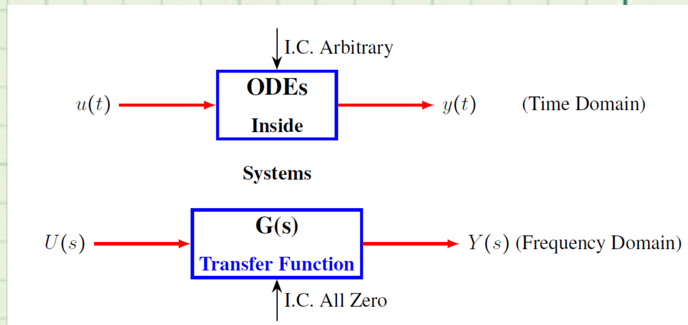
Summary: Transfer Functions map inputs to outputs.

$$\dddot{y} + a_2\ddot{y} + a_1\dot{y} + a_0 y = b_0 u + b_1\dot{u} + b_2\ddot{u}$$



$$G(s) = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0} = \frac{Y(s)}{U(s)}$$

ALL I.C.s Zero

$$Y(s) = G(s)\cdot U(s)$$

Transfer Function

$$\dot{x} = Ax + bu \qquad x(0^-) = 0 \Rightarrow G(s) = c(sI - A)^{-1} b$$
$$y = cx$$

BIBO STABILITY of $G(s) = \frac{N(s)}{D(s)}$, no common factors, $\deg(N(s)) \le \deg(D(s))$ iff all poles (roots of $D(s)$) negative real parts. Zeros (roots of $N(s)$) do not PLAY a ROLE in stability (stay tuned)

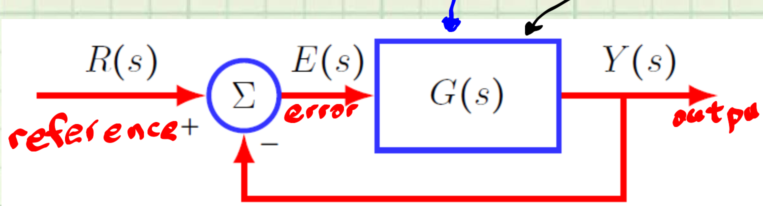
$s^2 + a_1 s + a_0 = 0$ has all roots w/ neg. real parts if, and only if $a_1 > 0$, $a_0 > 0$

$s^3 + a_2 s^2 + a_1 s + a_0 = 0$ has all roots w/ neg. real parts if, and only if $a_2 > 0$, $a_0 > 0$, & $a_1 - \frac{a_0}{a_2} > 0$

Unity Feedback System

open-loop transfer function

also called forward path t.f.



Graphical Representation of Linear Equations

$$\left.\begin{aligned} E(s) &= R(s) - Y(s) \\ Y(s) &= G(s)\cdot E(s) \end{aligned}\right\} \Rightarrow \quad \frac{Y(s)}{R(s)} = \frac{G(s)}{1+G(s)} \Leftrightarrow Y(s) = \frac{G(s)}{1+G(s)} R(s)$$

Example: $G(s) = \dfrac{s+2}{s^2-1}$ unstable $\left[s^2 + \underset{a_1}{\underset{\uparrow}{0}}\cdot s + \underset{a_0}{\underset{\uparrow}{(-1)}}\right]$

$$\frac{G(s)}{1+G(s)} = \frac{\frac{s+2}{s^2-1}}{1+\frac{s+2}{s^2-1}} \frac{(s^2-1)}{(s^2-1)} = \underbrace{\frac{s+2}{s^2+s+1}}_{\text{BIBO stable}}$$

Can go the other way

$G(s) = \dfrac{-3}{s^2+s+1}$ BIBO

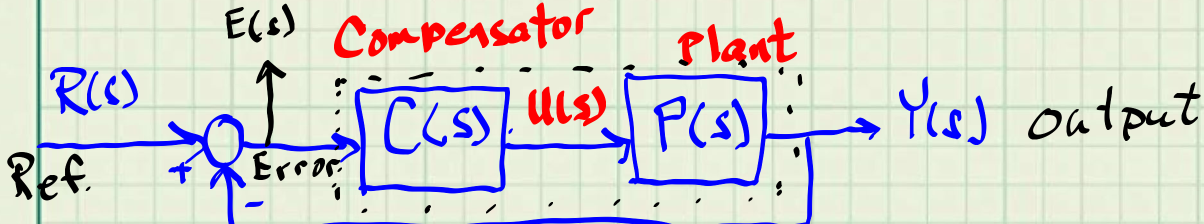
$$\frac{G(s)}{1+G(s)} = \frac{-3}{1+\frac{-3}{s^2+s+1}} = \frac{-3}{s^2+s-2}$$ Unstable

---

Today:
- Proportional Derivative Control
- How to "tune" it

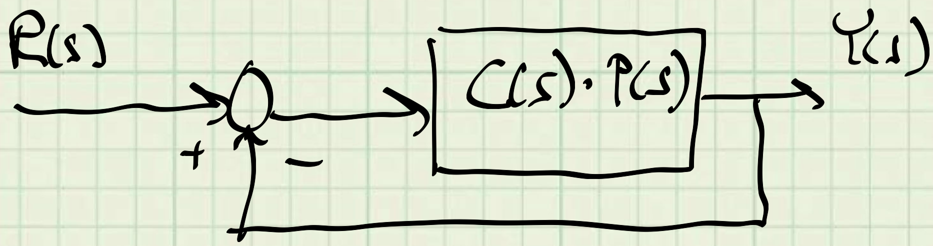
> - Can do most of Project 3 after today
> - No feedback control on the Final Exam

Cascade Control Architecture



Plant is the generic term for the

system that is to be controlled.



$$G(s) = C(s)\cdot P(s) \implies \frac{Y(s)}{R(s)} = \frac{C(s)P(s)}{1+C(s)P(s)}$$

Proportional Derivative Controller [PD]

Derivative Gain

$$C(s) = K_p + K_D s \longleftrightarrow K_p + K_D \frac{d}{dt}$$

Proportional Gain

$$U(s) = C(s)E(s) = (K_p + K_D s)E(s)$$

$$u(t) = \left(K_p + K_D \frac{d}{dt}\right)e(t) = K_p e(t) + K_D \dot{e}(t)$$

Example: $P(s) = \dfrac{1}{s^2-2s+1}$ unstable

We select a PD controller

$C(s) = K_p + K_D s$

$$\frac{Y(s)}{R(s)} = \frac{C(s)\,P(s)}{1 + C(s)\,P(s)} = \frac{(K_p + K_D s)\cdot \frac{1}{s^2 - 2s + 1}}{1 + (K_p + K_D s)\cdot \frac{1}{s^2-2s+1}}$$

$$= \frac{K_p + K_D s}{s^2 - 2s + 1 + (K_p + K_D s)}$$

$$= \frac{K_p + K_D s}{s^2 + (K_D - 2)s + (1 + K_p)}$$

Find values for K_p and K_d so that the closed-loop is BIBO stable

$K_D - 2 > 0 \iff K_D > 2$

$1 + K_p > 0 \iff K_p > -1$

Question: Which values to choose for K_p and K_D ???

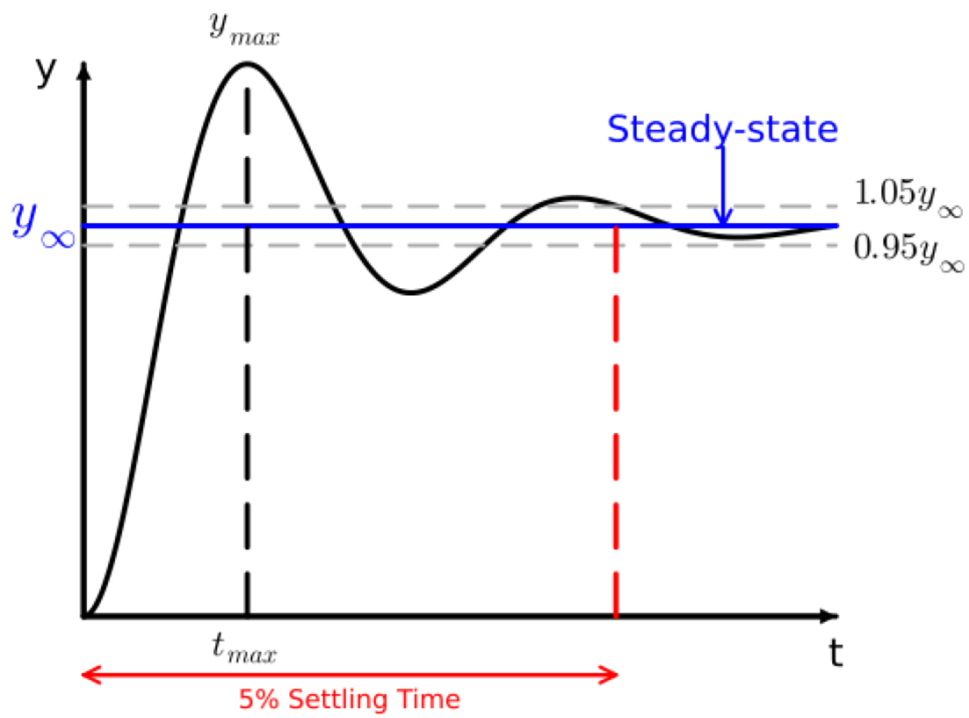


Figure 10.8: Typical step response of an underdamped system. The 5% settling time is the time it takes for the system to enter and then remain within the interval $[0.95y_\infty, 1.05y_\infty]$.

Key Qualitative Features

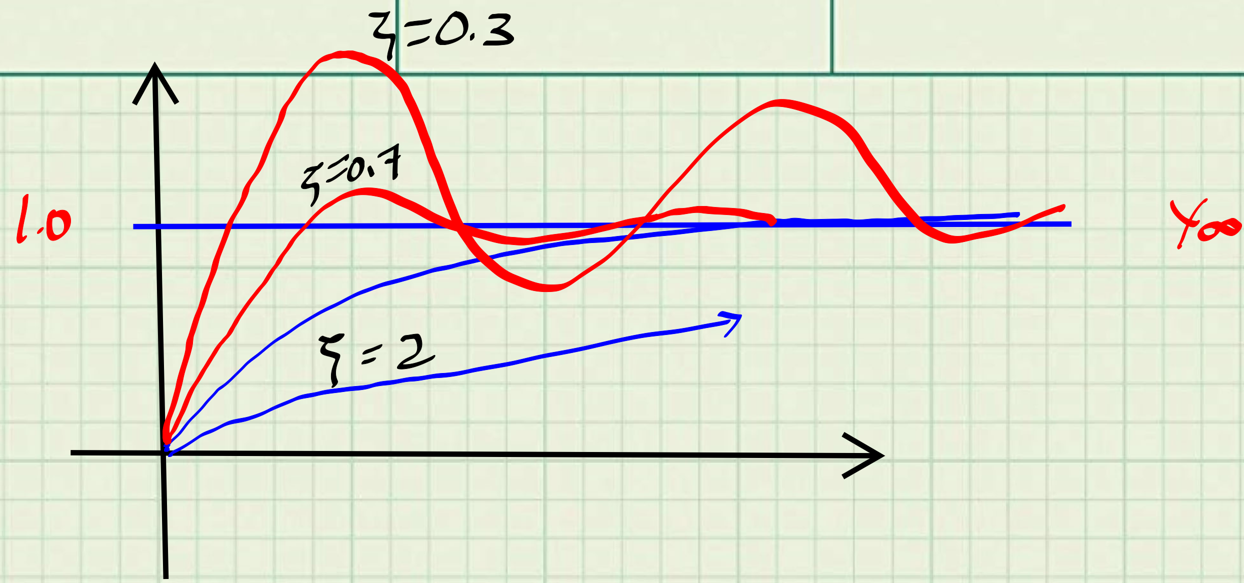
- % overshoot := $\frac{y_{\max} - y_\infty}{y_\infty} \times 100\%$
- 5% Settling time := $\min_{T>0} \{|y(t) - y_\infty| \leq 0.05y_\infty \ \text{ for all } \ t \geq T\}$

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

standard second-order

ζ = damping ratio

ω_n = undamped natural freq.



$$\zeta \quad \leftrightarrow \quad \text{percent overshoot} \quad = \quad 100 e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}}\ \%$$

$$\omega_n \quad \leftrightarrow \quad 5\%\ \text{settling time} \quad \approx \quad \frac{3}{\zeta\omega_n}.$$

5%

$$T_s = \frac{3}{\zeta\omega_n}$$

$$\omega_n = \frac{3}{\zeta \cdot T_s}$$

; $\zeta \to 1$ yields no overshoot. Common numbers to keep in
ntrol system designers use the exact formula for the dam
g will be required in the end no matter what because the m

ζ	$\approx$ Overshoot	True Overshoot
1.0	0 %	0.00 %
0.9	0 %	0.15 %
0.8	2 %	1.5 %
0.7	5 %	4.6 %
0.6	10 %	9.5 %
0.5	15 %	16.3 %

$$G_{CL} \approx \frac{(K_P + K_D s)\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{K_D\left(s + \frac{K_P}{K_D}\right)\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \longleftrightarrow \underline{y(t) = \left(1 - e^{at}\sin(\omega t + \theta)\right)}$$

$\zeta = 0.7,\ T_s = 3$

$$G_{CL}(s) = \frac{K_D(s + z)\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \qquad z = \frac{K_P}{K_D}$$

PD adds a zero in the numerator

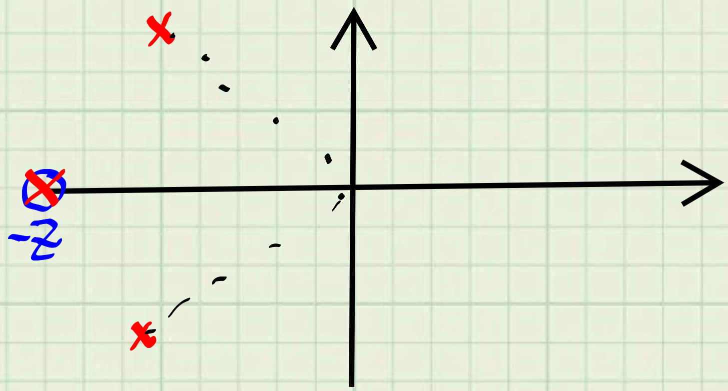
$$y_{CL} \longleftrightarrow K_D(s + z) \cdot y(t) = K_D\left(\frac{d}{dt} + z\right) y(t) =$$

$$= zK_D\, y(t) + \underbrace{K_D \frac{d}{dt} y(t)}$$

$$\frac{d}{dt}\left(1 - e^{at}\sin(\omega t + \theta)\right) = \boxed{\begin{aligned} &-a e^{at}\sin(\omega t + \theta) \\ &- e^{at} \cdot \omega \sin(\omega t + \theta) \end{aligned}}$$

Additional Oscillations

Left Half Plane



Can we remove the deleterious effects of the zero ~~added~~ by the PD controller?

Bottom line, we can cancel a stable zero with a stable pole

Missing Ingredients

- Segway is a NL model.

$$\dot{x} = \underline{f(x)} + b\,\underline{u} \qquad x = \begin{bmatrix} \theta \\ \varphi \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix}$$

$$x_e = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{equilibrium}$$

$$f(x_e) = 0$$

$A := 4\times 4$

$$f(x_e + \delta x) \approx \underbrace{\cancel{f(x_e)}^{\,0} + \overbrace{J_f(x_e)} \cdot \delta x}_{\text{Linearization of } f}$$

$$\boxed{\delta \dot{x} = A\,\delta x + b u}$$

Linear Approx of the ODE

Segway also comes from the Robot Equations

$$q = \begin{bmatrix} \theta \\ \varphi \end{bmatrix} \quad \begin{matrix} \text{lean angle} \\ \text{wheel angle} \end{matrix}$$

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \Gamma$$ ← external motor torque

The linearized robot equations are

$$\delta q = q - q_e$$
$$\delta \dot{q} = \dot{q} - 0$$
$$\delta \ddot{q} = \ddot{q} - 0$$

$$\boxed{D(q_e)\,\delta\ddot{q} + \frac{\partial G}{\partial q}(q_e)\cdot \delta q = \Gamma}$$

$$\delta x = \begin{bmatrix} \delta q \\ \delta \dot{q} \end{bmatrix} = \begin{bmatrix} \delta x_1 \\ \delta x_2 \end{bmatrix}$$

$$\delta \dot{x} = \begin{bmatrix} \delta \dot{q} \\ \delta \ddot{q} \end{bmatrix} = \begin{bmatrix} \delta x_2 \\ D(q_e) \backslash \left[-\frac{\partial G}{\partial q}(q_e)\,\delta x_1 + \Gamma \right] \end{bmatrix}$$

$$= \begin{bmatrix} 0_{2\times 2} & I_2 \\ A_{21} & 0_{2\times 2} \end{bmatrix} \begin{bmatrix} \delta x_1 \\ \delta x_2 \end{bmatrix} + bu$$

$$A_{21} = D(q_e) \backslash \left(-\frac{\partial G}{\partial q}(q_e) \right)$$

$$b = \begin{bmatrix} 0_{2\times 1} \\ D(q_c) \backslash \Gamma \end{bmatrix}$$

Linear Models $\rightarrow$ Transfer Functions

$\rightarrow$ PD Controller $\rightarrow$ Pre-compensator

Beautiful balancing controller for the Segway/BallBot