THE UNIVERSITY *of* EDINBURGH
informatics

Advanced Robotics

On the Design of Controllers
(Ref: Ch. 11 of K.M. Lynch & F.C. Park, *Modern Robotics*)

Subramanian Ramamoorthy
School of Informatics
University of Edinburgh

Types of Control Objectives

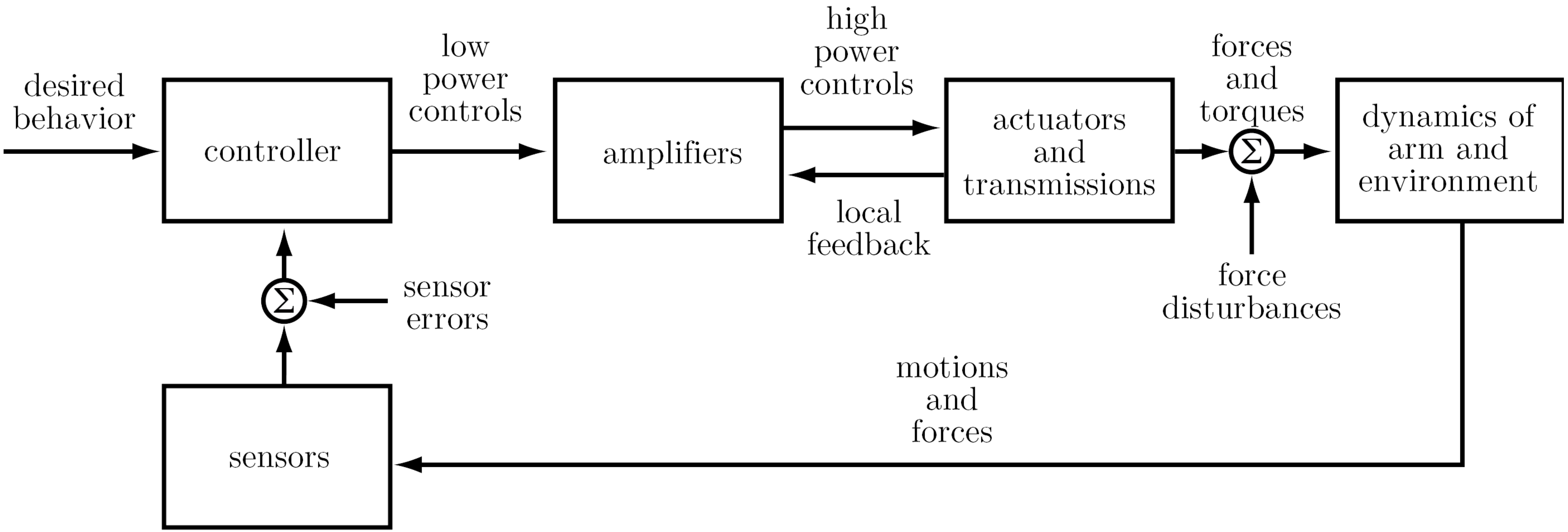
The same control structure (e.g. PD control) can be applied to many objectives:

- ❑ motion control
- ❑ force control
- ❑ hybrid motion-force control
- ❑ impedance control

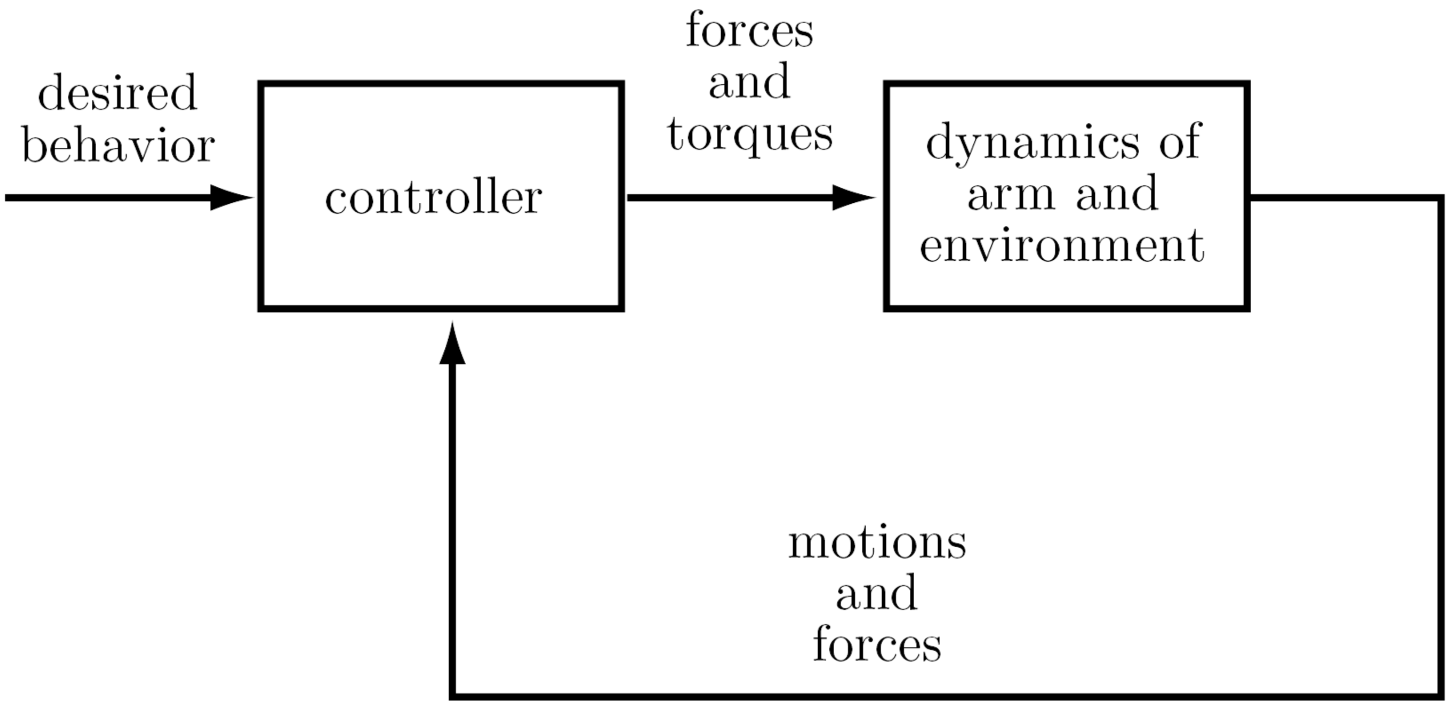
Consider the types of control for the following

- ❑ Shaking hands with a human
- ❑ Erasing a whiteboard
- ❑ Spray painting
- ❑ Back massage
- ❑ Pushing an object across the floor with a mobile robot
- ❑ Opening a refrigerator door
- ❑ Inserting a peg in a hole
- ❑ Polishing with a polishing wheel
- ❑ Folding laundry

Control System Block Diagram



A Simplified Block Diagram



Design: What do we Need to **<u>Deduce</u>** from Dynamics Models?

- Long-term dynamic behaviour
  - Stability: Will the dynamics converge? Will it come to rest?
  - Transient Response: How much will the state fluctuate in response to perturbations?
  - Given a certain family of control strategies, can this system be stabilized?
- Global Properties
  - Given nonlinearities, what kinds of phase space trajectories are possible?
  - What is the local structure along the various paths?

Design Concept: "Dynamic Response"

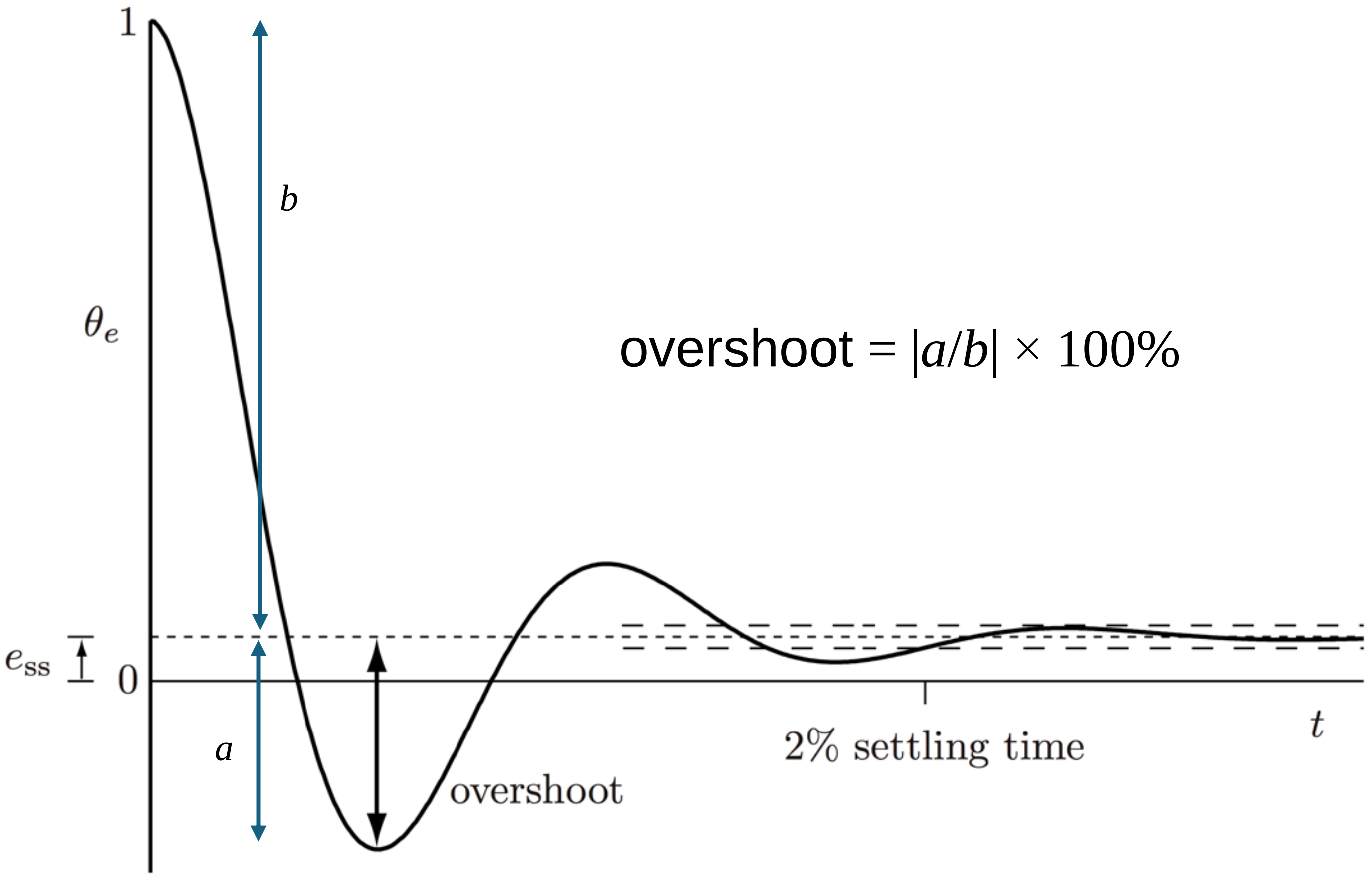
For motion control,

reference: $\theta_d(t)$
actual: $\theta(t)$
error: $\theta_e(t) = \theta_d(t) - \theta(t)$

Unit step error response:
$\theta_e(t)$ starting from $\theta_e(0) = 1$

Steady-state error response: e_{ss}

Transient error response:
overshoot, settling time



overshoot = $|a/b| \times 100\%$

Concept: **Error** Response

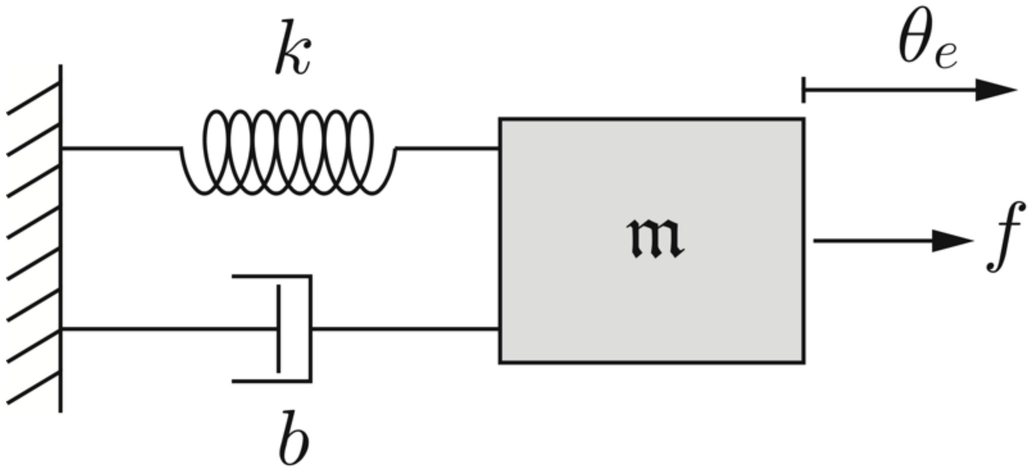
System dynamics, feedback controllers, and **error** response are often modeled by **linear ordinary differential equations**.

The simplest linear ODE exhibiting overshoot is second order, e.g.,

$$\mathfrak{m}\ddot{\theta}_e + b\dot{\theta}_e + k\theta_e = f$$

or, if $f = 0$,

$$\ddot{\theta}_e + \frac{b}{\mathfrak{m}}\dot{\theta}_e + \frac{k}{\mathfrak{m}}\theta_e = 0$$



k and b depend on the control law

A more general p^{th}-order linear ODE:

$$a_p\theta_e^{(p)} + a_{p-1}\theta_e^{(p-1)} + \cdots + a_2\ddot{\theta}_e + a_1\dot{\theta}_e + a_0\theta_e = c$$ **nonhomogenous**

$$a_p\theta_e^{(p)} + a_{p-1}\theta_e^{(p-1)} + \cdots + a_2\ddot{\theta}_e + a_1\dot{\theta}_e + a_0\theta_e = 0$$ **homogeneous**

$$\theta_e^{(p)} + a'_{p-1}\theta_e^{(p-1)} + \cdots + a'_2\ddot{\theta}_e + a'_1\dot{\theta}_e + a'_0\theta_e = 0$$

$$\theta_e^{(p)} = -a'_{p-1}\theta_e^{(p-1)} - \cdots - a'_2\ddot{\theta}_e - a'_1\dot{\theta}_e - a'_0\theta_e$$

Defining a state vector $x = (x_1, x_2, ..., x_p)$, you can write the p^{th}-order ODE as p first-order ODEs (a vector ODE).

$$
\begin{aligned}
x_1 &= \theta_e, \\
x_2 &= \dot{x}_1 = \dot{\theta}_e, \\
\vdots \quad & \quad \vdots \\
x_p &= \dot{x}_{p-1} = \theta_e^{(p-1)}
\end{aligned}
$$

$$\dot{x}(t) = Ax(t) \rightarrow x(t) = e^{At}x(0)$$

$$
A = \begin{bmatrix}
0 & 1 & 0 & \cdots & 0 & 0 \\
0 & 0 & 1 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 1 & 0 \\
0 & 0 & 0 & \cdots & 0 & 1 \\
-a'_0 & -a'_1 & -a'_2 & \cdots & -a'_{p-2} & -a'_{p-1}
\end{bmatrix} \in \mathbb{R}^{p \times p}
$$

$$\dot{x}(t) = Ax(t) \rightarrow x(t) = e^{At}x(0)$$

If $\mathrm{Re}(s) < 0$ for all eigenvalues s of A, then the error dynamics are **stable** (the error decays to zero).

The eigenvalues are the roots of the **characteristic equation**

$$\det(sI - A) = s^p + a'_{p-1}s^{p-1} + \cdots + a'_2 s^2 + a'_1 s + a'_0 = 0$$

Necessary conditions for stability: each $a'_i > 0$.

These necessary conditions are also **sufficient** for first- and second-order systems.

Discuss:

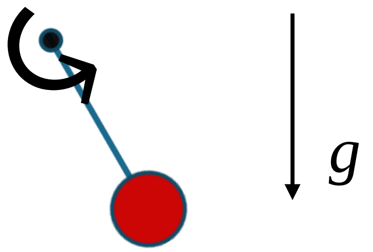
If the error dynamics characteristic equation is $(s + 3 + 2j)(s + 3 - 2j)(s - 2) = 0$, does the error converge to zero?

Note: if x_1 = error and $x = (x_1, x_2, x_3)$, then $\dot{x} = Ax$, where

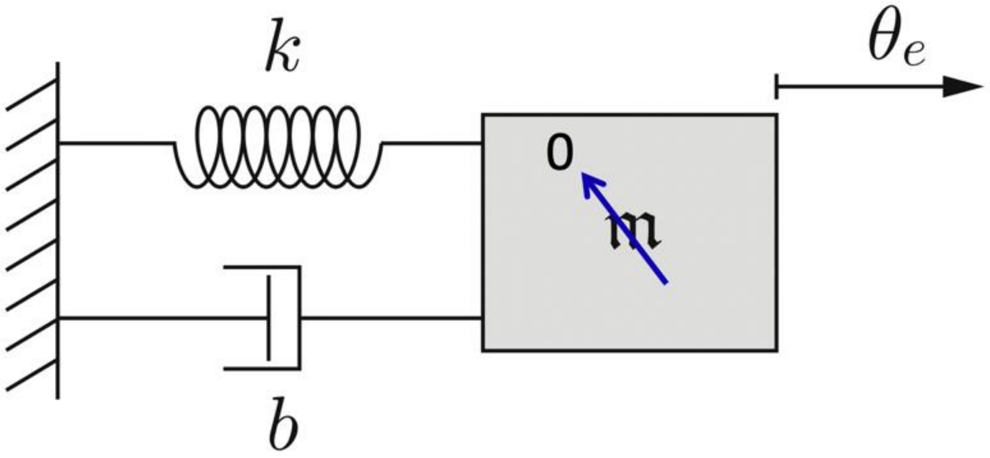
$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 26 & -1 & -4 \end{bmatrix}$$

Discuss:

You can choose a control law to be a virtual spring, a virtual damper, a virtual spring plus damper, or nothing. Which of these could stabilize an actuated pendulum with viscous friction to the upright configuration? To a horizontal configuration? To the downward configuration? Describe the transient and steady-state error response for each.



First-order **Error** Dynamics



0

$$\mathfrak{m}\ddot{\theta}_e + b\dot{\theta}_e + k\theta_e = 0$$

$$\dot{\theta}_e(t) + \frac{k}{b}\theta_e(t) = 0$$

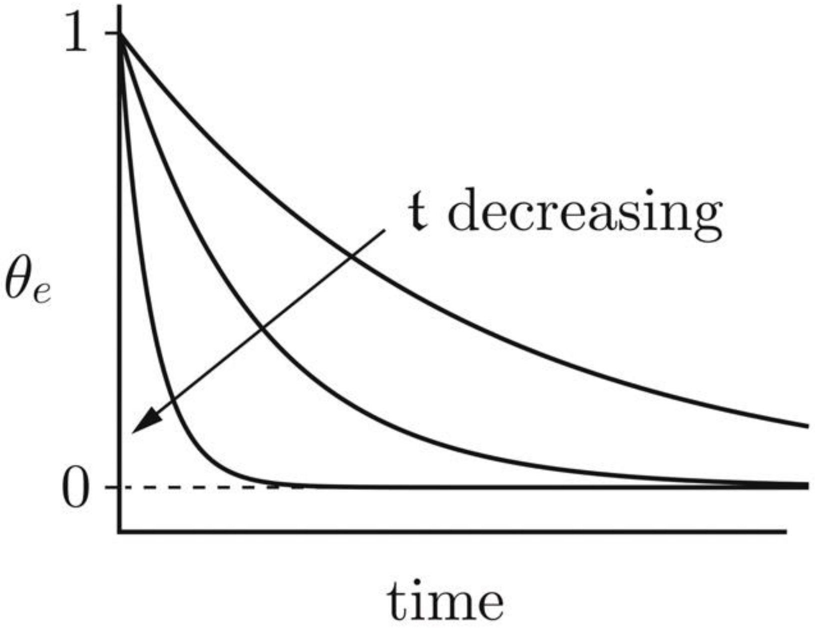
standard first-order form

time constant

$$\mathfrak{t} = b/k$$

$$\dot{\theta}_e(t) + \frac{1}{\mathfrak{t}}\theta_e(t) = 0$$

First-order **Error** Dynamics



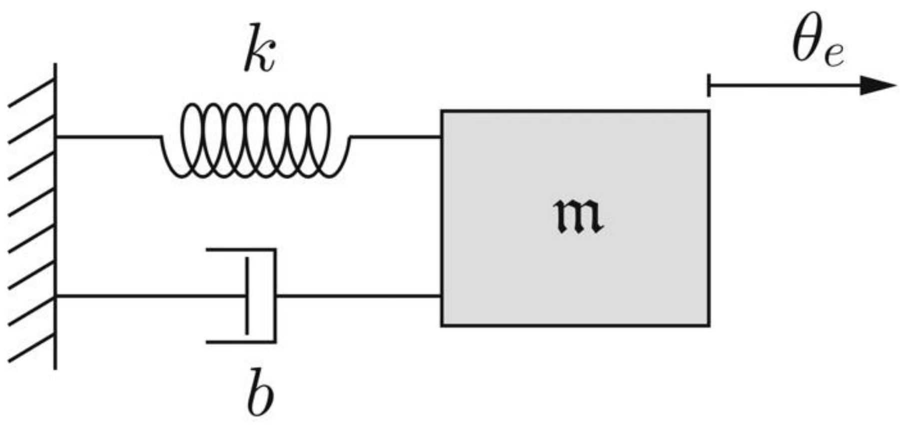
$$\theta_e(t) = e^{-t/\tau}\theta_e(0)$$

$$\theta_e(0) = 1$$

$$\frac{\theta_e(t)}{\theta_e(0)} = 0.02 = e^{-t/\tau}$$

$$\ln 0.02 = -t/\tau \quad \rightarrow \quad t = 3.91\tau$$

Second-order Error Dynamics



$$\ddot{\theta}_e(t) + \frac{b}{\mathfrak{m}}\dot{\theta}_e(t) + \frac{k}{\mathfrak{m}}\theta_e(t) = 0$$

natural frequency **damping ratio**

$$\omega_n = \sqrt{k/\mathfrak{m}} \qquad \zeta = b/(2\sqrt{k\mathfrak{m}})$$

$$\ddot{\theta}_e(t) + 2\zeta\omega_n\dot{\theta}_e(t) + \omega_n^2\theta_e(t) = 0$$

standard second-order form

Second-order **Error** Dynamics

$$\ddot{\theta}_e(t) + 2\zeta\omega_n\dot{\theta}_e(t) + \omega_n^2\theta_e(t) = 0$$

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$$

$$s_{1,2} = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1}$$

$\zeta > 1$: Overdamped

$\zeta = 1$: Critically damped

$\zeta < 1$: Underdamped

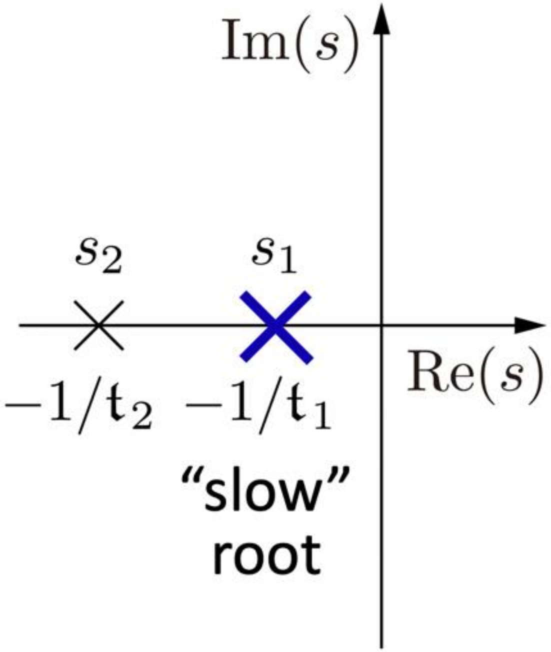
Overdamped behaviour

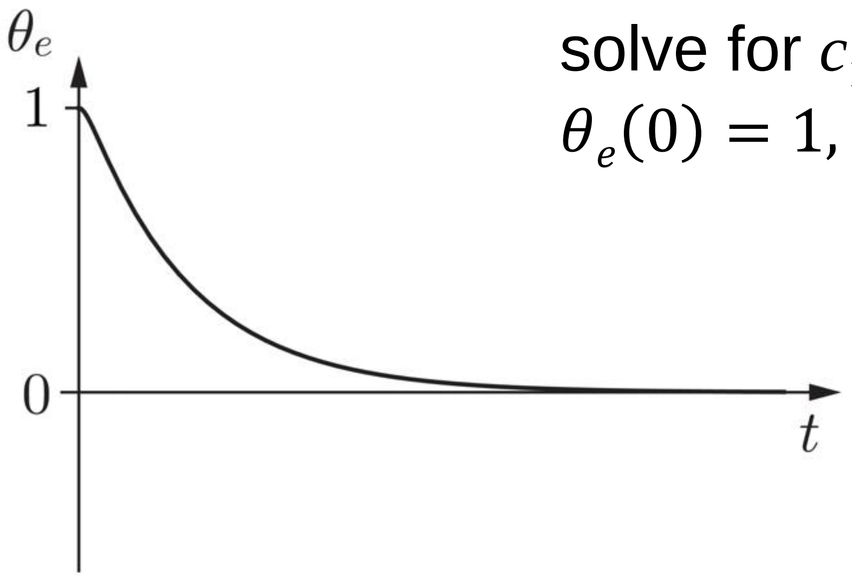
$\zeta > 1$: Overdamped

$$\theta_e(t) = c_1 e^{s_1 t} + c_2 e^{s_2 t}$$

$$s_1 = -\zeta\omega_n + \omega_n\sqrt{\zeta^2 - 1}$$

$$s_2 = -\zeta\omega_n - \omega_n\sqrt{\zeta^2 - 1}$$





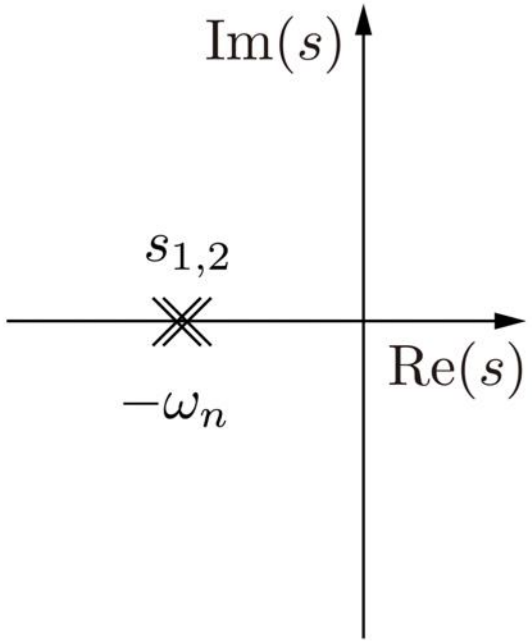
solve for c_1 and c_2 using
$\theta_e(0) = 1, \dot{\theta}_e(0) = 0$

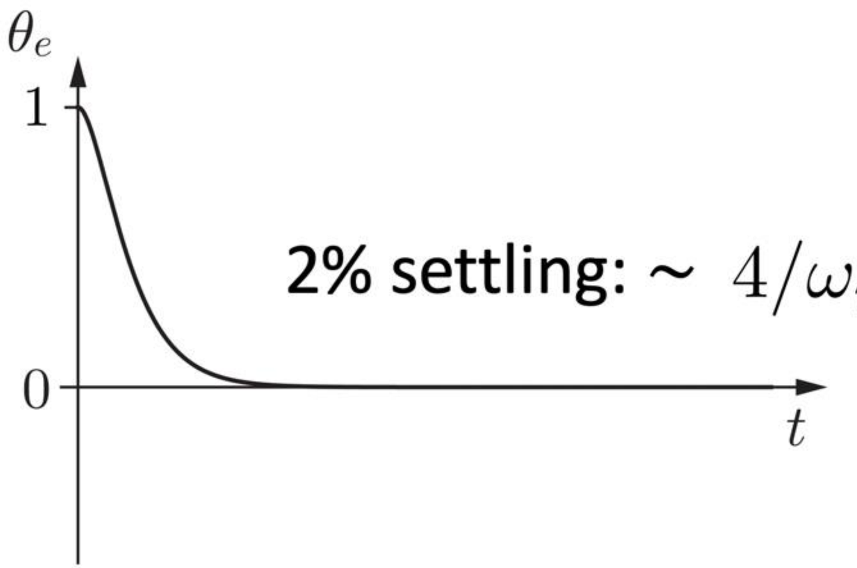
Critically damped behaviour

$\zeta = 1:$ Critically damped

$$\theta_e(t) = (c_1 + c_2 t)e^{-\omega_n t}$$

$$s_{1,2} = -\omega_n$$





2% settling: ~ $4/\omega_n$

Underdamped behaviour

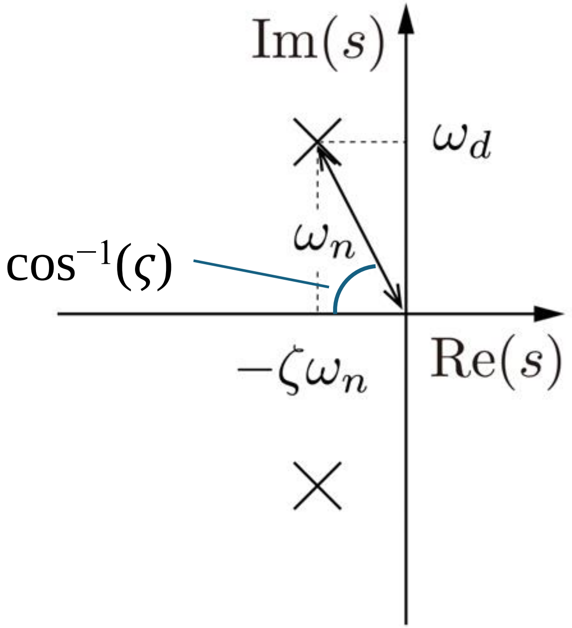
$\zeta < 1$: **Underdamped**

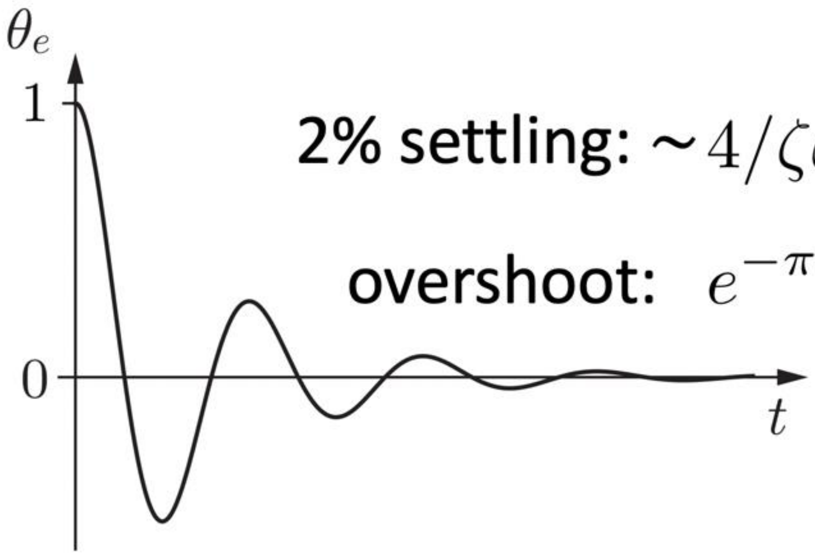
$$\theta_e(t) = (c_1 \cos \omega_d t + c_2 \sin \omega_d t)\, e^{-\zeta \omega_n t}$$

damped natural frequency

$$\omega_d = \omega_n \sqrt{1 - \zeta^2}$$

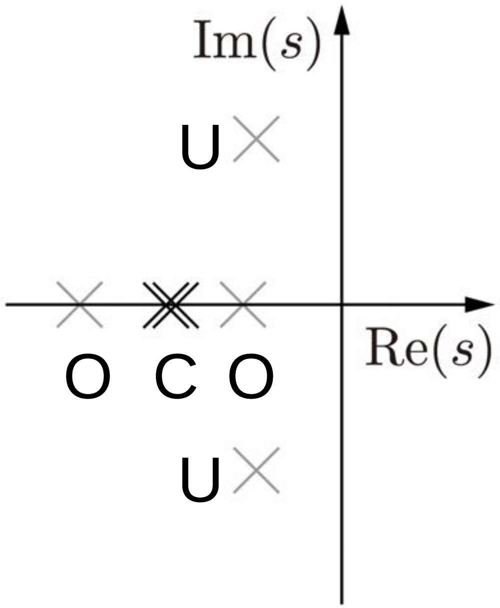
$$s_{1,2} = -\zeta \omega_n \pm j \omega_d$$

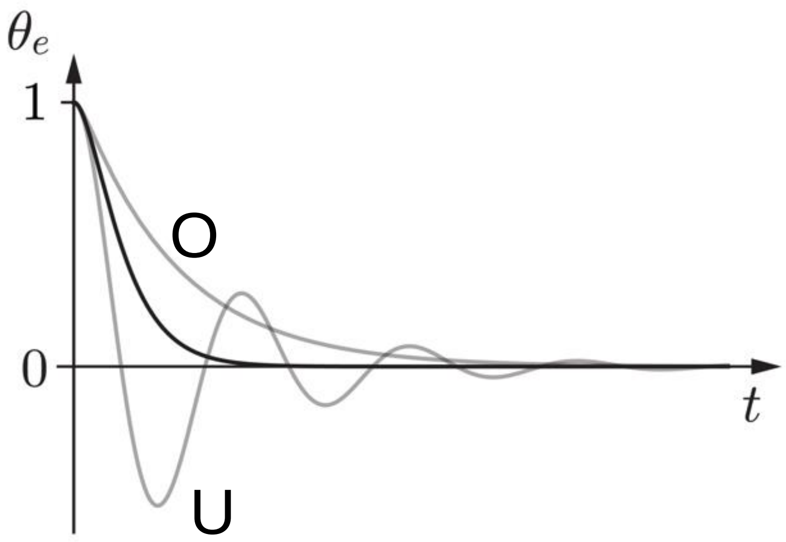




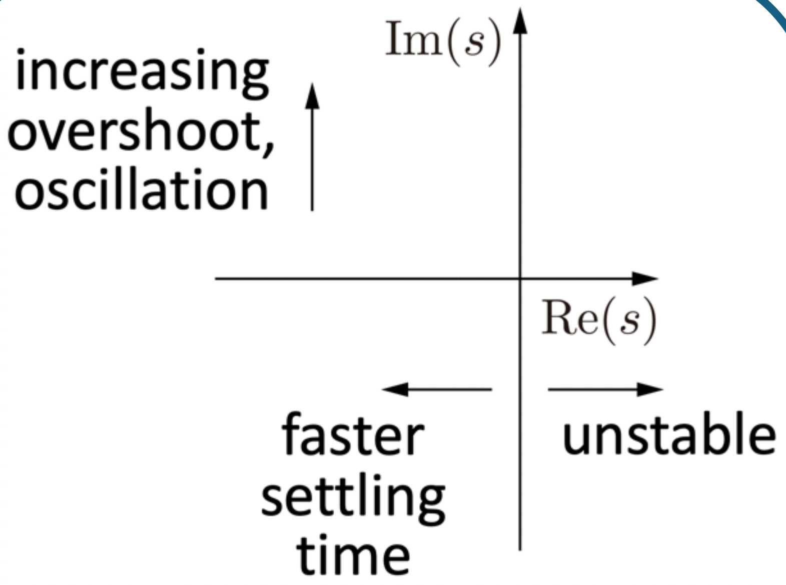
2% settling: $\sim 4/\zeta\omega_n$

overshoot: $e^{-\pi\zeta/\sqrt{1-\zeta^2}} \times 100\%$





second-order systems



time response relative to root locations for general systems

$$\omega_n = \sqrt{k/\mathfrak{m}} \qquad \zeta = b/(2\sqrt{k\mathfrak{m}})$$

2% settling: $\sim 4/\zeta\omega_n$

$$\ddot{\theta}_e(t) + 2\zeta\omega_n\dot{\theta}_e(t) + \omega_n^2\theta_e(t) = 0$$

overshoot: $e^{-\pi\zeta/\sqrt{1-\zeta^2}} \times 100\%$

When controlling a robot joint, what do b, k, and $\mathfrak{m}$ usually correspond to?

How do you change $\mathfrak{m}$ to decrease settling time? k, b?

How do you change $\mathfrak{m}$ to decrease overshoot? k, b?

Back to the PID controller

Let error be $e = x_{ref} - x$, PID controller in continuous time

$$\boldsymbol{u}(t) = k_p e + k_d \dot{e} + k_I \int e \quad dt$$

Recall, elements of the PID:

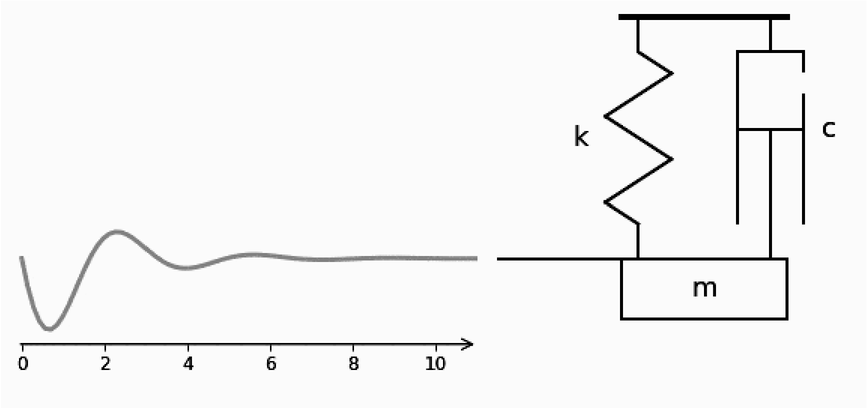
1. P: proportional control, control effort is linearly proportional to the system error;
2. I : integral control, control effort is linearly proportional to the integral of error over a period of time;
3. D: derivative control, control effort is linearly proportional to the rate of change of error, which gives a sharp response to a sudden change of signals.

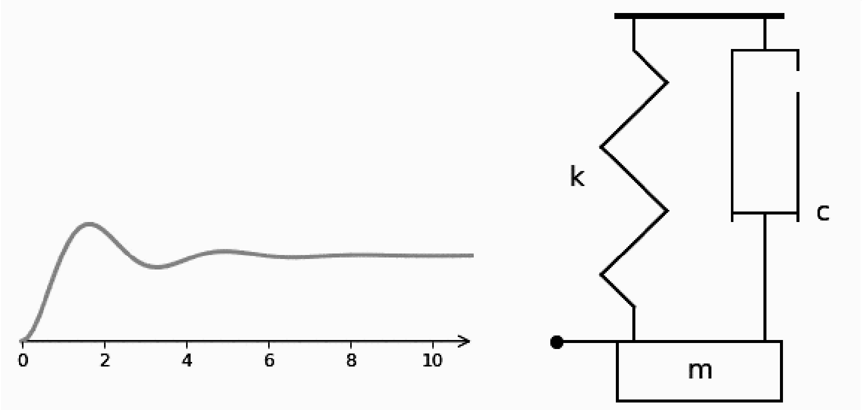
Focusing on the PD control components

$$\boldsymbol{u}(t) = k_p e + k_d \dot{e}$$

$$\Downarrow$$

$$\boldsymbol{u}(k) = k_p(0 - x(k)) + k_d(0 - \dot{x}(k))$$



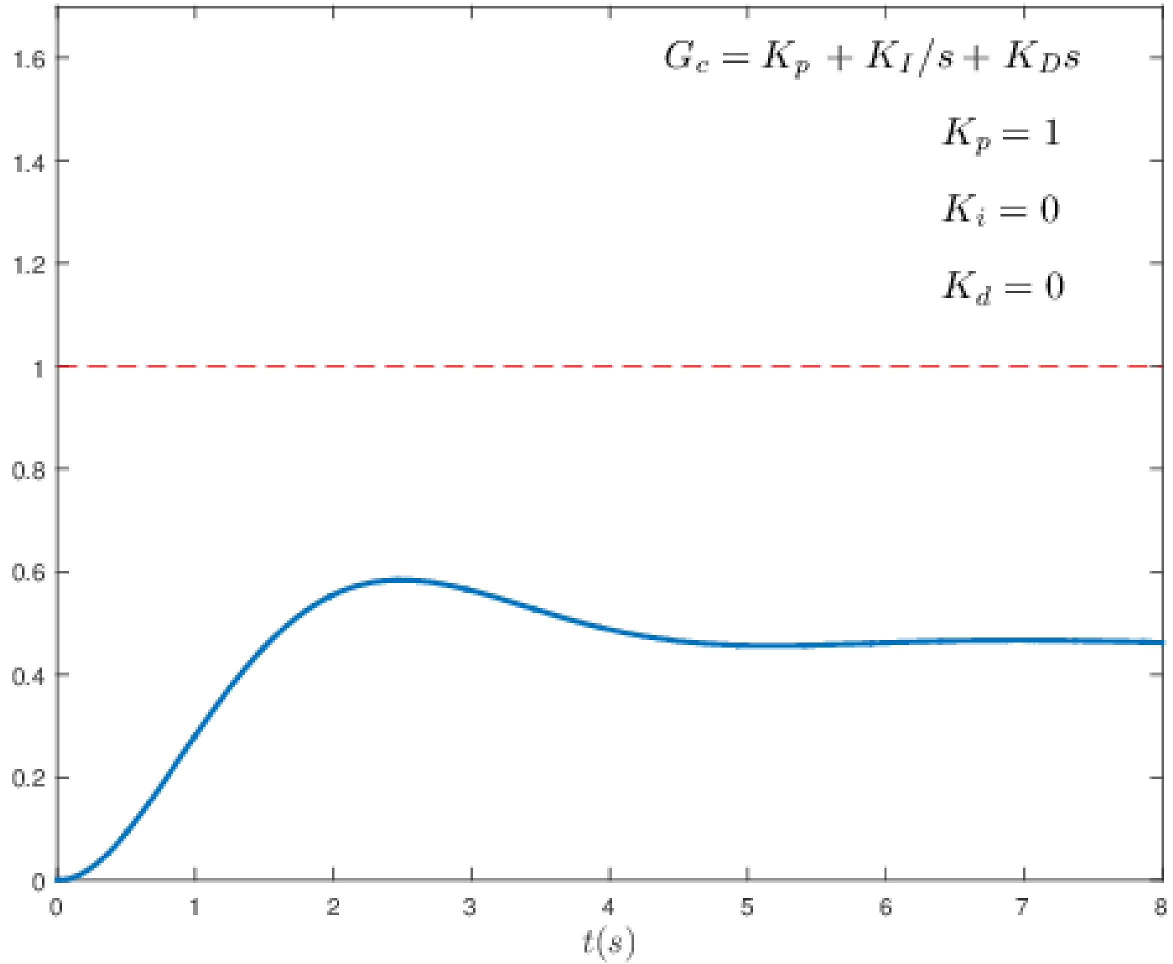


Effects of PID gains

Parameter	Rise time	Overshoot	Settling time	Steady-state error	Stability
k_p	Decrease	Increase	Small change	Decrease	Degrade
k_i	Decrease	Increase	Increase	Eliminate	Degrade
k_d	Slightly increase	Decrease	Decrease	No effect	Improve if velocity signal is good (not noisy, little delay)

- k_d term predicts system behaviour in *one tick*, which gives a control effort with the anticipation of the change during the next sampling time.
- In theory, given any k_p gain, there is always a k_d gain that can ensure critical damping of the response. However, due to the noise and delay of velocity, k_d cannot to be too large otherwise noise in amplified. Therefore, k_p gain can't be too large either.

Effects of gains



Examples: Response for Different Target Paths

Simulation of PD control, tracking a sawtooth signal.

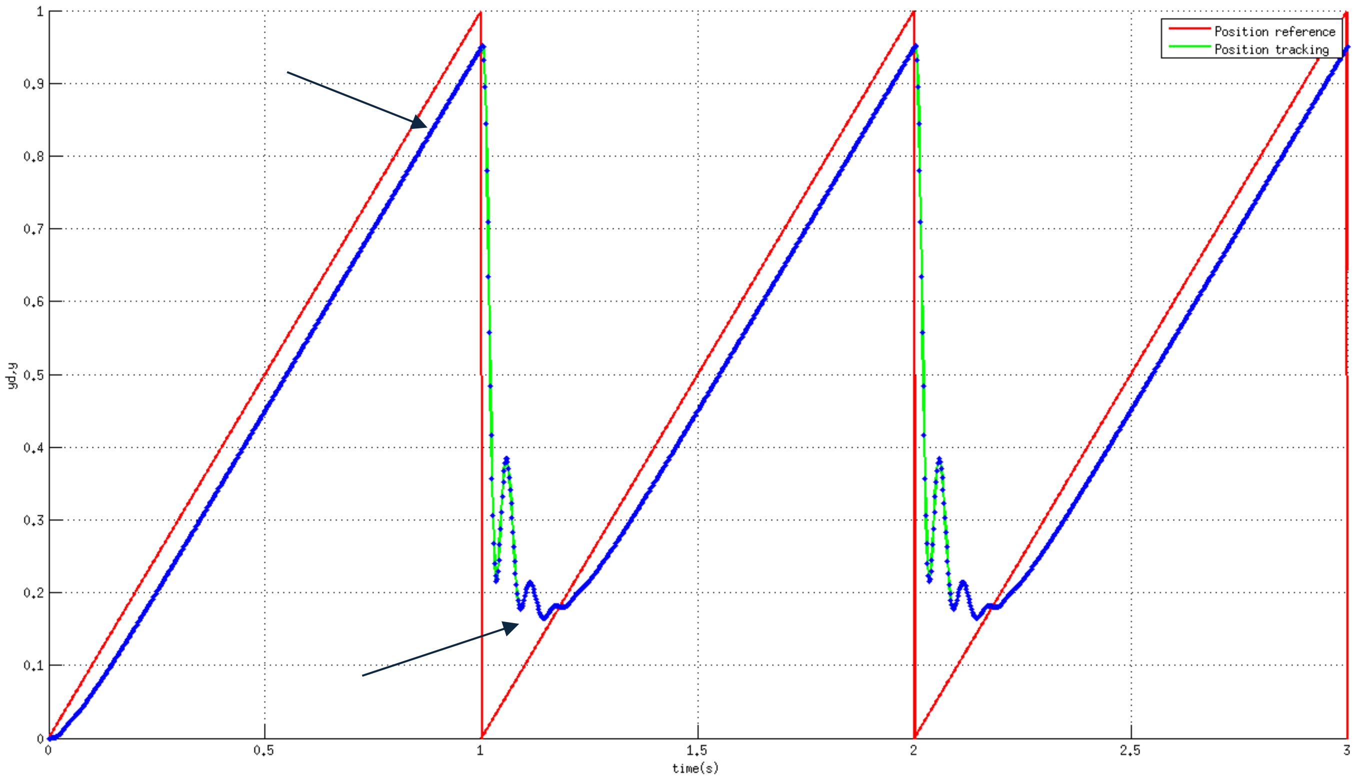
Low PD gain

No integral

kp=0.4;

ki=0.0;

kd=0.01;



Examples: Response for Different Target Paths

Simulation of PD control, tracking a sawtooth signal.

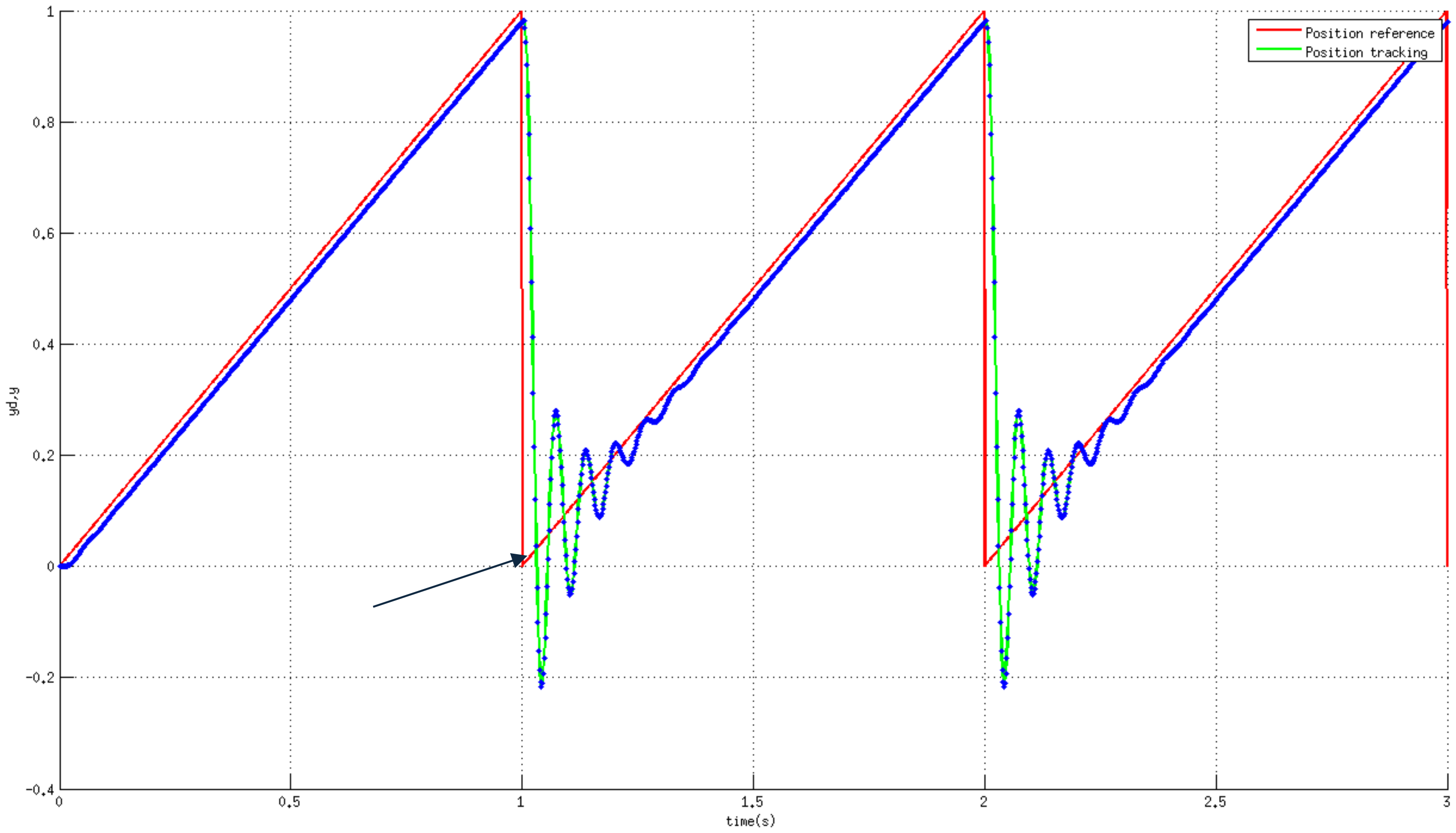
High PD gain

No integral

kp=1.0;

ki=0.0;

kd=0.01;



Examples: Response for Different Target Paths

Simulation of PD control, tracking a sawtooth signal.

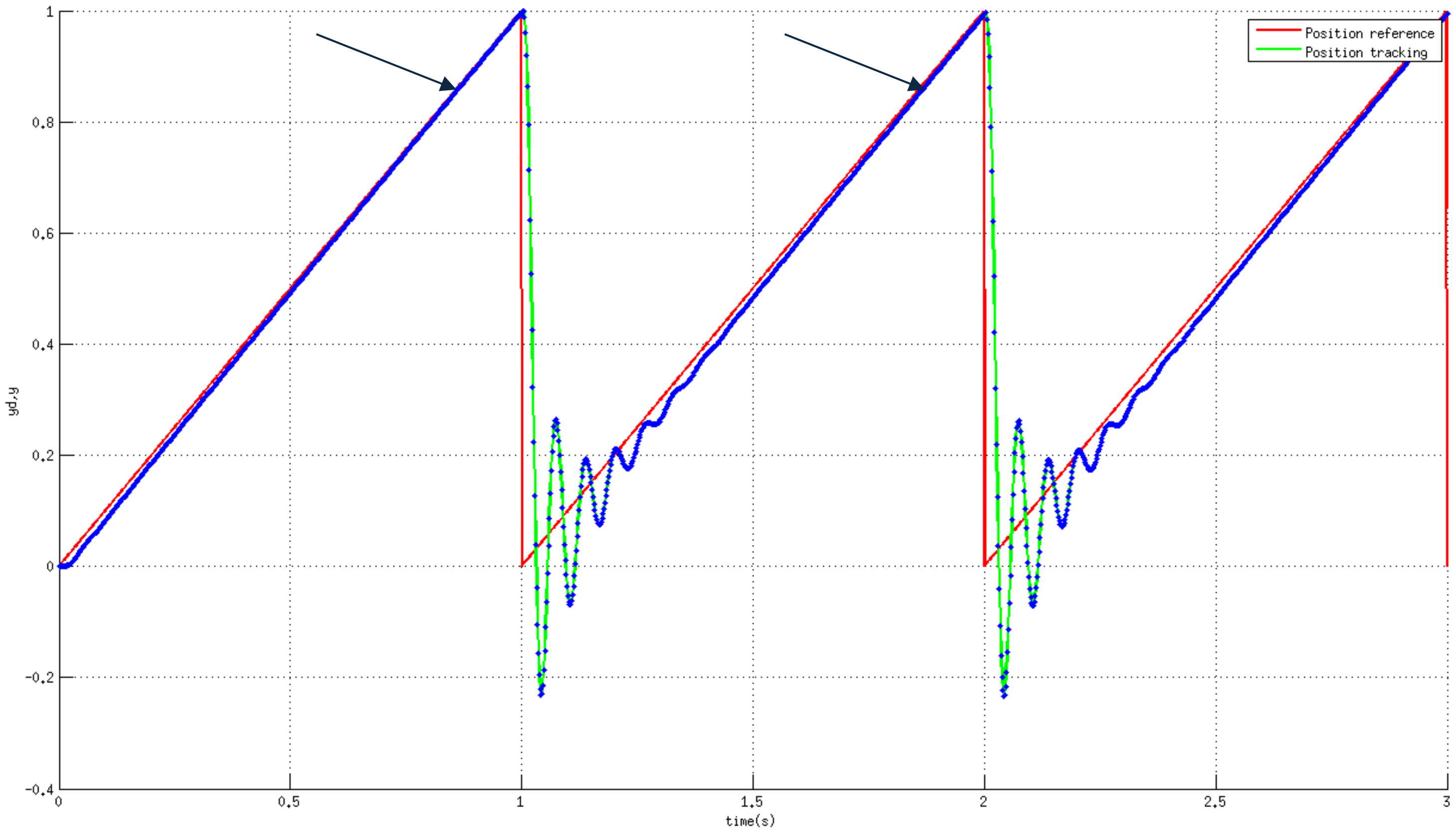
High PD gain

With integral

kp=1.0;

ki=2.0;

kd=0.005;



Examples: Response for Different Target Paths

Simulation of PD control, tracking a sawtooth signal.

High PD gain

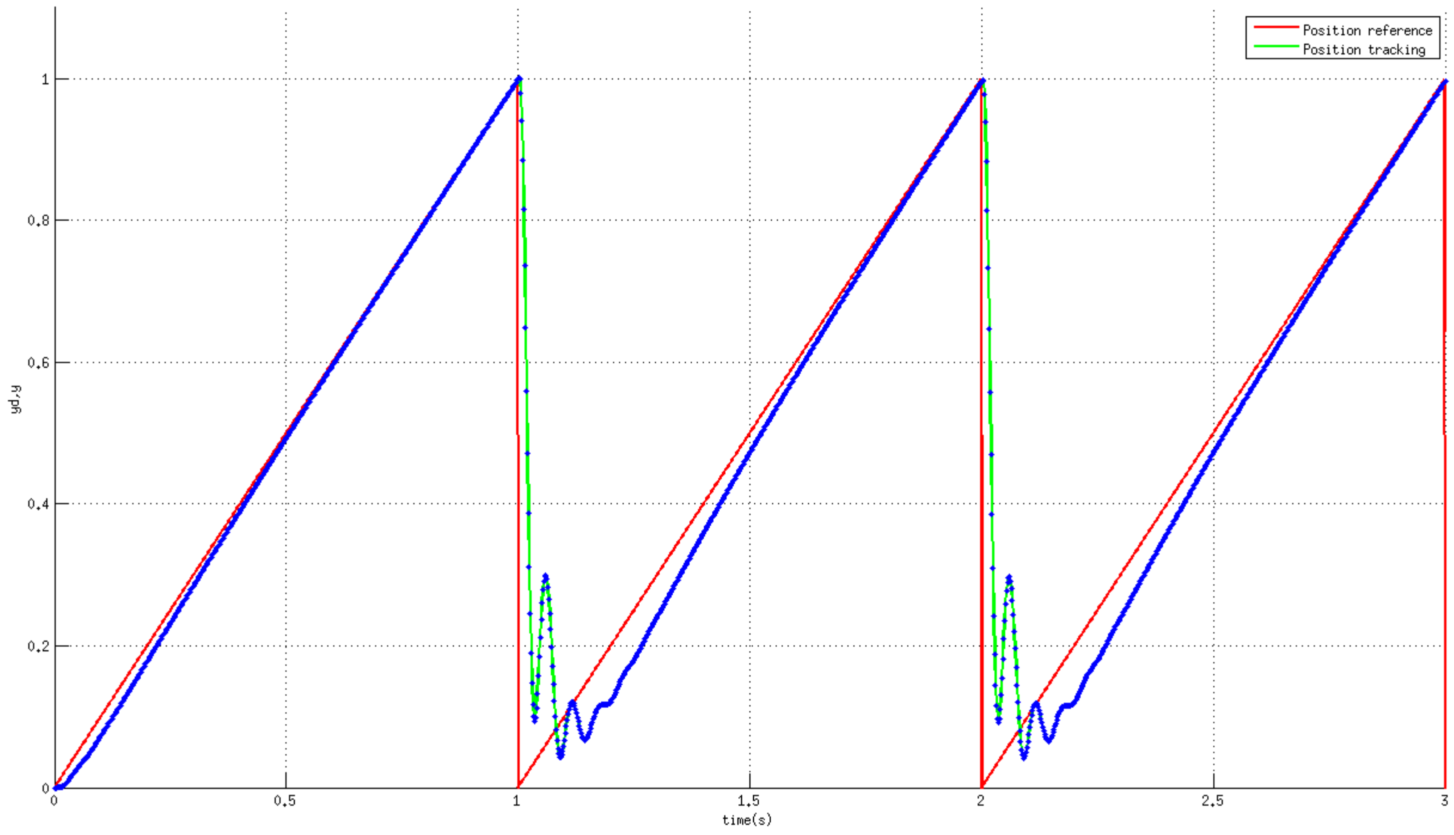
With integral

kp=0.6;

ki=2.0;

kd=0.01;

Empirically tuned gains.



Examples: Response for Different Target Paths

Simulation of PD control, tracking a sinusoidal signal.

High PD gain

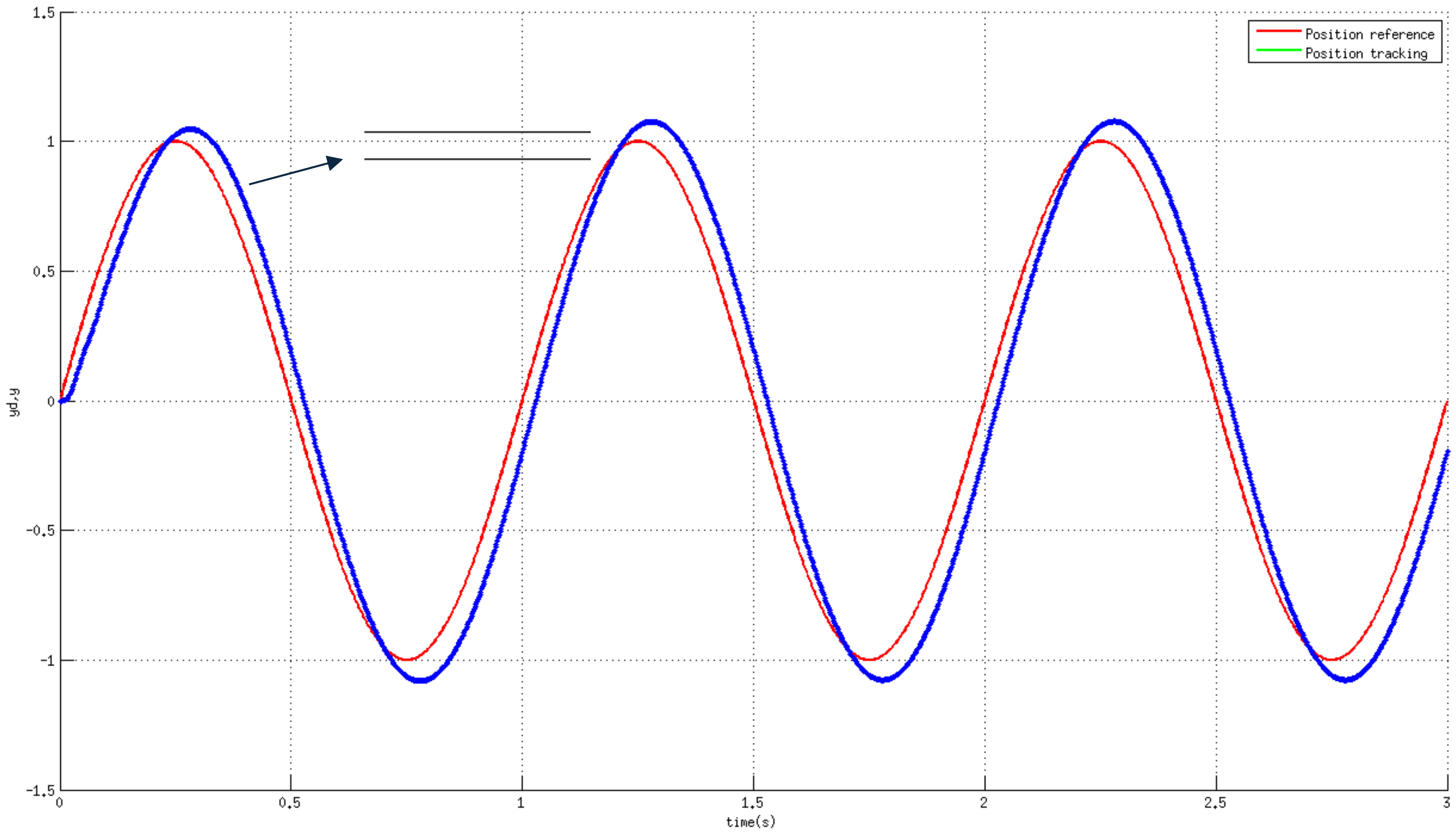
With integral

kp=0.6;

ki=2.0;

kd=0.01;

Overshooting problem caused by integral



Remark about *Digital* Implementation

Controls are often implemented in computer-based systems or by digital computation, e.g. micro-controllers, DSP, FPGA etc.
A digital control system only ‘sees’ the sensory information and command the control action at times, at a constant time interval.

$$\boldsymbol{u}(t) = k_p e + k_d \dot{e} + k_I \int e \, dt$$

The continuous PID control law

can be rewritten with appropriately adjusted coefficients as:

$$\boldsymbol{u}(k) = k_p e(k) + k_d \dot{e}(k) + k_i \sum e(i), i = 0, \ldots, k$$

Digital PID controller

Using backward Euler method:

$$\dot{e}(k) = \frac{[e(k) - e(k-1)]}{T}$$

(usually, derivative terms are filtered)

$$\int e\ dt = T \sum e(i)$$

note, in k^{th} control loop, range of i is: $i = 0, \ldots, k$

PID in continuous time $\boldsymbol{u}(t) = k_p e + k_d \dot{e} + k_I \int e\ dt$

PID in discrete time $\boldsymbol{u}(k) = k_p e(k) + k_d \frac{[e(k) - e(k-1)]}{T} + k_I T \sum e(i)$

$$\boldsymbol{u}(k) = k_p e(k) + k_d \dot{e}(k) + k_i \sum e(i), i = 0, \ldots, k$$

Concept: Feedback vs. Feedforward

For a single joint with the joint velocity as the control:

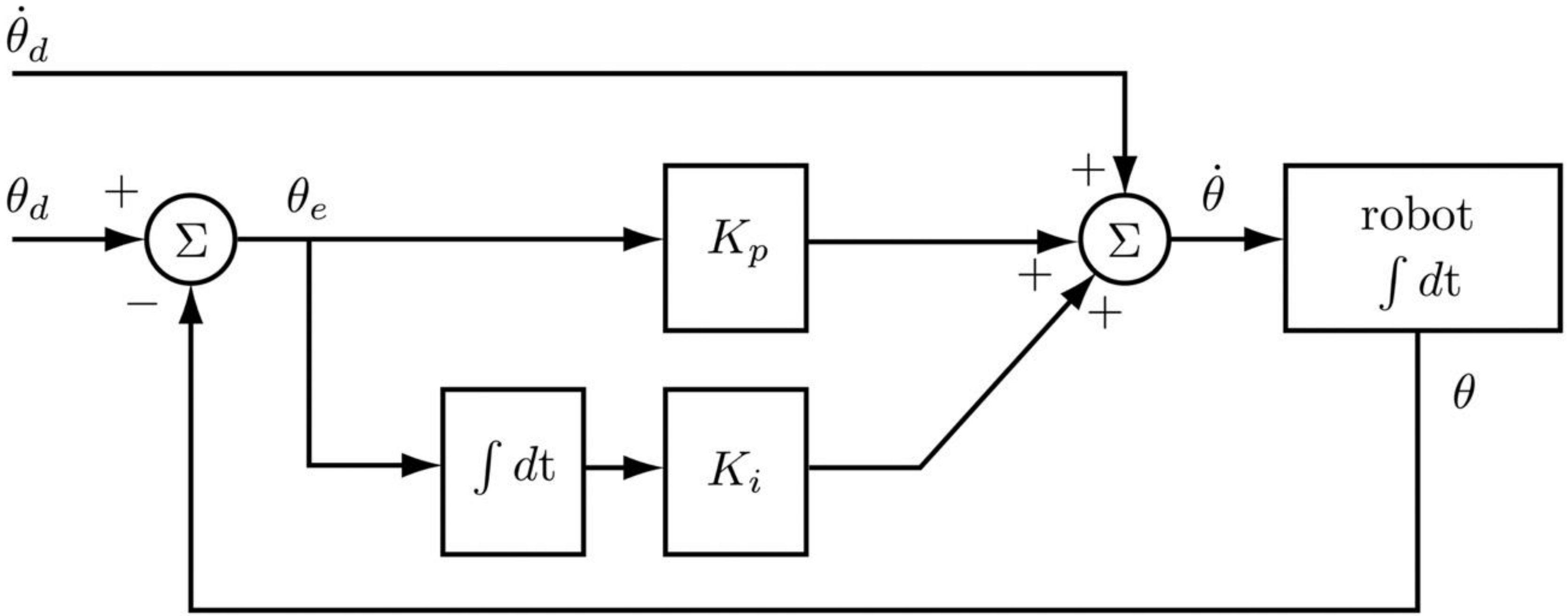
- **Open-loop (feedforward) control**: $\dot{\theta}(t) = \dot{\theta}_d(t)$
- **Closed-loop (feedback) control**: $\dot{\theta}(t) = f(\theta_d(t), \theta(t))$
- **FF + Proportional-Integral (PI) FB control**:

$$\dot{\theta}(t) = \dot{\theta}_d(t) + K_p \theta_e(t) + K_i \int_0^t \theta_e(\mathrm{t})\, d\mathrm{t}\ ,\ \ K_p, K_i \geq 0$$

  - *reduces to FF control if* $K_p, K_i = 0$
  - *if no FF term:* **P control** *when* $K_i = 0$, **I control** *when* $K_p = 0$

Discuss:
What is the point of FF control in this control law?

Block Diagram: Feedback and Feedforward



$$\dot{\theta}(t) = \dot{\theta}_d(t) + K_p\theta_e(t) + K_i \int_0^t \theta_e(\mathrm{t})\, dt$$