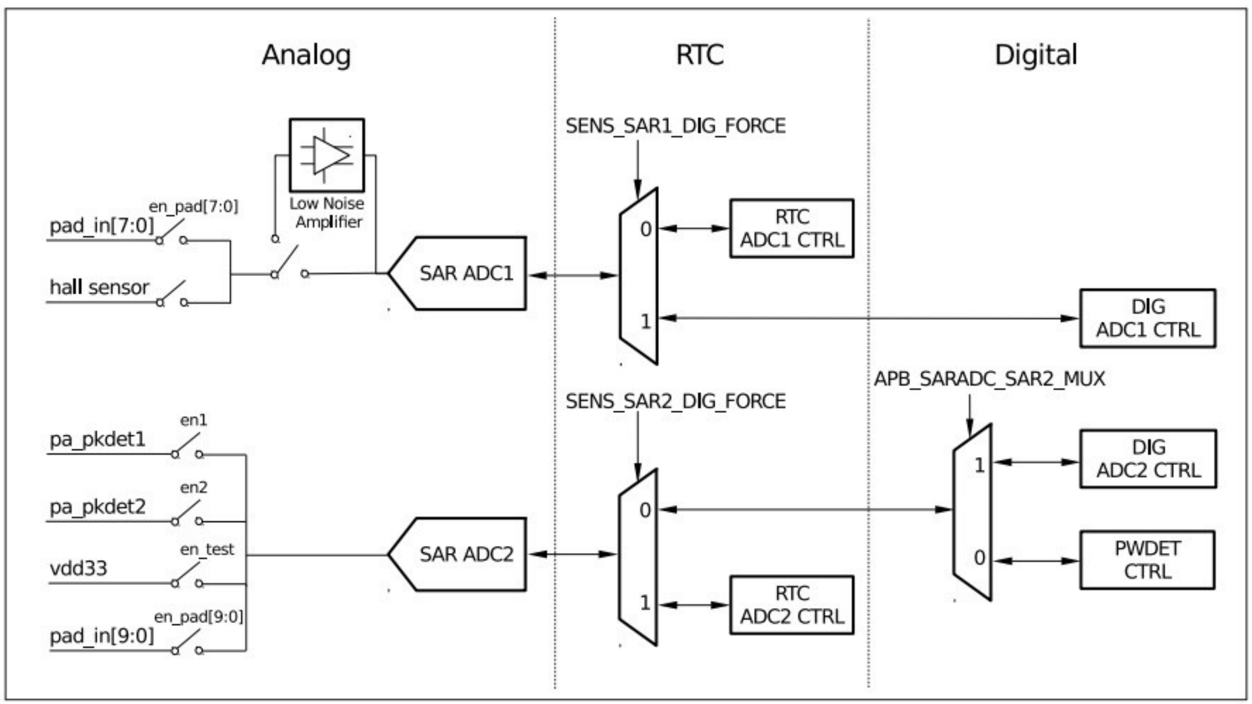


Figure 144: SAR ADC Outline of Function

EECS192 Lecture 5
Motor Control and Modelling
Feb. 16, 2021

Notes:
- 2/16 Quiz 1 (10 minutes) line camera timing issues

Topics
- Checkpoint 4: line camera hints and kinks
- Checkpoint 5,6
- HW1 Line finding (Python)
- Quiz 1 (bcourses)
- Velocity control (intro)
- Project Proposal Feedback
- Battery safety
- Motor model

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 Fig 8 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. The car should do 3 laps in 45 seconds or less.

(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.

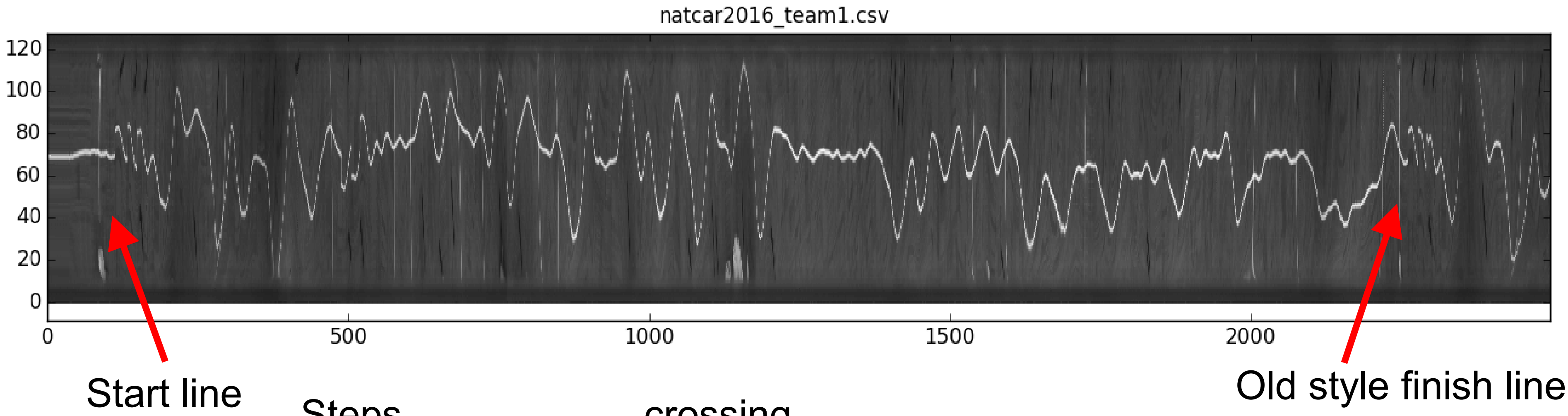
C6.3 All members must fill out the checkpoint survey before the checkoff close. Completion is individually graded.

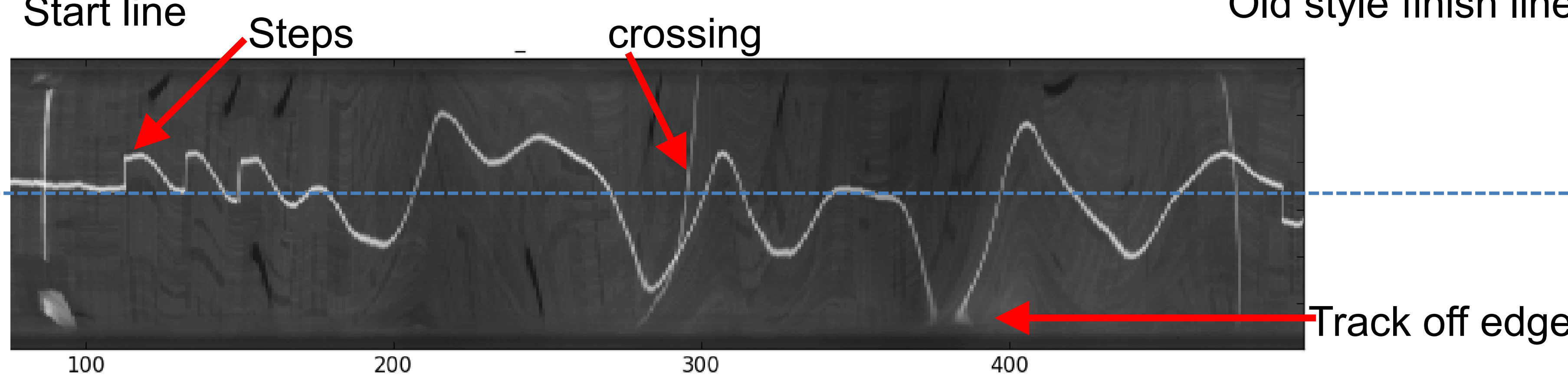
EECS192 Lecture 5
Motor Control and Modelling
Feb. 16, 2021

Notes:
- 2/16 Quiz 1 (10 minutes) line camera timing issues

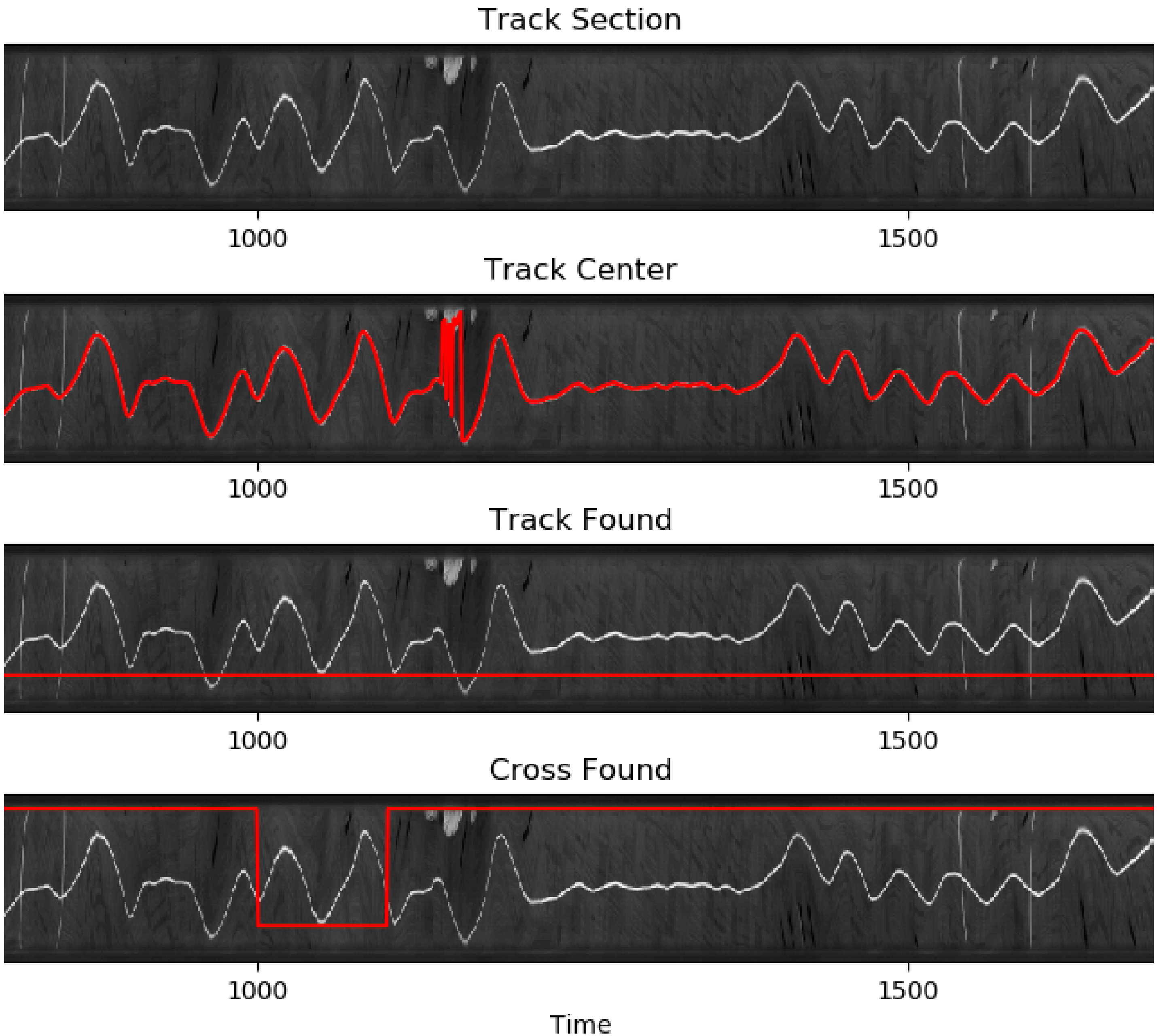
Topics
- Checkpoint 4: line camera hints and kinks
- Checkpoint 5,6
- HW1 Line finding (Python)
- Quiz 1 (bcourses)
- Velocity control (intro)
- Project Proposal Feedback
- Battery safety
- Motor model

Example Line camera data





HW1 Line Sensing



EECS192 Lecture 5
Motor Control and Modelling
Feb. 16, 2021

Notes:
- 2/16 Quiz 1 (10 minutes) line camera timing issues

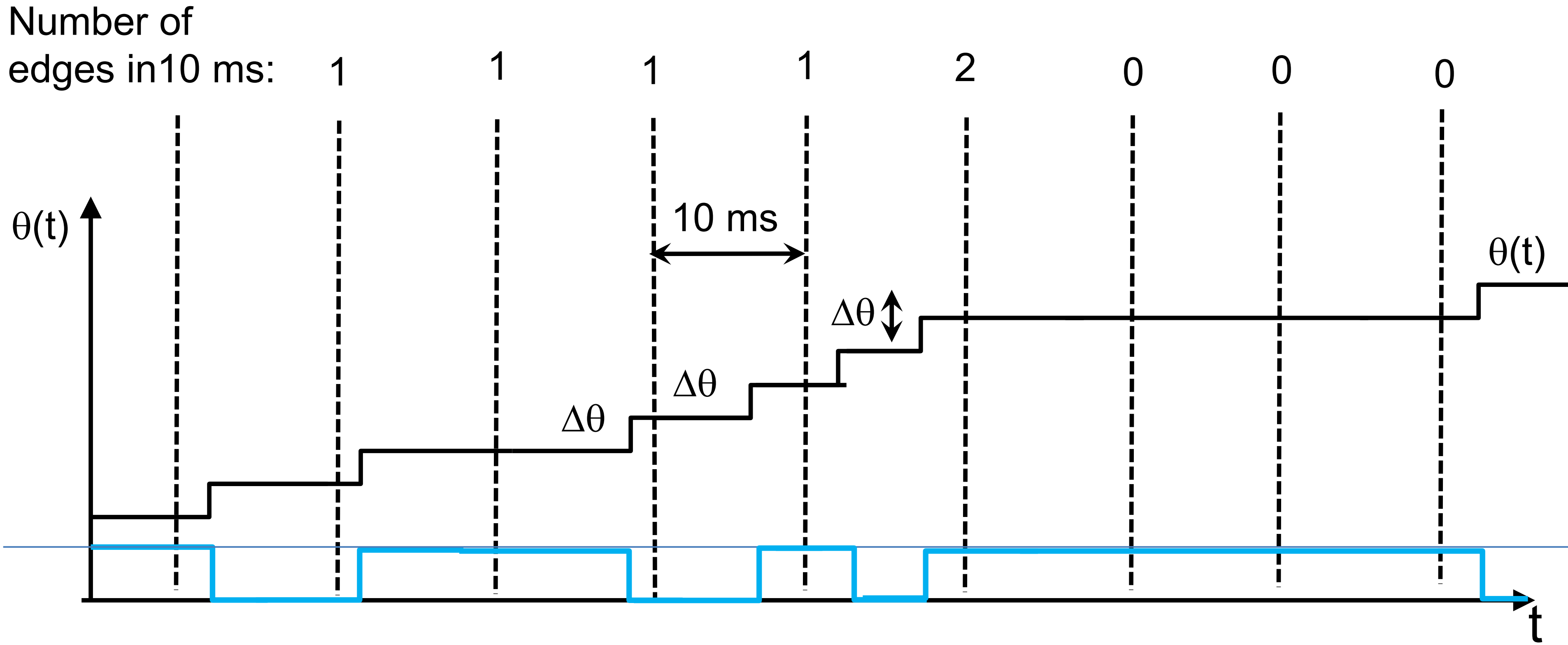
Topics
- Checkpoint 4: line camera hints and kinks
- Checkpoint 5,6
- HW1 Line finding (Python)
- Quiz 1 (bcourses)
- Velocity control (intro)
- Project Proposal Feedback
- Battery safety
- Motor model

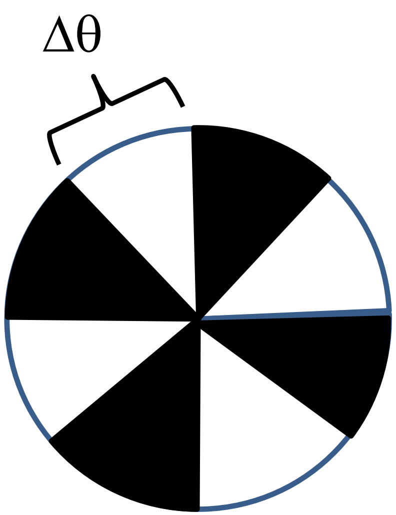
EECS192 Lecture 5
Motor Control and Modelling
Feb. 16, 2021

Notes:
- 2/16 Quiz 1 (10 minutes) line camera timing issues

Topics
- Checkpoint 4: line camera hints and kinks
- Checkpoint 5,6
- HW1 Line finding (Python)
- Quiz 1 (bcourses)
- Velocity control (intro)
- Project Proposal Feedback
- Battery safety
- Motor model

Velocity sensing: method #1 uniform sampling





$$V_{uniform} = \frac{N r_{wheel} \Delta\theta}{10 \text{ ms}}$$

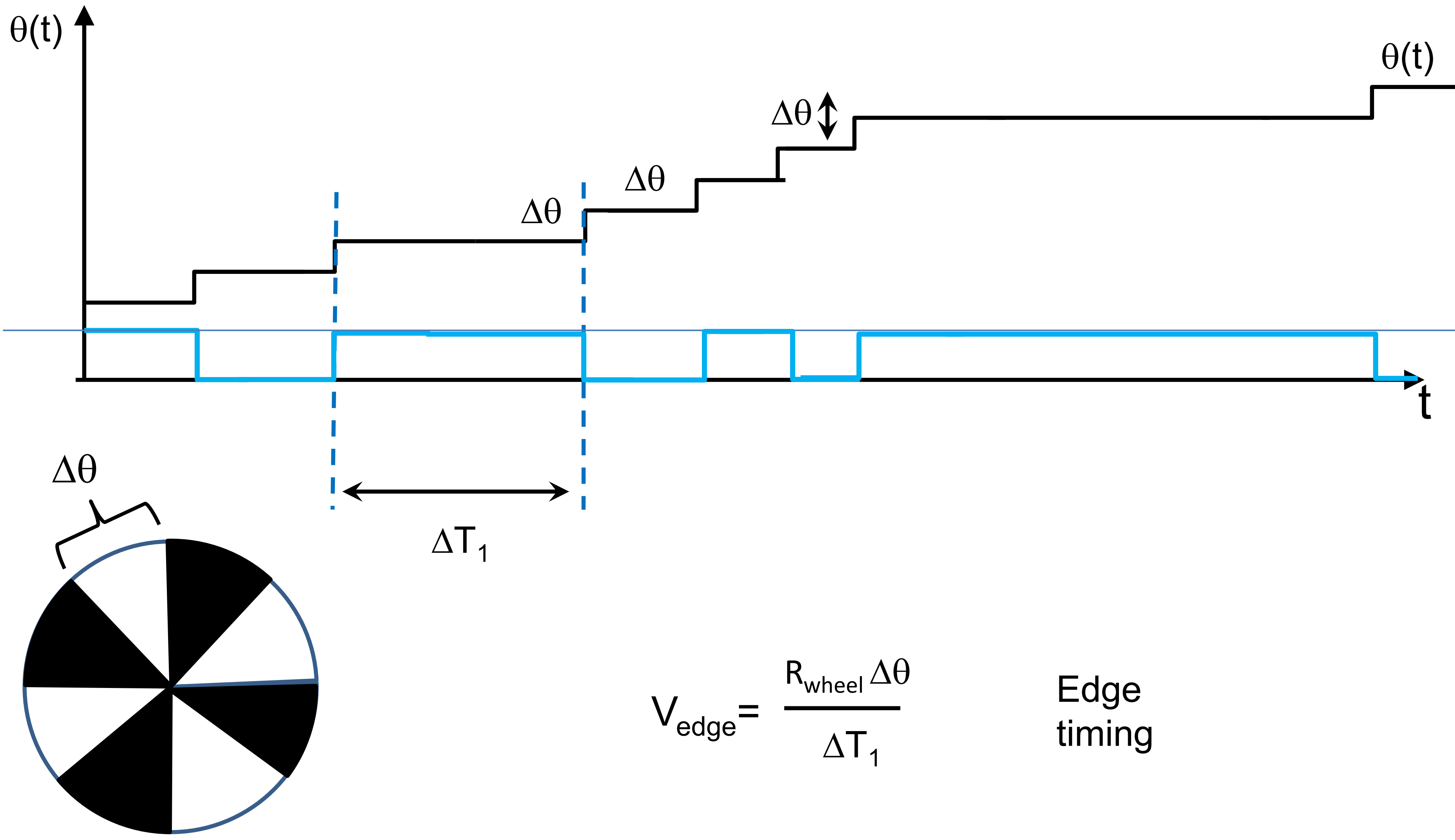
Uniform sampling

What is the problem with these parameters if

$\Delta x = 3 \text{ cm} = r_{wheel}\,\Delta\theta$

?

Velocity sensing: method #2 edge timing



$$V_{edge} = \frac{R_{wheel}\,\Delta\theta}{\Delta T_1}$$

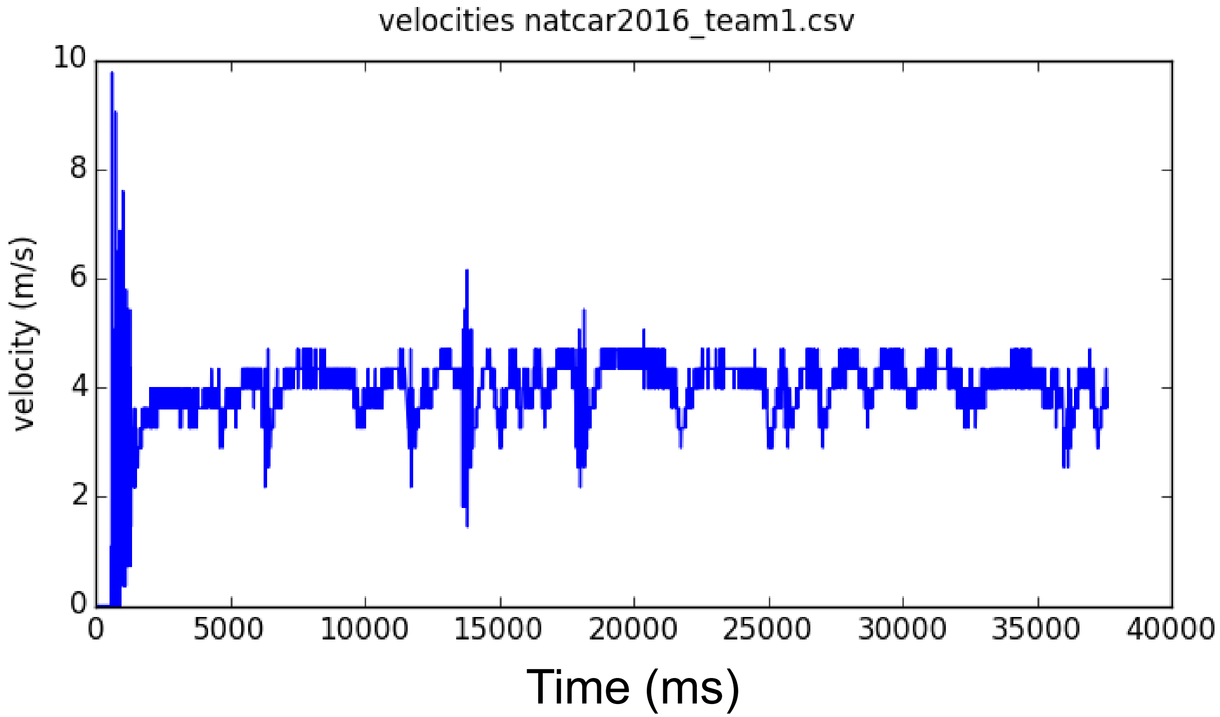
Edge timing

Velocity Sensing

estimating $\Delta x/\Delta T$

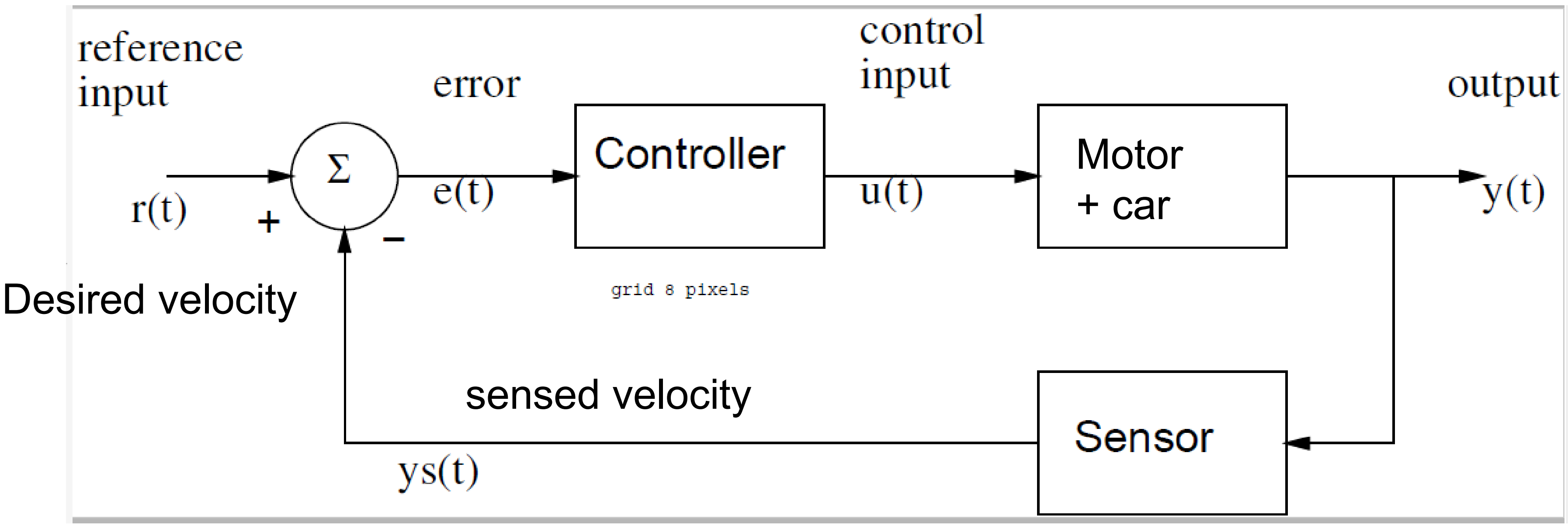
Can use moving average, e.g.

$$V_{ave}[n] = (v[n] + v[n-1] + v[n-2] + v[n-3])/4$$
$$= (x[n]-x[n-1])/T + (x[n-1]-x[n-2])/T + \ldots$$



Note: care about velocity sensing usually at cruise speed (also stopping)

Velocity control overview

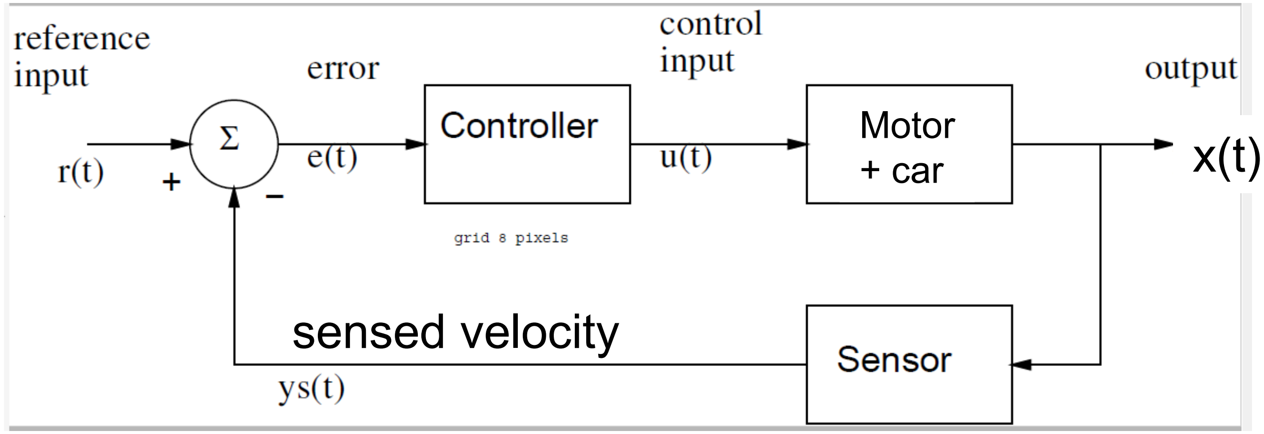


Proportional control:
u = kp*e = kp* (r-y);
Here: r is desired velocity, u is PWM % (1-2 ms)

Proportional + integral control

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

Example control- continuous time



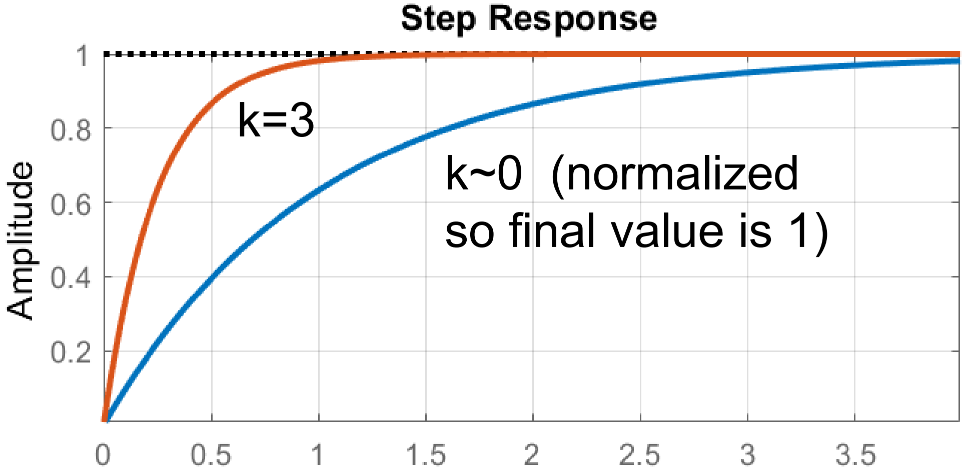
Car model,
x is velocity

$$\dot{x} = -x + u$$

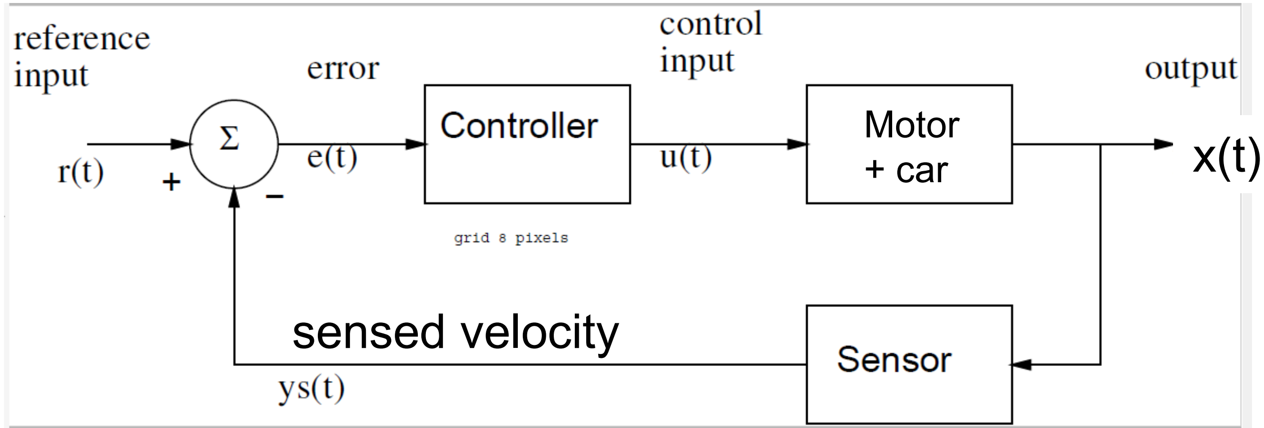
Let control input u=k(r-x), then dx/dt = -(k+1)x + kr

If r(t) is a unit step, then x(t)~ $1-e^{-(k+1)t}$

What is the limit on k?



Example control- discrete time



Car model,
x is velocity

$$\dot{x} = -x + u$$

Car still continuous time!

Let sample time T = 1 second. Let x(0) = 0 (car stopped).

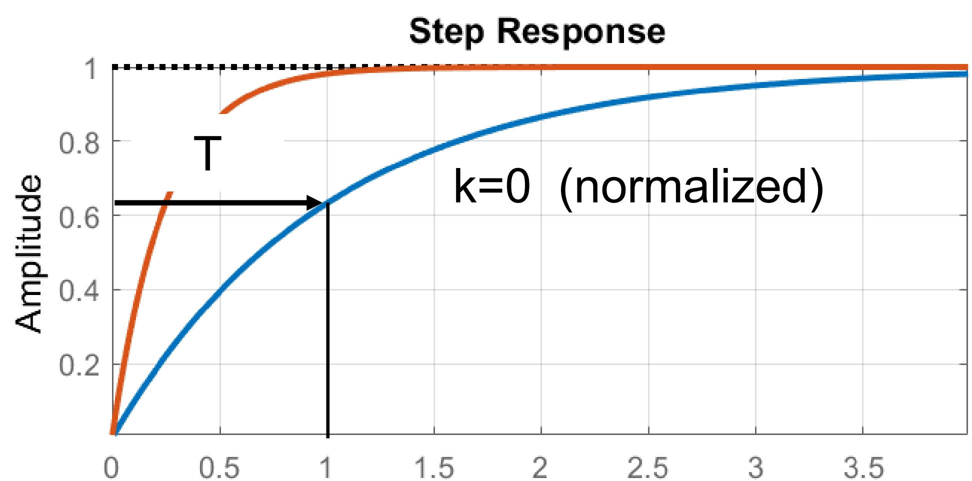
u(0)=u[0] = k(r[0]-x[0])

u[T] = k(r[T]-x[T])

u[2T]=k(r[2T]-x[2T])

etc

u(t) is constant from 0 to T^-



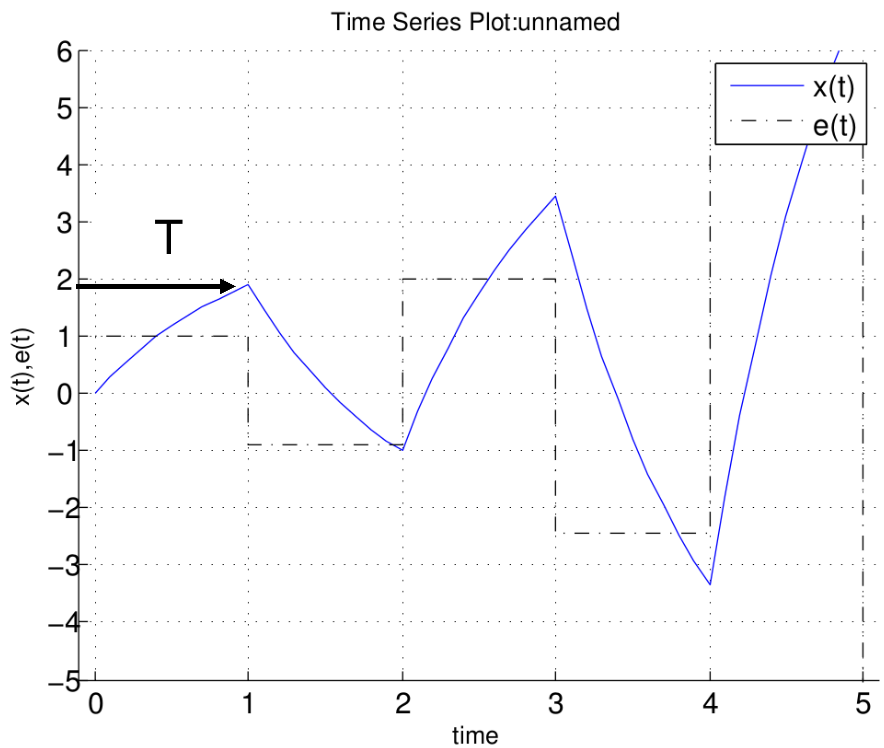
Example control- discrete time

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



EECS192 Lecture 5
Motor Control and Modelling
Feb. 16, 2021

Notes:
- 2/16 Quiz 1 (10 minutes) line camera timing issues

Topics
- Checkpoint 4: line camera hints and kinks
- Checkpoint 5,6
- HW1 Line finding (Python)
- Quiz 1 (bcourses)
- Velocity control (intro)
- Project Proposal Feedback
- Battery safety
- Motor model

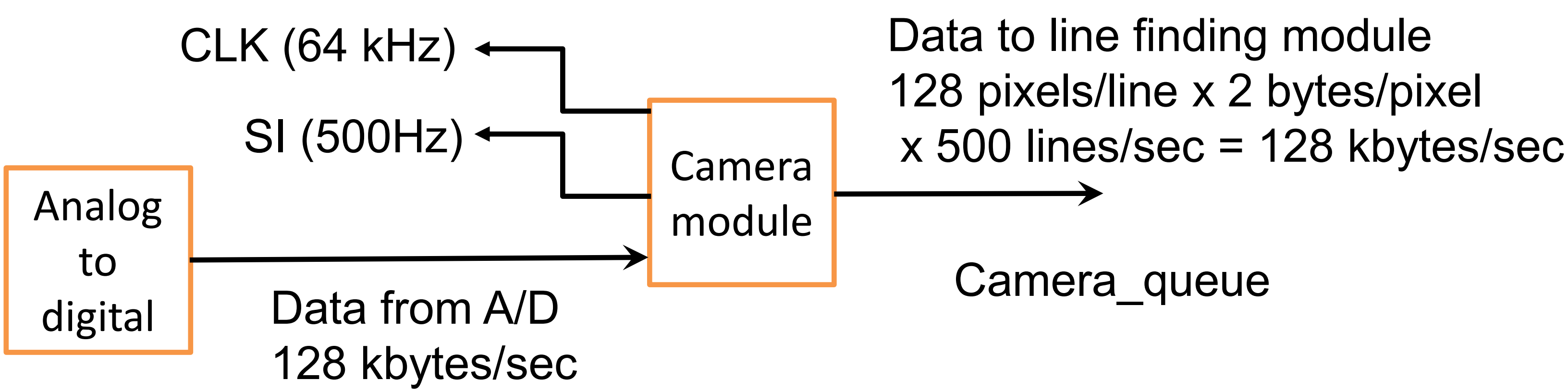
Project proposal feedback

Optional debrief on project proposal- slots on Wed 2/17 pm

Hardware Design:

Right idea, but wheel moves up and down

Software Structure: camera module data rates. Each line is 128 pixels. A/D converter is 12 bits, so 2bytes/pixel. Camera ~ 2 ms

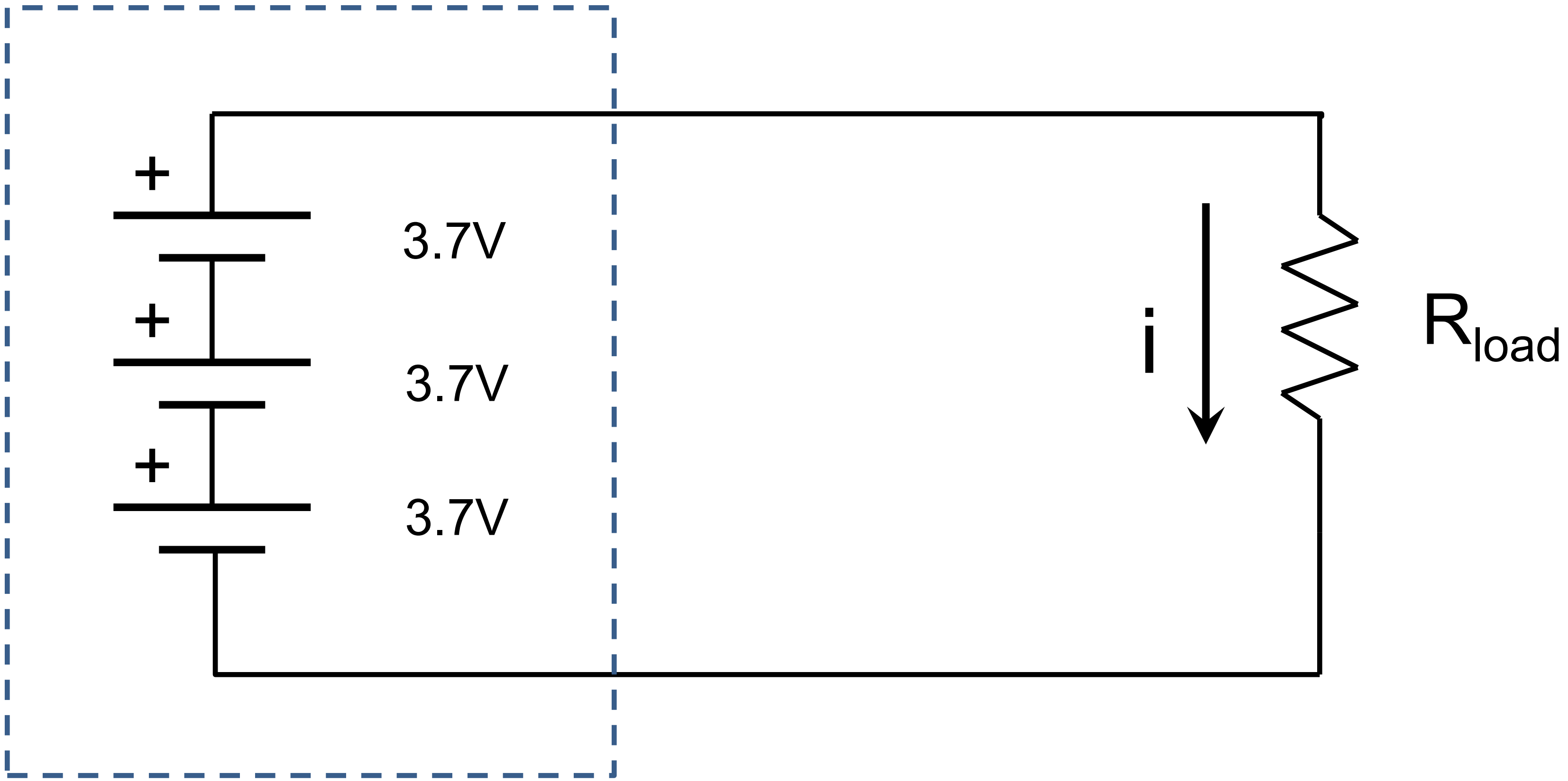


EECS192 Lecture 5
Motor Control and Modelling
Feb. 16, 2021

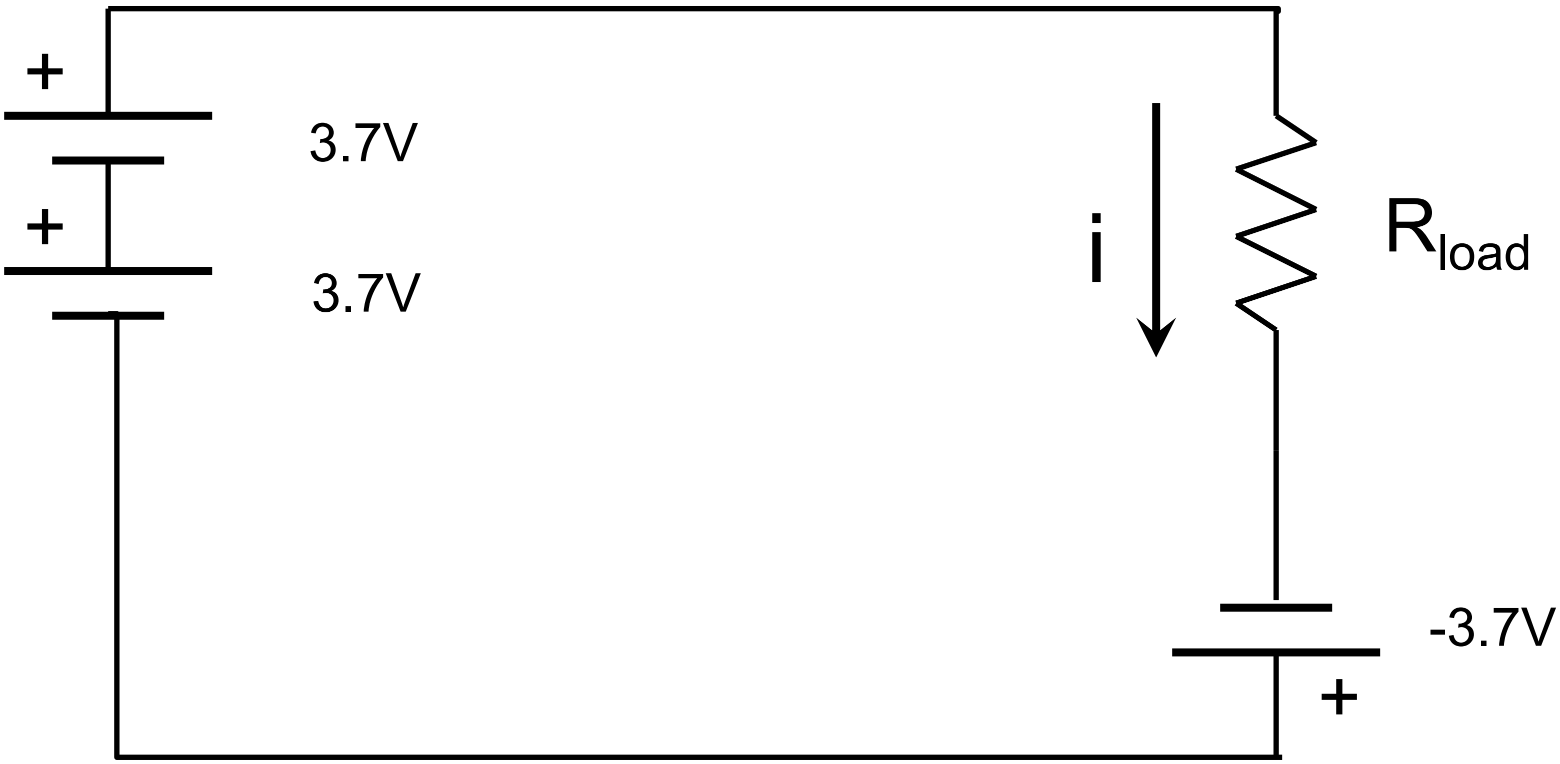
Notes:
- 2/16 Quiz 1 (10 minutes) line camera timing issues

Topics
- Checkpoint 4: line camera hints and kinks
- Checkpoint 5,6
- HW1 Line finding (Python)
- Quiz 1 (bcourses)
- Velocity control (intro)
- Project Proposal Feedback
- Battery safety
- Motor model

Battery Model- 3S



Battery Model- 3S
avoid weakly charged cell

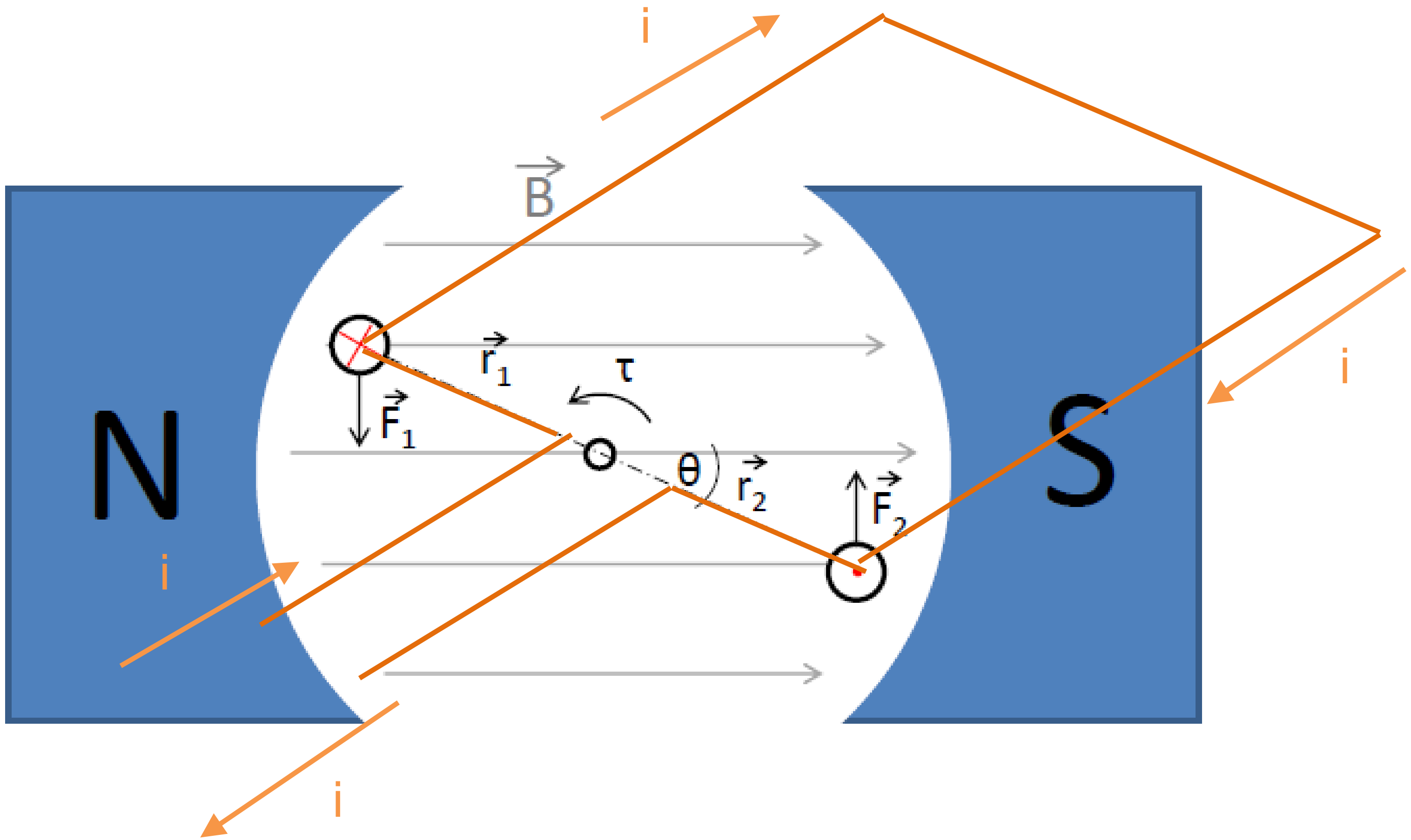


EECS192 Lecture 5
Motor Control and Modelling
Feb. 16, 2021

Notes:
- 2/16 Quiz 1 (10 minutes) line camera timing issues

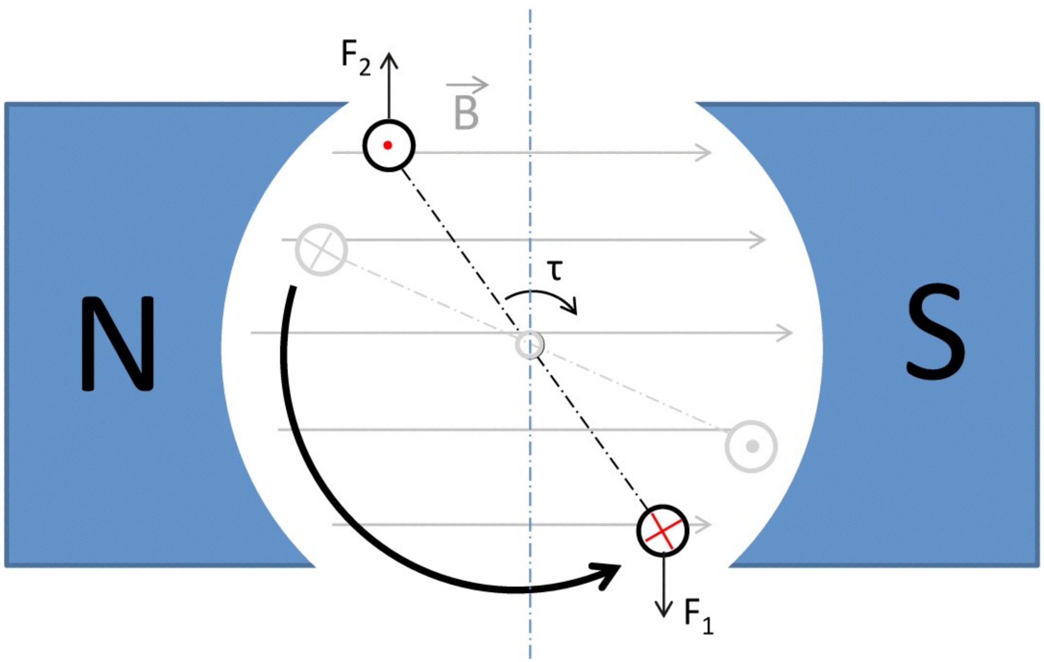
Topics
- Checkpoint 4: line camera hints and kinks
- Checkpoint 5,6
- HW1 Line finding (Python)
- Quiz 1 (bcourses)
- Velocity control (intro)
- Project Proposal Feedback
- Battery safety
- Motor model

DC Motor Physical Model

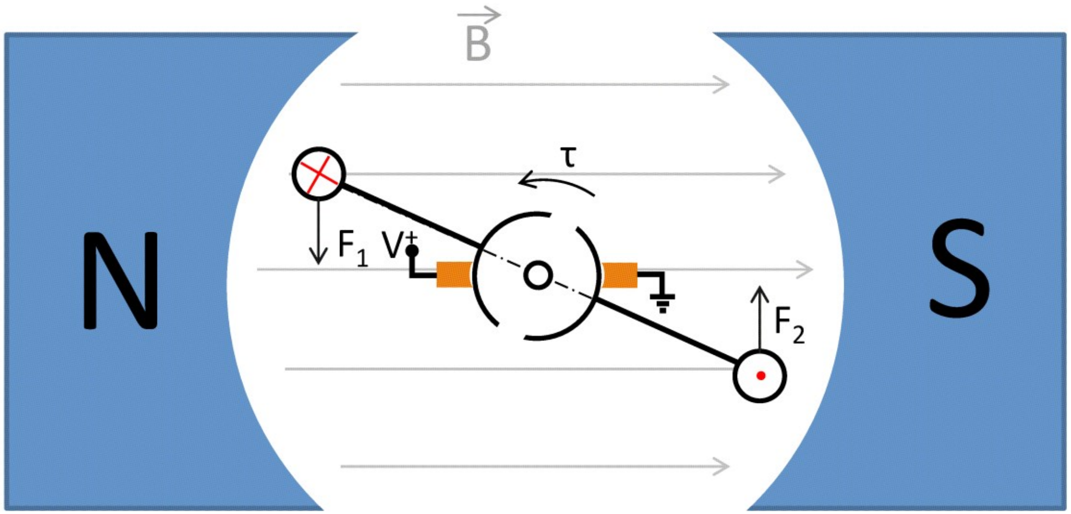


$$\vec{F} = i\vec{l} \times \vec{B}$$

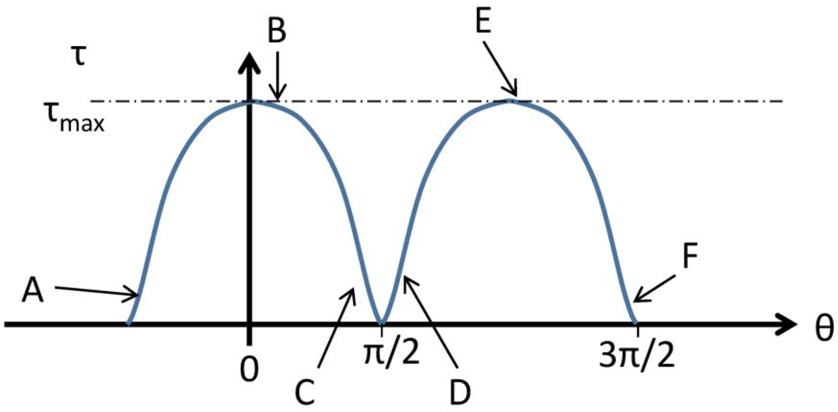
$$\tau = \vec{r}_1 \times \vec{F}_1 + \vec{r}_2 \times \vec{F}_2$$



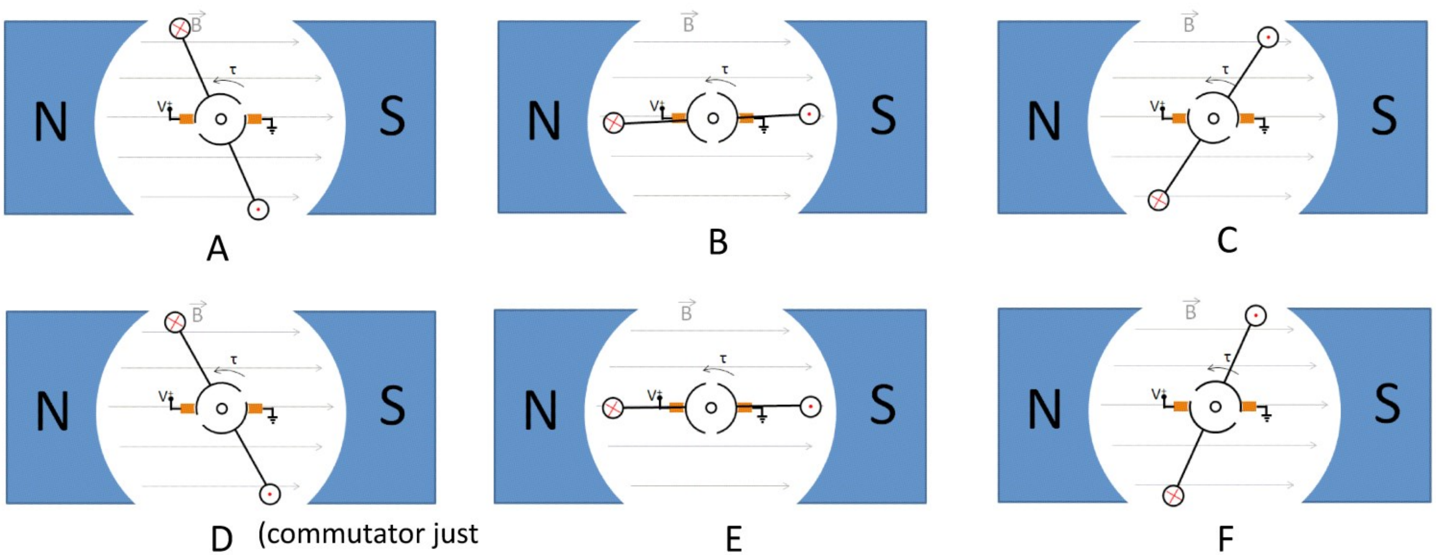
Rotating past the θ = π/2 line (blue) is a problem if current in the loop is in the same direction, because it would cause the reversal of the torques.



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

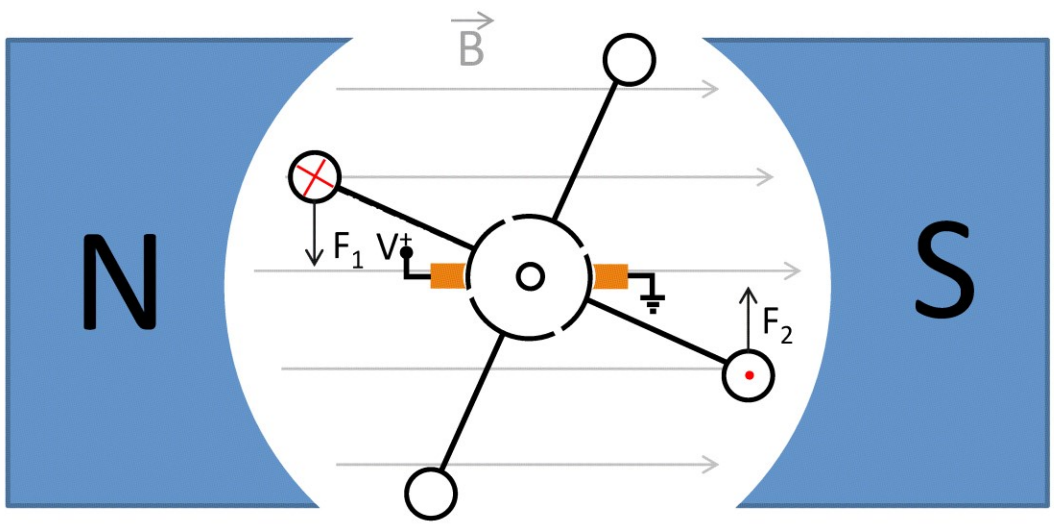


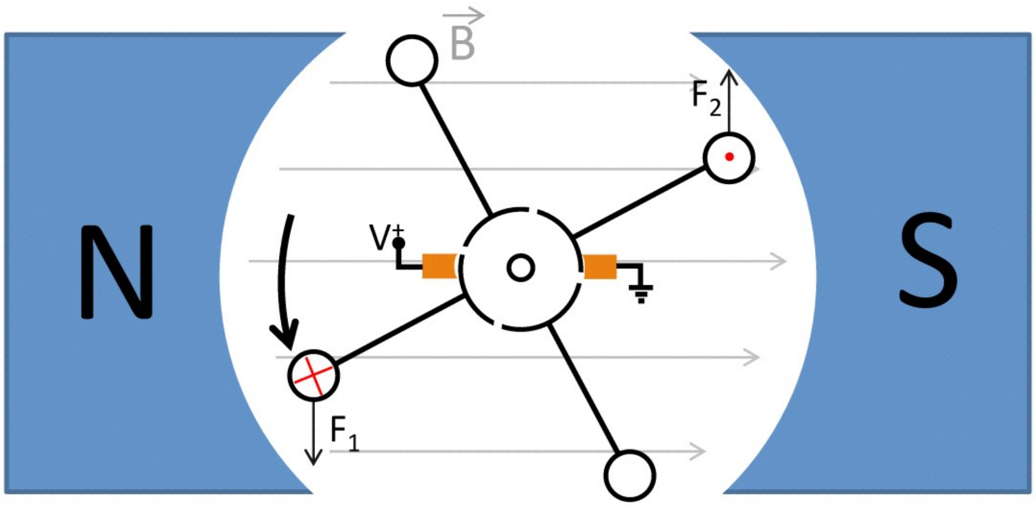
It works, but big torque ripple with only two segment commutator.



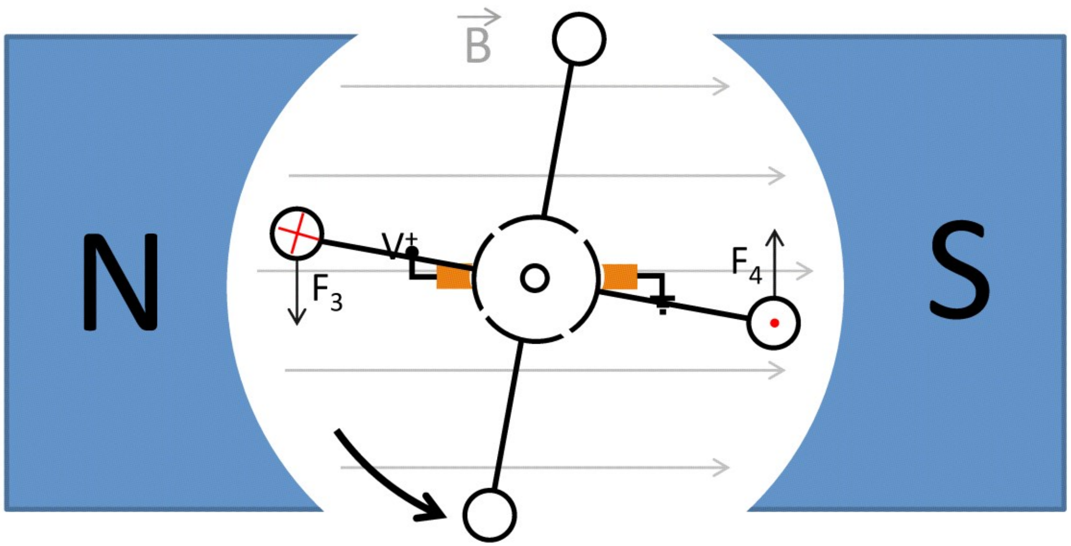
(commutator just reversed the current directions)

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

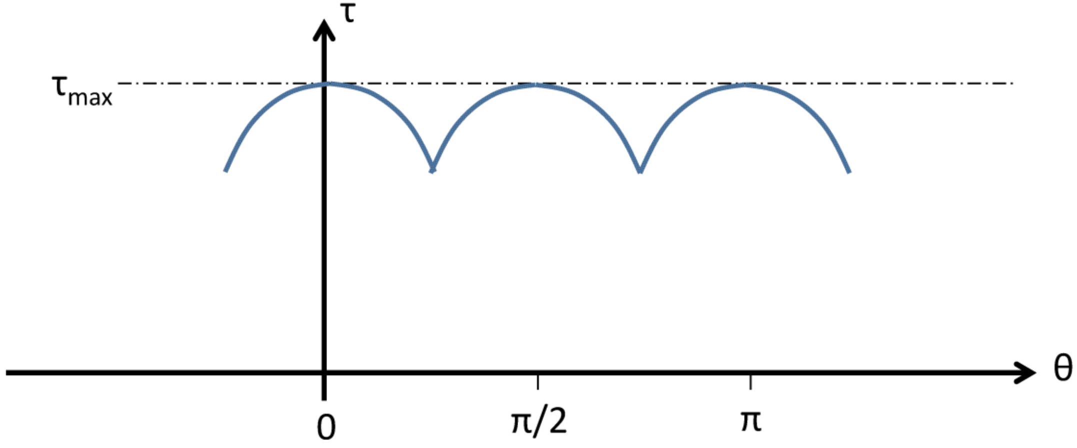




B
F2
N
V+
S
F1



$\vec{B}$
N
S
V+
F_3
F_4



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

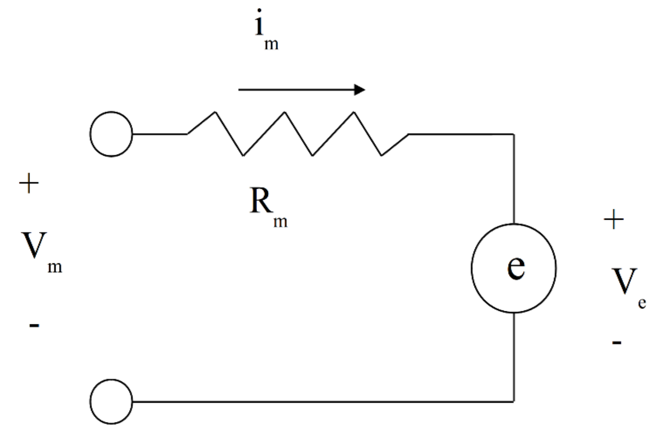
EECS192 Lecture 5- Summary
Motor Control and Modelling
Feb. 16, 2021

Notes:
- 2/16 Quiz 1 (10 minutes) line camera timing issues

Topics
- Checkpoint 4: line camera hints and kinks
- Checkpoint 5,6
- HW1 Line finding (Python)
- Quiz 1 (bcourses)
- Velocity control (intro)
- Project Proposal Feedback
- Battery safety
- Motor model

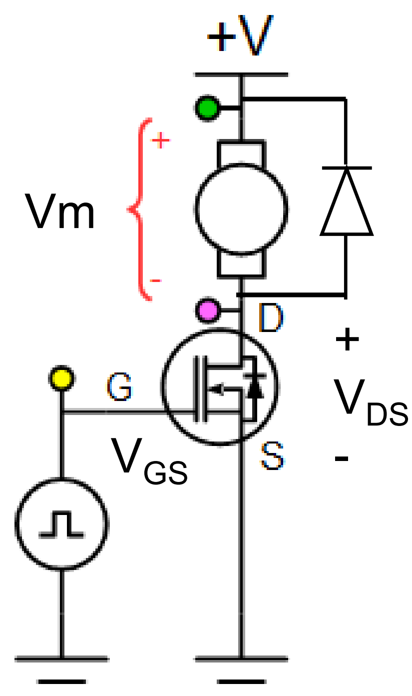
Extra Slides

Motor Electrical Model (neglect inductor)



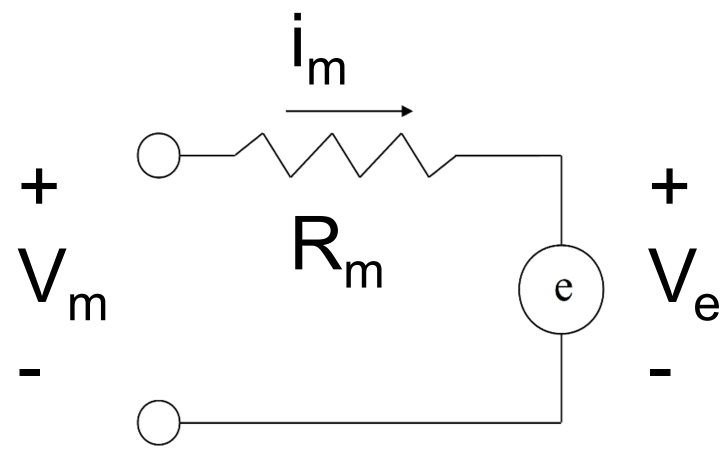
Motor Electrical Model
Back EMF
Motor electromechanical behavior

Continued on board
Also- see motor worksheet……



Note: missing e-stop!

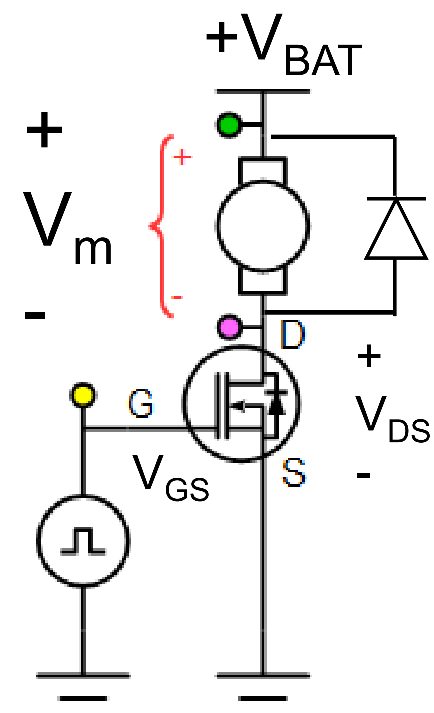
Motor Electrical Model



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?

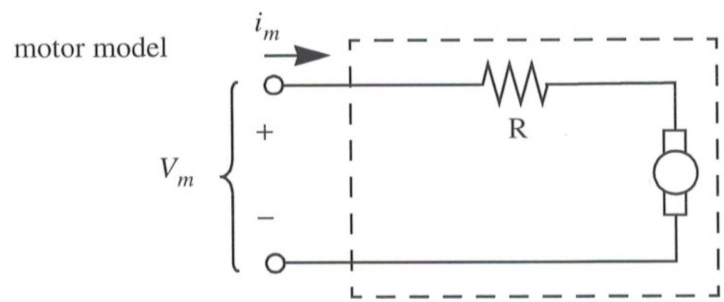
Conclusion:

$<i_m>=?$

Motor model

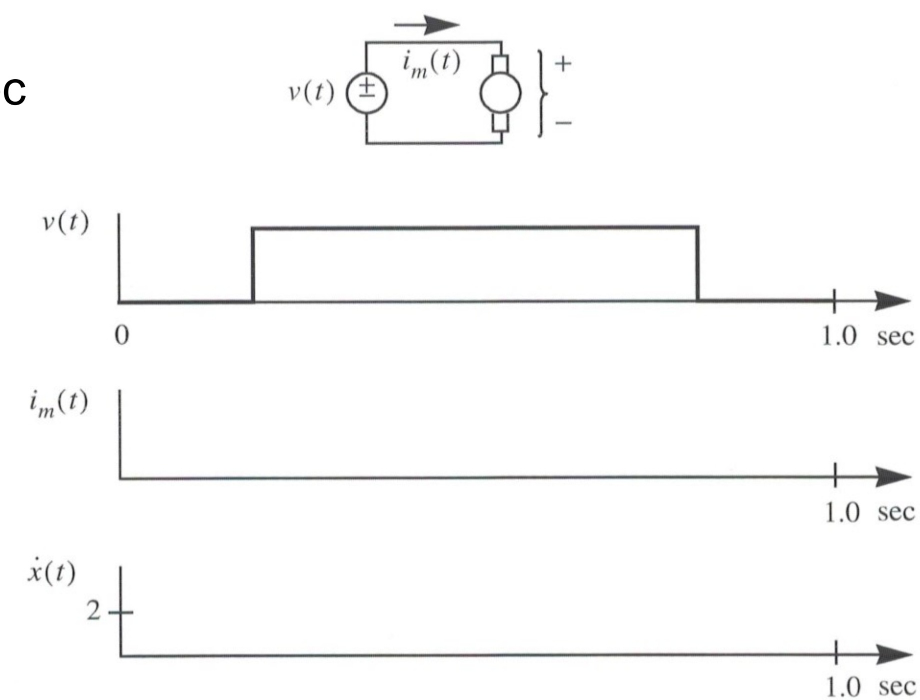
For this problem, consider a DC permanent magnet motor (as used in your car). The car is on a carpet and moves in a straight line with no slip between the wheels and the carpet. The car is initially moving at a speed of 2 meters per second.

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



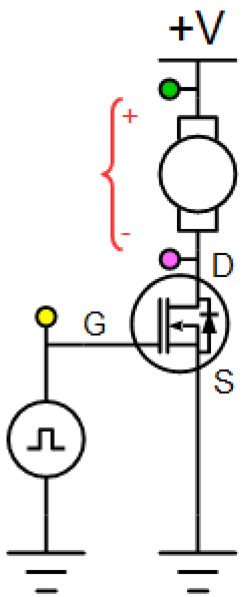
[4 pts.] **a)** Consider the motor driven from a voltage source with voltage $v(t)$, as shown. Sketch car velocity $\dot{x}(t)$ and motor terminal current for the time indicated.

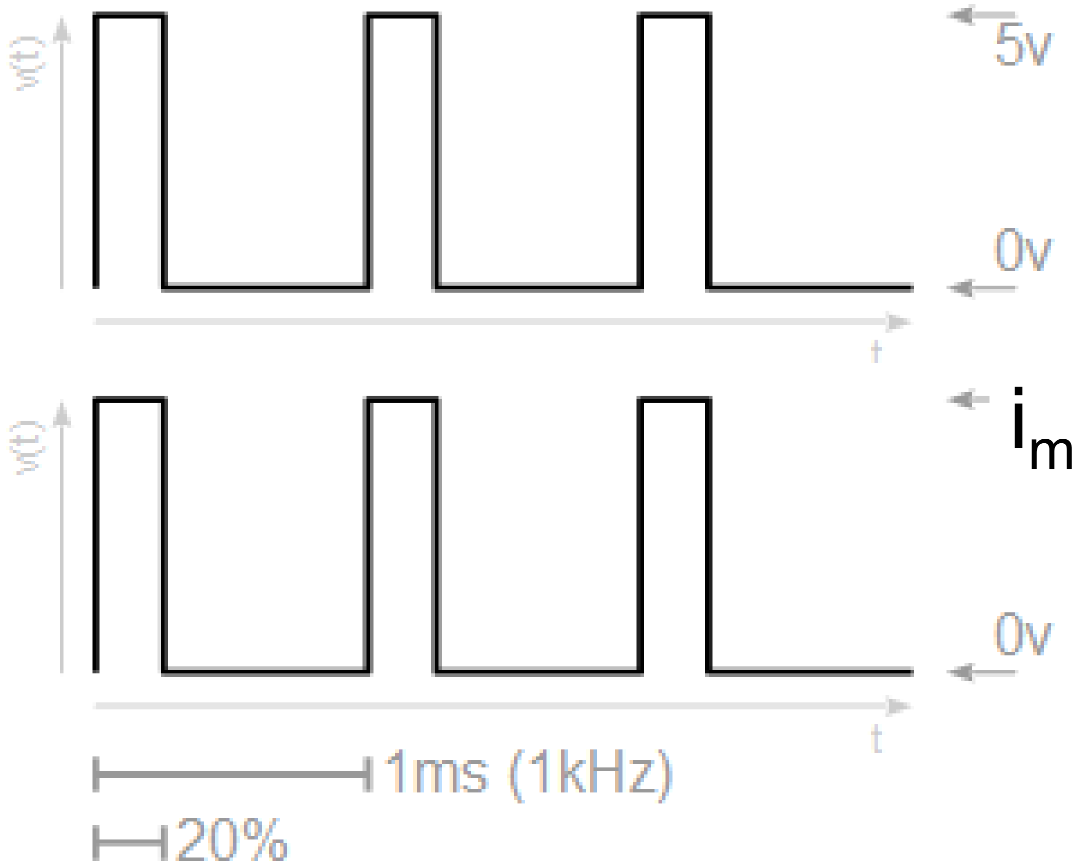
Let peak speed = 5 m/sec
Accel = 5 m/s^2
k_e = 1 v/(m/sec)
On board



(for answer
see sp99 final solution)

PWM for Main Motor control

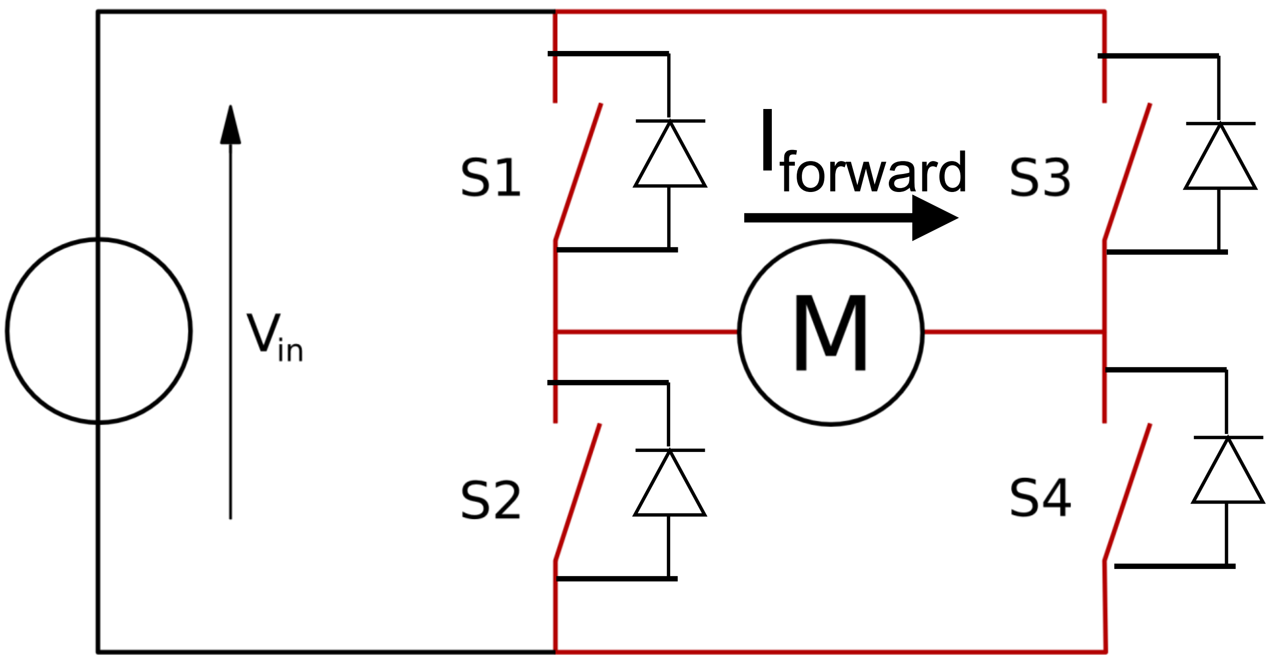




$<i_m> = (T/T_o)\, i_{max}$

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	

What does high side transistor do?

Latchup

Make sure Huzzah32 powered first,
before ESC,
before position encoder

1.4 Voltage and current operating ratings

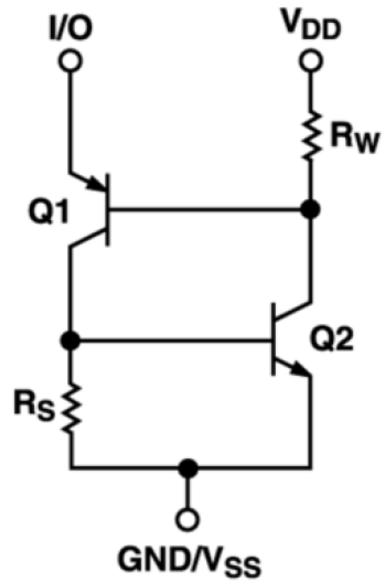
LATCHUP!

Table 4. Voltage and current operating ratings

Symbol	Description	Min.	Max.	Unit
V_{DD}	Digital supply voltage	–0.3	3.8	V
I_{DD}	Digital supply current		120	mA
V_{IO}	IO pin input voltage	–0.3	V_{DD} + 0.3	V

Caution: input voltage from sensor may be greater than 0.3V when CPU is off
VDD = 0!

Latchup phenomena:
make sure Vin always less than Vdd



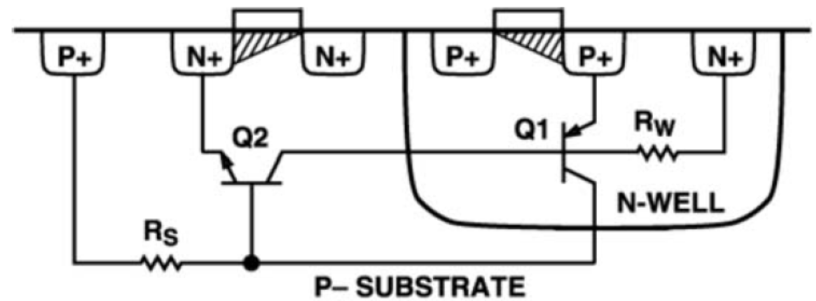
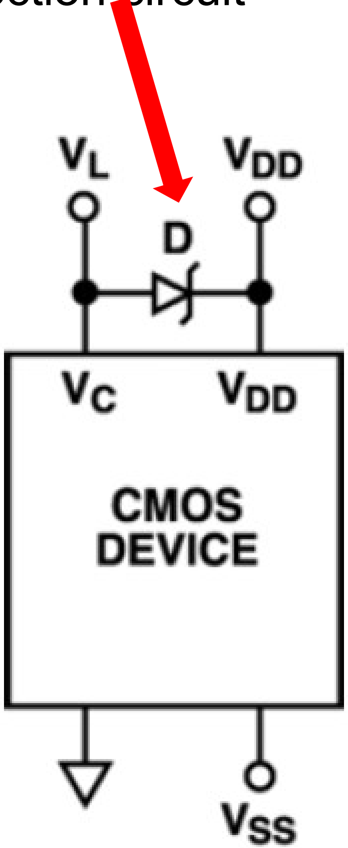


Figure 2. Cross-section of PMOS and NMOS devices, showing parasitic transistors Q1 and Q2.

Protection circuit



Schottky diode

Protection circuit