Root Locus
II

Outline

Root Locus Design Method

- Root locus (review)
- Dynamic Compensation
  - Lead compensation
  - Lag compensation
  - Notch compensation
- Time Delay (In AnaDig 3)
  - Padé approximation
  - Direct application
- The Discrete Root Locus

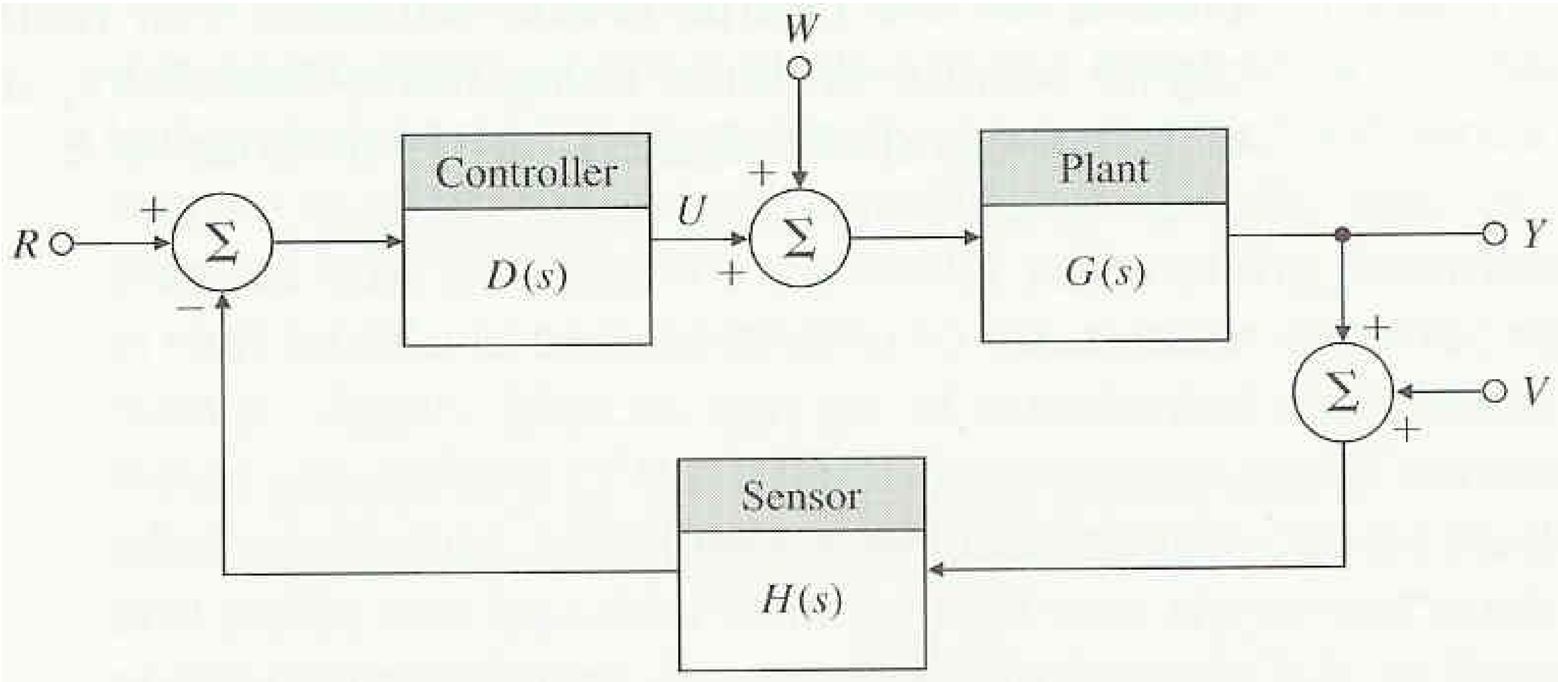
Root Locus

Closed loop transfer function

$$\frac{Y(s)}{R(s)} = T(s) = \frac{D(s)G(s)}{1 + D(s)G(s)H(s)}$$

Characteristic equation, roots are poles in *T(s)*

$$1 + D(s)G(s)H(s) = 0$$



Root Locus

Definition 1

- The root locus is the values of *s* for which $1+KL(s)=0$ is satisfied as *K* varies from 0 to infinity (pos.).

Definition 2

- The root locus of $L(s)$ is the points in the s-plane where the phase of $L(s)$ is 180°.
  - The angle to the test point from zero number *i* is ψ_i.
  - The angle to the test point from pole number *i* is ϕ_i.
  - Therefore, $\sum \psi_i - \sum \phi_i = 180° + 360°(l-1)$

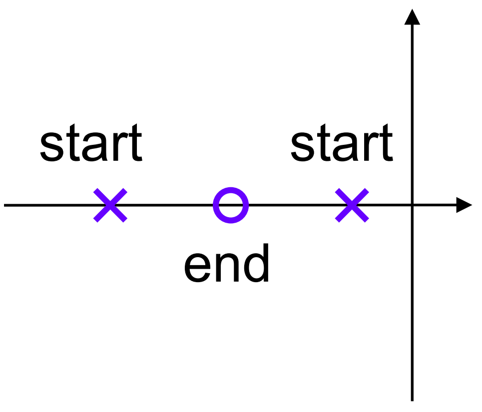
In def. 2, notice, $L(s) = -1/K\,, \quad \angle(-1/K) = 180°$

Root Locus

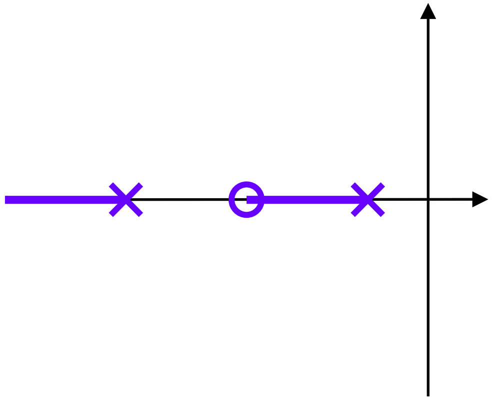
n poles
m zeroes

Rules for sketching a root locus

- Rule 1: The *n* branches of the locus start at the poles *L(s)* and *m* of these branches end on the zeros of *L(s)*.

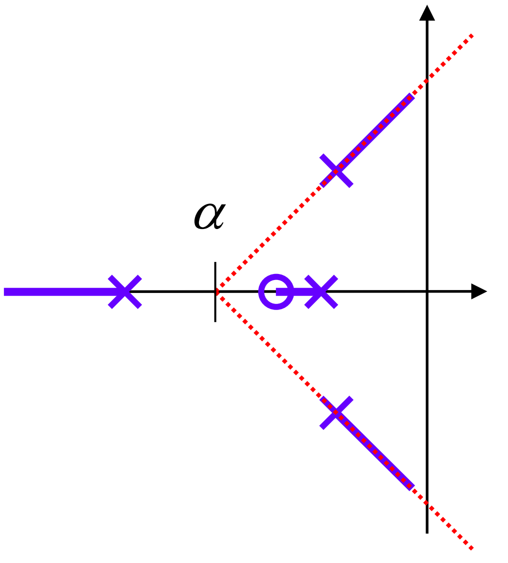


- Rule 2: The loci on the real axis are to the left of an odd number of (real axis) poles plus zeros.



Root Locus

- Rule 3: For large *s*, the loci are asymptotic to lines at angles ϕ_l radiating out from the center point $s = \alpha$.



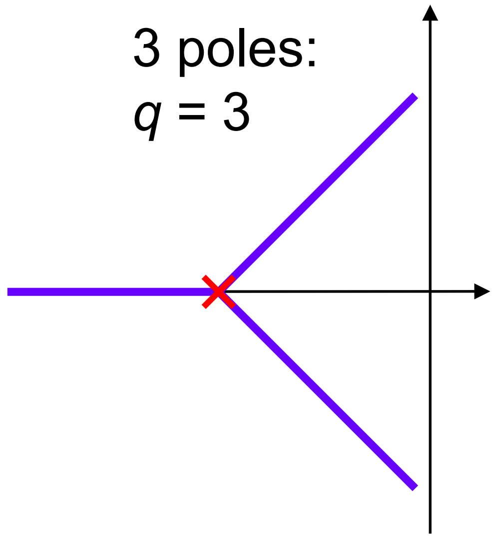
$$\phi_l = \frac{180° + 360°(l-1)}{n-m}, \quad l = 1, 2, \ldots, n-m$$

$$\alpha = \frac{\sum p_i - \sum z_i}{n-m}$$

ϕ_1 = 60 deg.
ϕ_2 = 180 deg.
ϕ_3 = 300 deg.

Root Locus

- Rule 4: The angles of departure (arrival) of a branch of the locus from a pole (zero) of multiplicity q.



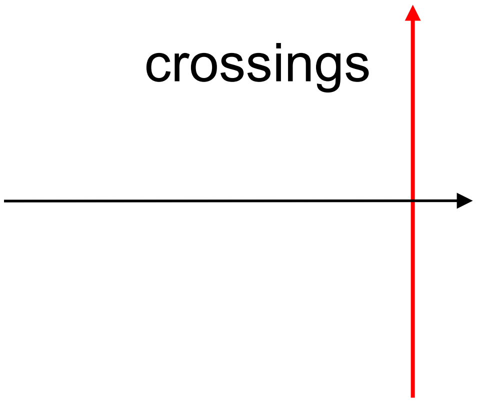
$$q\phi_{l,dep} = \sum \psi_i - \sum_{i \neq l} \phi_i - 180° - 360°(l-1)$$

$$q\psi_{l,dep} = \sum \phi_i - \sum_{i \neq l} \psi_i + 180° + 360°(l-1)$$

$\phi_{1,dep}$ = 60 deg.
$\phi_{2,dep}$ = 180 deg.
$\phi_{3,dep}$ = 300 deg.

Root Locus

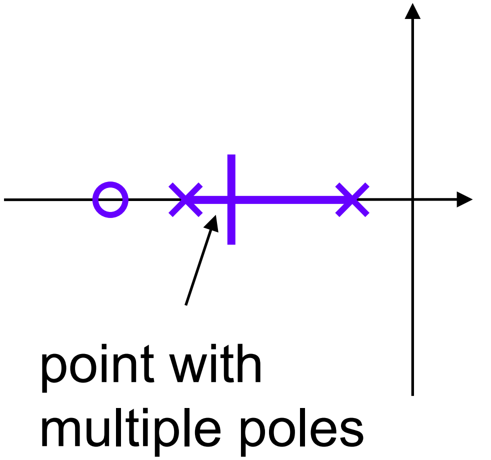
- Rule 5: The locus crosses the $j\omega$ axis at points found by Routh's criterion.



- Rule 6: Identification of multiple roots on the locus.

(1) $$\left(b\frac{da}{ds} - a\frac{db}{ds} \right) = 0$$

(2) $$\frac{180^\circ + 360^\circ(l-1)}{q}$$



Dynamic Compensation

Some facts

- We are able to determine the roots of the characteristic equation (closed loop poles) for a varying parameter K.
- The location of the roots determine the dynamic characteristics (performance) of the closed loop system.
- It might not be possible to achieve the desired performance with $D(s) = K$.

Controller design using root locus

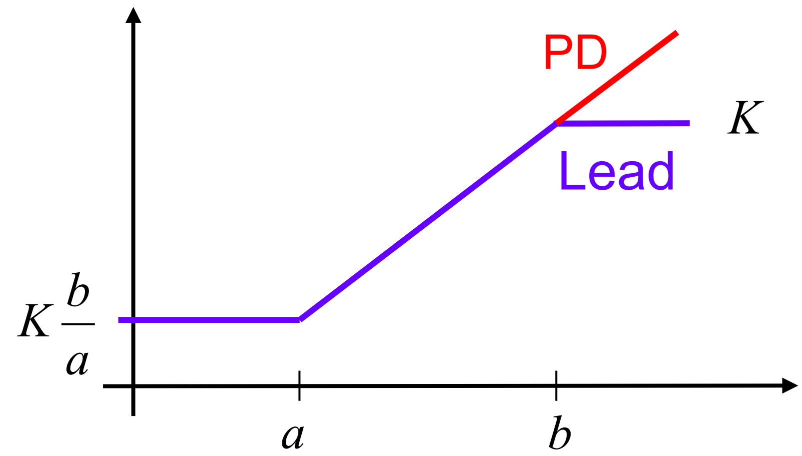
- Lead compensation (similar to PD control)
  - overshoot, rise time requirements
- Lag compensation (similar to PI control)
  - steady state requirements
- Notch compensation

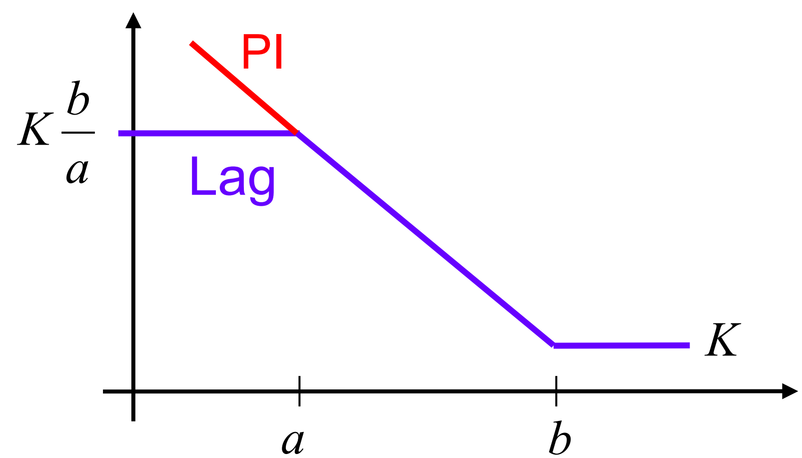
Dynamic Compensation

$$D(s) = K\frac{s+a}{s+b} = K\frac{b}{a}\frac{(s/a+1)}{(s/b+1)}$$

Lead compensation $\quad a < b$

Lag compensation $\quad a > b$





Lead Compensation

Example system

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

(blue) with requirement

$$\omega_n \cong 1.9$$

(green circle)

P - control

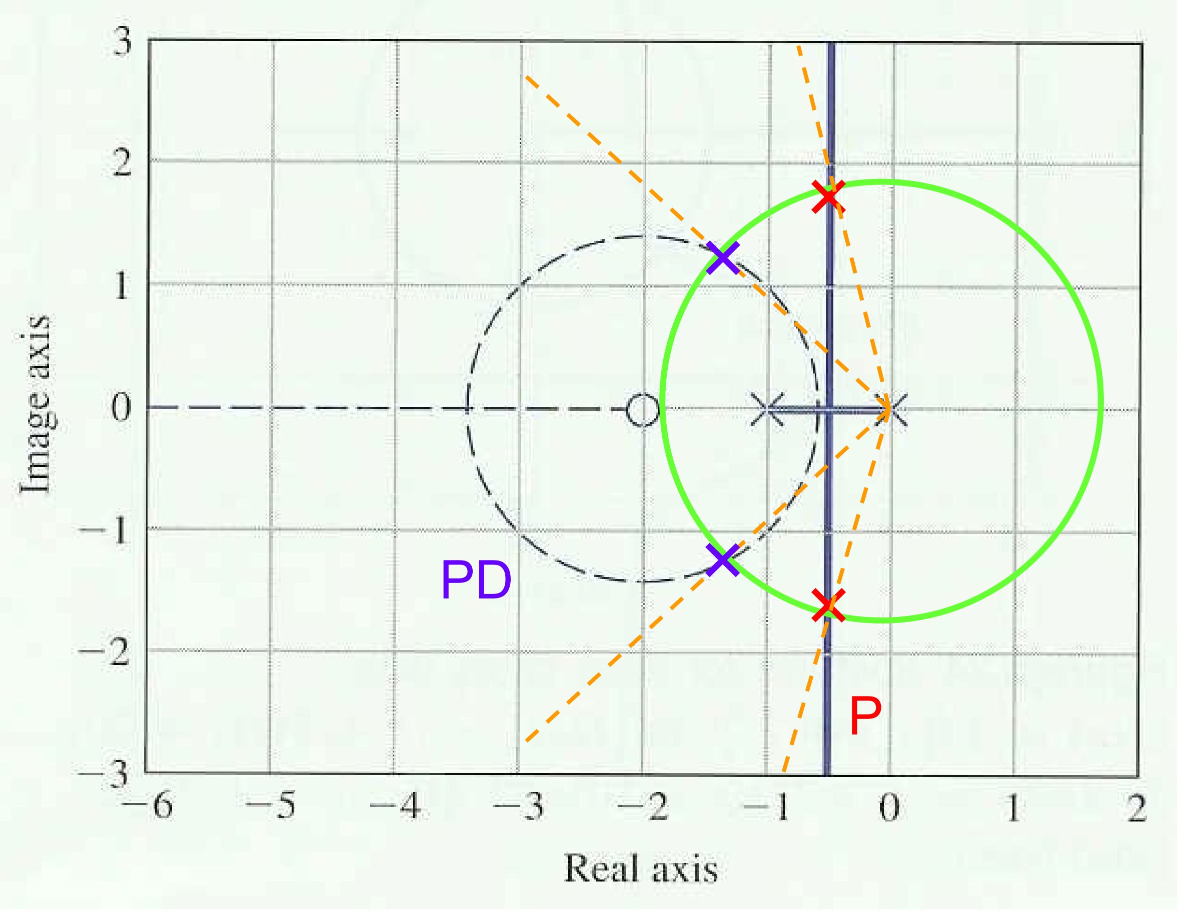
$$D(s) = K$$

(red marks)

PD - control

$$D(s) = K(s+2)$$

(blue marks)

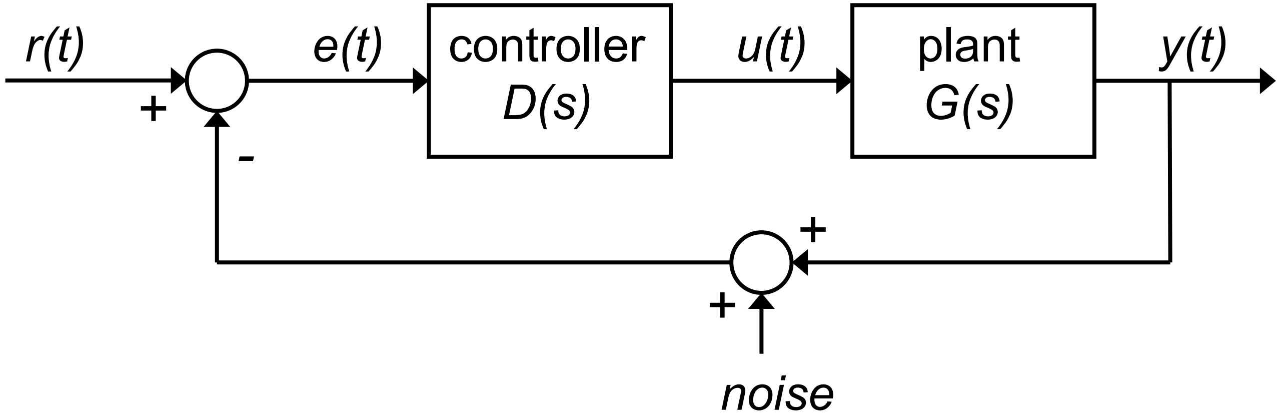


Lower damping ratio with PD !

Lead Compensation

PD control

- Pure differentiation
- Output noise has a great effect on *u(t)*
- Solution: Insert a pole at a higher frequency



Lead Compensation

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

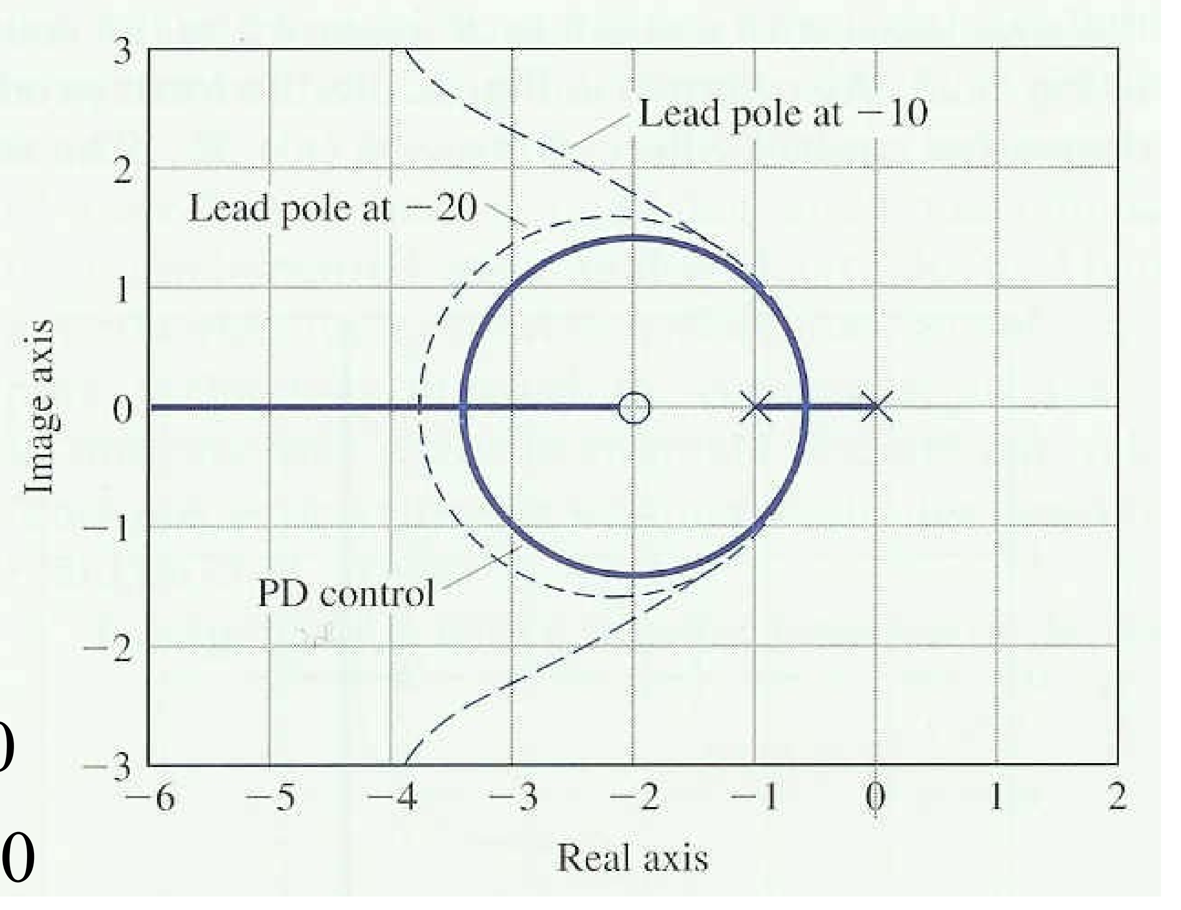
PD - control

$$D(s) = K(s+2)$$

Lead compensation

$$D(s) = K\frac{s+2}{s+p},$$

$$\text{try}: \; p = \begin{cases} 5 \, z = 10 \\ 10 \, z = 20 \end{cases}$$



Almost identical for small gains, for calculations look ex 5.4

Lead Compensation

Selecting *p* and *z*

- Usually, trial and error
- The placement of the zero is determined by dynamic requirements (rise time, overshoot).
- The exact placement of the pole is determined by conflicting interests.
  - suppression of output noise
  - effectiveness of the zero
- In general,
  - the zero z is placed around desired closed loop ω_n
  - the pole p is placed between 5 and 20 times of z

Lead Compensation

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

Requirements

Overshoot : $\zeta \geq 0.5$

Rise time : $\omega_n \approx 7$

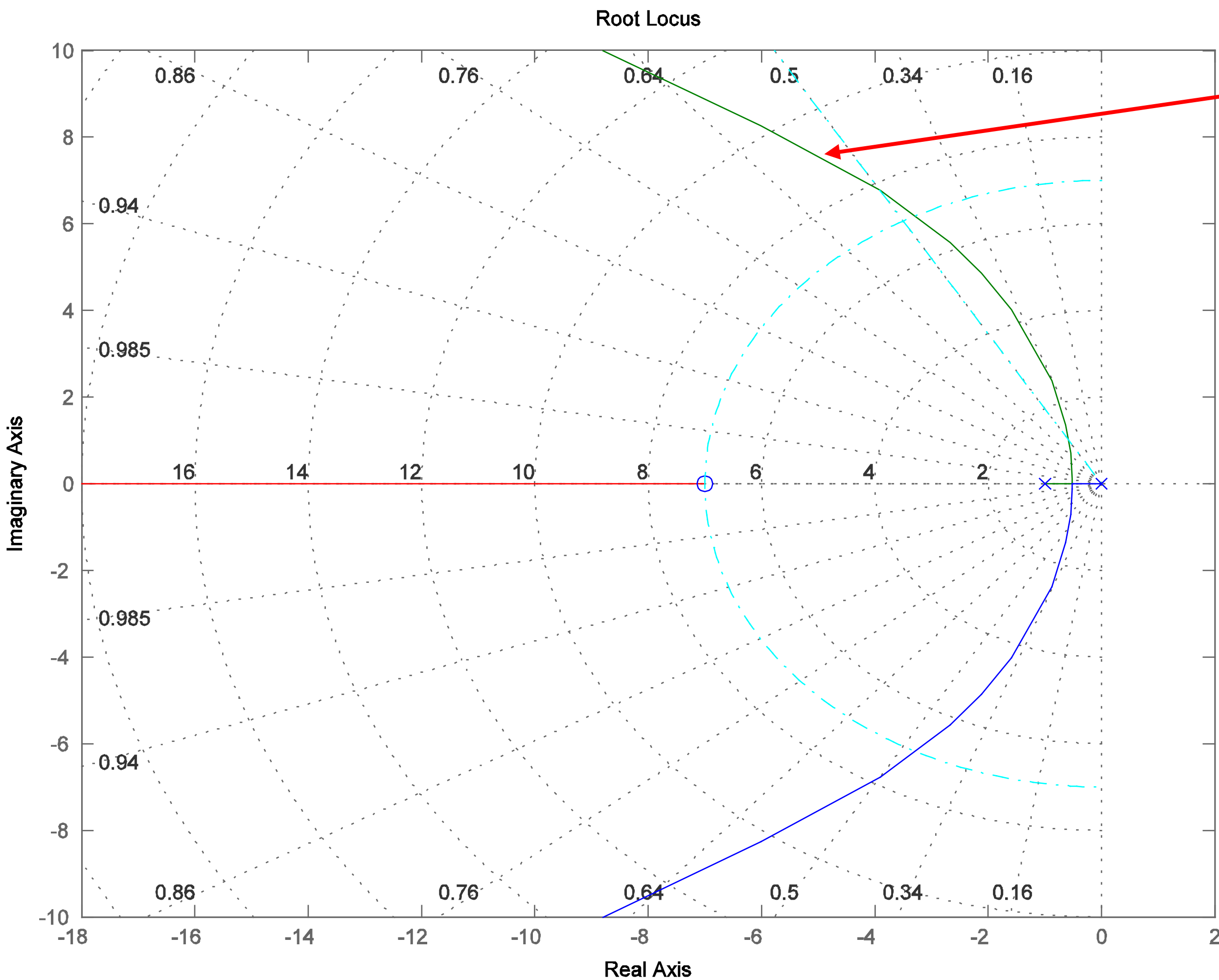
Lead compensation

$$D(s) = K\frac{s+z}{s+p},$$

Design approach (A)

- Initial design $z=\omega_n$, $p=5z$
- Noise: additional requirement, $max(p)=20z$
- Iterations
  - First design: $z=\omega_n$, $p=5z$ (neglecting the add. req.)
  - Second design: $z=\omega_n$, $p=20z$
  - Third design: new z, $p=20z$

Lead Compensation



Possible pole location

First design:

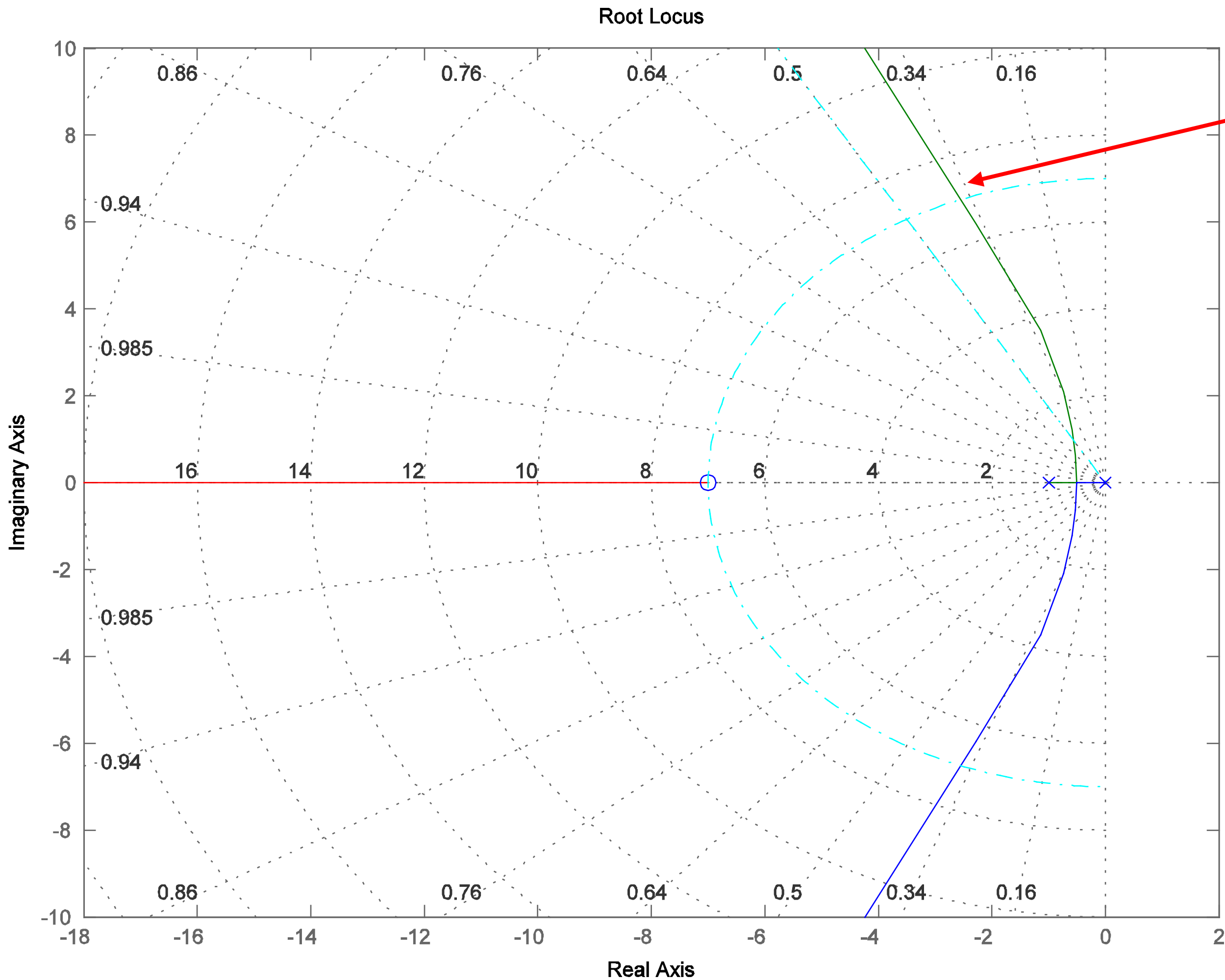
z=wn =7

p=5z=35

Matlab – design 1

```
sysG = tf([1],[1 1 0])
sysD = tf([1 7],[1 5*7])
rlocus(sysD*sysG)
```

Lead Compensation



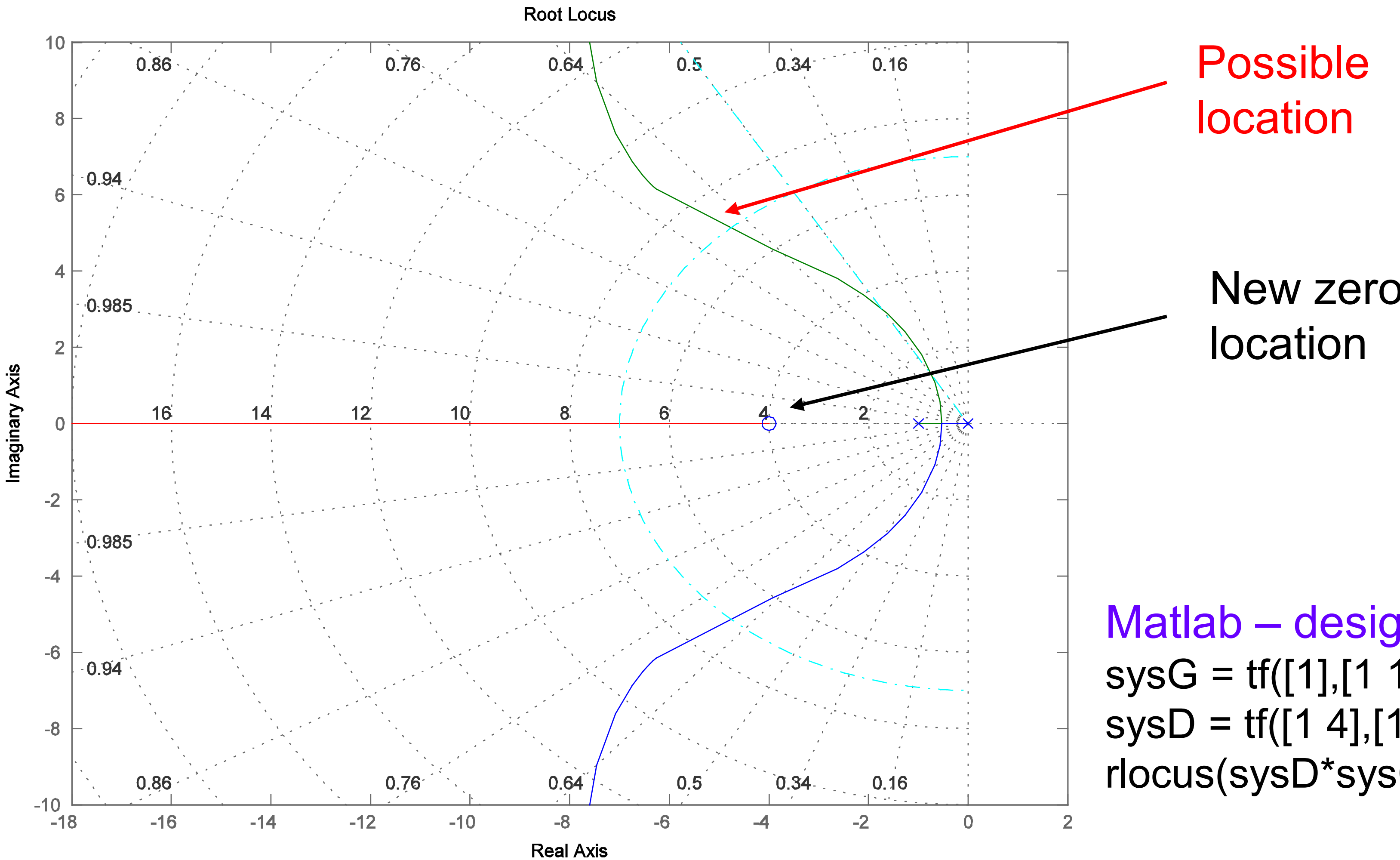
No possible pole location !

Second design:
z=wn, p=20z

Matlab – design 2

sysG = tf([1],[1 1 0])
sysD = tf([1 7],[1 20*7])
rlocus(sysD*sysG)

Lead Compensation



Matlab – design 3

sysG = tf([1],[1 1 0])

sysD = tf([1 4],[1 20*4])

rlocus(sysD*sysG)

Lead Compensation

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

Requirements

Overshoot : $\zeta \geq 0.5$

Rise time : $\omega_n \approx 7$

Lead compensation

$$D(s) = K\frac{s+z}{s+p},$$

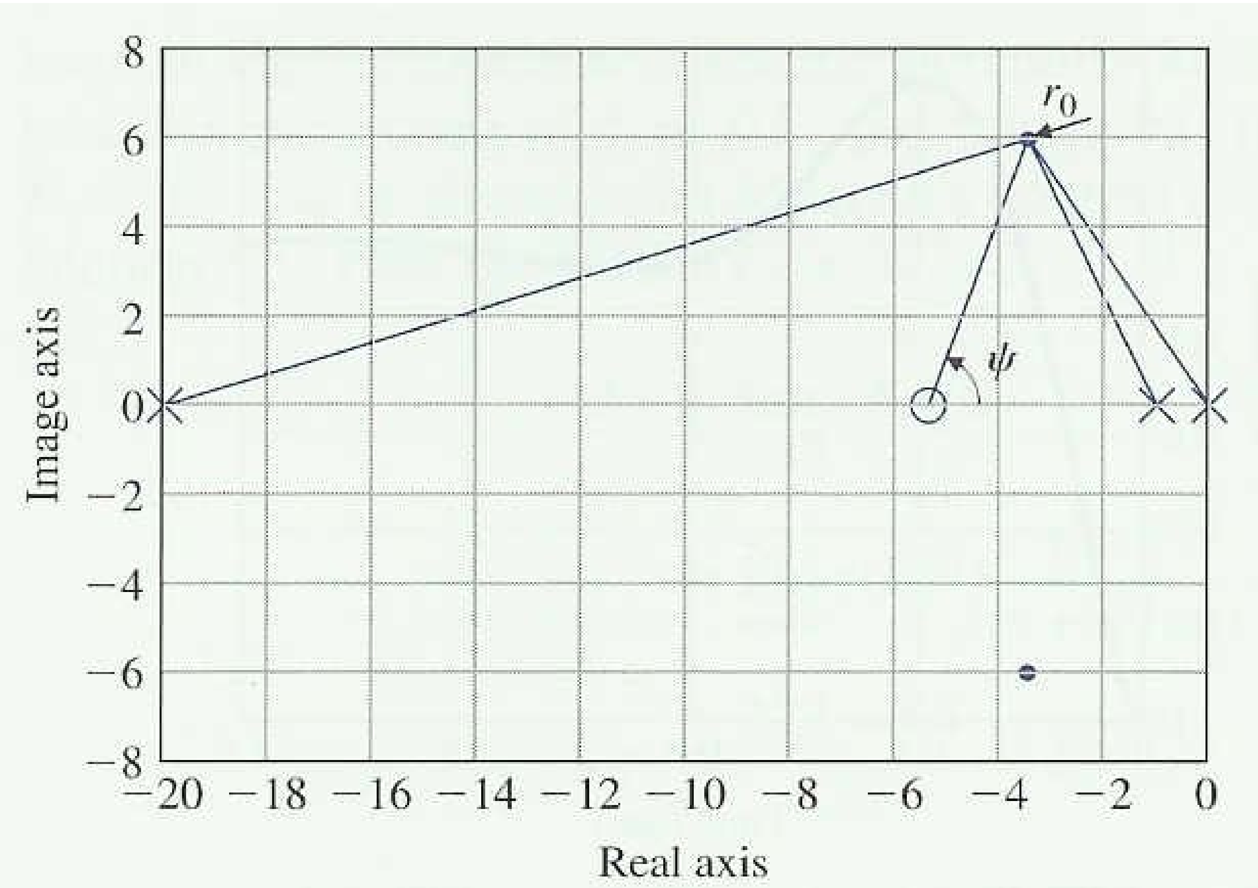
Design approach (B)

- Pole placement from the requirements

  $$r_0 = -3.5 + j3.5\sqrt{3}$$

- Noise: Additional requirement, *max(p)=20.* So, *p=20* to minimize the effect of the pole.

Lead Compensation



All poles are fixed.

r_0 is fixed.

$\Downarrow$

Possible to calculate ψ

as $\angle D(s)G(s) = 180°$

$\Downarrow$

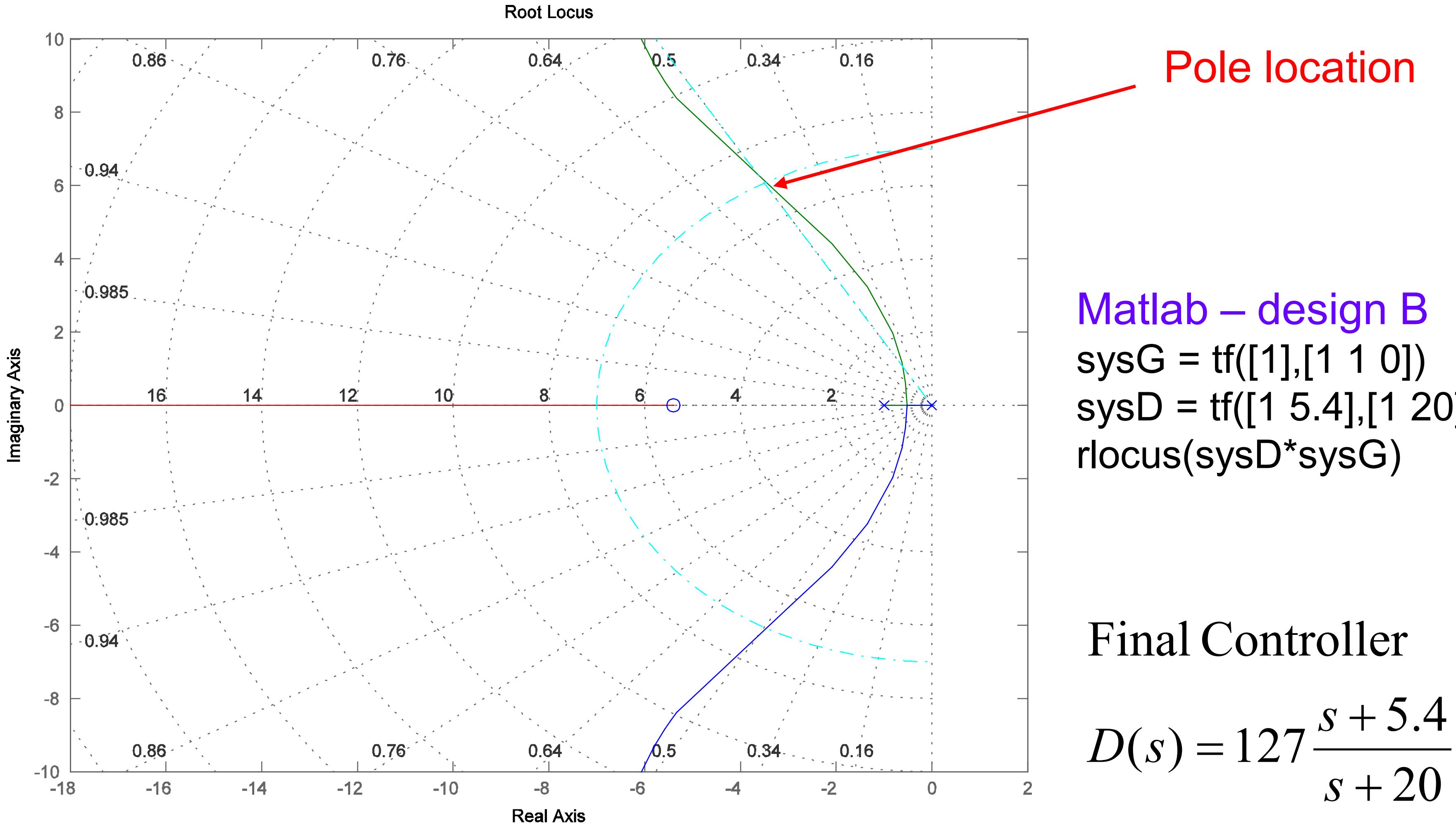
$\psi = 72.6°$

$\Downarrow$

Gives location of z

Result : $z = 5.4$

Lead Compensation



Matlab – design B

sysG = tf([1],[1 1 0])
sysD = tf([1 5.4],[1 20])
rlocus(sysD*sysG)

Final Controller

$$D(s) = 127\frac{s+5.4}{s+20}$$

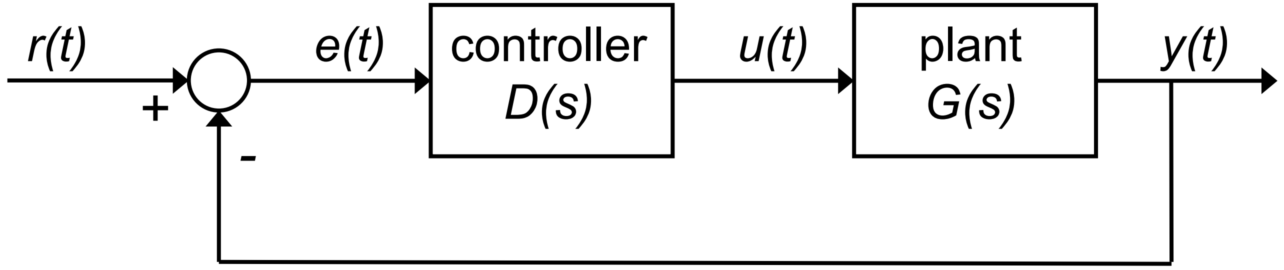
Lead Compensation

Exercise

- Design a lead compensation *D(s)* to the plant *G(s)* so that the dominant poles are located at $s = -2 \pm 2j$

$$D(s) = K\frac{s+z}{s+p} \qquad G(s) = \frac{1}{s^2}$$

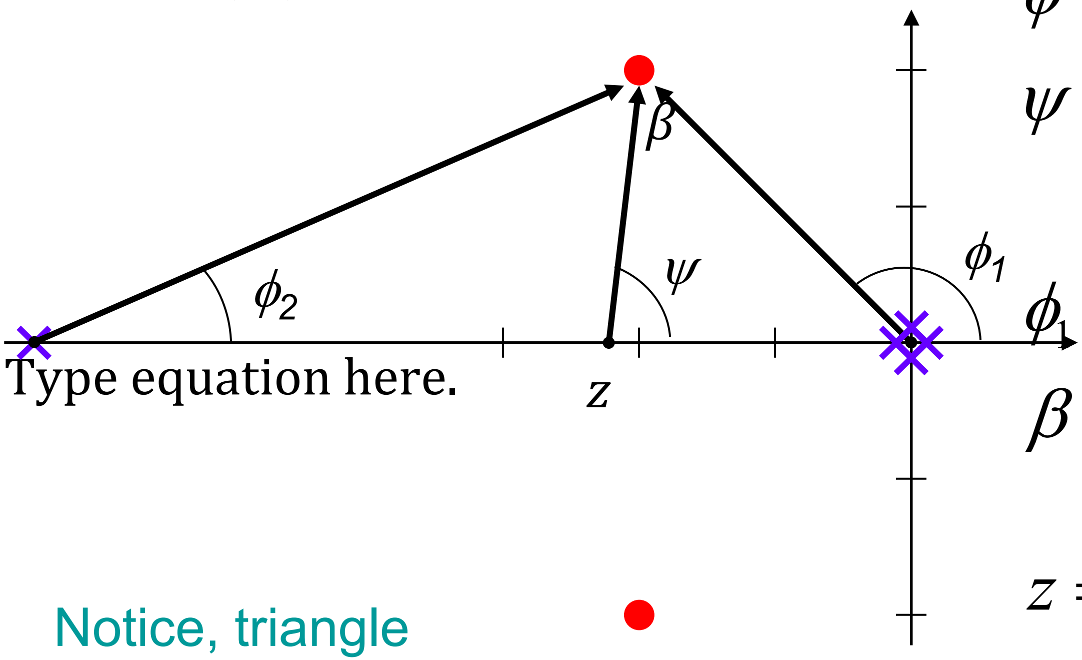
You can use, triangle a/sin(A) = b/sin(B)



Lead Compensation

Answer

- we have three poles 0,0,20 and one zero



Notice, triangle
a/sin(A) = b/sin(B)

Root Locus Condition

$$\angle D(s)G(s) = 180°$$

$$\phi_1 = 135°, \phi_2 \approx 10°$$

$$\psi - 2\phi_1 - \phi_2 = 180° \Rightarrow$$

$$\psi = 180° + 2 \cdot 135°$$
$$+10° = 460° = 100°$$

$$\phi_1 = 135° \Rightarrow$$

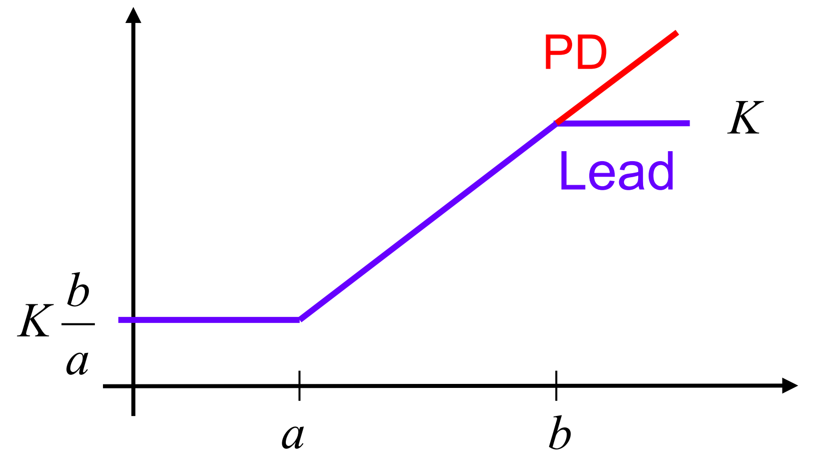
$$\beta = 180° - (180° - 135°) = 35°$$

$$z = -\frac{\sqrt{2^2 + 2^2}}{\sin(100°)} \sin(35°)$$
$$= -1.65$$

BREAK

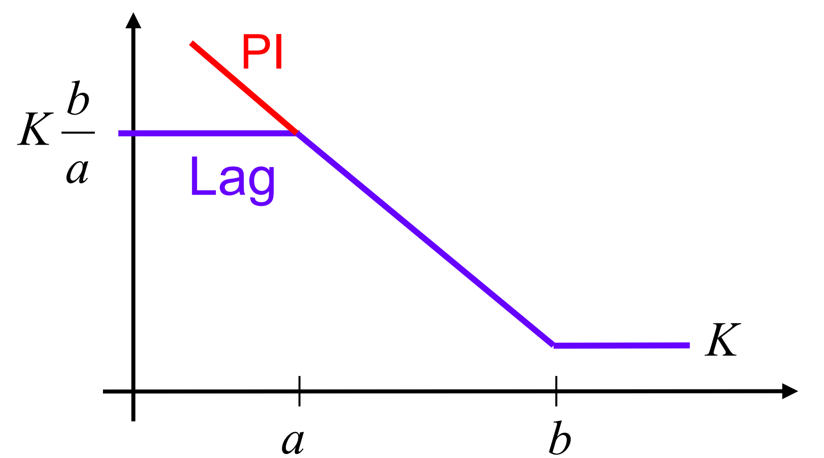
Dynamic Compensation

$$D(s) = K\frac{s+a}{s+b} = K\frac{b}{a}\frac{(s/a+1)}{(s/b+1)}$$



Lead compensation $\quad a < b$

Lag compensation $\quad a > b$



Lag Compensation

Lag compensation

- Steady state characteristics (requirements)
  - Position error constant K_p
  - Velocity error constant K_v
- Let us continue using the example system

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

- and the designed controller (lead compensation)

$$D(s) = 127\frac{s+5.4}{s+20}$$

Lag Compensation

Calculation of error constants

- Suppose we want $K_v \geq 100$
- We need additional gain at low frequencies !

$$D(s)G(s) = 127\frac{s+5.4}{s+20}\frac{1}{s(s+1)}$$

$$K_p = \lim_{s\to 0} D(s)G(s) = \infty$$

$$\text{Step input: } e_{ss} = \frac{1}{1+K_p} = 0$$

$$K_v = \lim_{s\to 0} sD(s)G(s) = 127\frac{5.4}{20} = 34.3$$

$$\text{Ramp input: } e_{ss} = \frac{1}{K_v} = 0.03$$

Lag Compensation

Selecting *p* and *z*

- To minimize the effect on the dominant dynamics *z* and *p* must chosen as low as possible (i.e. at low frequencies)

- To minimize the settling time *z* and *p* must chosen as high as possible (i.e. at high frequencies)

- Thus, the lag pole-zero location must be chosen at as high a frequency as possible without causing any major shifts in the dominant pole locations

Lag Compensation

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

Requirement

Velocity error

constant $K_v \geq 100$

Lag compensation

$$D(s) = K\frac{s+z}{s+p}$$

Low freq. gain

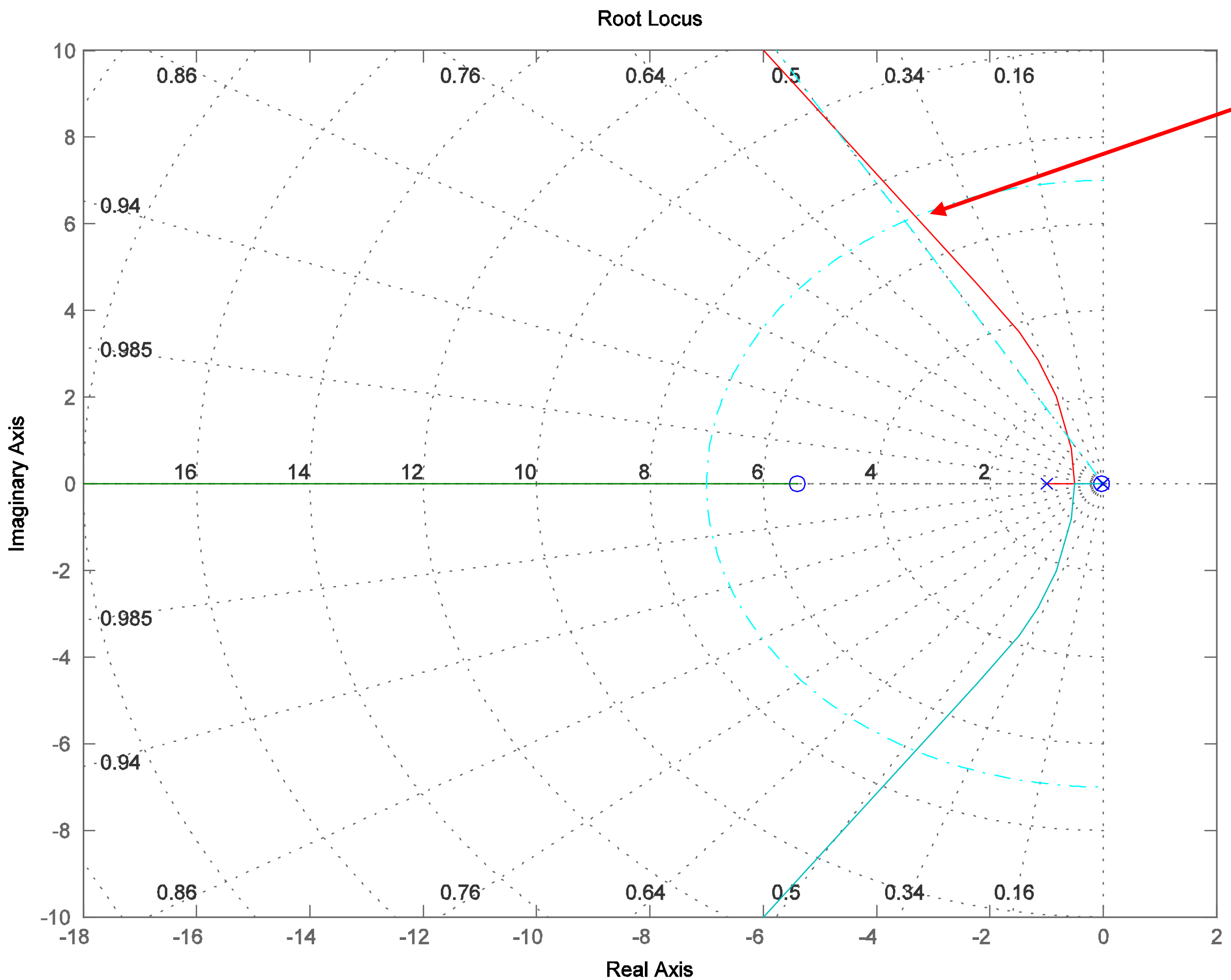
$$D(0) = K\frac{z}{p}$$

Design approach

- Increase the error constant by increasing the low frequency gain
- Choose *z* and *p* somewhat below ω_n

- Example system:
  - $K = 1$ (the proportional part has already been chosen)
  - $z/p = 3$, with $z = 0.03$, and $p = 0.01$.

Lag Compensation



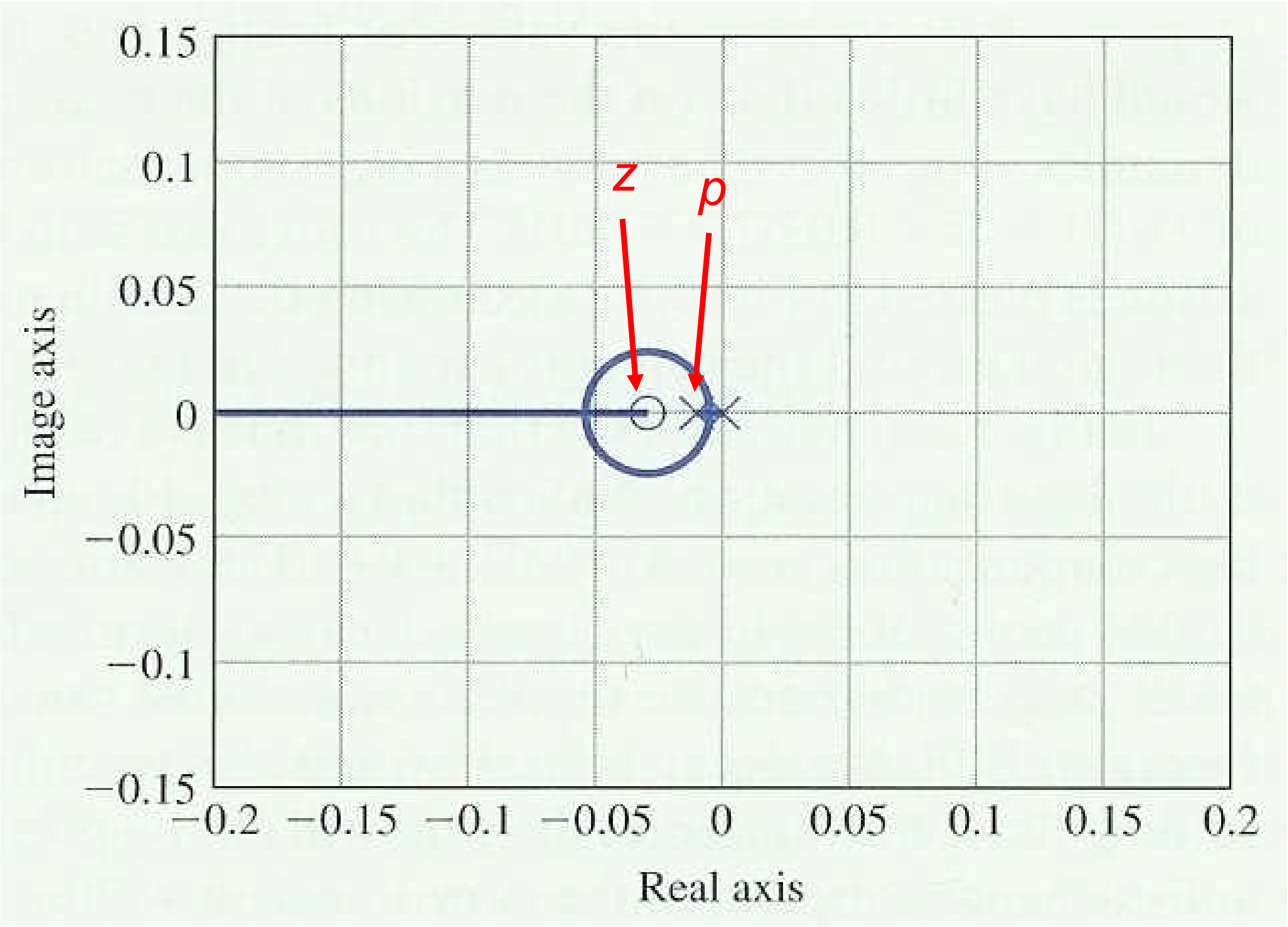
Notice, a slightly different pole location

Lead-lag Controller

$$D(s) =$$

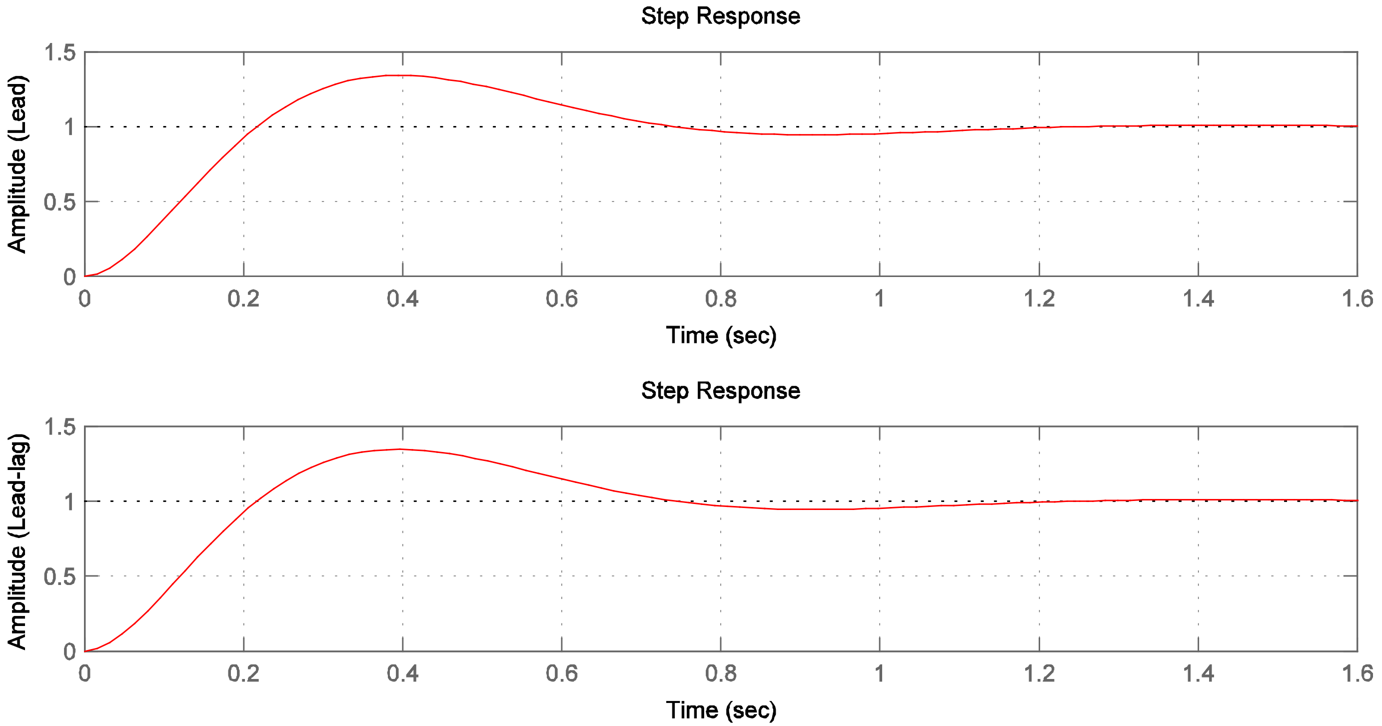
$$127\frac{s+5.4}{s+20}\frac{s+0.03}{s+0.01}$$

Lag Compensation



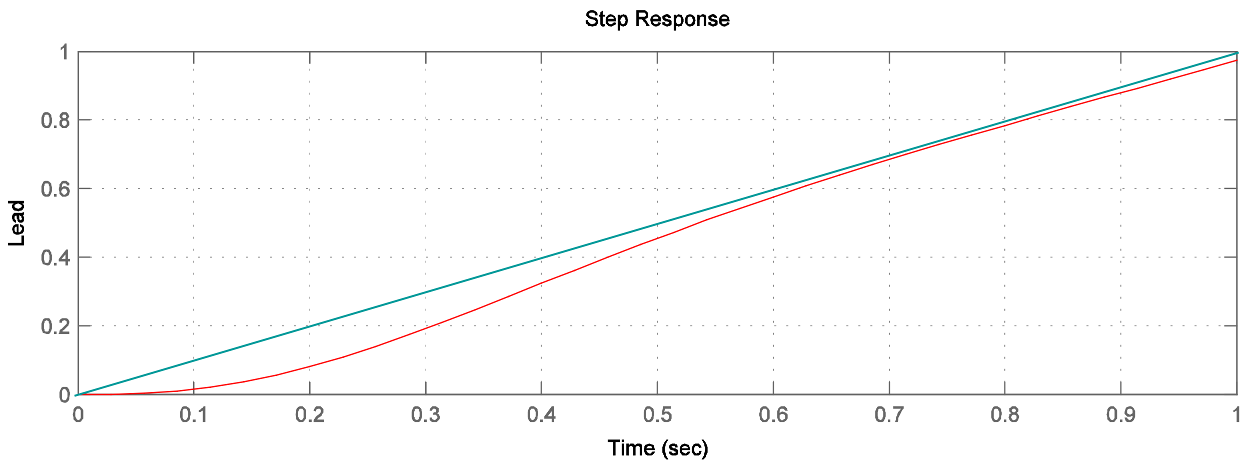
Lag Compensation

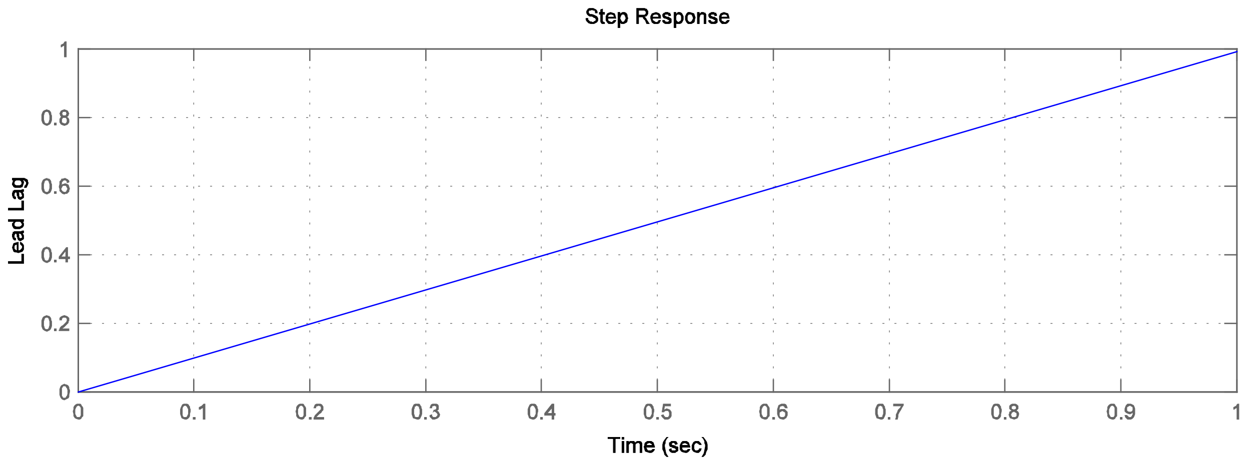
Step



Lag Compensation

Ramp





Notch compensation

Example system

- We have successfully designed a lead-lag controller

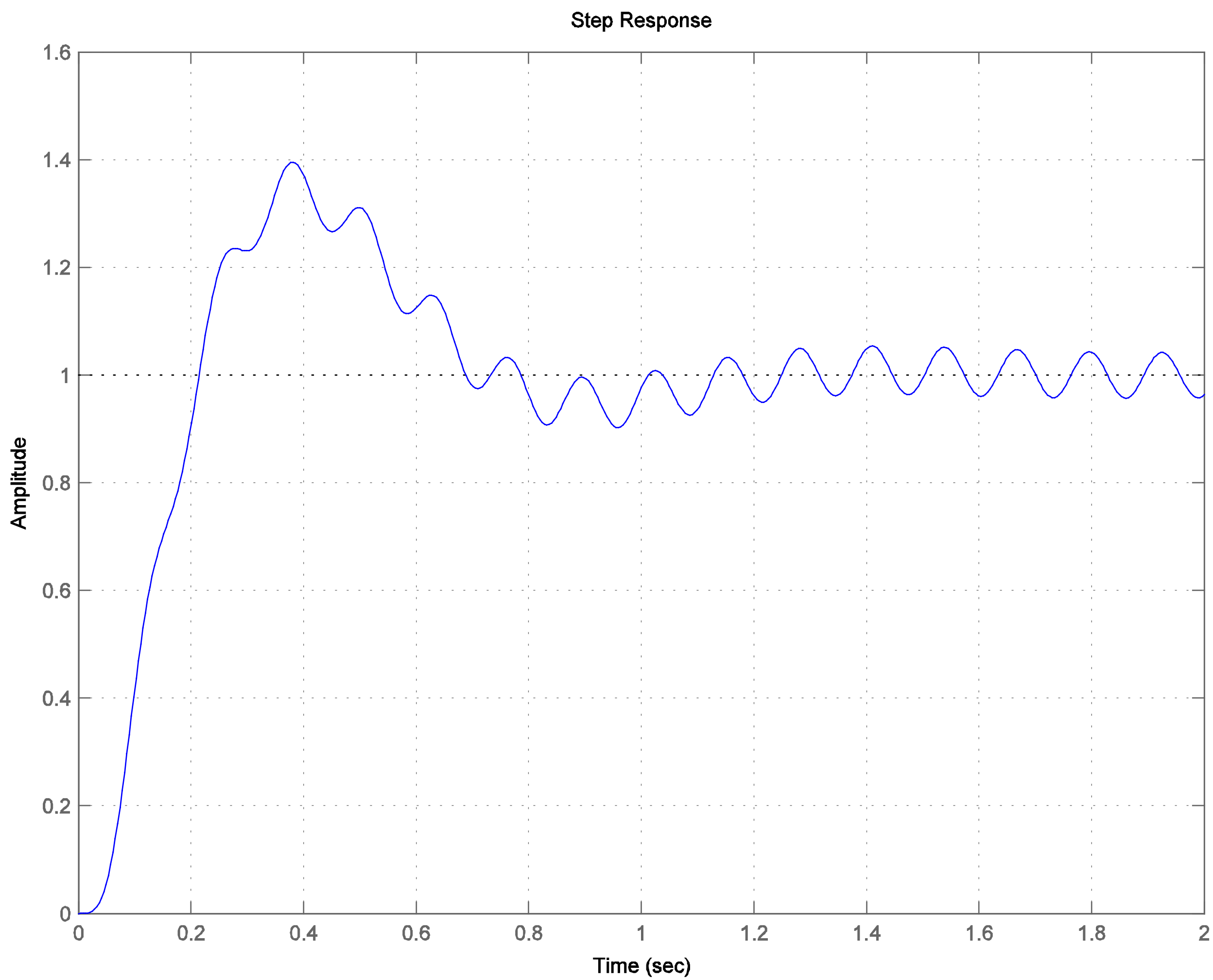
$$D(s) = 127\frac{s+5.4}{s+20}\frac{s+0.03}{s+0.01}$$

- Suppose the real system has a rather undampen oscillation about 50 rad/sec.
- Include this oscillation in the model

$$G(s) = \frac{1}{s(s+1)}\frac{2500}{(s^2+s+2500)}$$

- Can we use the original controller ?

Notch compensation



Step response using the lead-lag controller

Notch compensation

Aim: Remove or dampen the oscillations

Possibilities

- Gain stabilization
  - Reduce the gain at high frequencies
  - Thus, insert poles above the bandwidth but below the oscillation frequency – might not be feasible
- Phase stabilization (*notch compensation*)
  - A zero near the oscillation frequency
  - A zero increases the phase, and $\zeta \approx PM/100$
  - Possible transfer function

$$D_n(s) = \frac{s^2 + 2\zeta_n \omega_0 s + \omega_0^2}{(s + \omega_0)^2}$$

If $\zeta_n < 1$, complex zeros and double pole at ω_0

Notch compensation

$$G(s) = \frac{1}{s(s+1)} \frac{2500}{(s^2 + s + 2500)}$$

$$= \frac{1}{s(s+1)} \frac{\omega_r^2}{(s^2 + 2\zeta_r \omega_r s + \omega_r^2)}$$

$$\omega_r = 50 \quad , \quad \zeta_r = 0.01$$

Notch compensation

$$D_n(s) = \frac{s^2 + 2\zeta_n \omega_0 s + \omega_0^2}{(s + \omega_0)^2}$$

Let us choose

$$\omega_r = \omega_0 \quad , \quad \zeta_r = \zeta_n$$

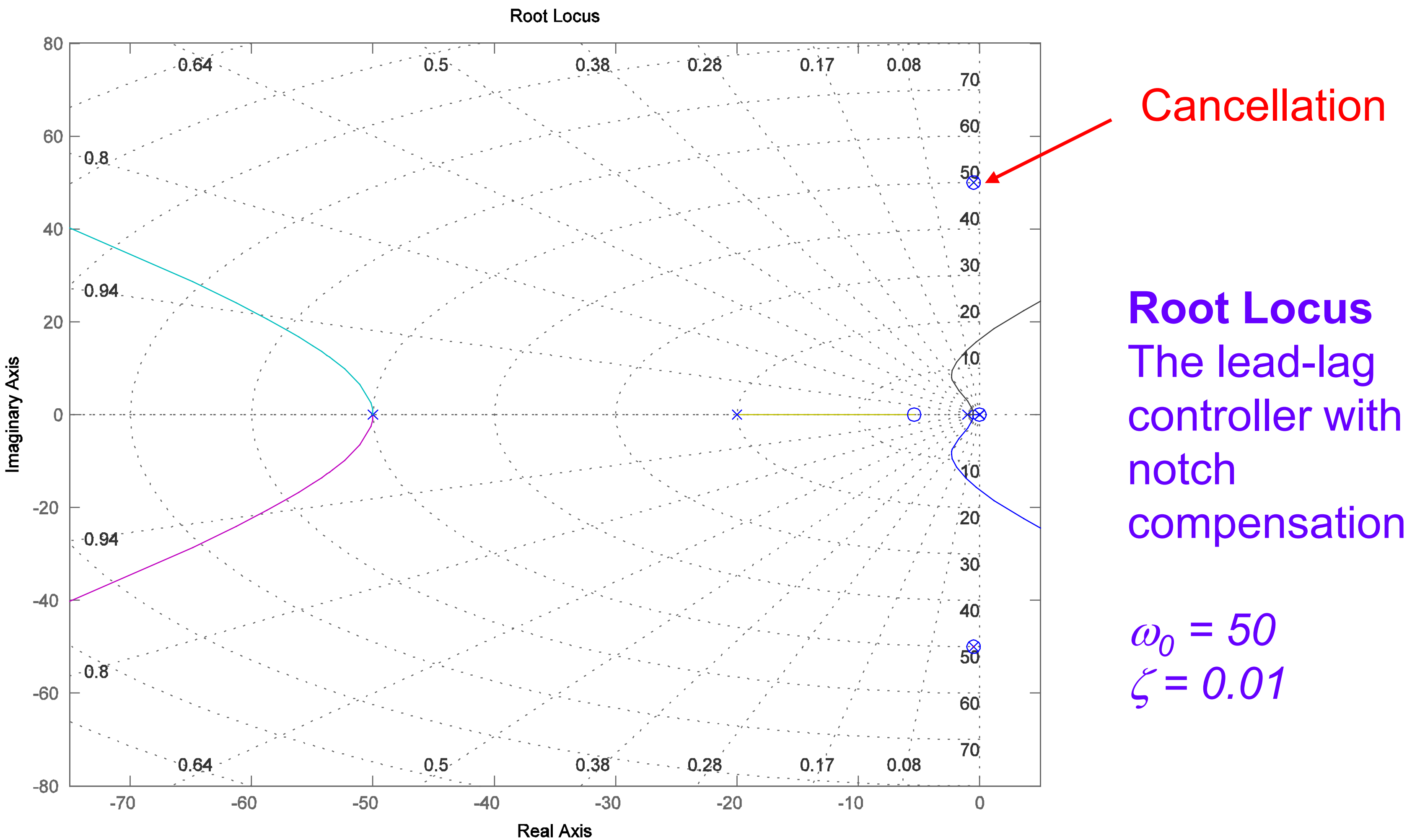
The undamped poles are cancelled

Result

$$D_n(s) G(s)$$

$$= \frac{1}{s(s+1)} \frac{\omega_r^2}{(s + \omega_r)^2}$$

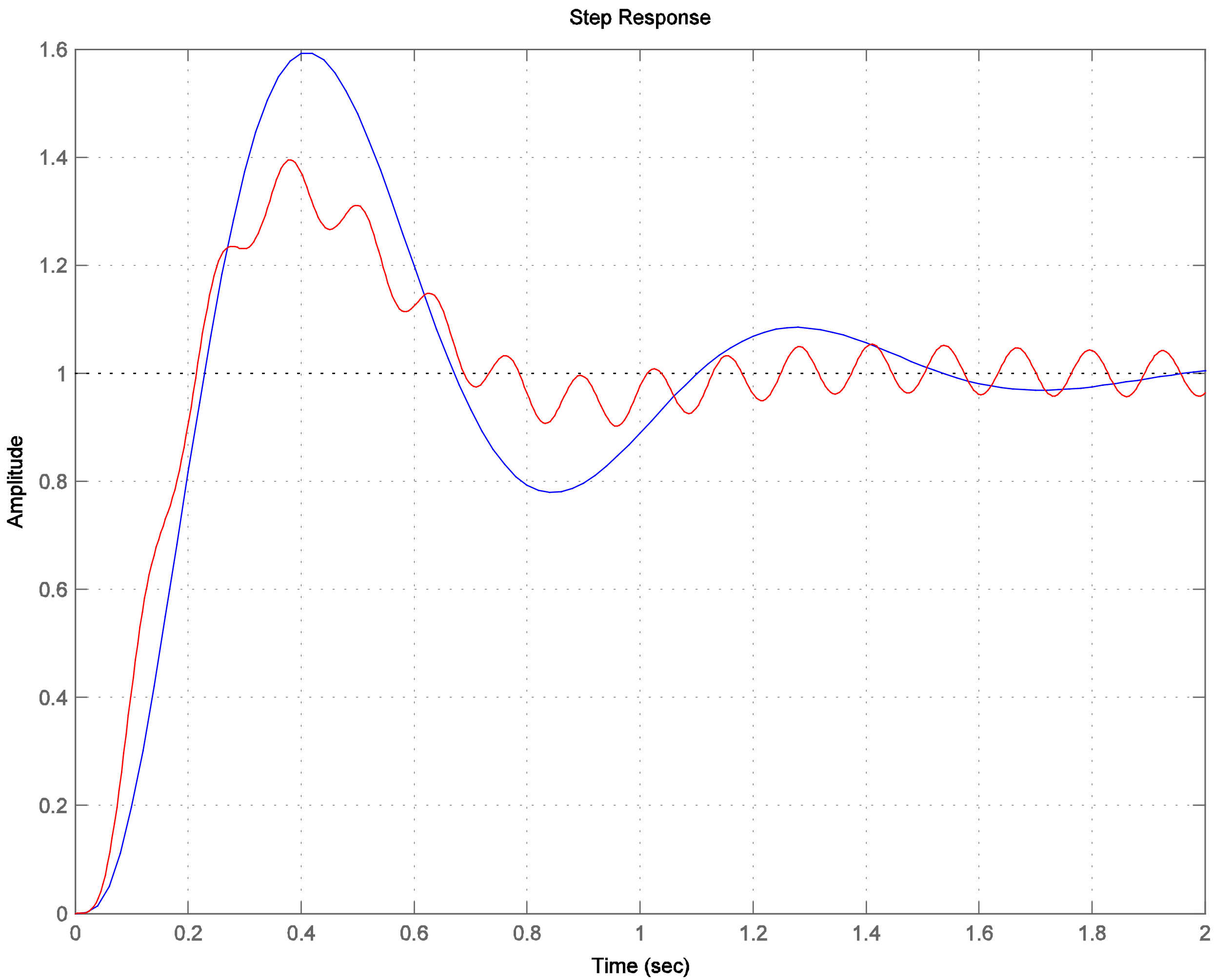
Notch compensation



Root Locus
The lead-lag controller with notch compensation

$\omega_0 = 50$
$\zeta = 0.01$

Notch compensation



Too much overshoot !!

Also, exact cancellation might cause problems due to modelling errors

Thus, we need different parameters

Notch compensation

Modified parameters

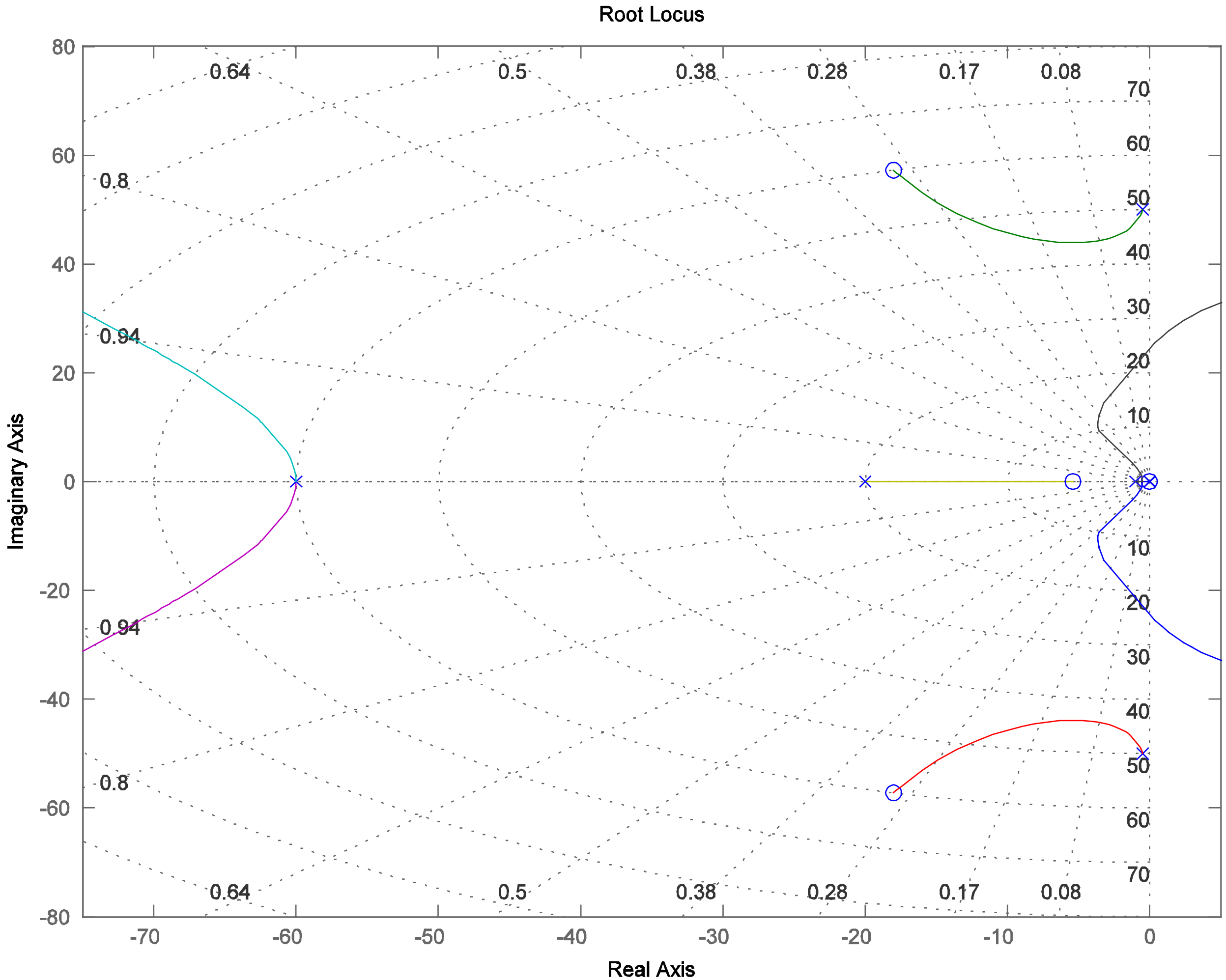
- Notch compensation

$$D_n(s) = \frac{s^2 + 2\zeta\omega_0 s + \omega_0^2}{(s + \omega_0)^2}$$

- Dynamic characteristics
  - Increase ζ_n to obtain less overshoot
  - Make sure that the roots move into the LHP
  - In general, obtain a satisfactory dynamic behavior
- After some trial and error

$$\omega_0 = 60 \quad , \quad \zeta_n = 0.3$$

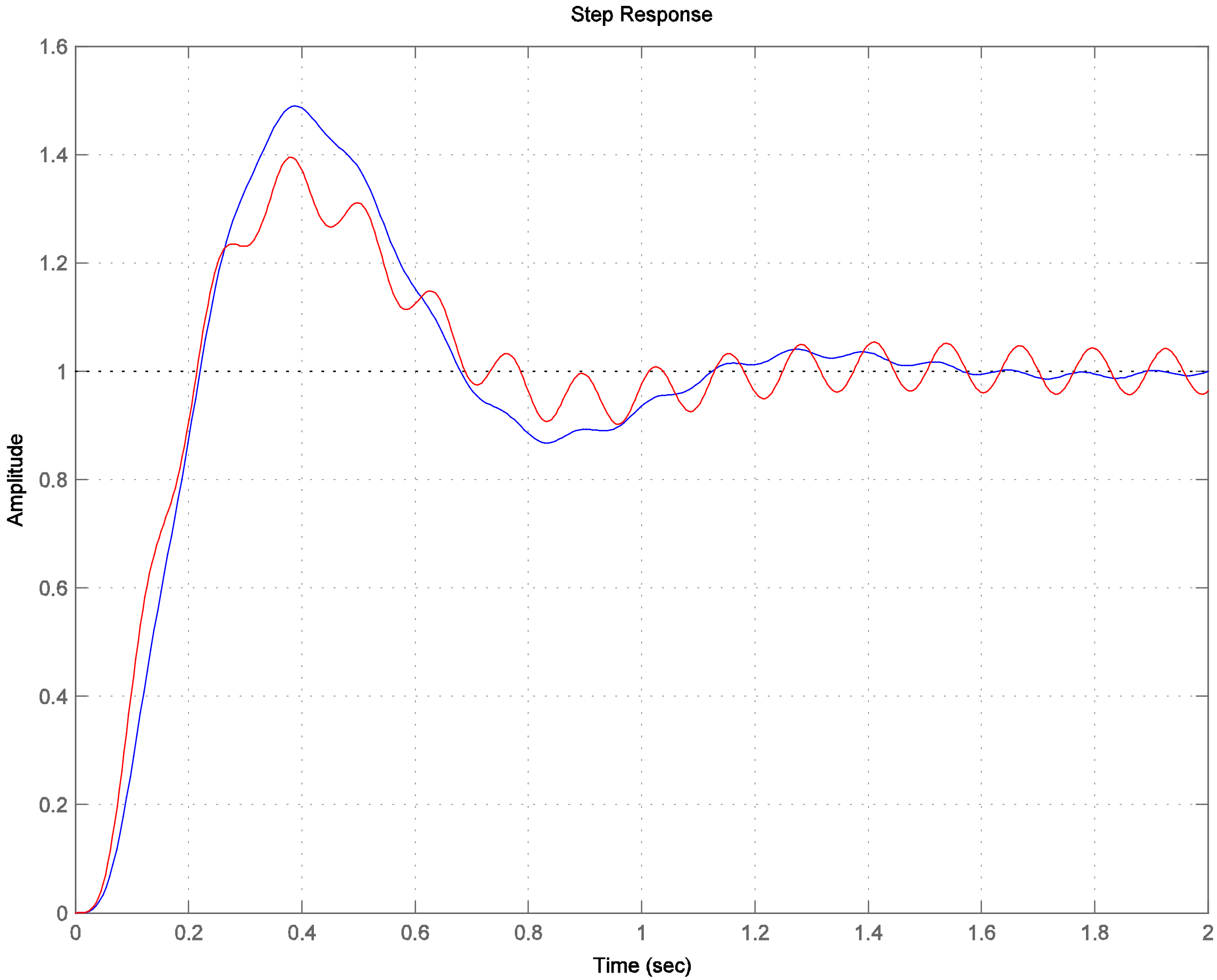
Notch compensation



Root Locus
The lead-lag controller with notch compensation

$\omega_0 = 60$
$\zeta = 0.3$

Notch compensation



The overshoot has been lowered to an acceptable level !

The oscillations have almost been removed !

So, we have reached a final design !

Time Delay

Notice

- Time delay always reduces the stability of a system !
- Important to be able to analyze its effect
- In the s-domain a time delay is given by $e^{-\lambda s}$
- Most applications contain delays (sampled systems)

Root locus analysis

- The original method does only handle polynomials

Solutions

- Approximation (Padé) of $e^{-\lambda s}$
- Modifying the root locus method (direct application)

Time Delay

First approximation

(1,1) Padé approximant

$$e^{-s} \approx \frac{b_0 s + b_1}{a_0 s + 1}$$

McLauren series

$$e^{-s} = 1 - s + \frac{s^2}{2} + \frac{s^3}{3!} + \frac{s^4}{4!} + \ldots$$

$$\frac{b_0 s + b_1}{a_0 s + 1} = b_1 + (b_0 - a_0 b_1) s$$

$$- a_0 (b_0 - a_0 b_1) s^2 + a_0^2 (b_0 - a_0 b_1) s^3$$

$$b_1 = 1$$

$$b_0 - a_0 b_1 = -1$$

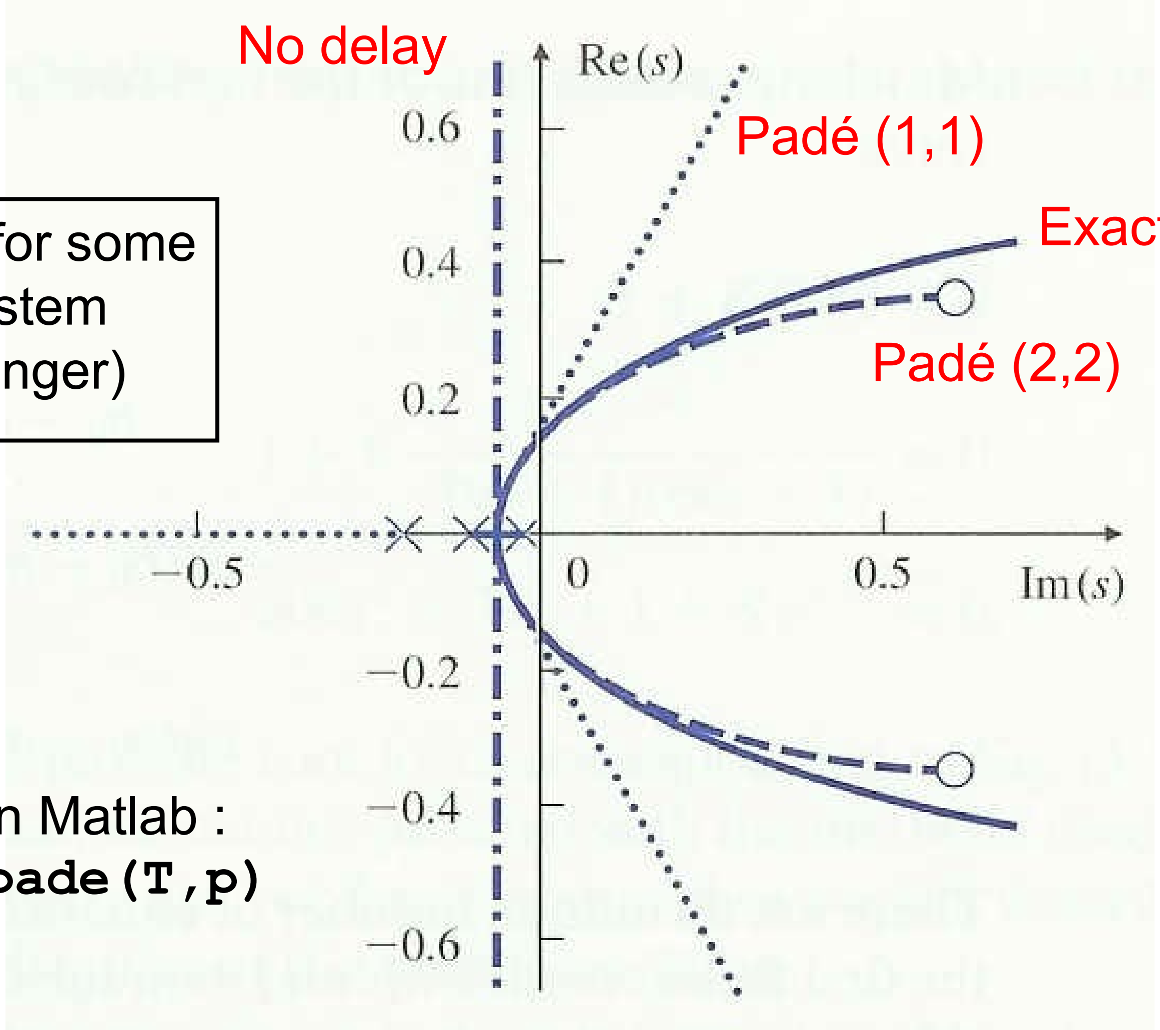
$$- a_0 (b_0 - a_0 b_1) = \frac{1}{2}$$

$$a_0^2 (b_0 - a_0 b_1) = -\frac{1}{6}$$

$\Downarrow$ with $s \equiv T_d s$

$$e^{-T_d s} \cong \frac{1 - (T_d s / 2)}{1 + (T_d s / 2)}$$

Time Delay



Root locus for some example system (heat exchanger)

(p,p) Approx. in Matlab :

```
[num,den]=pade(T,p)
```

Time Delay

Direct approach (exact calculation)

Process

$$G(s) = e^{-T_d s}\, G_0(s)$$

Notice, $s = \sigma + j\omega$

$$\angle e^{-T_d s} = \angle (e^{-T_d \sigma} e^{-jT_d \omega}) = -T_d \omega$$

Modified root locus condition

$$\angle D(s)\,G(s) = \angle \left(D(s)\,G_0(s)\right) + \angle \left(e^{-T_d s}\right) = 180^\circ$$

$$\Downarrow$$

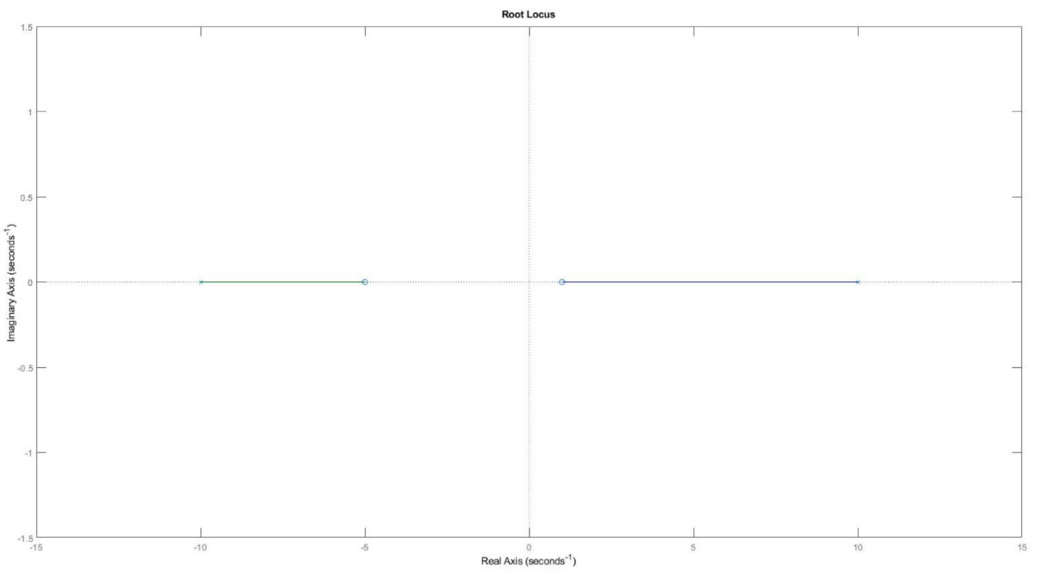
$$\angle D(s)\,G(s) = 180^\circ + T_d \omega$$

However, Matlab does not support this approach...

Unstable systems - example

$$G(s) = \frac{(s-1)(s+5)}{(s+10)(s-10)}$$

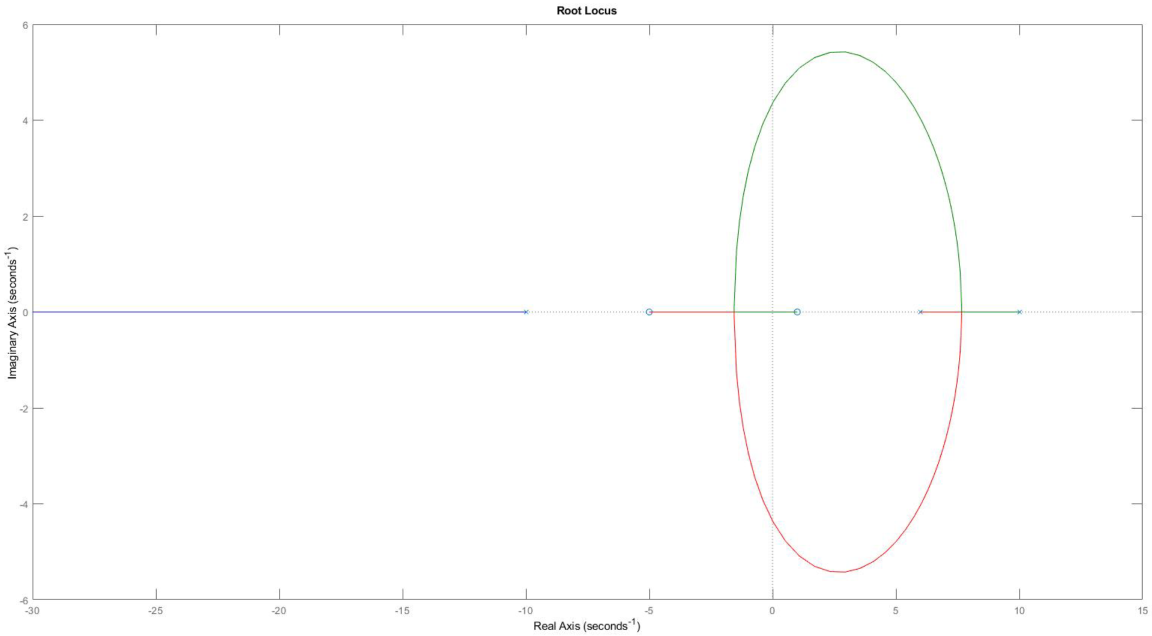
an unstable pole and an unstable zero



How can we stabilize the system

Dirty trick -Insert an unstable pole

$$G(s) = \frac{(s-1)(s+5)}{(s+10)(s-10)(s-6)}$$



The Discrete Root Locus

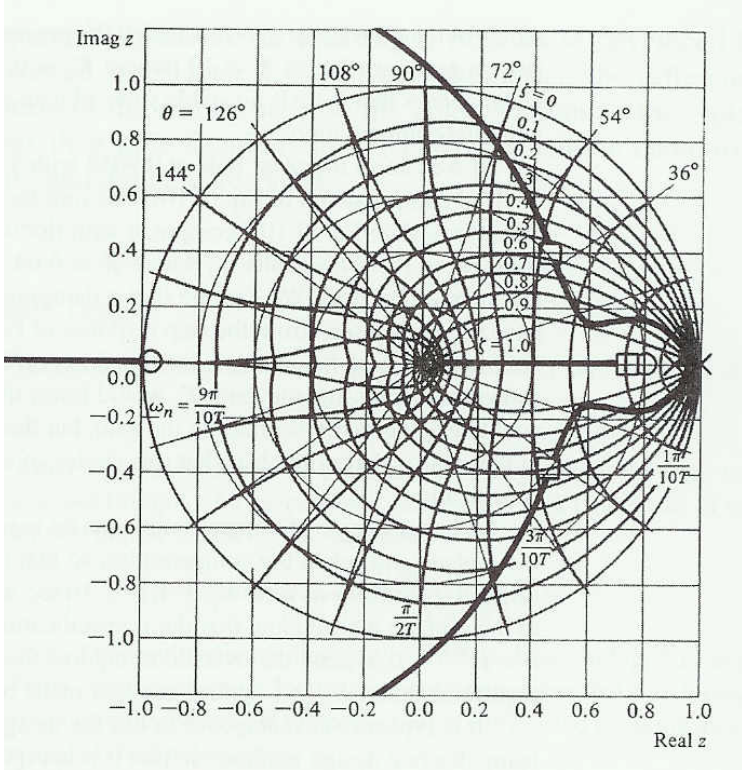
Discrete (z-domain)

- Closed loop transfer function

$$\frac{D(z)G(z)}{1+D(z)G(z)}$$

- Characteristic equation

$$1+D(z)G(z)=0$$



- Thus, same sketching techniques as in the s-domain, the unit circle is the stable region.
- *However, different interpretation !!!*