# MAE/ECE 5320 Mechatronics

2023 Spring

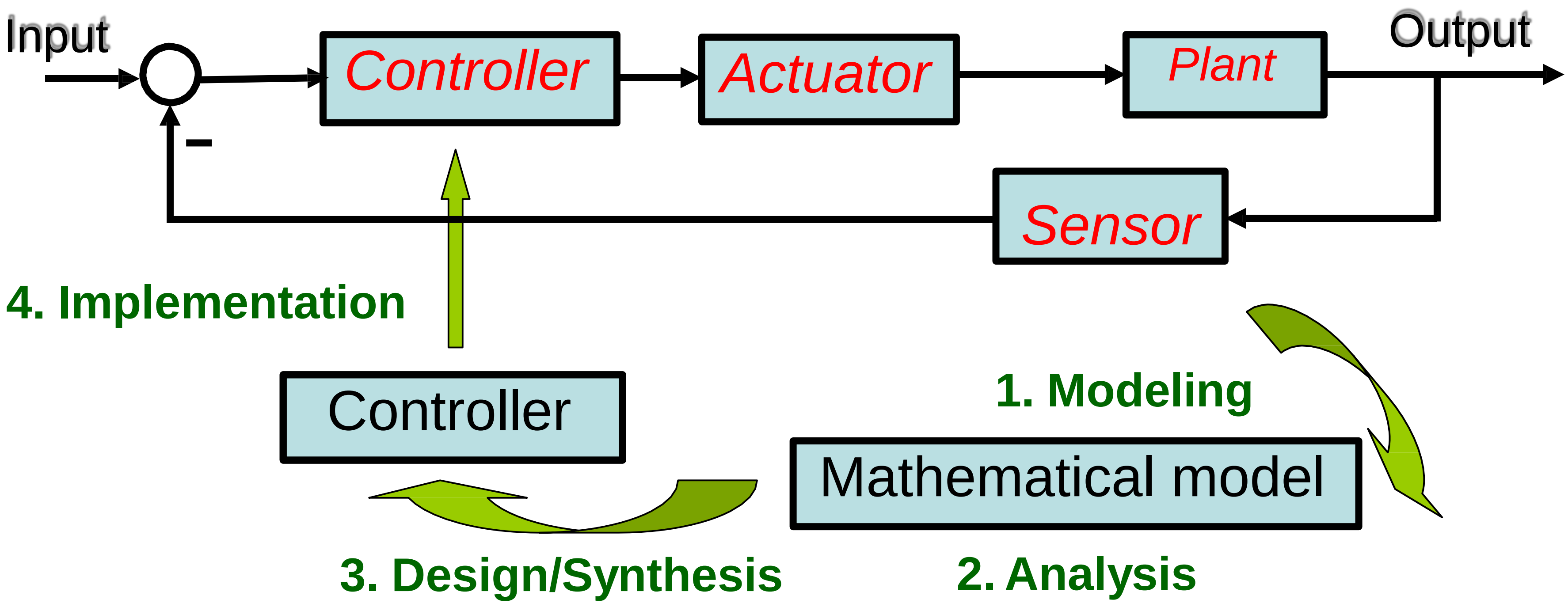
Lecture 02

Tianyi He
Utah State University

# Content

- Background
- System Stability
- First Order System Responses
- Second Order System Responses
- PID control tuning

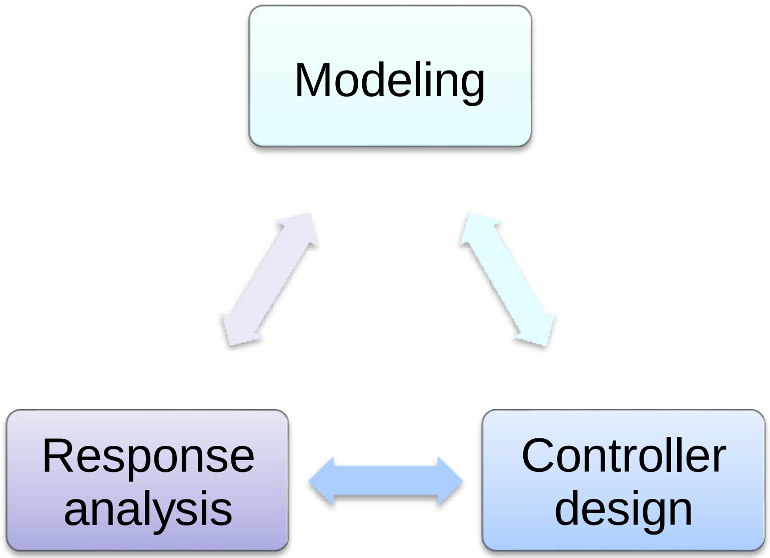
# Closed-loop Control System



# Overview

**Models:**
- transfer function
- state-space

- mechanical
- electrical
- electro-mechanical




**Stability**
- Routh-Hurwitz
- Nyquist
- Lyapunov, BIBO

**Time response**
- Transient
- Steady-state

**Frequency response**
- Bode plot
- Nyquist

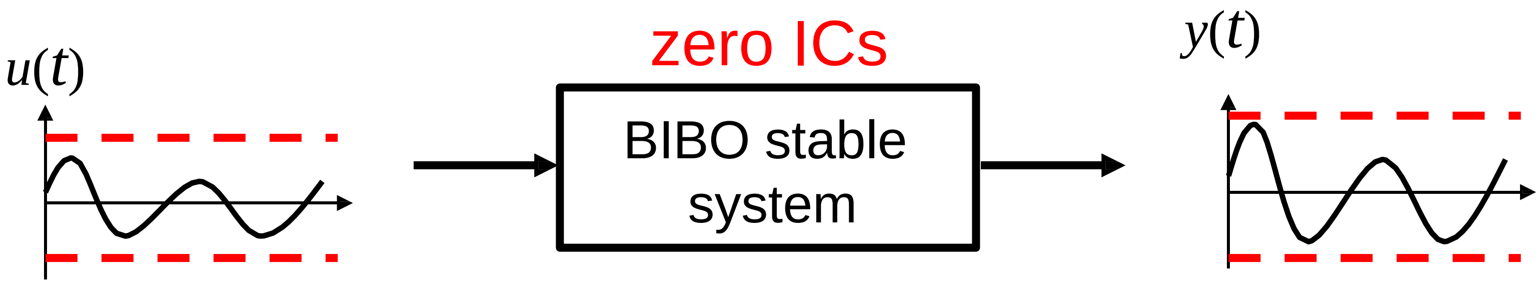
- **Achieve desired time/frequency performance**

- **Using Root Locus or**
- **Frequency domain**

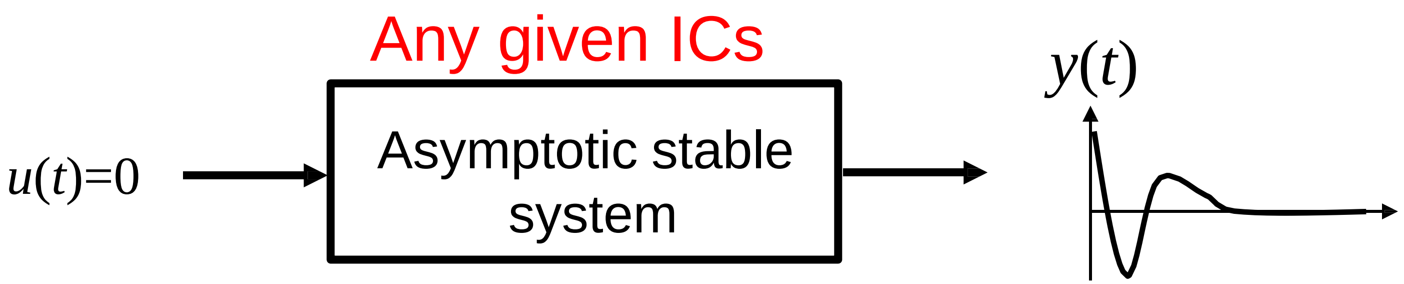
- **PID & Lead-lag controllers**
- **PID tune**

# Stability Definition

- BIBO (Bounded-Input-Bounded-Output) stability : *Any bounded input generates a bounded output.*




- Asymptotic stability : *Any given ICs generates y(t) converging to zero.*

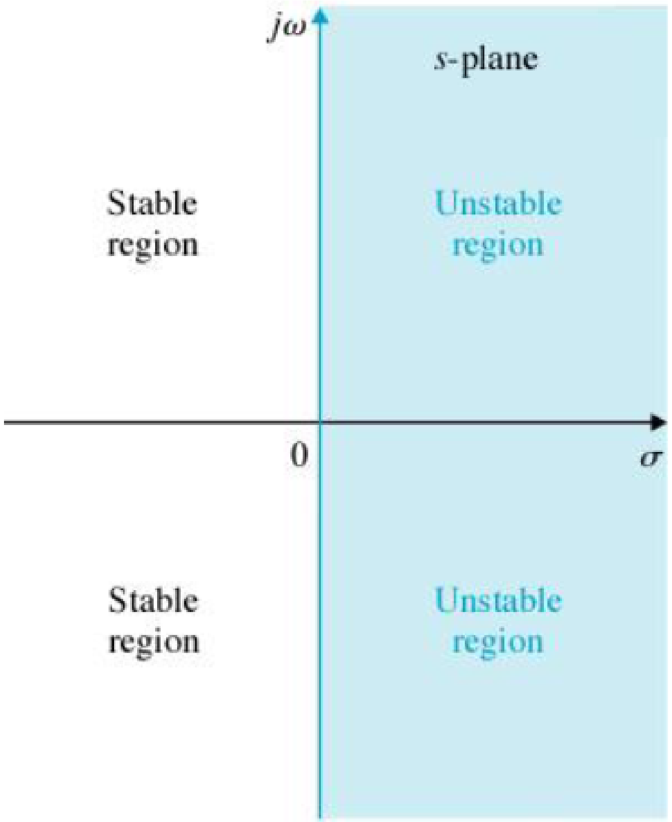


# Stability – "s" Domain Stability

For a system by a transfer function $\boldsymbol{G}(\boldsymbol{s})$, *Let* $s_i$ *be poles of* $G$. *Then,* $G$ *is*

BIBO stable $\Leftrightarrow$ $Re(s_i) < 0$ for all $i$ $\Leftrightarrow$ asymptotically stable

- (BIBO and asymptotically) stable if $Re(s_i) < 0$ for all $i$.
- marginally stable if
  - $Re(s_i) \leq 0$ for all $i$, and
  - simple root for $Re(s_i) = 0$
- unstable if
  it is neither stable nor marginally stable.



# Stability – Routh-Hurwitz Criterion

- Consider a polynomial $Q(s) = a_n s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0$
- Assume $a_0 \neq 0$
  - If this assumption does not hold, Q can be factored as

$$Q(s) = s^m \underbrace{(\hat{a}_{n-m} s^{n-m} + \hat{a}_{n-m-1} s^{n-m-1} + \cdots + \hat{a}_1 s + \hat{a}_0)}_{\hat{Q}(s)}$$

$where\ \hat{a}_0 \neq 0$

  - The following method applies to the polynomial $\hat{Q}(s)$

$$\begin{array}{l|llll l}
s^n & a_n & a_{n-2} & a_{n-4} & a_{n-6} & \cdots \\
s^{n-1} & a_{n-1} & a_{n-3} & a_{n-5} & a_{n-7} & \cdots \\
s^{n-2} & b_1 & b_2 & b_3 & & \\
s^{n-3} & c_1 & c_2 & c_3 & & \\
\vdots & \vdots & \vdots & & & \\
s^2 & k_1 & k_2 & & & \\
s^1 & l_1 & & & & \\
s^0 & m_1 & & & &
\end{array}$$

1 From the given polynomial

2

$$b_1 = \frac{a_{n-2}a_{n-1} - a_n a_{n-3}}{a_{n-1}}$$

$$b_2 = \frac{a_{n-4}a_{n-1} - a_n a_{n-5}}{a_{n-1}}$$

$$\vdots$$

3

$$c_1 = \frac{a_{n-3}b_1 - a_{n-1}b_2}{b_1}$$

$$c_2 = \frac{a_{n-5}b_1 - a_{n-1}b_3}{b_1}$$

$$\vdots$$

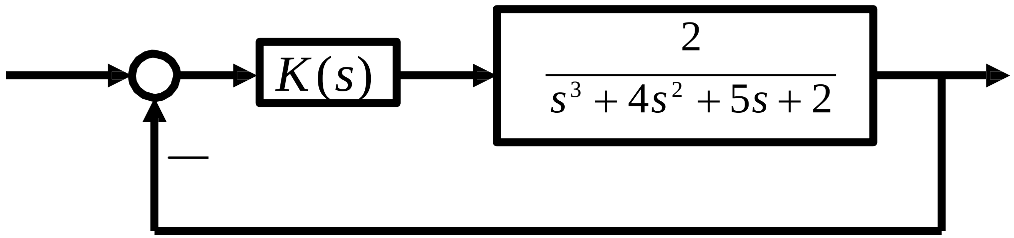
4

*The number of unstable roots*

*=*

*the number of sign changes in the first column*

# Example – Routh-Hurwitz Criterion




*Design $K(s)$ that stabilizes the closed-loop system*

$K(s) = K$ *(constant)*

Characteristic equation

$$1+\frac{2K}{s^3+4s^2+5s+2}=0 \qquad s^3+4s^2+5s+2(K+1)=0$$

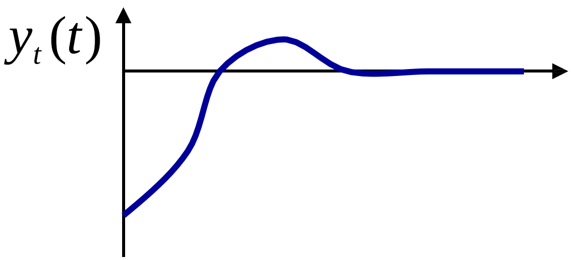
Routh array

$$\begin{array}{c|ll}
s^3 & 1 & 5 \\
s^2 & 4 & 2(K+1) \\
s^1 & \frac{9-K}{2} & >0 \\
s^0 & 2(K+1) & >0
\end{array}
\left.\vphantom{\begin{array}{c}1\\1\end{array}}\right\} \Rightarrow -1<K<9$$

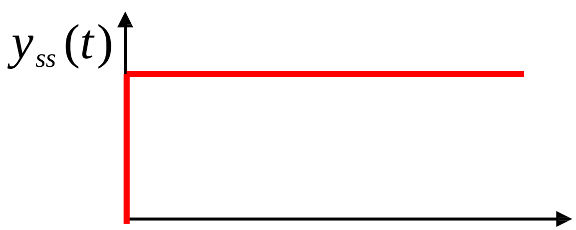
# Time Response – Input Output Relationship


- analyze a system property by applying an *input r(t)* and observing a time response y(t). Common-used inputs : step, impulse, ramp, sinusoidal.
- Time response can be divided as

$$y(t) = \underbrace{y_t(t)}_{\text{Transient response}} + \underbrace{y_{ss}(t)}_{\text{Steady-state response}}$$



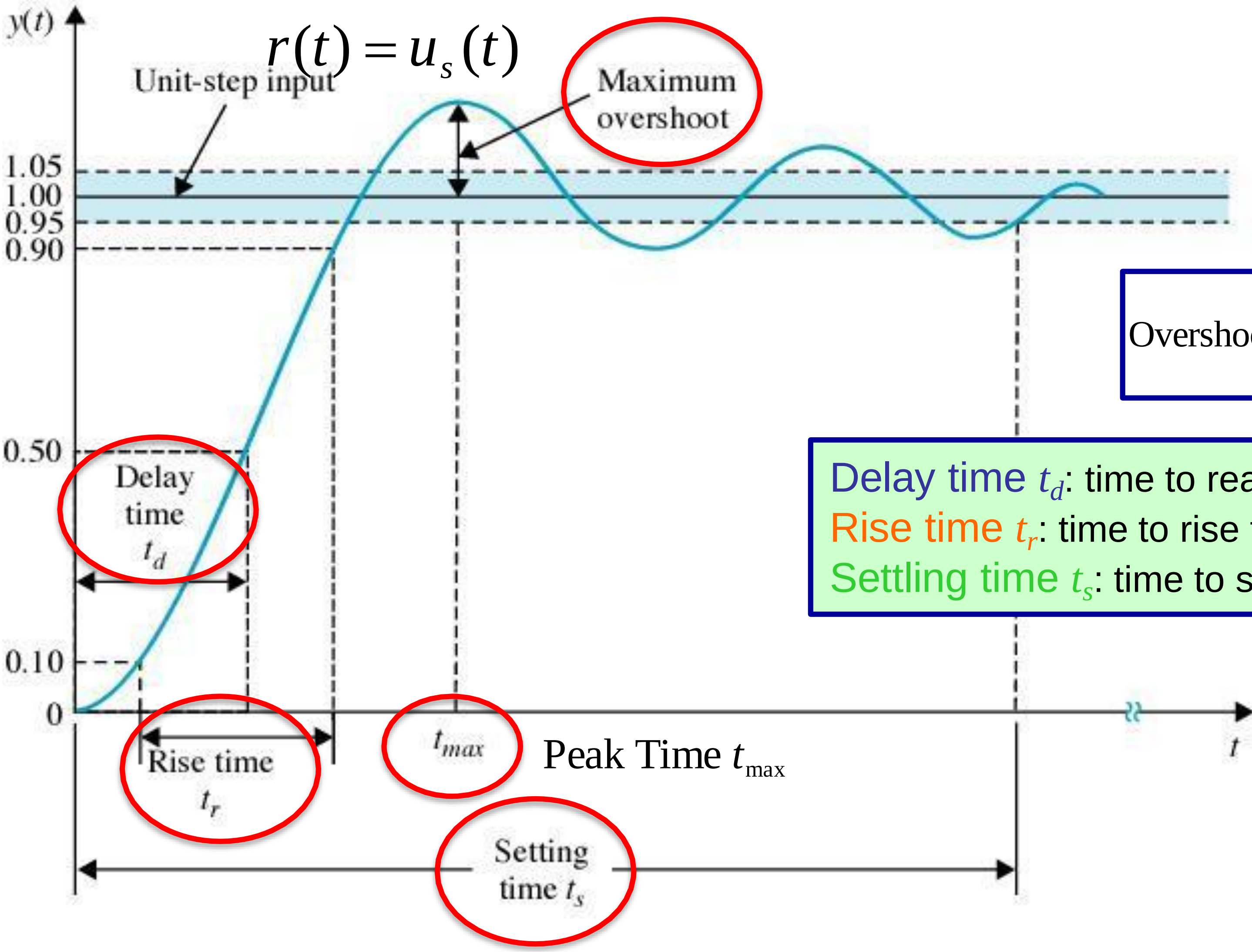

$y_t$ dies out, y(t) converges to $y_{ss}$




*Suppose* $\boldsymbol{G(s)}$ *is stable, by the final value theorem:*

$$\lim_{t\to\infty} y_t(t) = 0 \qquad y_{ss} = \lim_{s\to 0} sG(s)\frac{R}{s} = RG(0)$$

# TR-Response performance measure




$$r(t) = u_s(t)$$

Peak Time $t_{\max}$

$$SS_{err} = 1 - y_{ss}$$

$$\text{Overshoot (\%)} = \frac{y_{\max} - y_{ss}}{y_{ss}} \times 100$$

Delay time $t_d$: time to reach 0.5 $y_{ss}$
Rise time $t_r$: time to rise from 0.1 $y_{ss}$ to 0.9$y_{ss}$
Settling time $t_s$: time to settle within 5% of $y_{ss}$

# TR – Time Response Remarks

- Response speed is measured by *Rise time, delay time, and settling time*
- Relative stability is measured by *Percent overshoot*
- In general
  - Fast response → Large percent overshoot
  - Large percent overshoot → small stability margin
- We need to take **trade-off** between response speed and stability.

*Remarks*

- *Analytical system responses are often difficult to obtain, other than 1st, 2nd order system.*

- *Useful MATLAB command:*
  *step(sys), stepinfo(sys), impulse(sys), lsim(sys,u,t), initial(sys,x0)…*

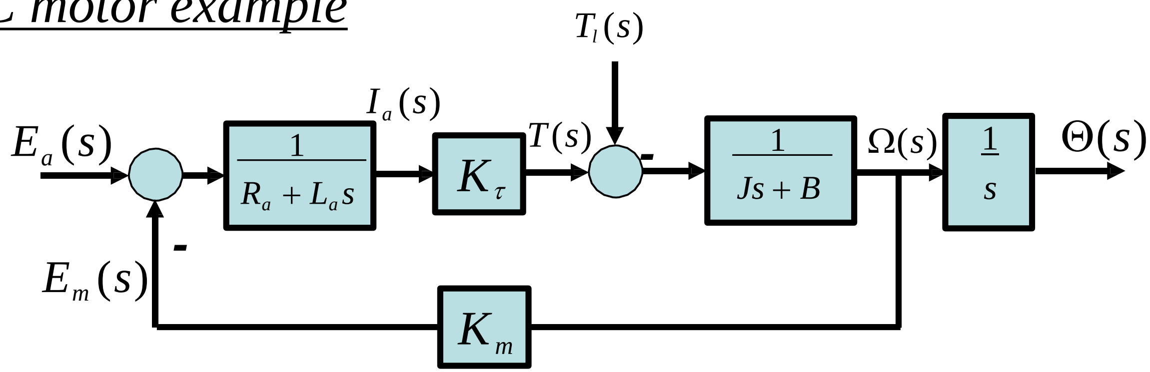
- Alternatively, use MATLAB SIMULINK to simulate time responses

# TR – 1st Order System Response

- A standard form of the first-order system:

$$G(s) = \frac{K}{Ts+1}$$

*DC motor example*



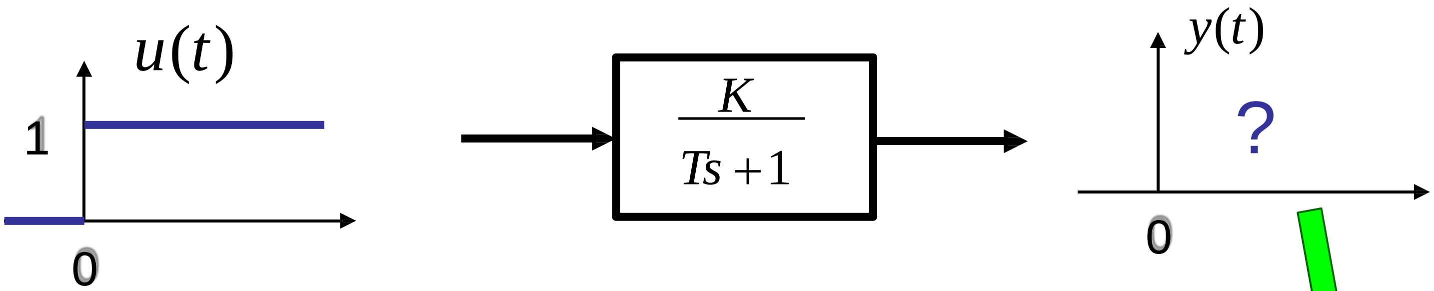

*Transfer Function from motor input voltage $E_a$ to motor speed $\Omega$*

Note: If $L_a << R_a$. We can approximate the DC motor using a first order system by setting $L_a = 0$.

$$\frac{\Omega(s)}{E_a(s)} = \frac{K_\tau}{(L_a s + R_a)(Js+B) + K_\tau K_m} \approx \frac{K_\tau}{R_a(Js+B) + K_\tau K_m}$$

$$=: \frac{K}{Ts+1} \left( K := \frac{K_\tau}{R_a B + K_m K_\tau},\ T = \frac{R_a J}{R_a B + K_m K_\tau} \right)$$

# TR – Step Resp. of 1st Order System

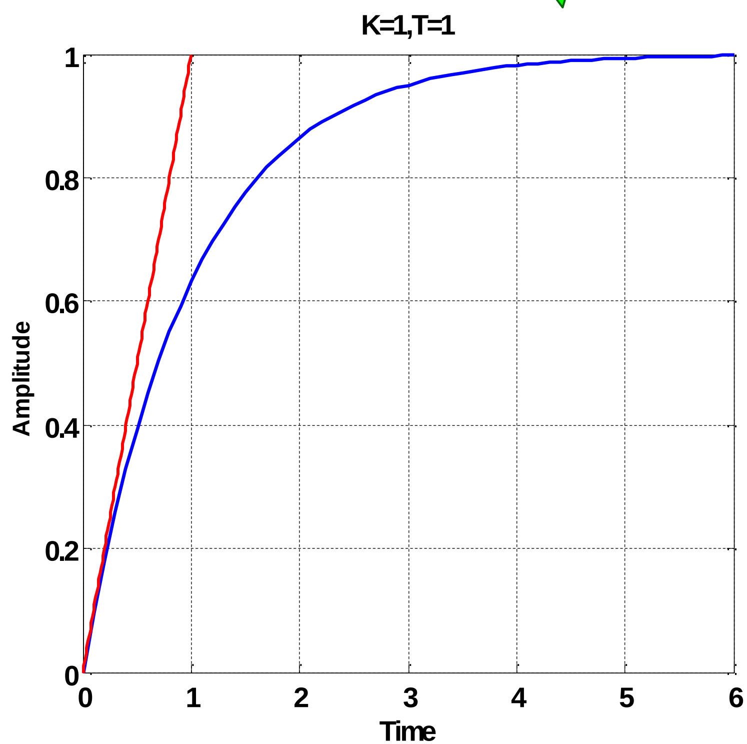



$$
\begin{aligned}
Y(s) &= G(s)U(s) \\
&= \frac{K/T}{s+1/T}\cdot\frac{1}{s} \\
&= \frac{K}{S} + \frac{-K}{s+1/T}
\end{aligned}
$$

(Partial fraction )

$L^{-1}$ *Inverse Laplace Transform*

$$
\begin{aligned}
y(t) &= L^{-1}[Y(s)] \\
&= K(1-e^{-t/T})\ (t \geq 0)
\end{aligned}
$$



# TR – physical meaning of K and T

- $K$: DC gain, amplifier ratio
- stable system $G(s)$, DC gain is $G(0)$
  - Final value theorem

$$\lim_{t\to\infty} y(t) = \lim_{s\to 0} sG(s)\frac{1}{s} = G(0)$$

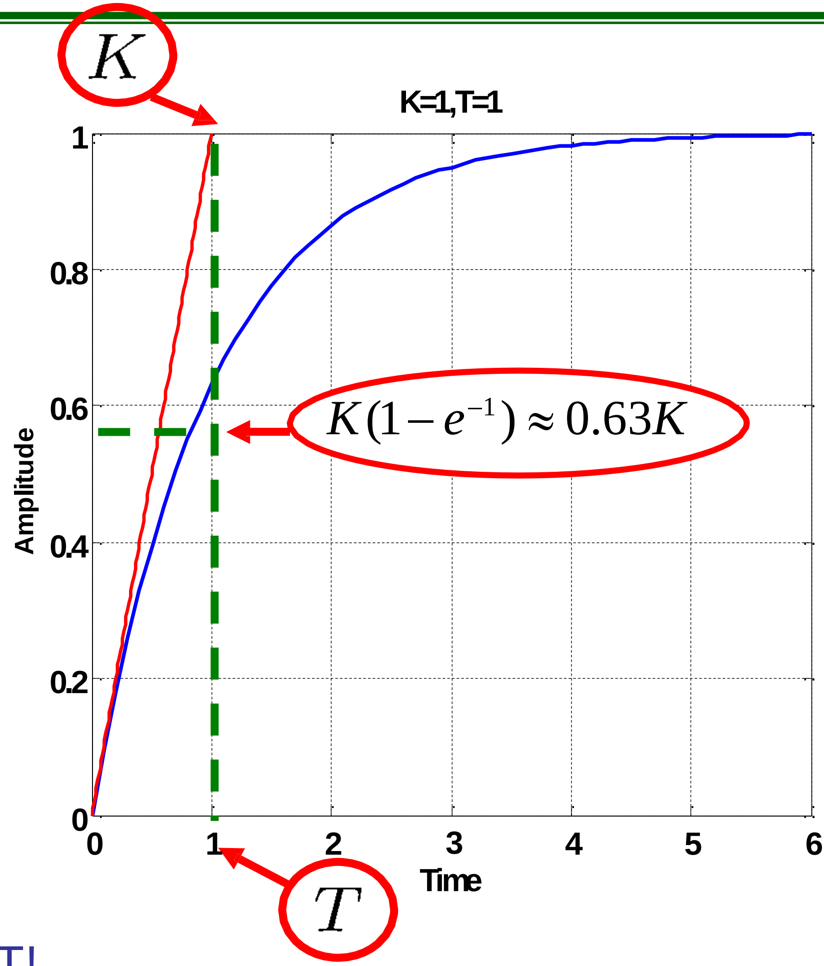
- $T$: Time constant
  - Time when response reaches 63% of final value;
  - convergence speed;

smaller $T$ , faster response speed

| $t$ | $e^{-t/T}$ |
|---|---|
| 0 | 1 |
| $T$ | 0.3679 |
| $2T$ | 0.1353 |
| $3T$ | 0.0498 |
| $4T$ | 0.0183 |
| $5T$ | 0.0067 |

5% settling time @ 3T!

2% settling time @ 4T!



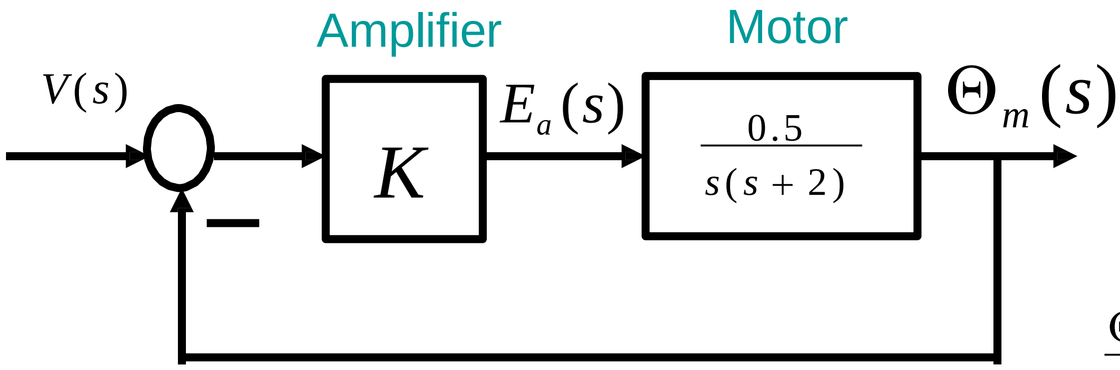

$$y(t) = K(1-e^{-t/T})$$

# TR – 2nd Order System response

- A standard form of the second-order system

$$G(s) = \frac{\omega_n^2}{s^2 + 2\varsigma\omega_n s + \omega_n^2} \begin{cases} \varsigma : & \text{damping ratio} \\ \omega_n : & \text{undamped natural frequency} \end{cases}$$

*DC motor position control example*

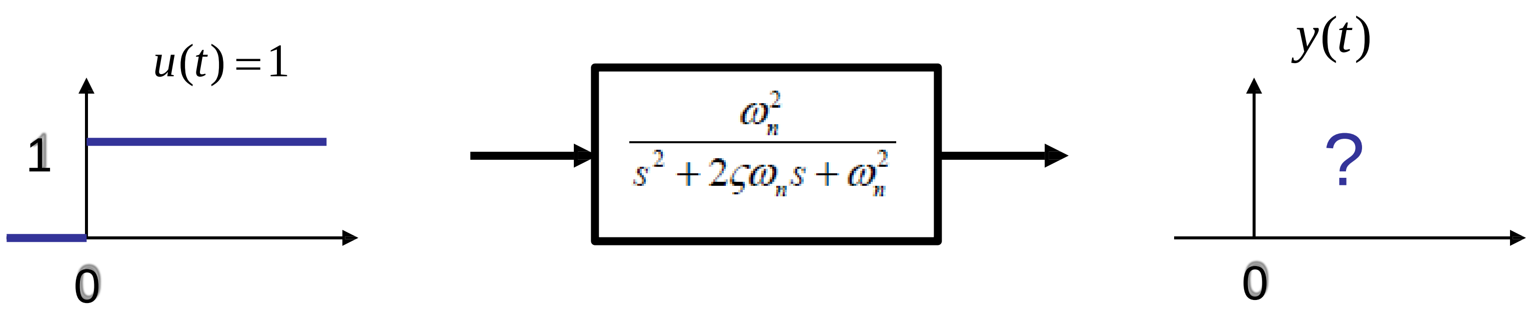



Closed-loop TF

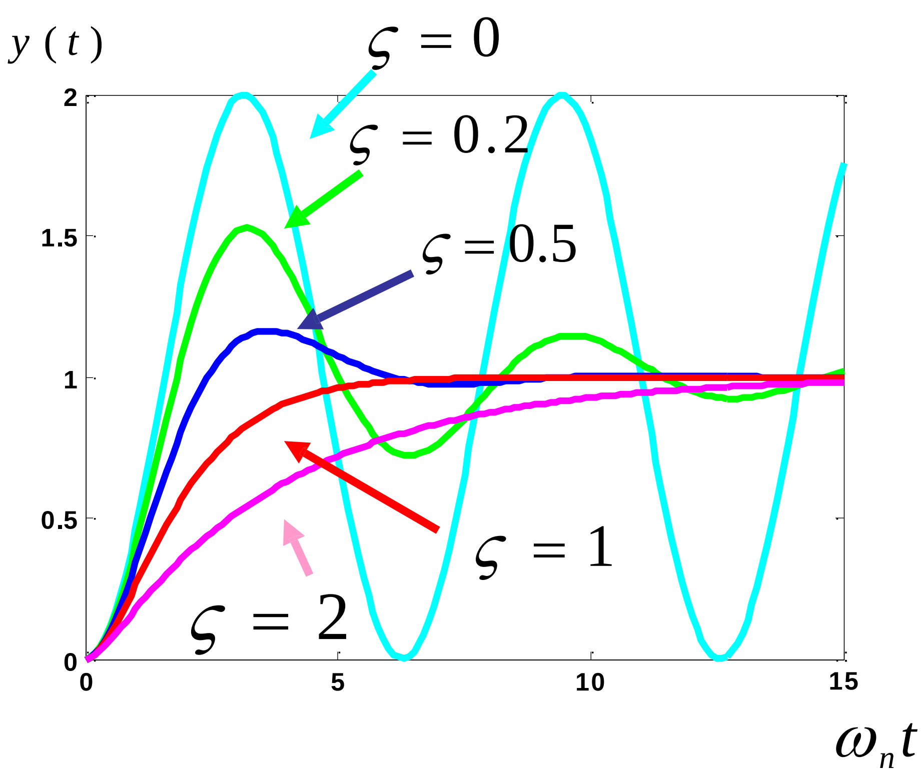
$$\frac{\Theta_m(s)}{V(s)} = \frac{0.5K}{s^2 + 2s + 0.5K}$$

*Transfer function from voltage to angular position of motor shaft*

# TR – Step Response of 2nd Order System




- Undamped $\varsigma = 0$
- Underdamped $0 < \varsigma < 1$
- Critically damped $\varsigma = 1$
- Overdamped $\varsigma > 1$



# Steady-state step responses

$$y_{ss} = \lim_{s\to 0} s\,Y(s) = \lim_{s\to 0} s\left(\frac{1}{s}\right)H(s) = \lim_{s\to 0} \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} = \frac{\omega_n^2}{\omega_n^2} = 1$$

More generally, if the numerator is not $\omega_n^2$, but some $K$ :

$$H(s) = \frac{K}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad \Rightarrow \quad \boxed{y_{ss} = \frac{K}{\omega_n^2}}$$

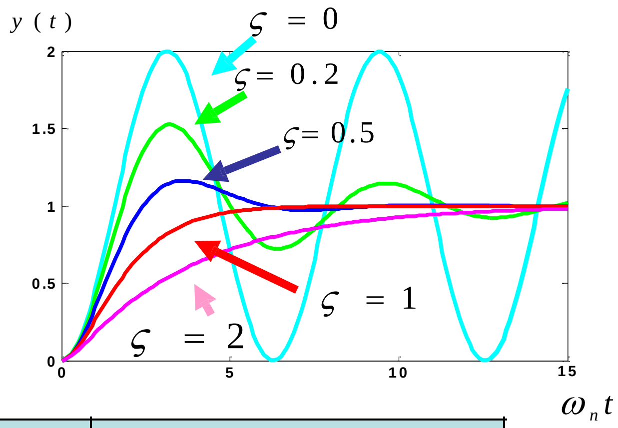
# TR – Step Response of 2nd-Order System

- Analytical solution of $y(t)$ for underdamped case

$$Y(s)=\frac{\omega_n^2}{s^2+2\varsigma\omega_n s+\omega_n^2}\cdot\frac{1}{s},\qquad 0<\varsigma<1$$

$$L^{-1}$$

$$\boxed{y(t)=1-\frac{e^{-\varsigma\omega_n t}}{\sqrt{1-\varsigma^2}}\sin(\omega_d t+\cos^{-1}\varsigma)}$$

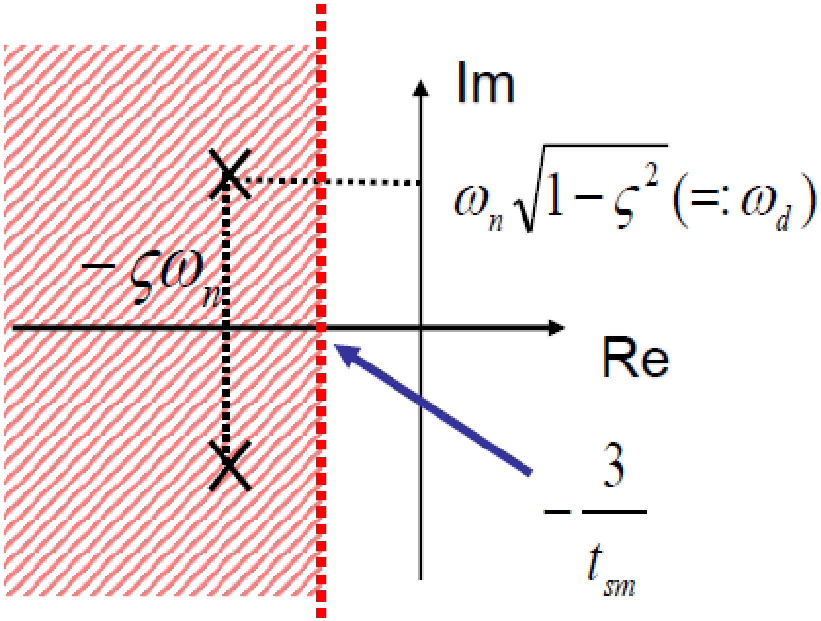



| | 1st Order | 2nd Order |
|---|---|---|
| Peak time | $\infty$ | $\frac{\pi}{\omega_d}=\frac{\pi}{\omega_n\sqrt{1-\varsigma^2}}$ |
| Peak value | 1 | $1+e^{-\frac{\varsigma\pi}{\sqrt{1-\varsigma^2}}}$ |
| Percent Overshoot | 0 | $100e^{-\frac{\varsigma\pi}{\sqrt{1-\varsigma^2}}}$ |
| Settling time (5%) | 3T | $\frac{3}{\varsigma\omega_d}$ |
| Settling time (2%) | 4T | $\frac{4}{\varsigma\omega_d}$ |

# TR – 2nd Order System ( Design example)

- Require 5% settling time $t_s < t_{sm}$ (given):

$$t_s \approx \frac{3}{\varsigma\omega_n} < t_{sm} \Rightarrow \boxed{\varsigma\omega_n > \frac{3}{t_{sm}}}$$

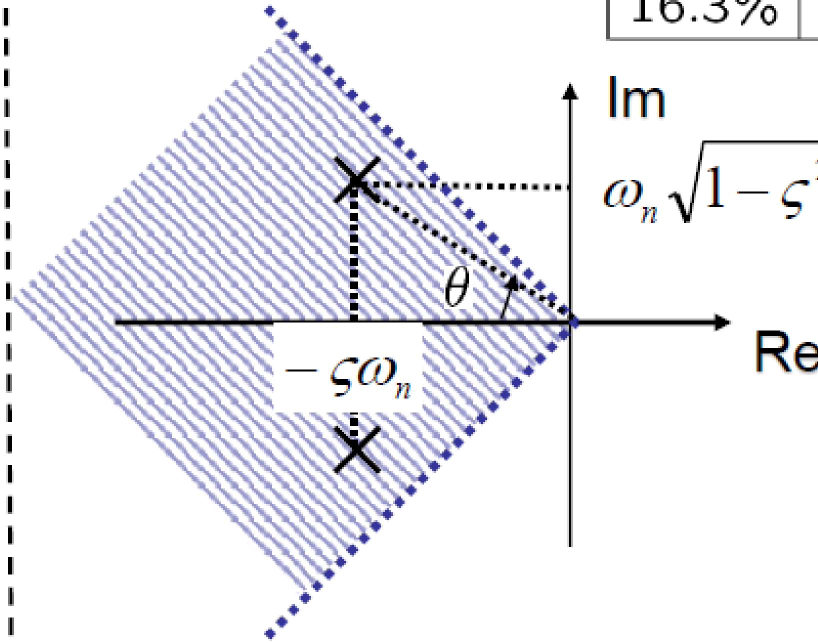



- Require $PO$<$PO_{max}$(given):

$$PO = 100e^{\frac{-\varsigma\pi}{\sqrt{1-\varsigma^2}}} < PO_{\max} \Rightarrow \boxed{\theta < \theta_m}$$

$$= 100e^{\frac{-\pi}{\tan\theta}}$$

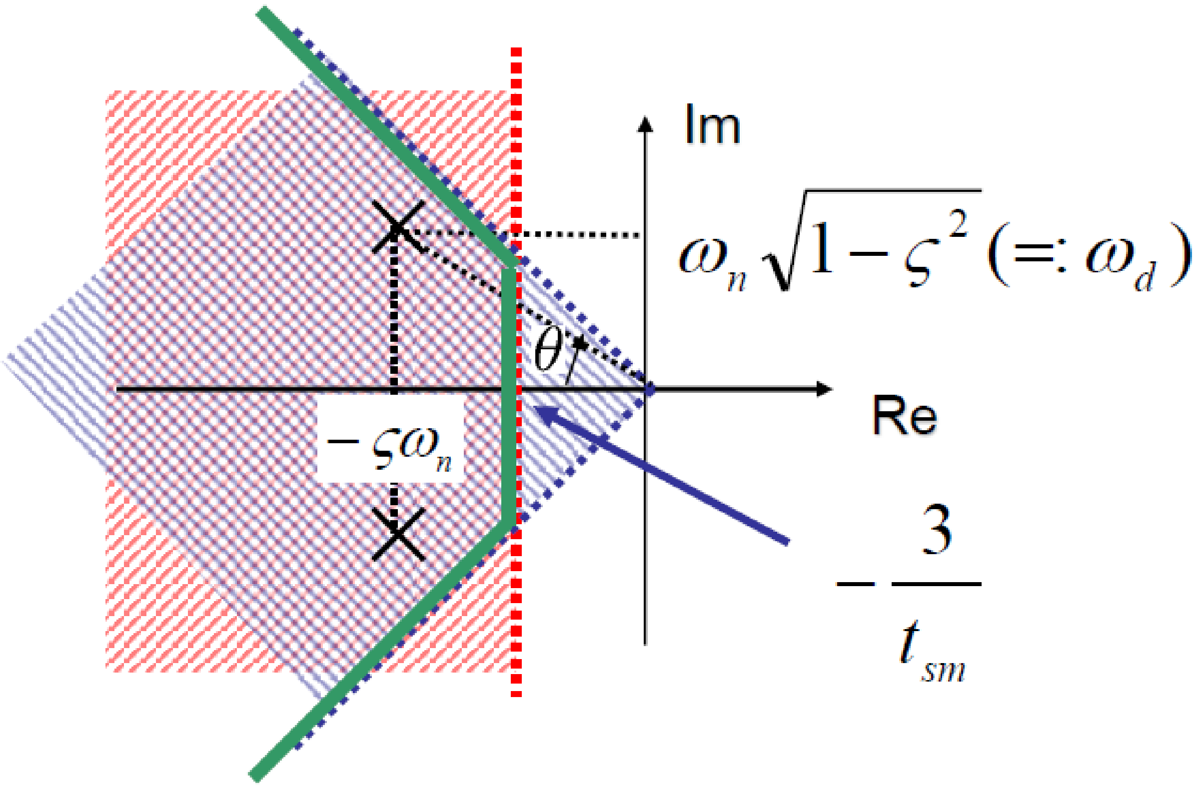
| $PO_m$ | $\theta_m$ |
|---|---|
| 4.3% | $\pi/4$ |
| 16.3% | $\pi/3$ |



# TR – 2nd Order System ( Design example)

- Combination of two requirements

$$\boxed{\varsigma\omega_n > \frac{3}{t_{sm}}} \quad \& \quad \boxed{\theta < \theta_m}$$



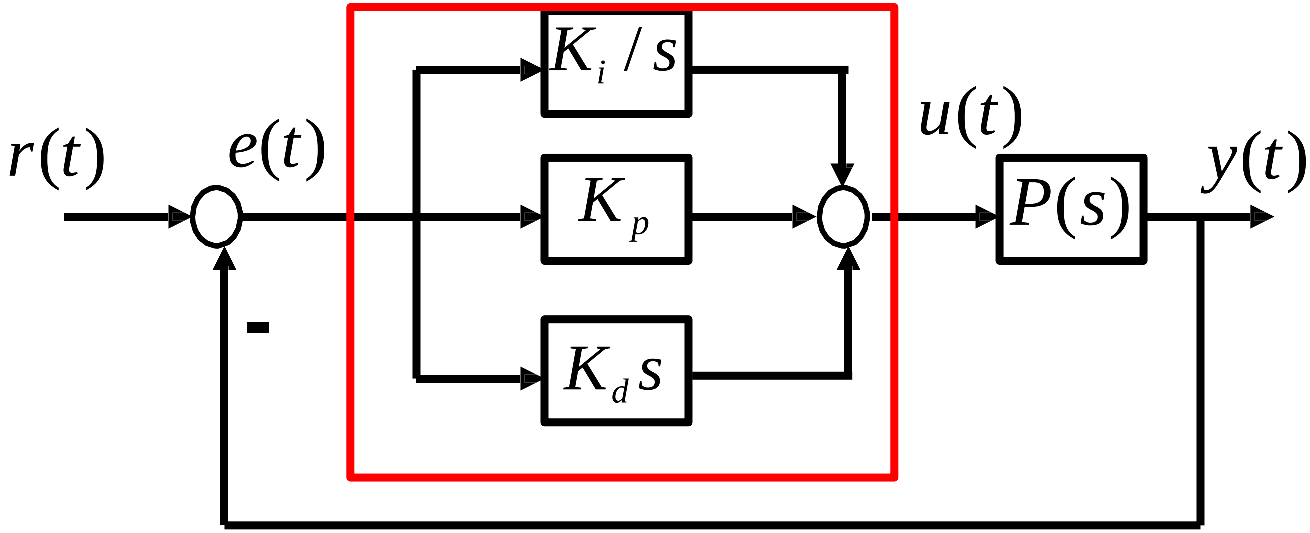
# TR – 2nd Order System (Summary)

- Transient response of 2nd order system is characterized by
  - Damping ratio $\zeta$ and undamped natural frequency $\omega$
  - Or in other words, pole locations

- Delay time and rise time are not so easy to characterize, and thus not covered in this course.

- For transient responses of high order systems, we need computer simulations.

# PID Controller




$$u(t) = \underbrace{K_p e(t)}_{\text{Proportional}} + \underbrace{K_i \int_0^t e(\tau) d\tau}_{\text{Integal}} + \underbrace{K_d \frac{de(t)}{dt}}_{\text{Derivative}}$$

$$C(s) = K_p + \frac{K_i}{s} + K_d s = K_p \left( 1 + \frac{1}{K_I s} + K_D s \right)$$

# PID Controller Remarks

- Most popular in process and robotics industries
  - Good performance
  - Functional simplicity (Operators can easily tune.)
- To avoid high frequency noise amplification, derivative term is implemented as

$$K_d s \approx \frac{K_d s}{\tau_d s + 1}$$

with $\tau_d$ much smaller than plant time constant.

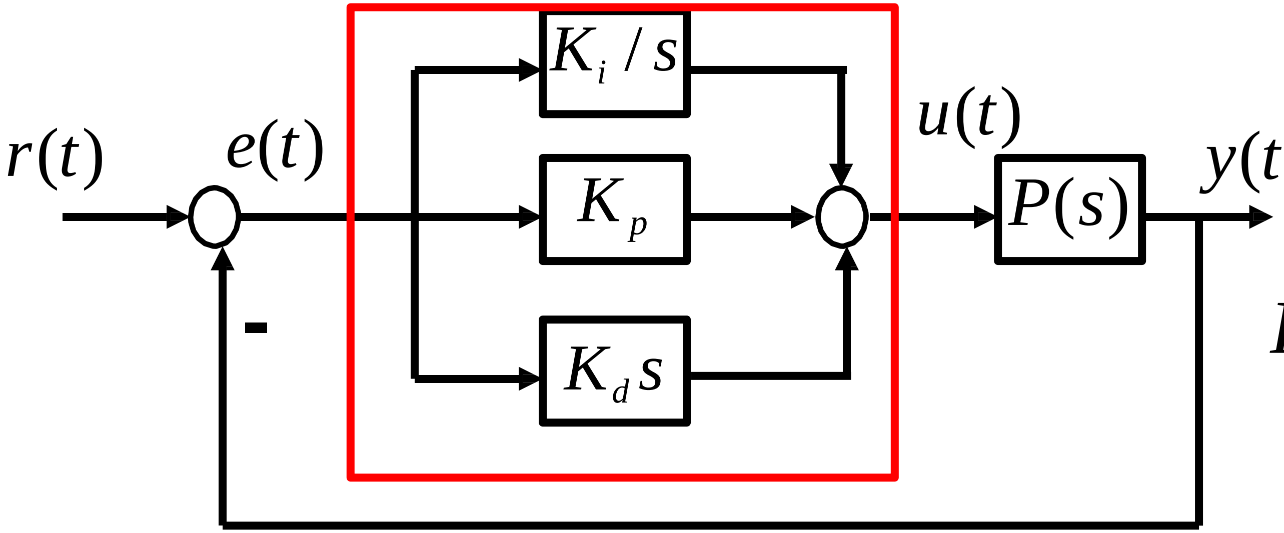
- PI controller

$$C(s) = K_p + \frac{K_i}{s}$$

- PD controller

$$C(s) = K_p + K_d s$$

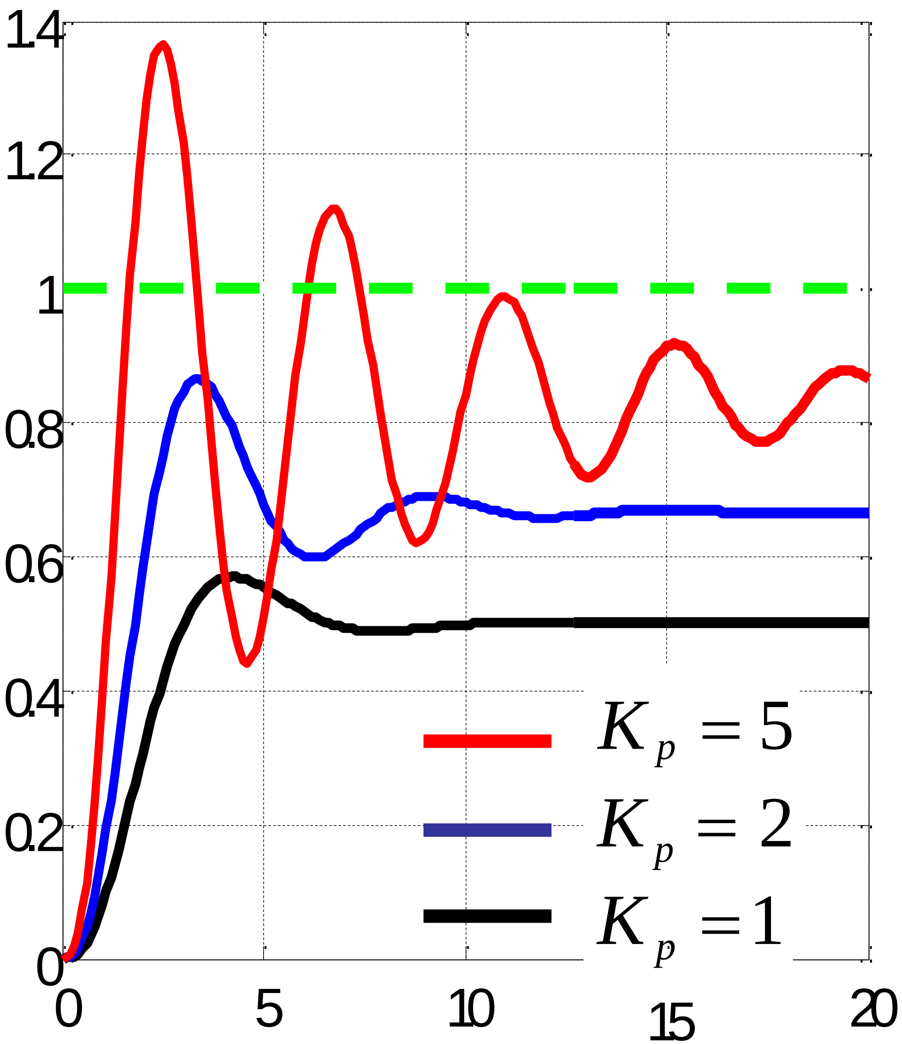
# A Simple Example (1)




$$P(s) = \frac{1}{(s+1)^3}$$

- We plot $y(t)$ for step reference $r(t)$ with
  - P controller
  - PI controller
  - PID controller

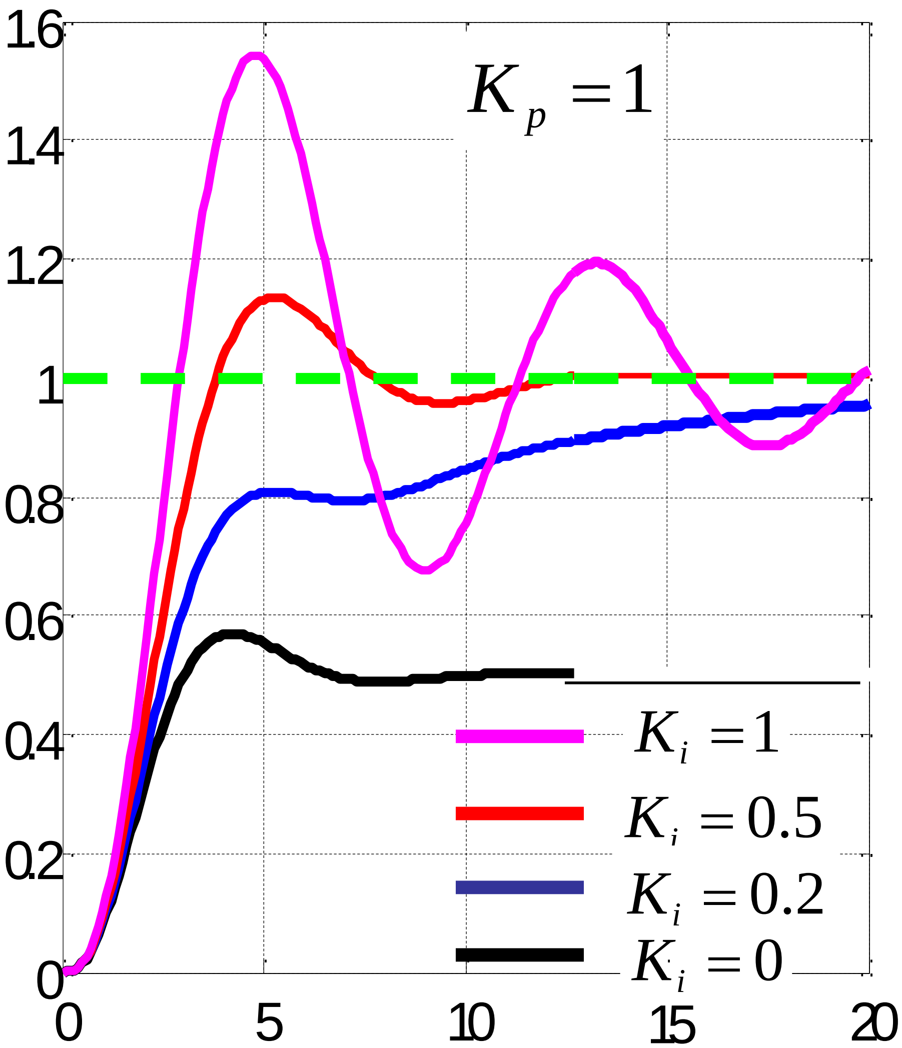
# A Simple Example (P Controller (2))




$$C(s) = K_p$$

- Simple
- Steady state error
  - Higher gain gives smaller error
- Stability
  - Higher gain gives faster and more oscillatory response

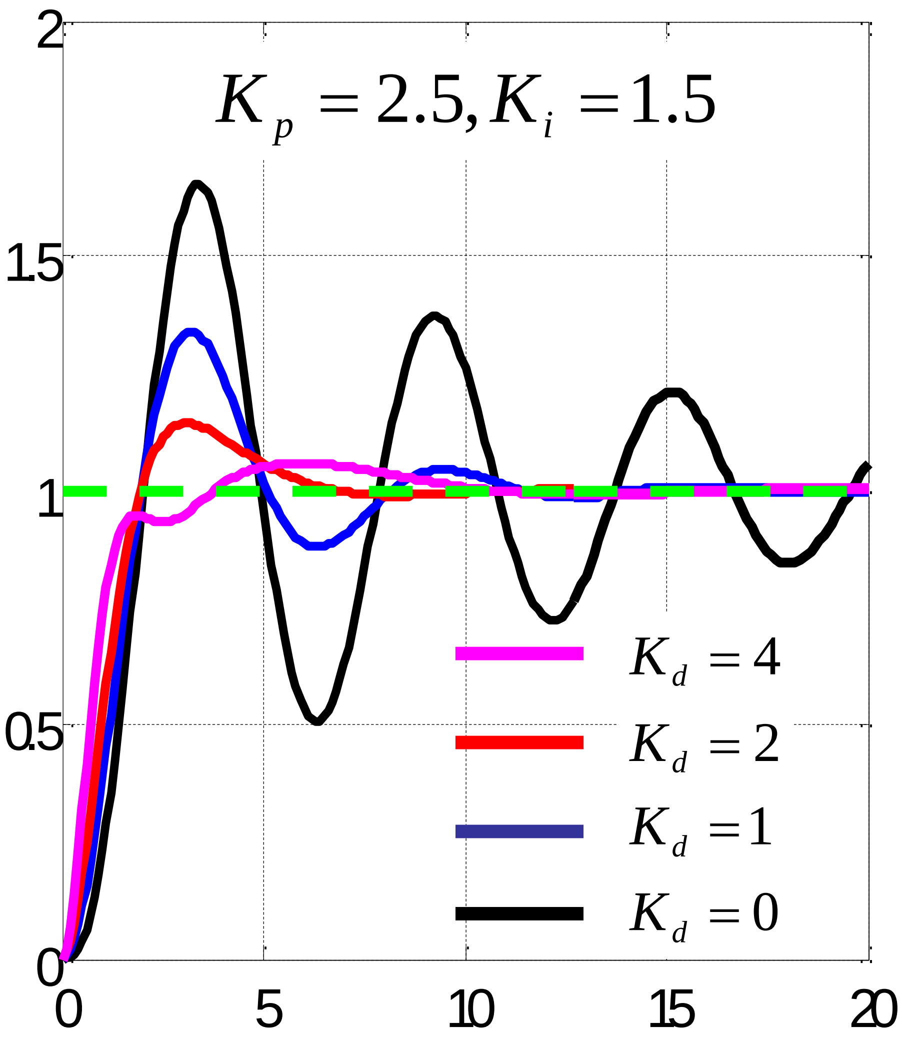
# A Simple Example (PI Controller (3))




$$C(s) = K_p + \frac{K_i}{s}$$

- Zero steady state error (provided that CL is stable.)
- Stability
  - Higher gain gives faster and more oscillatory response

# A Simple Example (PID Controller (4))




$$C(s) = K_p + \frac{K_i}{s} + K_d s$$

- Zero steady state error (due to integral control)
- Stability
  - Higher gain gives more damped response
- Too high gain worsen performance.

# How to Turn PID Parameters?

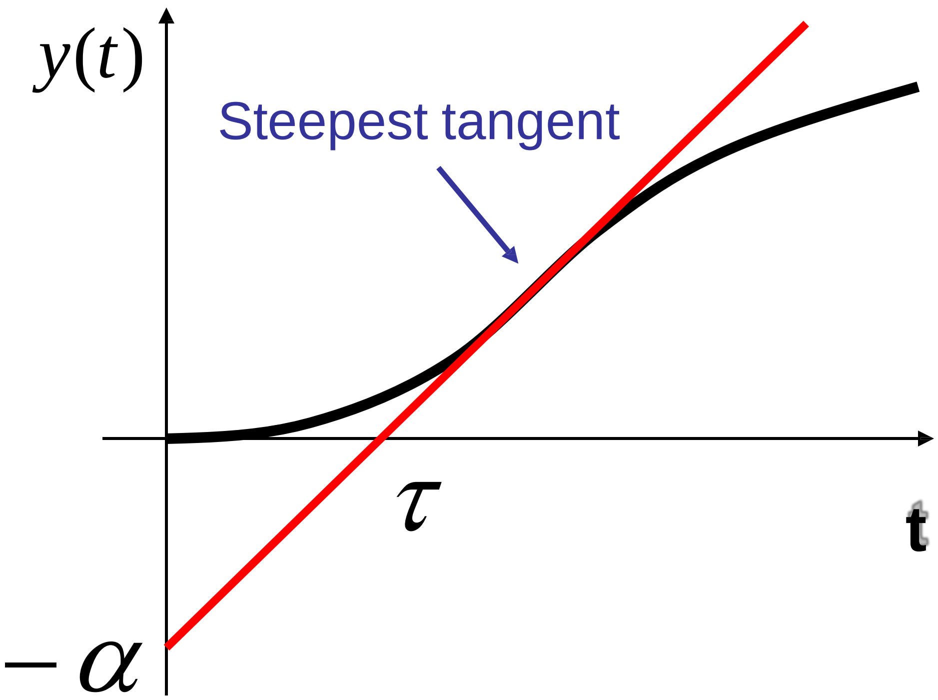
- Model-based
  - Root locus
  - Frequency response approach
  - Useful only when a model is available
  - Necessary if a system has to work at the first trial

- Empirical (without model)
  - Ziegler-Nichols tuning rule (1942)
  - Simple
  - Useful even if a system is too complex to model
  - Useful only when trial-and-error tuning is allowed

# Ziegler-Nichols PID Tuning Rules (1)

- Step response method (for only stable systems)

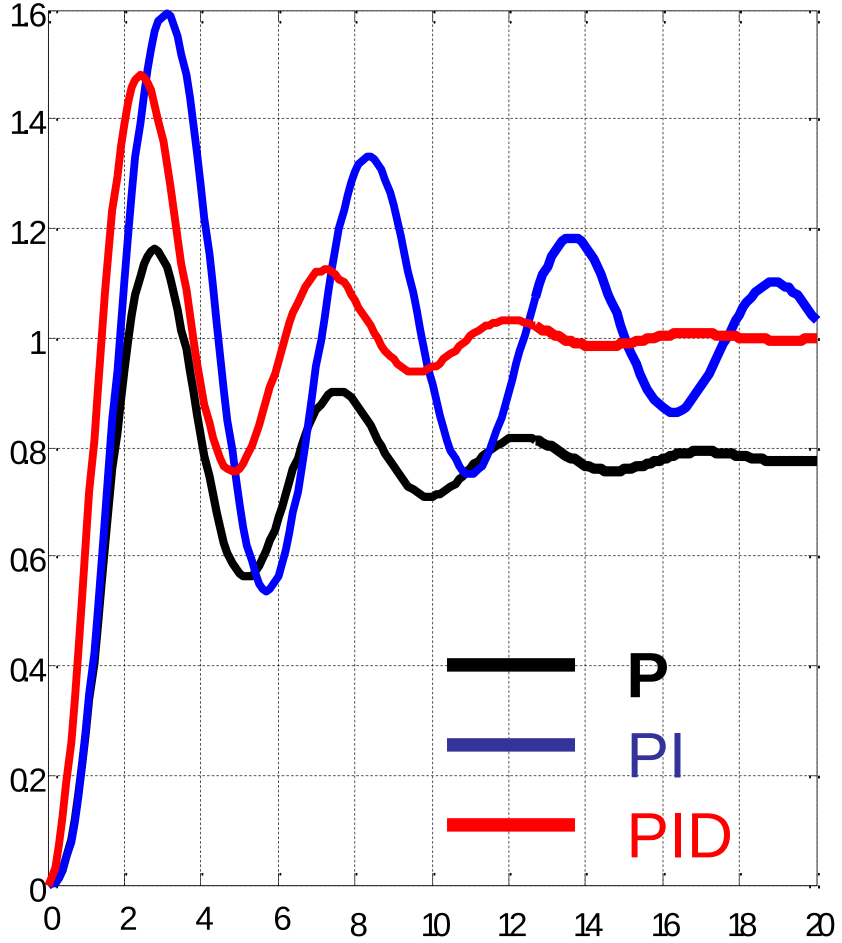
Open-loop step response → PID parameters




$$C(s) = K_p\left(1 + \frac{1}{T_I s} + T_d s\right)$$

| Type | $K_P$ | $T_I$ | $T_D$ |
|---|---|---|---|
| $P$ | $1/\alpha$ | | |
| $PI$ | $0.9/\alpha$ | $3\tau$ | |
| $PID$ | $1.2/\alpha$ | $2\tau$ | $0.5\tau$ |

# A Simple Example (Revisited (5))



# Direct PID Controller Tuning

- **Step 1:** Start with all three gains ($K_P, K_I, K_D$) equal to zero
- **Step 2:** Increase $K_P$ graduately until the system is marginally stable (slightly oscillation in output response observed), and set $K_P$ to half of the corresponding valve.
- **Step 3:** Do the same for $K_I$ (if required steady state response is not satisfactory) but try to keep $K_I$ as small as possible with satisfactory steady-state error.
- **Step 4:** Do the same for $K_D$ (if required transient response is not satisfactory) but try to keep $K_D$ as small as possible.
- **Step 5:** May repeat the process from Steps 2 to 4.

# PID Control Summary

- PID control
  - Most popular controller in industry
  - Model-free methods for design are available.
  - Simple controller structure
  - Simple controller tuning
  - Widely applicable

- Ziegler-Nichols tuning rules provide a starting point for fine tuning, rather than final settings of controller parameters in a single shot.

- Direct tuning is an easy way for PID controller tuning but require certain experience;

  Observe->tune-> observe->tune…