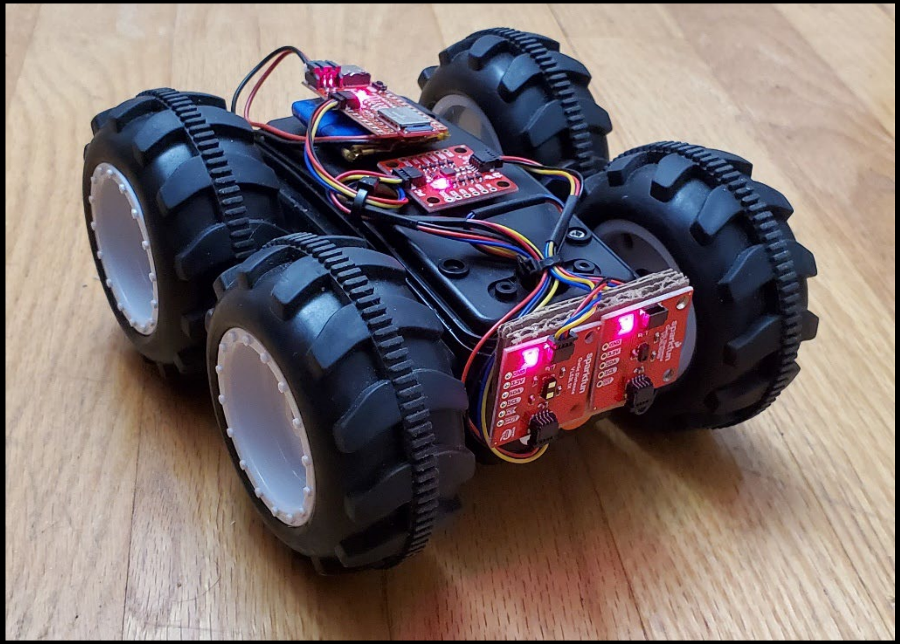
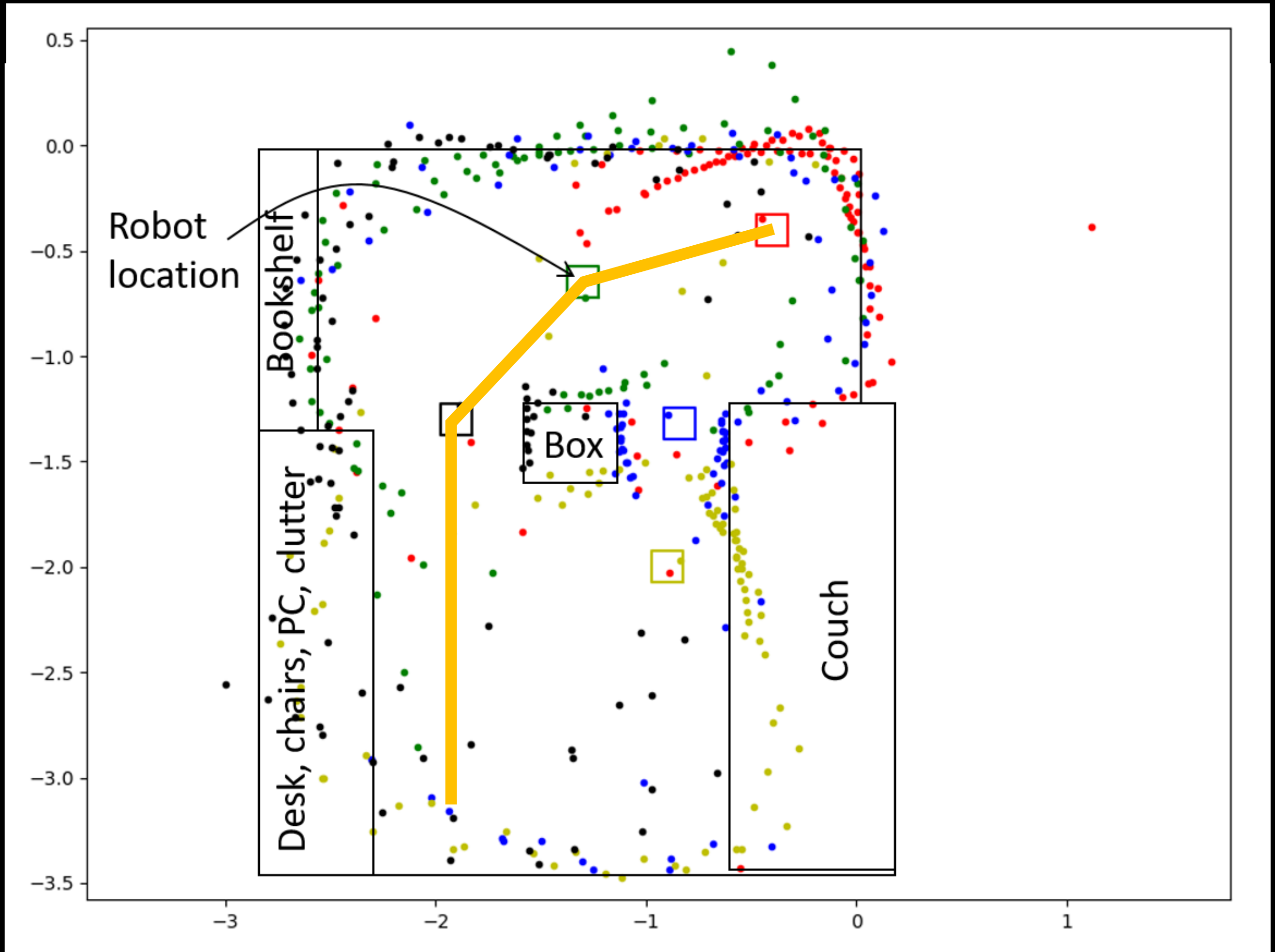
Prof. Kirstin Hagelskjær Petersen
kirstin@cornell.edu

Fast Robots

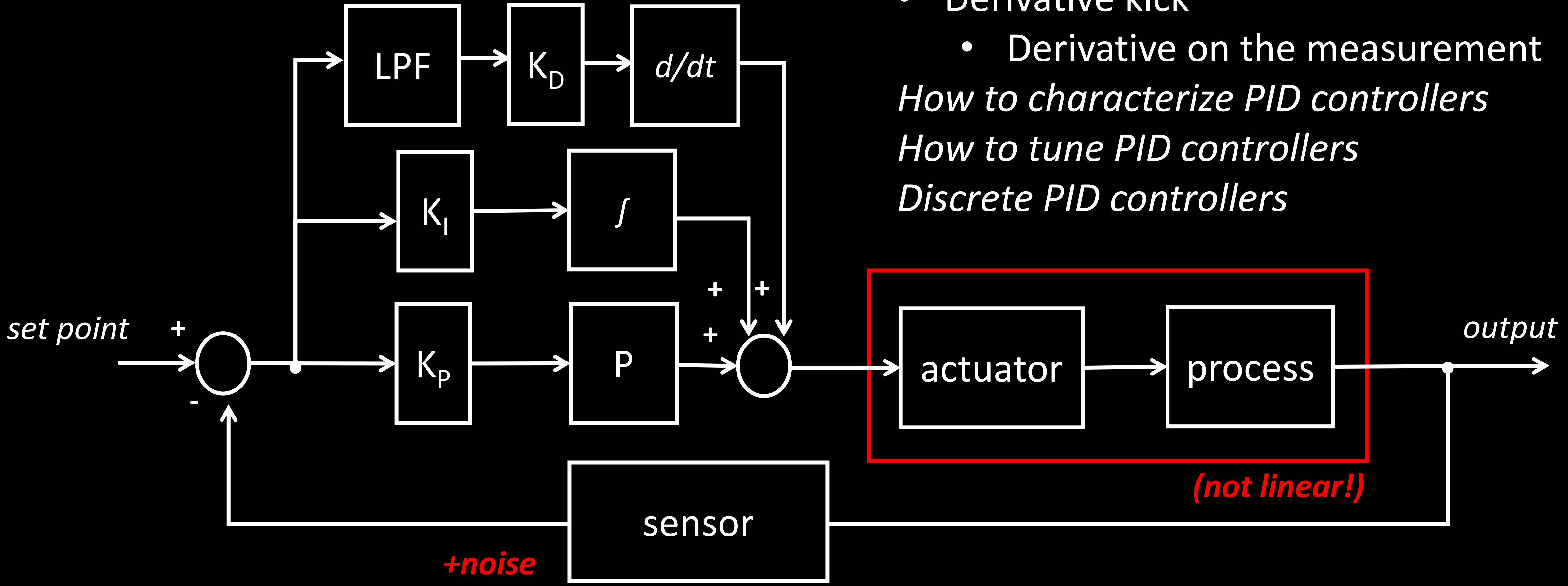
Feedback Control

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





PID



Common things to address:

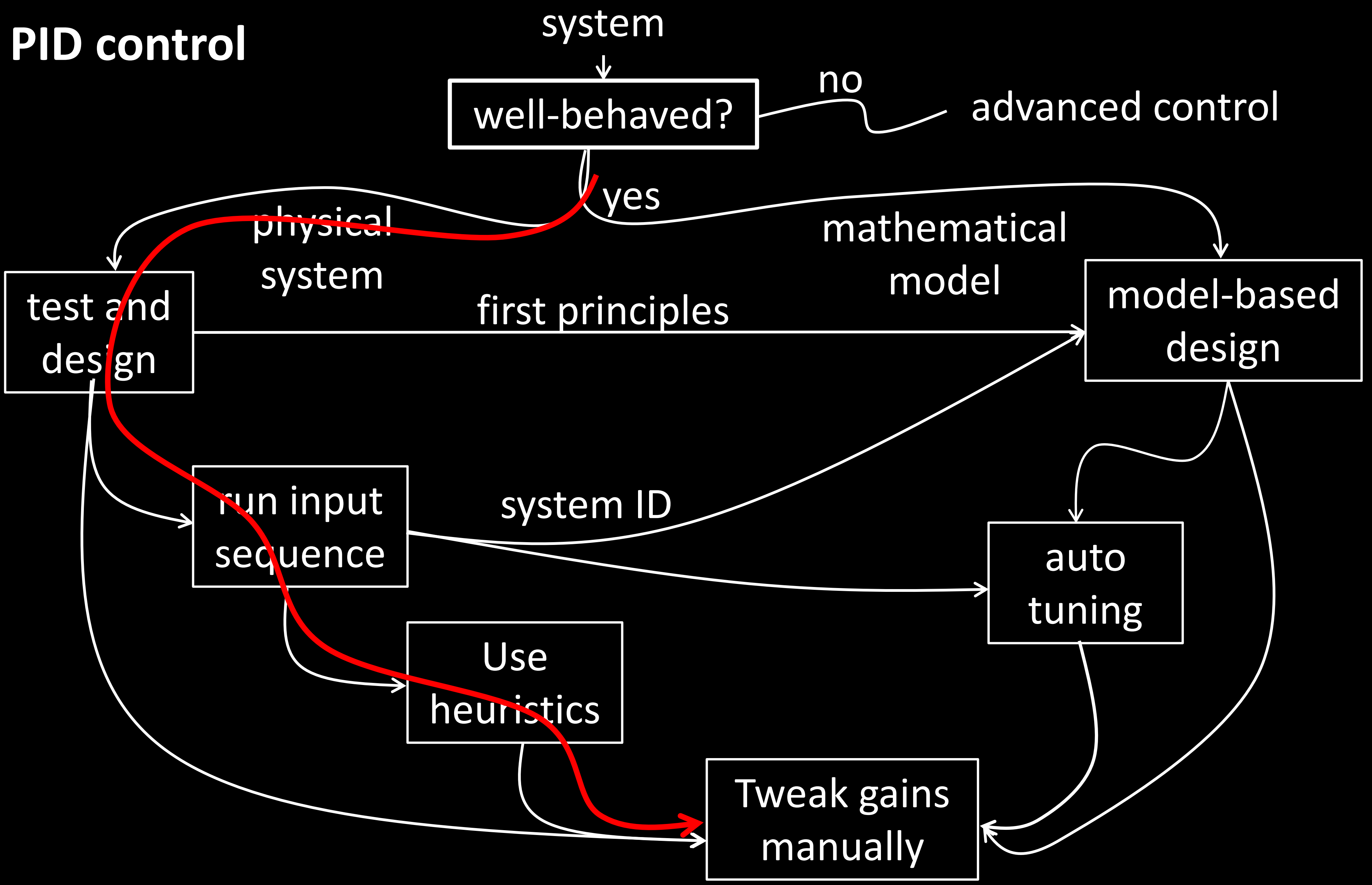
- Integrator wind-up
- Derivative low pass filter
- Derivative kick
  - Derivative on the measurement

How to characterize PID controllers

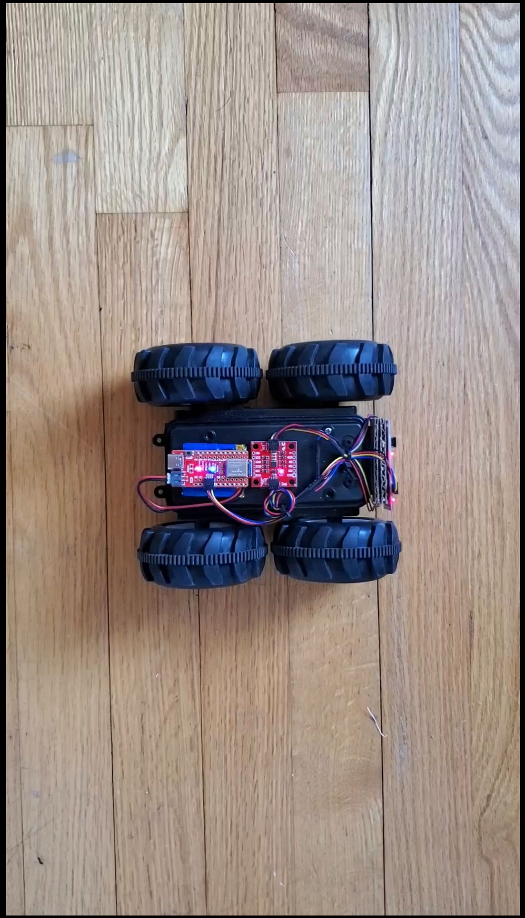
How to tune PID controllers

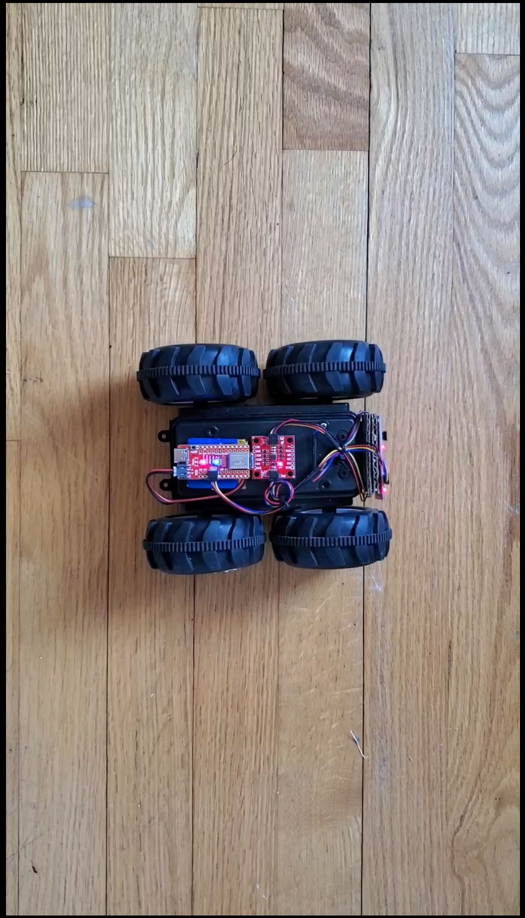
Discrete PID controllers

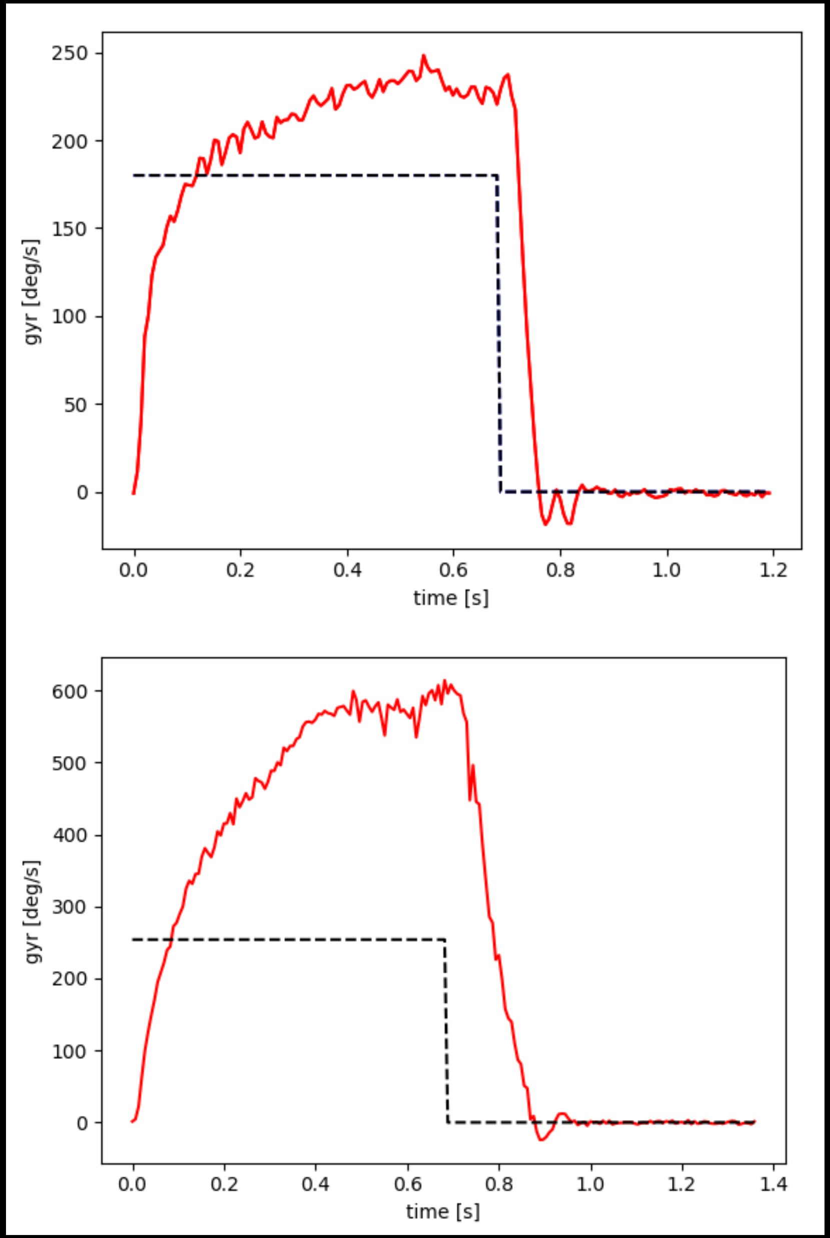
Tuning PID control



Tuning PID control







Tuning PID control

- Chien, Hornes, and Reswick method

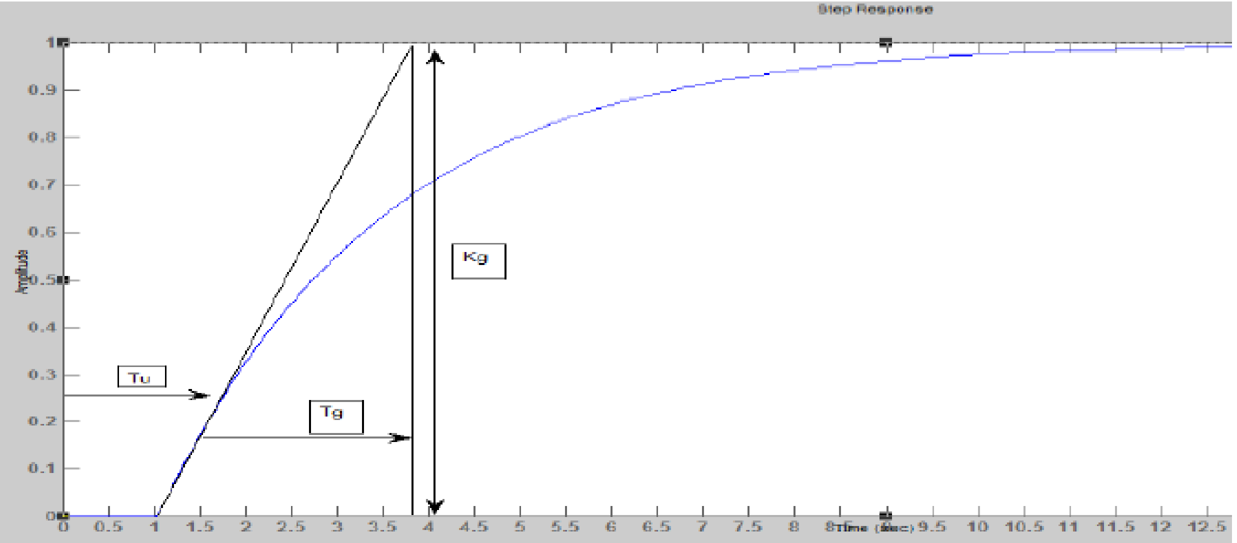
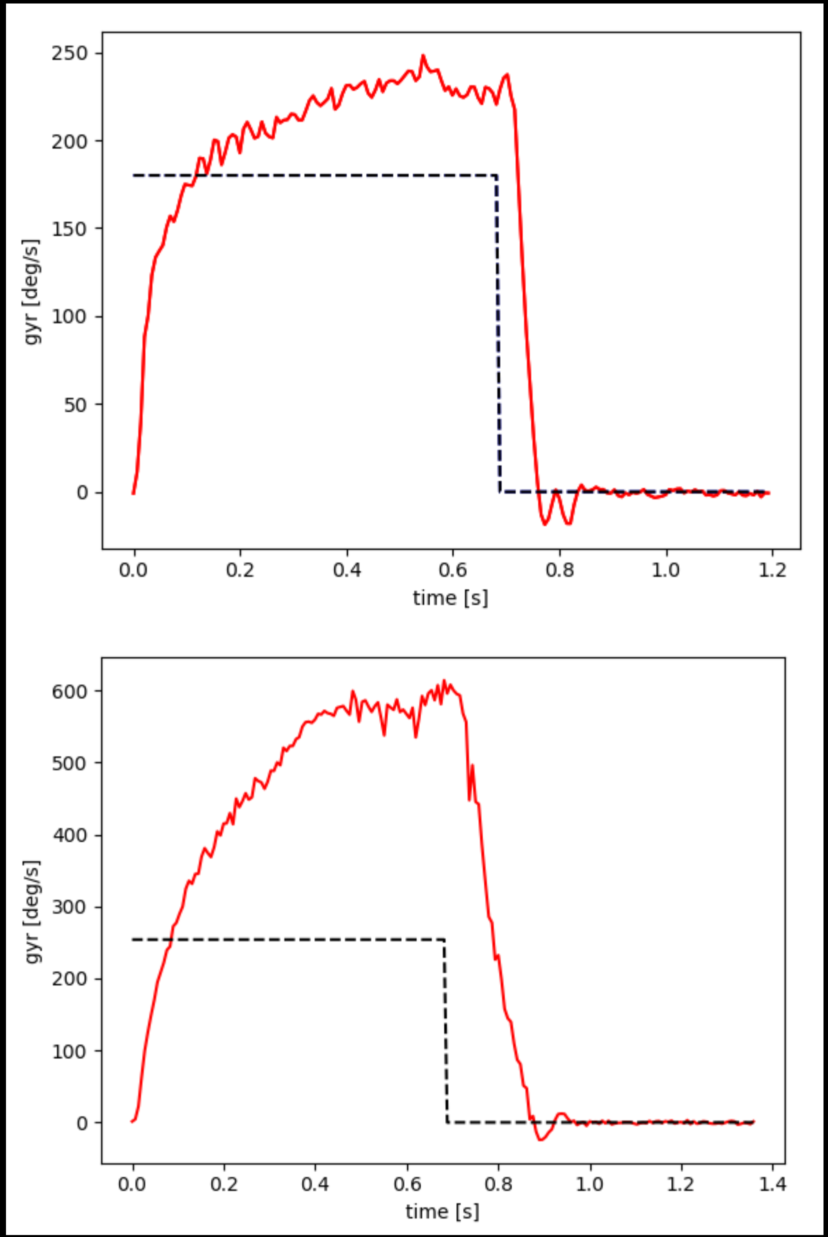


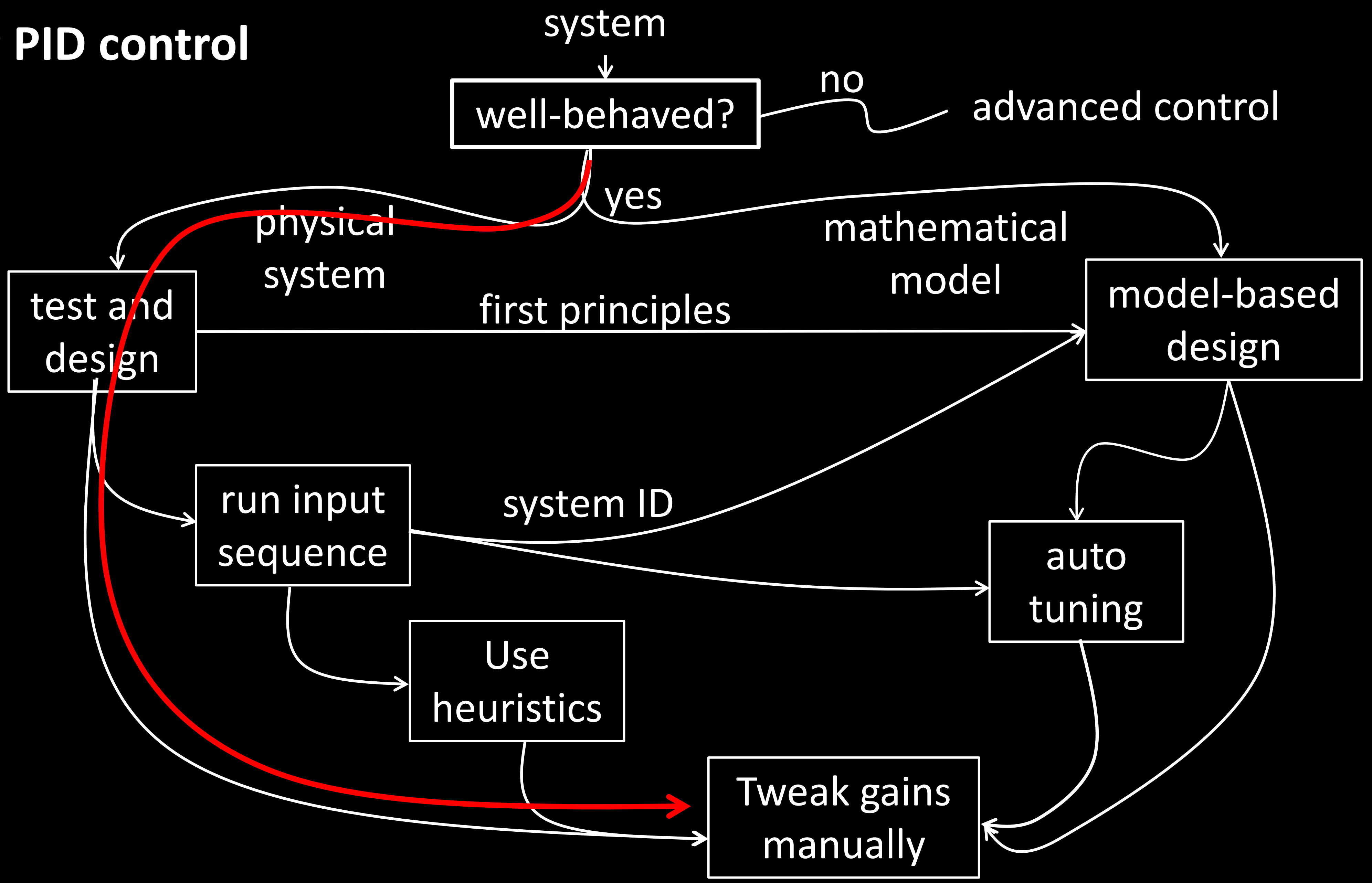
Fig.7. Open loop response of CHR method

Table.11. CHR Compensator

Type of controller	K_p	T_i	T_d
PID	$0.6T_g/T_uKg$	T_g	$0.5T_u$



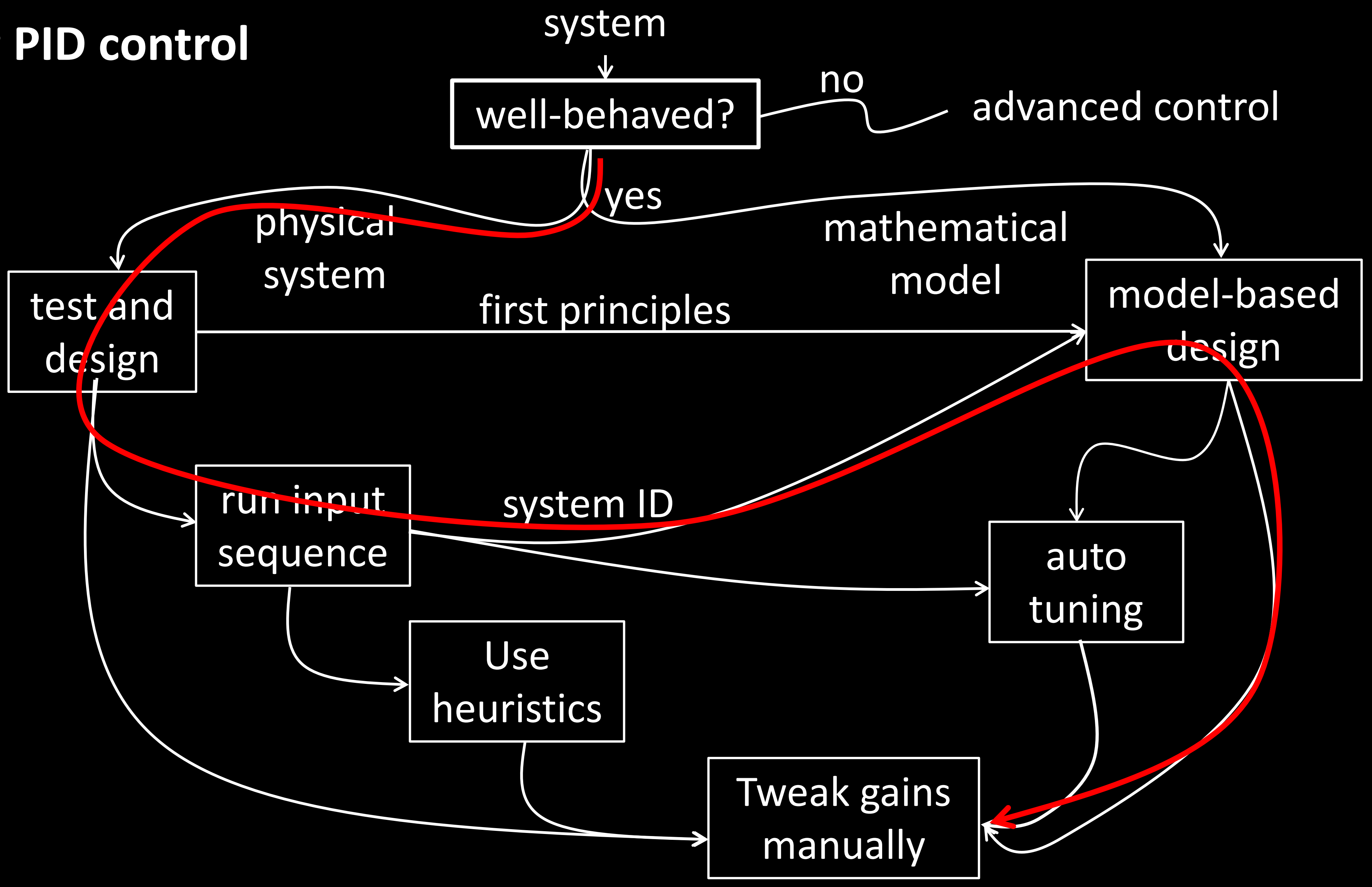
Tuning PID control



PID control

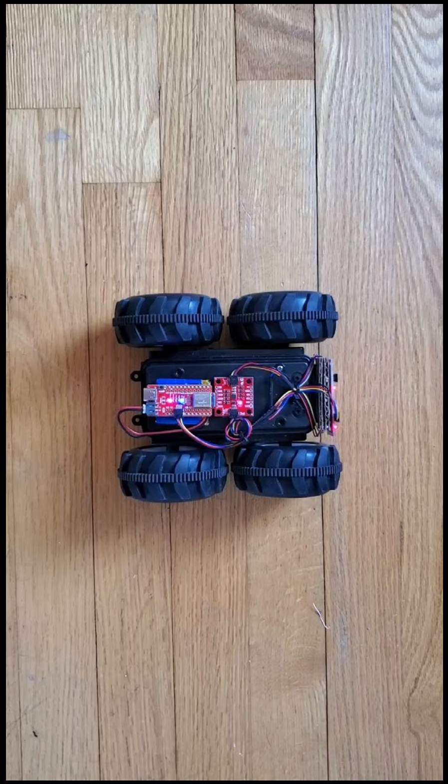
- **Heuristic procedure #1:**
  - Set Kp to small value, KD and KI to 0
  - Increase KD until oscillation, then decrease by factor of 2-4
  - Increase KP until oscillation or overshoot, decrease by factor of 2-4
  - Increase KI until oscillation or overshoot
  - Iterate
- **Heuristic procedure #2:**
  - Set KD and KI to 0
  - Increase KP until oscillation, then decrease by factor of 2-4
  - Increase KI until loss of stability, then back off
  - Increase KD to increase performance in response to disturbance
  - Iterate

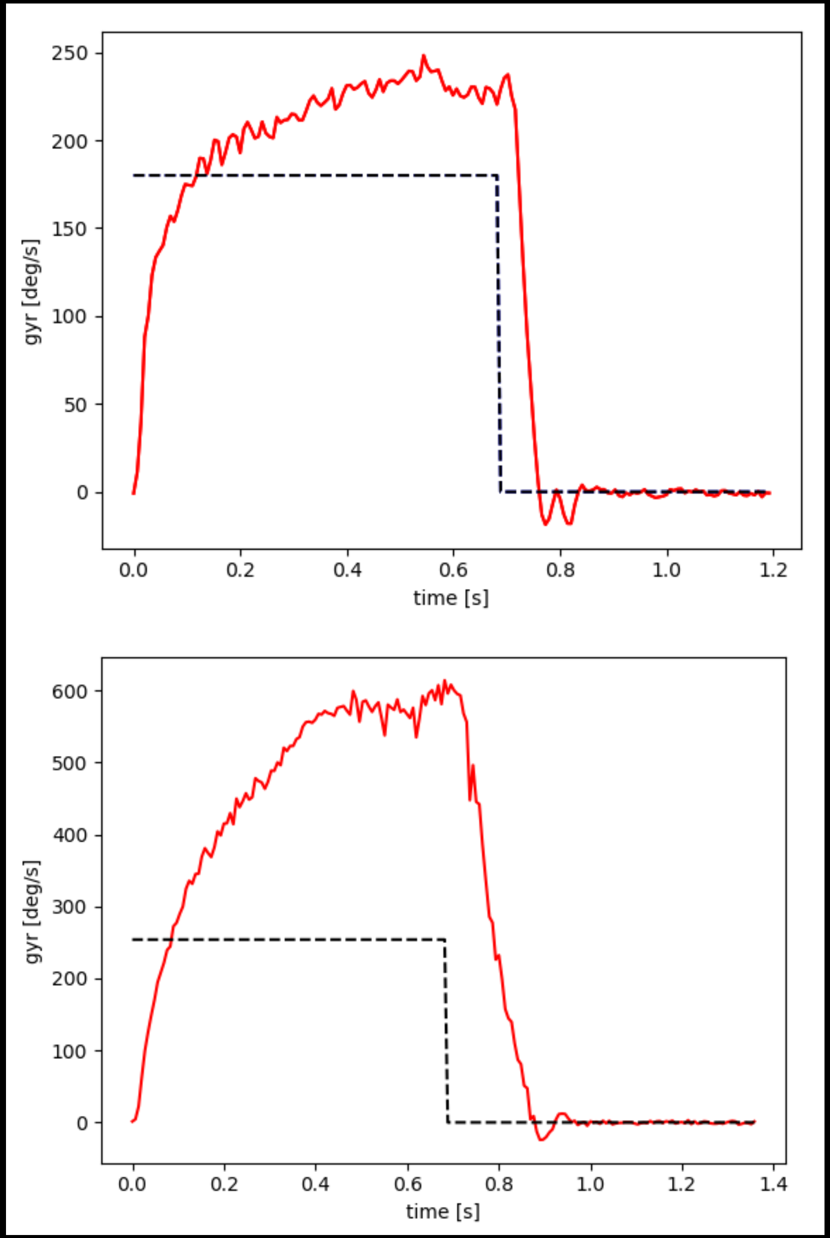
Tuning PID control



Tuning PID control

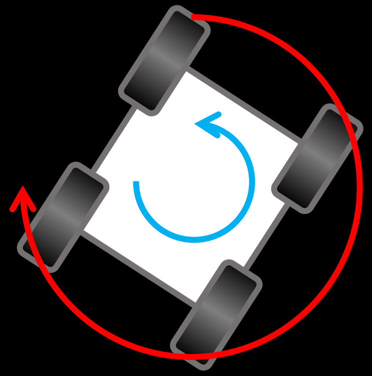
- Equations of motion
  - First order system…





PID control for constant angular speed, $\dot{\theta}$

- Equations of motion
  - $x = \dot{\theta}$
  - $x = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}$



$$F = ma$$

$$\tau = I\alpha$$

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

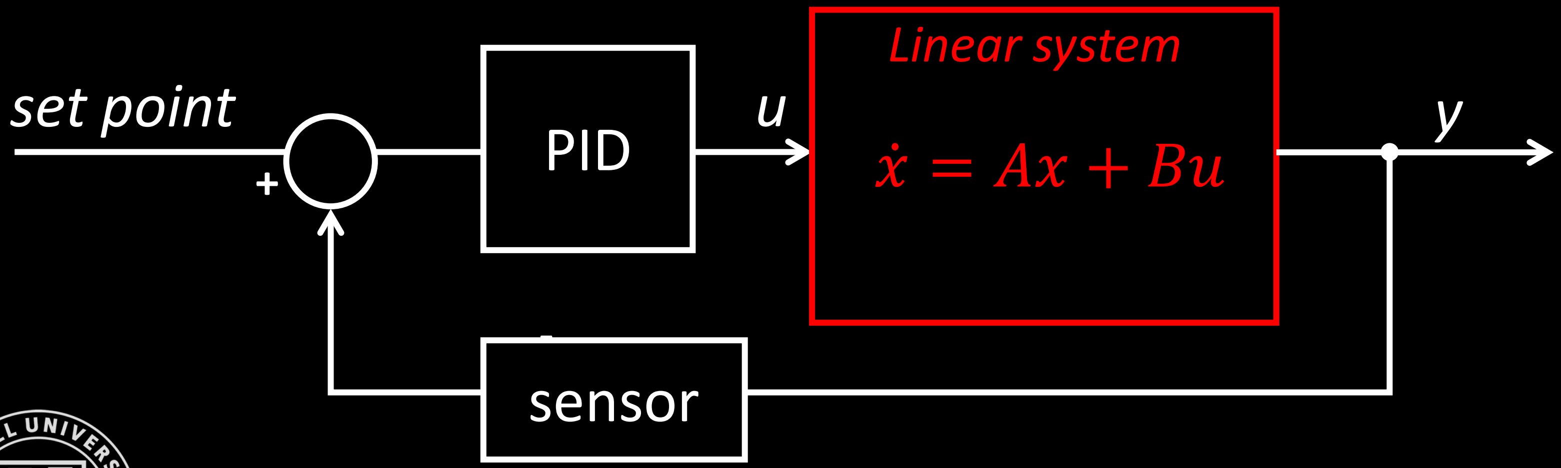
$$u - \dot{\theta}c = I\ddot{\theta}$$

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

$$[\ddot{\theta}] = \left[\frac{-c}{I}\right][\dot{\theta}] + \left[\frac{1}{I}\right]u$$

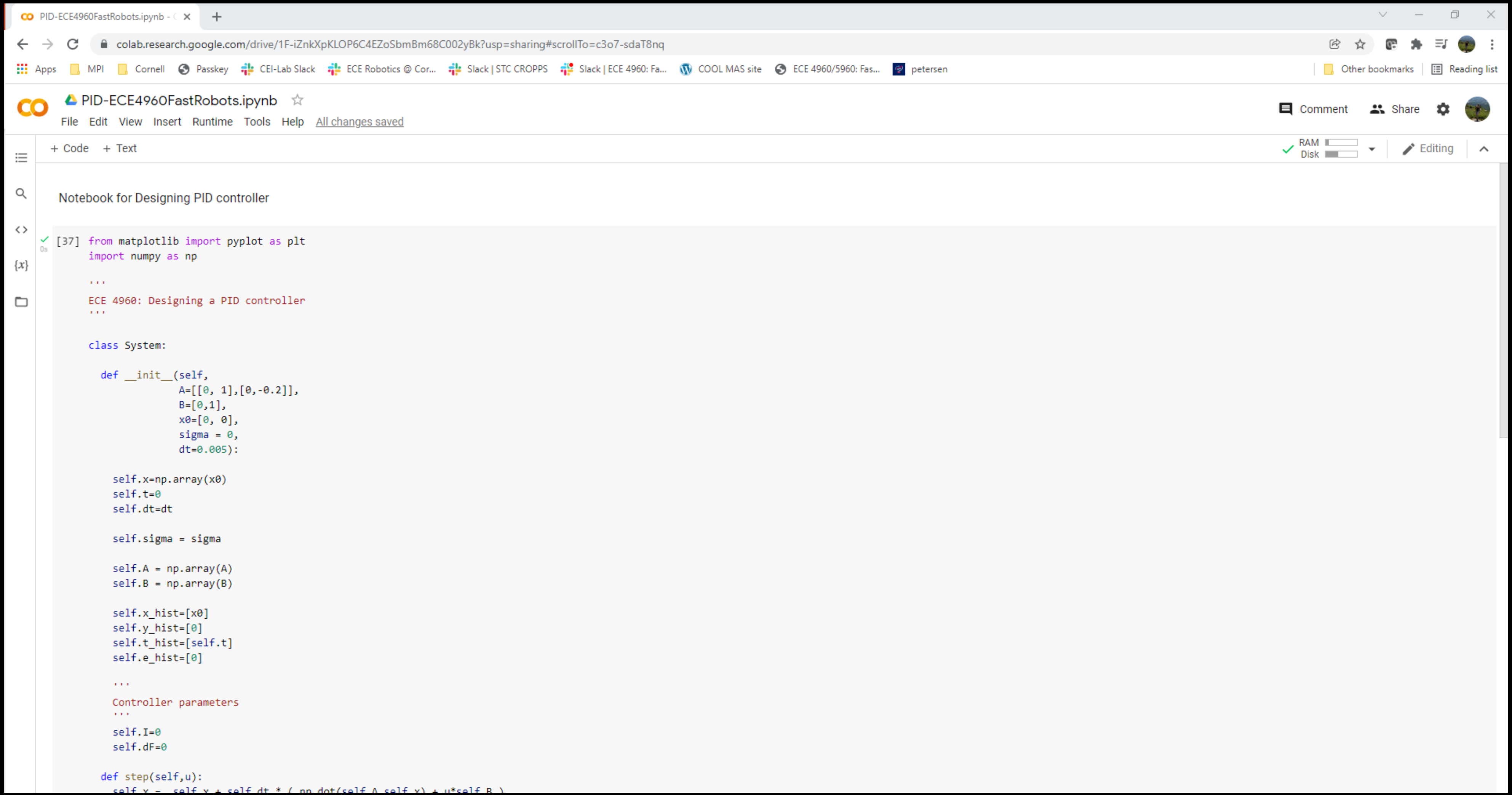
$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & \frac{-c}{I} \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{I} \end{bmatrix} u$$

A B



PID control for constant angular speed, $\dot{\theta}$

- https://bit.ly/3LIAxae




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

'''
ECE 4960: Designing a PID controller
'''

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]

    '''
    Controller parameters
    '''
    self.I=0
    self.dF=0

  def step(self,u):
    self.x = self.x + self.dt * ( np.dot(self.A,self.x) + u*self.B )
```

PID control for constant angular speed, $\dot{\theta}$

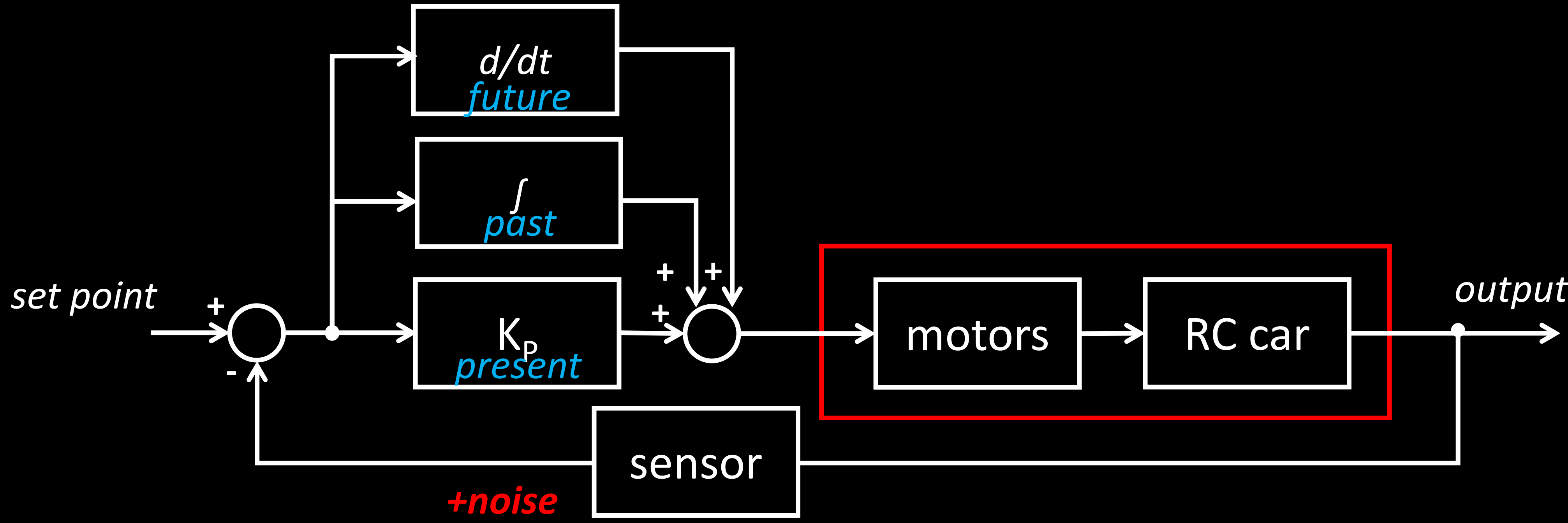
- https://bit.ly/3LIAxae

- **Heuristic procedure #1:**
  - Set Kp to small value, KD and KI to 0
  - Increase KD until oscillation, then decrease by factor of 2-4
  - Increase KP until oscillation or overshoot, decrease by factor of 2-4
  - Increase KI until oscillation or overshoot
  - Iterate
- **Heuristic procedure #2:**
  - Set KD and KI to 0
  - Increase KP until oscillation, then decrease by factor of 2-4
  - Increase KI until loss of stability, then back off
  - Increase KD to increase performance in response to disturbance
  - Iterate

PID control for constant angular speed, $\dot{\theta}$

- https://bit.ly/3LIAxae
- Overshoot (K_p = 10, K_I = 100)
- Dampening (K_p = 10, K_I = 100, K_D = 0.8)
- Noise (sigma = 0.1)
- LPF (alpha = 0.05)
- Derivative kick (alpha = 1, sigma = 0)

PID control of a 2nd order system



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

Lab 6, PID control

- PID control on angular speed (gyroscope)
  - Lab 9 mapping (as slow as possible)
- PID control on speed (accelerometer, tof)
  - Lab 13 path execution
- PID control on distance from wall (gyroscope and tof)
  - Lab 13 path execution
- PID control on a position (tof)
- PID control on an angle (gyroscope)

Biggest limitation?

- Sensor sampling time
- PID control is preferably 5-10 times faster than your system
- *Lab 7 Kalman Filter*
- *Lab 8 Stunts*
  - *Open loop category*
  - *Closed-loop category*

Next three lectures

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

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

These should look familiar from..

- MATH 2940 Linear Algebra
- ECE3250 Signals and systems
- ECE5210 Theory of linear systems
- MAE3260 System Dynamics
- etc…