Feedback control II

Fast Robots, ECE4160/5160, MAE 4190/5190

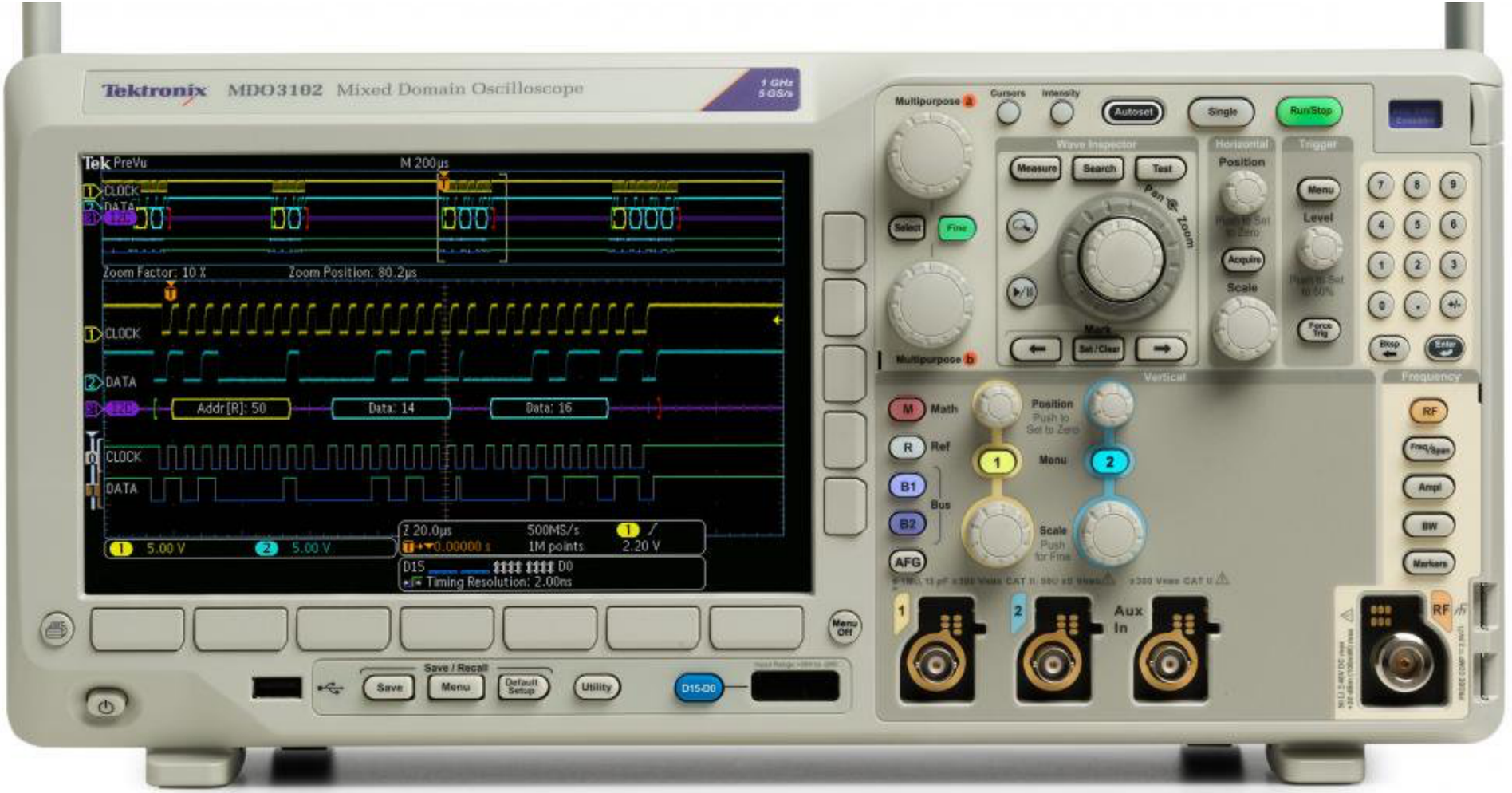
E. Farrell Helbling, 2/12/25
Credit: Prof. Kirstin Petersen

Class Action Items

- Please check our open hours if you need to go back to the lab to solder!
- Reminder: there is no lab next week and lab 3 is not due until Feb. 24-25. If you used a slip week for lab 2, this is still due next week!
- Lab 4 also has a significant soldering component. If you can, work on this early (i.e., next week after February break).
- Note: we will typically return grades/ feedback on your write-ups two weeks after the original submission date!

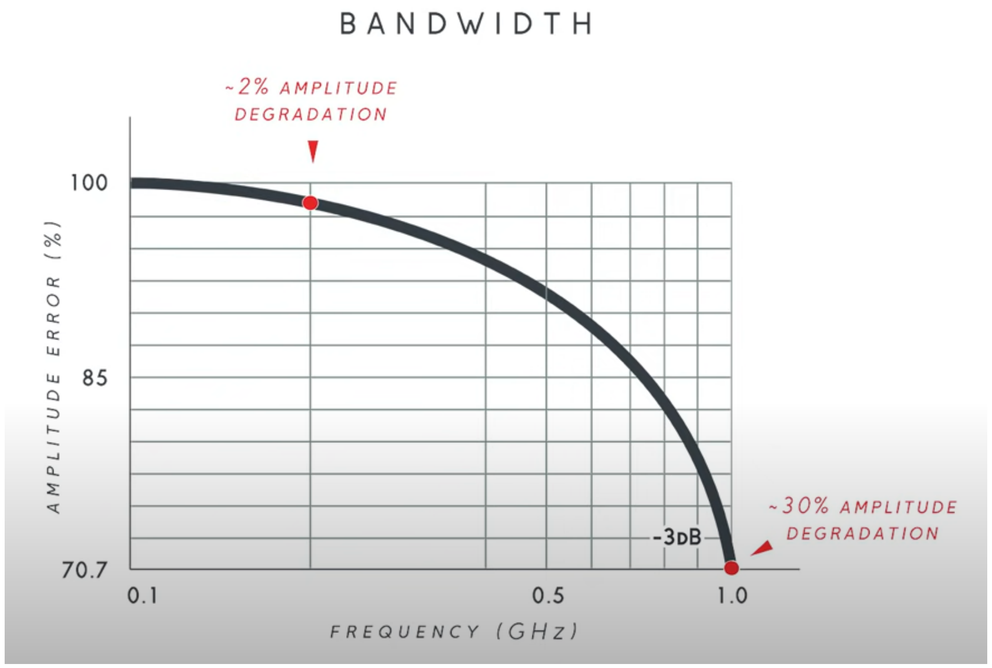
Oscilloscopes

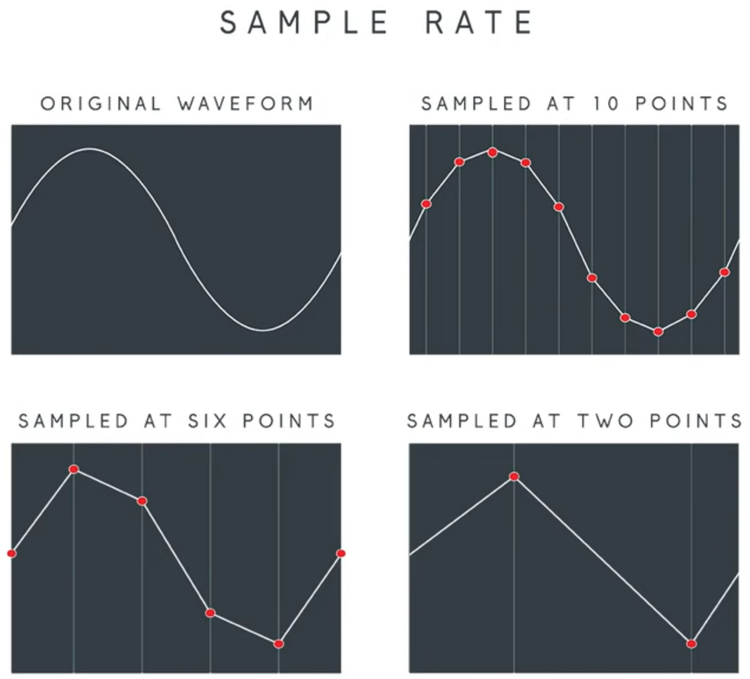
Oscilloscope setup



Oscilloscope setup

- Bandwidth
- Sample rate
- Resolution

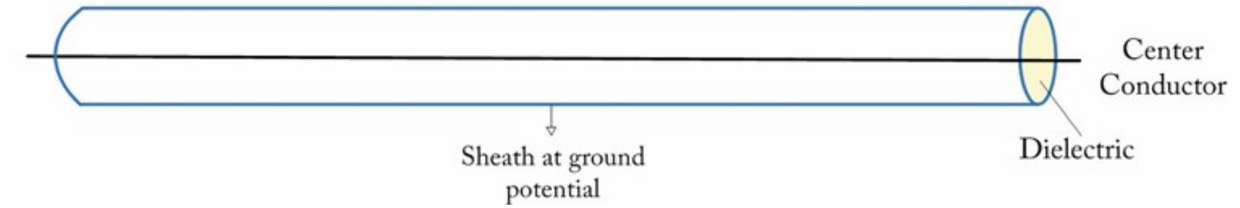




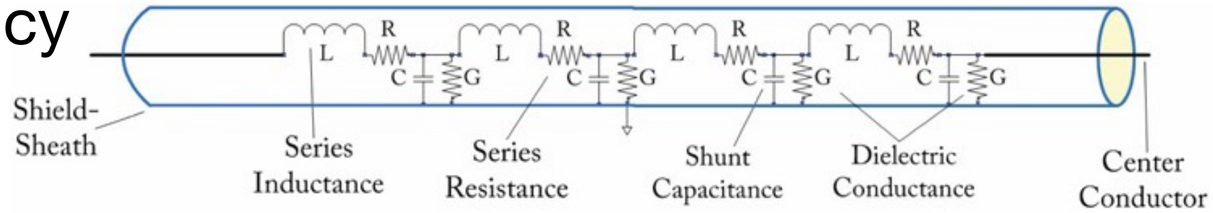
Oscilloscope Probes

- Scope inputs resemble a 16pF capacitor in parallel with a 1MOhm resistor
- At high frequencies the coax cable acts as a low pass filter
- 1x attenuation for low amplitude, low frequency signals
- 10x attenuation for load-sensitive circuits, high-frequency or high-amplitude signals

Low frequency coax cable

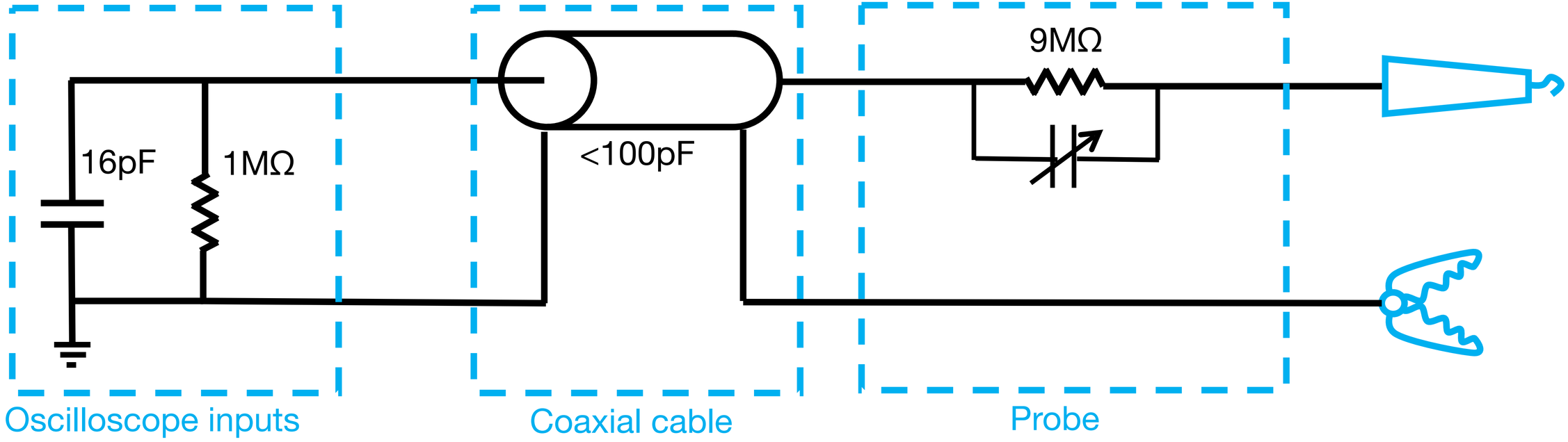


High frequency circuit



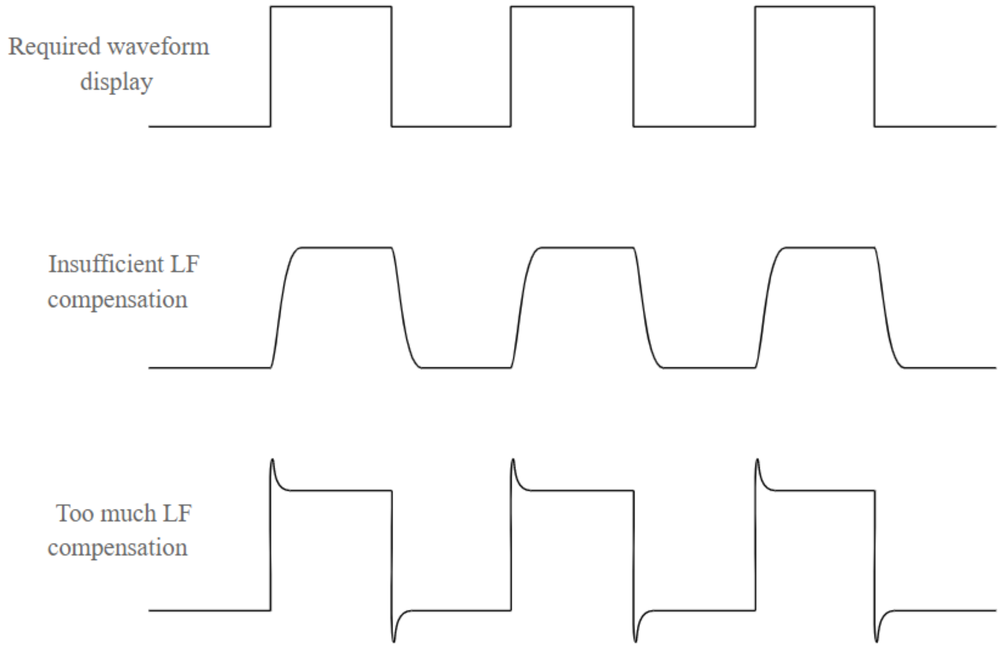
Oscilloscope Probes

- Scope inputs resemble a 16pF capacitor in parallel with a 1MOhm resistor
- At high frequencies the coax cable acts as a low pass filter
- 1x attenuation for low amplitude, low frequency signals
- 10x attenuation for load-sensitive circuits, high-frequency or high-amplitude signals

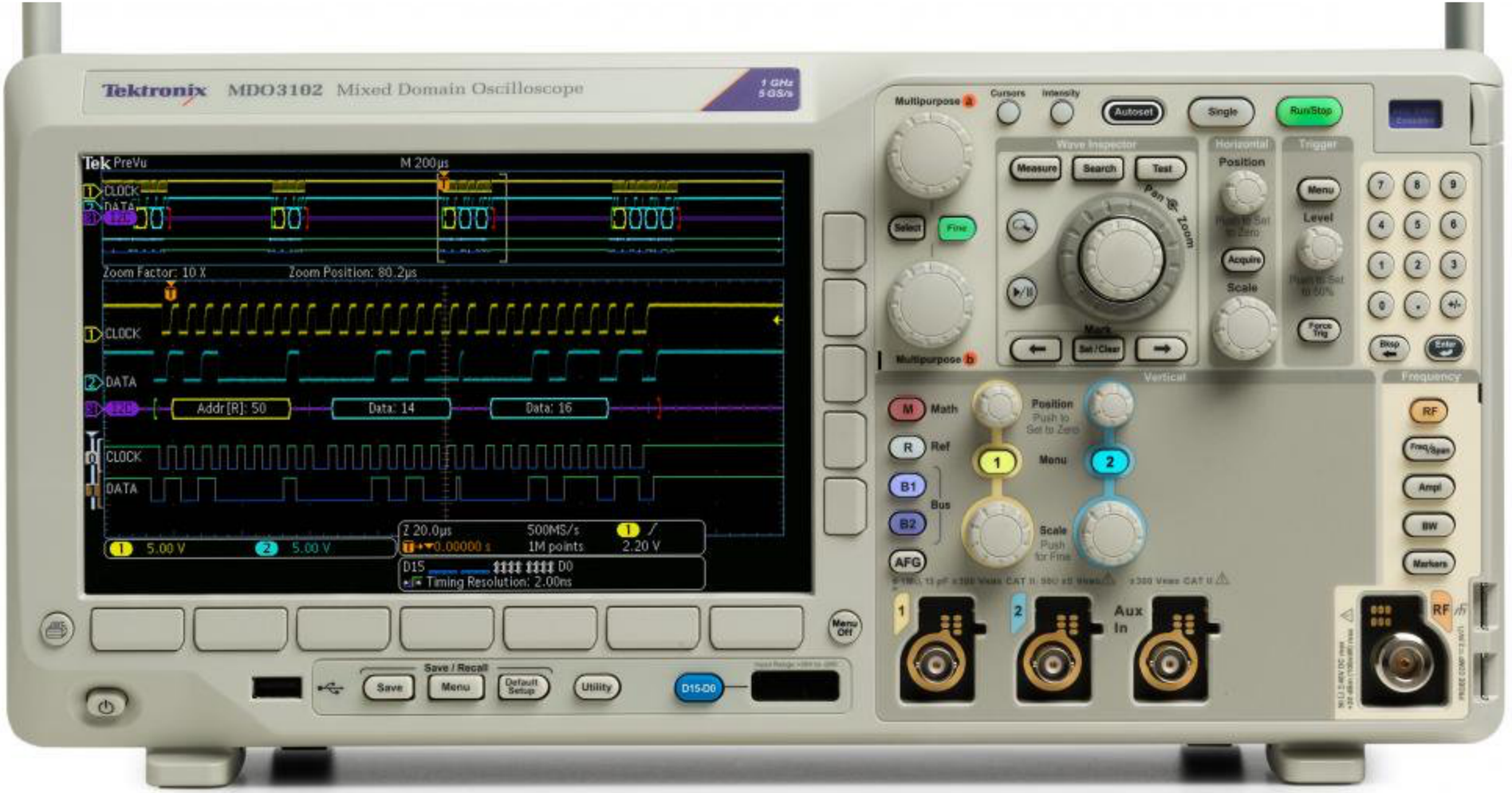


Oscilloscope Probes

- 10x probe calibration
  - Use the built-in square wave generator
  - Adjust capacitor until the square wave looks square!



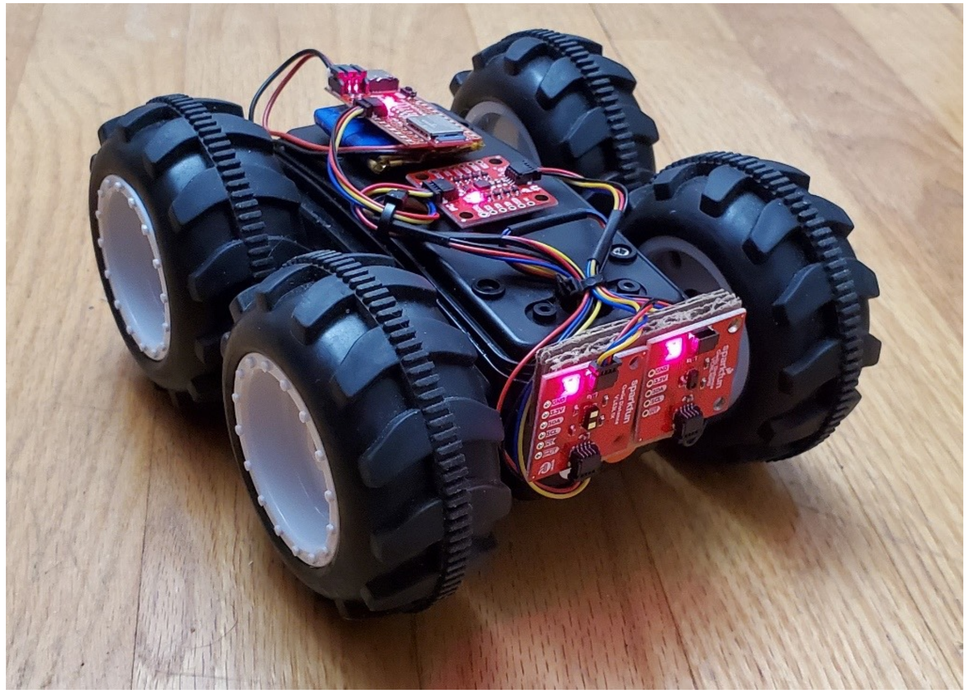
Oscilloscope setup

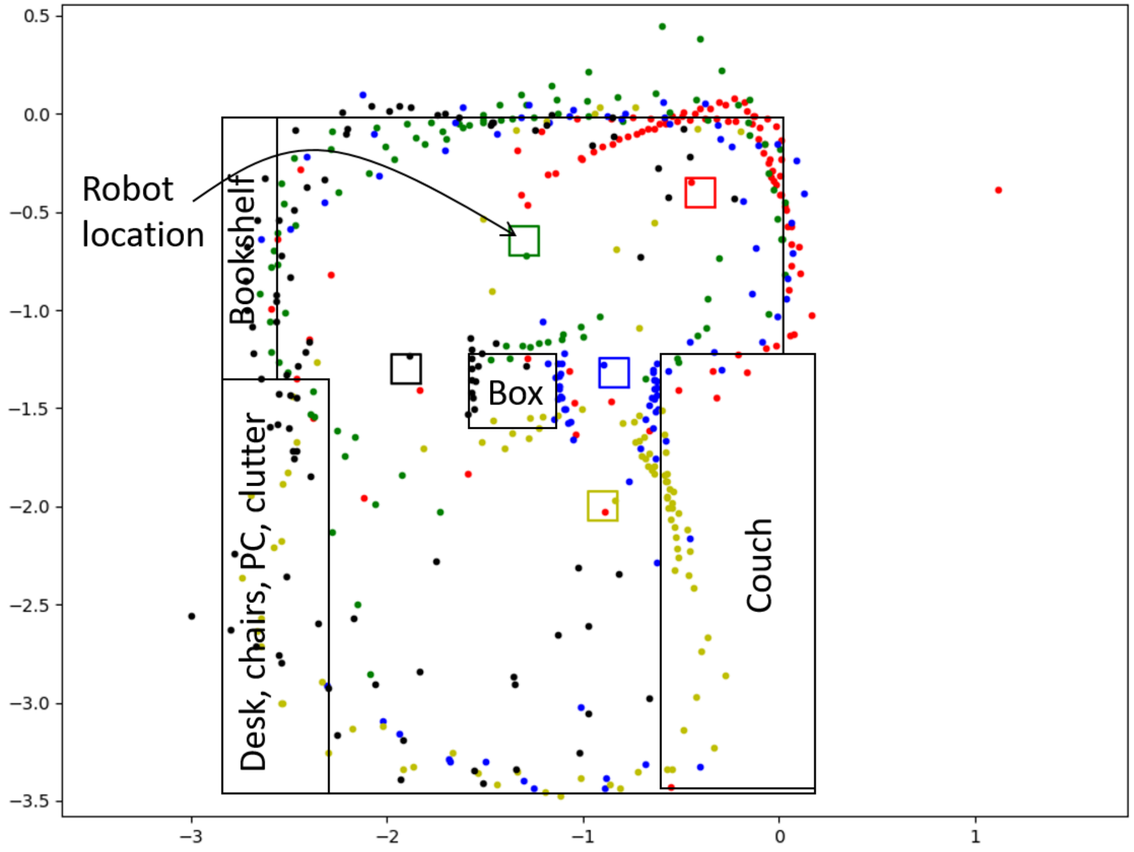


PID continued

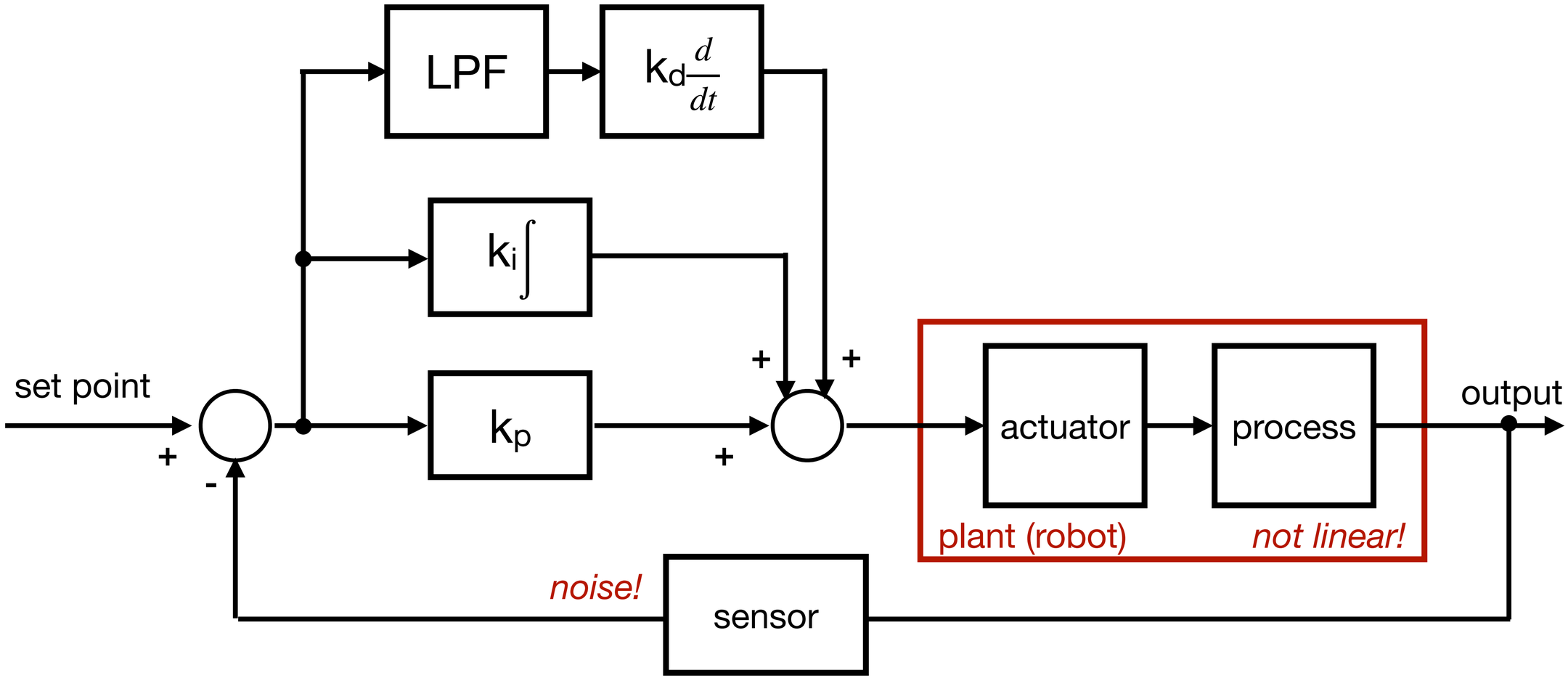
Feedback Control

- Stunts: maintain speed prediction at different battery levels, over different surfaces
- Mapping: evenly spaced out sensor readings
- Path execution: adhere to generated path plans

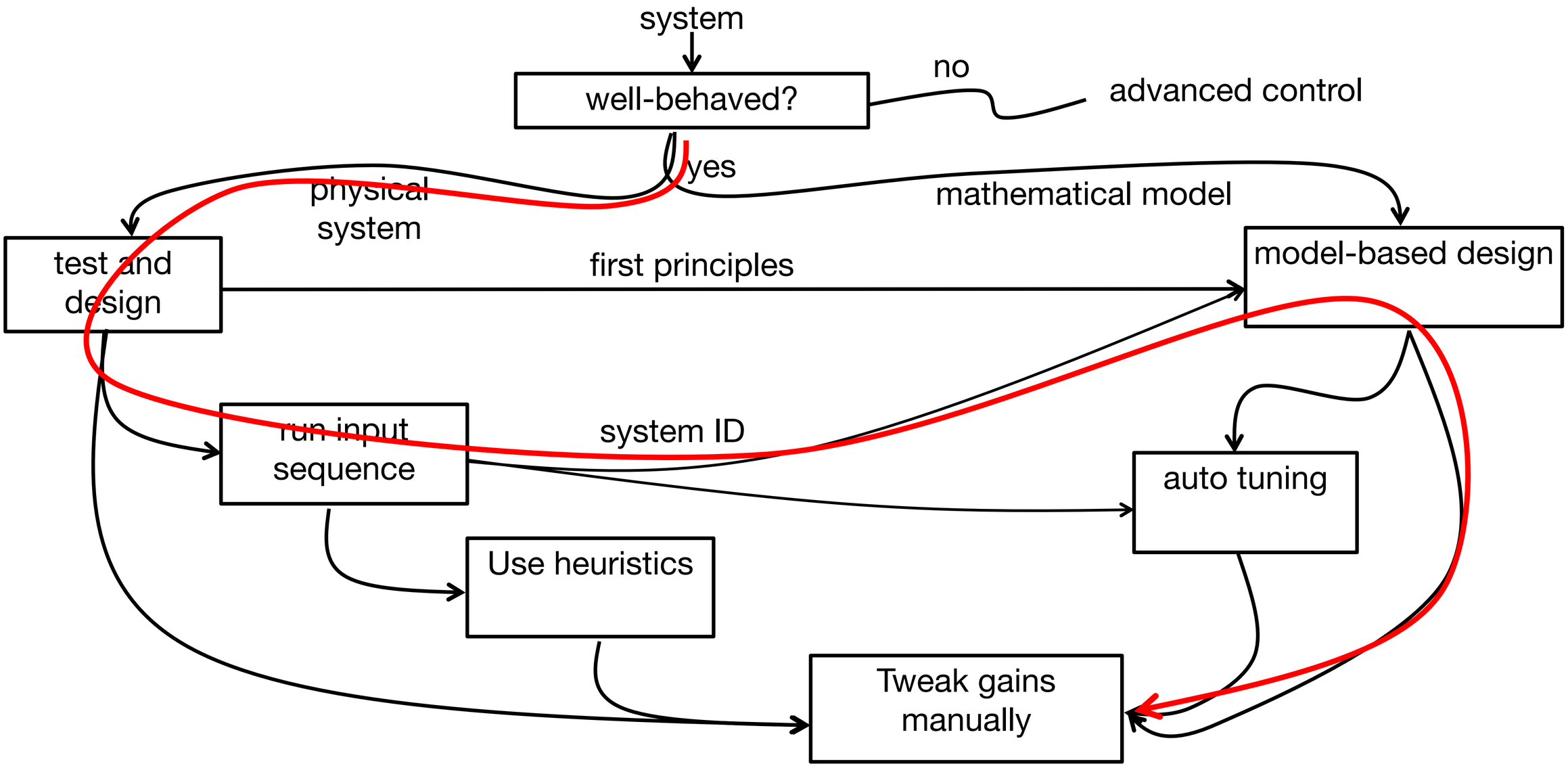




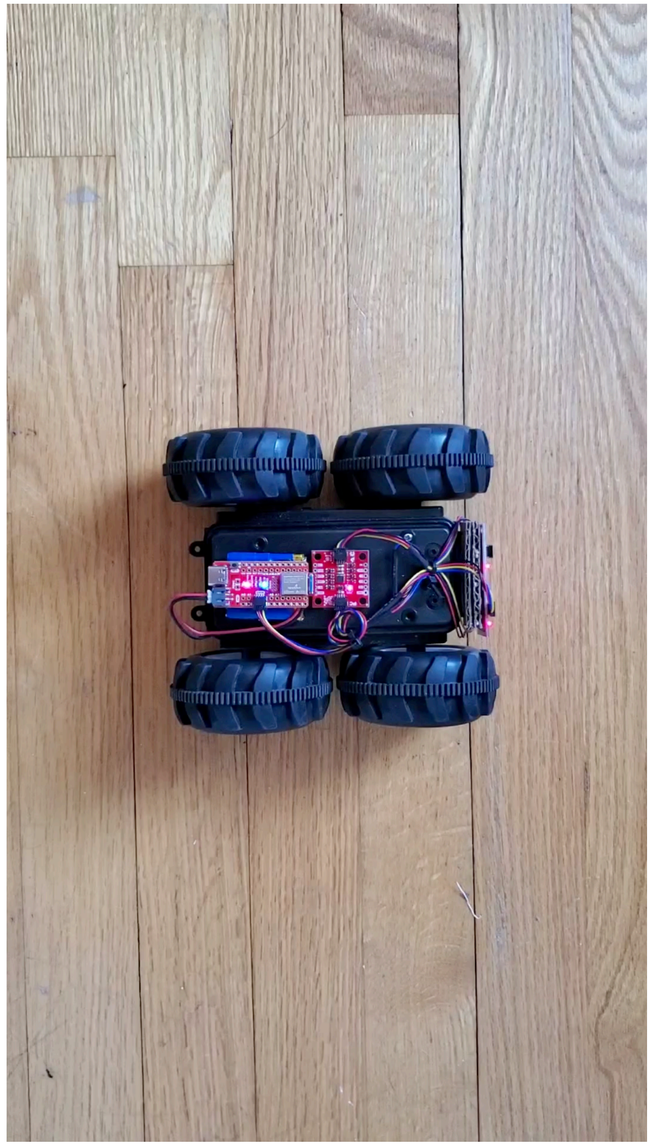
PID

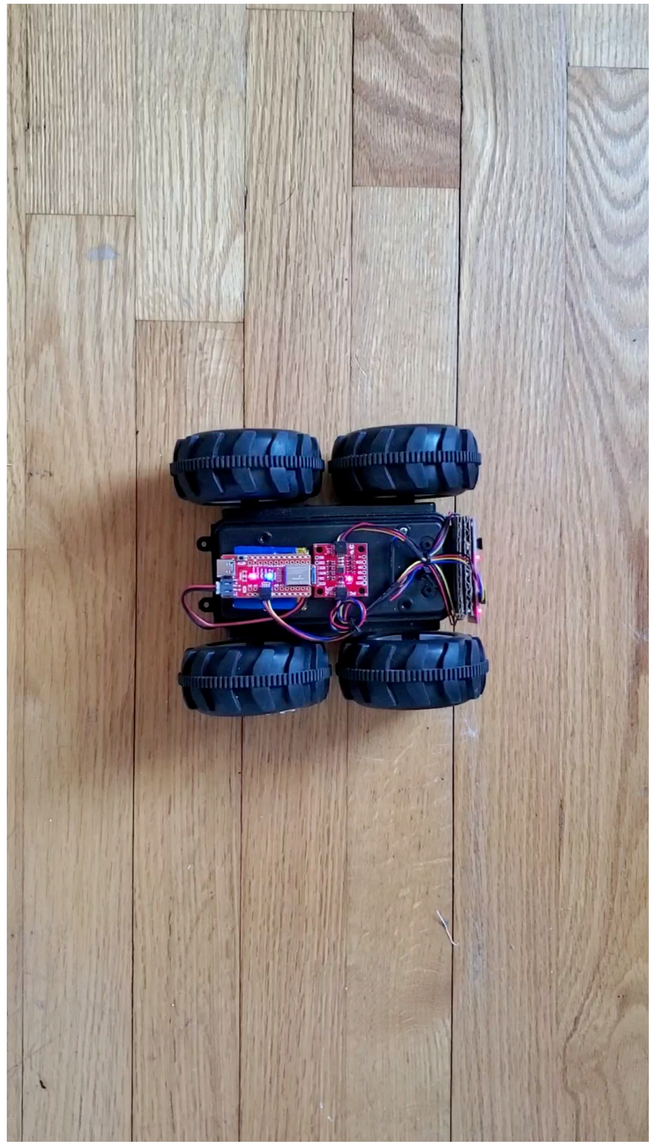

LPF
$k_d \frac{d}{dt}$
$k_i \int$
k_p
set point
+
-
+
+
+
actuator
process
output
plant (robot)
not linear!
noise!
sensor

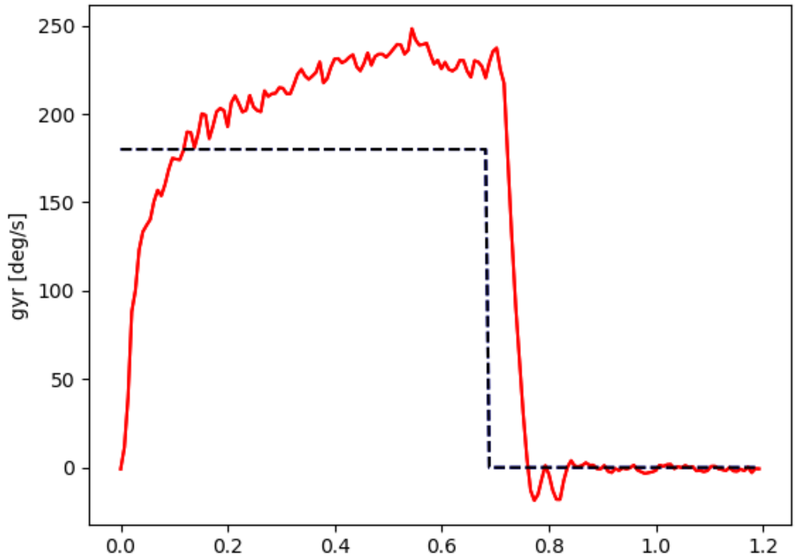
Tuning PID control

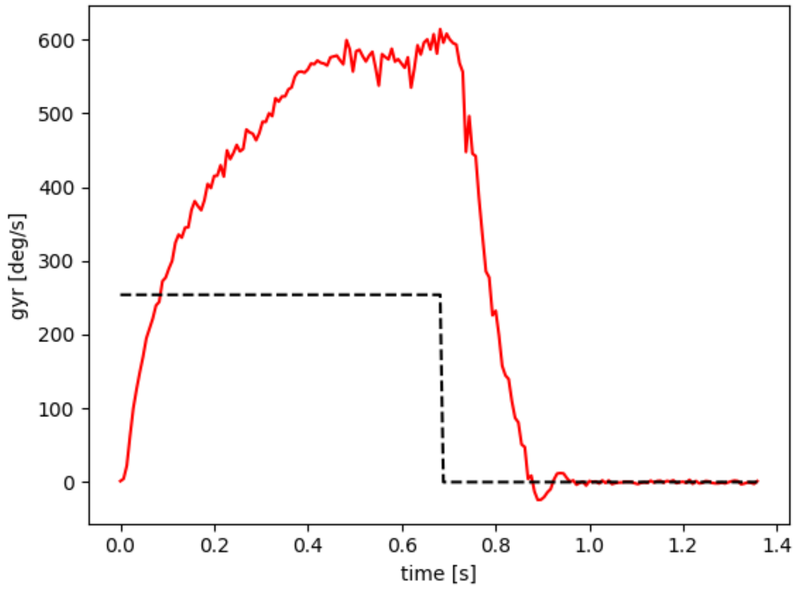


Simple model



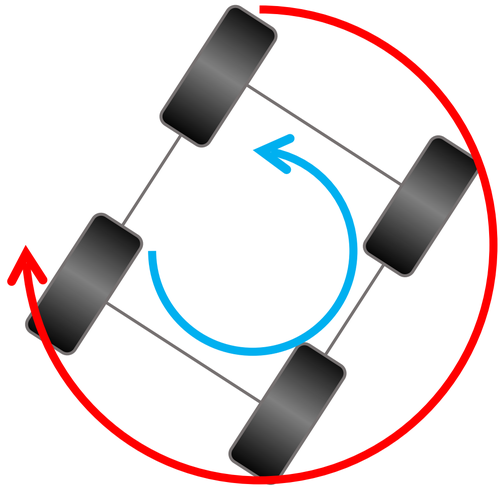






Equations of motion, angular speed

$$x = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}$$



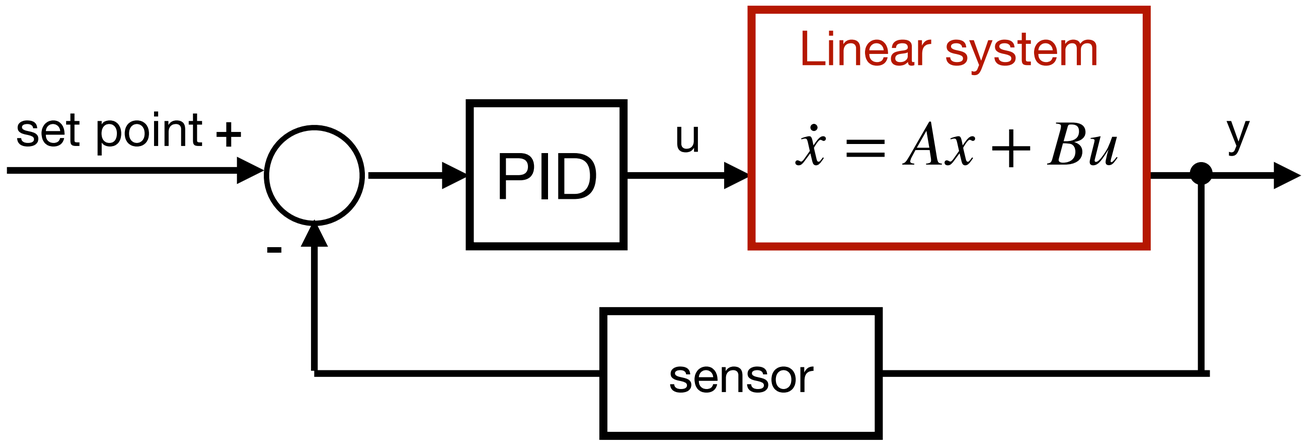
$$F = ma$$

$$\tau = I\alpha$$

$$\tau = I\ddot{\theta}$$

$$u - \dot{\theta}c = I\ddot{\theta}$$

$$\ddot{\theta} = \begin{bmatrix} \frac{-c}{I} \end{bmatrix} \dot{\theta} + \begin{bmatrix} \frac{1}{I} \end{bmatrix} u$$



$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & \frac{-c}{I} \end{bmatrix}}_{A} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{I} \end{bmatrix}}_{B} u$$

PID control, angular speed

https://tinyurl.com/yc2wkckn

PID-FastRobots.ipynb

File Edit View Insert Runtime Tools Help

Share

+ Code + Text

RAM Disk

Gemini

Notebook for Designing PID controller. Minor formatting modifications from 2023, originally written by Prof. Kirstin Petersen.

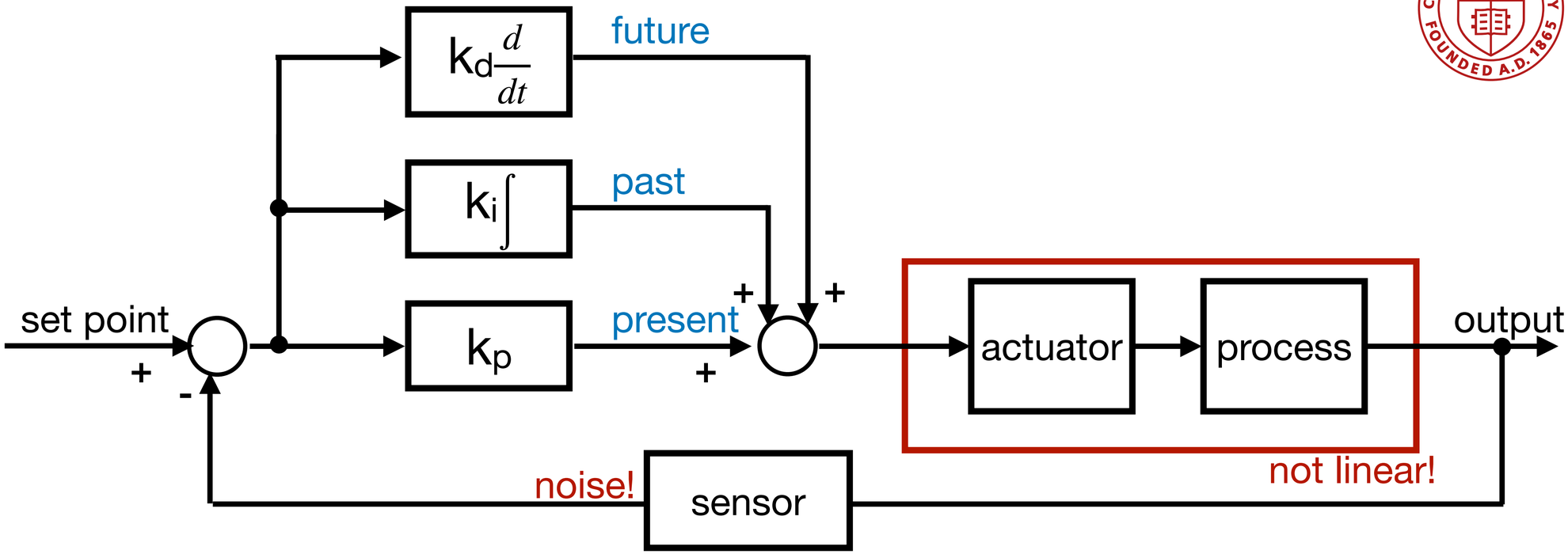
```
from matplotlib import pyplot as plt
import numpy as np

'''
ECE 4160/5160, MAE 4190/5190: Designing a PID controller
Example: I=1, c=0.2
'''

class System:

  def __init__(self,
               A=[[0, 1],[0,-0.2]],
               B=[0,1],
               x0=[0, 0],
               sigma = 0,
               dt=0.005):

    self.x=np.array(x0)
    self.t=0
    self.dt=dt

    self.sigma = sigma

    self.A = np.array(A)
    self.B = np.array(B)

    self.x_hist=[x0]
    self.y_hist=[0]
    self.t_hist=[self.t]
    self.e_hist=[0]

    '''
```

PID control, angular speed

https://tinyurl.com/yc2wkckn

- Heuristic procedure #1:
  - Set k_p to small value, k_d and k_i to 0
  - Increase k_d until oscillation, then decrease by a factor of 2-4
  - Increase k_p until oscillation or overshoot, decreases by a factor of 2-4
  - Increase k_i until oscillation or overshoot
  - Iterate
- Heuristic procedure #2:
  - Set k_d and k_i to 0
  - Increase k_p until oscillation, then decrease by factor of 2-4
  - Increase k_i until loss of stability, then back off
  - Increase k_d to increase performance in response to disturbance
  - Iterate

PID



- 1st order system:

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & \frac{-c}{I} \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{I} \end{bmatrix} u$$

- 2nd order system:

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \text{const} & \frac{-c}{I} \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{I} \end{bmatrix} u$$

Labs 5 and 6 PID control

- Control for fast mode
  - Stunt: orientation
  - Stunt: speed control
- Control for slow mode
  - Mapping: angular sped
  - Path execution: position control

Biggest limitation:

- Sensor noise
- Sensor sampling time
- PID control is 5-10x faster than system
- Lab 7 kalman filter
- Lab 8 stunt

Next three lectures

Control theory

- Linear systems
- Eigenvectors
- Stability
- Controllability
- Observability
- Kalman filters

$$\dot{x} = Ax + Bu$$

These should look familiar from:

- MATH2940 Linear Algebra
- ECE3250 Signals and Systems
- ECE5210 Theory of Linear Systems
- MAE3260 System Dynamics
- and many others…

Have a good break!