EECS192 Lecture 5
Feb. 19, 2020

Announcements

1. 3/3 Quiz 3: TSL1401 line camera
2. CalDay Sat. April 18 10 am @ UCB
3. Be safe with LiPo
4. Be safe with mounting

Topics

- Upcoming checkpoints
- Q1 Solution
- Project Proposal Feedback
- Q2
- Velocity Sensing Recap
- Velocity Control (intro)
- Line Sensor TSL1401
- HW1 Track Detection
- Steering control (intro to intro)

Upcoming Checkpoints

2/21 C4: easy, work ahead!
C4.2: Line camera image capture with exposure control.
C4.4.4 Line camera calibration: measure track lateral displacement in mm
HW 1 line detection (due 3/3)

2/28 C5: may be harder, mounting, prototyping velocity sensor, writing control code
C5.3: BBBL, motor driver, velocity sensor mounted to car
C5.4: Basic track detection and wheels turn toward track (benchtop)
C5.5: basic velocity sensor, estimation and benchtop control: 3 m/s.

3/6 C6.3: The vehicle must complete the figure-8 course completely autonomously in under 3 minutes.
C6.3.4: running with velocity control

Topics

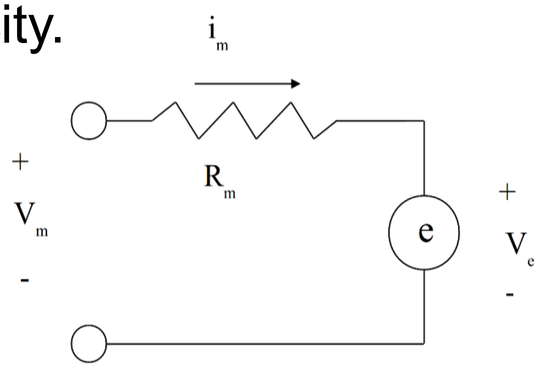
- Upcoming checkpoints
- Q1 Solution
- Project Proposal Feedback
- Q2
- Velocity Sensing Recap
- Velocity Control (intro)
- Line Sensor TSL1401
- HW1 Track Detection
- Steering control (intro to intro)

Q1 Solution

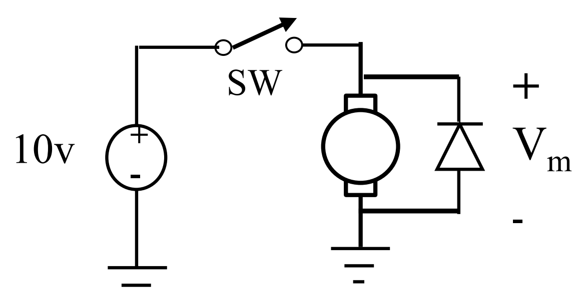
Consider a DC permanent magnet motor (as used in your car).
The car is initially at rest. The motor is connected as shown below.
Neglect battery and switch resistance. Neglect motor inductance. Assume diode is ideal.
Assume motor resistance = 0.2 ohm, and that the car accelerates to 4 m/s in 2 seconds.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

On board

Topics

- Upcoming checkpoints
- Q1 Solution
- Project Proposal Feedback
- Q2
- Velocity Sensing Recap
- Velocity Control (intro)
- Line Sensor TSL1401
- HW1 Track Detection
- Steering control (intro to intro)

Project proposal feedback: Motor Driver

Schematic

- Estop: what to switch?
- G_EN = 5V not 3.3V.
- Snubbing capacitors and diode
- Drive/brake/enable/dir- shoot through protection
- Need accel/coast/brake to avoid burning out motor/transistors
- Connectors- battery, motor, IO connections+ground
- Make sure capacitors on MC33883 are connected (needed for charge pump) and appropriate Vcc.
- Watch out for RC time constant on gate.
  Ciss Input Capacitance =8970pF.

Circuit Layout

- Mounting holes
- Big wires, short distances
- QFN vs SOIC package, SMD vs through hole
- Heat sinks- horizontal transistor is more robust
- Estop switch
- Signal connectors- include ground
- Power connectors

Project proposal feedback:

2.2 Mechanical Mount

- Encoder location: find easily accessible location, not in suspension
- Materials: 3D printed parts, and acrylic: heavy and brittle. Consider Styrofoam or thin-wall tube

3.1 IO Lines

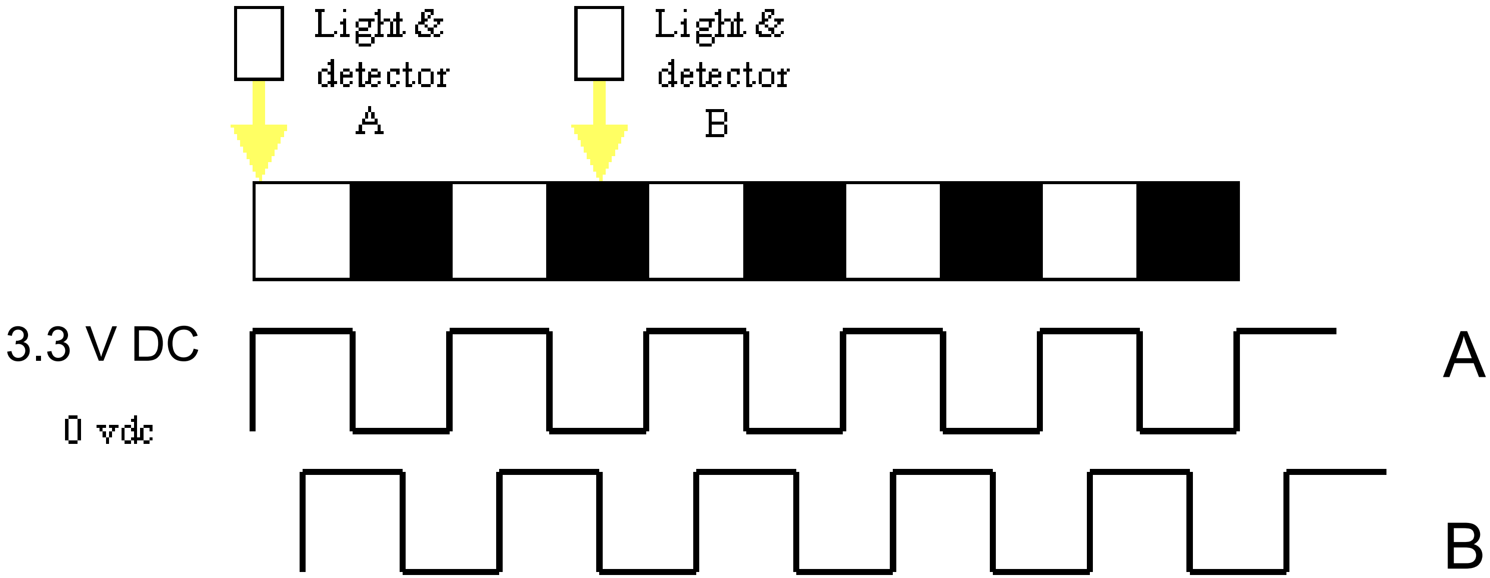
- PWM? GPIO line won’t be real time. Need PWM hardware, or PRU. PWM0 is available on ``GPS’’ connector through librobotcontrol.

- Encoder? GPIO line is not setup to count edges and may miss fast samples. QEP avail if using quadrature (maybe hack with delay for using single detector). Could also use PRU and a GPIO input line. (PRU0 already setup for Chan 4 quad encoding).
  - Maybe use timer 7 to latch event of rising edge of signal (ch 20)
  - Maybe use eCAP to catch rising edge (Ch 15)

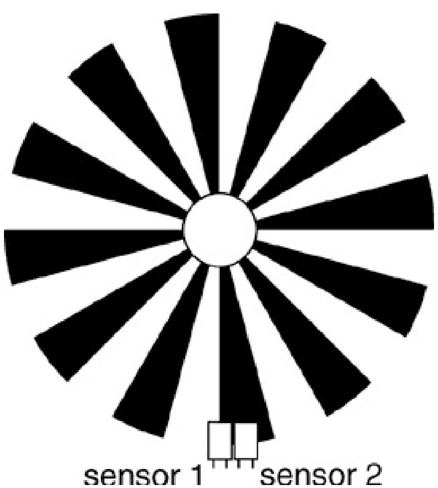
Topics

- Upcoming checkpoints
- Q1 Solution
- Project Proposal Feedback
- Q2
- Velocity Sensing Recap
- Velocity Control (intro)
- Line Sensor TSL1401
- HW1 Track Detection
- Steering control (intro to intro)

Quadrature Encoder



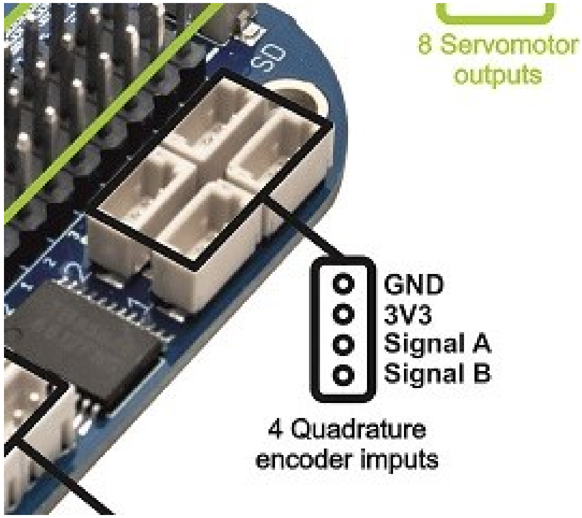
https://www.sinotech.com/wp-content/uploads/quadrature-encoder.gif



Fab suggestion: aluminum foil covered with black paper with slots. 4 slots probably enough. Note: sensors can be placed where convenient- don’t need to look at same slot.

http://codeforfree.weebly.com/uploads/1/0/1/6/10160088/5858962_orig.png

Beagle Bone Blue Quad Encoder




```
int rc_encoder_read  (int ch)
```

Returns
The current position (signed 32-bit integer)
or -1 and prints an error message is there is a problem.
Ch 1-3 are available
Examples:
rc_test_encoders.c.

Sharp GPS260

Fig.9 Test Circuit for Response Time

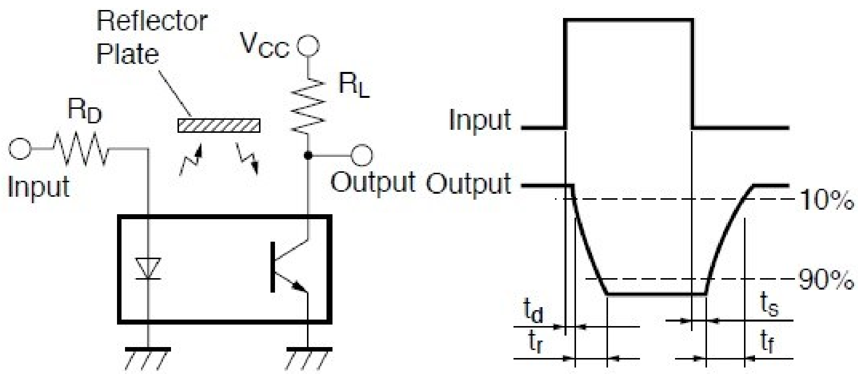
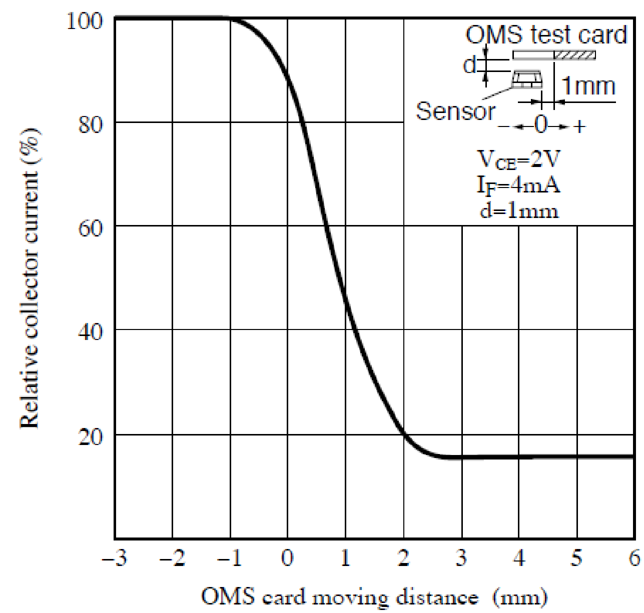


Fig.13 Detecting Position Characteristics (2)

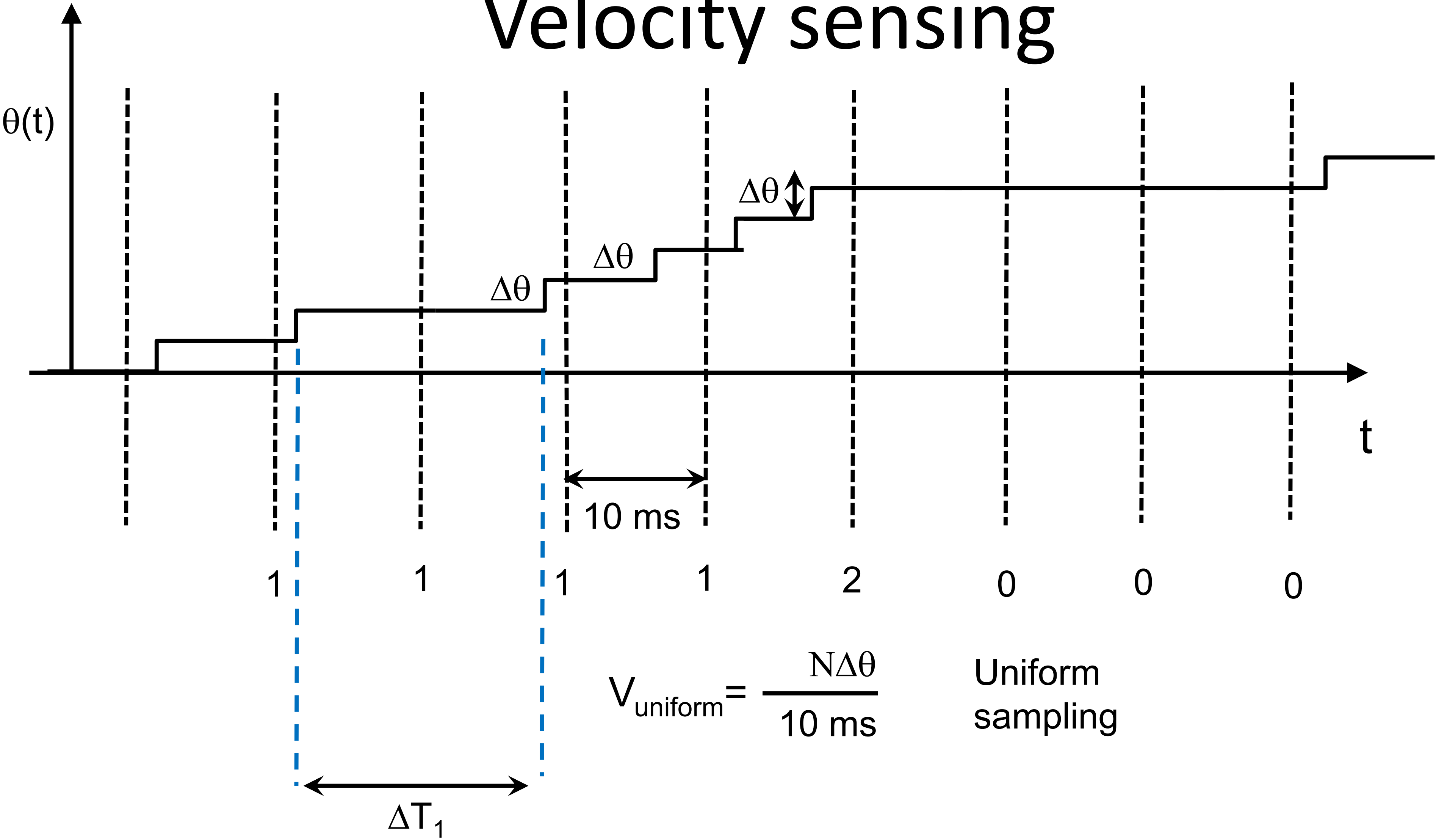


- Choose current 4 mA in LED
- Vcc = 3.3 V
- May want regulated/clean voltage for Vcc

100 us response time

velocity sensing

Velocity sensing



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

Uniform sampling

$$V_{edge} = \frac{\Delta\theta}{\Delta T_1}$$

Edge timing

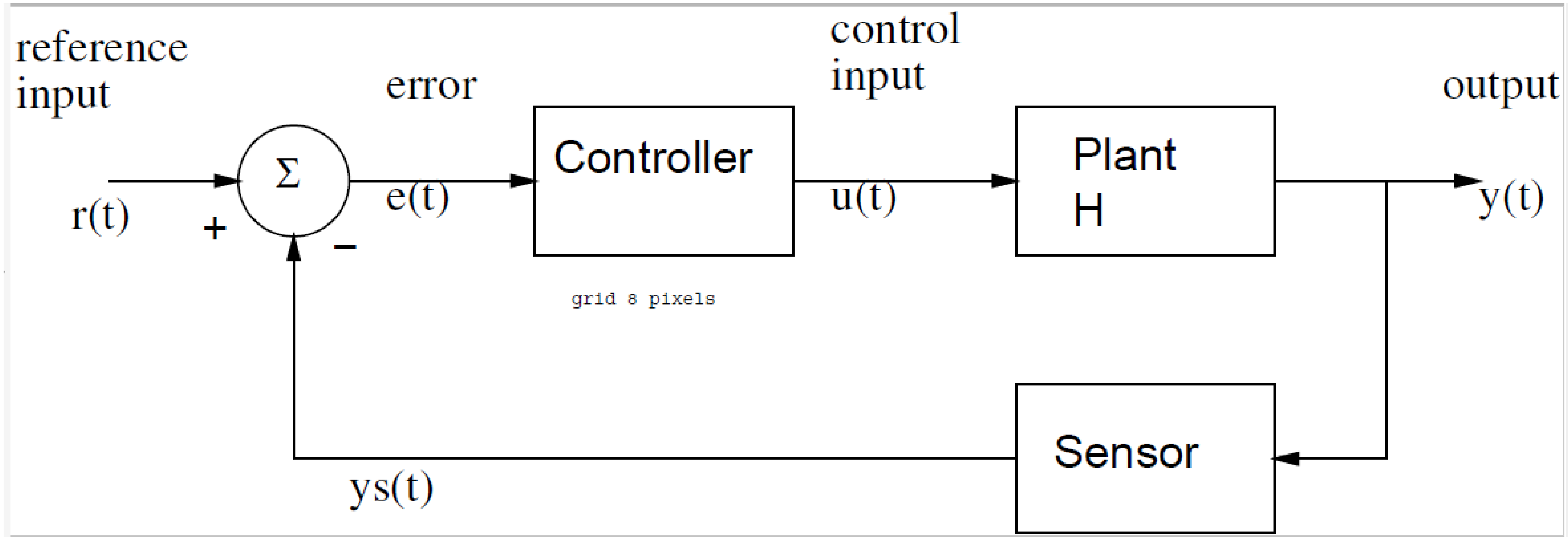
Velocity sensing (recap)

V~ (change in angle)/(change in time)

On board…

Topics
- Upcoming checkpoints
- Q1 Solution
- Project Proposal Feedback
- Q2
- Velocity Sensing Recap
- Velocity Control (intro)
- Line Sensor TSL1401
- HW1 Track Detection
- Steering control (intro to intro)

Velocity control overview



On board…

Proportional control:

U = kp*e = kp* (r-y);

Here: r is desired velocity, U is PWM %

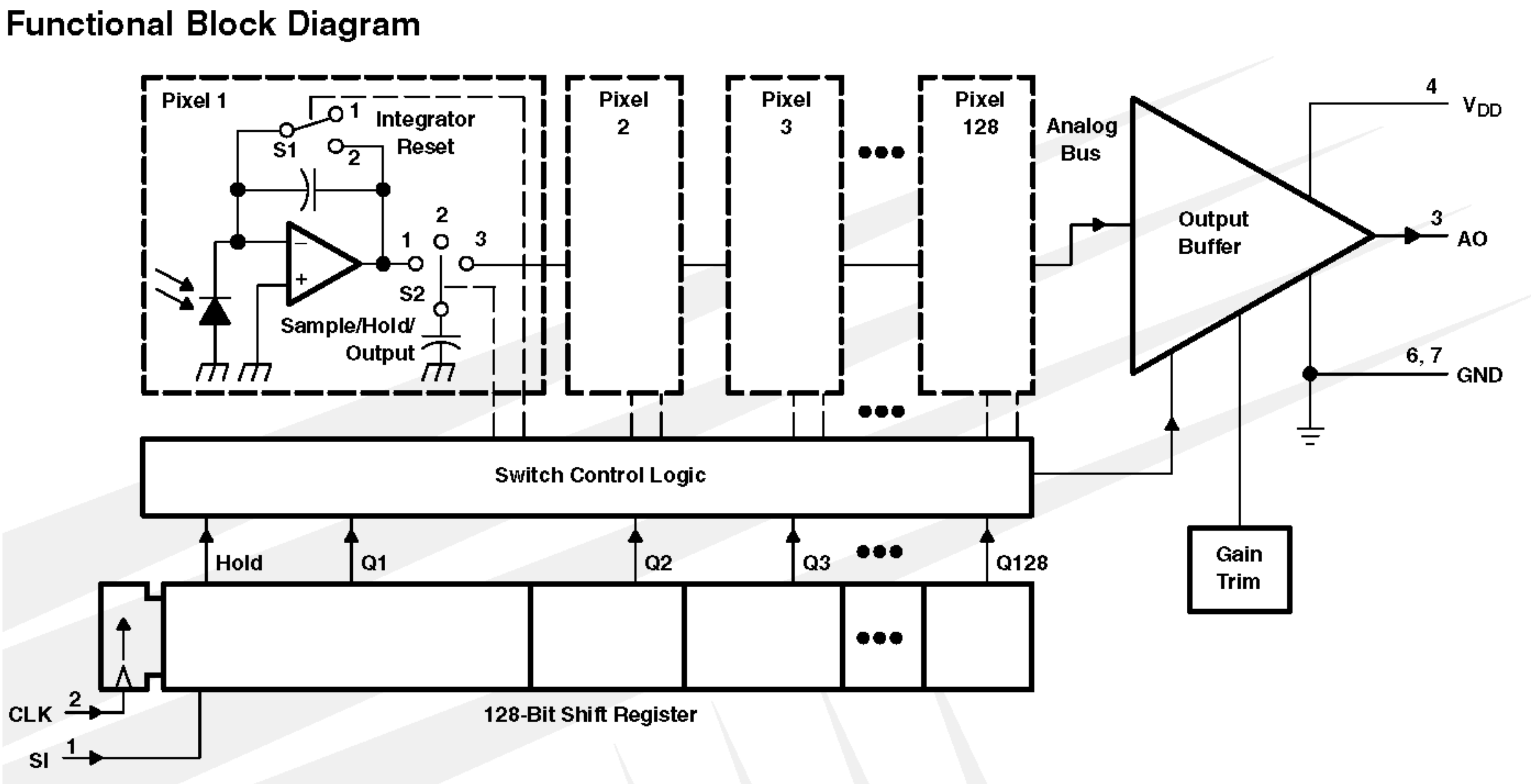
Proportional + integral control

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

Topics

- Upcoming checkpoints
- Q1 Solution
- Project Proposal Feedback
- Q2
- Velocity Sensing Recap
- Velocity Control (intro)
- Line Sensor TSL1401
- HW1 Track Detection
- Steering control (intro to intro)

TSL 1401 line sensor



TSL 1401 line sensor

PARAMETER MEASUREMENT INFORMATION

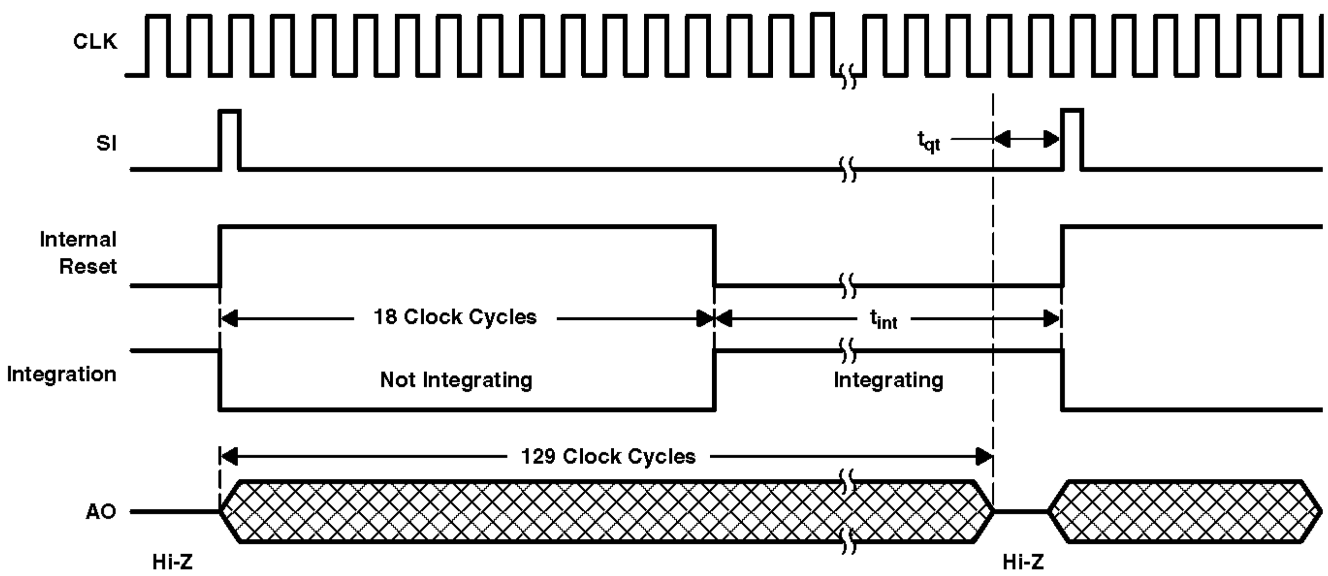
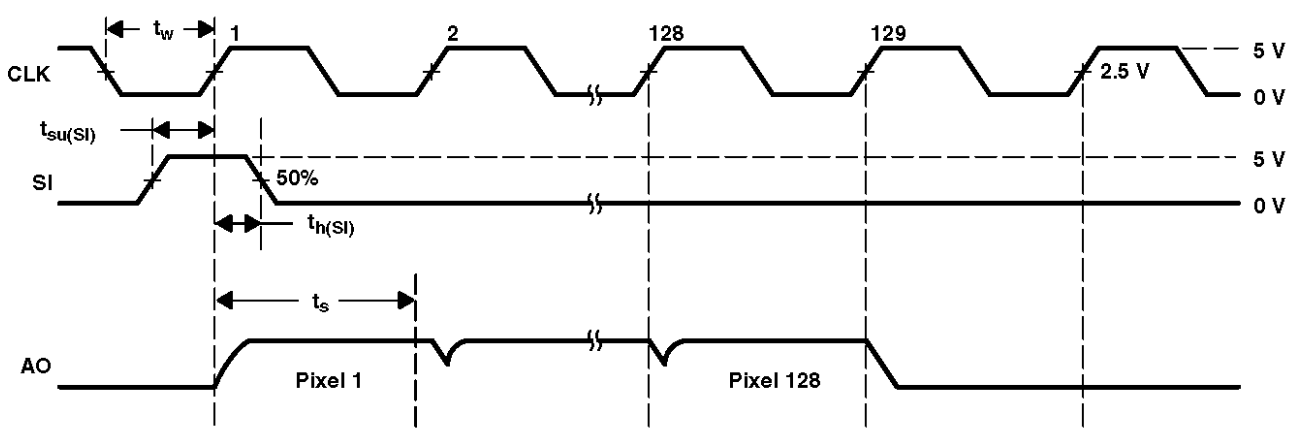


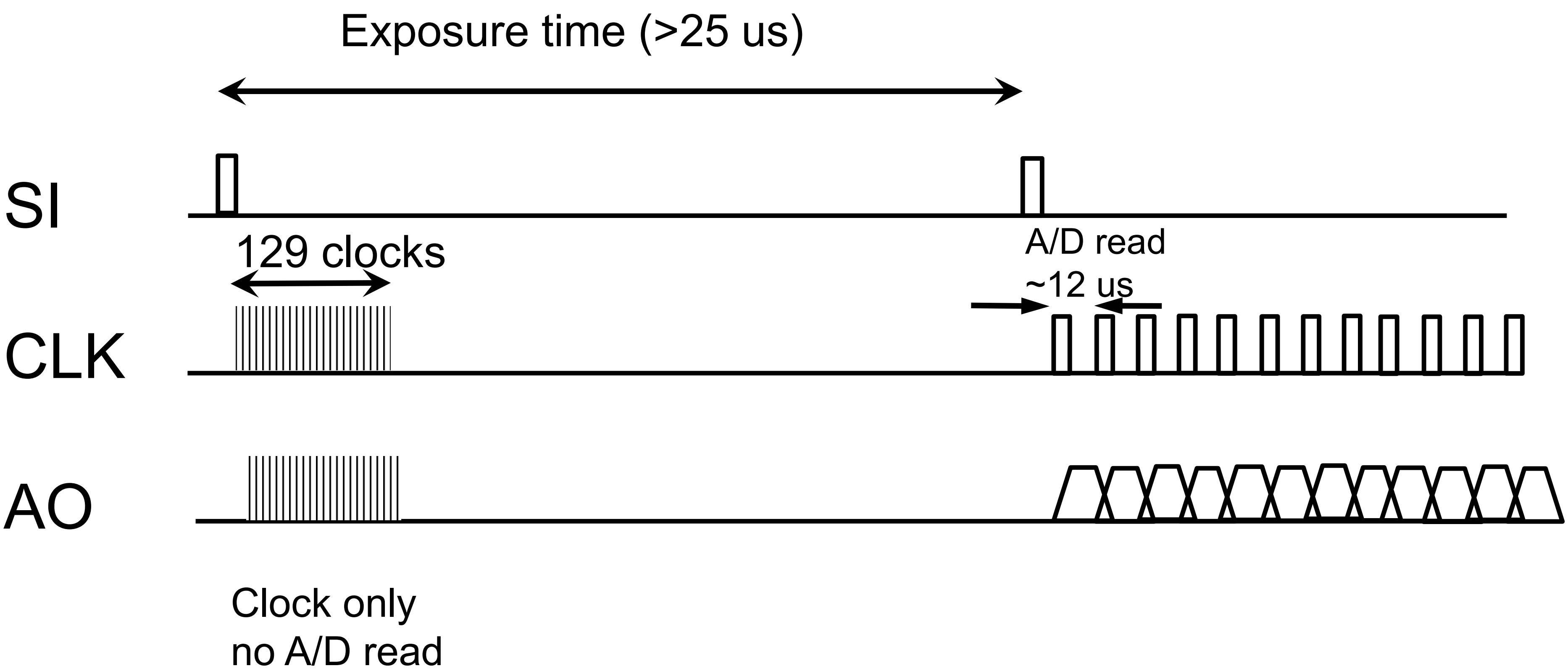
Figure 1. Timing Waveforms



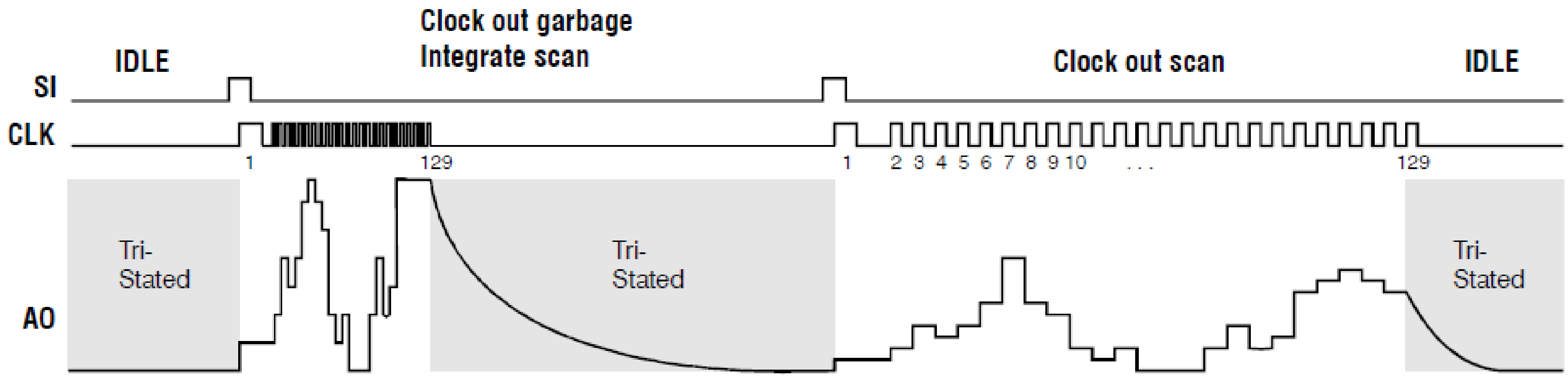
Line sensor

Figure 2. Operational Waveforms

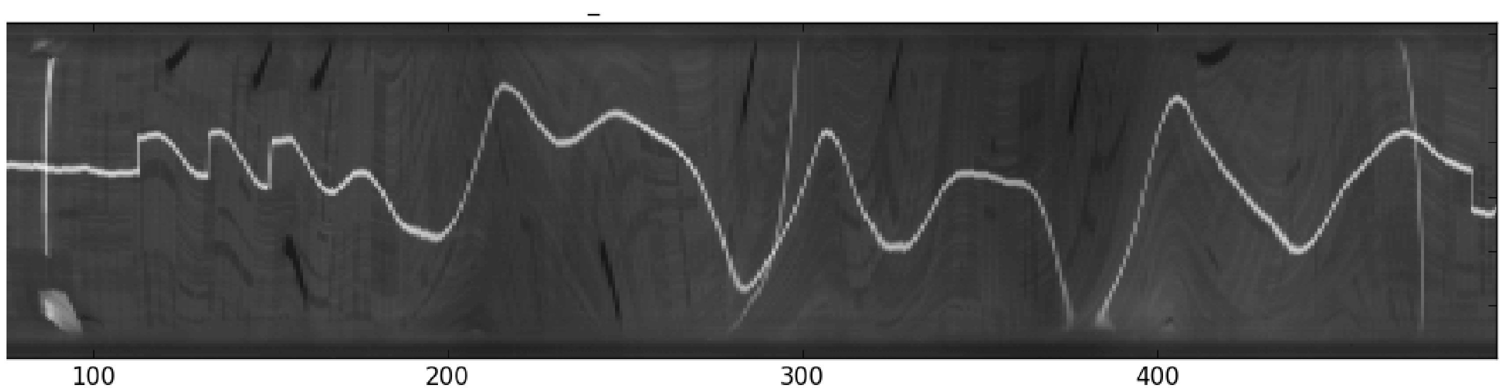
TSL 1401 line sensor- option exposure control
(would need to modify PRU code)



Automatic Gain Control



In all the discussion that follows, we will be using one-shot imaging.



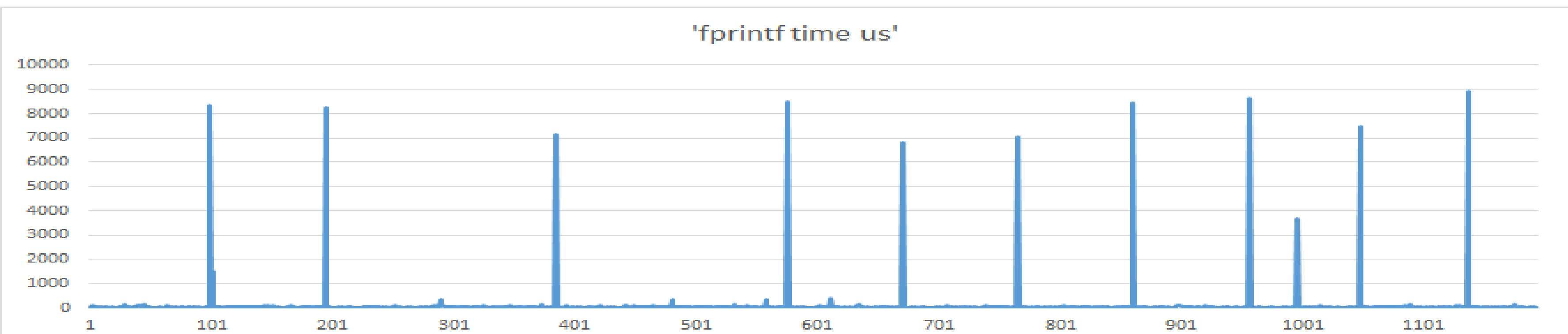
- Choose exposure time based on average illumination
- Keep frame rate constant e.g. read sensor twice 1+4 ➔ 4 +1 ms
- (Constant time is important for control- will see later)

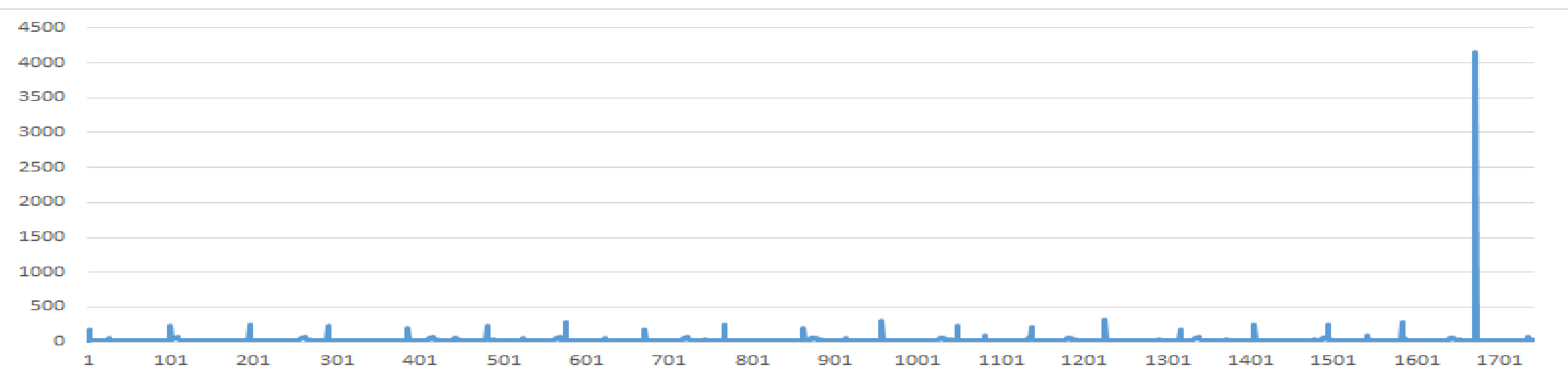
Debian Processes/Delay

```
htop
# systemctl disable avahi-daemon
# systemctl stop avahi-daemon

connmanctl> services
connmanctl> disable wifi

sudo kill -9 {avahi-daemon, rc_battery_monitor,
apache2}.
```



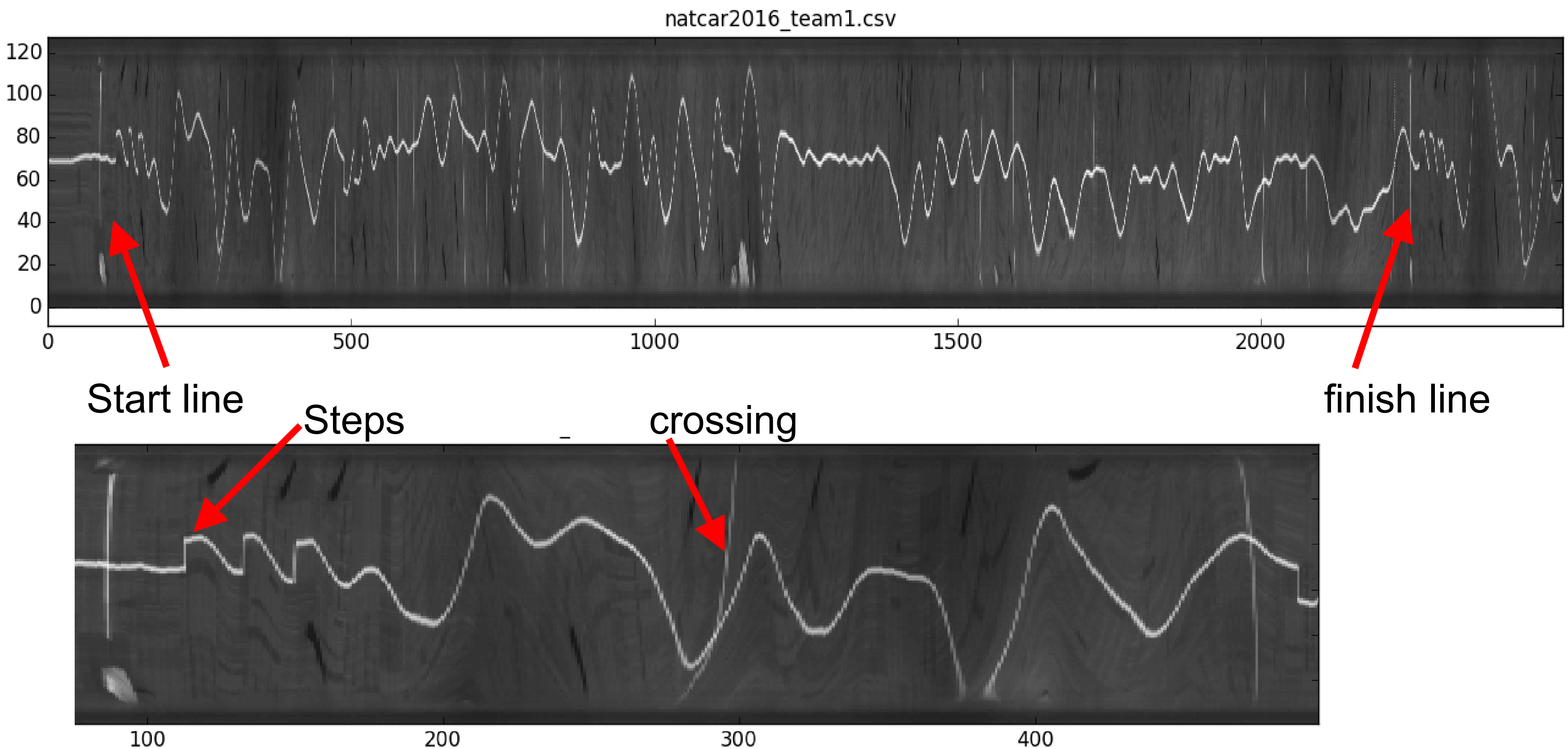


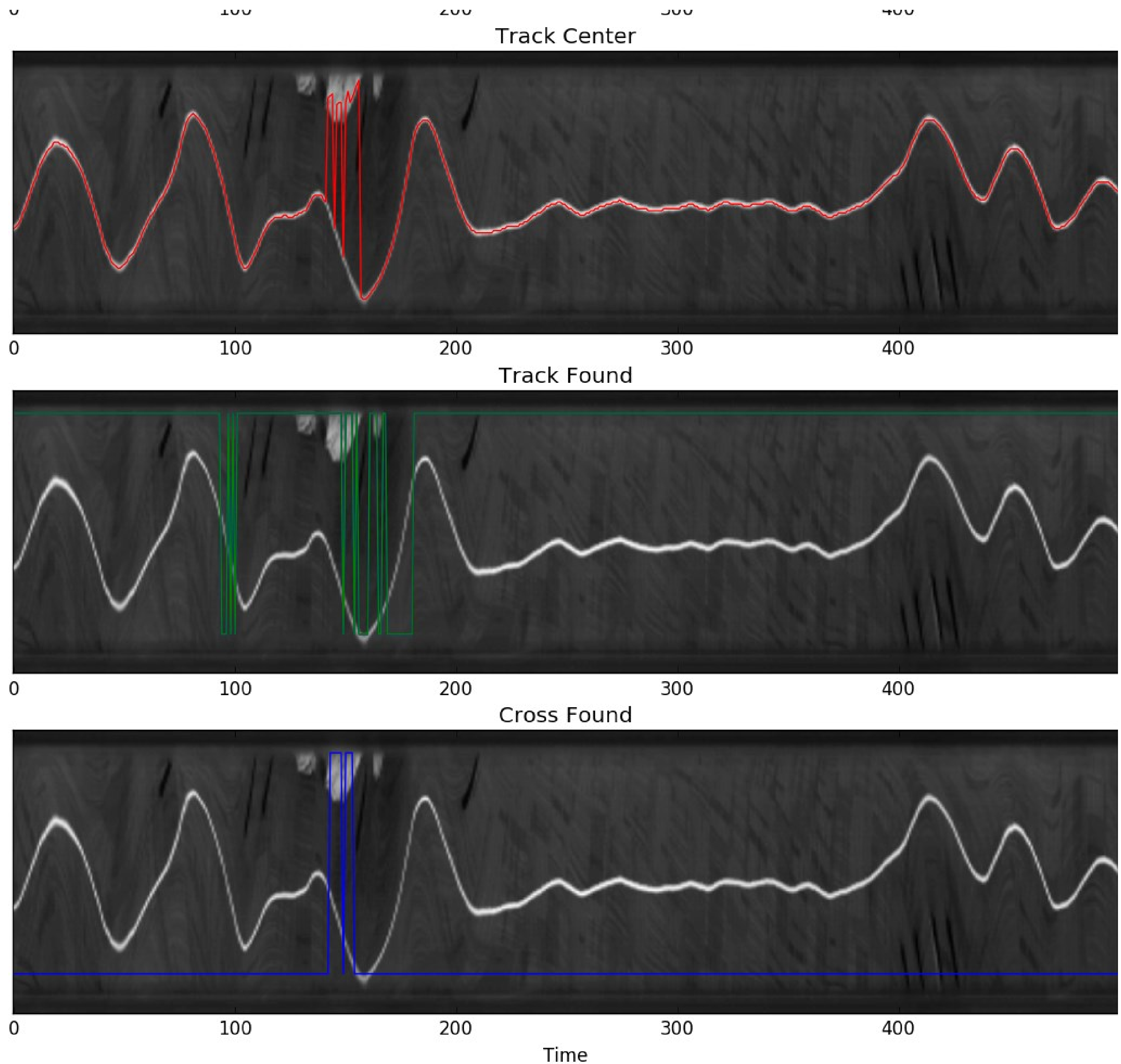
Topics

- Upcoming checkpoints
- Q1 Solution
- Project Proposal Feedback
- Q2
- Velocity Sensing Recap
- Velocity Control (intro)
- Line Sensor TSL1401
- HW1 Track Detection
- Steering control (intro to intro)

Python Template for Track Finding

\# track_center_list - A length n array of integers from 0 to 127.
Represents the predicted center of the line in each frame.
\# track_found_list - A length n array of Booleans {10,100}.
Represents whether or not each frame contains a detected line.
\# cross_found_list - A length n array of Booleans {10,100}.
Represents whether or not each frame contains a crossing.

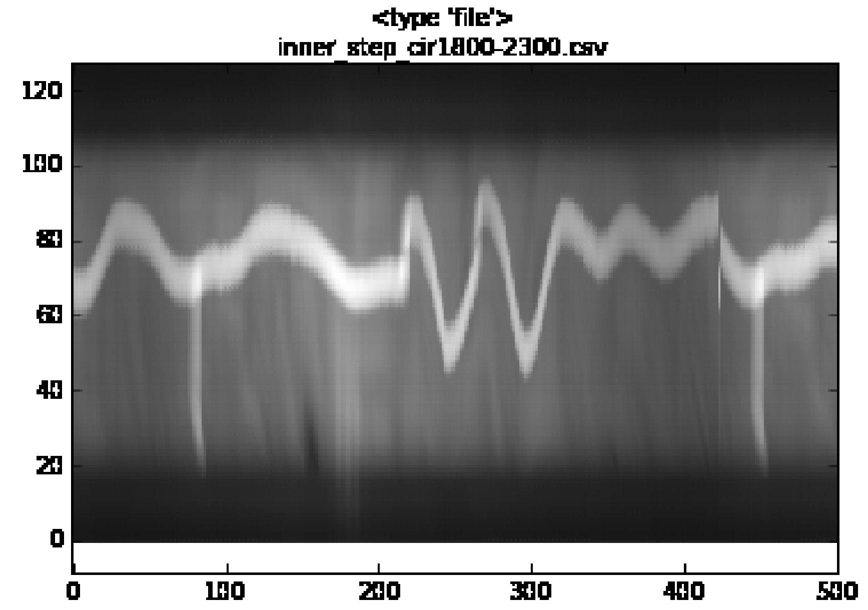



0
100
200
300
400
Track Center
0
100
200
300
400
Track Found
0
100
200
300
400
Cross Found
0
100
200
300
400
Time

Possible algorithms for line detection

e.g. scipy.signal.filter. Many options. Here 3 suggestions:

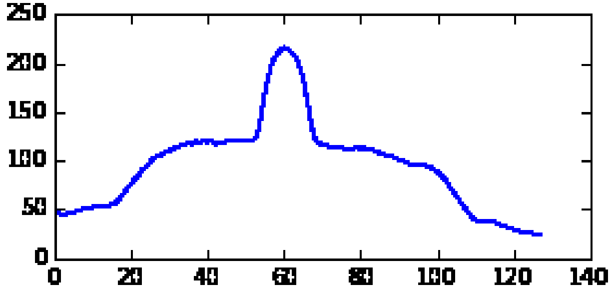
- Subtraction- to find left and right edge of line (ok if not noisy, somewhat lighting invariant)
- Difference of gaussians (idea is to smooth then differentiate)
- Correlation (best match position for known features)
  - scipy.signal.correlate



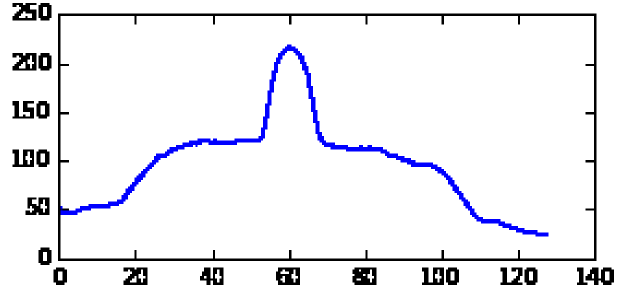
Alternative #1 frame subtraction

TSL 1401 line sensor 8 bit

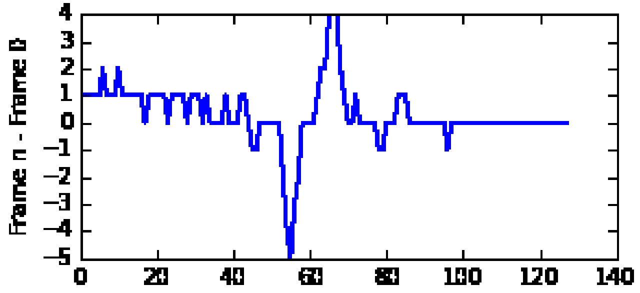
Frame 0



Frame 1



Frame 1-Frame 0



Frame 0

Frame 2

Frame 2-Frame 0

Notes: peak shows edge of track. Noisy, only 1 pixel resolution.

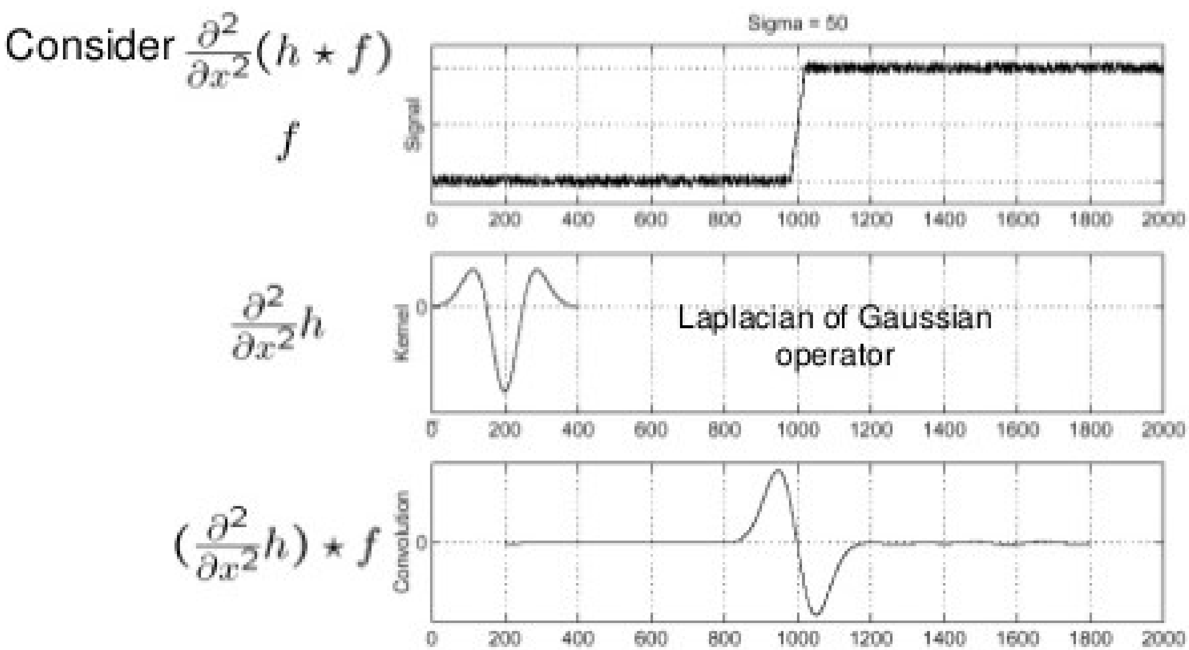
Alternative #2 Difference of Gaussians

Laplacian of Gaussian

$$\triangle[G_\sigma(x,y) * f(x,y)] = [\triangle G_\sigma(x,y)] * f(x,y) = LoG * f(x,y)$$

Convolve with Difference of Gaussians kernel (approx. to LoG)

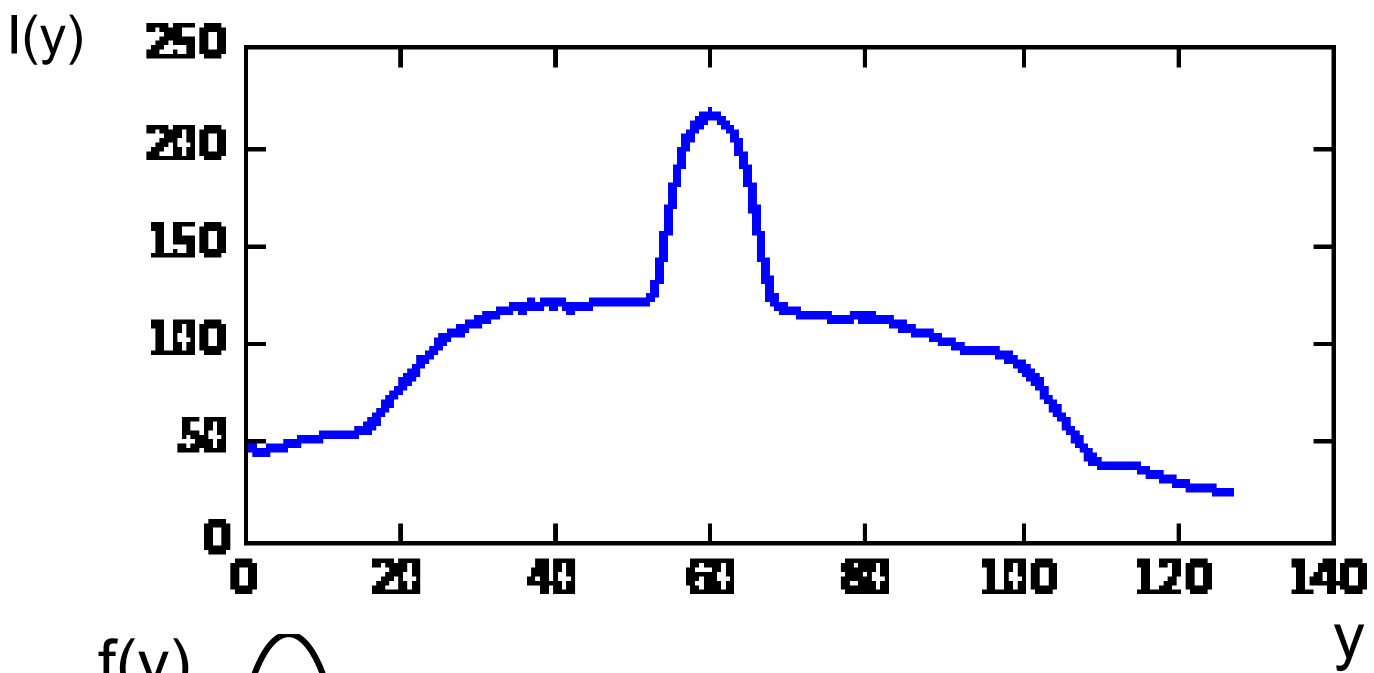
$$\Gamma_{\sigma_1,\sigma_2}(x) = I * \frac{1}{\sigma_1\sqrt{2\pi}}\, e^{-(x^2)/(2\sigma_1^2)} - I * \frac{1}{\sigma_2\sqrt{2\pi}}\, e^{-(x^2)/(2\sigma_2^2)}.$$



https://image.slidesharecdn.com/cbirfeatures-150705141111-lva1-app6892/95/cbir-features-47-638.jpg?cb=1436105787

Notes: zero crossing is edge location

Alternative #3 Correlation



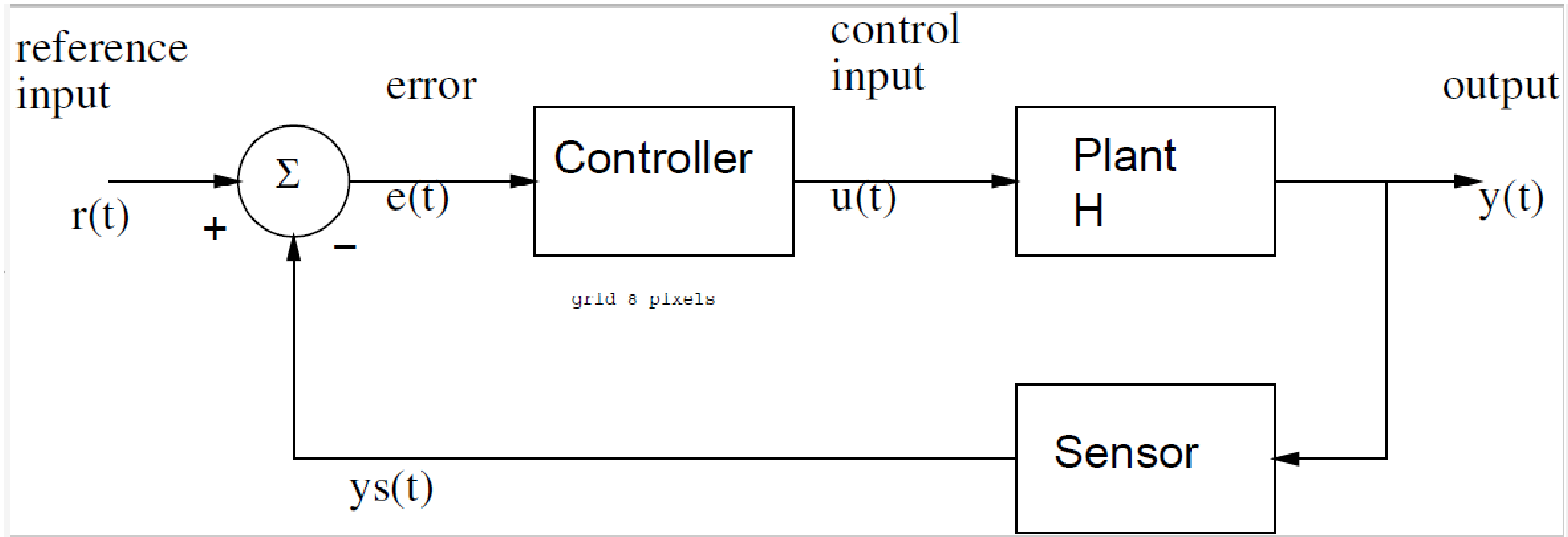
$$\underset{\Delta y}{\arg\min} \; \| I(y) - f(y - \Delta y) \|_2$$

Notes: normalize, find by least squares or search. Can use $\Delta y(n-1)$ to initialize

Topics

- Upcoming checkpoints
- Q1 Solution
- Project Proposal Feedback
- Q2
- Velocity Sensing Recap
- Velocity Control (intro)
- Line Sensor TSL1401
- HW1 Track Detection
- Steering control (intro to intro)

Steering Control overview



On board…

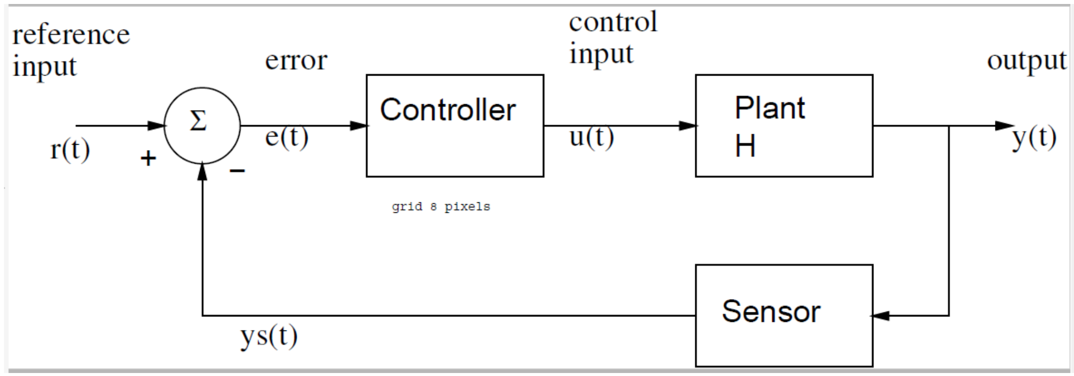
Proportional control:

U = kp*e = kp* (r-y);

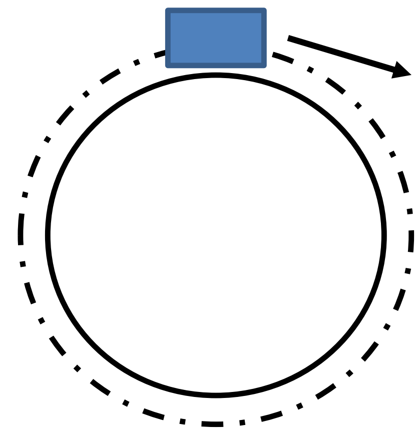
Proportional + integral control

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

Bicycle Steering Control



Note steady state error:
car follows larger radius

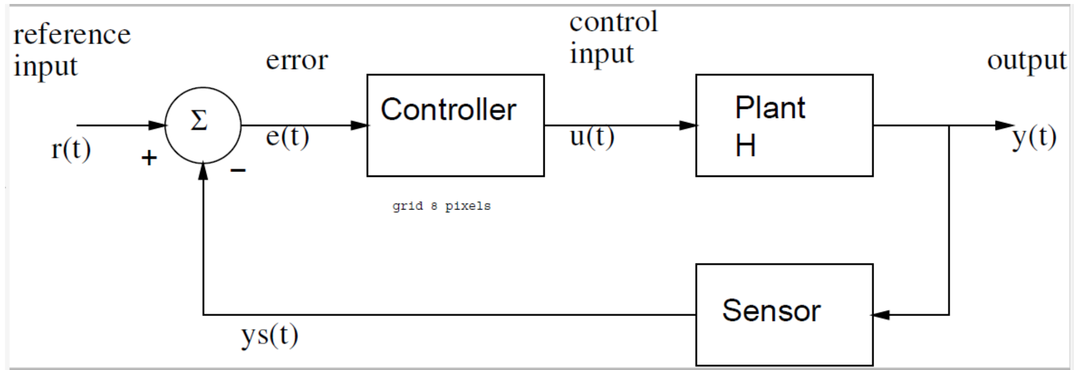


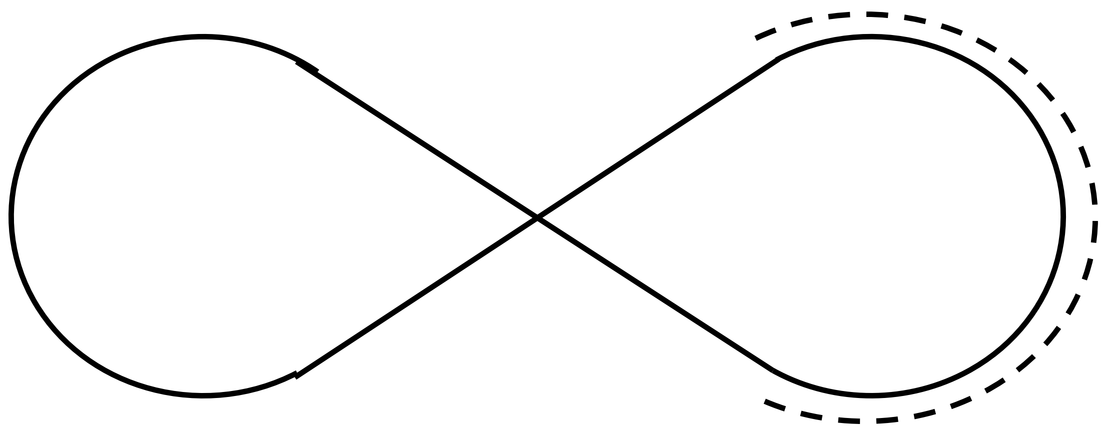
Proportional control:
r = 0 (to be on straight track)
δ=u = kp*e

Proportional+derivative

P+I+D

Proportional + Integral

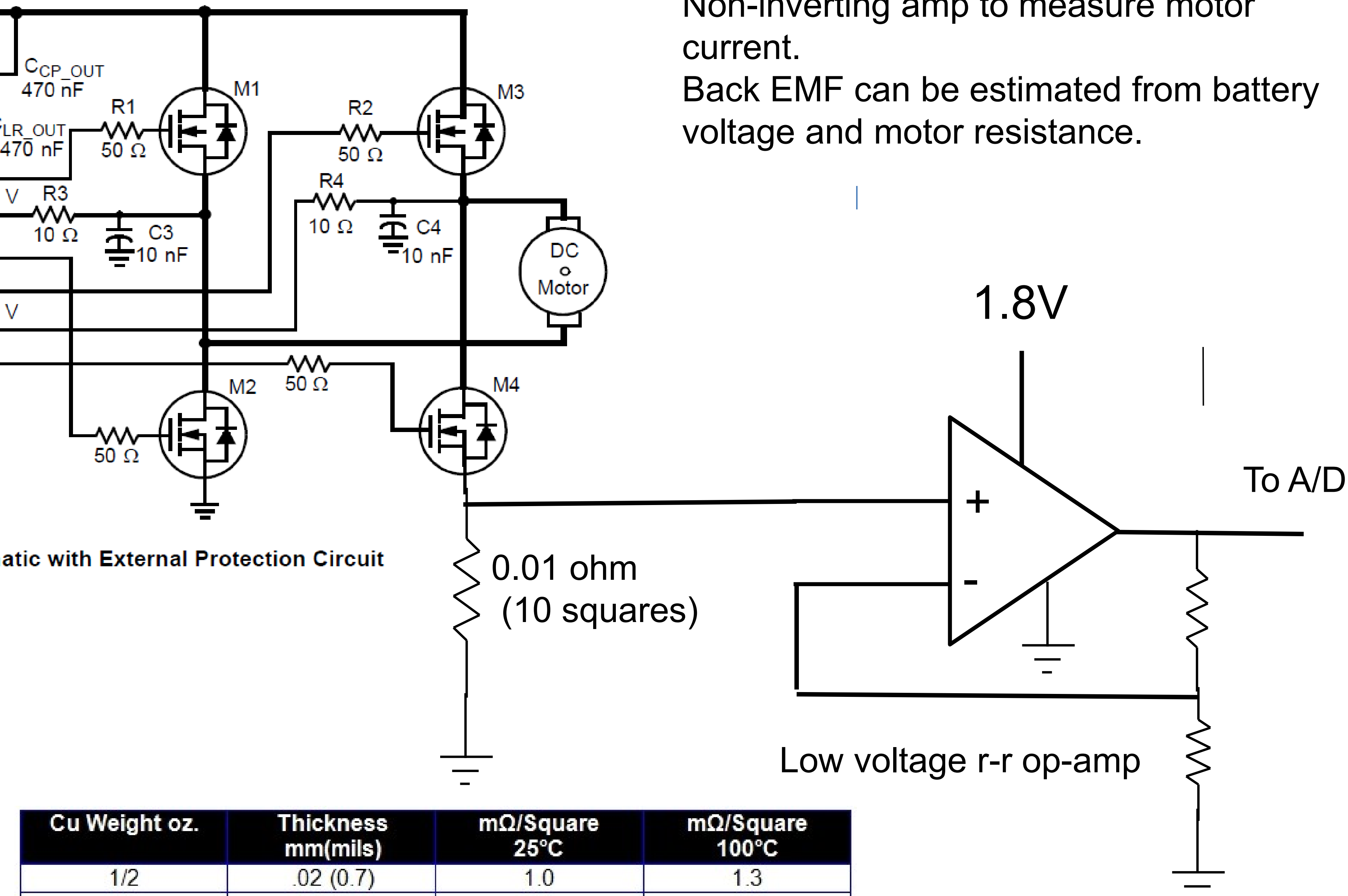




On board Anti-windup

Extra Slides

Non-inverting amp to measure motor current.
Back EMF can be estimated from battery voltage and motor resistance.

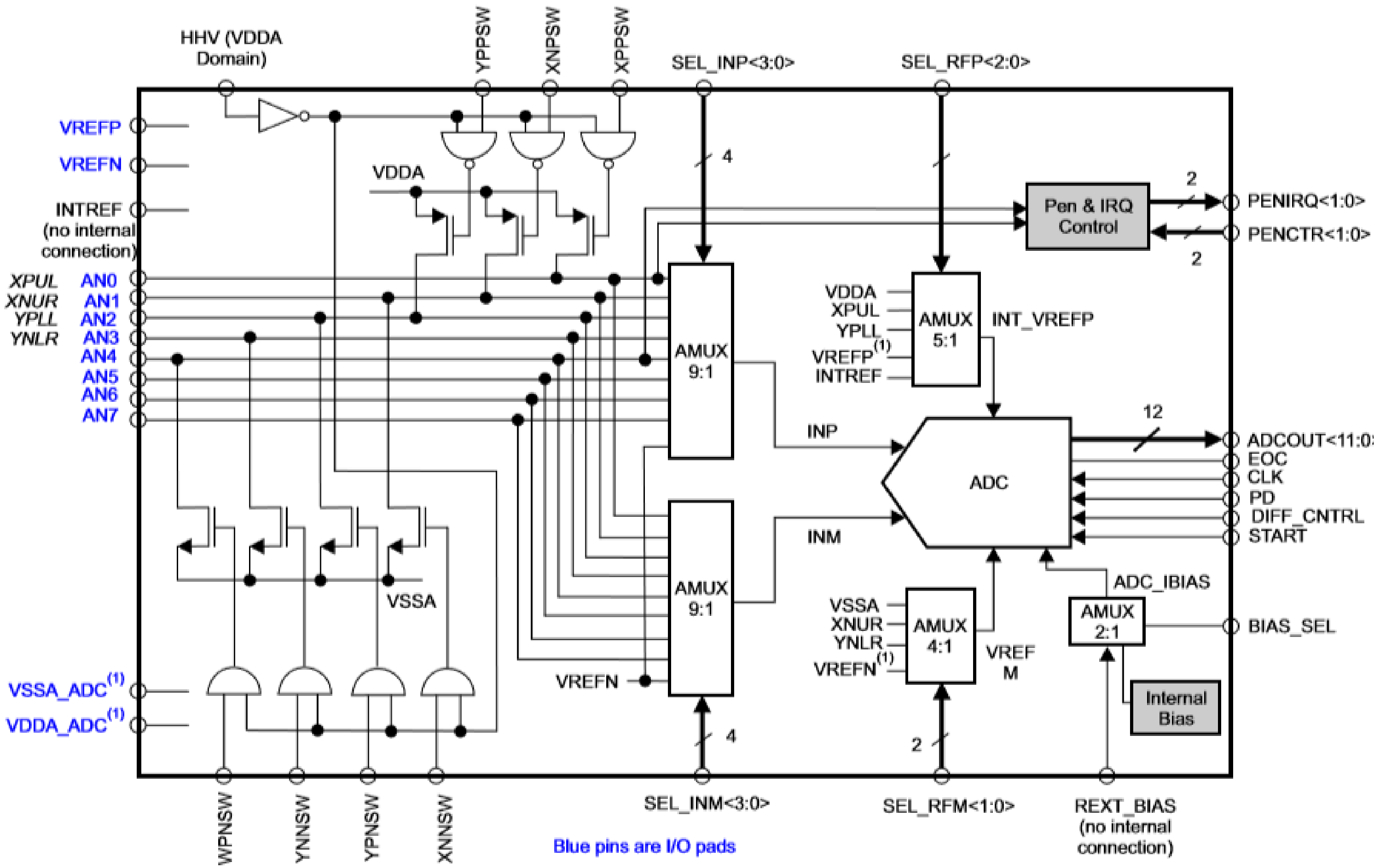


Cu Weight oz.	Thickness mm(mils)	mΩ/Square 25°C	mΩ/Square 100°C
1/2	.02 (0.7)	1.0	1.3
1	.04 (1.4)	0.5	0.65

Back EMF velocity sensing

Analog/Digital Overview

Figure 12-2. Functional Block Diagram



(1) In the device-specific datasheet:

- VDDA_ADC and VSSA_ADC are referred to as "Internal References"
- VREFP and VREFN are referred to as "External References"

Caution: 1.8V MAXIMUM

Note: lots of sequencing/delays.

Set up in PRU0 to read 128 elements at ~8 us/sample.