ME 343 – Control Systems

Lecture 13

September 21, 2009

Laplace Transform

Function *f(t)* of time

Piecewise continuous and exponential order $|f(t)| < Ke^{bt}$

$$F(s) = \int_{0-}^{\infty} f(t)e^{-st}dt \qquad L^{-1}[F(s)] = f(t) = \frac{1}{2\pi j}\int_{\alpha-j\infty}^{\alpha+j\infty} F(s)e^{st}ds$$

0- limit is used to capture transients and discontinuities at $t=0$

s is a complex variable $(\sigma + j\omega)$

There is a need to worry about regions of convergence of the integral

Units of s are sec^{-1}=Hz

A frequency

If *f(t)* is volts (amps) then *F(s)* is volt-seconds (amp-seconds)

Laplace Transform Properties

Linearity: (absolutely critical property)

$$L\{Af_1(t)+Bf_2(t)\}=AL\{f_1(t)\}+BL\{f_2(t)\}=AF_1(s)+BF_2(s)$$

Integration property: $$L\left\{\int_0^t f(\tau)d\tau\right\}=\frac{F(s)}{s}$$

Differentiation property: $$L\left\{\frac{df(t)}{dt}\right\}=sF(s)-f(0-)$$

$$L\left\{\frac{d^2f(t)}{dt^2}\right\}=s^2F(s)-sf(0-)-f'(0-)$$

$$L\left\{\frac{d^mf(t)}{dt^m}\right\}=s^mF(s)-s^{m-1}f(0-)-s^{m-2}f'(0-)-\cdots-f^{(m)}(0-)$$

Laplace Transform Properties

Translation properties:

s-domain translation: $$L\{e^{-\alpha t}f(t)\}=F(s+\alpha)$$

t-domain translation: $$L\{f(t-a)u(t-a)\}=e^{-as}F(s) \text{ for } a>0$$

Initial Value Property: $$\lim_{t\to 0+} f(t)=\lim_{s\to\infty} sF(s)$$

Final Value Property: $$\lim_{t\to\infty} f(t)=\lim_{s\to 0} sF(s)$$

If all poles of $F(s)$ are in the LHP

Laplace Transform Properties

Time Scaling: $L\{f(at)\} = \frac{1}{|a|}F(\frac{s}{a})$

Multiplication by time: $L\{tf(t)\} = -\frac{dF(s)}{ds}$

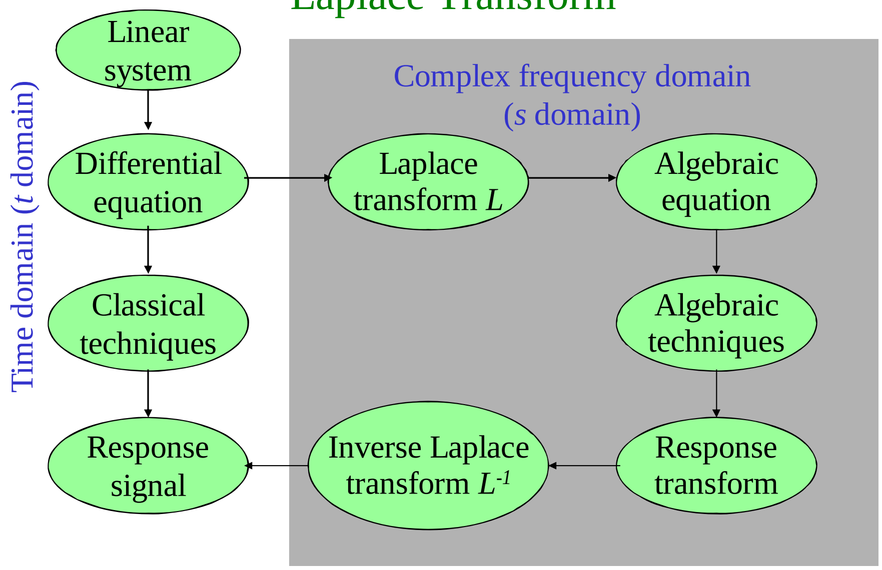
Convolution: $L\{\int_0^t f(\tau)g(t-\tau)d\tau\} = F(s)G(s)$

Time product: $L\{f(t)g(t)\} = \frac{1}{2\pi j}\int_{\sigma-j\omega}^{\sigma+j\omega} F(s)G(s-\lambda)d\lambda$

Laplace Transform Table

Signal	Waveform	Transform
impulse	$\delta(t)$	1
step	$u(t)$	$\frac{1}{s}$
ramp	$tu(t)$	$\frac{1}{s^2}$
exponential	$e^{-\alpha t}u(t)$	$\frac{1}{s+\alpha}$
damped ramp	$te^{-\alpha t}u(t)$	$\frac{1}{(s+\alpha)^2}$
sine	$\sin(\beta t)u(t)$	$\frac{\beta}{s^2+\beta^2}$
cosine	$\cos(\beta t)u(t)$	$\frac{s}{s^2+\beta^2}$
damped sine	$e^{-\alpha t}\sin(\beta t)u(t)$	$\frac{\beta}{(s+\alpha)^2+\beta^2}$
damped cosine	$e^{-\alpha t}\cos(\beta t)u(t)$	$\frac{s+\alpha}{(s+\alpha)^2+\beta^2}$

Laplace Transform



The diagram commutes

Same answer whichever way you go

Solving LTI ODE's via Laplace Transform

$$y^{(n)}+a_{n-1}y^{(n-1)}+\cdots+a_0y=b_mu^{(m)}+b_{m-1}u^{(m-1)}+\cdots+b_0u$$

Initial Conditions: $y^{(n-1)}(0),\ldots,y(0),u^{(m-1)}(0),\ldots,u(0)$

Recall $$L\left\{\frac{d^k f(t)}{dt^k}\right\}=s^kF(s)-\sum_{j=0}^{k-1}f^{(k-1-j)}(0)s^j$$

$$s^nY(s)-\sum_{j=0}^{n-1}y^{(n-1-j)}(0)s^j+\sum_{i=0}^{n-1}a_i\left[s^iY(s)-\sum_{j=0}^{i-1}y^{(i-1-j)}(0)s^j\right]=\sum_{i=0}^{m}b_i\left[s^iU(s)-\sum_{j=0}^{i-1}u^{(i-1-j)}(0)s^j\right]$$

$$Y(s)=\frac{b_ms^m+b_{m-1}s^{m-1}+\cdots+b_1s+b_0}{s^n+a_{n-1}s^{n-1}+\cdots+a_1s+a_0}U(s)+\frac{\sum_{i=0}^{n-1}a_i\sum_{j=0}^{i-1}y^{(i-1-j)}(0)s^j-\sum_{i=0}^{m}b_i\sum_{j=0}^{i-1}u^{(i-1-j)}(0)s^j}{s^n+a_{n-1}s^{n-1}+\cdots+a_1s+a_0}$$

For a given rational $U(s)$ we get $Y(s)=Q(s)/P(s)$

Computing Transfer Functions via Laplace Transform

$$y^{(n)} + a_{n-1}y^{(n-1)} + \cdots + a_0 y = b_{m-1}u^{(m-1)} + \cdots + b_0 u$$

Assume all <u>Initial Conditions Zero</u>:

$$\left(s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0\right)Y(s) = \left(b_{m-1}s^{m-1} + \cdots + b_1 s + b_0\right)U(s)$$

Output Input

$$Y(s) = \frac{b_{m-1}s^{m-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0}U(s) = \frac{B(s)}{A(s)}U(s)$$

$$H(s) = \frac{Y(s)}{U(s)} = \frac{b_{m-1}s^{m-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0}$$

$$= K\frac{(s-z_1)(s-z_2)\cdots(s-z_m)}{(s-p_1)(s-p_2)\cdots(s-p_n)}$$

Rational Functions

We shall mostly be dealing with TFs which are rational functions – ratios of polynomials in s

$$F(s) = \frac{b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0}{a_n s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0}$$

$$= K\frac{(s-z_1)(s-z_2)\cdots(s-z_m)}{(s-p_1)(s-p_2)\cdots(s-p_n)}$$

p_i are the poles and z_i are the zeros of the function

K is the scale factor or (sometimes) gain

A proper rational function has $n \geq m$

A strictly proper rational function has $n > m$

An improper rational function has $n < m$

Partial Fraction Expansion - Residues at Simple Poles

Functions of a complex variable with isolated, finite order poles have *residues* at the poles

$$F(s) = K\frac{(s-z_1)(s-z_2)\cdots(s-z_m)}{(s-p_1)(s-p_2)\cdots(s-p_n)} = \frac{k_1}{(s-p_1)} + \frac{k_2}{(s-p_2)} + \cdots + \frac{k_n}{(s-p_n)}$$

$$(s-p_i)F(s) = \frac{k_1(s-p_i)}{(s-p_1)} + \frac{k_2(s-p_i)}{(s-p_2)} + \cdots + k_i + \cdots + \frac{k_n(s-p_i)}{(s-p_n)}$$

Residue at a simple pole: $k_i = \lim_{s\to p_i}(s-p_i)F(s)$

Partial Fraction Expansion - Residues at multiple poles

$$F(s) = K\frac{(s-z_1)(s-z_2)\cdots(s-z_m)}{(s-p_1)^r} = \frac{k_1}{(s-p_1)} + \frac{k_2}{(s-p_1)^2} + \cdots + \frac{k_r}{(s-p_1)^r}$$

Residue at a multiple pole:

$$k_j = \frac{1}{(r-j)!}\lim_{s\to p_i}\frac{d^{r-j}}{ds^{r-j}}\left[(s-p_i)^r F(s)\right], \qquad j = 1\cdots r$$

Partial Fraction Expansion - Residues at Complex Poles

Compute residues at the poles $\lim_{s \to a} (s-a)F(s)$

Bundle complex conjugate pole pairs into second-order terms if you want … but you will need to be careful!

$$(s-\alpha-j\beta)(s-\alpha+j\beta) = \left[s^2 - 2\alpha s + \left(\alpha^2+\beta^2\right)\right]$$

Inverse Laplace Transform is a sum of complex exponentials. But the answer will be <u>real</u>.

Inverting Laplace Transforms in Practice

We have a table of inverse LTs

Write F(s) as a partial fraction expansion

$$\begin{aligned}F(s) &= \frac{b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0}{a_n s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0} \\ &= K\frac{(s-z_1)(s-z_2)\cdots(s-z_m)}{(s-p_1)(s-p_2)\cdots(s-p_n)} \\ &= \frac{\alpha_1}{(s-p_1)} + \frac{\alpha_2}{(s-p_2)} + \frac{\alpha_{31}}{(s-p_3)} + \frac{\alpha_{32}}{(s-p_3)^2} + \frac{\alpha_{33}}{(s-p_3)^3} + \ldots + \frac{\alpha_q}{(s-p_q)}\end{aligned}$$

Now appeal to linearity to invert via the table

- Surprise!
- Nastiness: computing the partial fraction expansion is best done by calculating the residues

Not Strictly Proper Laplace Transforms

Find the inverse LT of $F(s)=\dfrac{s^3+6s^2+12s+8}{s^2+4s+3}$

Convert to polynomial plus strictly proper rational function

Use polynomial division

$$\begin{aligned} F(s) &= s+2+\frac{s+2}{s^2+4s+3} \\ &= s+2+\frac{0.5}{s+1}+\frac{0.5}{s+3} \end{aligned}$$

Invert as normal

$$f(t)=\left[\frac{d\delta(t)}{dt}+2\delta(t)+0.5e^{-t}+0.5e^{-3t}\right]u(t)$$

Impulse Response

Dirac's delta: $\int_0^\infty u(\tau)\delta(t-\tau)d\tau=u(t)$

Integration is a limit of a sum

$\Downarrow$

u(t) is represented as a sum of impulses

By superposition principle, we only need unit impulse response

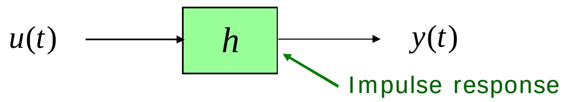
$h(t-\tau)$ Response at t to an impulse applied at τ of amplitude $u(\tau)$

System Response: $u(t) \longrightarrow \boxed{h} \longrightarrow y(t)$

$$y(t)=\int_0^\infty u(\tau)h(t-\tau)d\tau$$

Impulse Response

t-domain:

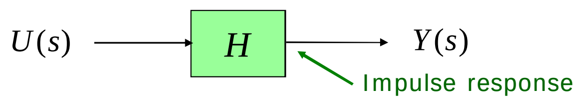


$$y(t)=\int_0^\infty u(\tau)h(t-\tau)d\tau \qquad u(t)=\delta(t)\Rightarrow y(t)=h(t)$$

The system response is obtained by convolving the input with the impulse response of the system.

Convolution: $L\{\int_0^\infty u(\tau)h(t-\tau)d\tau\}=H(s)U(s)$

s-domain:

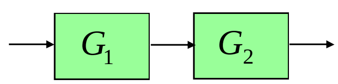


$$Y(s)=H(s)U(s) \qquad u(t)=\delta(t)\Rightarrow U(s)=1\Rightarrow Y(s)=H(s)$$

The system response is obtained by multiplying the transfer function and the Laplace transform of the input.

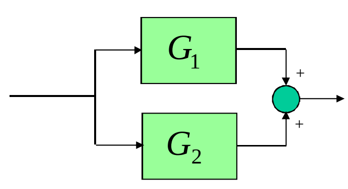
Block Diagrams

Series:



$$G=G_1G_2$$

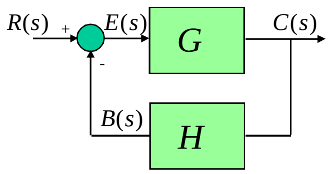
Parallel:



$$G=G_1+G_2$$

Block Diagrams

Negative Feedback:



R	Reference input
$E = R - B$	Error signal
$C = GE$	Output
$B = HC$	Feedback signal

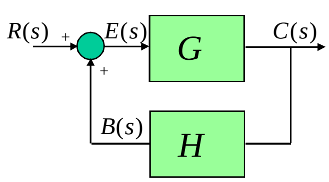
$$C = GR - GHC \Rightarrow (1+GH)C = GR \Rightarrow \frac{C}{R} = \frac{G}{(1+GH)}$$

$$E = R - HGE \Rightarrow (1+GH)E = R \Rightarrow \frac{E}{R} = \frac{1}{(1+GH)}$$

Rule: Transfer Function=Forward Gain/(1+Loop Gain)

Block Diagrams

Positive Feedback:



R	Reference input
$E = R + B$	Error signal
$C = GE$	Output
$B = HC$	Feedback signal

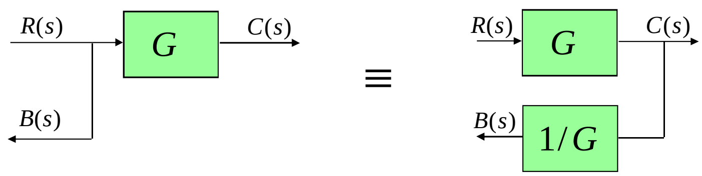
$$C = GR + GHC \Rightarrow (1-GH)C = GR \Rightarrow \frac{C}{R} = \frac{G}{(1-GH)}$$

$$E = R + HGE \Rightarrow (1-GH)E = R \Rightarrow \frac{E}{R} = \frac{1}{(1-GH)}$$

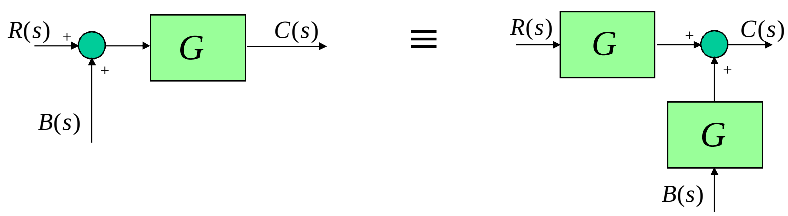
Rule: Transfer Function=Forward Gain/(1-Loop Gain)

Block Diagrams

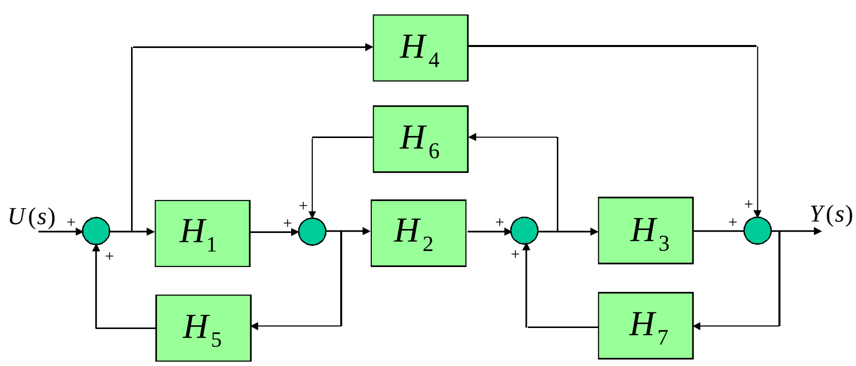
Moving through a branching point:



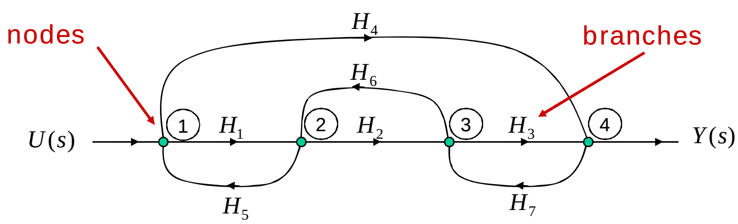
Moving through a summing point:



Mason's Rule



Signal Flow Graph



Mason's Rule

Path: a sequence of connected branches in the direction of the signal flow without repetition
Loop: a closed path that returns to its starting node
Forward path: connects input and output

$$G(s)=\frac{Y(s)}{U(s)}=\frac{1}{\Delta}\sum_i G_i\Delta_i$$

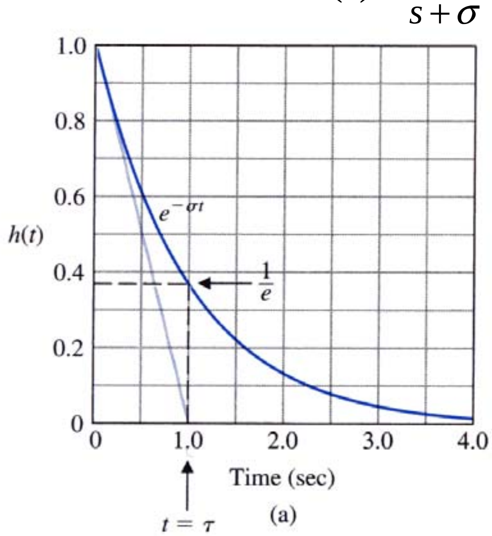
G_i = gain of the ith forward path
Δ = the system determinant

$$\begin{aligned}&= 1-\sum(\text{all loop gains})\\ &+\sum(\text{gain products of all possible two loops that do not touch})\\ &-\sum(\text{gain products of all possible three loops that do not touch})\\ &+\cdots\end{aligned}$$

Δ_i = value of Δ for the part of the graph that does not touch the ith forward path

Time Response vs. Poles

Real pole: $H(s)=\dfrac{1}{s+\sigma}\Rightarrow h(t)=e^{-\sigma t}$ Impulse Response



$\sigma>0$ Stable

$\sigma<0$ Unstable

$\tau=\dfrac{1}{\sigma}$ Time Constant

Time Response vs. Poles

Real pole:

$$H(s) = \frac{\sigma}{s+\sigma} \Rightarrow h(t) = \sigma e^{-\sigma t}$$ Impulse Response

$$\tau = \frac{1}{\sigma}$$ Time Constant

$$Y(s) = \frac{\sigma}{s+\sigma}\frac{1}{s} \Rightarrow y(t) = 1 - e^{-\sigma t}$$ Step Response

Time Response vs. Poles

Complex poles: $$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$ Impulse Response

$$= \frac{\omega_n^2}{(s+\zeta\omega_n)^2 + \omega_n^2(1-\zeta^2)}$$

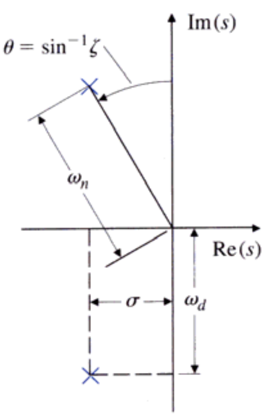
ω_n : Undamped natural frequency

ζ : Damping ratio

$$H(s) = \frac{\omega_n^2}{(s+\sigma+j\omega_d)(s+\sigma-j\omega_d)}$$

$$= \frac{\omega_n^2}{(s+\sigma)^2 + \omega_d^2}$$

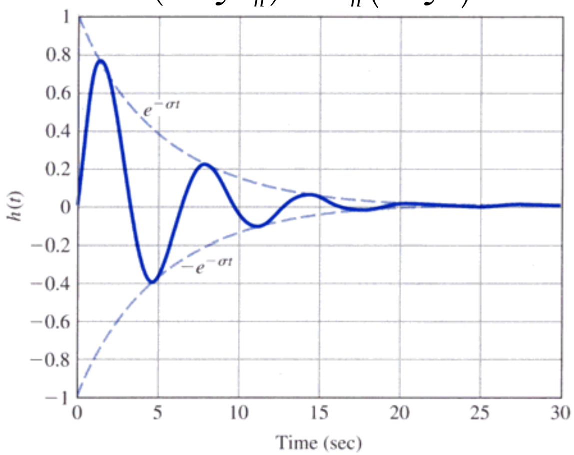
$$\sigma = \zeta\omega_n, \omega_d = \omega_n\sqrt{1-\zeta^2}$$



Time Response vs. Poles

Complex poles:

$$H(s) = \frac{\omega_n^2}{(s+\zeta\omega_n)^2 + \omega_n^2\left(1-\zeta^2\right)} \rightarrow h(t) = \frac{\omega_n}{\sqrt{1-\zeta^2}} e^{-\sigma t}\sin(\omega_d t)$$



Impulse Response

$\sigma > 0$ Stable

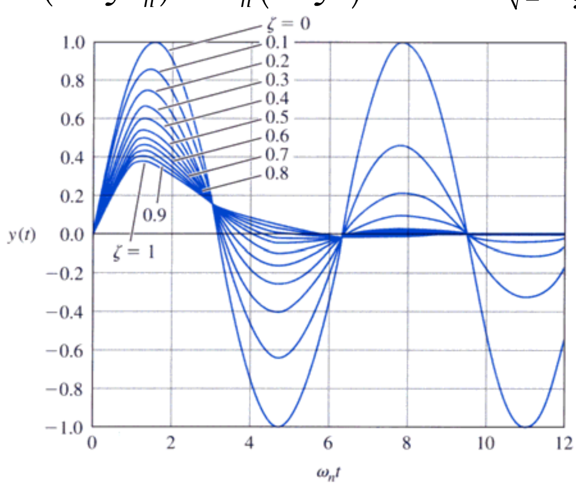
$\sigma < 0$ Unstable

ME 343 – Control Systems – Fall 2009 171

Time Response vs. Poles

Complex poles:

$$H(s) = \frac{\omega_n^2}{(s+\zeta\omega_n)^2 + \omega_n^2\left(1-\zeta^2\right)} \rightarrow h(t) = \frac{\omega_n}{\sqrt{1-\zeta^2}} e^{-\sigma t}\sin(\omega_d t)$$



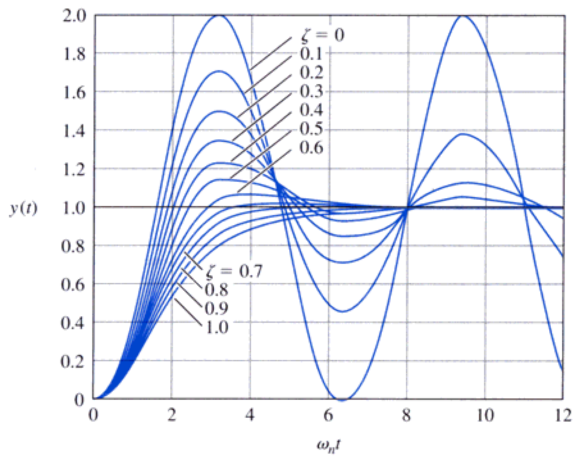
Impulse Response

ME 343 – Control Systems – Fall 2009 172

Time Response vs. Poles

Complex poles:

$$Y(s)=\frac{\omega_n^2}{(s+\zeta\omega_n)^2+\omega_n^2(1-\zeta^2)}\frac{1}{s}\rightarrow y(t)=1-e^{-\sigma t}\left[\cos(\omega_d t)+\frac{\sigma}{\omega_d}\sin(\omega_d t)\right]$$



Step Response

Time Response vs. Poles

Complex poles:

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

$$=\frac{\omega_n^2}{(s+\zeta\omega_n)^2+\omega_n^2(1-\zeta^2)}$$

CASES:

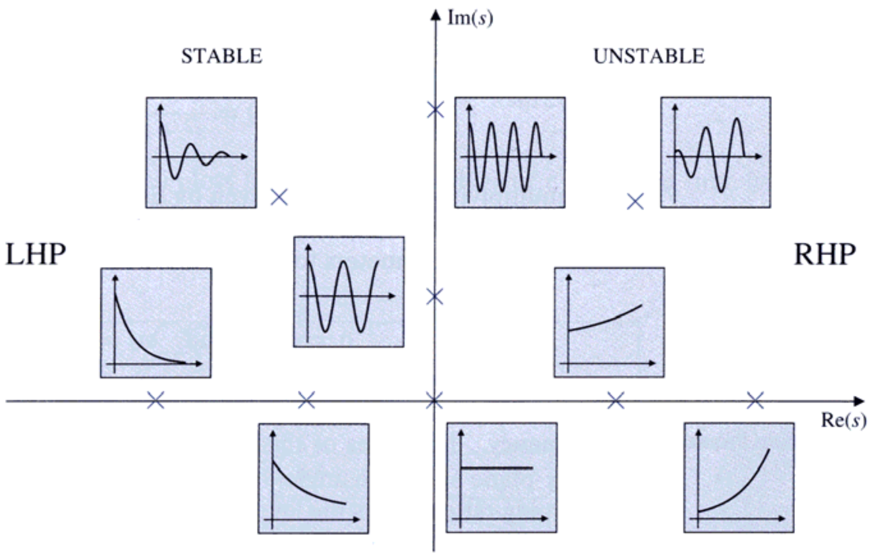
$\zeta=0: s^2+\omega_n^2$ Undamped

$\zeta<1: (s+\zeta\omega_n)^2+\omega_n^2(1-\zeta^2)$ Underdamped

$\zeta=1: (s+\omega_n)^2$ Critically damped

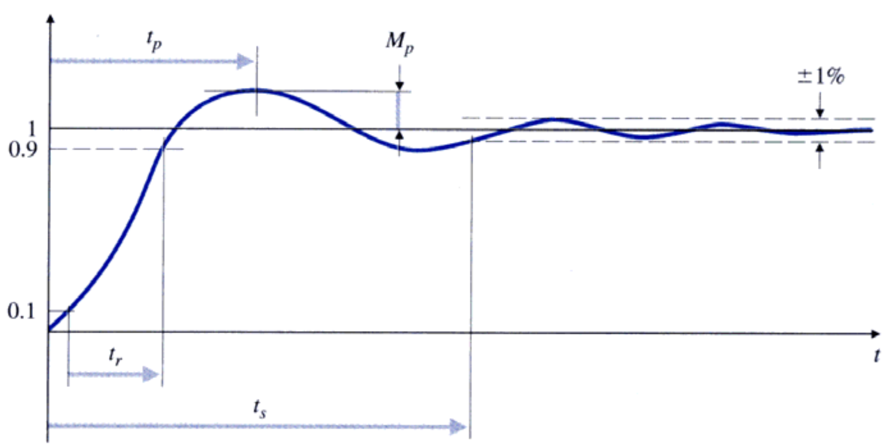
$\zeta>1: \left[s+\left(\zeta+\sqrt{\zeta^2-1}\right)\omega_n\right]\left[s+\left(\zeta-\sqrt{\zeta^2-1}\right)\omega_n\right]$ Overdamped

Time Response vs. Poles



ME 343 – Control Systems – Fall 2009 175

Time Domain Specifications



ME 343 – Control Systems – Fall 2009 176

Time Domain Specifications

1- The **rise time** t_r is the time it takes the system to reach the vicinity of its new set point

2- The **settling time** t_s is the time it takes the system transients to decay

3- The **overshoot** M_p is the maximum amount the system overshoot its final value divided by its final value

4- The **peak time** t_p is the time it takes the system to reach the maximum overshoot point

$$t_p = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} \qquad t_r \cong \frac{1.8}{\omega_n}$$

$$M_p = e^{-\pi\frac{\zeta}{\sqrt{1-\zeta^2}}} \qquad t_s = \frac{4.6}{\zeta\omega_n}$$

Time Domain Specifications

Design specification are given in terms of

$$t_r, t_p, M_p, t_s$$

These specifications give the position of the poles

$$\omega_n, \zeta \Rightarrow \sigma, \omega_d$$

Example: Find the pole positions that guarantee

$$t_r \le 0.6\sec, M_p < 10\%, t_s \le 3\sec$$

Time Domain Specifications

Additional poles:

1- can be neglected if they are sufficiently to the left of the dominant ones.

2- can increase the rise time if the extra pole is within a factor of 4 of the real part of the complex poles.

Zeros:

1- a zero near a pole reduces the effect of that pole in the time response.

2- a zero in the LHP will increase the overshoot if the zero is within a factor of 4 of the real part of the complex poles (due to differentiation).

3- a zero in the RHP (nonminimum phase zero) will depress the overshoot and may cause the step response to start out in the wrong direction.

Stability

$$\frac{Y(s)}{R(s)} = \frac{b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0}$$

$$\frac{Y(s)}{R(s)} = K\frac{(s-z_1)(s-z_2)\cdots(s-z_m)}{(s-p_1)(s-p_2)\cdots(s-p_n)}$$

$$\frac{Y(s)}{R(s)} = \frac{k_1}{(s-p_1)} + \frac{k_2}{(s-p_2)} + \cdots + \frac{k_n}{(s-p_n)}$$

Impulse response:

$$R(s) = 1 \Rightarrow Y(s) = \frac{k_1}{(s-p_1)} + \frac{k_2}{(s-p_2)} + \cdots + \frac{k_n}{(s-p_n)}$$

$$y(t) = k_1 e^{p_1 t} + k_2 e^{p_2 t} + \cdots + k_n e^{p_n t}$$

Stability

$$y(t) = k_1 e^{p_1 t} + k_2 e^{p_2 t} + \cdots + k_n e^{p_n t}$$

We want: $e^{p_i t} \xrightarrow[t \to \infty]{} 0 \qquad \forall i = 1 \ldots n$

Definition: A system is asymptotically stable (a.s.) if

$$\mathrm{Re}\{p_i\} < 0 \qquad \forall i$$

Characteristic polynomial: $a(s) = s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0$

Characteristic equation: $a(s) = 0$

Stability

Necessary condition for asymptotical stability (a.s.):

$$a_i > 0 \qquad \forall i$$

Use this as the first test!

If any $a_i < 0$, the the system is UNSTABLE!

Routh's Criterion

Necessary and sufficient condition
Do not have to find the roots p_i!

Routh's Array:

$$
\begin{array}{lllll}
s^n & 1 & a_2 & a_4 & \cdots \\
s^{n-1} & a_1 & a_3 & a_5 & \cdots \\
s^{n-2} & b_1 & b_2 & b_3 & \\
s^{n-3} & c_1 & c_2 & c_3 & \\
s^{n-4} & d_1 & d_2 & & \\
\vdots & & & & \\
s^0 & a_n & & &
\end{array}
\quad a_n
$$

Depends on whether
n is even or odd

$$
b_1 = \frac{a_1 a_2 - a_3}{a_1}, \quad b_2 = \frac{a_1 a_4 - a_5}{a_1}, \quad b_3 = \frac{a_1 a_6 - a_7}{a_1} \quad \cdots
$$
$$
c_1 = \frac{b_1 a_3 - a_1 b_2}{b_1}, \quad c_2 = \frac{b_1 a_5 - a_1 b_3}{b_1}, \quad \cdots
$$
$$
d_1 = \frac{c_1 b_2 - b_1 c_2}{c_1}, \quad d_2 = \frac{c_1 b_3 - b_1 c_3}{c_1}, \quad \cdots
$$
$$
\vdots \qquad \vdots
$$

Routh's Criterion

How to remember this?

Routh's Array:

$$
\begin{array}{lllll}
s^n & m_{11} & m_{12} & m_{13} & \cdots \\
s^{n-1} & m_{21} & m_{22} & m_{23} & \cdots \\
s^{n-2} & m_{31} & m_{32} & m_{33} & \cdots \\
s^{n-3} & m_{41} & m_{42} & m_{43} & \cdots \\
\vdots & \vdots & \vdots & \vdots &
\end{array}
$$

$$
m_{1,j} = a_{2j-2},
$$
$$
m_{2,j} = a_{2j-1},
$$
$$
m_{i,j} = -\frac{\begin{vmatrix} m_{i-2,1} & m_{i-2,j+1} \\ m_{i-1,1} & m_{i-1,j+1} \end{vmatrix}}{m_{i-1,1}}, \forall i \geq 3
$$

Routh's Criterion

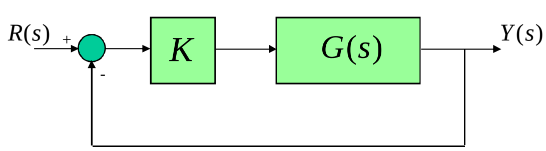
The criterion:

• The system is **asymptotically stable** if and only if all the elements in the first column of the Routh's array are positive

• The number of roots with positive real parts is equal to the number of sign changes in the first column of the Routh array

Routh's Criterion

Determine the range of K over which the system is stable



Routh's Criterion

Special Case I: Zero in the first column

We replace the zero with a small positive constant $\varepsilon>0$ and proceed as before. We then apply the stability criterion by taking the limit as $\varepsilon\to 0$

Special Case II: Entire row is zero

This indicates that there are complex conjugate pairs. If the ith row is zero, we form an auxiliary equation from the previous nonzero row:

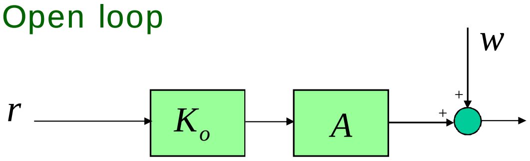
$$a_1(s)=\beta_1 s^{i+1}+\beta_2 s^{i-1}+\beta_3 s^{i-3}+\cdots$$

Where β_i are the coefficients of the $(i+1)$th row in the array. We then replace the ith row by the coefficients of the derivative of the auxiliary polynomial.

Properties of Feedback

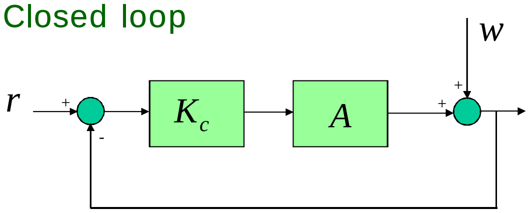
Disturbance Rejection:

Open loop



$$y=K_o A r+w$$

Closed loop



$$y=\frac{K_c A}{1+K_c A}r+\frac{1}{1+K_c A}w$$

Properties of Feedback

Disturbance Rejection:

Choose control s.t. for $w=0, y \approx r$

Open loop: $K_o = \frac{1}{A} \Rightarrow y = r + w$

Closed loop: $K_c >> \frac{1}{A} \Rightarrow y \approx r + 0w = r$

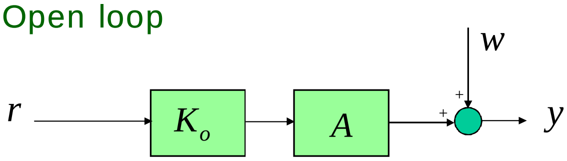
Feedback allows attenuation of disturbance without having access to it (without measuring it)!!!

IMPORTANT: High gain is dangerous for dynamic response!!!

Properties of Feedback

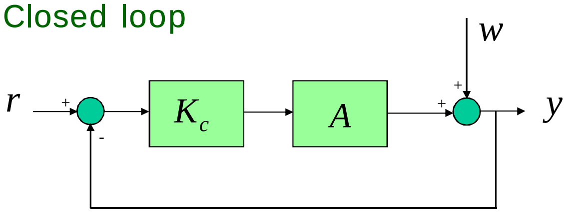
Sensitivity to Gain Plant Changes

Open loop



$$T_o = \left(\frac{y}{r}\right)_o = AK_o$$

Closed loop



$$T_c = \left(\frac{y}{r}\right)_c = \frac{AK_c}{1 + AK_c}$$

Properties of Feedback

Sensitivity to Gain Plant Changes

Let the plant gain be $A+\delta A$

Open loop: $$\frac{\delta T_o}{T_o}=\frac{\delta A}{A}$$

Closed loop: $$\frac{\delta T_c}{T_c}=\frac{\delta A}{A}\frac{1}{1+AK_c}<<\frac{\delta A}{A}=\frac{\delta T_o}{T_o}$$

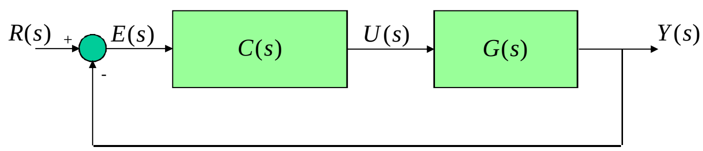
Feedback reduces sensitivity to plant variations!!!

Sensitivity: $$S_A^T=\frac{dT/T}{dA/A}=\frac{A}{T}\frac{dT}{dA}$$

Example: $$S_A^{T_c}=\frac{1}{1+AK_c},S_A^{T_o}=1$$

Steady-state Tracking

The Unity Feedback Case



Want $E(s)/R(s)=0$. $$\frac{E(s)}{R(s)}=\frac{1}{1+C(s)G(s)}$$

Test Inputs: $$r(t)=\frac{t^k}{k!}1(t)$$ $$R(s)=\frac{1}{s^{k+1}}$$

$k=0$: step (position)

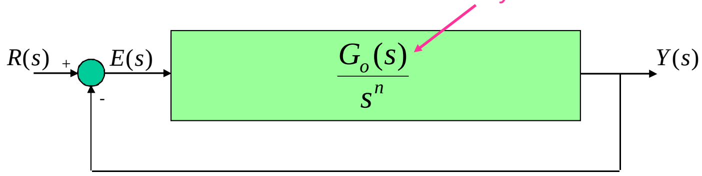
$k=1$: ramp (velocity)

$k=2$: parabola (acceleration)

Steady-state Tracking

The Unity Feedback Case

Type n
System



$$C(s)G(s) = \frac{G_o(s)}{s^n}, E(s) = \frac{1}{1+\frac{G_o(s)}{s^n}} R(s), R(s) = \frac{1}{s^{k+1}}$$

Steady State Error:

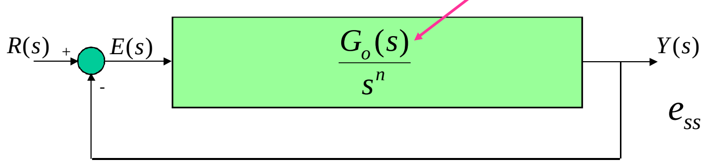
Final Value Theorem

$$e_{ss} = \lim_{t\to\infty} e(t) = \lim_{s\to 0} sE(s) = \lim_{s\to 0} s \frac{1}{1+\frac{G_o(s)}{s^n}} \frac{1}{s^{k+1}} = \lim_{s\to 0} \frac{s^n}{s^n + G_o(s)} \frac{1}{s^k} = \lim_{s\to 0} \frac{s^{n-k}}{s^n + G_o(0)}$$

Steady-state Tracking

The Unity Feedback Case

Type n
System



$$e_{ss} = \lim_{s\to 0} \frac{s^{n-k}}{s^n + G_o(0)}$$

Steady State Error:

Type (n)	Input (k): Step ($k=0$)	Ramp ($k=1$)	Parabola ($k=2$)
Type 0	$\frac{1}{1+G_o(0)} = \frac{1}{1+\lim_{s\to 0} C(s)G(s)} = \frac{1}{1+K_p}$	∞	∞
Type 1	0	$\frac{1}{G_o(0)} = \frac{1}{\lim_{s\to 0} sC(s)G(s)} = \frac{1}{K_v}$	∞
Type 2	0	0	$\frac{1}{G_o(0)} = \frac{1}{\lim_{s\to 0} s^2C(s)G(s)} = \frac{1}{K_a}$

Steady-state Tracking

$$K_p = \lim_{s\to 0} C(s)G(s) \qquad n=0 \qquad \text{Position Constant}$$

$$K_v = \lim_{s\to 0} sC(s)G(s) \qquad n=1 \qquad \text{Velocity Constant}$$

$$K_a = \lim_{s\to 0} s^2C(s)G(s) \qquad n=2 \qquad \text{Acceleration Constant}$$

n: Degree of the poles of *CG(s)* at the origin (the number of integrators in the loop with unity gain feedback)

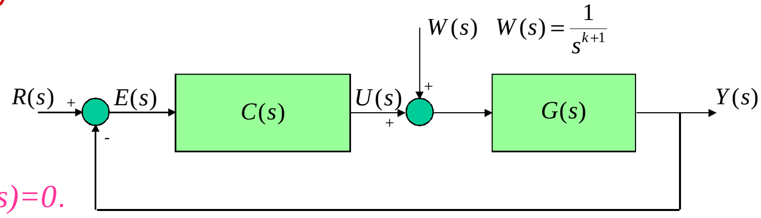
- Applying integral control to a plant with no zeros at the origin makes the system type ≥ I
- All this is true ONLY for unity feedback systems
- Since in Type I systems $e_{ss}=0$ for any *CG(s)*, we say that the system type is a robust property.

Steady-state Disturbance Rejection

The Unity Feedback Case

$$w(t) = \frac{t^k}{k!}1(t)$$

$$W(s) = \frac{1}{s^{k+1}}$$



Set *r=0*.
Want *Y(s)/W(s)=0*.

$$\frac{Y(s)}{W(s)} = \frac{G(s)}{1+C(s)G(s)} = T(s) = s^n T_o(s)$$

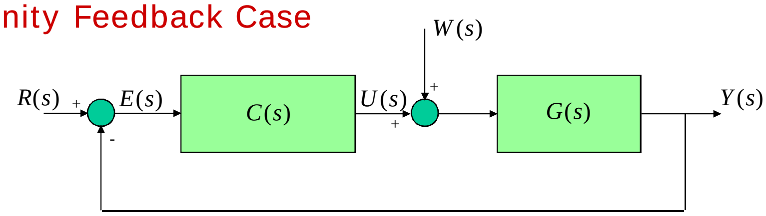
Steady State Error: *e=r-y=-y*

Final Value Theorem

$$-e_{ss} = y_{ss} = \lim_{t\to\infty} y(t) = \lim_{s\to 0} sY(s) = \lim_{s\to 0} sT(s)\frac{1}{s^{k+1}} = \lim_{s\to 0} T_o(s)\frac{s^n}{s^k}$$

Steady-state Disturbance Rejection

The Unity Feedback Case



Steady State Output:

Type (n)	Step ($k=0$)	Ramp ($k=1$)	Parabola ($k=2$)
	Disturbance (k)		
Type 0	*	∞	∞
Type 1	0	*	∞
Type 2	0	0	*

$0 < * < \infty$

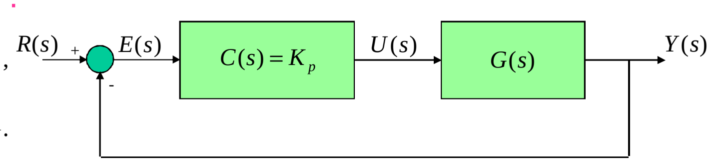
PID Controller

PID: Proportional – Integral – Derivative

P Controller:

$$\frac{Y(s)}{R(s)} = \frac{C(s)G(s)}{1+C(s)G(s)},$$

$$\frac{E(s)}{R(s)} = \frac{1}{1+C(s)G(s)}.$$



$$u(t) = K_p e(t), \quad U(s) = K_p E(s)$$

Step Reference:

$$R(s) = \frac{1}{s} \Rightarrow e_{ss} = \lim_{s\to 0} sE(s) = \lim_{s\to 0} s \frac{1}{1+K_p G(s)} \frac{1}{s} = \frac{1}{1+K_p G(0)}$$

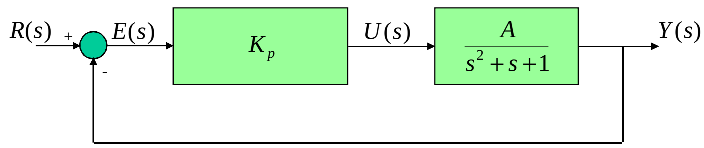
$e_{ss} = 0 \Leftrightarrow K_p G(0) \to \infty$ True when:

- Proportional gain is high
- Plant has a pole at the origin

High gain proportional feedback (needed for good tracking) results in underdamped (or even unstable) transients.

PID Controller

P Controller: Example (P_controller.m)



$$\frac{Y(s)}{R(s)} = \frac{K_p G(s)}{1+K_p G(s)} = \frac{K_p A}{s^2 + s + (1+K_p A)}$$

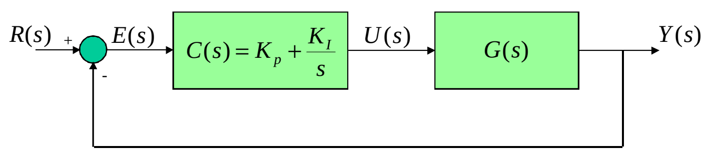
$$\begin{aligned} \omega_n^2 &= 1 + K_p A \\ 2\zeta\omega_n &= 1 \end{aligned} \quad \Rightarrow \zeta = \frac{1}{2\omega_n} = \frac{1}{2\sqrt{1+K_p A}} \xrightarrow[K_p \to \infty]{} 0$$

✓ Underdamped transient for large proportional gain
✓ Steady state error for small proportional gain

PID Controller

PI Controller:

$$\frac{Y(s)}{R(s)} = \frac{C(s)G(s)}{1+C(s)G(s)}, \quad \frac{E(s)}{R(s)} = \frac{1}{1+C(s)G(s)}.$$



$$u(t) = K_p e(t) + K_I \int_0^t e(\tau)d\tau, \qquad U(s) = \left(K_p + \frac{K_I}{s}\right)E(s)$$

Step Reference:

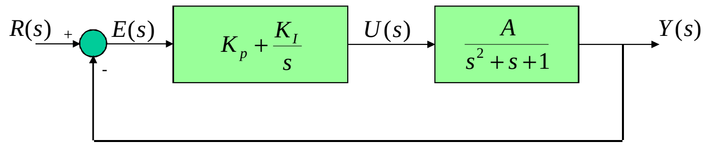
$$R(s) = \frac{1}{s} \Rightarrow e_{ss} = \lim_{s\to 0} sE(s) = \lim_{s\to 0} s \frac{1}{1+\left(K_p + \frac{K_I}{s}\right)G(s)} \frac{1}{s} = \lim_{s\to 0} \frac{1}{1+\left(K_p + \frac{K_I}{s}\right)G(s)} = 0$$

- It does not matter the value of the proportional gain
- Plant does not need to have a pole at the origin. The controller has it!

Integral control achieves perfect steady state reference tracking!!!
Note that this is valid even for $K_p=0$ as long as $K_i \neq 0$

PID Controller

PI Controller: Example (PI_controller.m)



$$\frac{Y(s)}{R(s)} = \frac{\left(K_p + \frac{K_I}{s}\right)G(s)}{1+\left(K_p + \frac{K_I}{s}\right)G(s)} = \frac{(K_p s + K_I)A}{s^3+s^2+(1+K_p A)s + K_I A}$$

DANGER: for large K_i the characteristic equation has roots in the RHP

$$s^3+s^2+(1+K_p A)s + K_I A = 0$$

Analysis by Routh's Criterion

PID Controller

PI Controller: Example (PI_controller.m)

$$s^3+s^2+(1+K_p A)s + K_I A = 0$$

Necessary Conditions: $1+K_p A > 0, K_I A > 0$

This is satisfied because $A > 0, K_p > 0, K_I > 0$

Routh's Conditions:

s^3	1	$1+K_p A$
s^2	1	$K_I A$
s^1	$1+K_p A - K_I A$	
s^0	$K_I A$	

$$1+K_p A - K_I A > 0$$

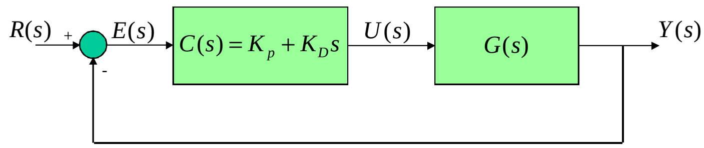
$$\Downarrow$$

$$K_I < K_P + \frac{1}{A}$$

PID Controller

PD Controller:

$$\frac{Y(s)}{R(s)} = \frac{C(s)G(s)}{1+C(s)G(s)},$$
$$\frac{E(s)}{R(s)} = \frac{1}{1+C(s)G(s)}.$$



$$u(t) = K_p e(t) + K_D \frac{de(t)}{dt}, \qquad U(s) = (K_p + K_D s)E(s)$$

Step Reference:

$$R(s) = \frac{1}{s} \Rightarrow e_{ss} = \lim_{s\to 0} sE(s) = \lim_{s\to 0} s \frac{1}{1+(K_p + K_D s)G(s)}\frac{1}{s} = \frac{1}{1+K_p G(0)}$$

$$e_{ss} = 0 \Leftrightarrow K_p G(0) \to \infty$$

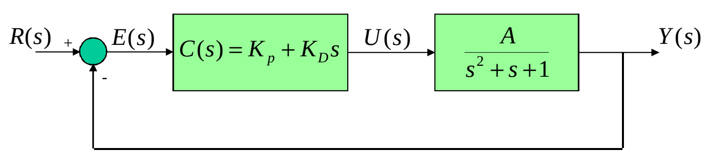
True when:

- Proportional gain is high
- Plant has a pole at the origin

PD controller fixes problems with stability and damping by adding "anticipative" action

PID Controller

PD Controller: Example (PD_controller.m)



$$\frac{Y(s)}{R(s)} = \frac{(K_p + K_D s)G(s)}{1+(K_p + K_D s)G(s)} = \frac{A(K_p + K_D s)}{s^2 + (1+K_D A)s + (1+K_p A)}$$

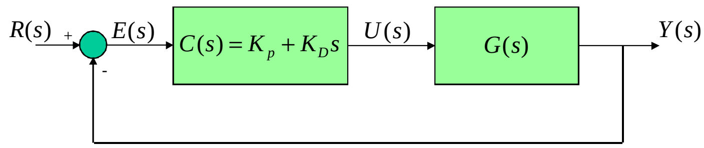
$$\omega_n^2 = 1 + K_p A$$
$$2\zeta\omega_n = 1 + K_D A \qquad \Rightarrow \zeta = \frac{1+K_D A}{2\omega_n} = \frac{1+K_D A}{2\sqrt{1+K_p A}}$$

✓ The damping can be increased now independently of K_p
✓ The steady state error can be minimized by a large K_p

PID Controller

PD Controller:

$$\frac{Y(s)}{R(s)} = \frac{C(s)G(s)}{1+C(s)G(s)},$$
$$\frac{E(s)}{R(s)} = \frac{1}{1+C(s)G(s)}.$$



$$u(t) = K_p e(t) + K_D \frac{de(t)}{dt}, \quad U(s) = (K_p + K_D s)E(s)$$

NOTE: cannot apply pure differentiation.
In practice,

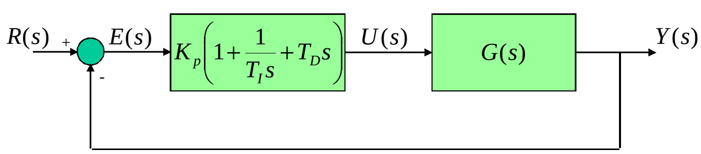
$$K_D s$$

is implemented as

$$\frac{K_D s}{\tau_D s + 1}$$

PID Controller

PID: Proportional – Integral – Derivative



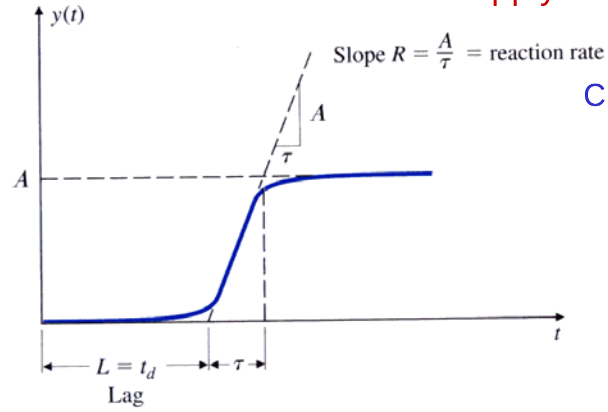
$$u(t) = K_p\left[e(t) + \frac{1}{T_I}\int_0^d e(\tau)d\tau + T_D\frac{de(t)}{dt}\right] \quad K_I = \frac{K_p}{T_I}, K_D = K_p T_D$$

$$\frac{U(s)}{E(s)} = K_p\left(1 + \frac{1}{T_I s} + T_D s\right)$$

PID Controller: Example (PID_controller.m)

PID Controller: Ziegler-Nichols Tuning

- Empirical method (no proof that it works well but it works well for simple systems)
- Only for stable plants
- You do not need a model to apply the method

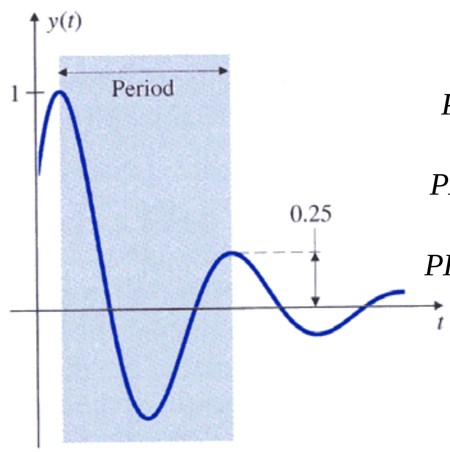


Class of plants:

$$\frac{Y(s)}{U(s)} = \frac{Ke^{-t_d s}}{\tau s + 1}$$

PID Controller: Ziegler-Nichols Tuning

METHOD 1: Based on step response, tuning to decay ratio of 0.25.



Tuning Table:

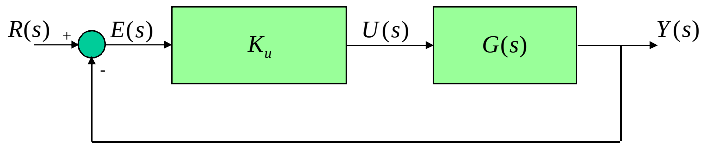
$$P: \quad K_p = \frac{\tau}{t_d}$$

$$PD: \quad K_p = 0.9\frac{\tau}{t_d}, T_I = \frac{t_d}{0.3}$$

$$PID: \quad K_p = 1.2\frac{\tau}{t_d}, T_I = 2t_d, T_D = 0.5t_d$$

PID Controller: Ziegler-Nichols Tuning

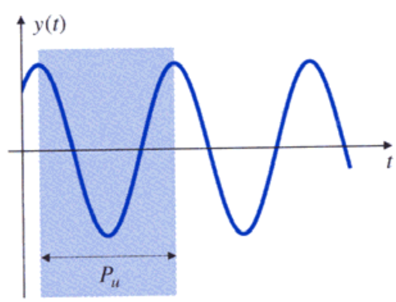
METHOD 2: Based on limit of stability, ultimate sensitivity method.



- Increase the constant gain K_u until the response becomes purely oscillatory (no decay – marginally stable – pure imaginary poles)
- Measure the period of oscillation P_u

PID Controller: Ziegler-Nichols Tuning

METHOD 2: Based on limit of stability, ultimate sensitivity method.



Tuning Table:

$$P: \quad K_p = 0.5K_u$$

$$PD: \quad K_p = 0.45K_u, T_I = \frac{P_u}{1.2}$$

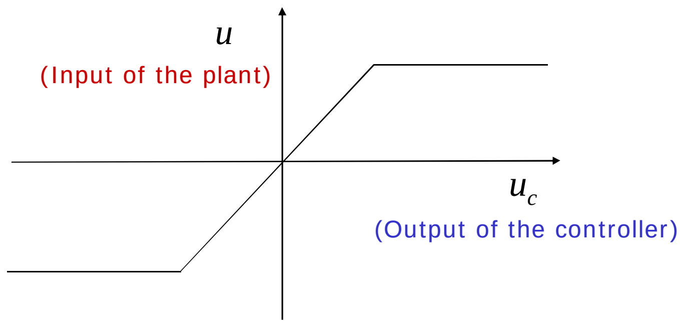
$$PID: \quad K_p = 0.6K_u, T_I = \frac{P_u}{2}, T_D = \frac{P_u}{8}$$

The Tuning Tables are the same if you make:

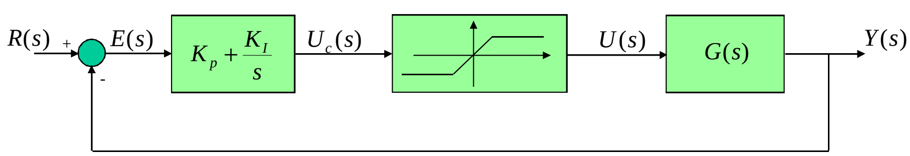
$$K_u = 2\frac{\tau}{t_d}, P_u = 4t_d$$

PID Controller: Ziegler-Nichols Tuning

Actuator Saturates:
- valve (fully open)
- aircraft rudder (fully deflected)



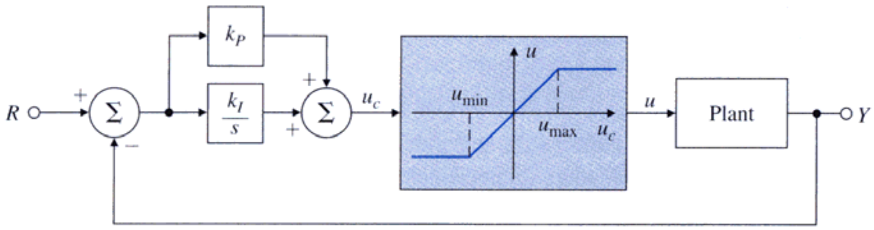
PID Controller: Ziegler-Nichols Tuning



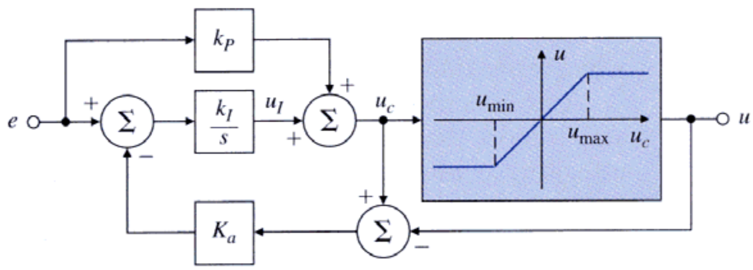
What happens?
- large step input in r
- large e
- large $u_c \rightarrow u$ saturates
- eventually e becomes small
- u_c still large because the integrator is “charged”
- u still at maximum
- y overshoots for a long time

PID Controller: Ziegler-Nichols Tuning

Plant without Anti-Windup:

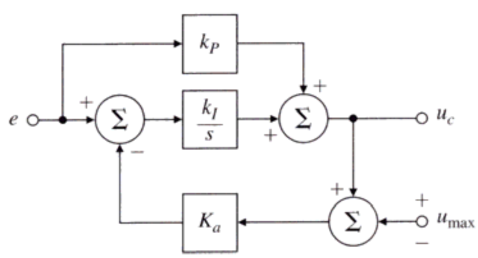


Plant with Anti-Windup:



PID Controller: Ziegler-Nichols Tuning

In saturation, the plant behaves as:



$$e \longrightarrow \boxed{\frac{k_P s + k_I}{s + K_a k_I}} \longrightarrow u_c$$

For large K_a, this is a system with very low gain and very fast decay rate, i.e., the integration is turned off.

Saturation/Antiwindup: Example (Antiwindup_sim.mdl)