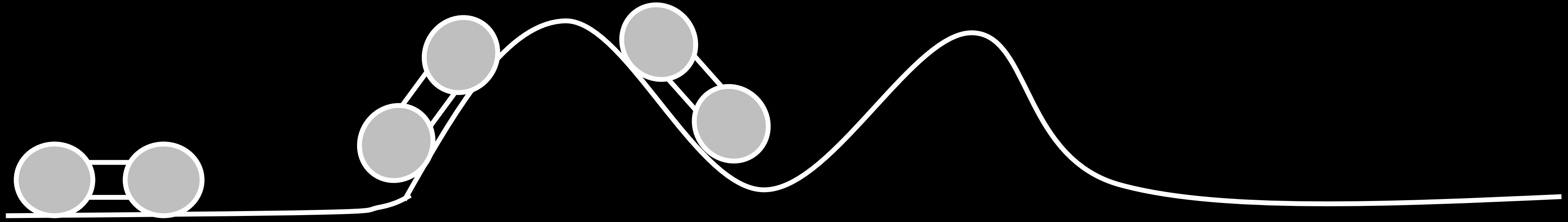
Prof. Kirstin Hagelskjær Petersen
kirstin@cornell.edu

Fast Robots

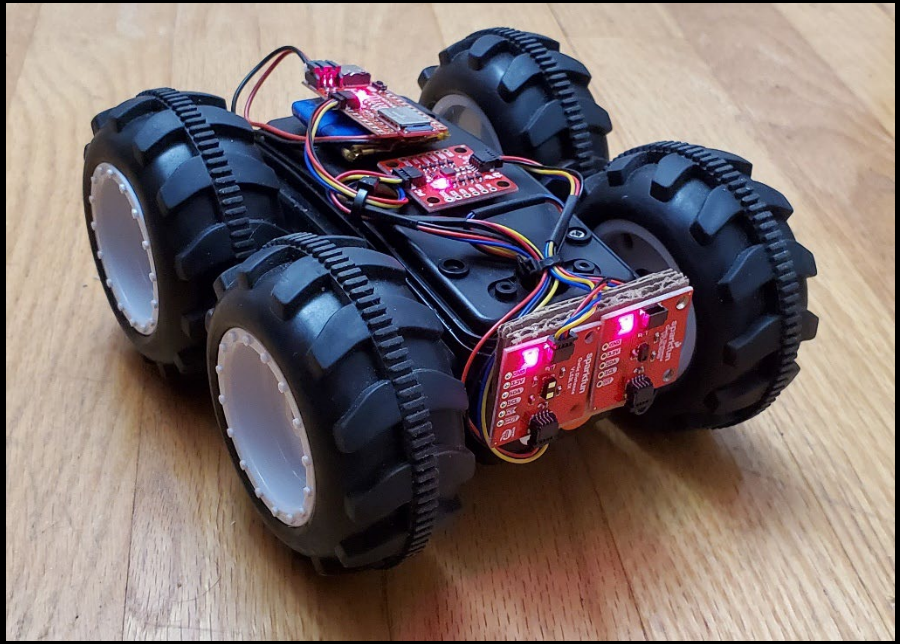
Feedback Control

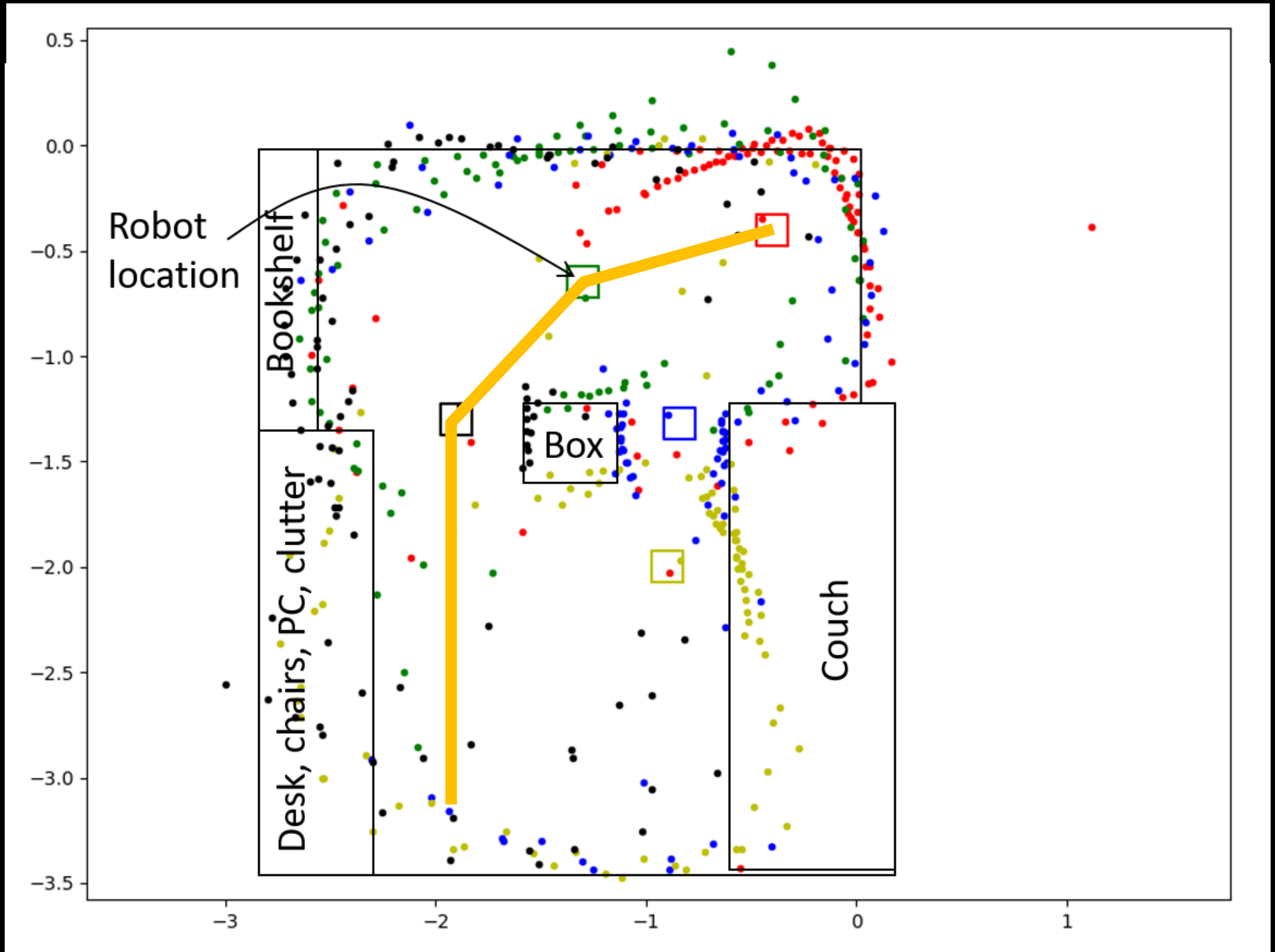
- Maintaining speed prediction at different battery levels, over different surfaces
- Maintaining position with respect to walls
- Etc.



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

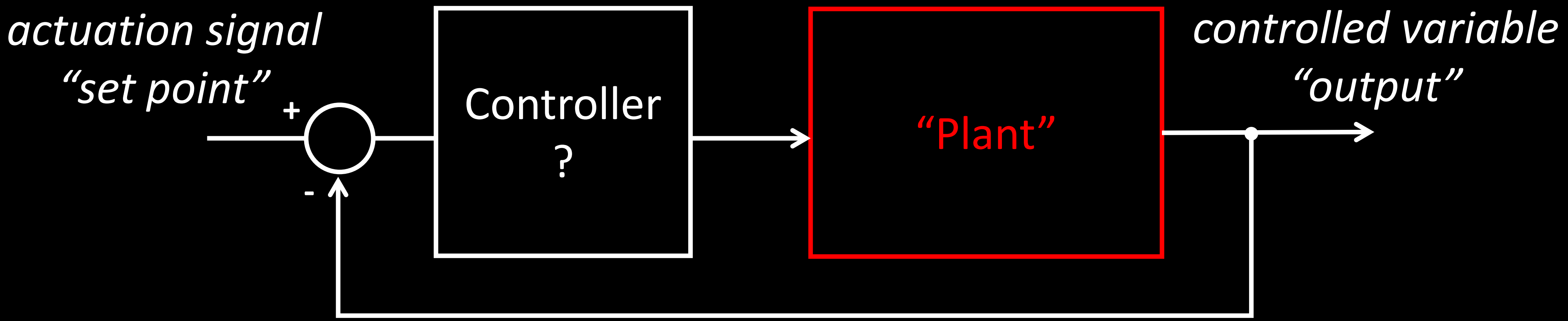




PID control

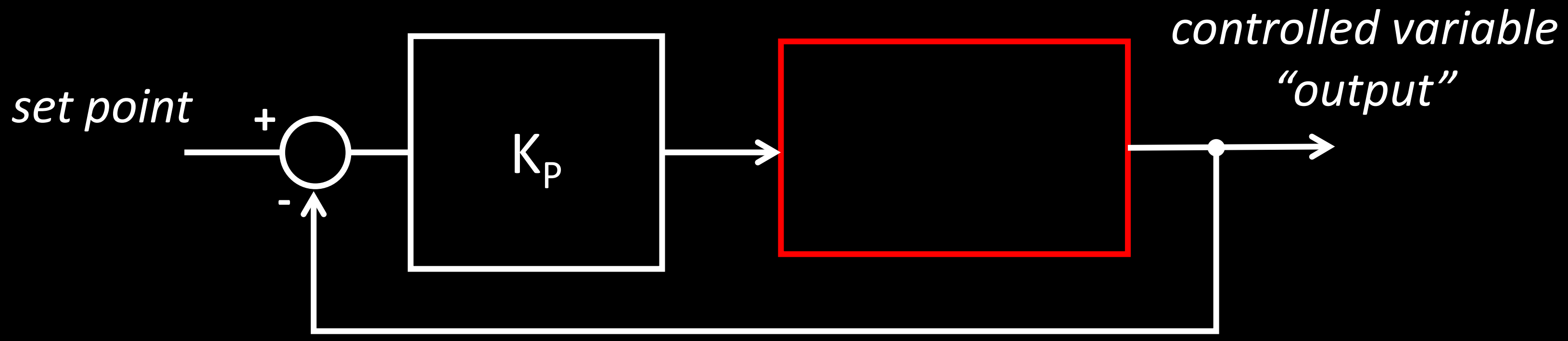
Heavily inspired by a Matlab Tech Talk: Understanding PID Control

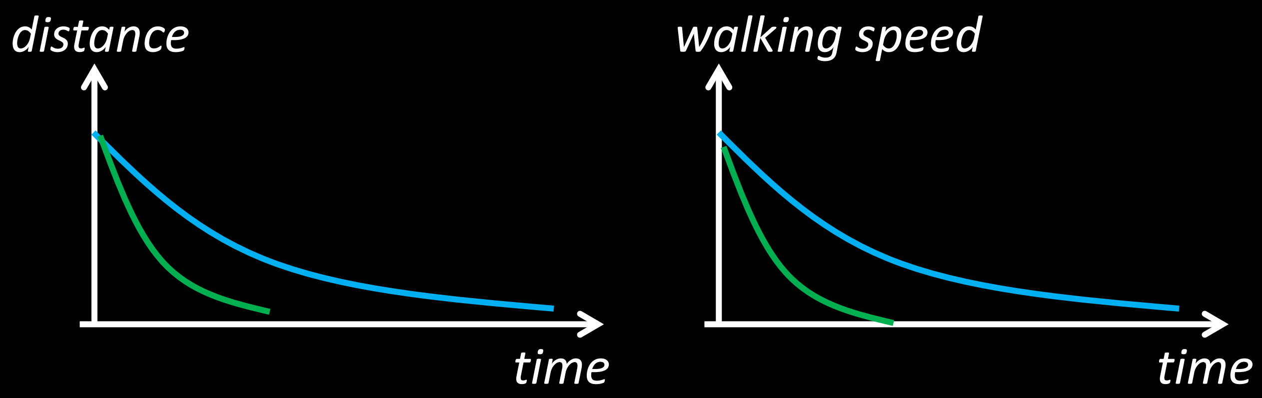
$$u(t) = K_P e(t) + K_I \int_0^t e(t)dt + K_D \frac{de(t)}{dt}$$



PID control

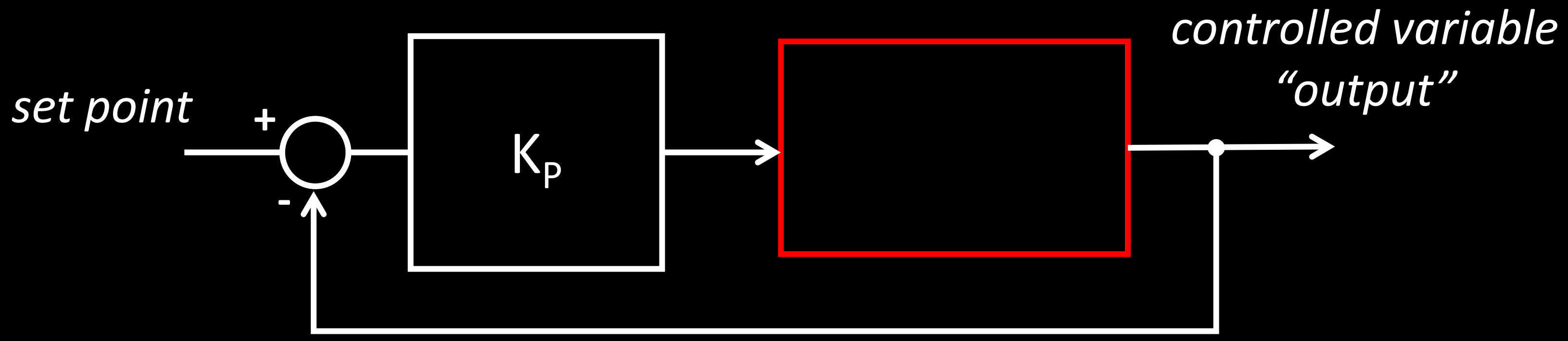
- Soccer field example





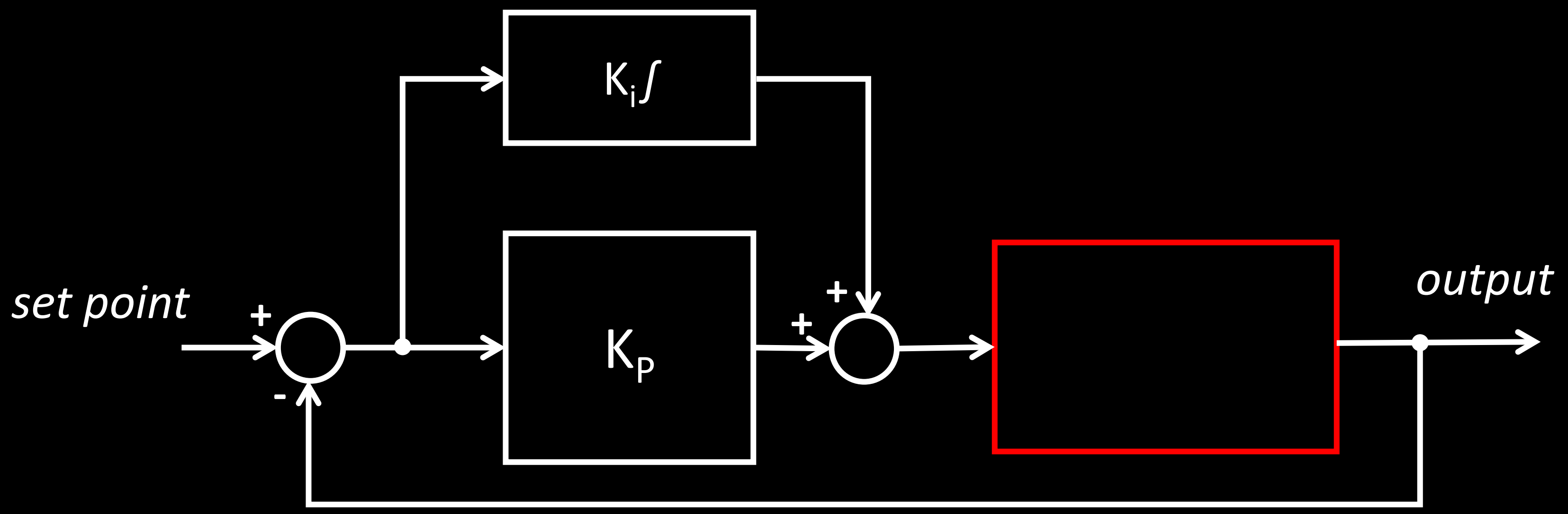
PID control

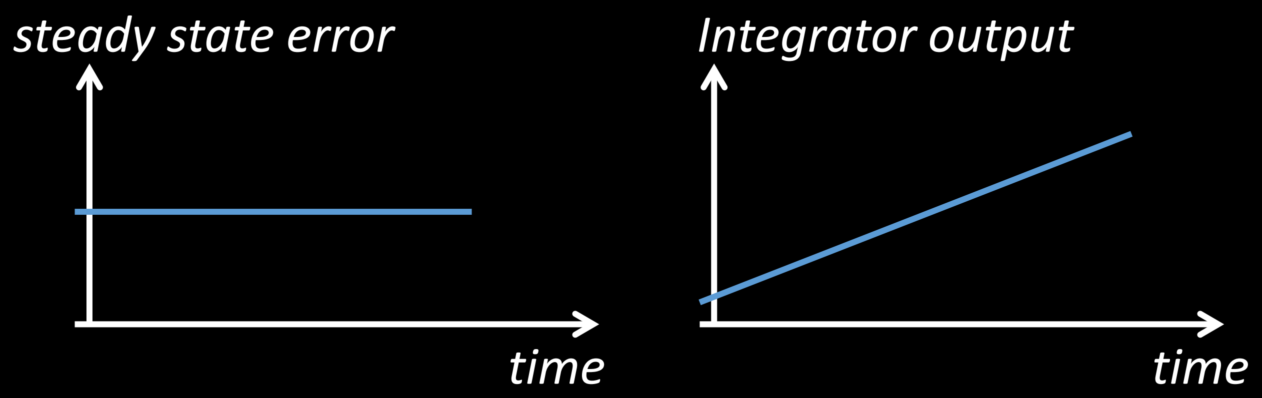
- Drone example
  - But there's gravity…
  - Hover at 100rpm
    - Kp = 2, a > 0m
    - Kp = 5, a = 30m
    - Kp = 10, a = 40m
    - Kp = 100, a = 49m
    - Steady state error



PID control

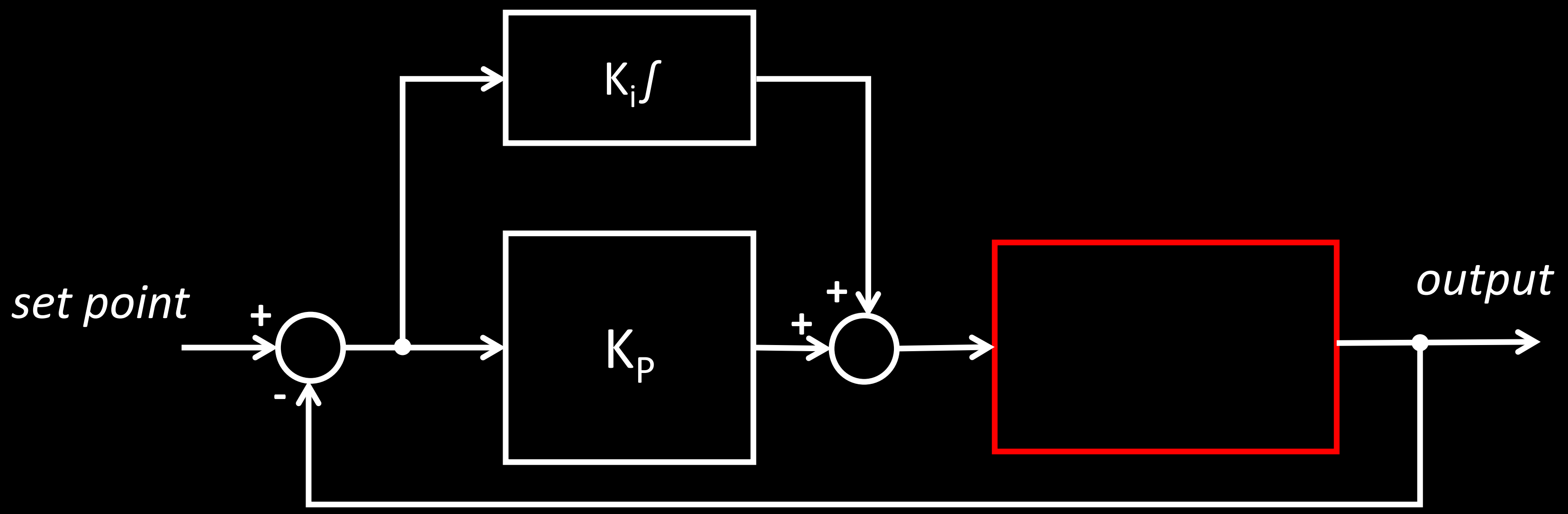
- Drone example
  - But there's gravity…
  - Hover at 100rpm
    - Kp = 2, a > 0m

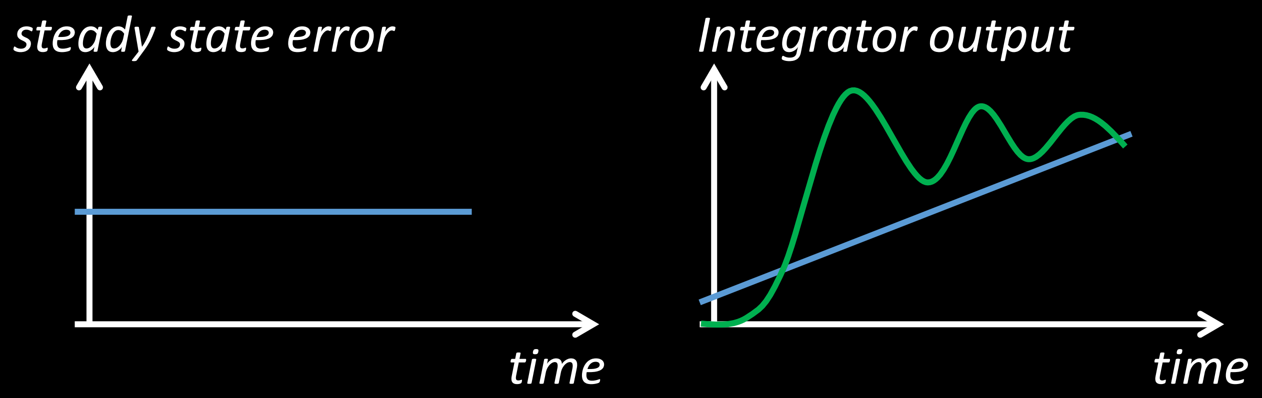




PID control

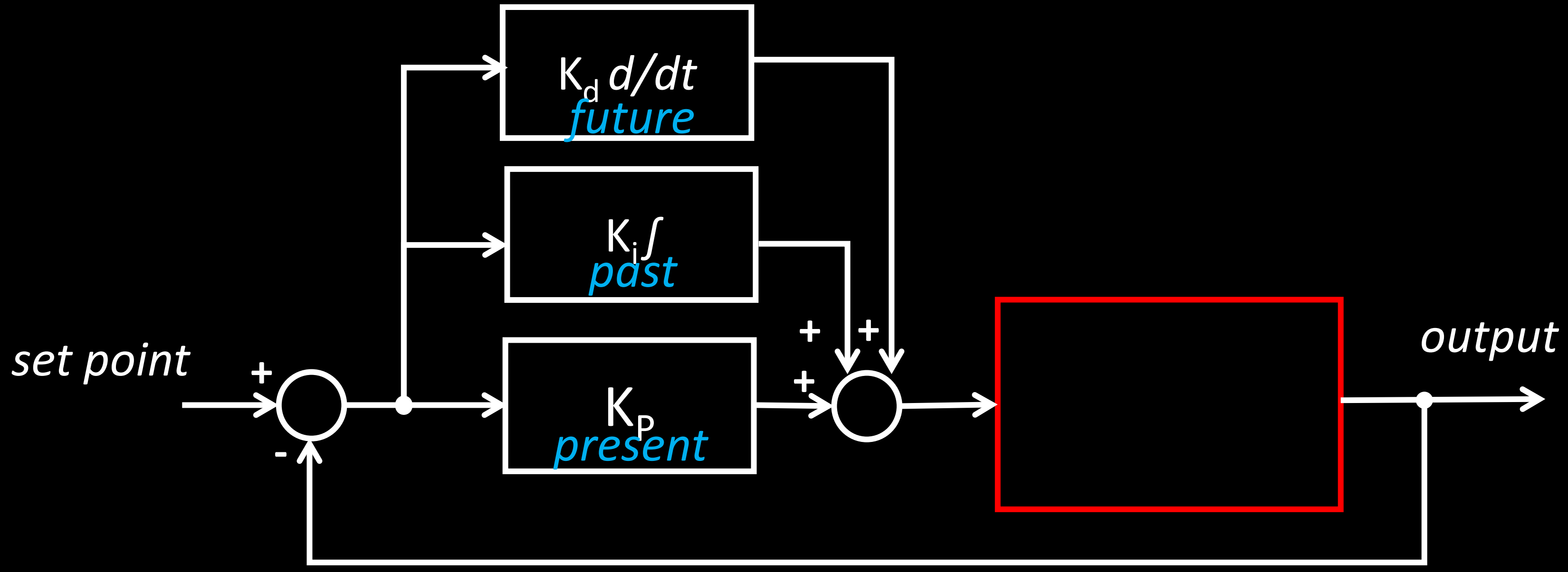
- Drone example

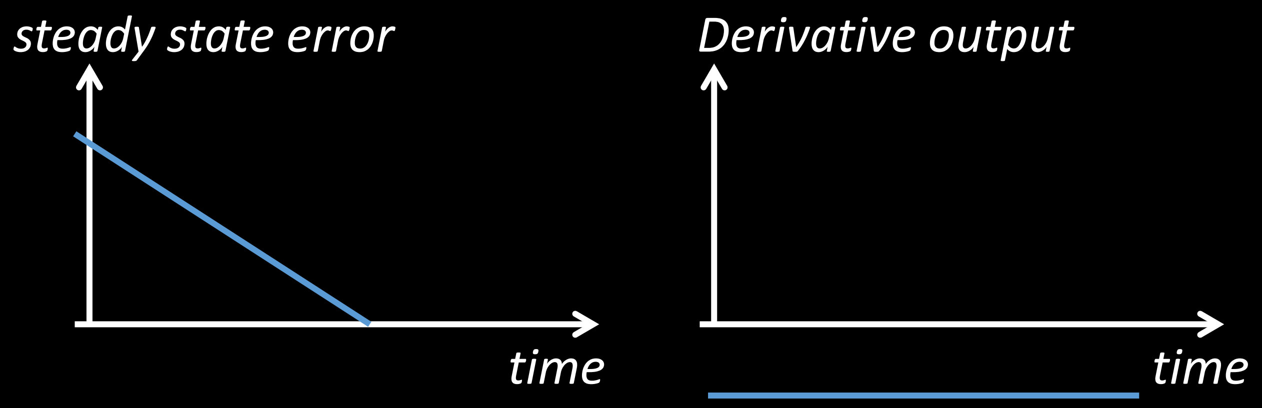




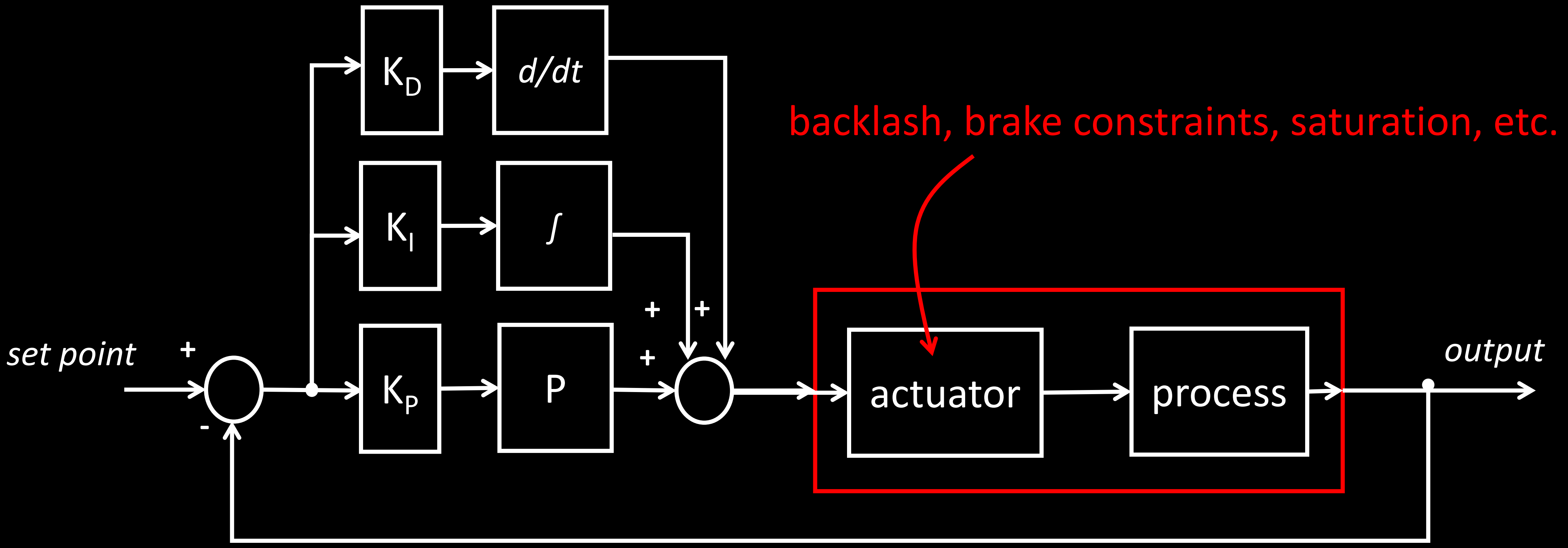
PID control

- Drone example

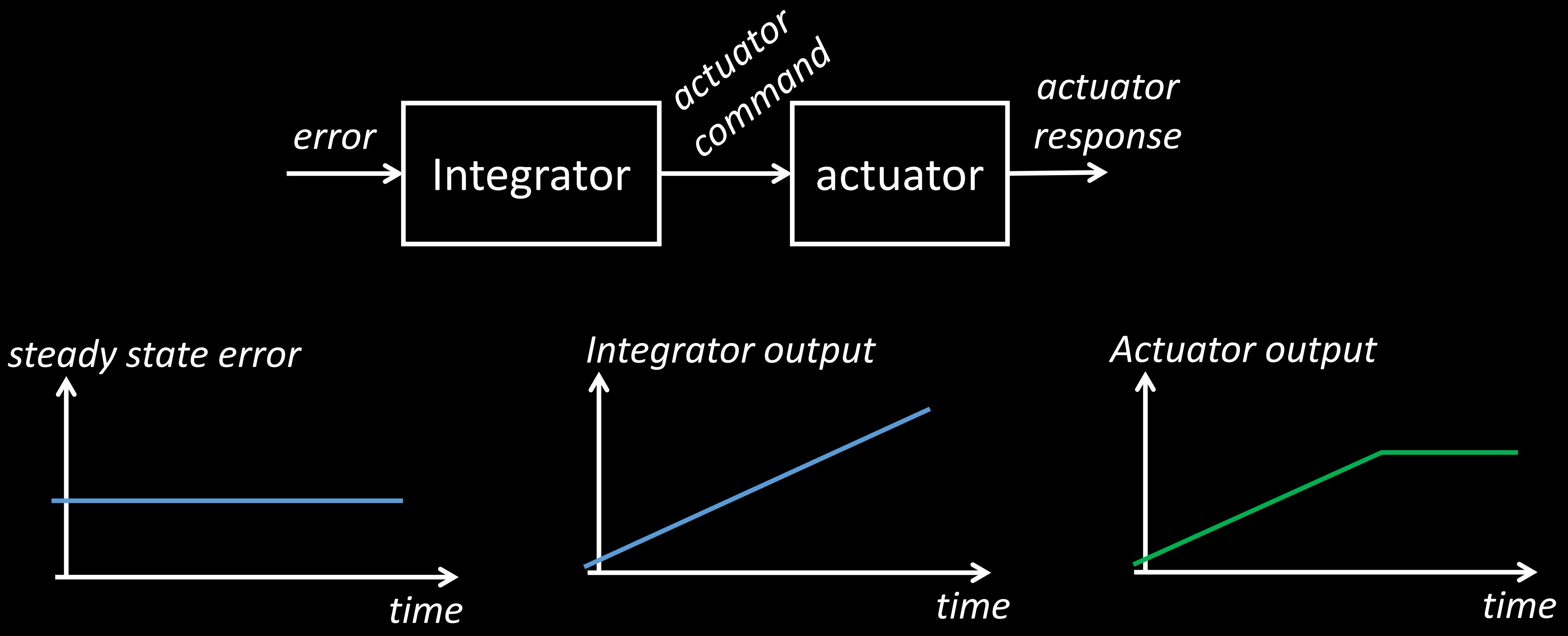




Real Systems are not linear!

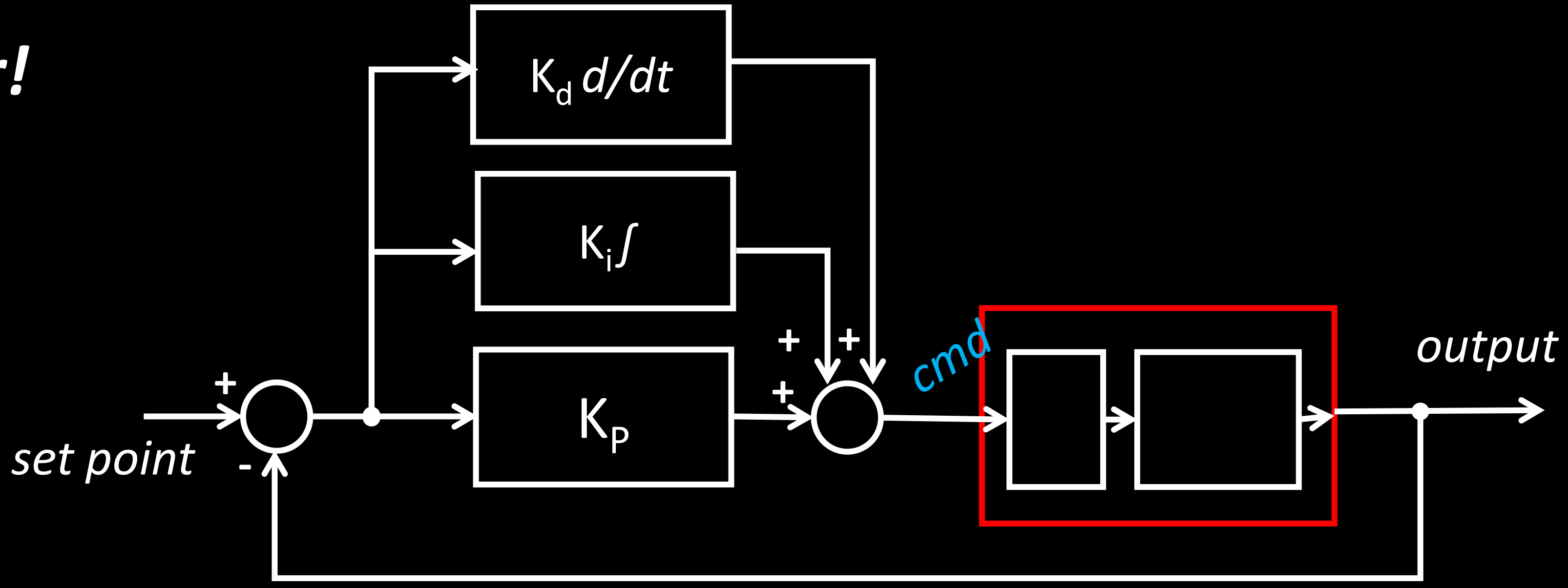


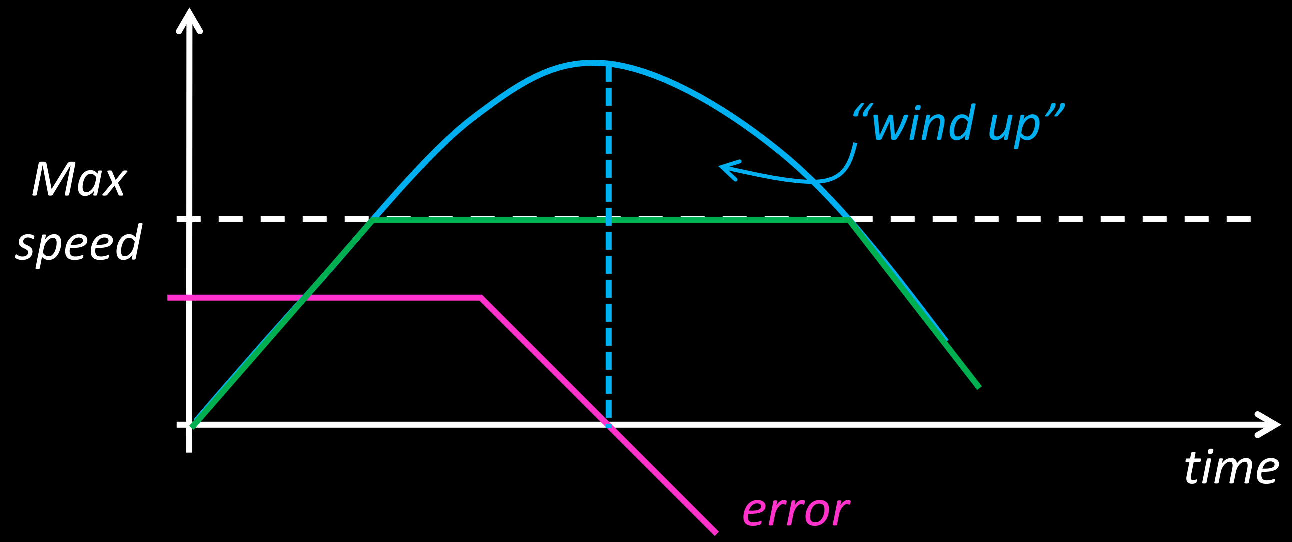
Real Systems are not linear!



Real systems are not linear!

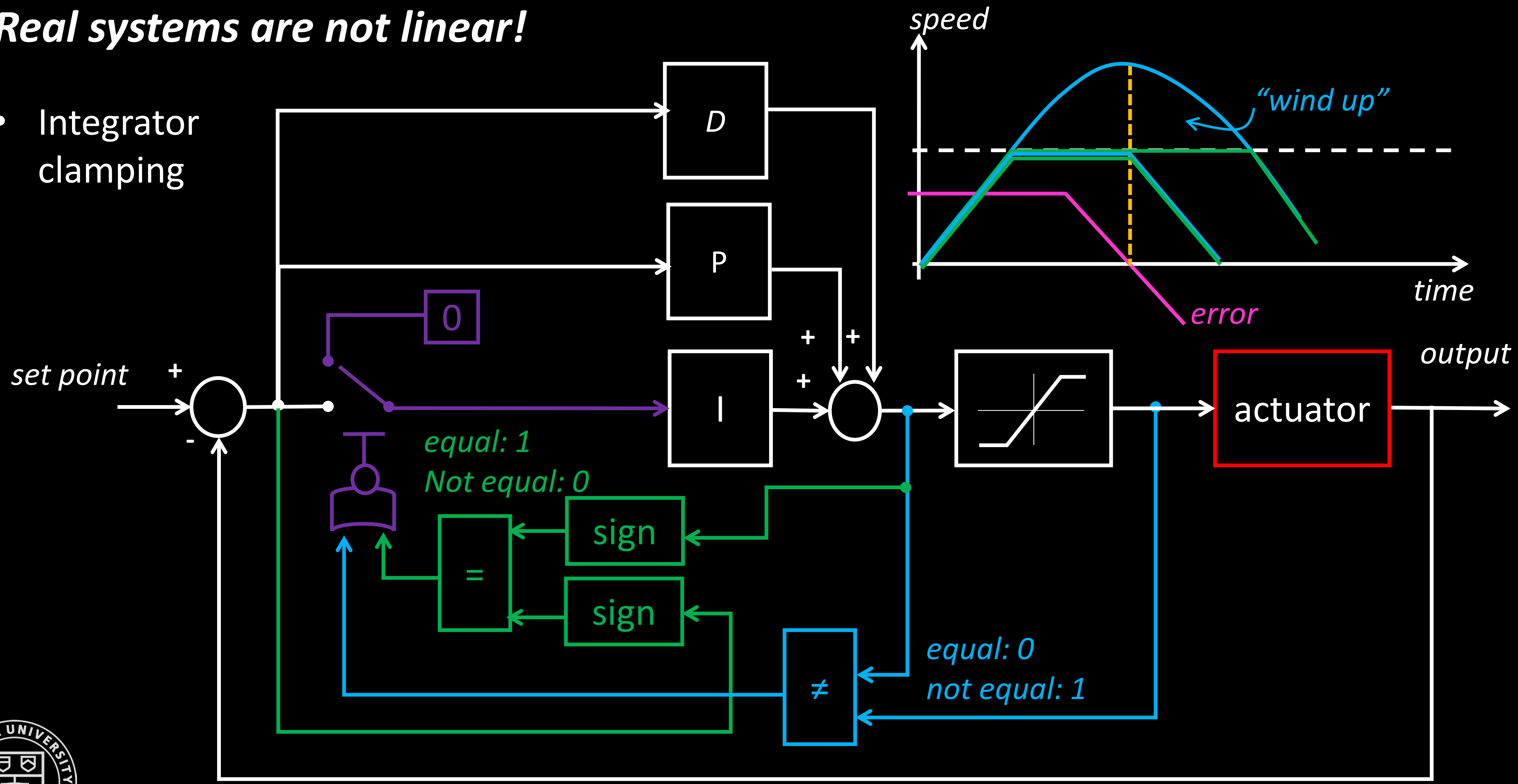
- Drone example
  - "Integral wind-up"
    - Clamping





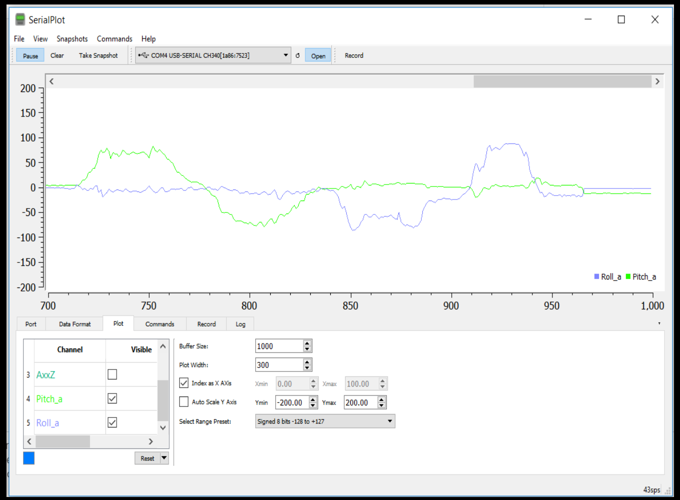
Real systems are not linear!

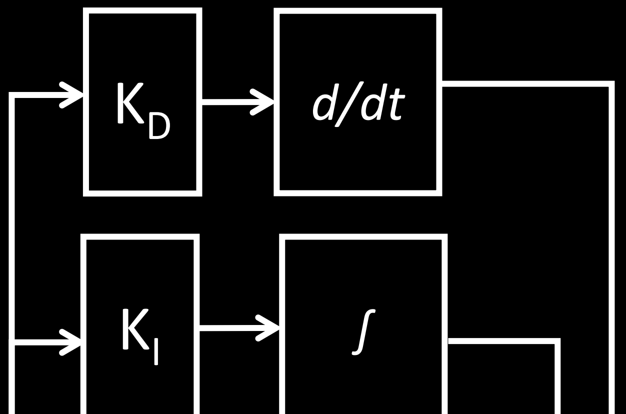
- Integrator clamping

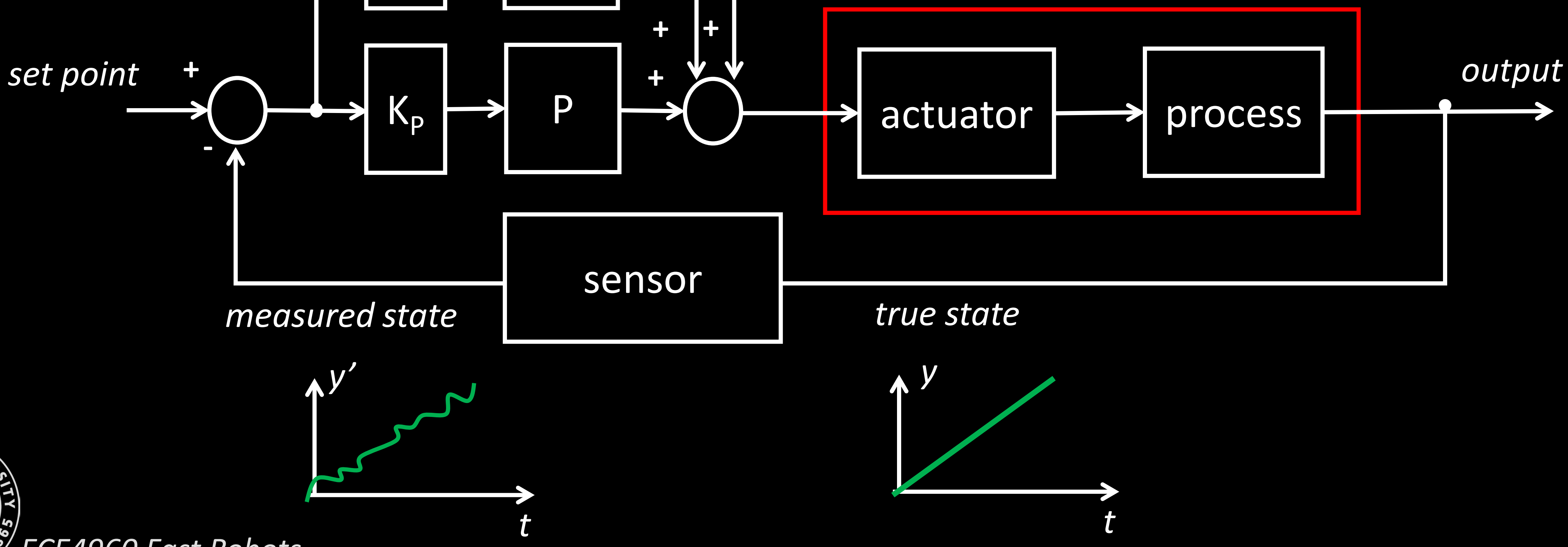


PID and Sensor noise

"noise":
- environment
- implementation
- defects

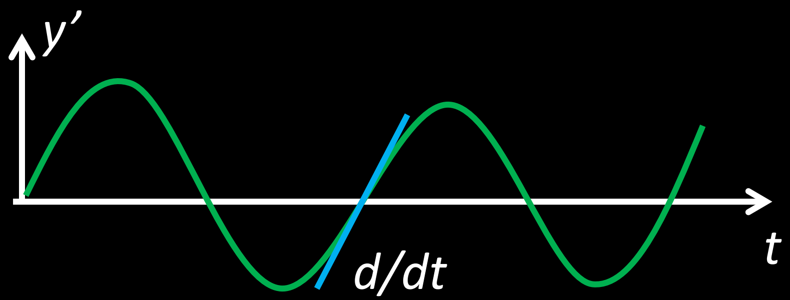


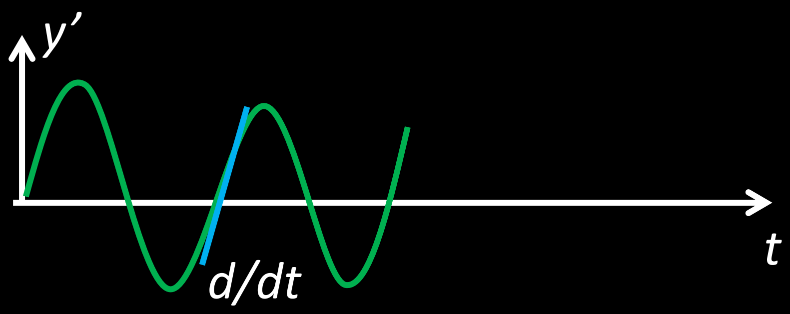




PID and Sensor noise

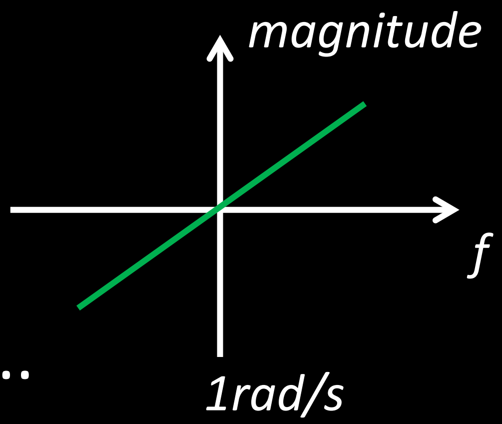
- Derivatives amplify HF signals more than LF signals





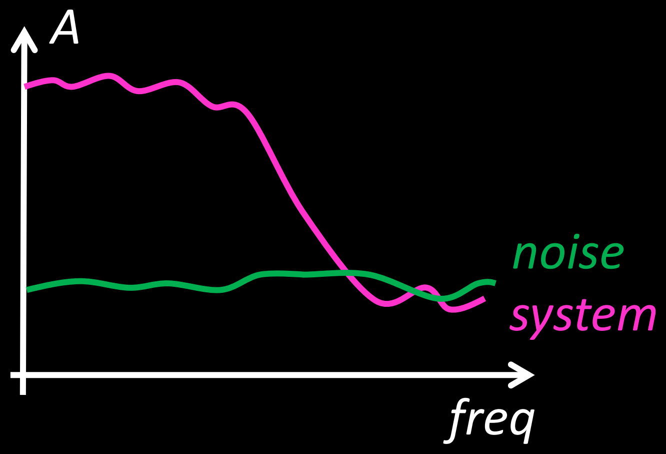
$$y(t) = A\sin(\omega_a t+\phi_a) + B\sin(\omega_b t+\phi_b) + \ldots$$

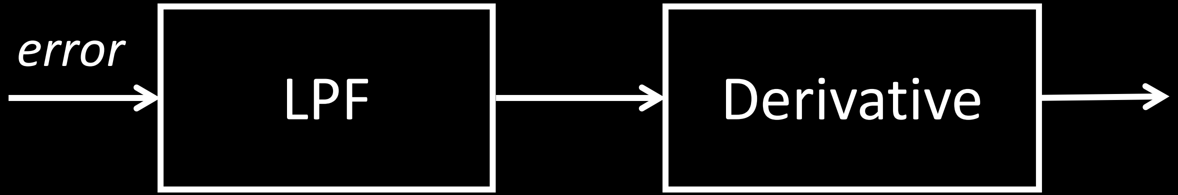
$$dy(t)/dt = A\omega_a\sin(\omega_a t+\phi_a+90^o) + B\omega_b\sin(\omega_b t+\phi_b+90^o) + \ldots.$$

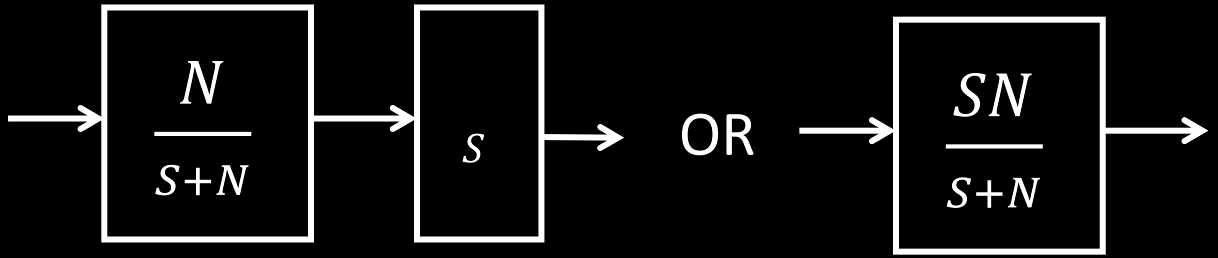


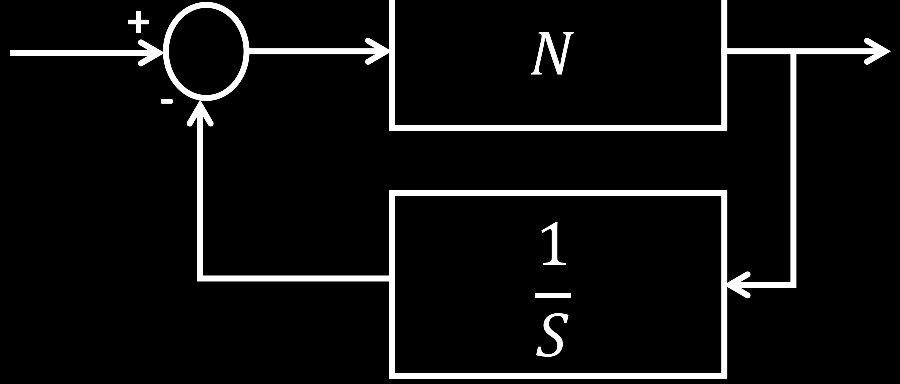
- if ω_a > 1rad/s, the amplitude will increase
- if ω_a < 1rad/s, the amplitude will decrease

PID and Sensor noise









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

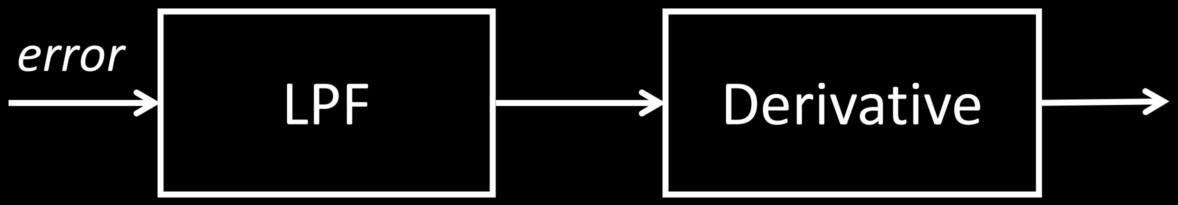
PID and Sensor noise

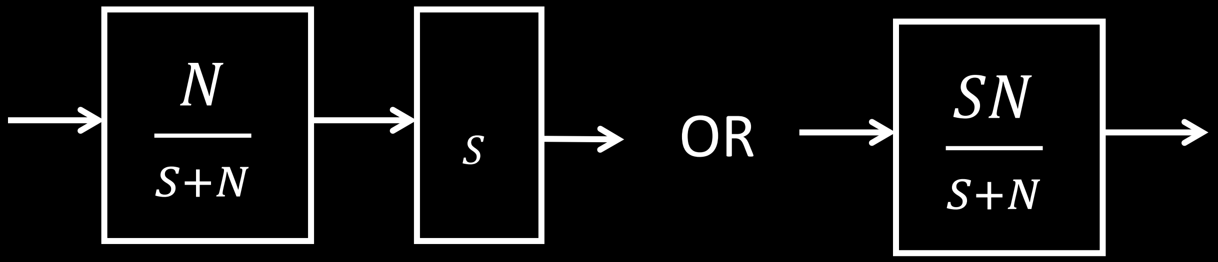
$$y = N\left(u - \frac{y}{s}\right)$$

$$y + \frac{Ny}{s} = Nu$$

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

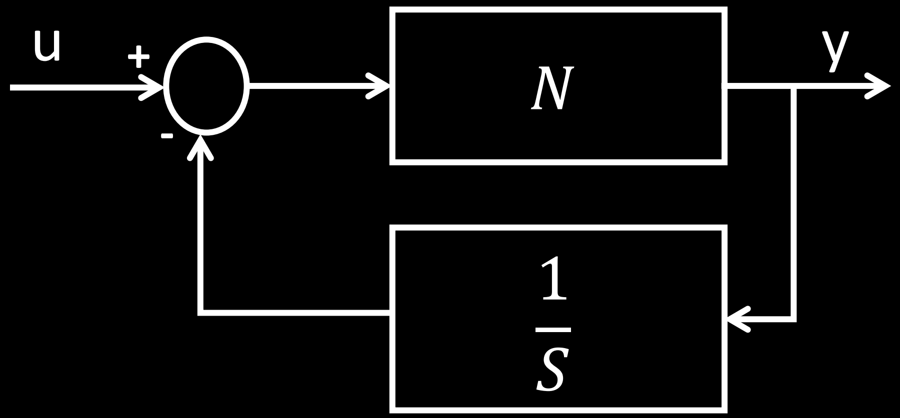
$$\frac{y}{u} = \frac{N}{1+N\frac{1}{s}}$$





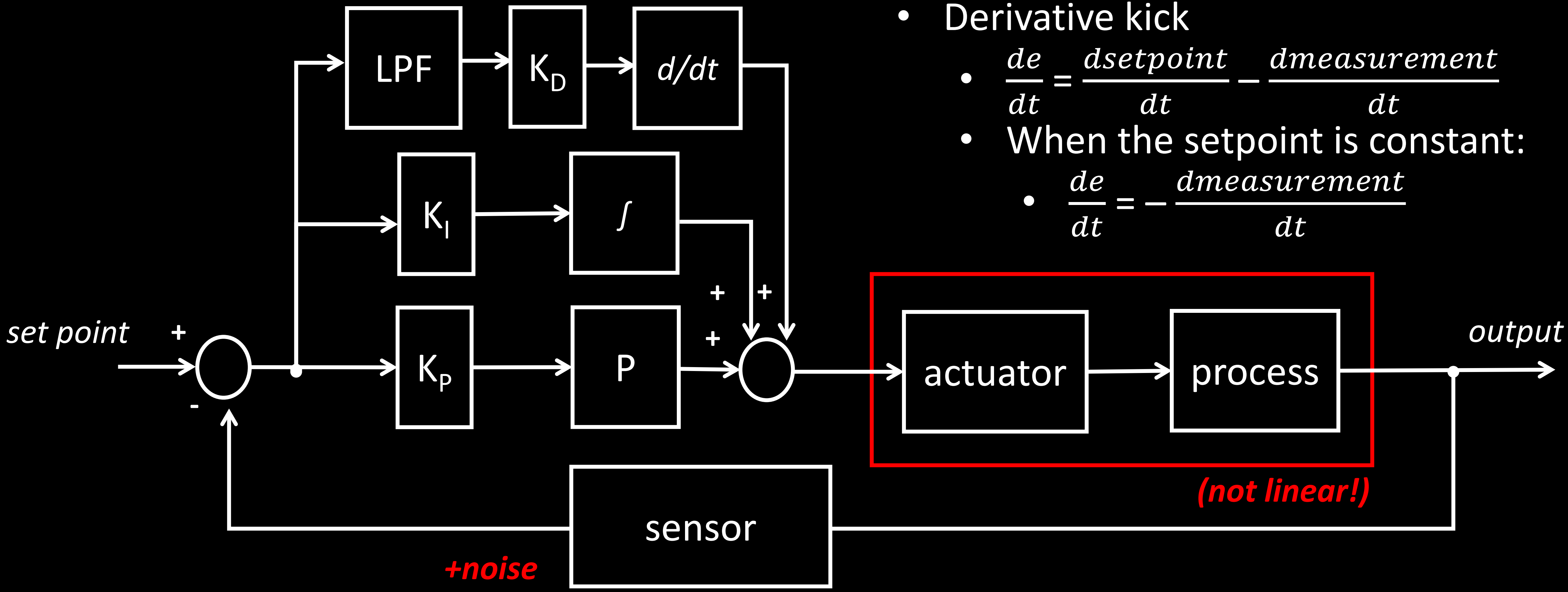
Time	Laplace
$\frac{d}{dt}$	S
$\int dt$	$\frac{1}{S}$

1st order LPF $\quad \frac{N}{S+N} = \frac{1}{\frac{1}{N}S+1} = \frac{1}{\tau S+1}$

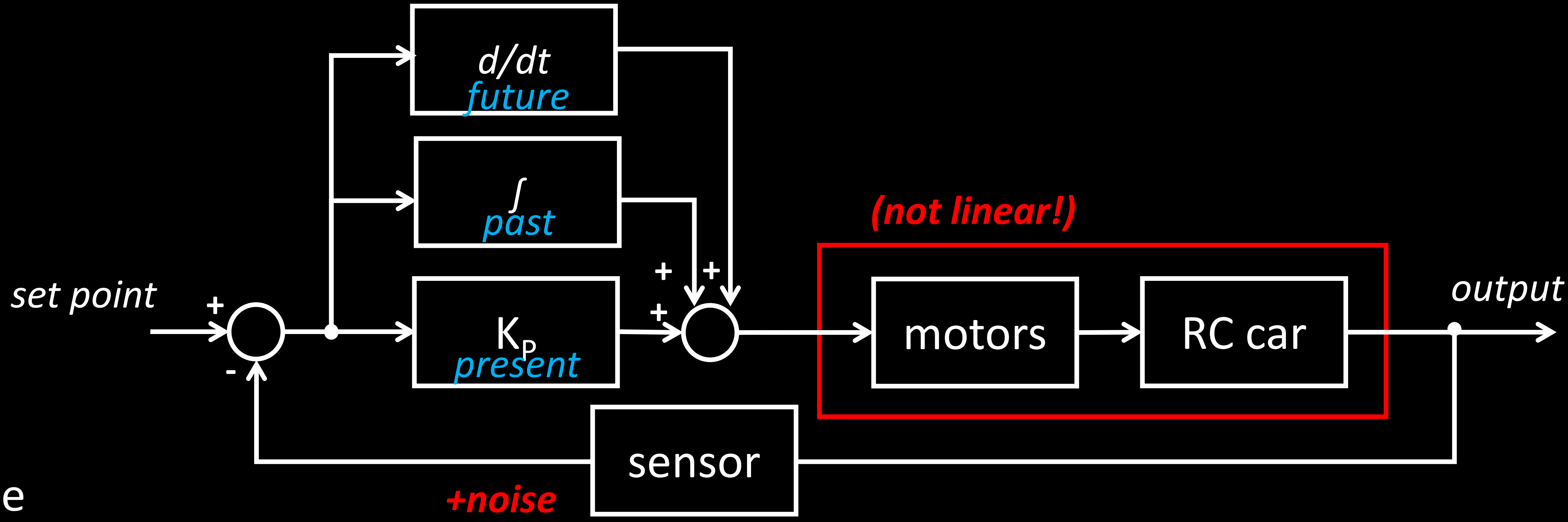


PID

- Integrator wind-up
- Derivative low pass filter
- Derivative kick
  - $\frac{de}{dt} = \frac{dsetpoint}{dt} - \frac{dmeasurement}{dt}$
  - When the setpoint is constant:
    - $\frac{de}{dt} = -\frac{dmeasurement}{dt}$



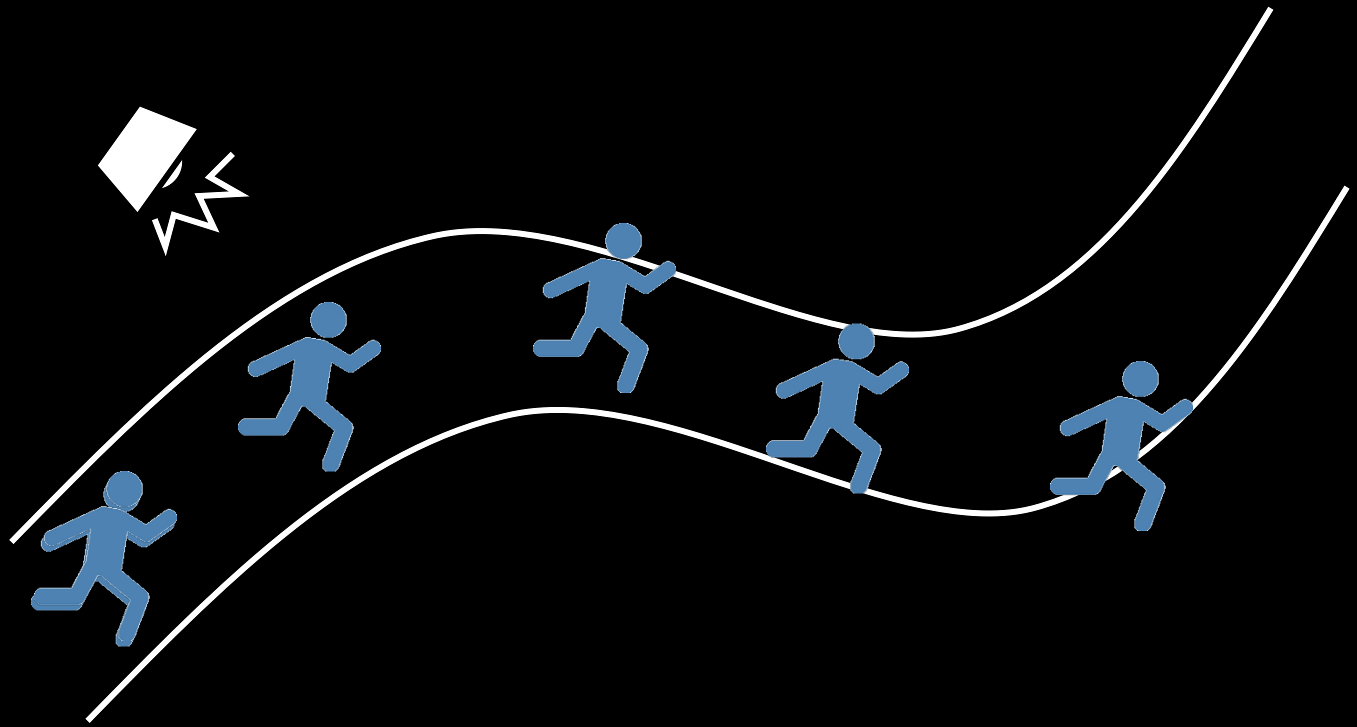
PID control



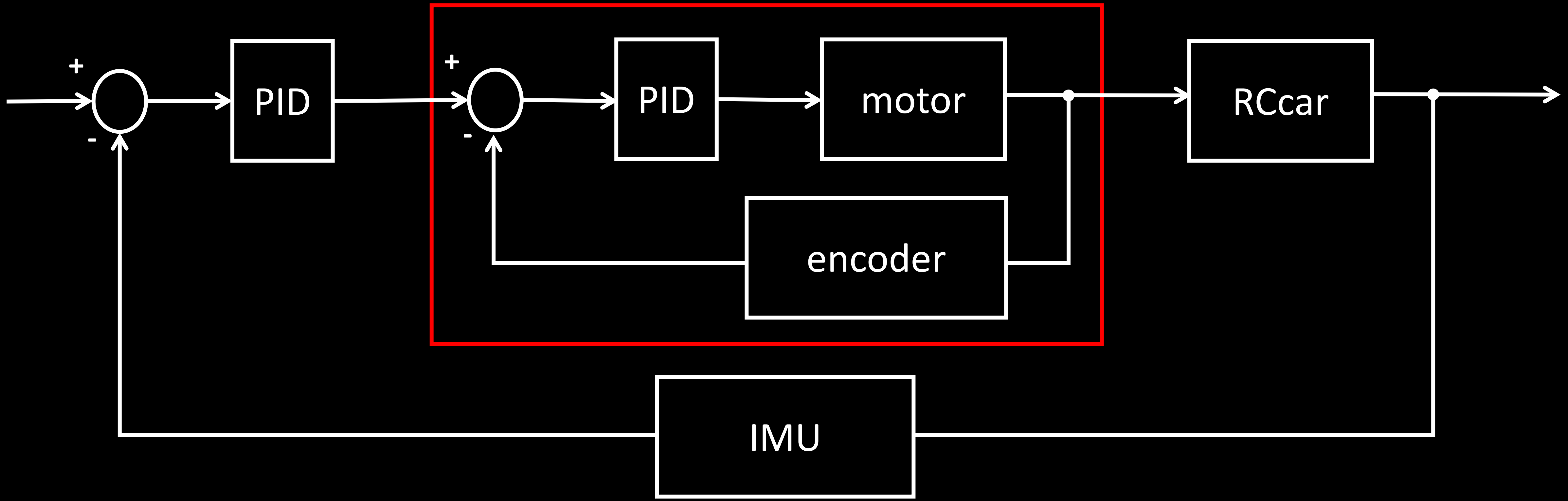
- Performance
  - Rise time/Response
    - Ex: 10% to 90% of final value
  - Peak time
    - Time to reach first peak
  - Overshoot
    - Amount in excess of final value
  - Settling time
    - Ex: Time before output settles to 1% of final value

Discrete PID Control

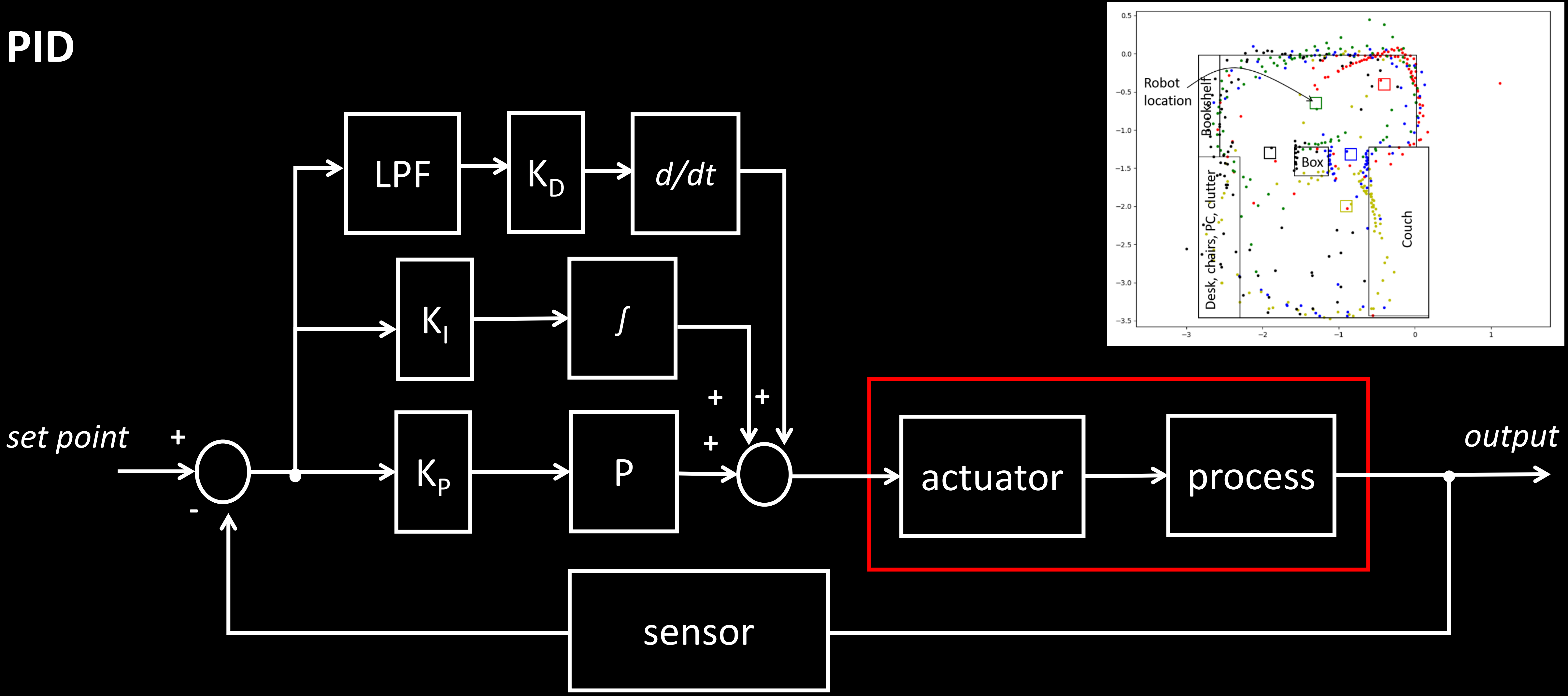
- Sampling time
- Control ~10 times faster than the system



Cascaded Control Loops

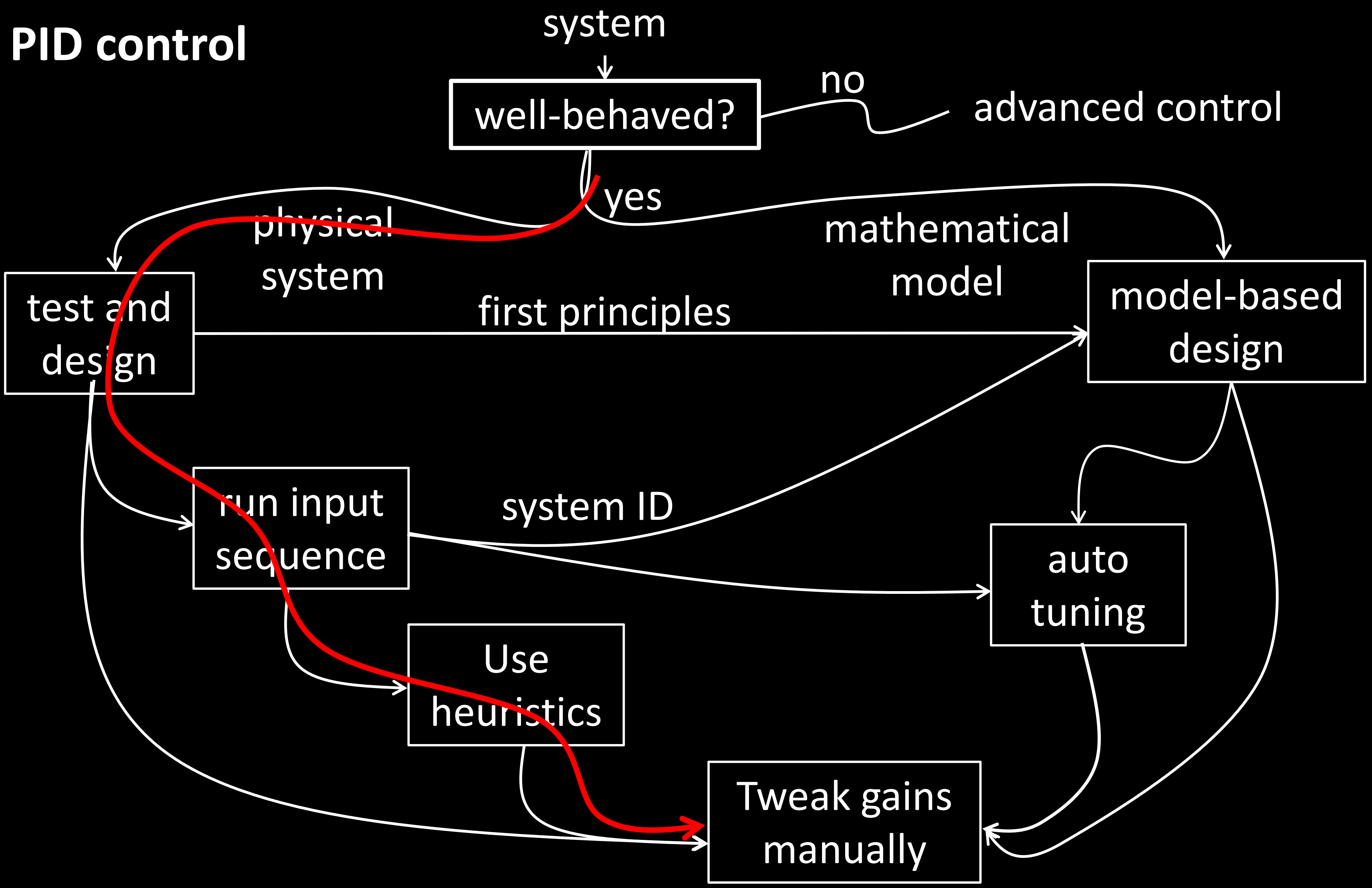


PID

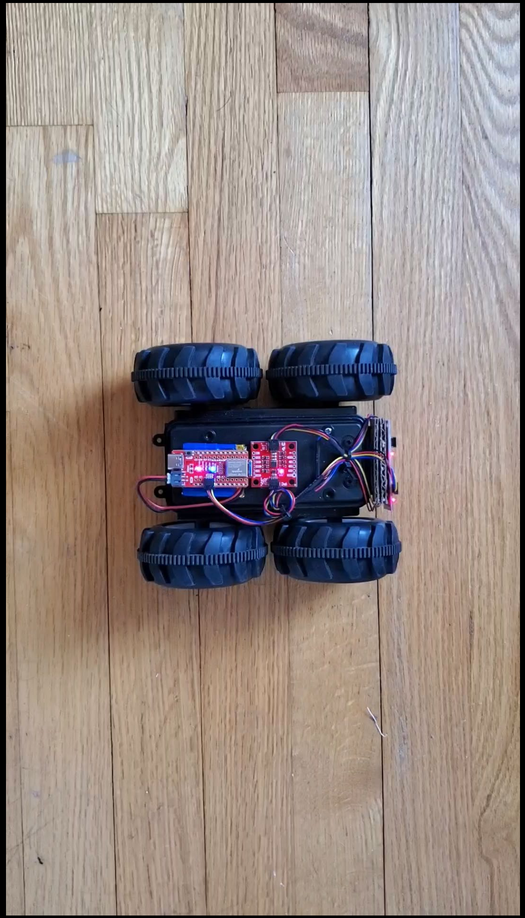


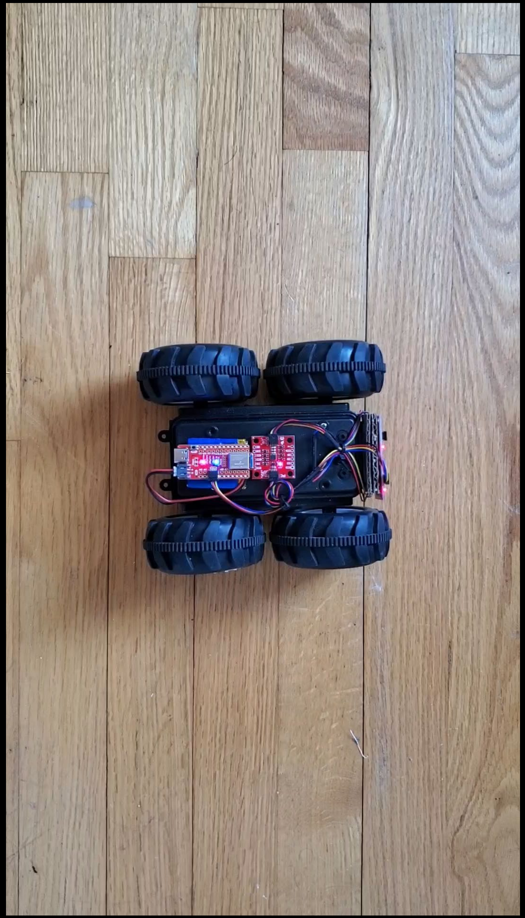
set point
+
-
LPF
K_D
d/dt
K_I
∫
K_P
P
+
+
+
actuator
process
output
sensor
Robot location
Bookshelf
Box
Couch
Desk, chairs, PC, clutter

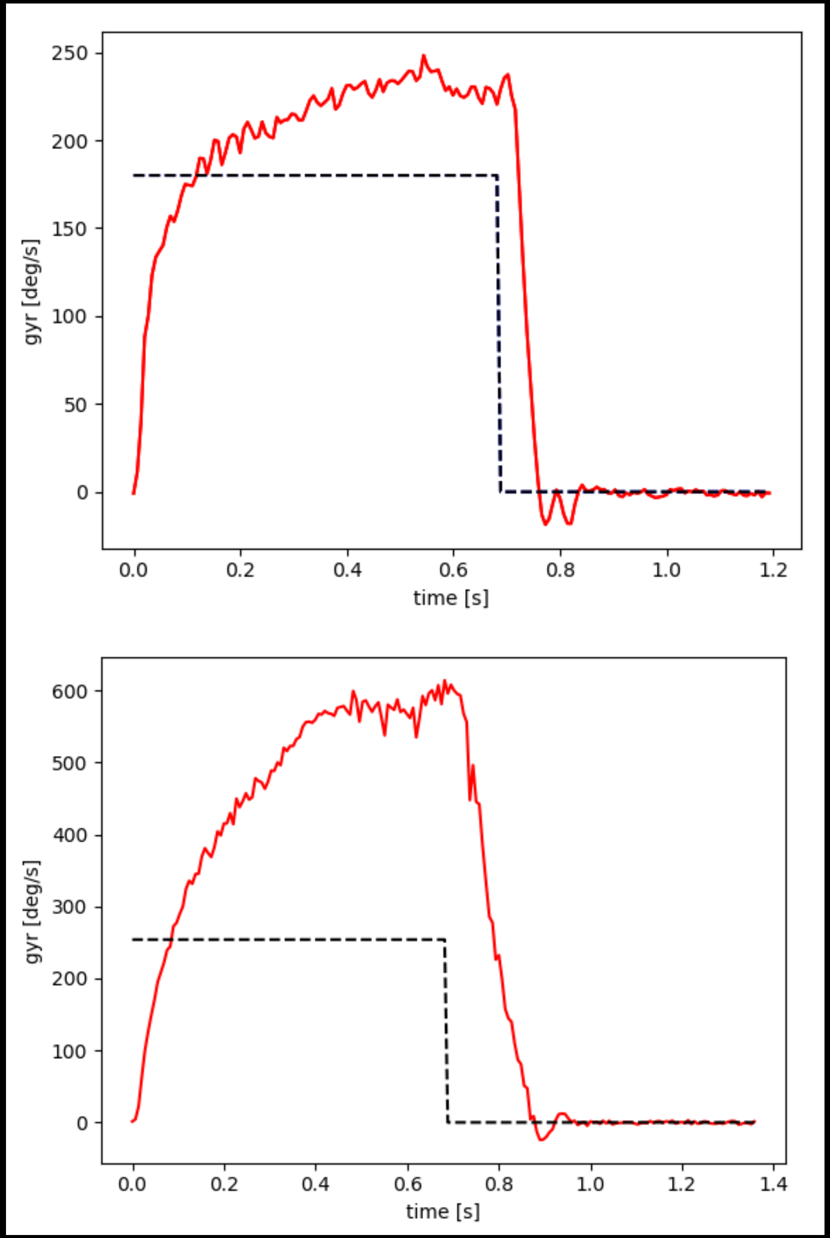
Tuning PID control



Tuning PID control







Tuning PID control

- Chien, Hornes, and Reswick method

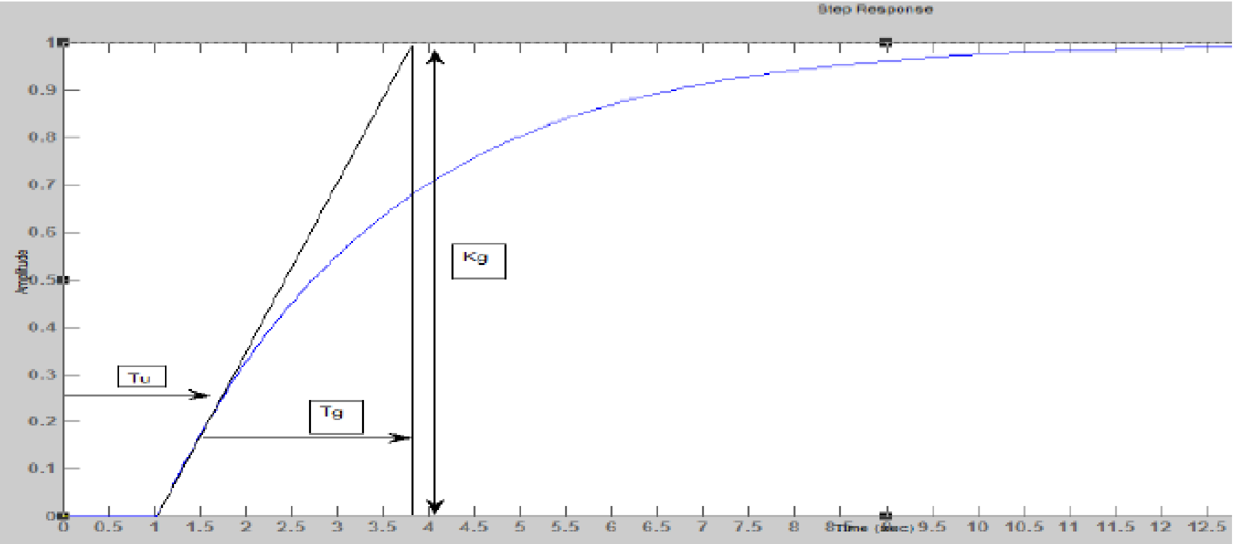
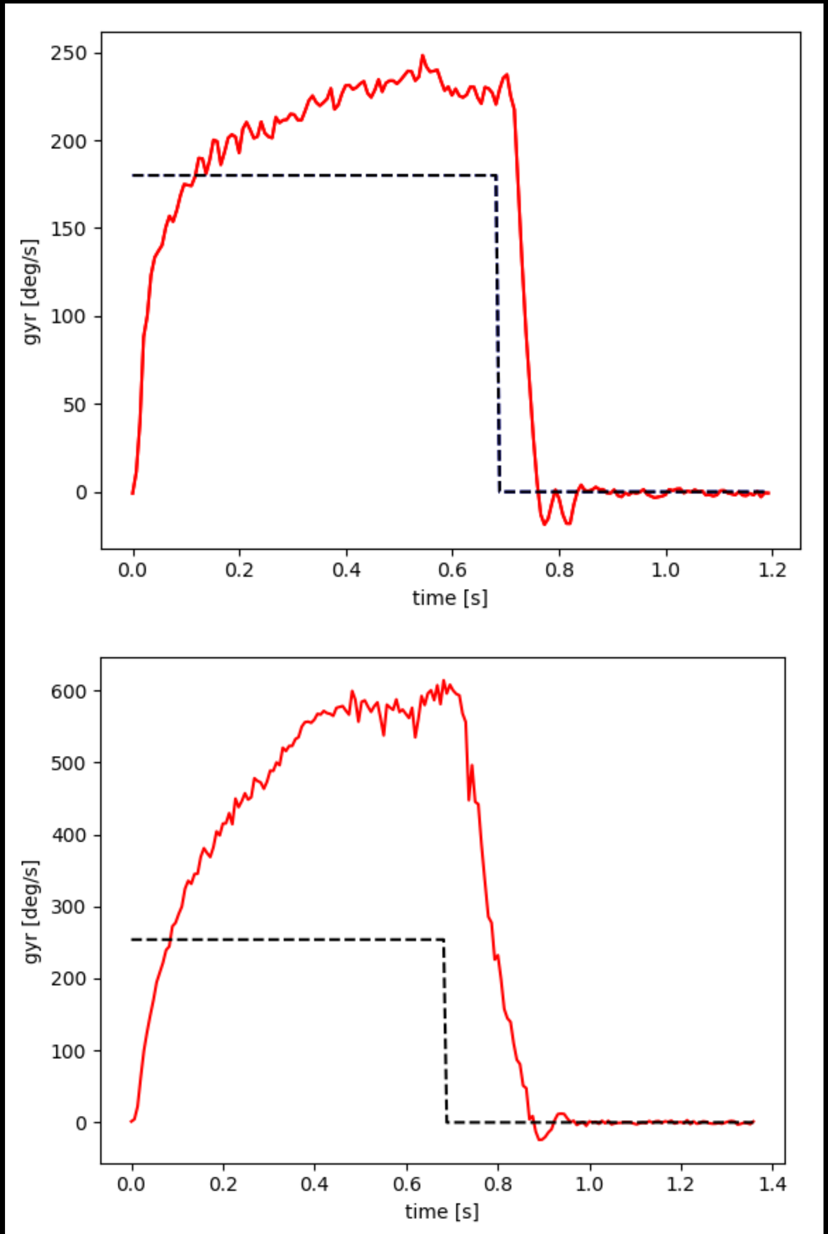


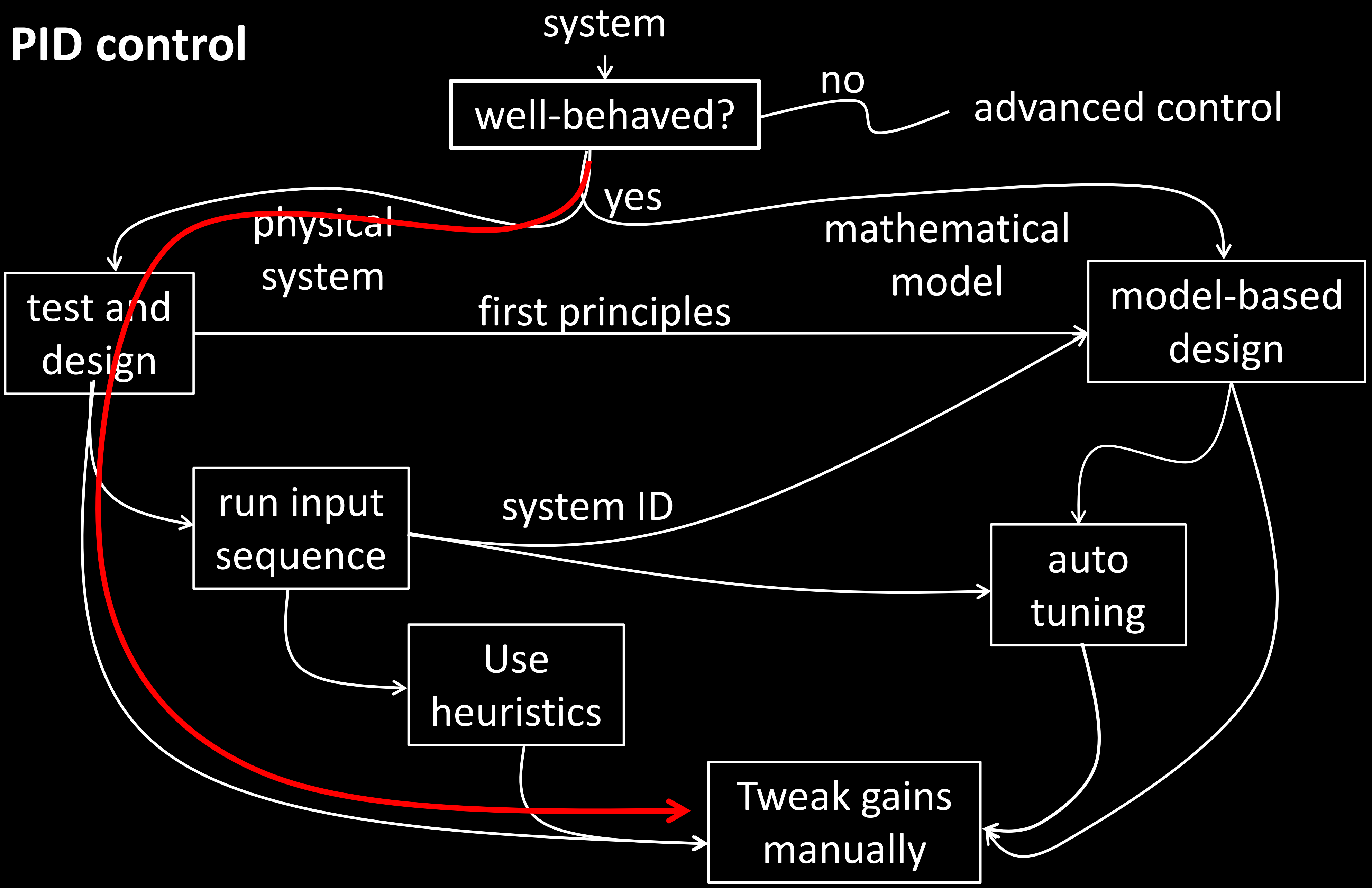
Fig.7. Open loop response of CHR method

Table.11. CHR Compensator

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



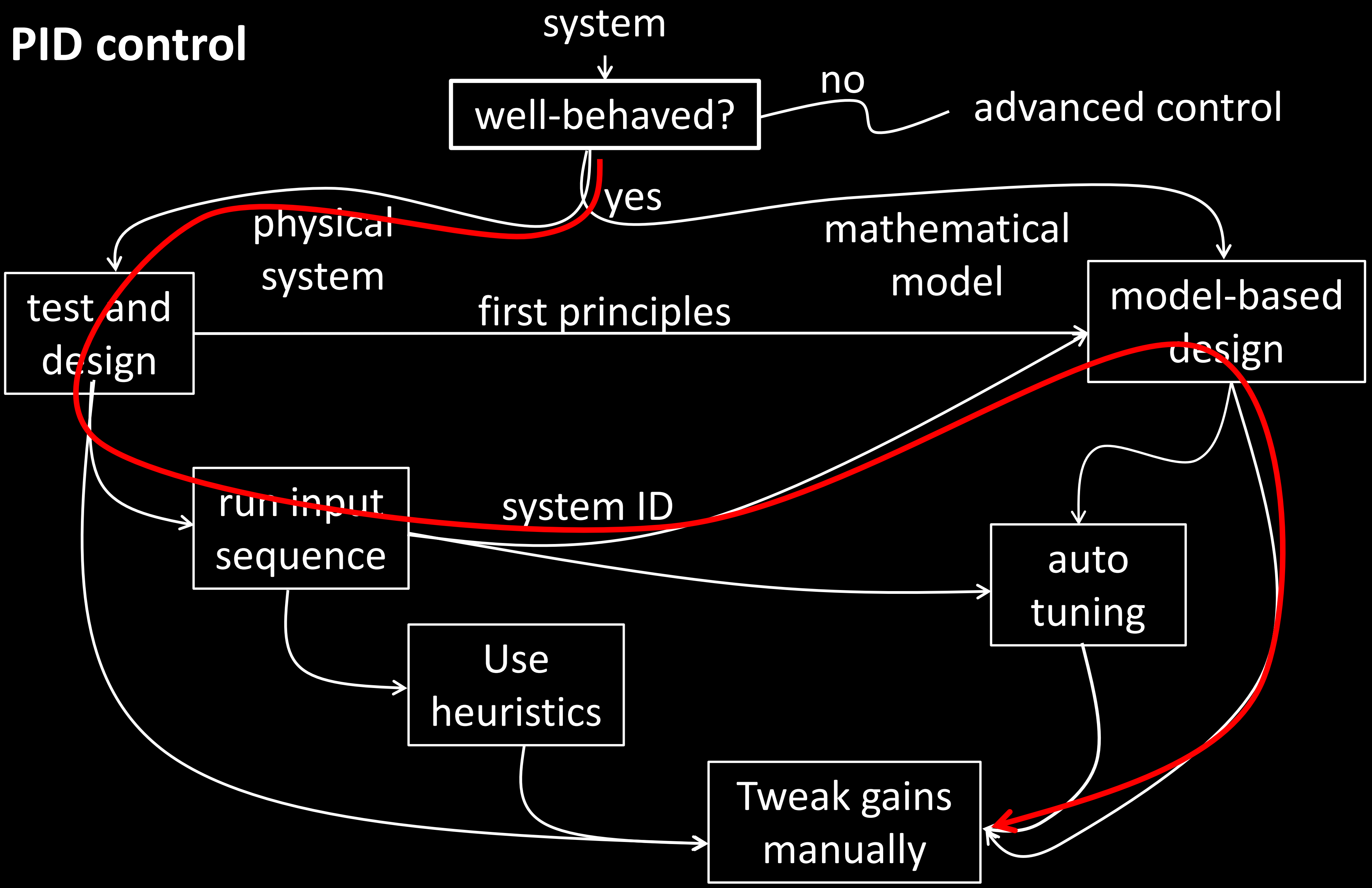
Tuning PID control



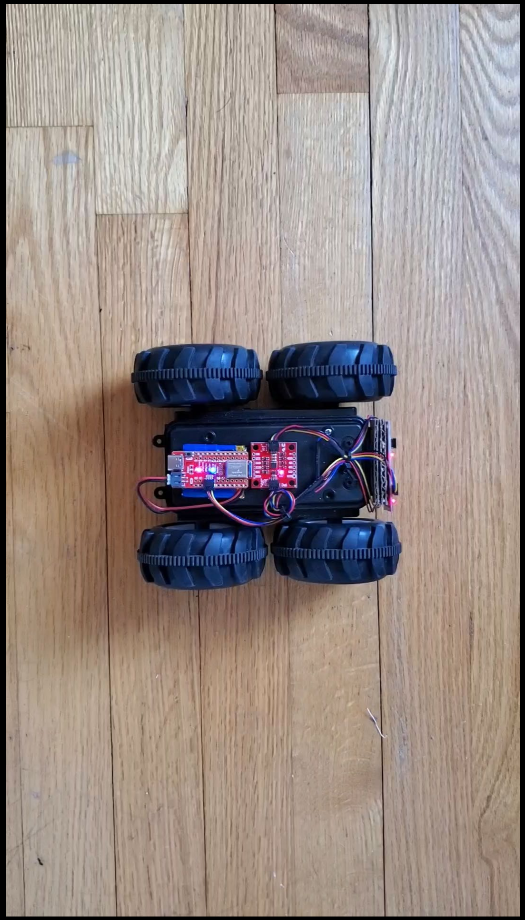
PID control

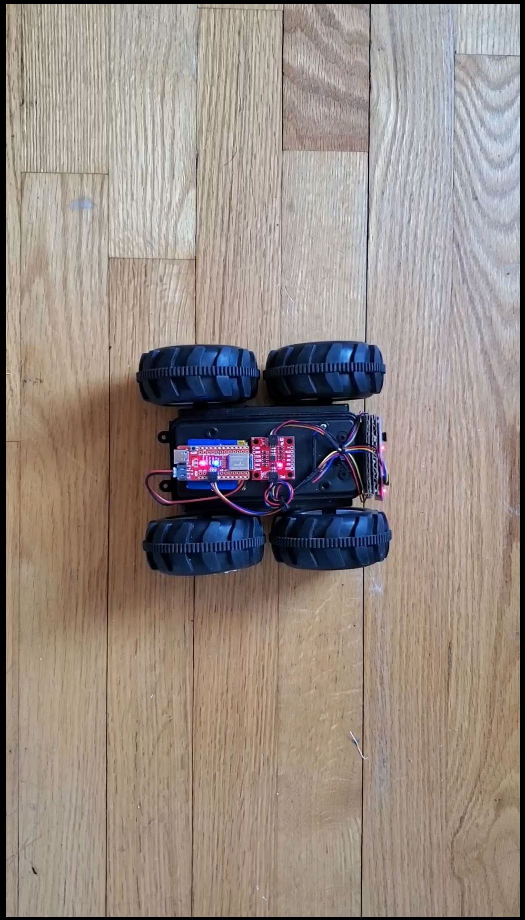
- **Heuristic procedure #1:**
  - Set Kp to small value, KD and KI to 0
  - Increase KD until oscillation, then decrease by factor of 2-4
  - Increase KP until oscillation or overshoot, decrease by factor of 2-4
  - Increase KI until oscillation or overshoot
  - Iterate
- **Heuristic procedure #2:**
  - 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

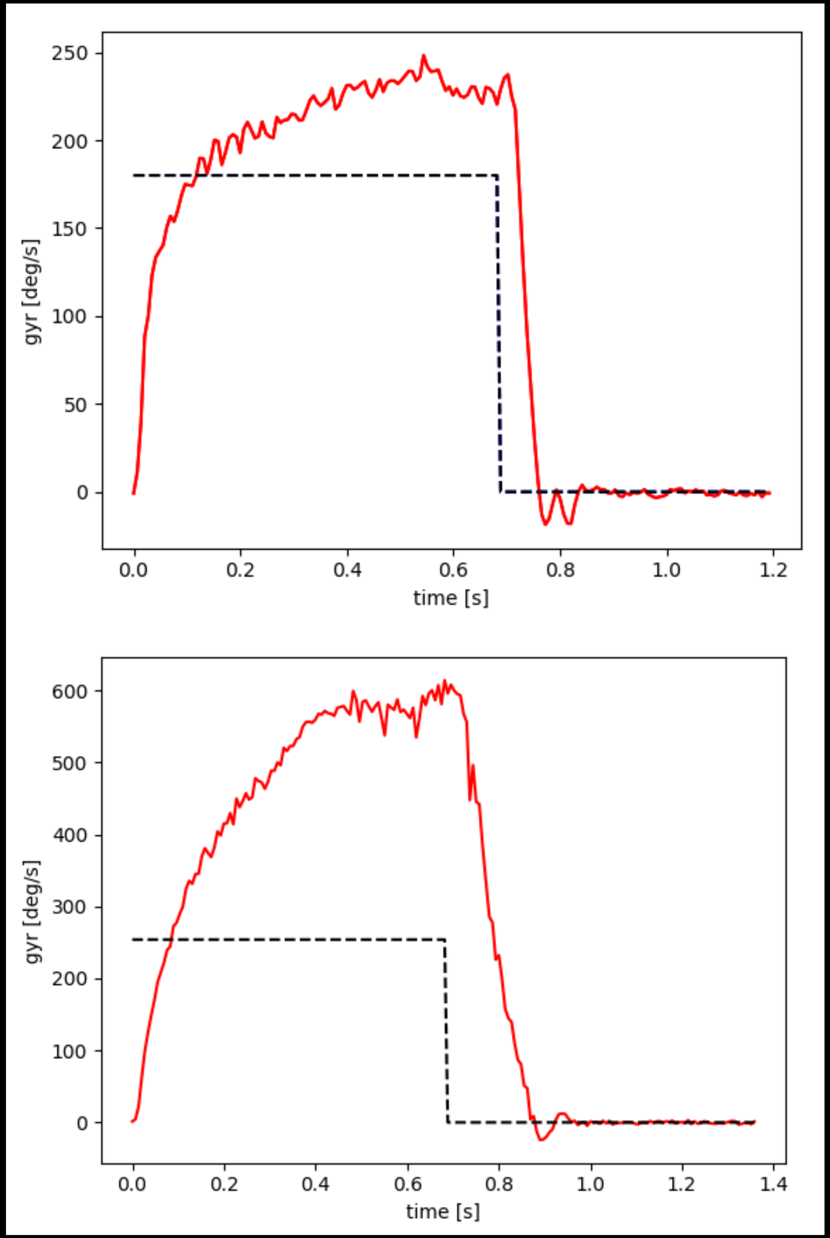
Tuning PID control



Tuning PID control

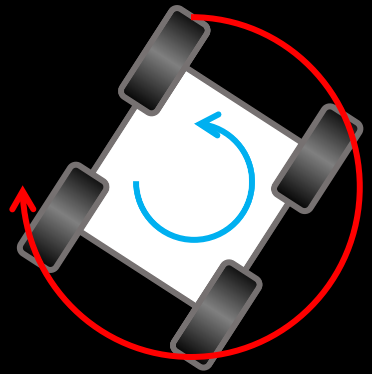






Tuning PID control

- Equations of motion
  - $x = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}$
- https://tinyurl.com/y67glgzk



$$F = ma$$

$$\tau = I\alpha$$

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

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

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

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