# Feedback control

## Fast Robots, ECE4160/5160, MAE 4190/5190

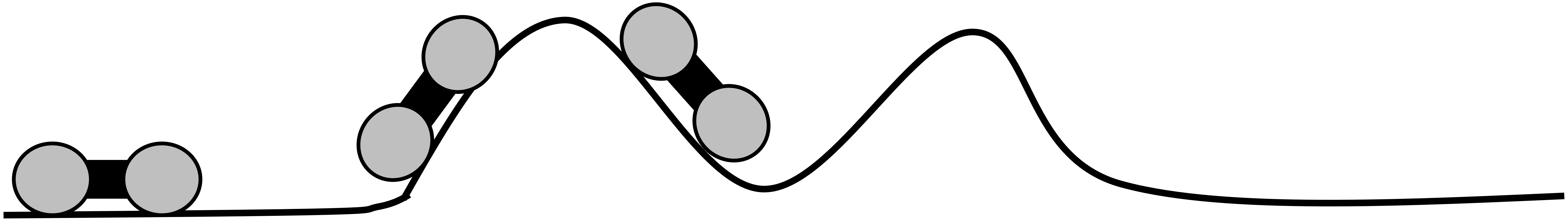
**E. Farrell Helbling, 2/10/26**
**Credit: Prof. Kirstin Petersen**

# Class Action Items

- Lab 3 starts today! Please aim to get your soldering done first!
  - Not everyone can solder at the same time (we only have 8 soldering irons)
  - If you have access to another soldering iron, feel free to use it.
- There is no lab next week and lab 3 is not due until Feb. 24-25, this is mostly due to Feb. break, but also because there is a lot of soldering to do over the next two labs. Please plan ahead!
  - Please if you can, work on lab 4 early (i.e., next week after February break)

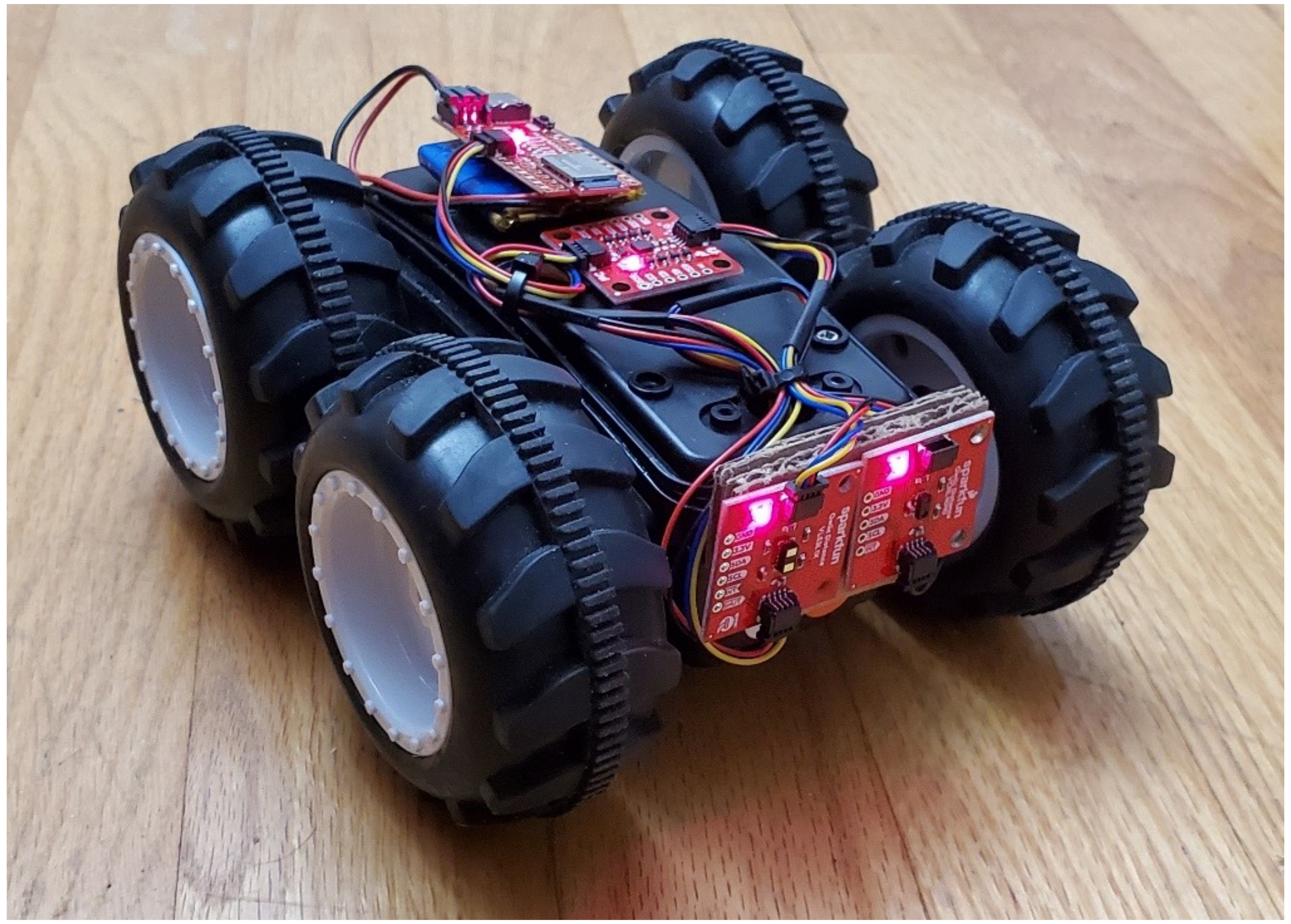
# Feedback Control

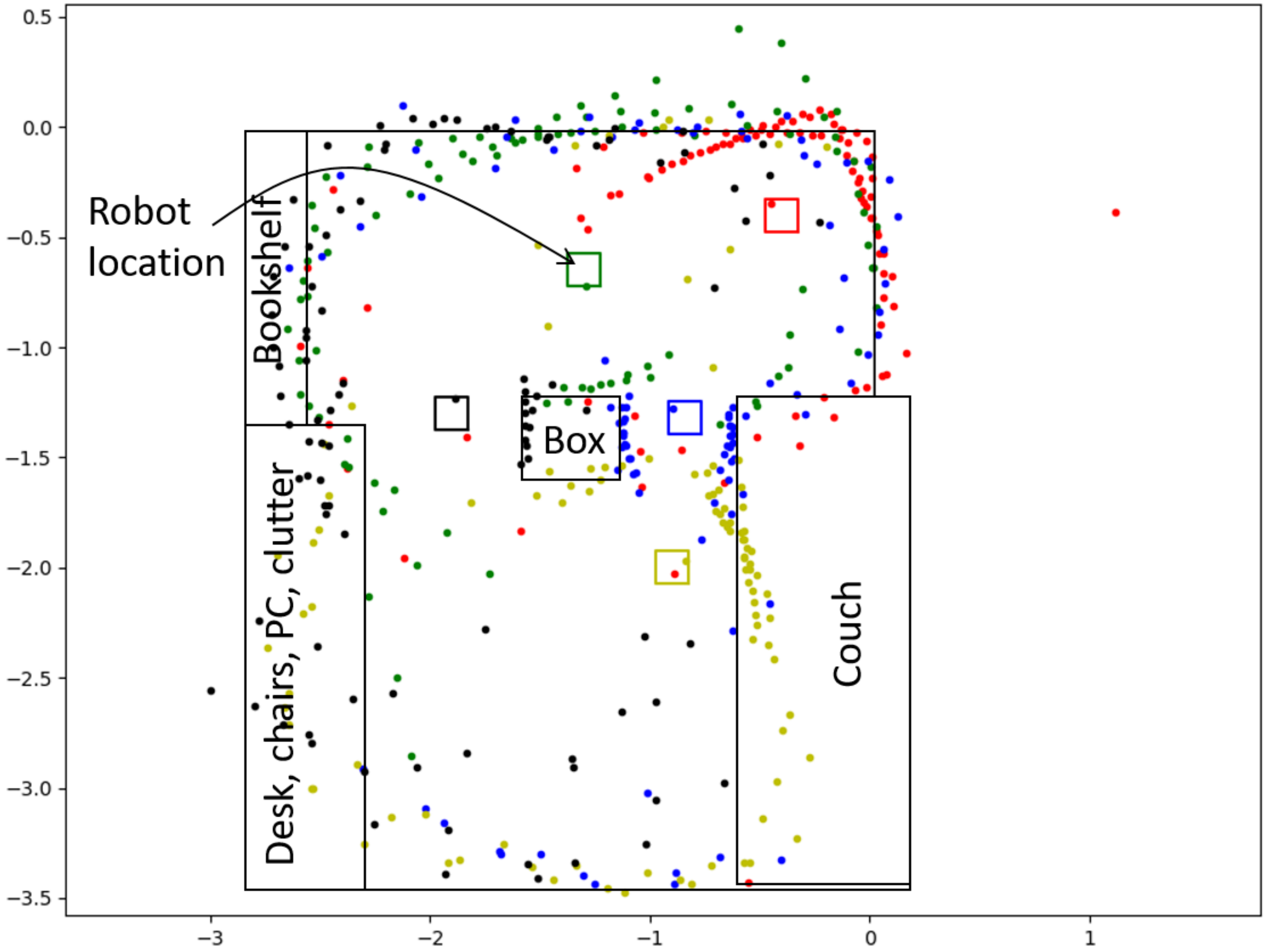
- Maintain speed prediction at different battery levels, over different surfaces
- Maintain position with respect to walls
- Maintain heading angle
- Etc.



# 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





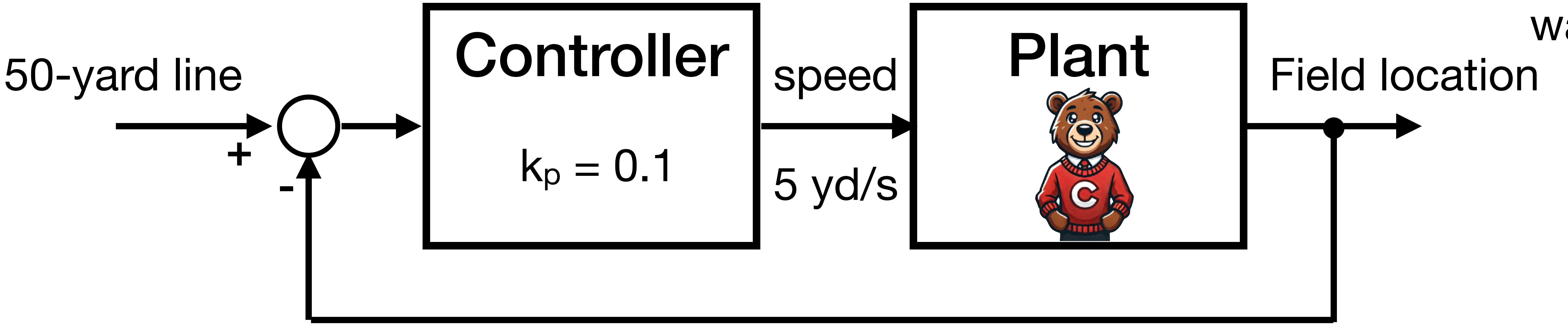
# PID control

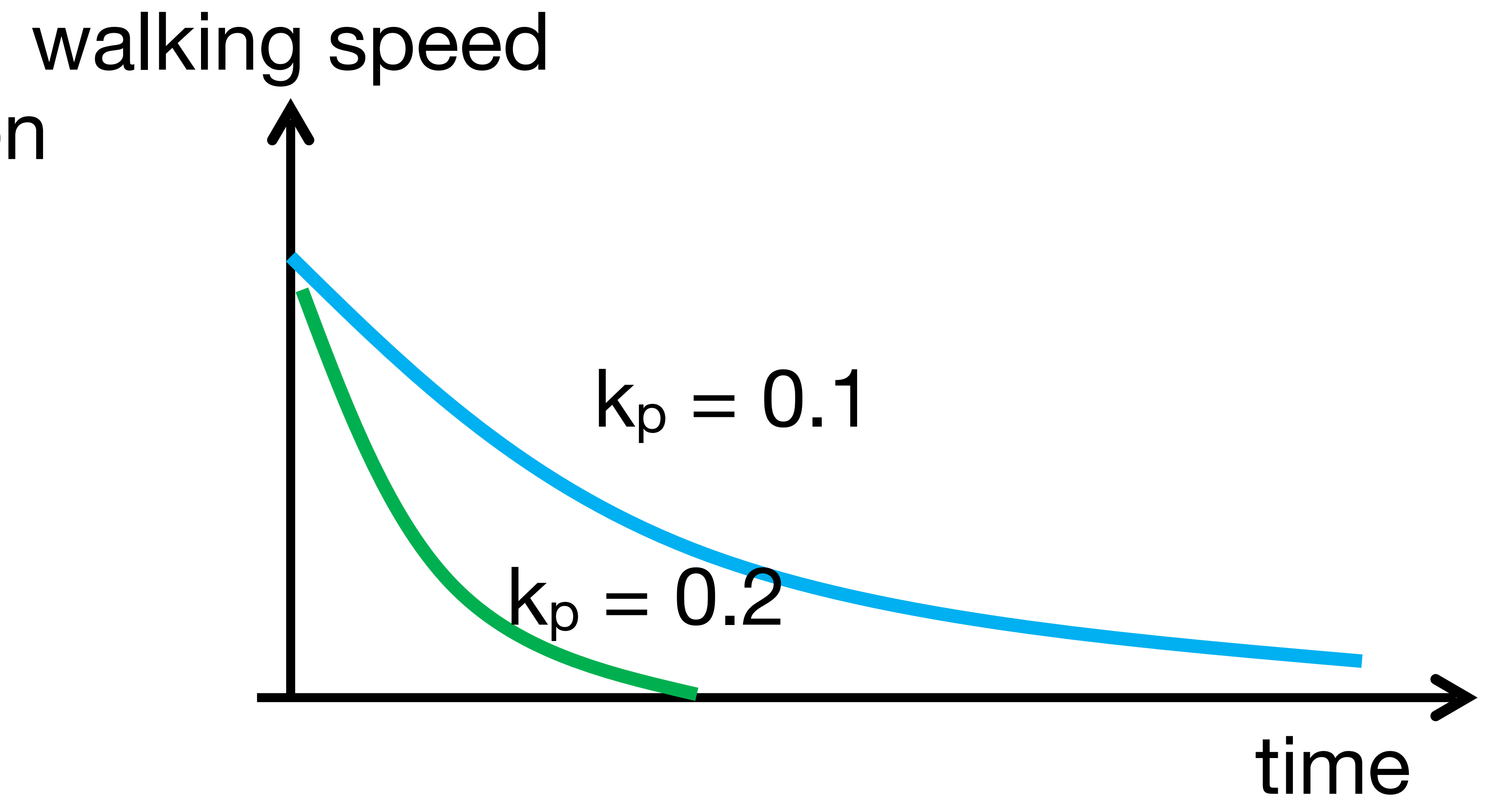
$$u(t) = K_P e(t) + K_I \int_0^t e(t)dt + K_D \frac{de(t)}{dt}$$

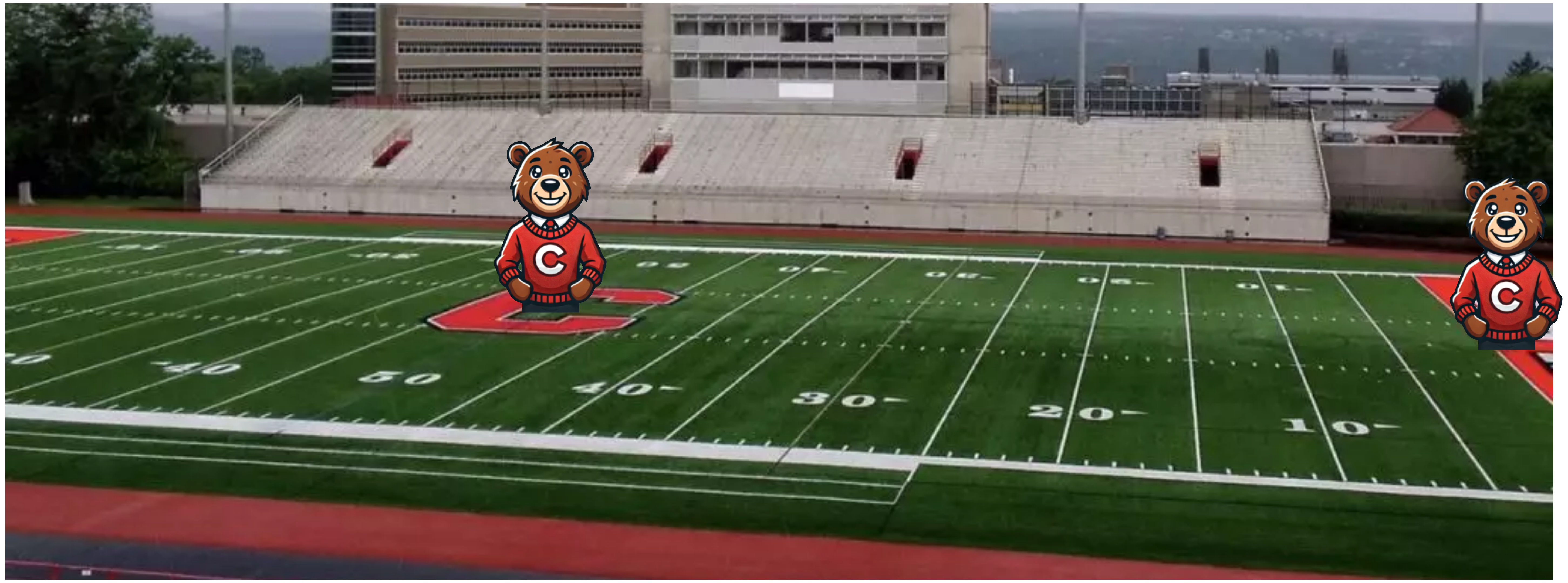

Based heavily on MATLAB's Tech Talk: Understanding PID control

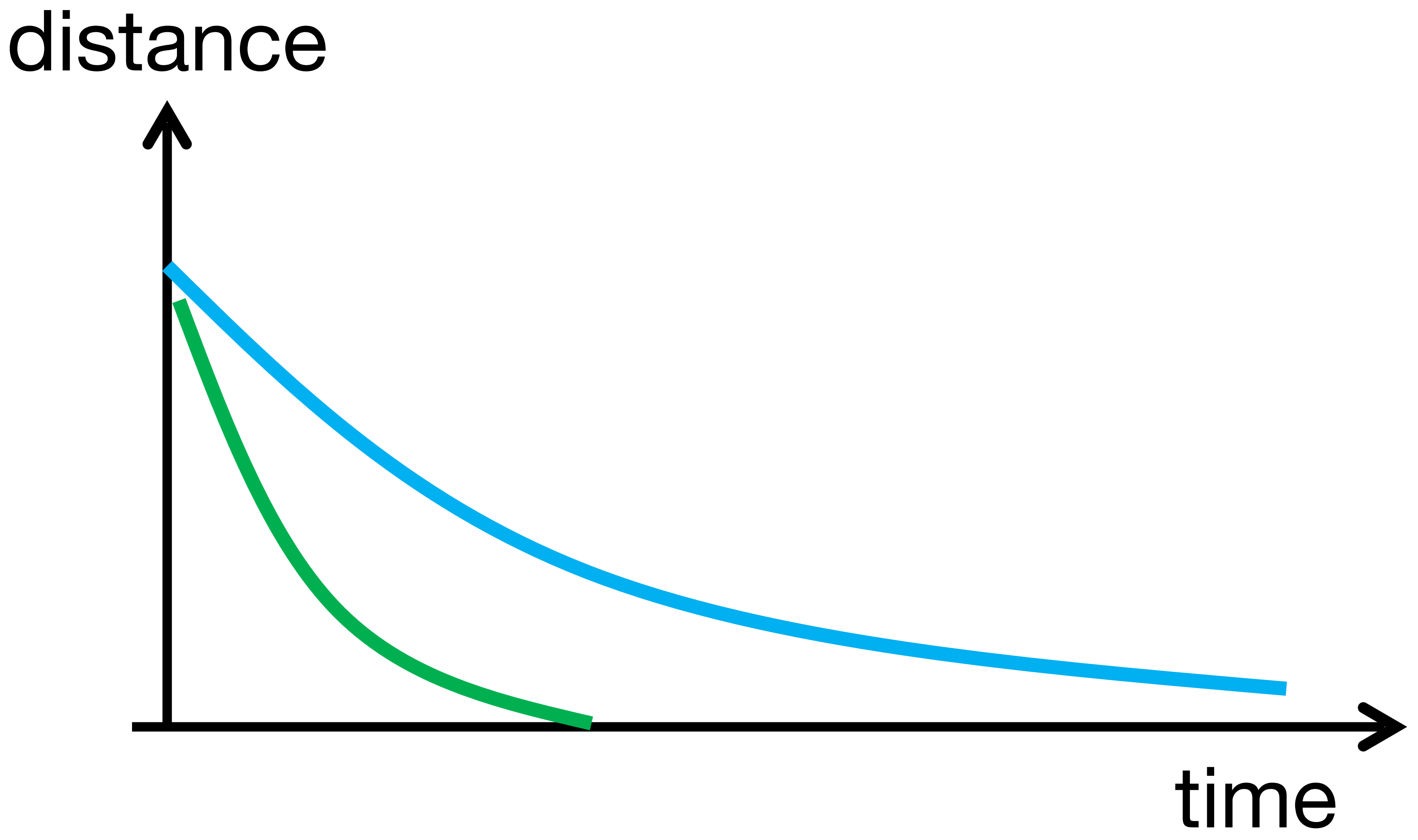
# PID control

## Football Field



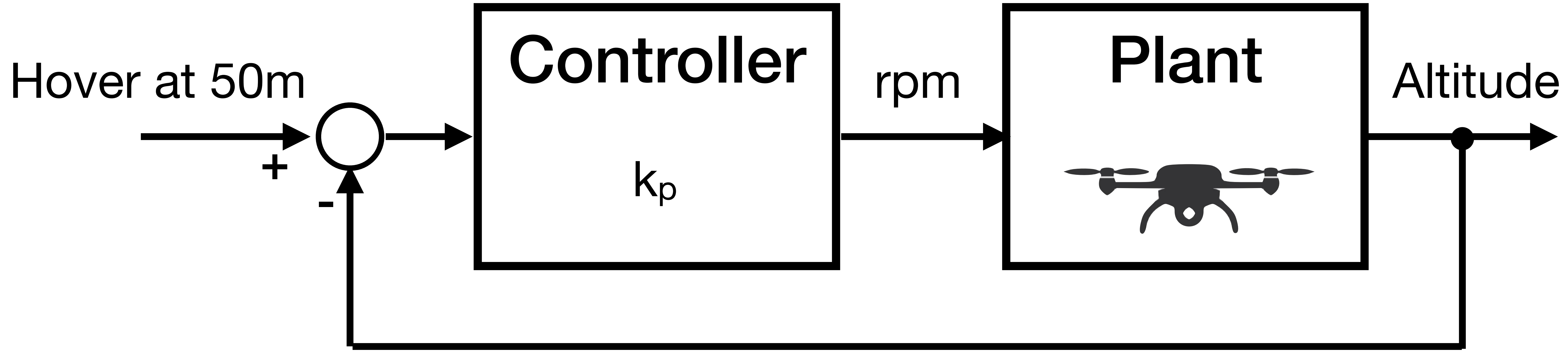



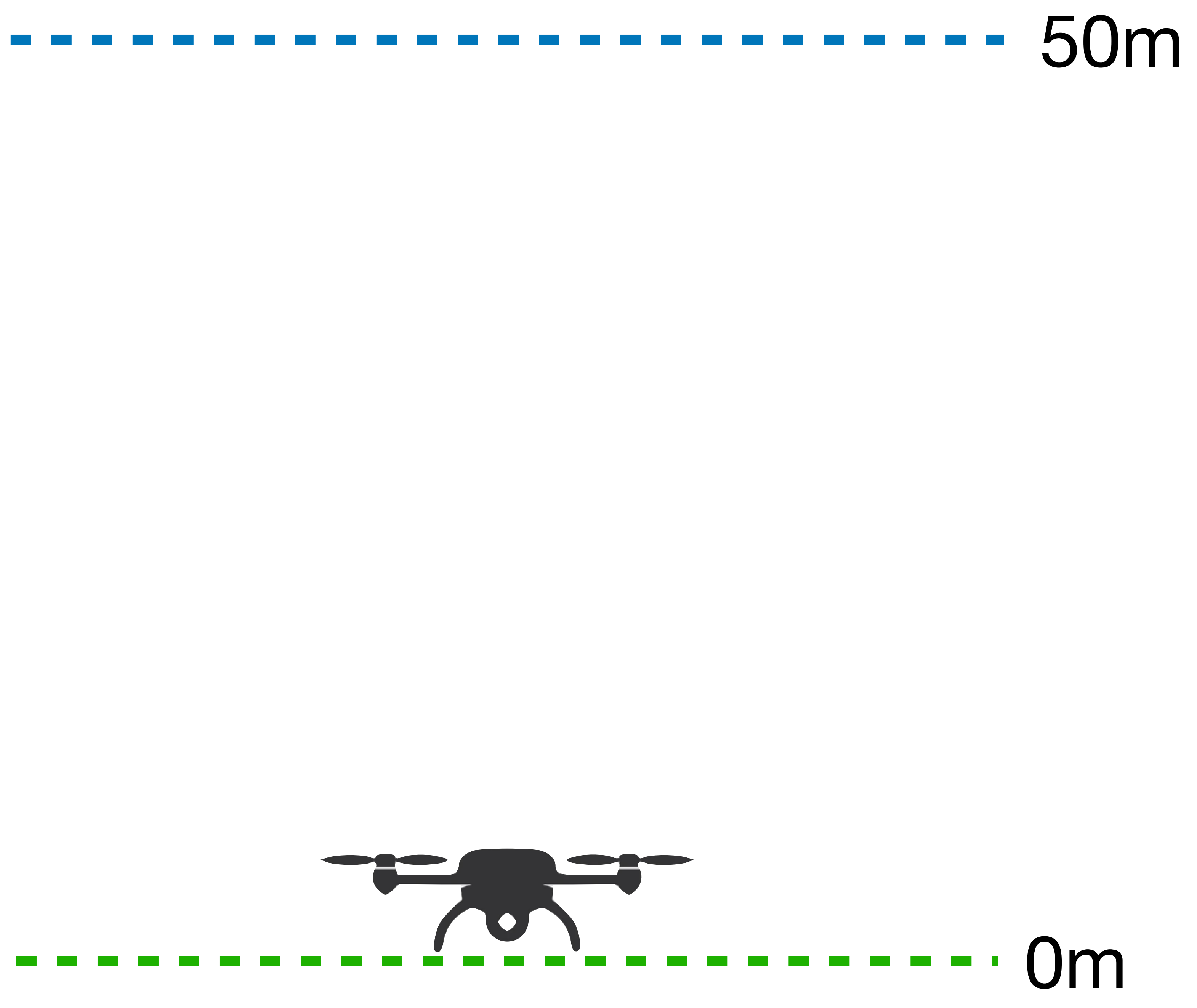





# PID control

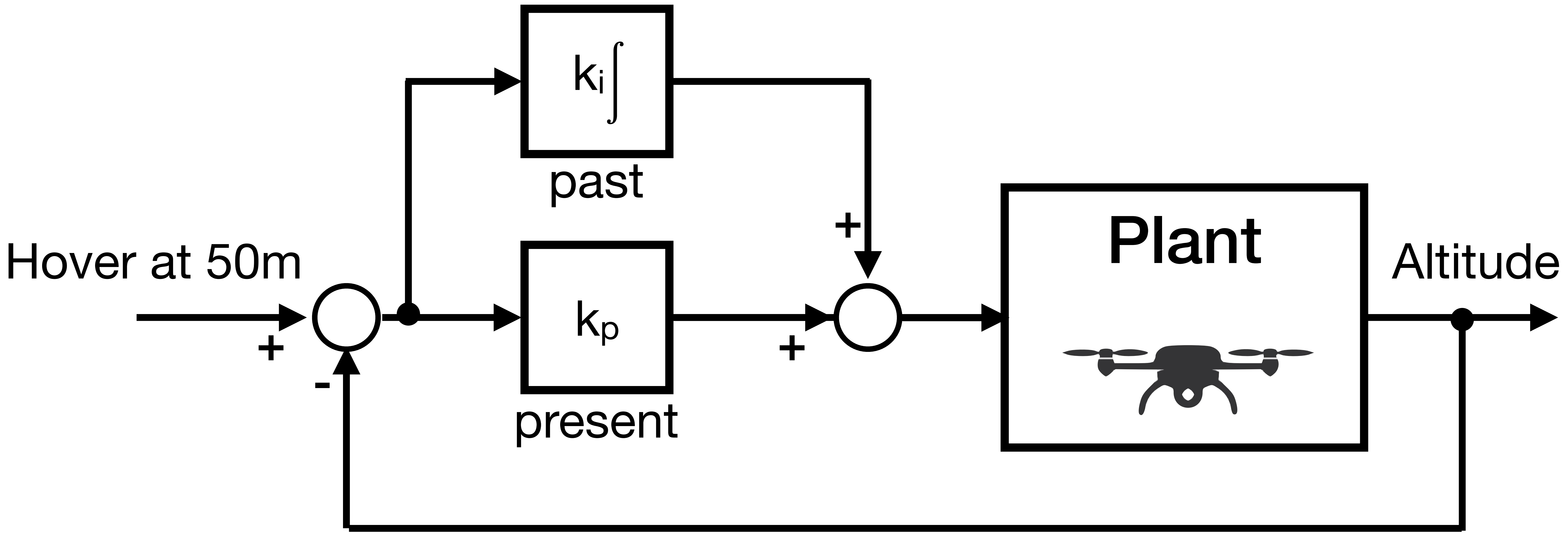
## Drone

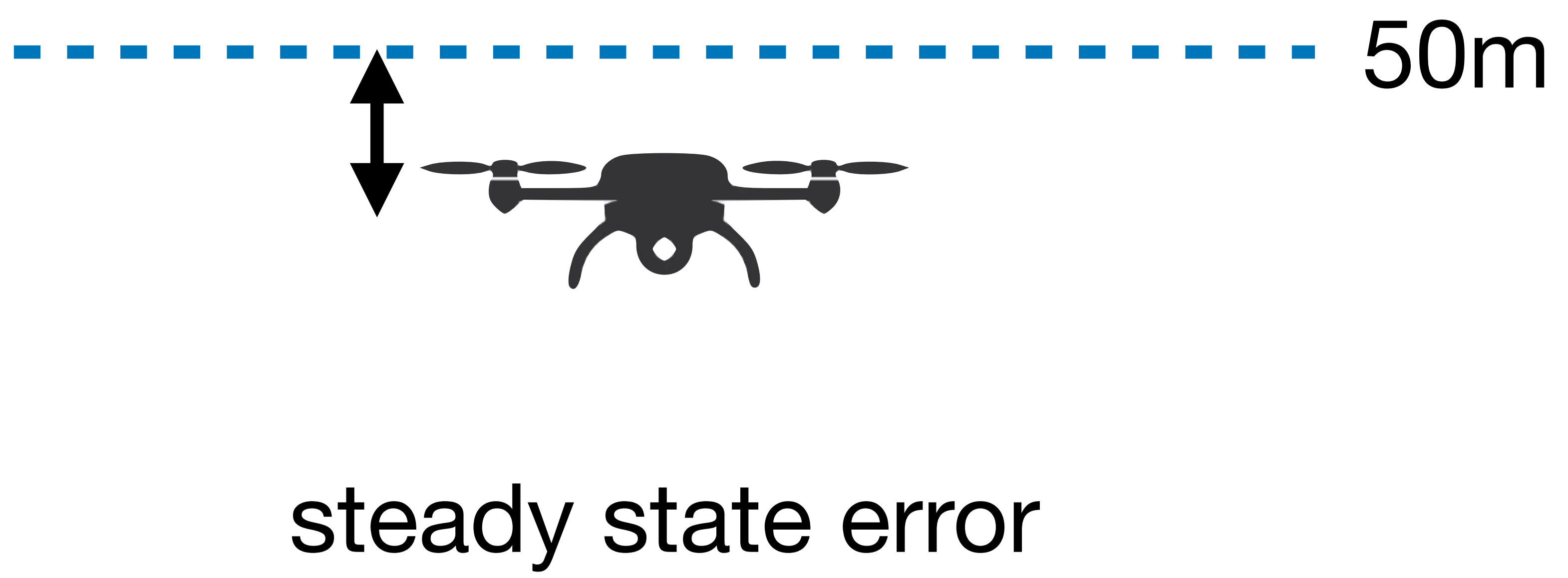


# PID control

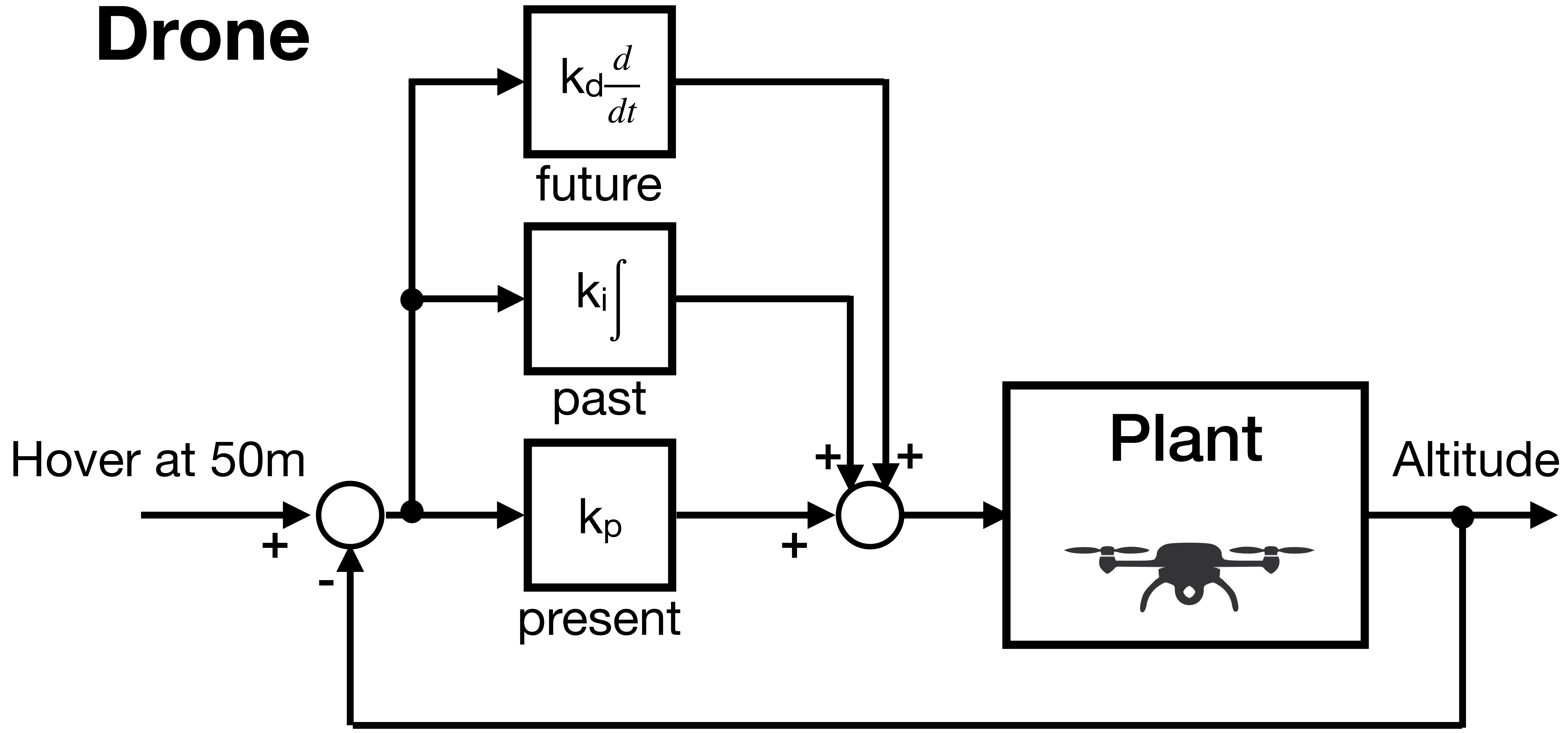
## Drone

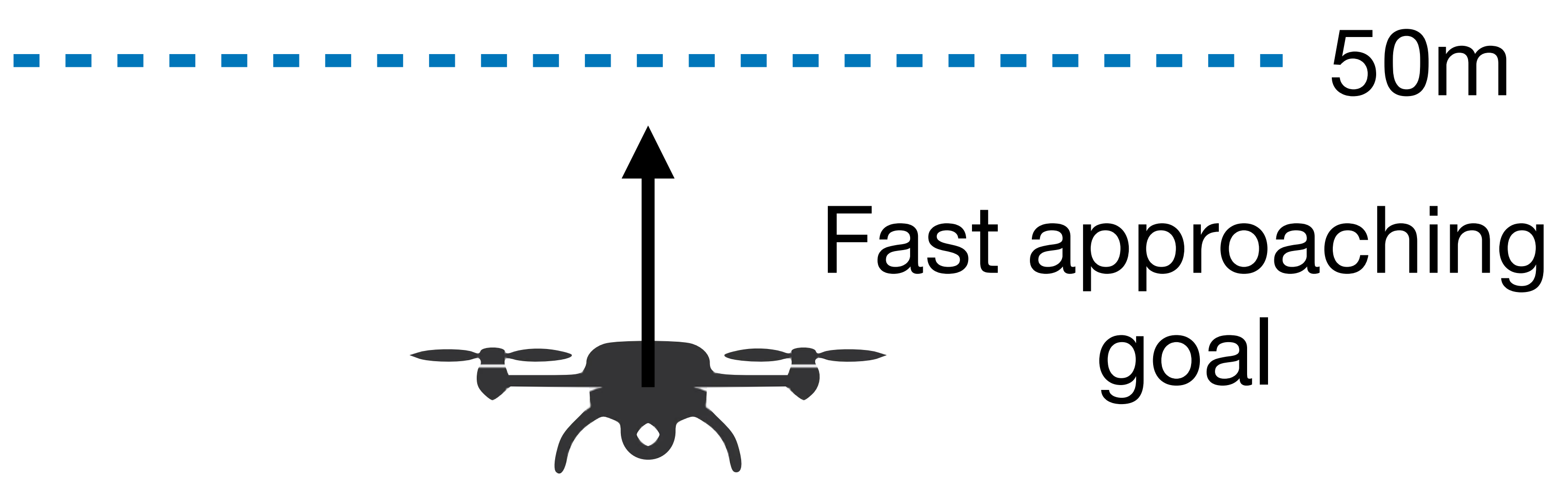


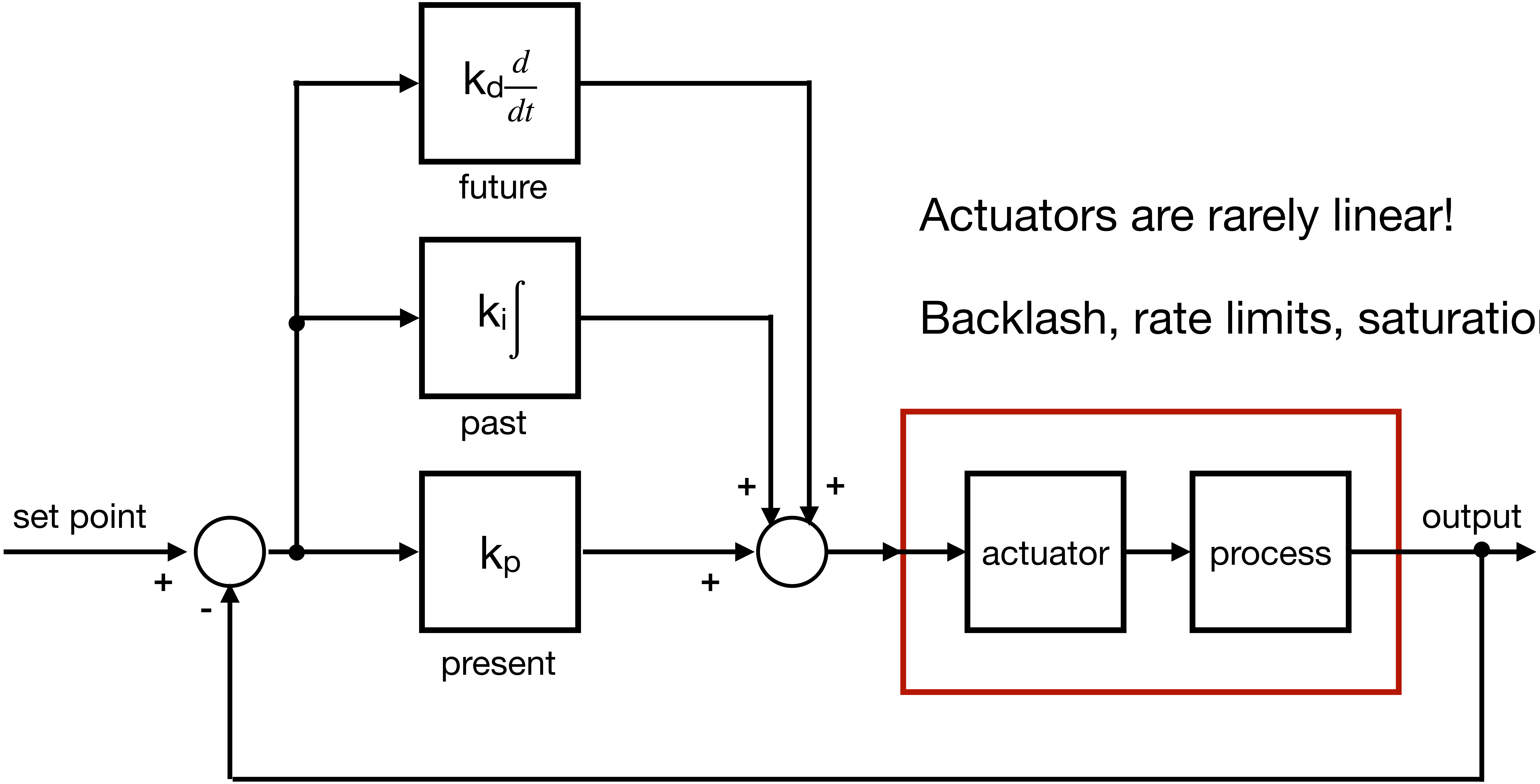
# PID control

## Drone



# PID Control

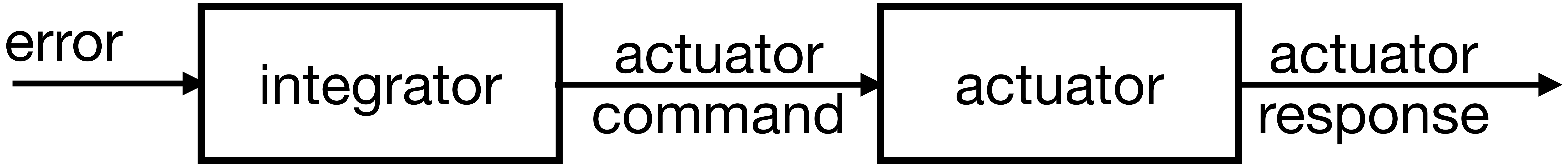


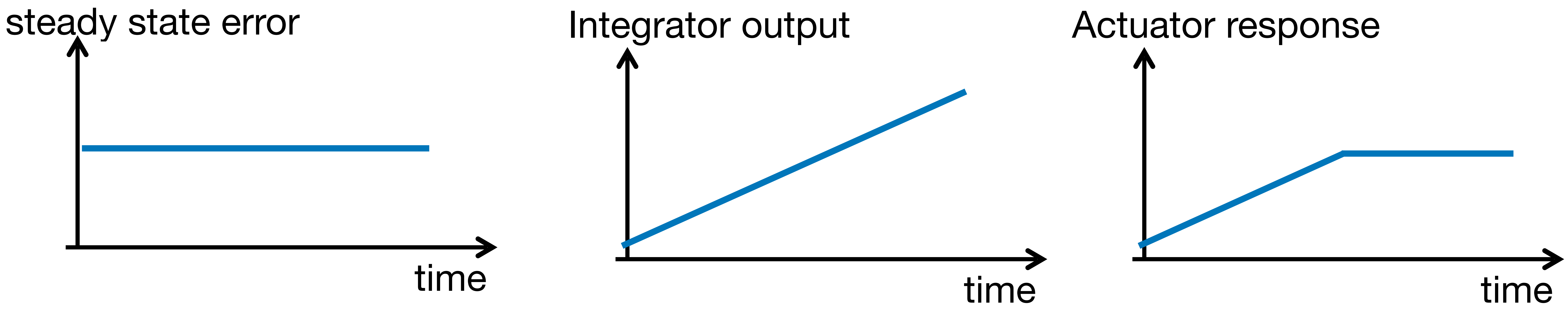

Actuators are rarely linear!

Backlash, rate limits, saturation

# Real systems are not linear

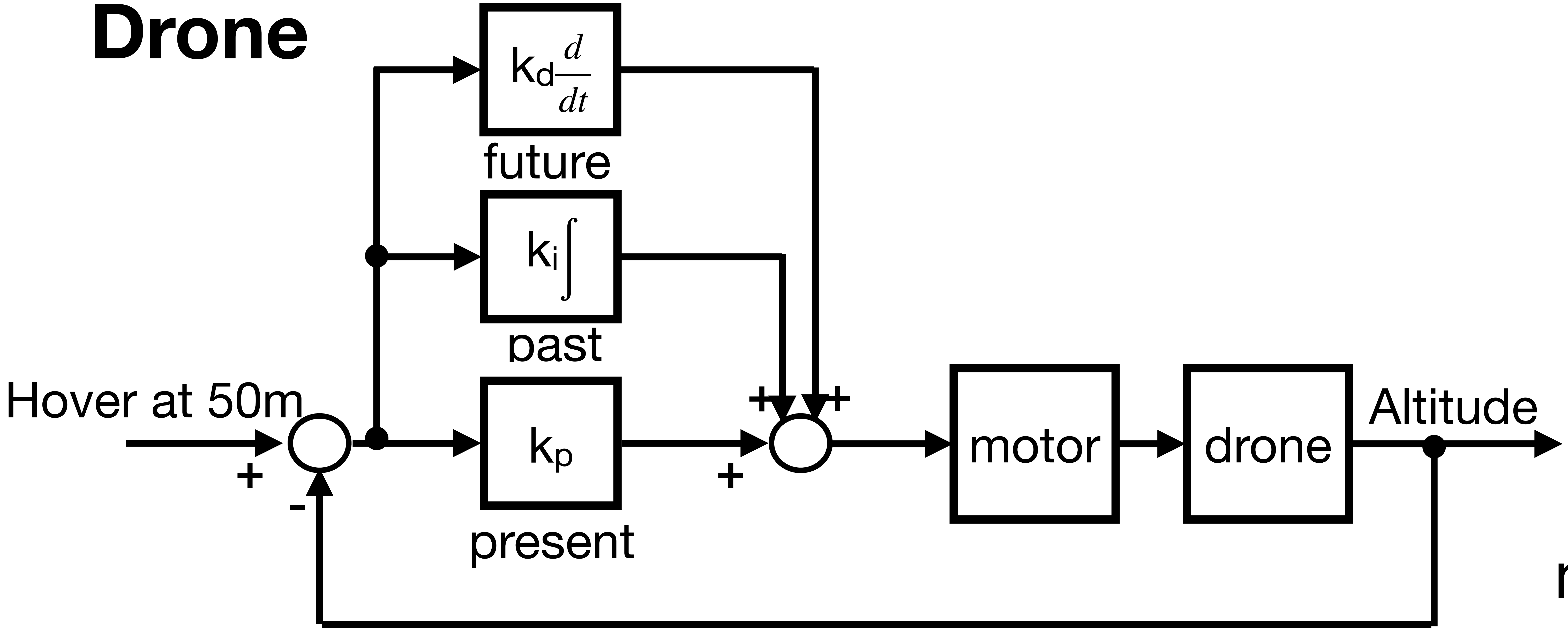
## Actuator response

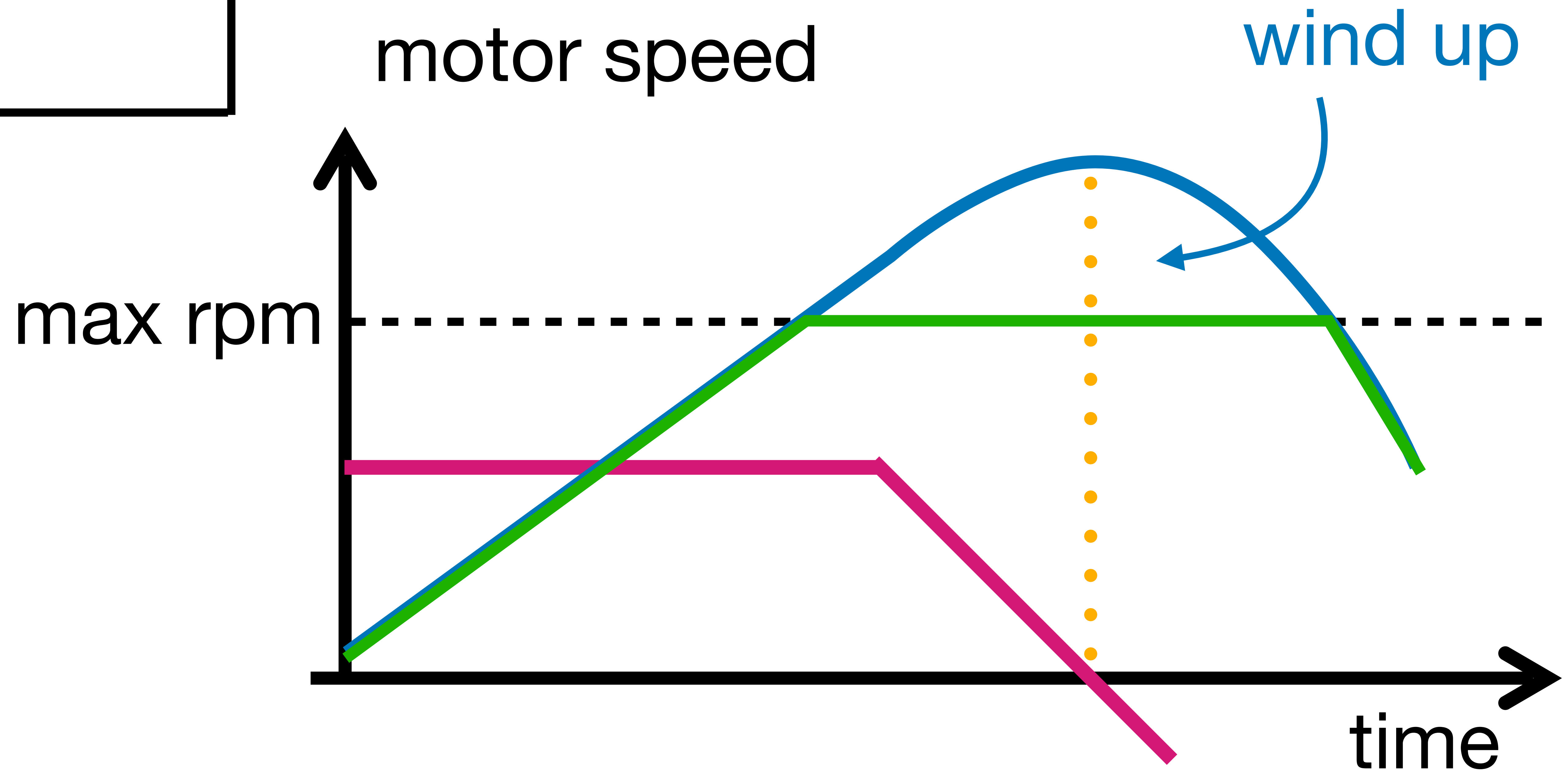


# Real systems are not linear

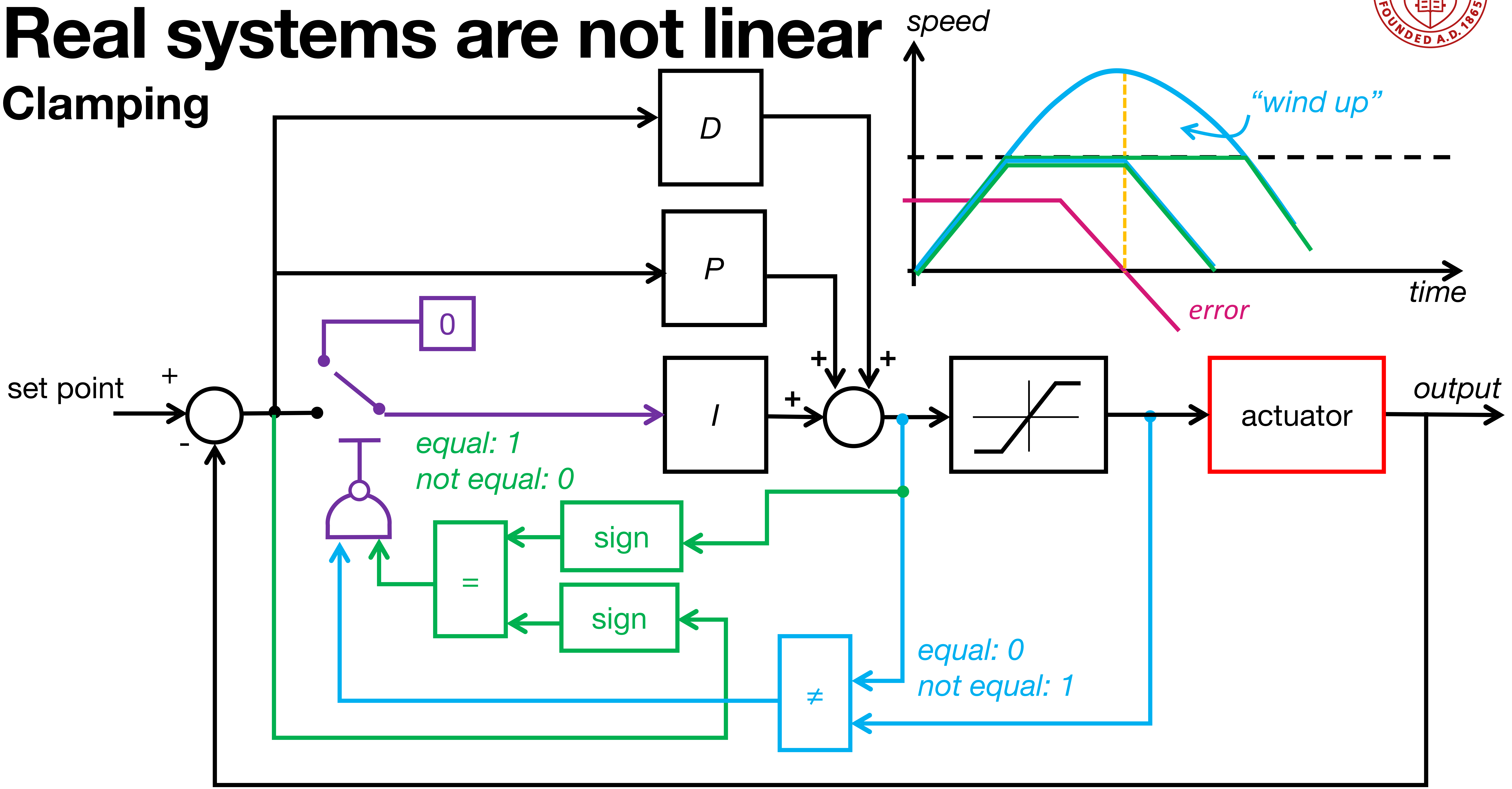
## Drone



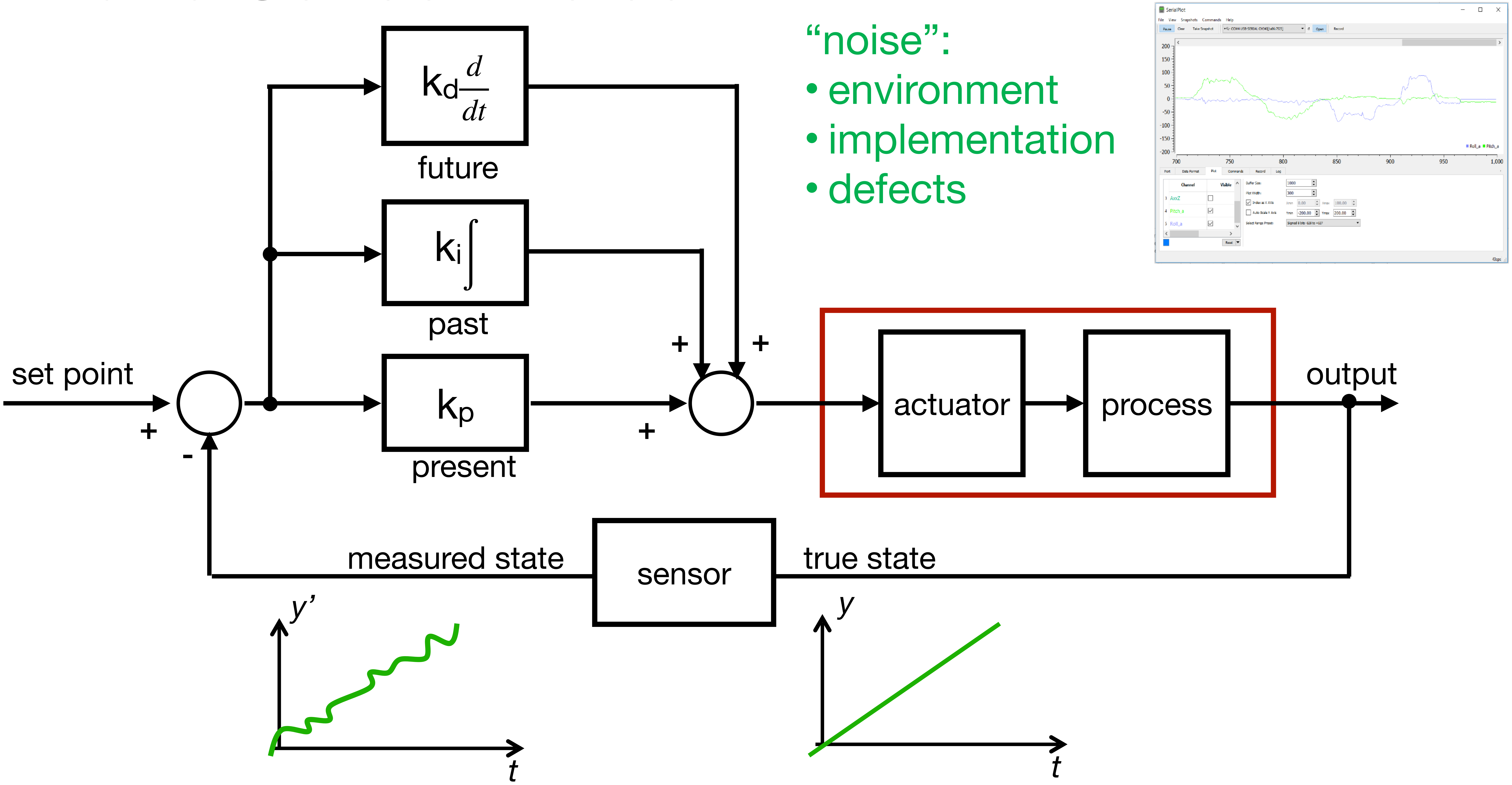



# Real systems are not linear

## Clamping



# PID and Sensor Noise

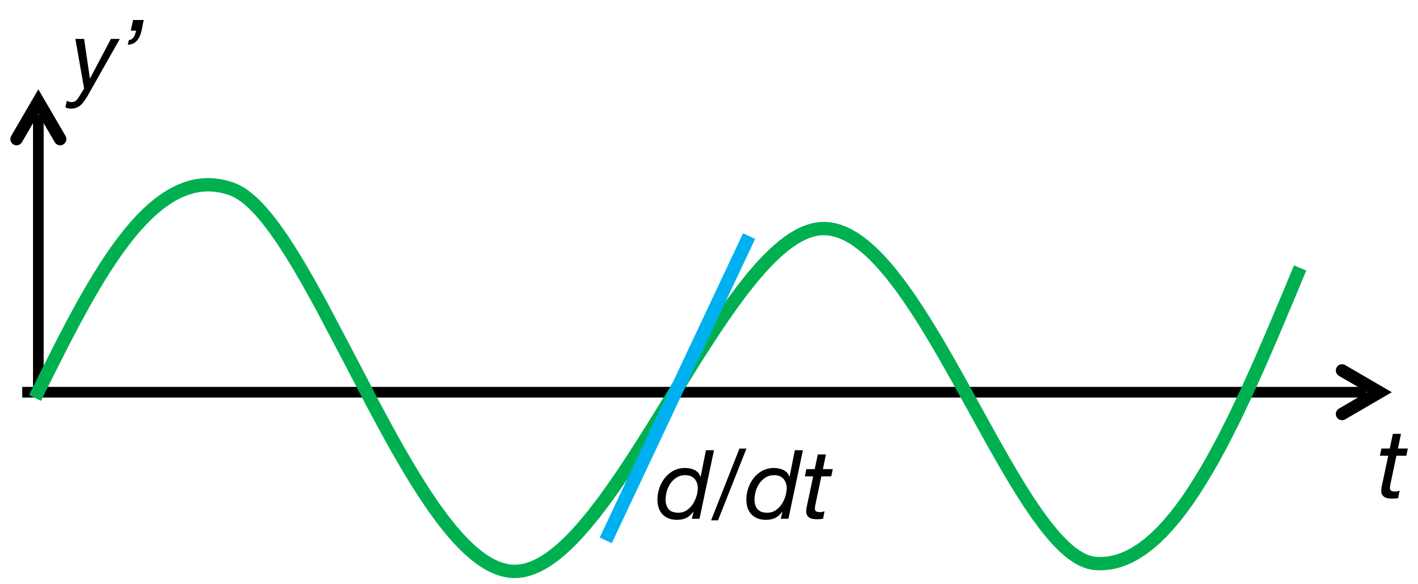


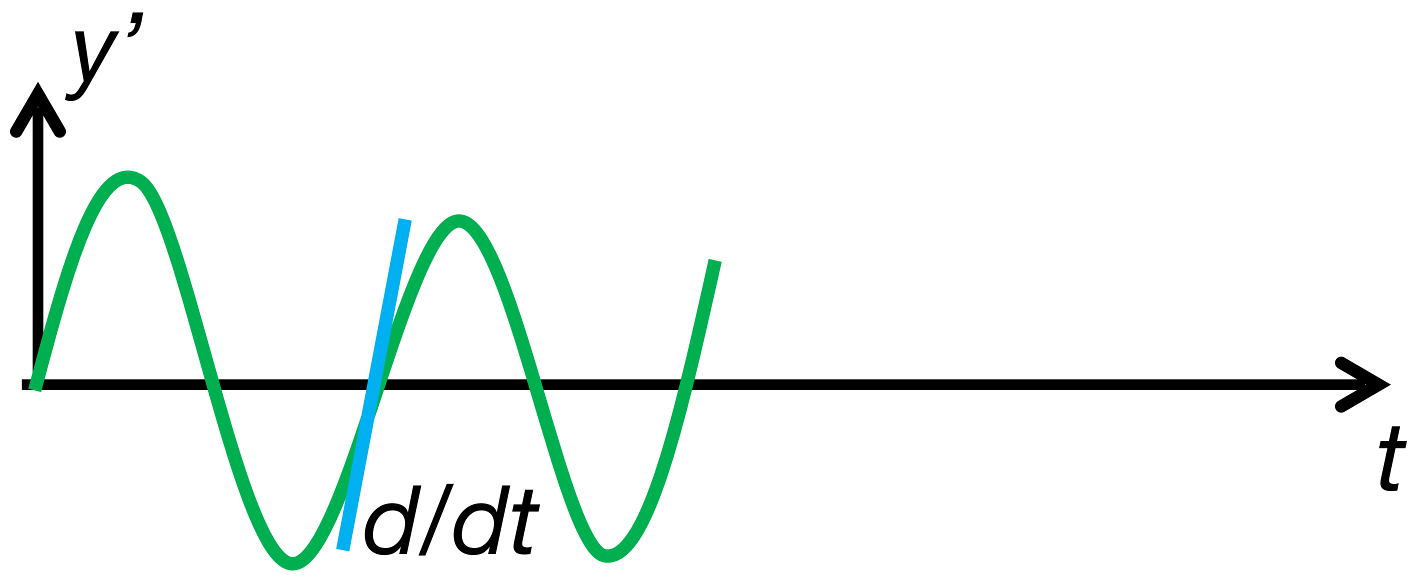

"noise":

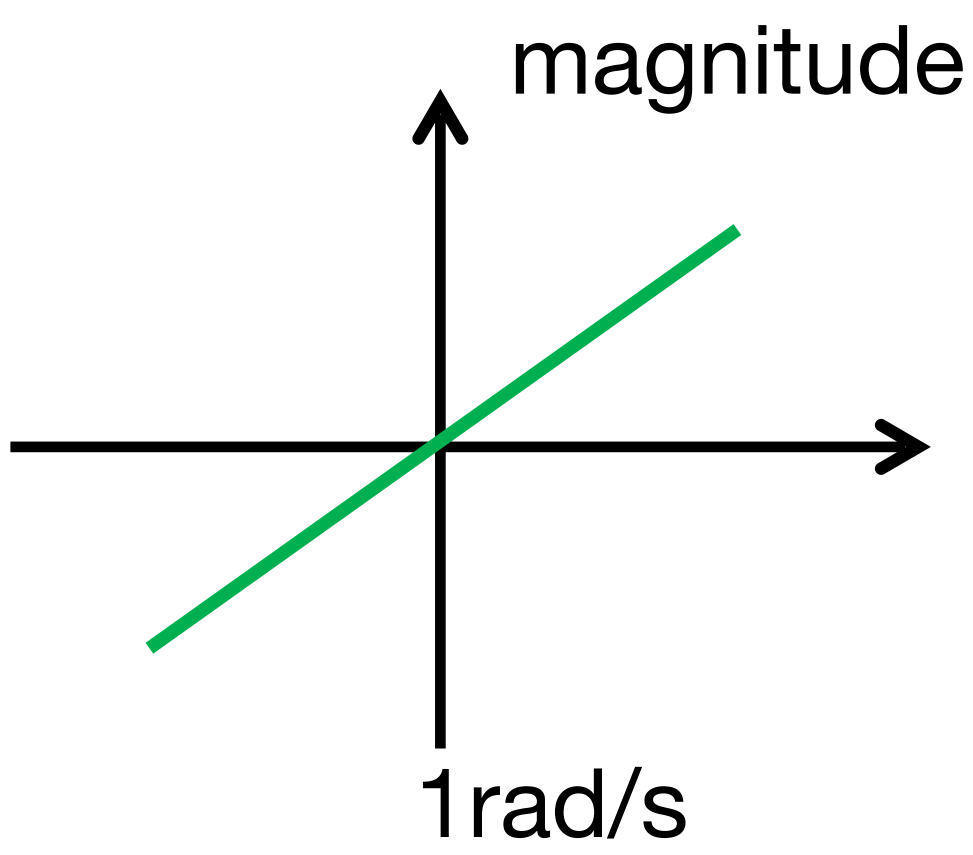
- environment
- implementation
- defects

# PID and Sensor Noise

## Derivatives amplify HF signals more than LF signals



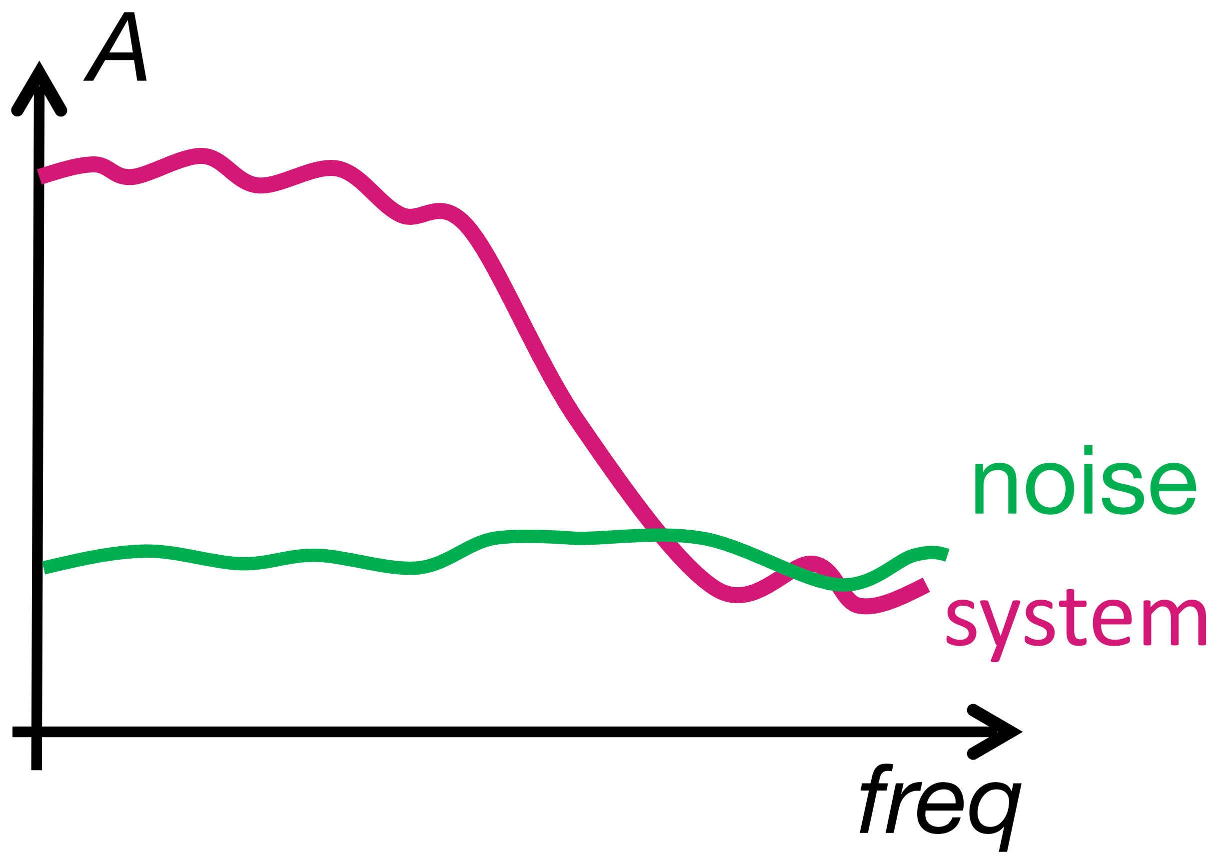



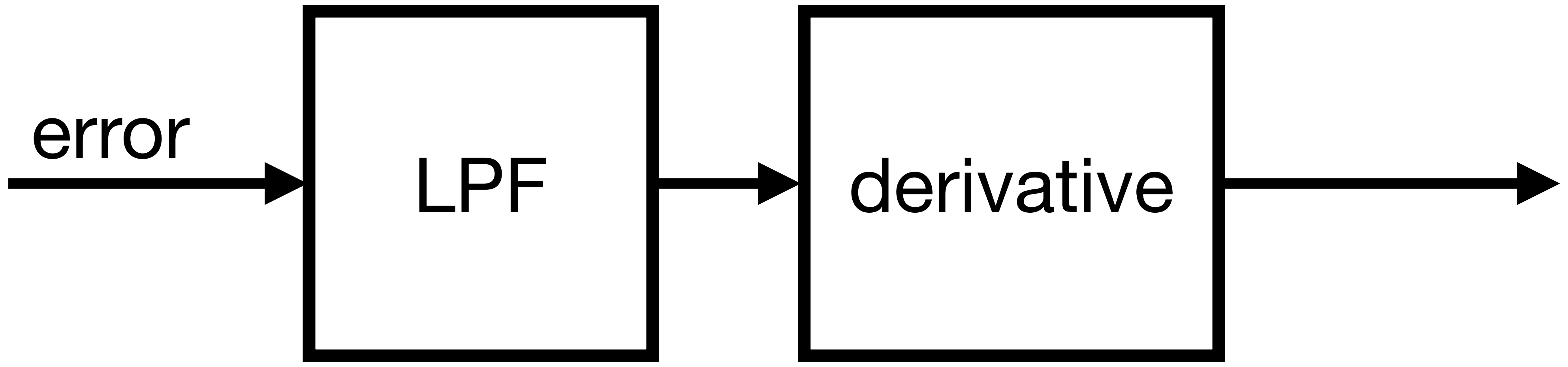

$y(t) = A\sin(\omega_a t + \phi_a) + B\sin(\omega_b t + \phi_b) + \ldots$

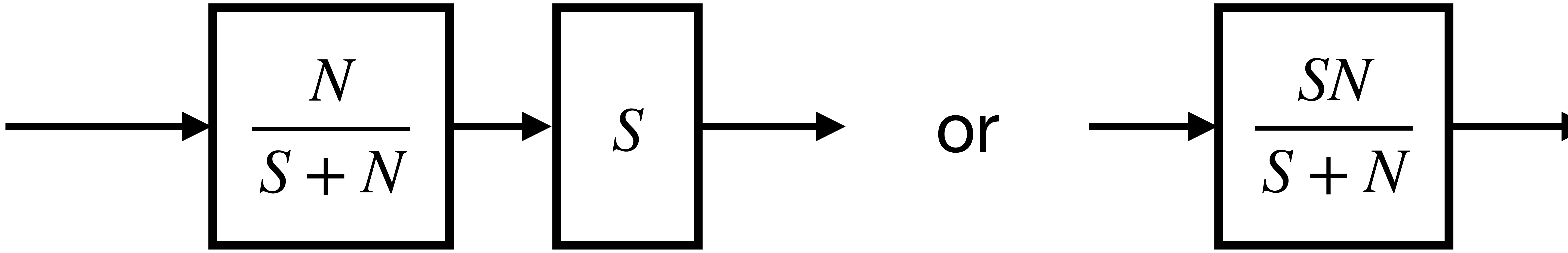
$dy(t)/dt = A\omega_a \sin(\omega_a t + \phi_a + 90^o) + B\omega_b \sin(\omega_b t + \phi_b + 90^o) + \ldots.$

- if $\omega_a$ >1rad/s, then amplitude will increase
- if $\omega_a$ <1rad/s, then amplitude will decrease

# PID and Sensor Noise

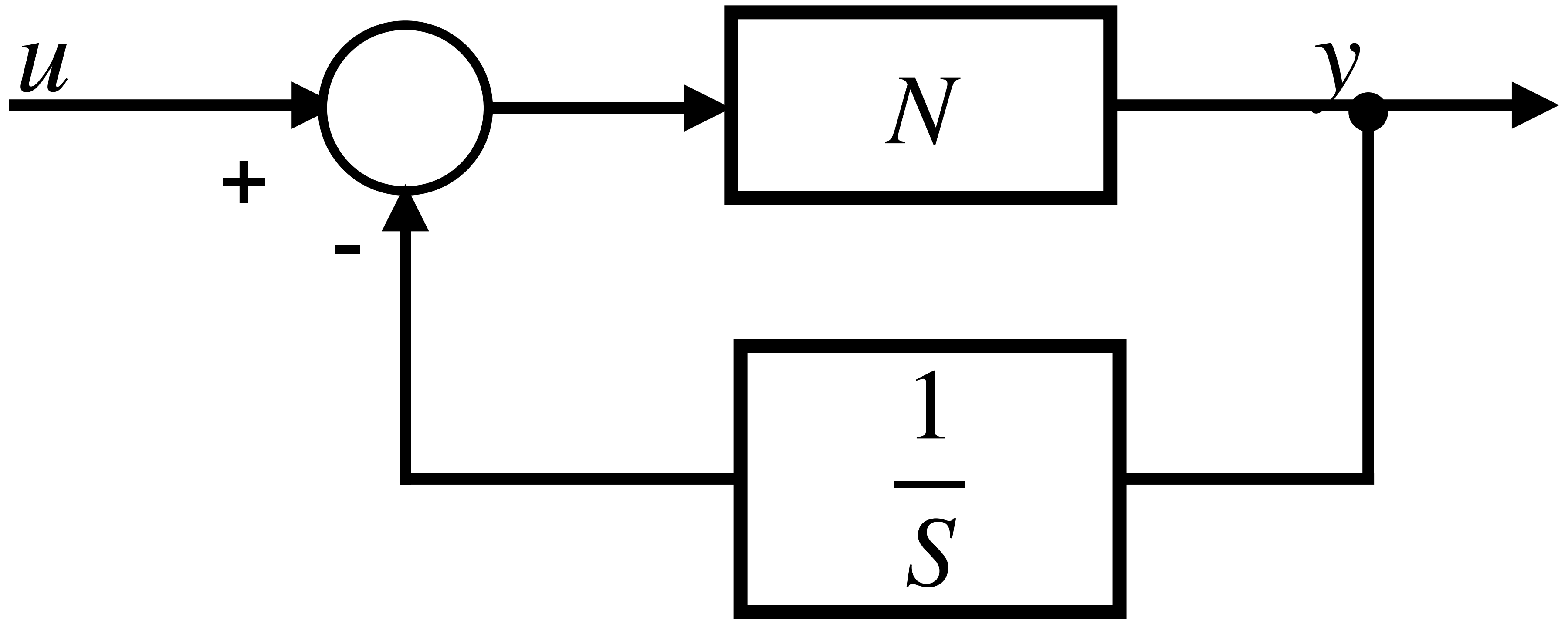



| Time | Laplace |
|---|---|
| $\frac{d}{dt}$ | $S$ |
| $\int dt$ | $\frac{1}{S}$ |
| | $\frac{N}{S+N}$ |

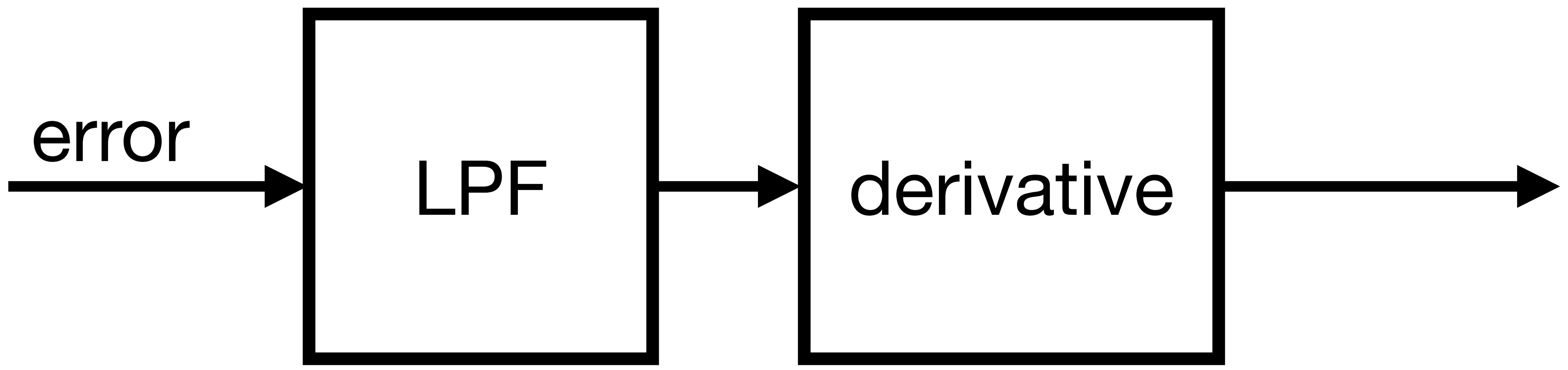
$$= \frac{1}{\frac{1}{N}S + 1} = \frac{1}{\tau S + 1}$$

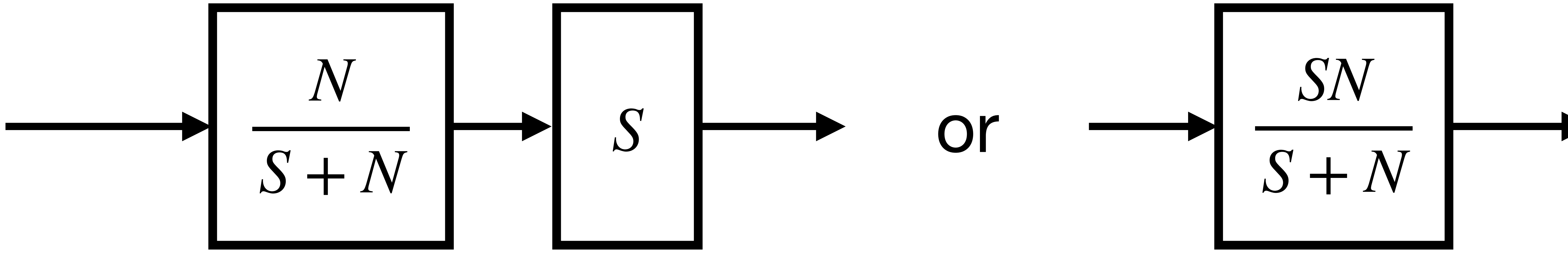


# PID and Sensor Noise

$$y = N\left(u - \frac{y}{s}\right) \qquad y = \frac{N}{1 + \frac{N}{s}}u$$

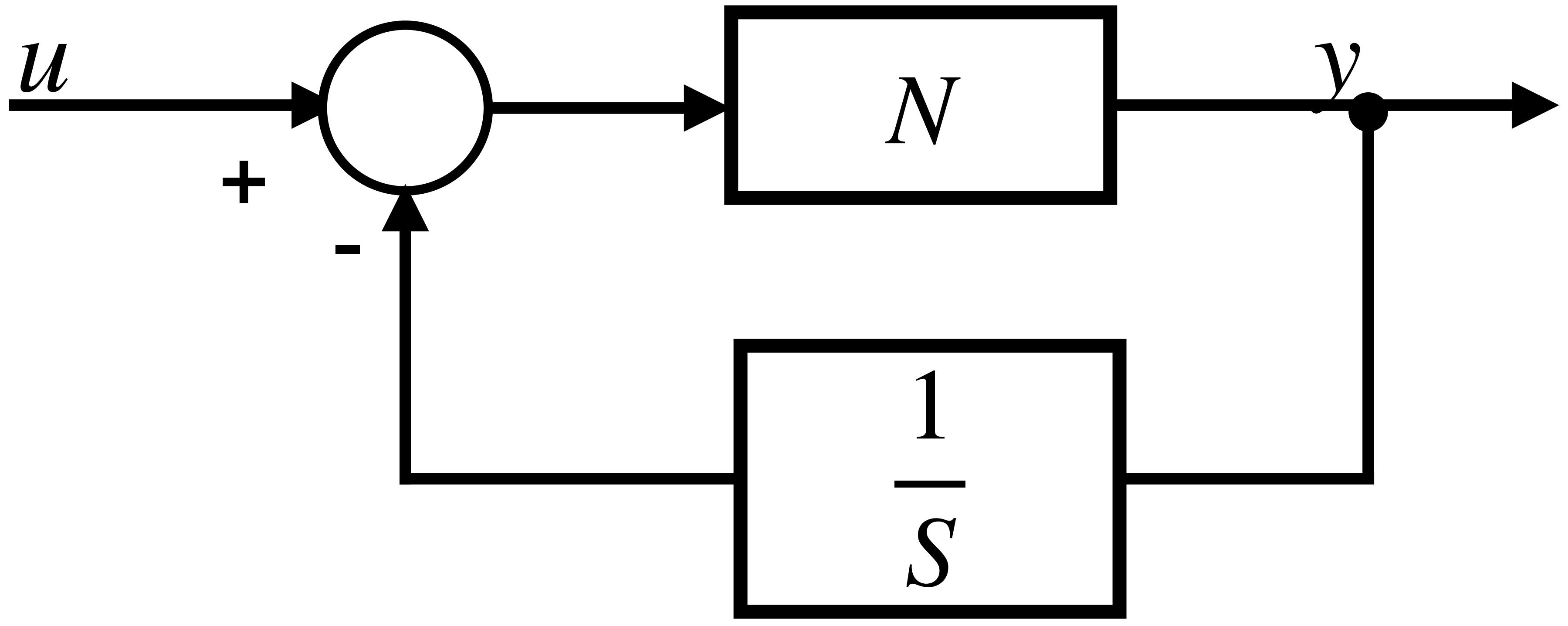
$$y + \frac{Ny}{s} = Nu \qquad \frac{y}{u} = \frac{N}{1 + N\frac{1}{s}}$$



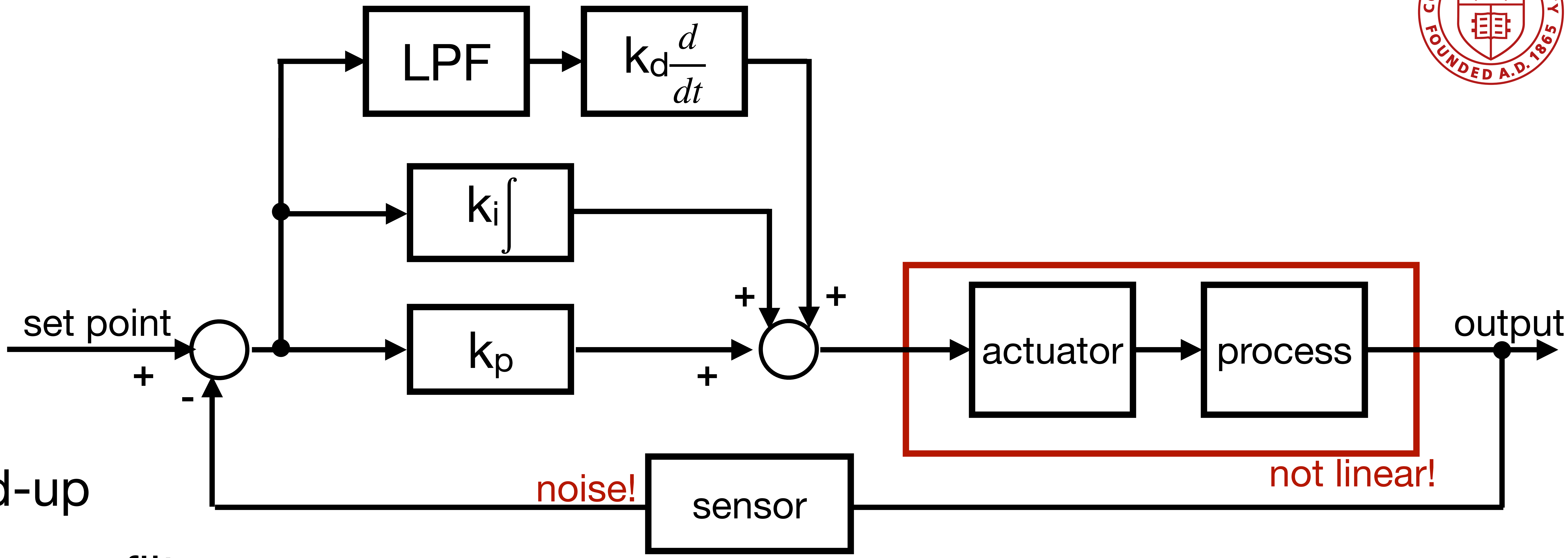




| Time | Laplace |
|---|---|
| $\frac{d}{dt}$ | $S$ |
| $\int dt$ | $\frac{1}{S}$ |
| 1st order LPF | $\frac{N}{S+N}$ |

$$= \frac{1}{\frac{1}{N}S + 1} = \frac{1}{\tau S + 1}$$

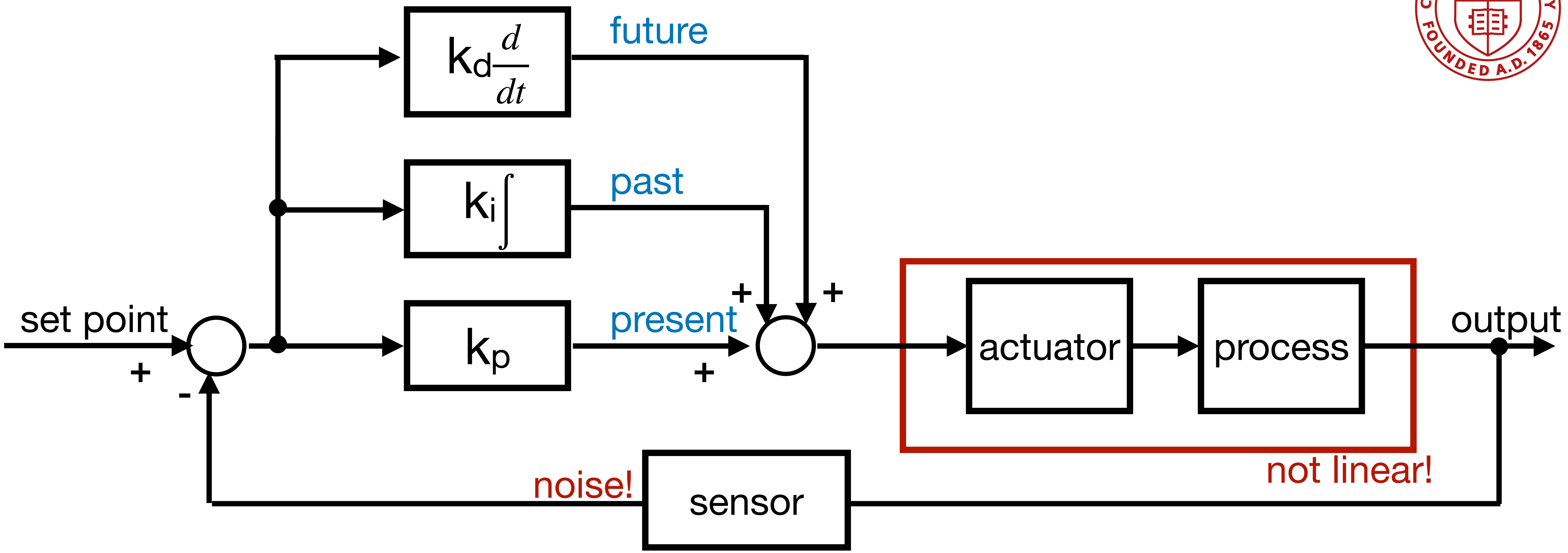


# PID




- Integrator wind-up
- Derivative low pass filter
- Derivative kick
- $$\frac{de}{dt} = \frac{dset}{dt} - \frac{dmeas}{dt}$$
- Constant setpoint: $\frac{de}{dt} = -\frac{dmeas}{dt}$

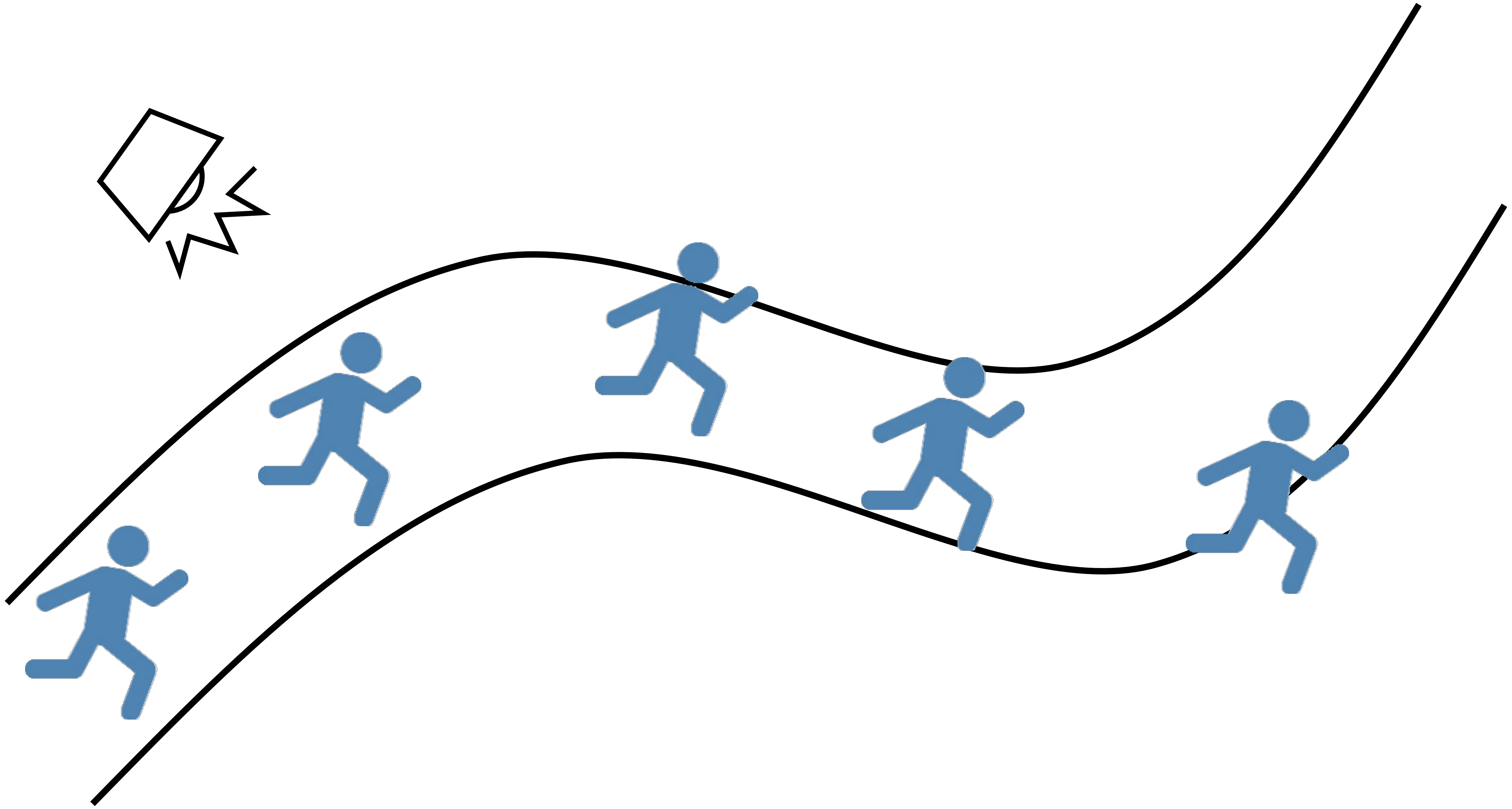
# PID



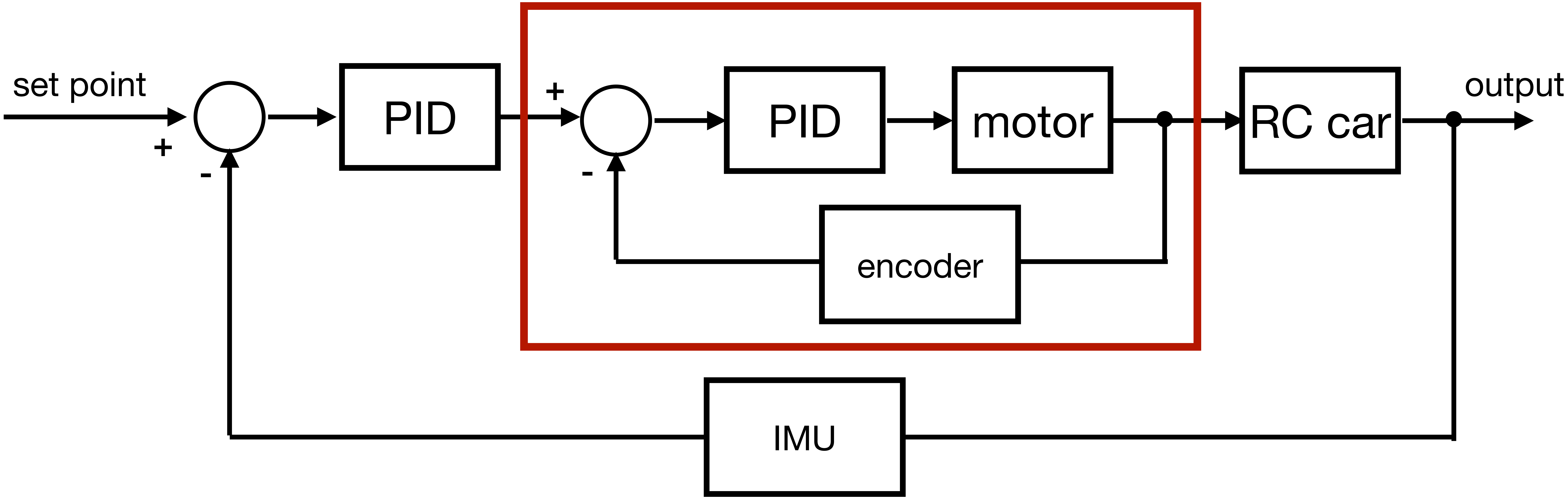

- Rise time/ response: some percent of final value
- Peak time: time to reach first peak
- Overshoot: amount in excess of final value
- Settling time: time before output settles to x% of final value

# Discrete PID Control

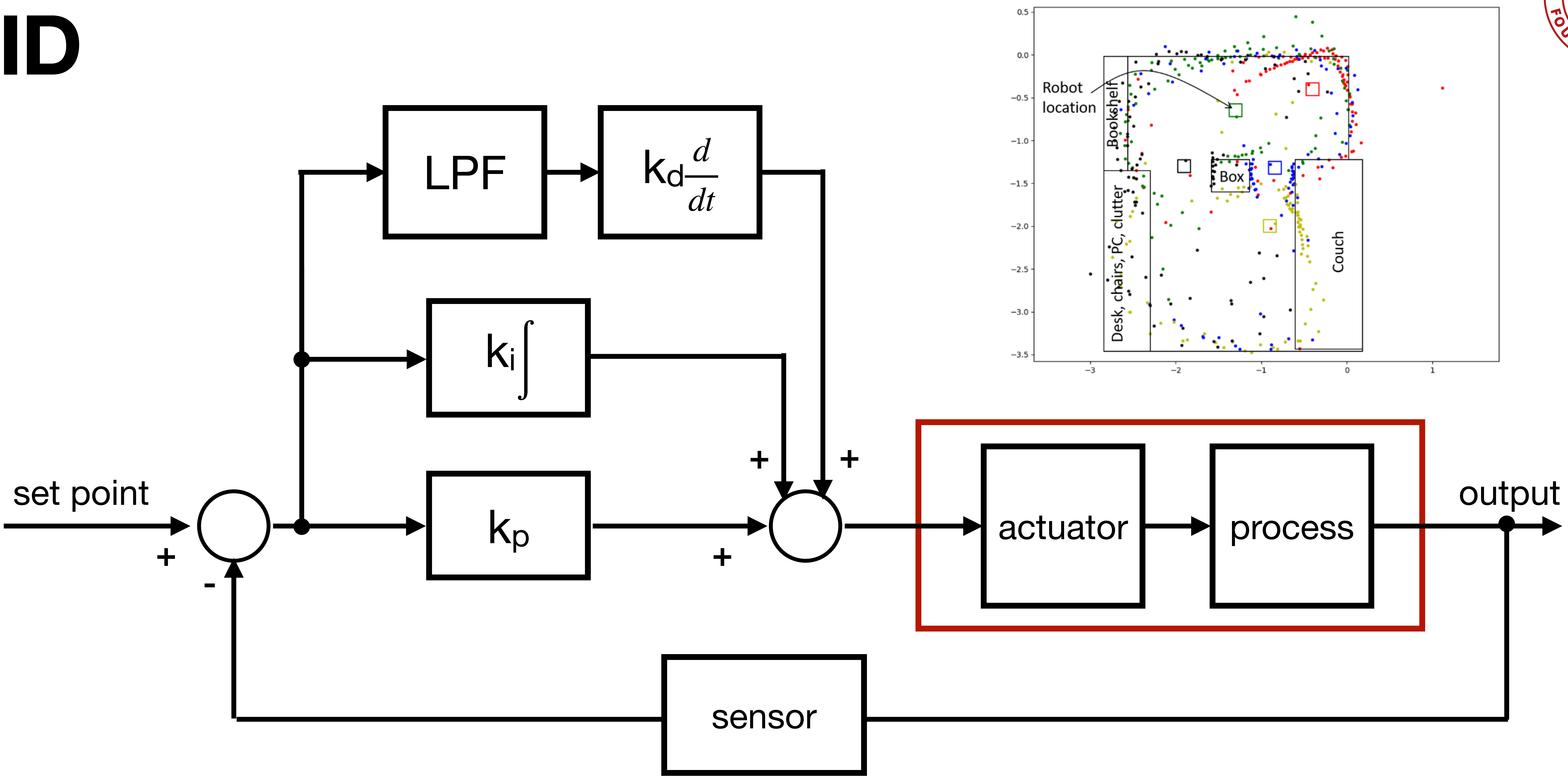
- Sampling time
- Control ~10 times faster than the system dynamics



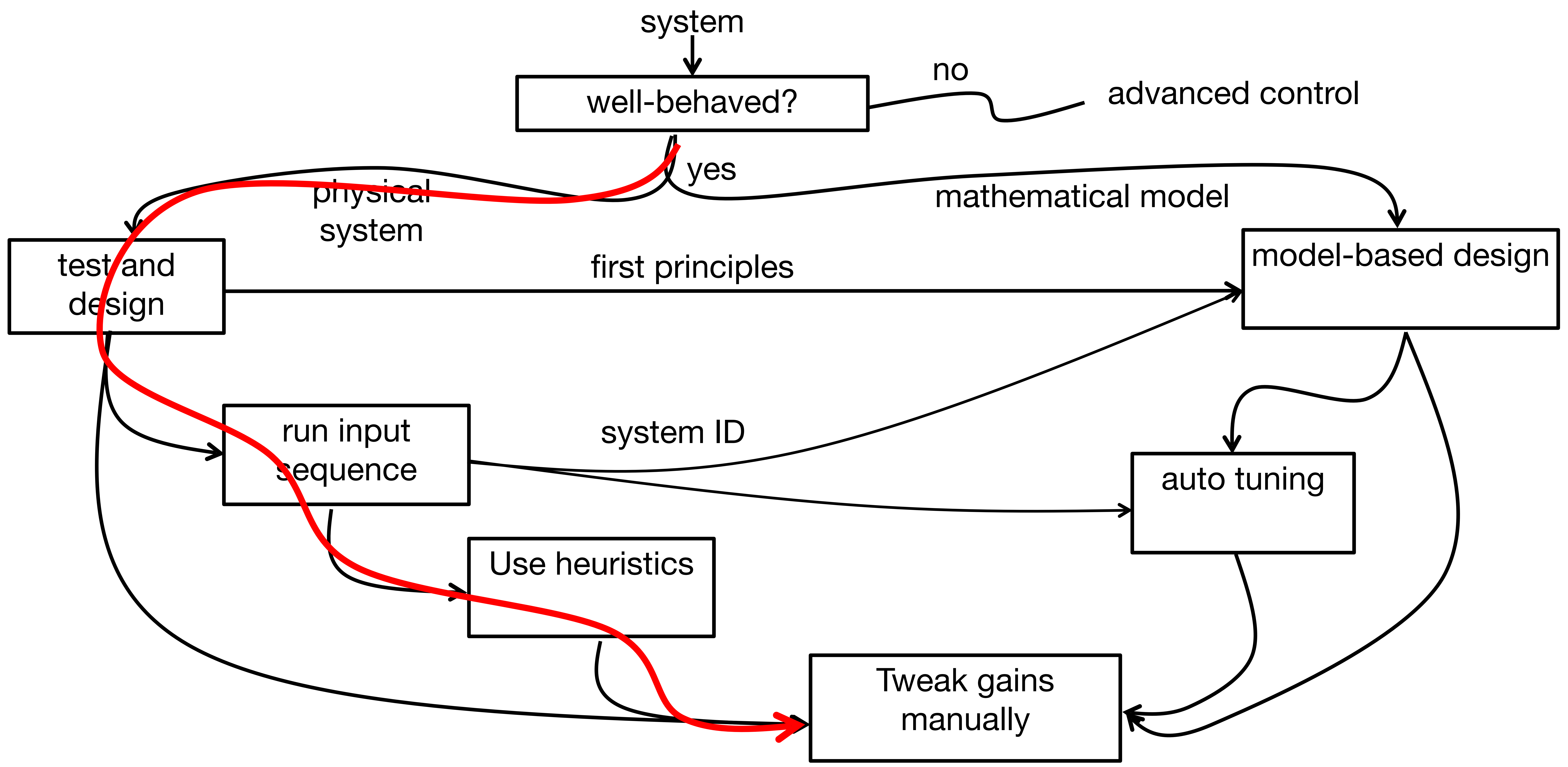
# Cascaded Control Loops



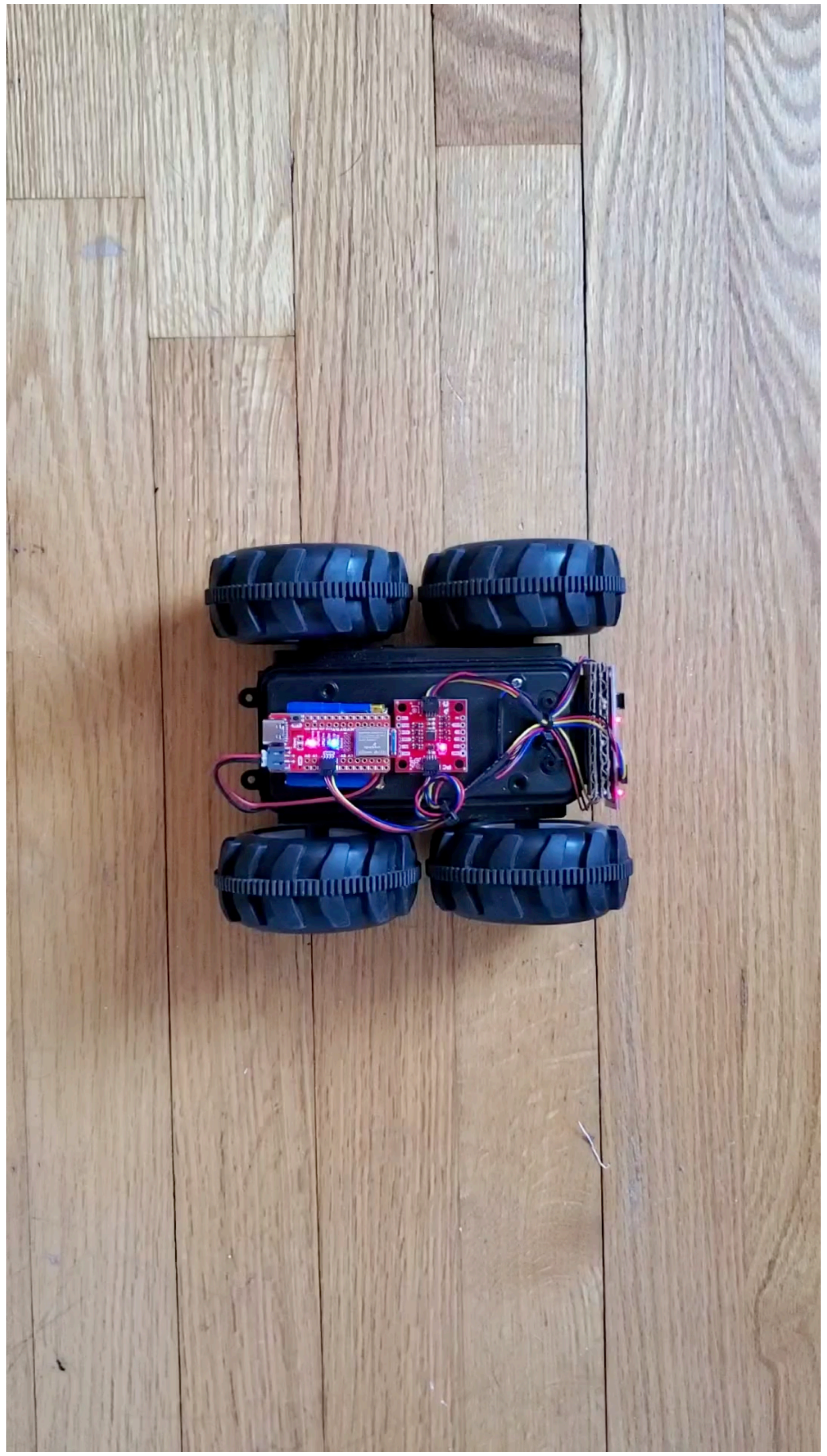
# PID

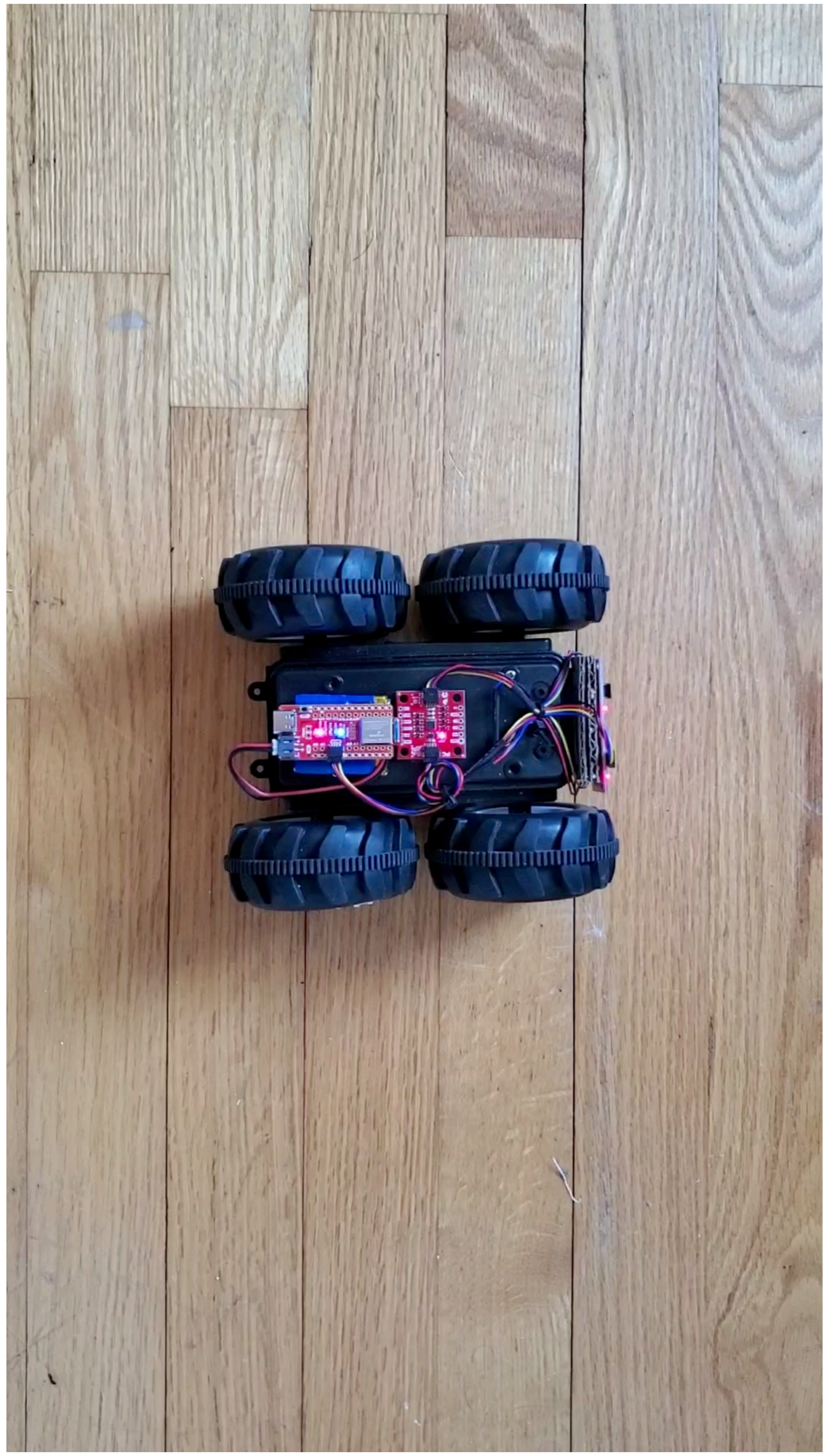

0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3.0
−3.5
−3
−2
−1
0
1
Robot location
Bookshelf
Box
Desk, chairs, PC, clutter
Couch
LPF
$k_d \frac{d}{dt}$
$k_i \int$
$k_p$
set point
+
-
+
+
+
actuator
process
output
sensor

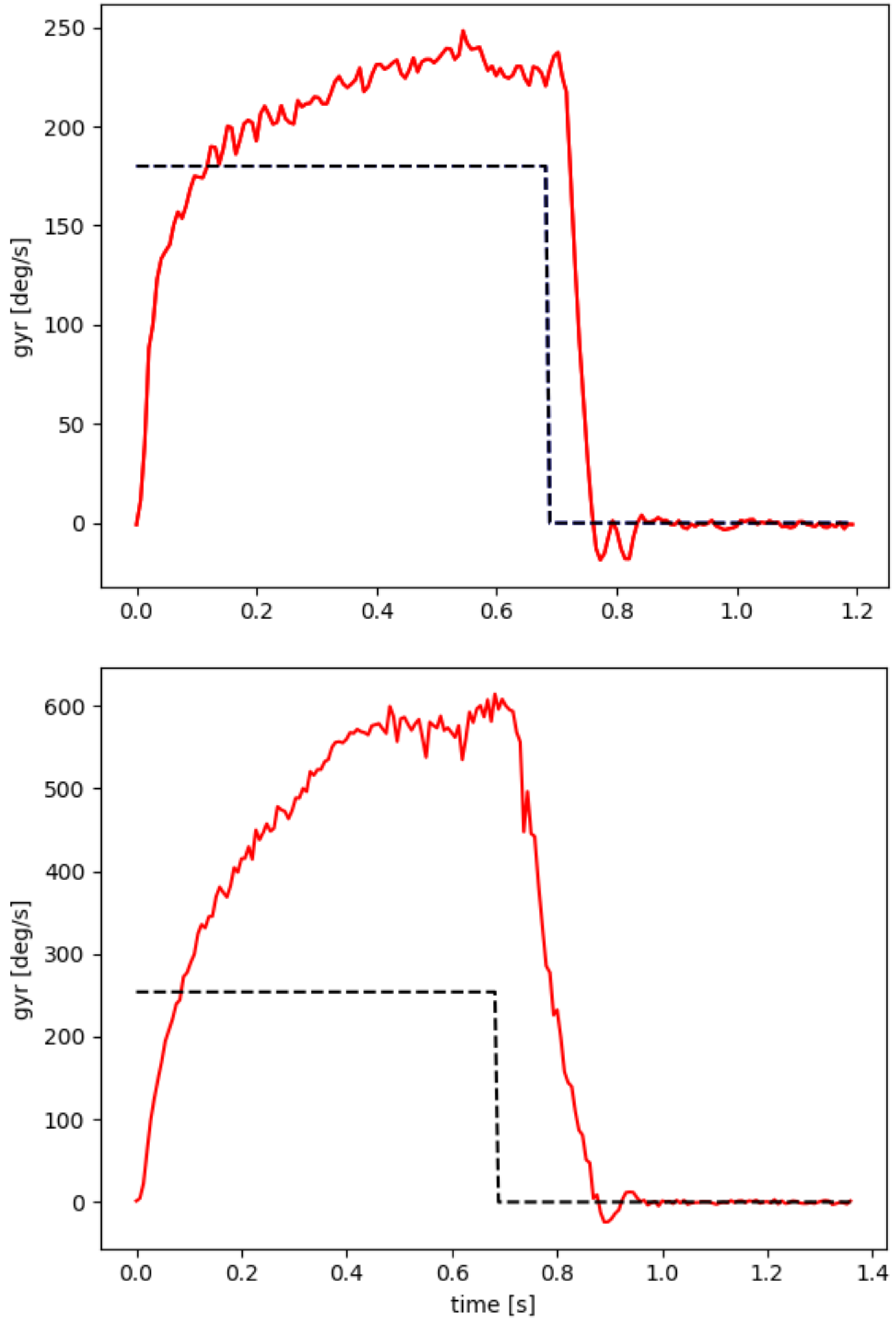
# Tuning PID control



# Tuning PID control







# Tuning PID control

## Chien, Hornes, and Reswick method

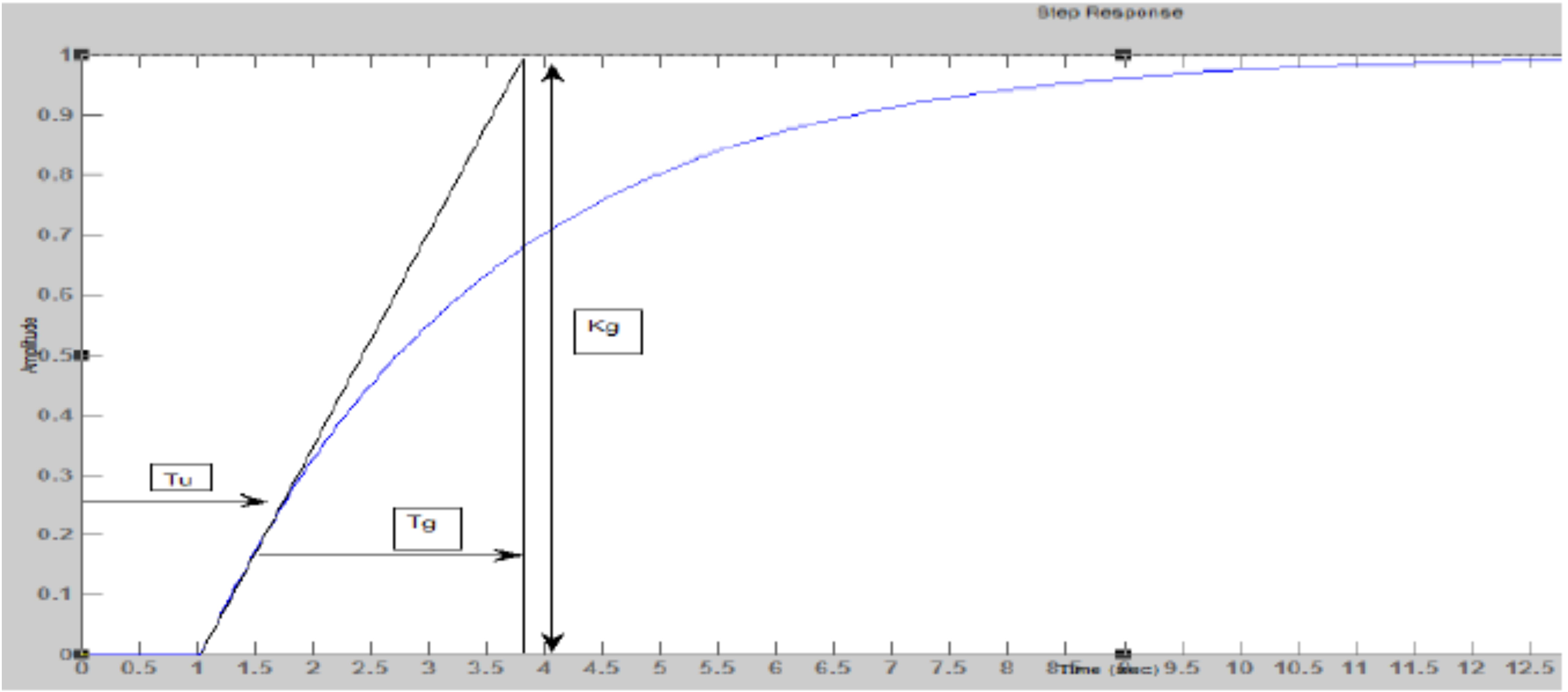
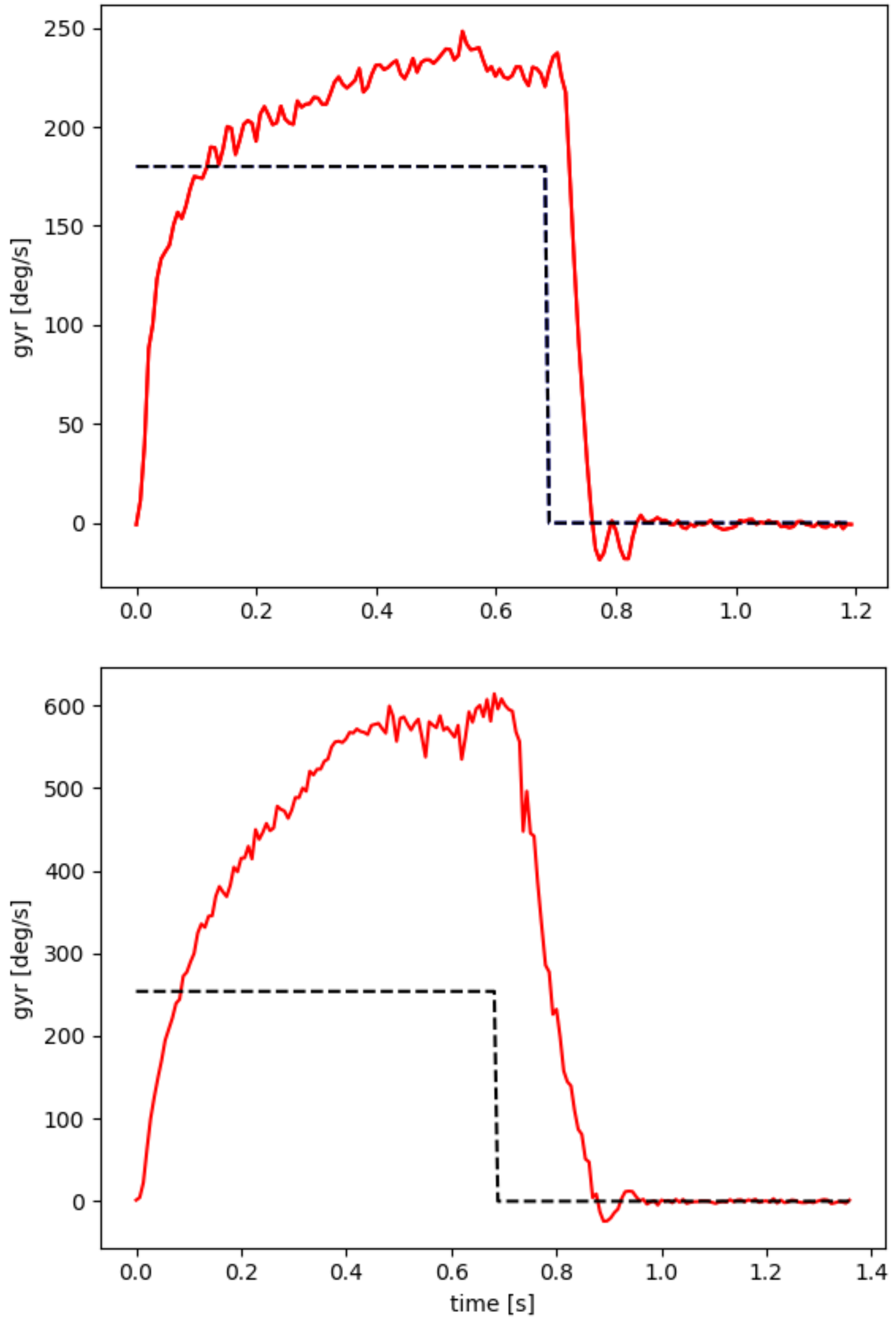


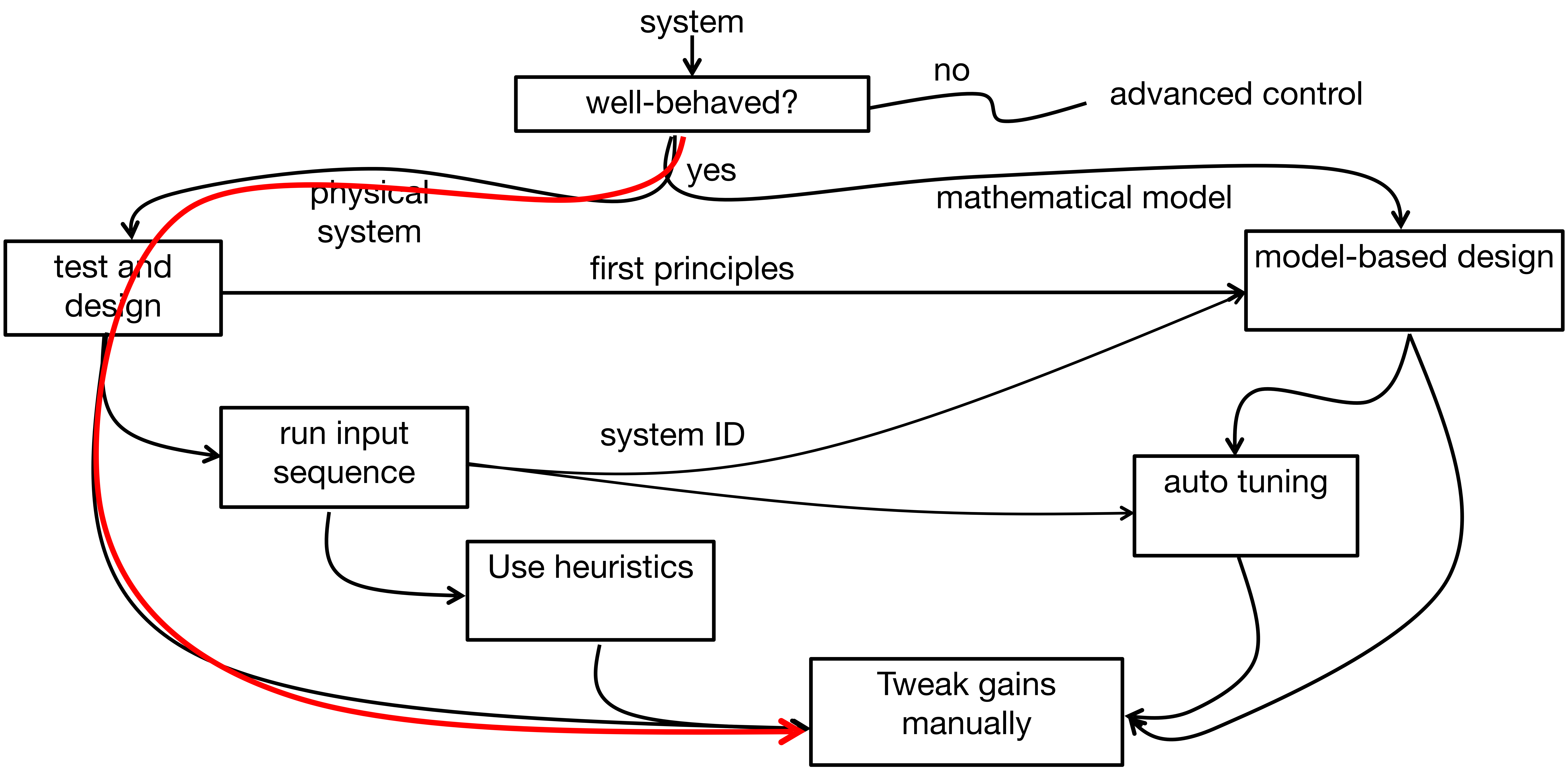

Fig.7. Open loop response of CHR method

Table.11. CHR Compensator

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



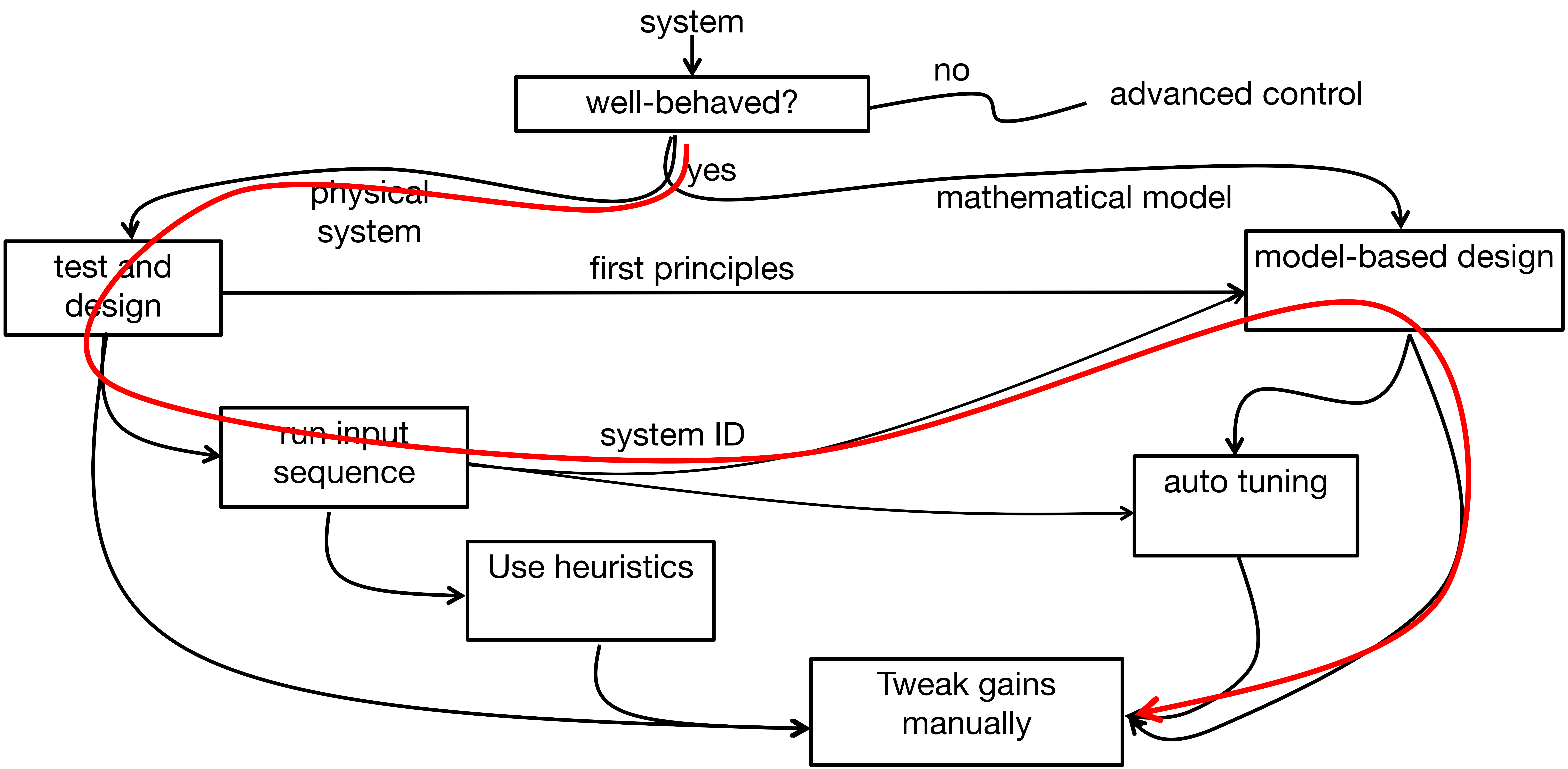
# Tuning PID control



# Tuning PID control

- 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

# Tuning PID control

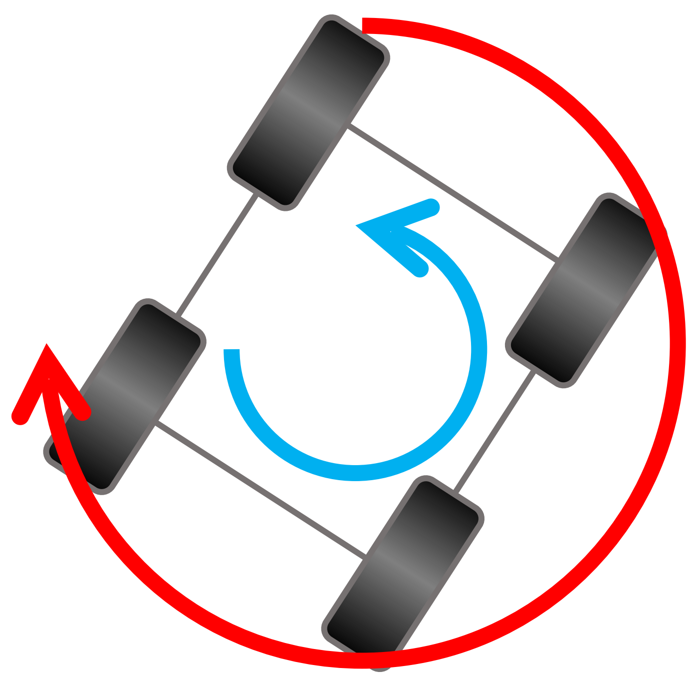


https://tinyurl.com/y67glgzk

# Tuning PID control

## Equations of motion

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



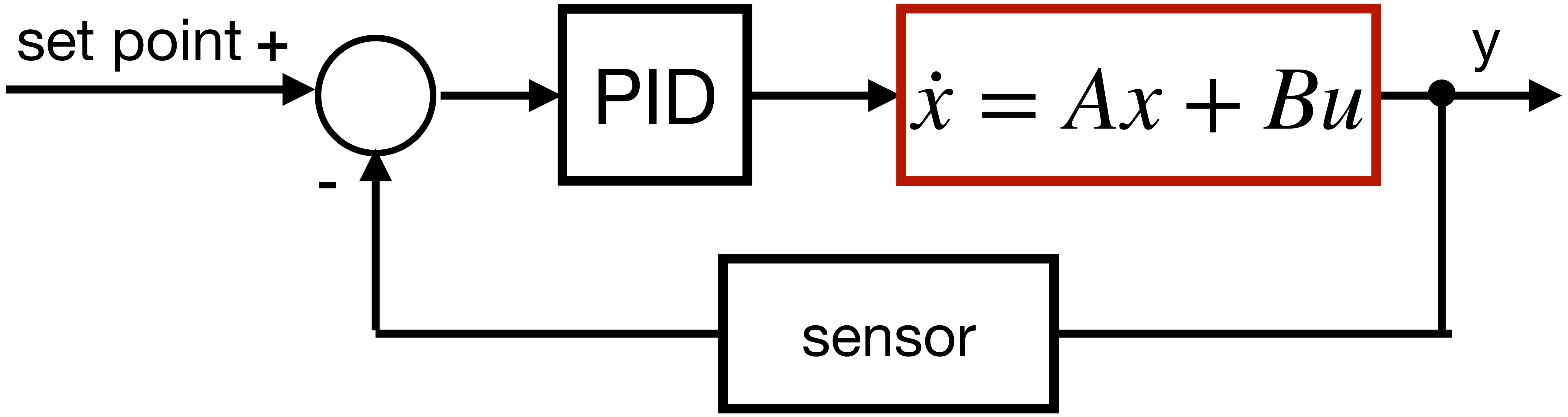
$$F = ma$$

$$\tau = I\alpha$$

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

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

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




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

# Lab 3

Good example from last year:
https://boltstrike.github.io/pages/lab3.html