EE 4323 – Industrial Control Systems

Module 9: Root Locus & Frequency-Domain Design

Dr. James H. Taylor
Department of Electrical Engineering
telephone: 453-5101
e-mail: jtaylor@unb.ca
web page: www.ee.unb.ca/jtaylor/

10 May 2017

Root Locus & Frequency-Domain Design

- Drawing Root Loci
- Pure Gain Compensation
- Rate Feedback Compensation
- Proportional Plus Rate Feedback Design
- Lead Compensation
  - Root Locus Design
  - Frequency-Domain Design
- Proportional/Integral (PI) Compensation
- Lag Compensation
  - Root Locus Design
  - Frequency-Domain Design

References:

1. Norman Nise, *Control Systems Engineering*, 4^{th} Edition, John Wiley & Sons, 2004.
2. Katsuhiko Ogata, *Modern Control Engineering*, Fourth Edition, Prentice Hall, 2002.
3. ...and many other basic controls texts.

Control System Design Problems

- The goal of control system design is to take the "givens", usually a plant, actuators and sensors, and devise a control strategy that will achieve acceptable performance (meet specifications)
- Control problems range from the very easy to the very difficult
- The two most common control objectives are (a) to improve the transient response (e.g., achieve a desired rise time and percent overshoot) and (b) to improve steady-state operation (e.g., meet a steady-state error specification)
- There are several motivations for good control in industrial processes: produce a product more rapidly, produce a higher-quality product, produce a product at lower cost, meet environmental impact requirements, ...
- There are several motivations for good control in consumer products: improve their performance, provide increased functionality (e.g., consider all the roles of controls in the modern automobile: achieving good handling qualities, good fuel economy, easier operation via cruise control and power steering, etc.), make them less expensive to operate, ...
- We will investigate **root locus** and **frequency-domain** methods for solving control problems

Root Locus Drawing Rules[1]

Given: Open-loop transfer function $G_{OL}(s)$ (perhaps $G_{OL}(s) = G(s) \cdot H(s)$),

$$G_{OL}(s) = \frac{K(s^m + \ldots + b_1 s + b_0)}{s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0} = \frac{K\,p(s)}{q(s)} \tag{1}$$

The following rules pertain to $K \geq 0$ [modified in square brackets for $K < 0$]:

1. Starting and Ending Points: Root loci start ($K = 0$) at the open-loop poles and end ($|\,K\,| \to \infty$) at the open-loop zeros (including $n - m$ zeros at ∞).

2. Segments on the Real Axis: Root locus segments lie on the real axis wherever there are an ODD [*EVEN*] number of open-loop poles and zeros to the right.

3. Asymptotes: $n - m$ root loci go to ∞ along straight-line asymptotes starting at $\sigma_A + j0$ where

$$\sigma_A = \frac{\Sigma(\text{poles of G}) - \Sigma(\text{zeros of G})}{n - m} = \frac{b_{m-1} - a_{n-1}}{n - m} \tag{2}$$

   at angles θ_A where

$$\theta_A = \frac{(2k+1)\pi}{n-m} \; [\, \frac{2k\pi}{n-m}] \;,\; k = 1, 2, \ldots (m - n) \tag{3}$$

4. Imaginary Axis Crossings: Use the Routh-Hurwitz method to determine $j\omega$-axis crossings and the corresponding value(s) of K (the "row of zeros" rule).

5. Break-away Points: Root loci break away from or break into real-axis segments wherever

$$p(s)\frac{dq(s)}{ds} = q(s)\frac{dp(s)}{ds} \tag{4}$$

6. Angles of Departure / Arrival: Assume a point s_θ arbitrarily near a pole (angle of departure) or zero (angle of arrival), then apply the fundamental angle relation

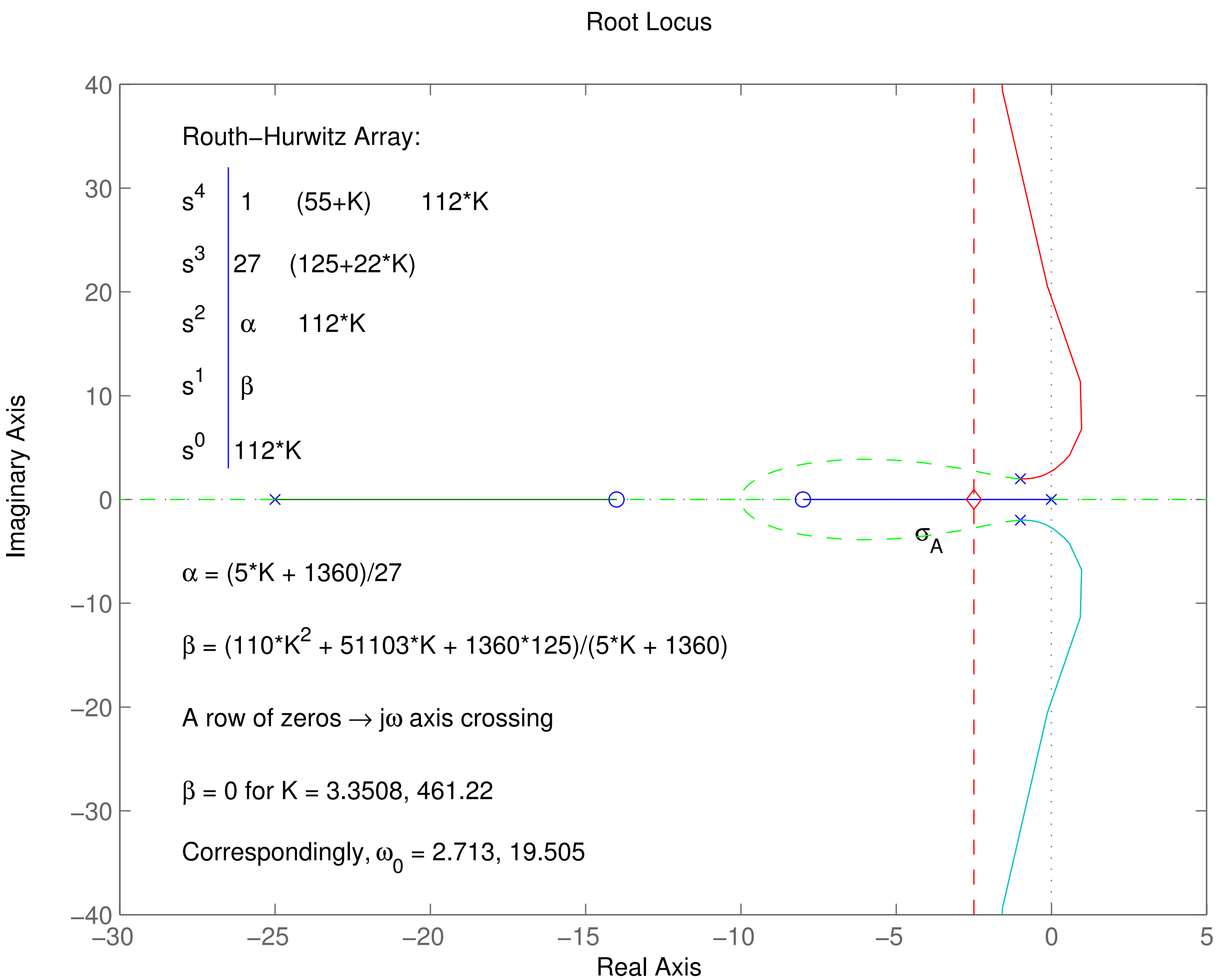
$$\Sigma(\text{angles from zeros}) - \Sigma(\text{angles from poles}) = (2k+1)\pi \;[\, 2k\pi\,] \tag{5}$$

7. Evaluating K: Pick a point s_d on the root locus, then determine the distances from there to the open-loop poles, $R_{p,i}$, i = 1, 2, ...and to the open-loop zeros, $R_{z,j}$, j = 1, 2, ...then $K = \Pi R_{p,i} / \Pi R_{z,j}$.

[1]Adapted from J. R. Rowland, *Linear Control Systems: Modeling, Analysis & Design,* John Wiley and sons, 1986.

Root Locus Example 1

$$G(s) = \frac{(s+8)(s+14)}{s(s^2+2s+5)(s+25)} \;\; ; \;\; \sigma_A = \frac{\Sigma p_i - \Sigma z_i}{n_p - n_z} = -2.5$$

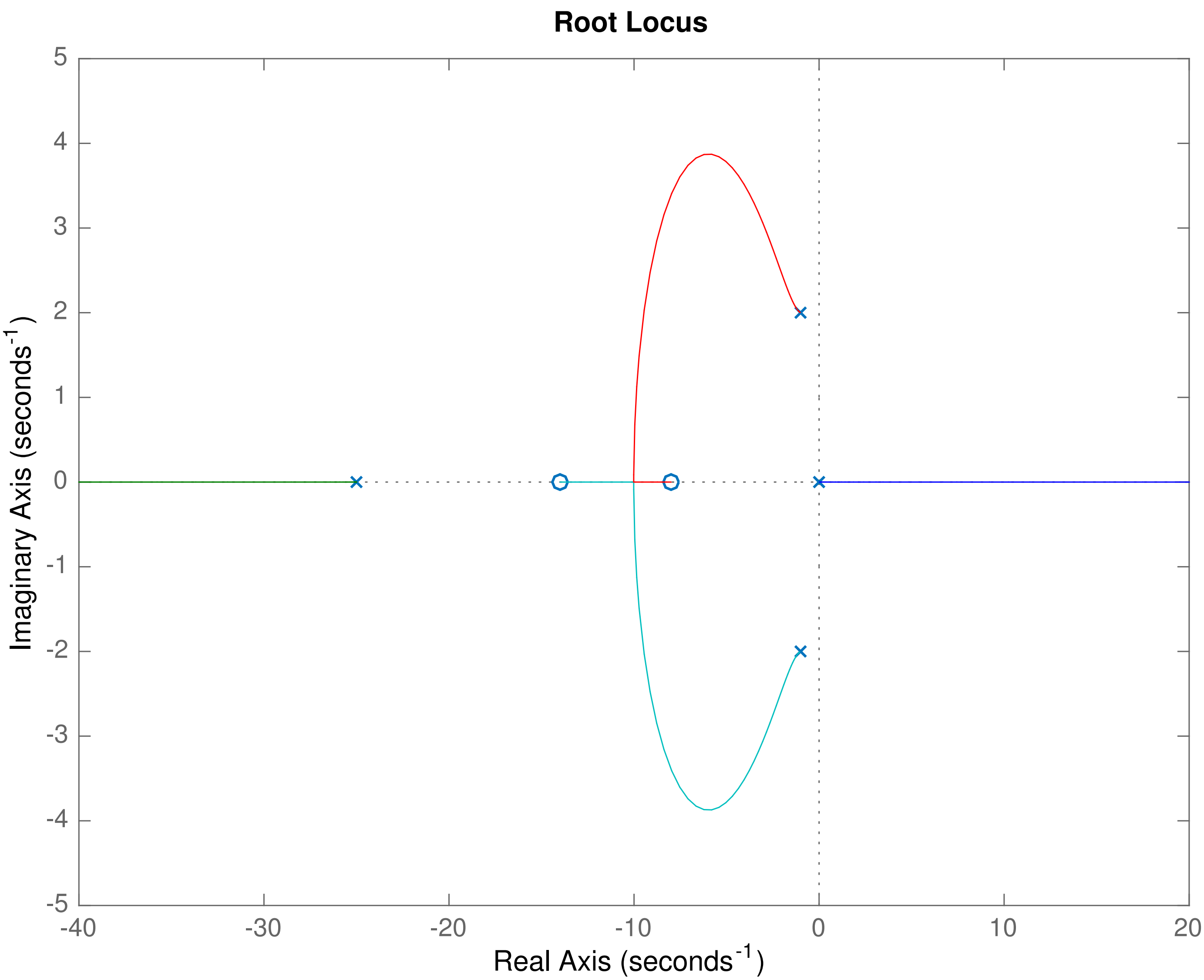


Some useful MATLAB commands:

```
n1 = [ 1  22  112 ];  % poly([ -8 -14 ]);
d1 = [ 1  27  55  125  0 ]; % poly([0 -1+2*j -1-2*j -25]);
figure(1); rlocus(n1,d1)
axis([-30 5 -40 40])

ro = length(d1) - length(n1); % relative order
% hint: a poly is s^n - sum(roots)s^(n-1) + ...
sga = (n1(2) - d1(2))/ro;
hold on; plot(sga,0,'rd')
plot([sga sga],[-40 40],'r--')
text(-4.4,-4.2,'\sigma_A')

% break-in / break-away points
dd1 = polyder(d1); % polynomial derivative!
dn1 = polyder(n1);
bkpol = conv(n1,dd1) - conv(d1,dn1);
roots(bkpol)
%% ans = -10.8937 (other roots complex => forget them)

figure(2)
rlocus(-n1,d1); % get neg RL data to plot
```

Root Locus Example 1 (Cont'd)

. . . and here is that root locus for negative K



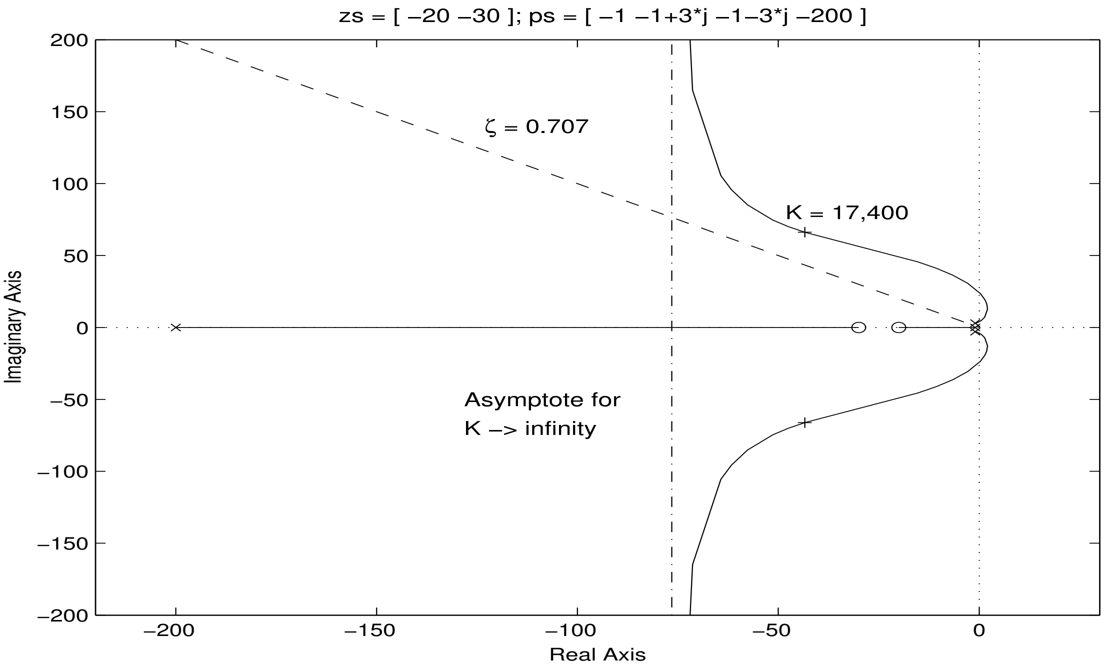
So, now the break in point is evident here

Control Methods to Improve Transient Response

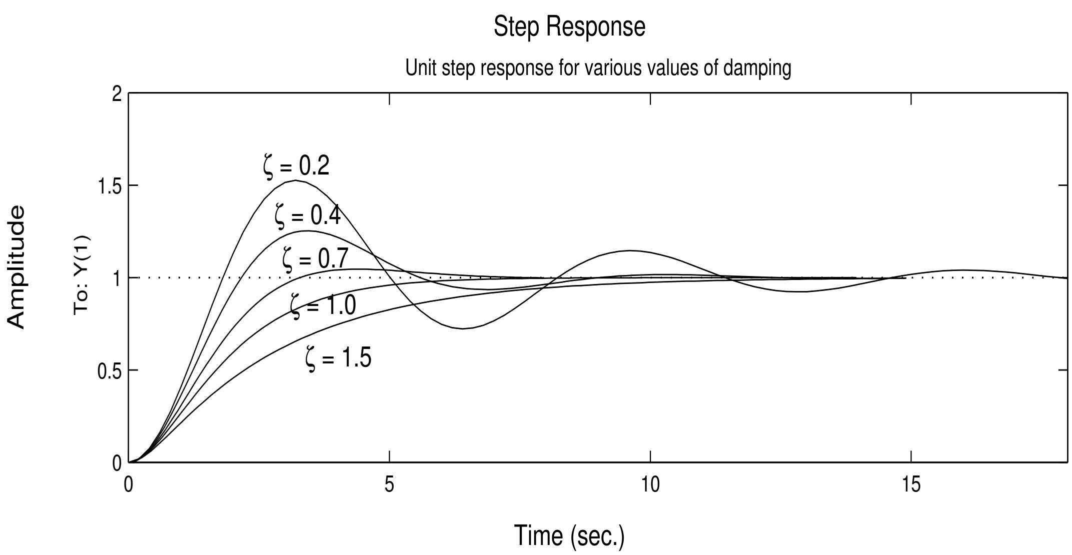
- One may take either a time-domain view (improve the rise time and/or percent overshoot) or a frequency-domain view (improve the system's bandwidth)
- One may use the **dominant pole** concept for the time-domain viewpoint; however, you must check the validity of this assumption during and after your design work
- Given closed-loop poles that are roots of $s^2 + 2\zeta\omega_n s + \omega_n^2$, the parameter ζ governs percent overshoot (next page) and ω_n determines speed of response (double ω_n to reduce rise time 50 %)
- Common control strategies for improving transient response include (a) proportional plus derivative (PD) control, (b) lead compensation and (c) proportional plus rate feedback control
- For proportional plus rate feedback control one first designs an inner rate feedback loop to improve the plant characteristics, and then designs an outer proportional control loop; we haven't covered this technique this year
- In PD and lead compensation one designs a single dynamic compensator in one step

Root Locus Example 2

$$G(s) = \frac{(s+20)(s+30)}{(s+1)(s^2+2s+10)(s+200)} \;\; ; \;\; \sigma_A = \frac{\Sigma p_i - \Sigma z_i}{n_p - n_z} = -76.5$$

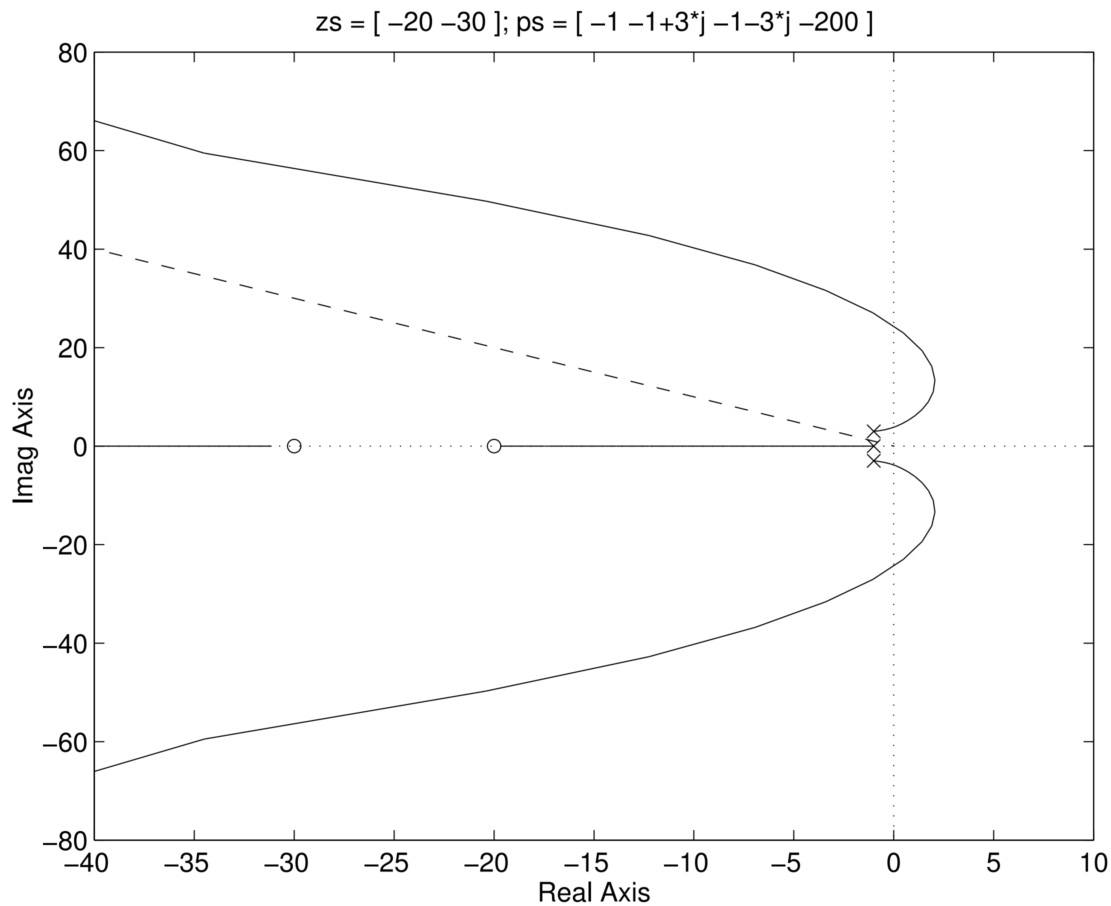


Decent damping is not possible (note: the line for constant ζ makes the angle $\theta = \cos^{-1}(\zeta)$ with respect to the real axis). Damping $\zeta = 0.707$ ($\theta = 45$ deg) is a common specification for low overshoot:



Root Locus Drawing (Cont'd)

Again, consider $G(s) = \dfrac{(s+20)(s+30)}{(s+1)(s^2+2s+10)(s+200)}$

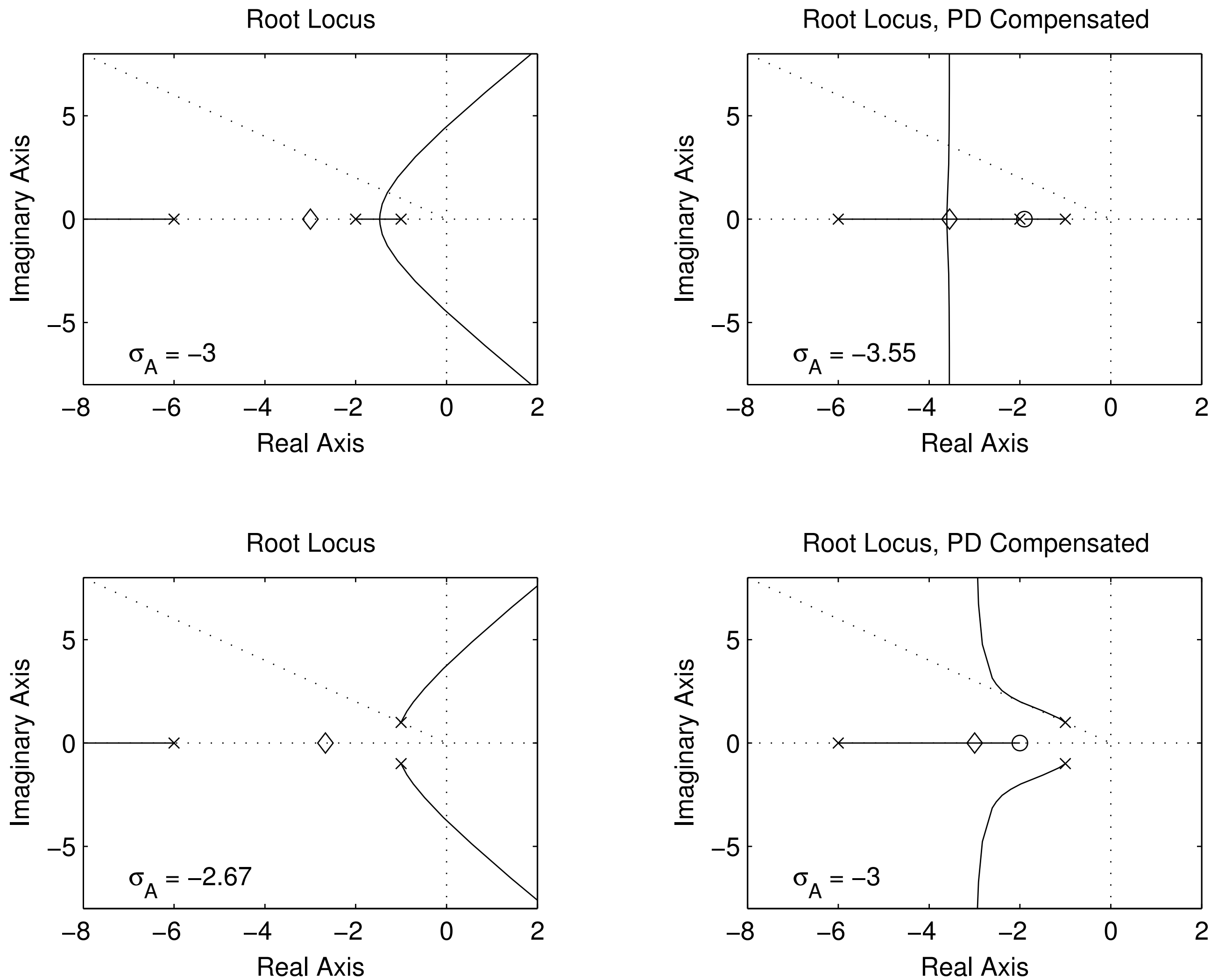


Close-up View $\Rightarrow$ **serious stability problems** in addition to poor damping ...so let's consider how we might improve the situation:

1. Use **Proportional + Derivative (PD) control**
2. Use **lead compensation**

Proportional / Derivative Compensation

Basic Idea: Adding an open-loop zero can be very beneficial:

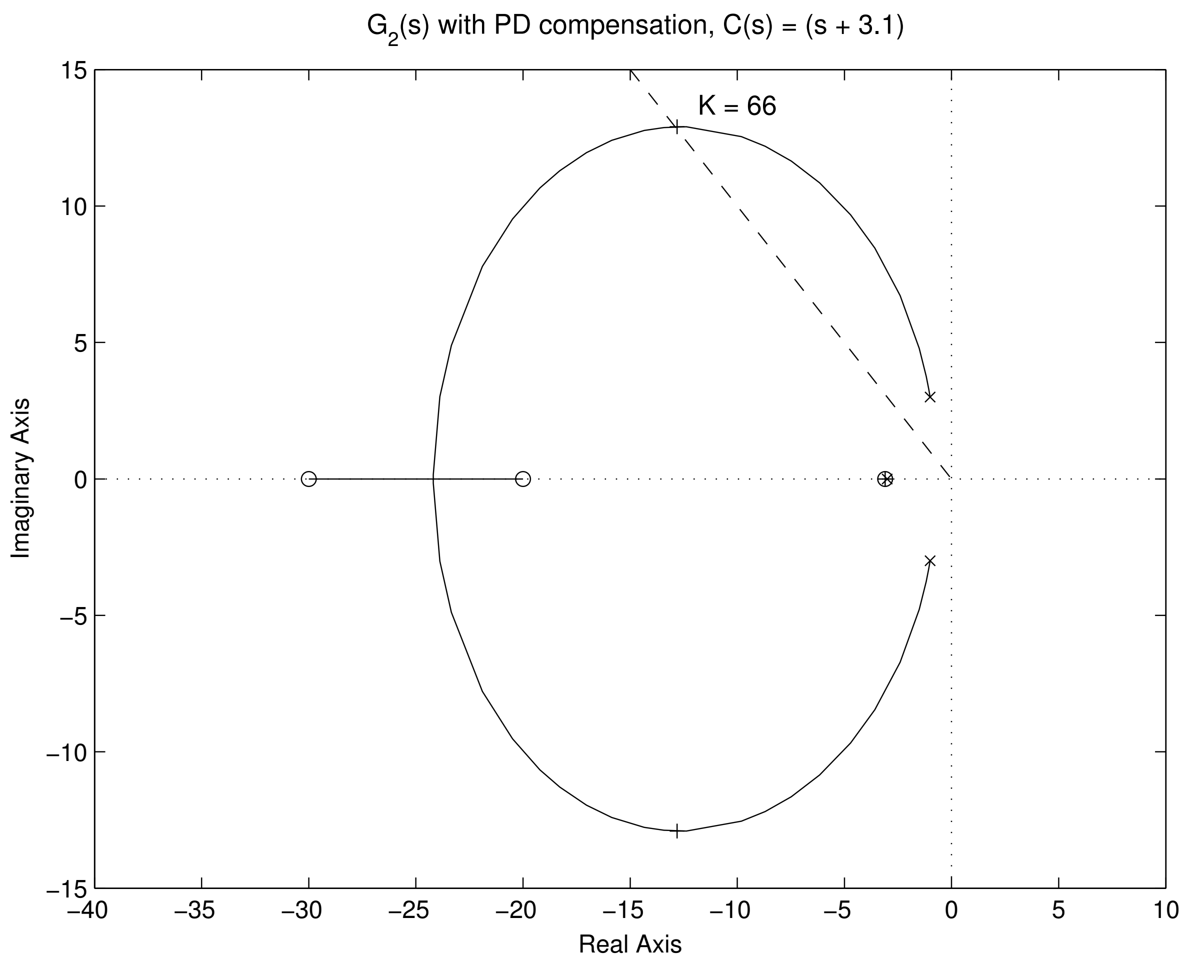


The added zero due to $C_{PD}(s) = K_D(s + \alpha)$ is beneficial:

- it can pull the root locus to the left, making the closed-loop system 2 to 3 times faster
- it may eliminate instability for high gain (e.g., if the original plant had $n - m = 3$ then the compensated open loop has $n - m = 2$, as above)
- it may ensure the "dominant poles hypothesis" if there is a near pole/zero cancellation (top case)

PD Compensation (Cont'd)

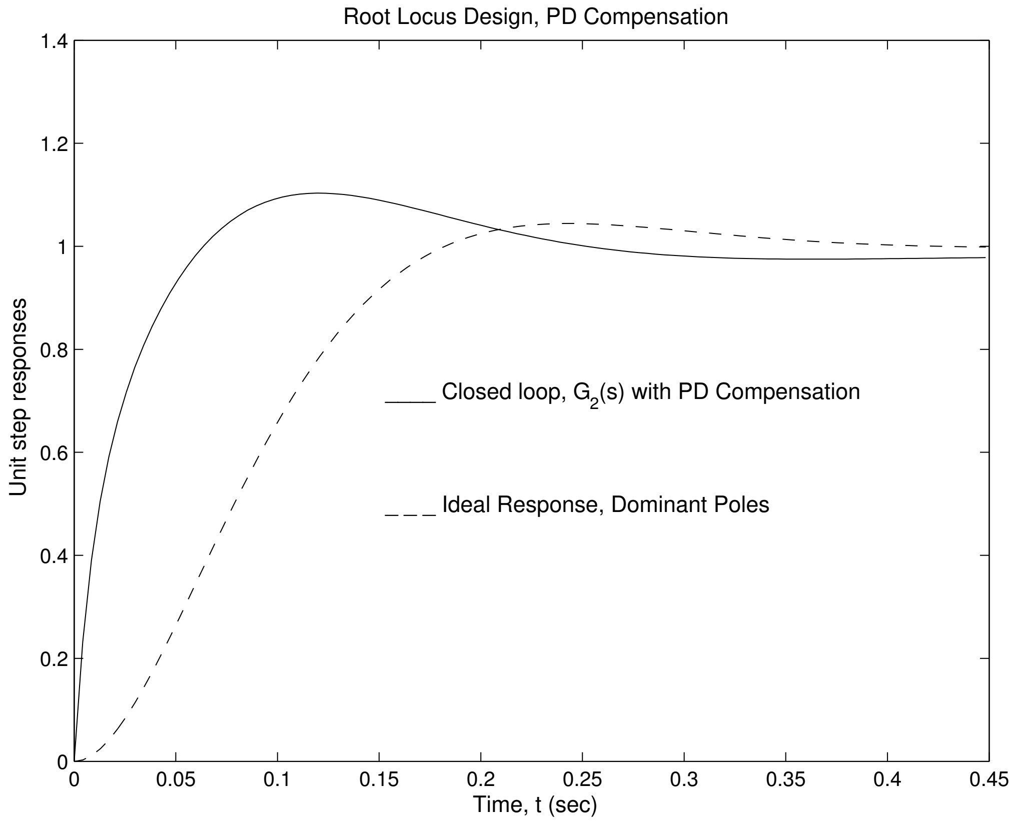
Let's apply PD compensation to Example 2:




```
pd = [ 1 3.1 ]; nc = conv(num,pd); % compensated numerator
rlocus(nc,den) % root locus of the compensated plant
hold on; plot([-15 0],[15 0],'--'); % plot zeta = 0.707 line
[K,clp] = rlocfind(nc,den); % determine K for desired zeta
```

The zero at $s = 3.1 \Rightarrow$ the root locus is pulled to the left; the near pole/zero cancellation $\Rightarrow$ the complex poles *might* be dominant; however, the zeros at -20 and -30 are close enough to the poles at $-12.8 \pm 12.8j$ to raise questions ...

Checking the PD Design




```
clnum = [ 0 K*nc ]; clden = den + clnum; % build closed-lp system
step(clnum,clden); % generate the step response
p = roots(clden); d2 = poly([p(2) p(3)]); % poly of complex poles
[y2,x2,t2] = step(d2(3),d2); % response of ideal 2nd-order system
hold on; plot(t2,y2,'--')
```

The actual PD design exhibits somewhat more overshoot than desired – however, the response is significantly faster than the "dominant poles" argument would suggest (we were lucky this time)

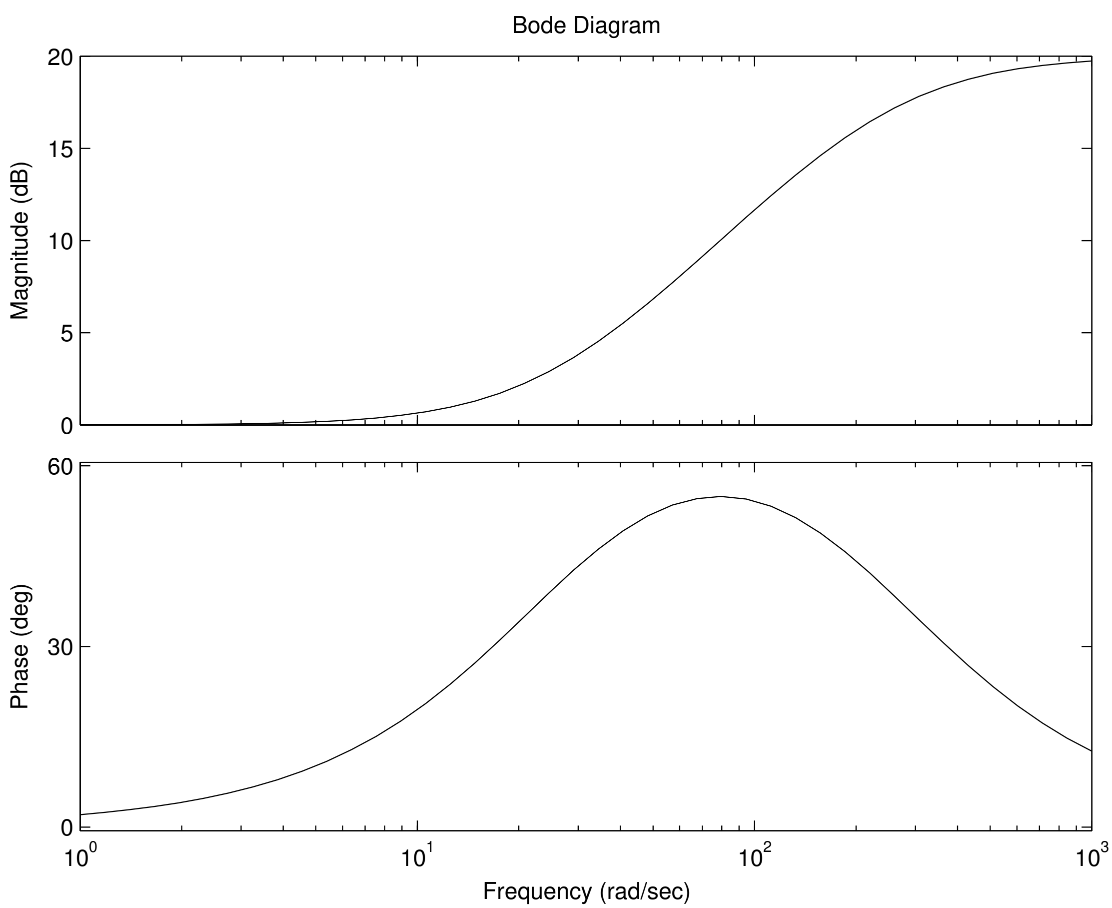
Overview of Lead Compensation

Basic Idea: The attainable closed-loop response is too slow; use lead compensation to **speed up the complex poles substantially** (increase ω_n). A lead compensator has the form

$$C_{LEAD} = \frac{1 + s/\alpha}{1 + s/R\alpha} = R\frac{s + \alpha}{s + R\alpha} \tag{6}$$

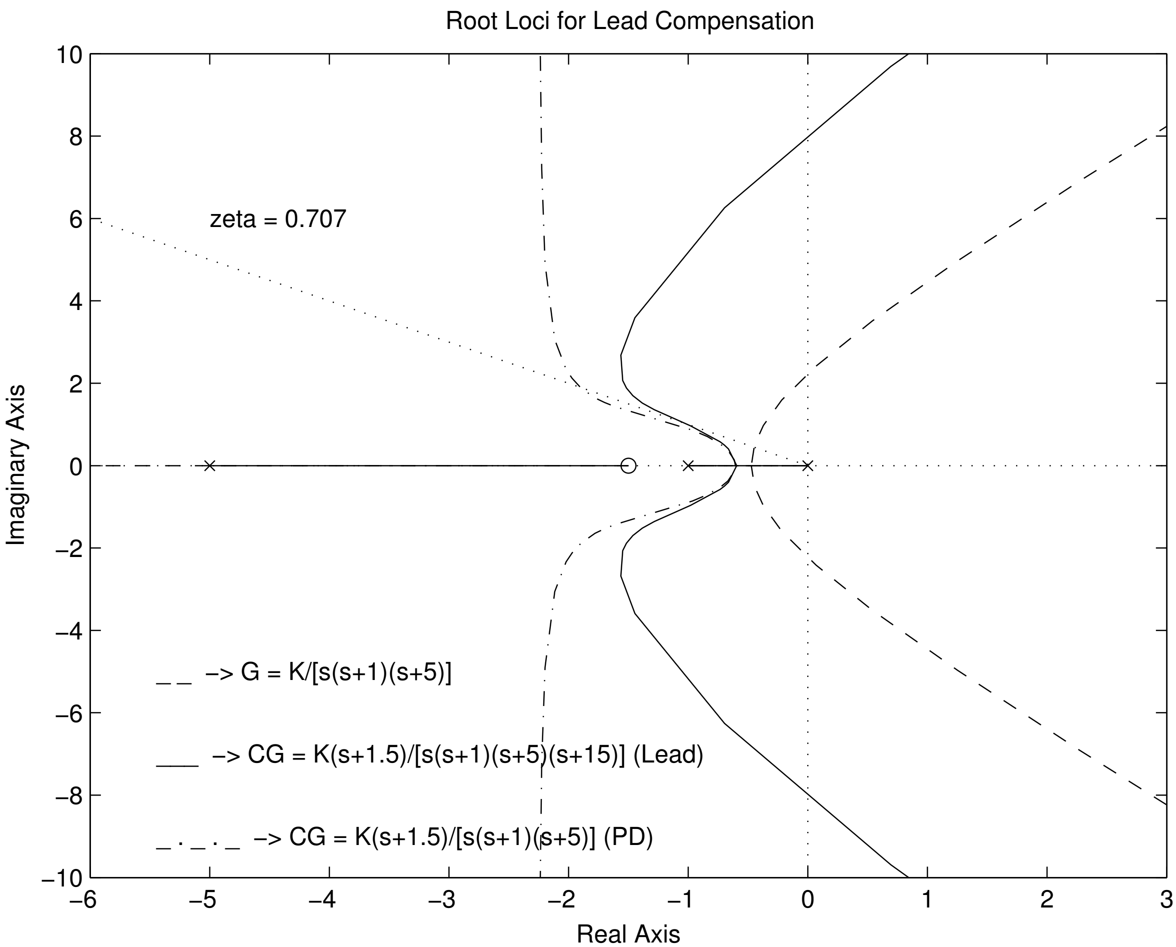
where R is the pole/zero ratio and α places the zero; we want to place the pole/zero pair so that the original root locus is "pulled to the left" by the zero – note that the low-frequency gain is unity

```
R = 10; alfa = 25;
num = [ R R*alfa ]; den = [ 1 R*alfa ];
bode(num,den)
```



Lead Compensation via Root Locus

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

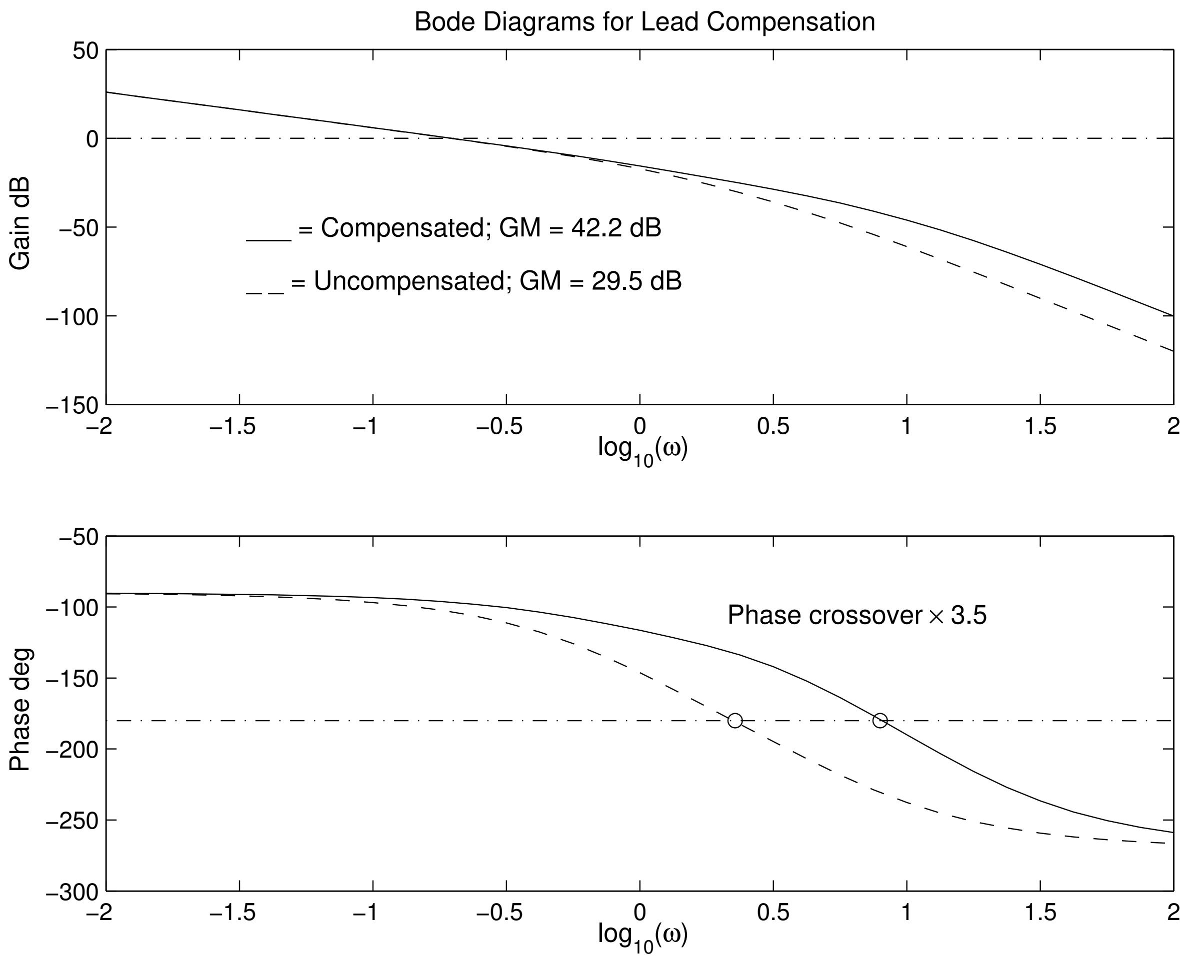


Detailed View: Pulling the dominant closed-loop poles "to the left" increases the break frequency ω_n which, if you can do so without changing the damping ζ, will make the response faster. Note that the asymptote shift is very beneficial:

$$\sigma_A = \frac{\Sigma(\text{poles of CG}) - \Sigma(\text{zeros of CG})}{n-m} = \sigma_A^{CG(s)} - \frac{(R-1)\alpha}{n-m}$$

Overview of Lead Compensation (Cont'd)

This is what lead compensation achieves in the frequency domain:

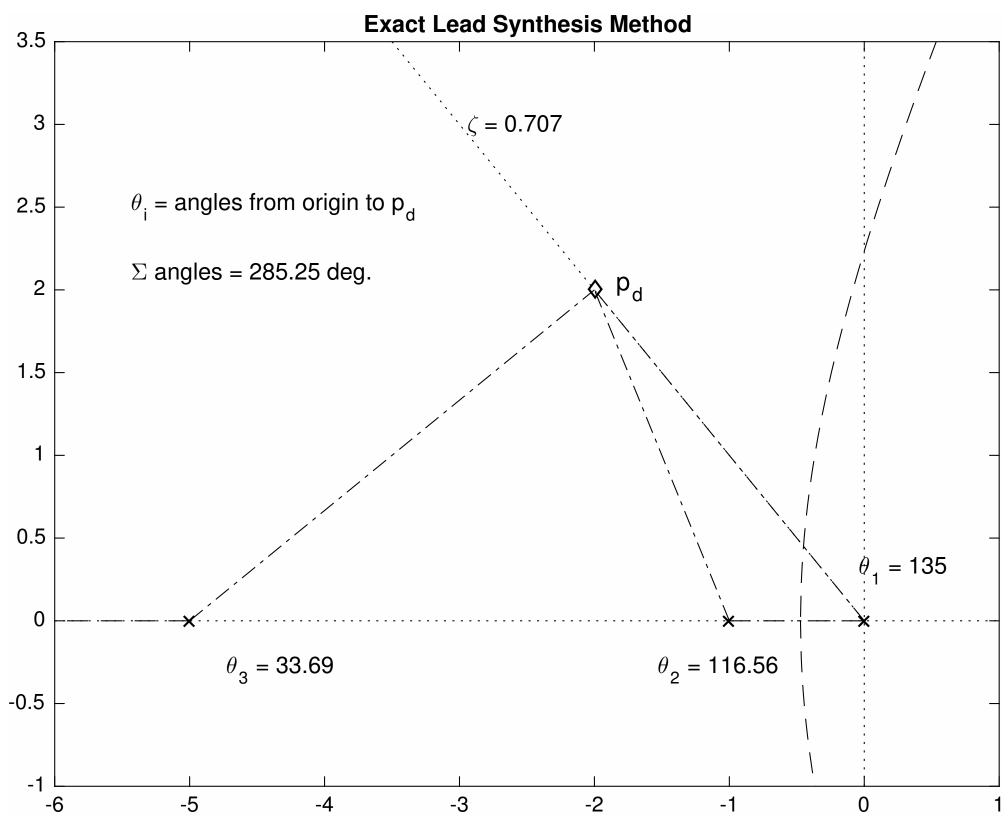


The gain-margin crossover point is at significantly higher frequency, and the gain margin is improved.

Exact Lead Compensation

Given a root locus with complex poles that are unsatisfactory it is possible to synthesize a lead compensator that moves them to a *reasonable* desired location as follows:

- Pick the desired location p_d
- Add up the angle contributions from the **given** plant poles and zeros and calculate the **angle deficit** θ_d, the amount this sum differs from ± 180 deg; example: $\theta_d = 105.25$ deg (see figure)
- Place the lead compensator pole and zero to make up the angle deficit ($\theta_{zc} - \theta_{pc} = \theta_d$)



Exact Lead Compensation Example

Proceeding on this example, we note that if we place the lead compensator zero to cancel the pole at -1 (i.e., choose $\alpha = 1$) we have $\theta_{zc} = 116.56$ deg, so we need $\theta_{pc} = \theta_{zc} - \theta_d = 11.31$ deg; we can thus solve for R via

$$\frac{2}{R\alpha - 2} = \tan(11.31) \;\; \rightarrow \;\; R = 12$$

Does this work? See for yourself:

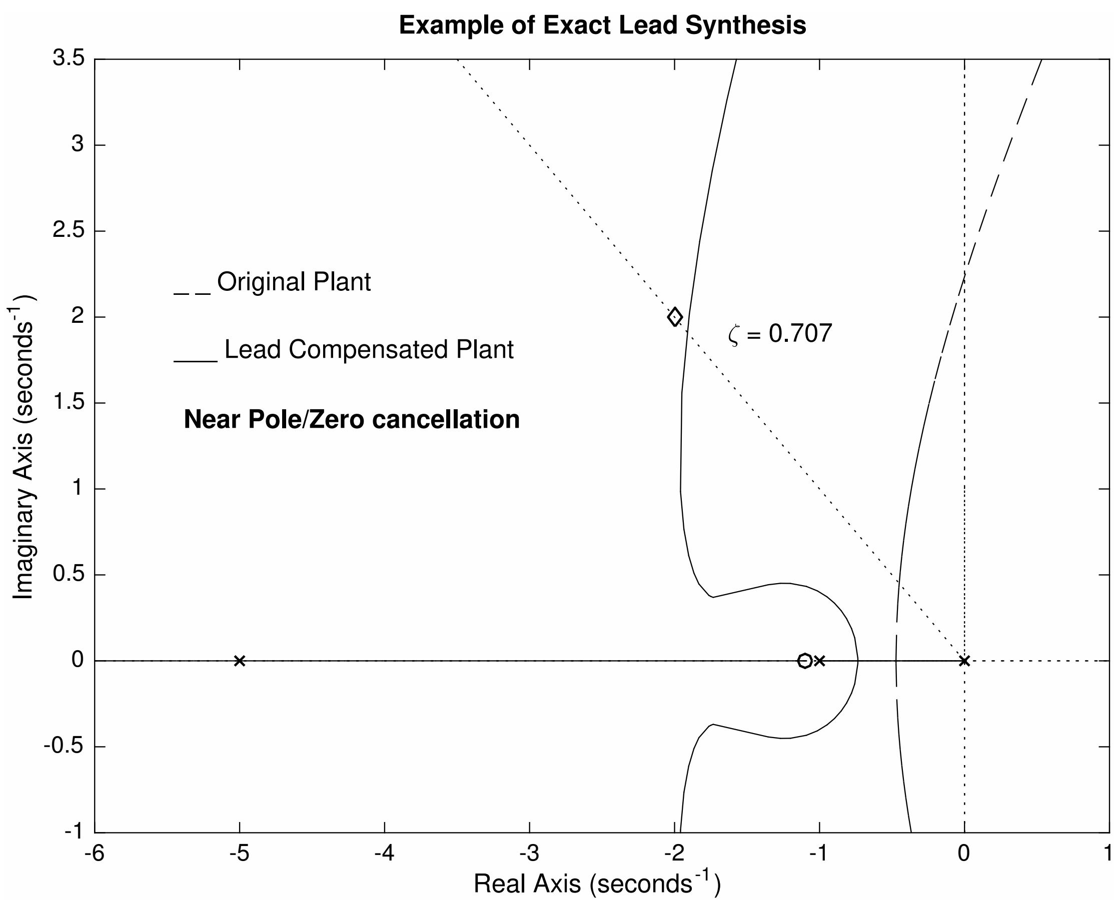
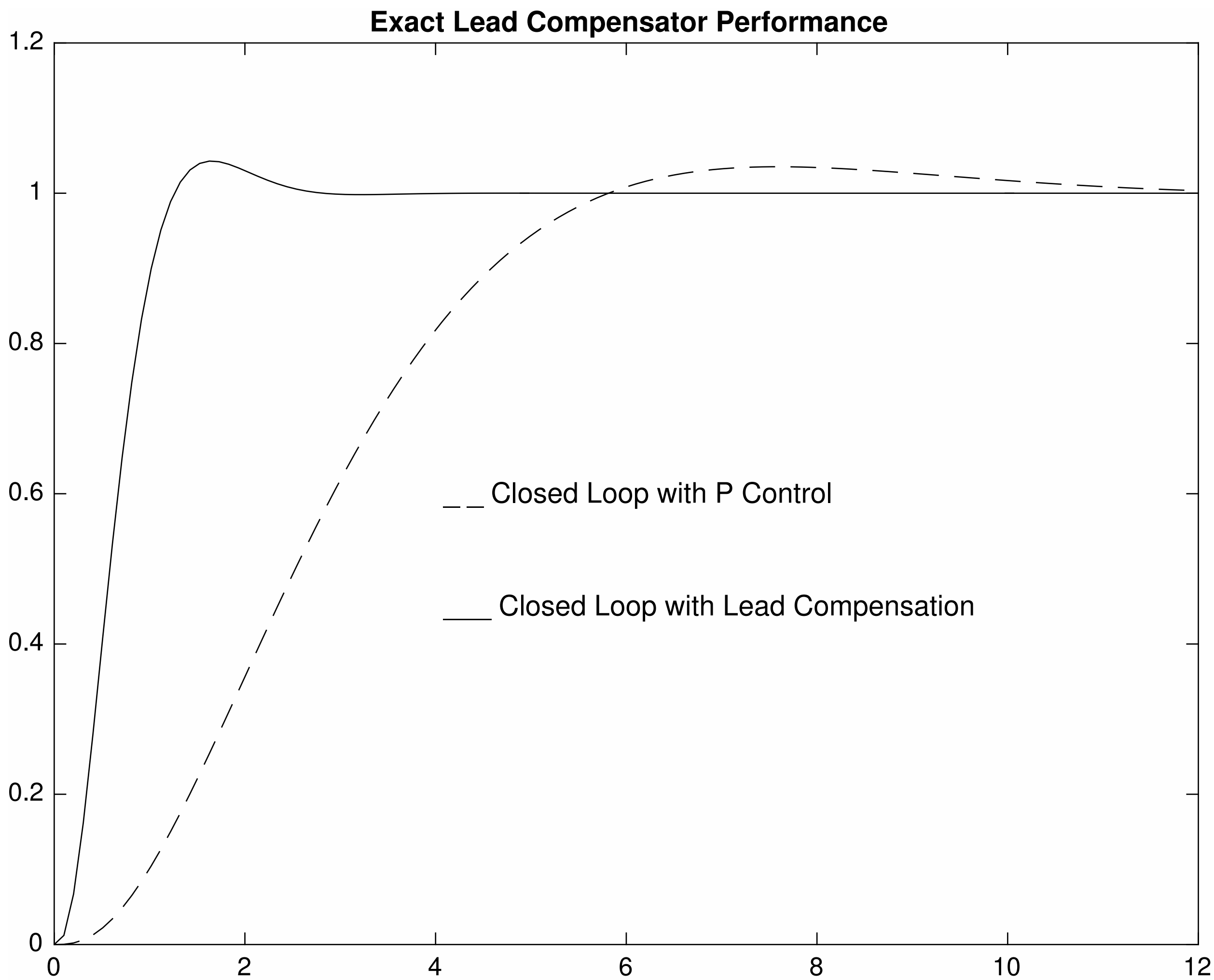


Figure 1: Exact Lead Compensator Design, Near Pole/Zero Cancellation

Note: exact or near pole/zero cancellation is often effective in lead compensator design

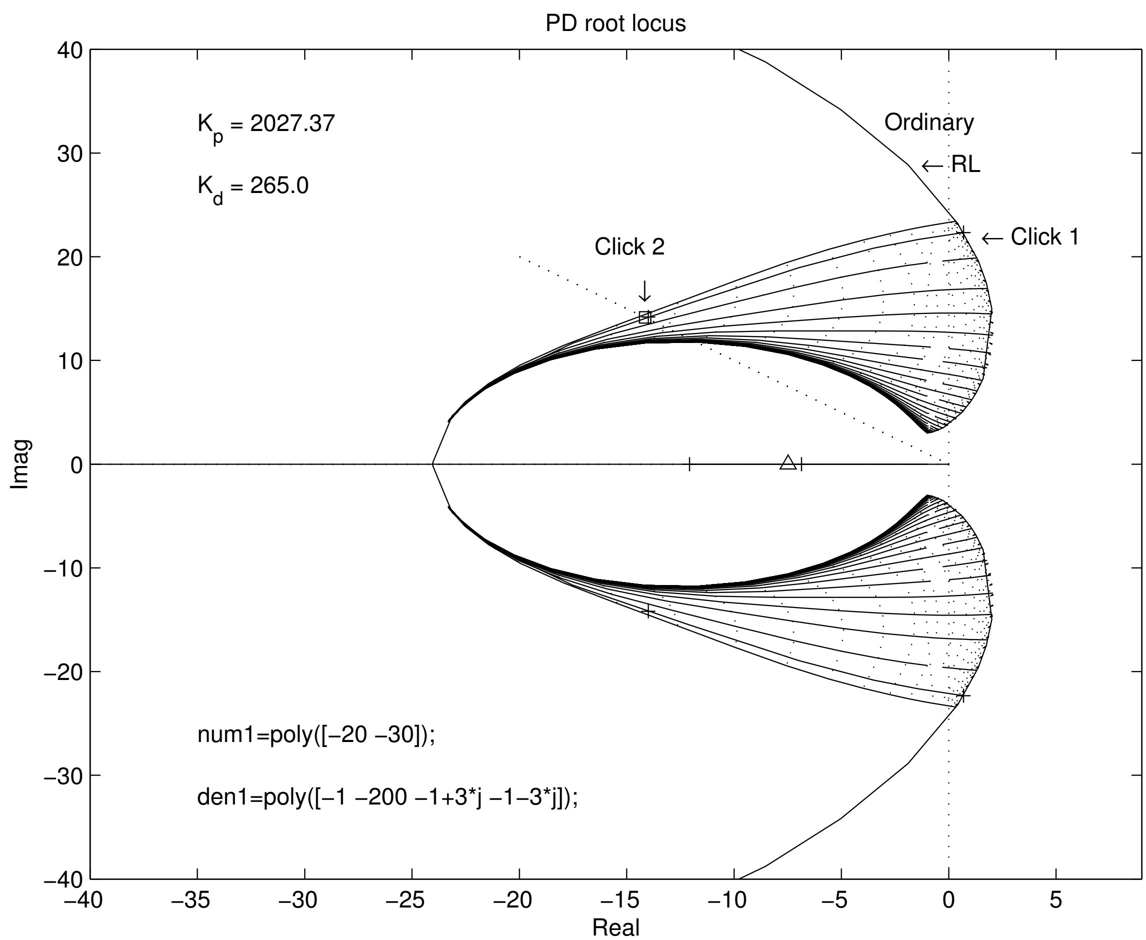
Exact Lead Compensation Example (Cont'd)

Does this work? **See for yourself:**



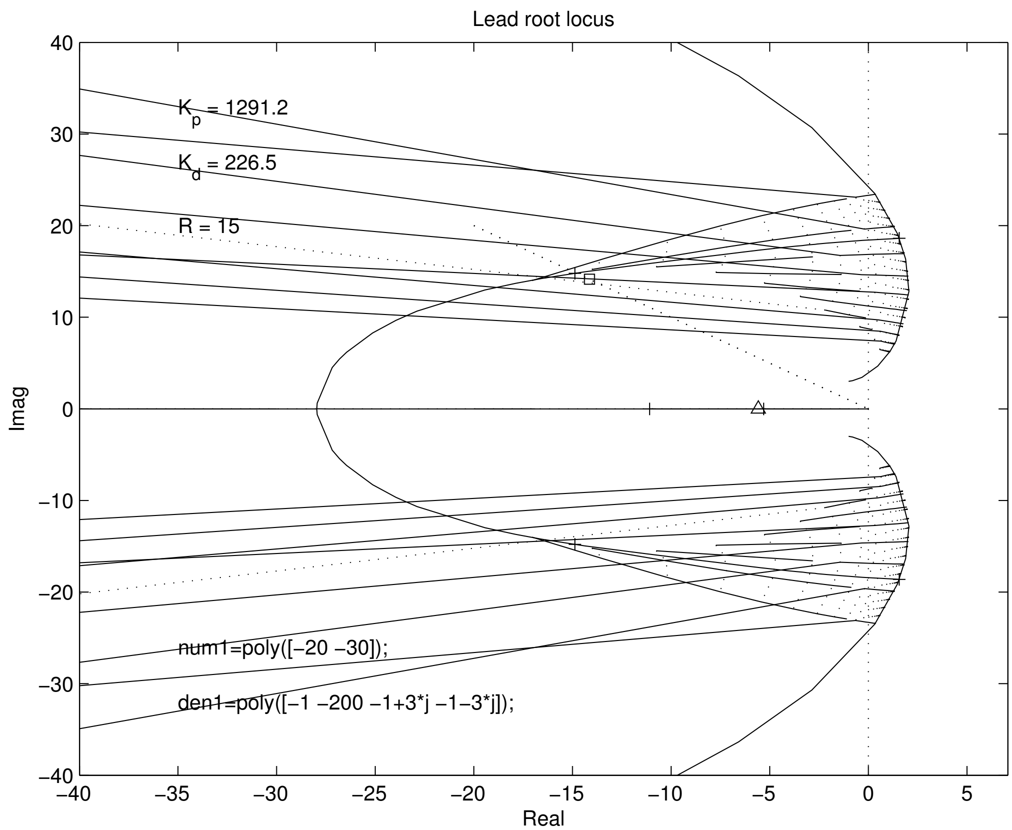
PD Compensation via New "Two-D Root Locus" Software

```
zeta = 0.707; omega_n = 15;
[Kp,Kd,pls]=rlocus2DPD(num,den,zeta,omega_n);}
```



First, click on a point on the ordinary root locus (magenta) that is the origin of a mesh-line passing through the target closed-loop pole location; then click on the new root locus plot (green) that is nearest to the target.

Lead Compensation via New "Two-D Root Locus" Software



First, click on a point on the ordinary root locus (magenta) that is the origin of a mesh-line passing through the target closed-loop pole location; then click on the new root locus plot (green) that is nearest to the target.

Improving Steady-State Behaviour

- Steady-state error can be studied using the Final Value Theorem of Laplace transforms: $y_{ss} = \lim_{s \to 0}(sY(s))$
- Therefore, the governing issue is: What is W_{LF} (see my handout[2] on constructing Bode plots – recall that the limit of $W(s)$ as $s \to 0$ is W_{LF})?
- If $W_{LF} = K$ (a type 0 system) and $W(s)$ is incorporated in a control system with unity feedback, then steady-state error for a unit step input is $e_{ss} = 1/(1 + K)$ and for a ramp input it is infinite
- If $W_{LF} = K/s$ (a type 1 system) and $W(s)$ is incorporated in a control system with unity feedback, then steady-state error for a unit step input is $e_{ss} = 0$ and for a ramp input it is $e_{ss} = 1/K$
- This demonstrates that e_{ss} for a given input (step, ramp etc.) can be **eliminated** if the **type** of the system is increased (e.g., type 0 to type 1), by adding an integrator or **greatly reduced** by substantially increasing the low-frequency gain K
- Proportional plus integral (PI) control increases the system type, and lag compensation may be used to substantially increase the low-frequency gain
- In applying either strategy, the usual objective is to improve steady-state error **without** much modification of the transient response

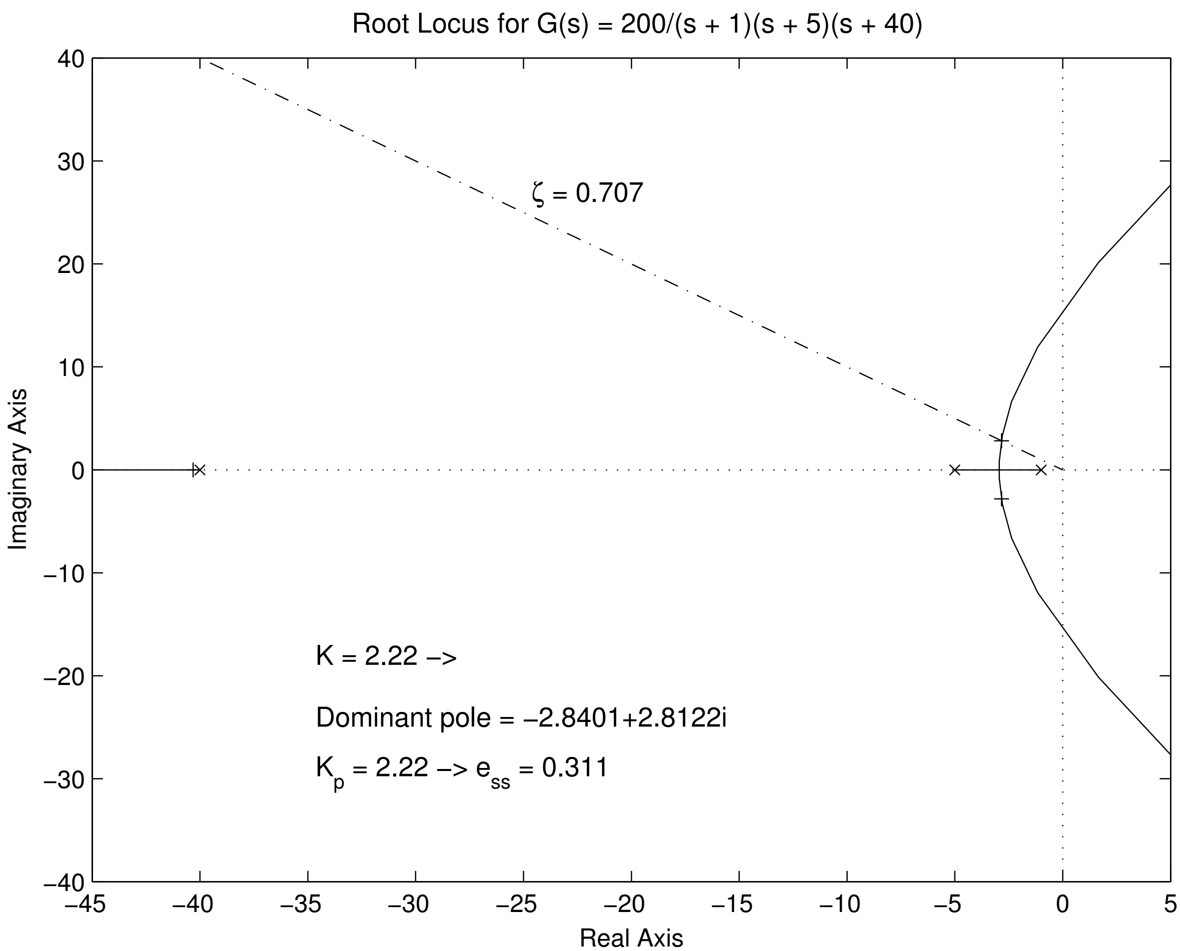
[2] In my handout on frequency response we were using $W(s)$ for the transfer function, here it's $G(s)$ – hopefully, that is not a problem . . .

Basics of PI Compensation

Basic Idea: PI Compensation completely eliminates steady-state error of the type existing in the uncompensated system.

Given the plant

$$G(s) = \frac{200K}{(s+1)(s+5)(s+40)}$$



With proportional control, $K_p = 2.21$ so $e_{ss} = 0.31$ for a step input, which is excessive. For a PI compensator,

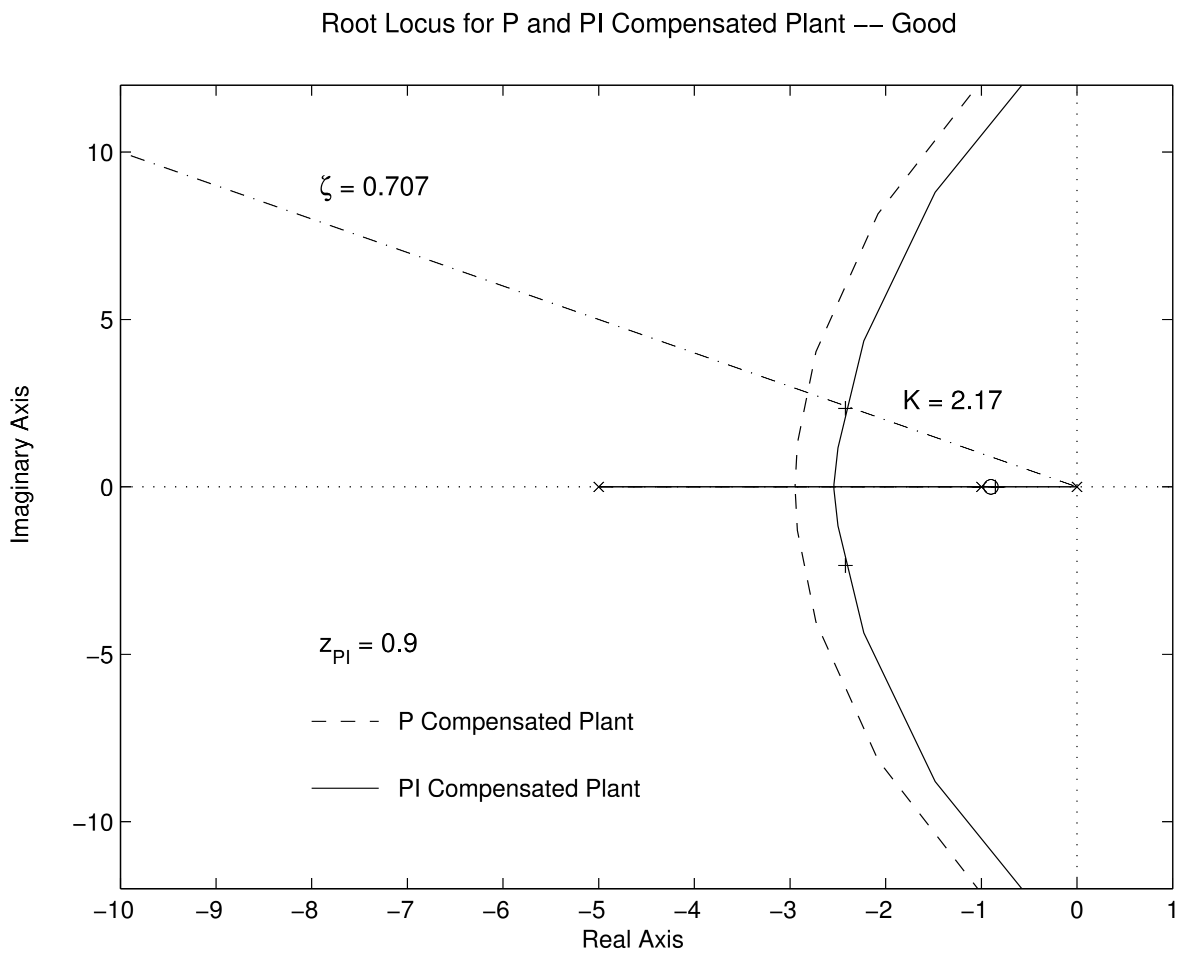
$$C(s) = K_p + K_i/s = \frac{K_p(1+sT_i)}{sT_i}$$

the question is: where to put the zero? There are several good strategies, including pole/zero cancellation (if feasible) and a "small influence angle" rule of thumb.

PI Compensation – P/Z Cancellation

A **good choice**: put the zero near the pole at -1

```
denpi = poly([ 0 -1 -5 -40 ]); %% new pole at s = 0
numpi1 = 200*[ 1/0.9 1 ];      %% factor (s/a + 1)
figure; rlocus(numpi1,denpi);
```

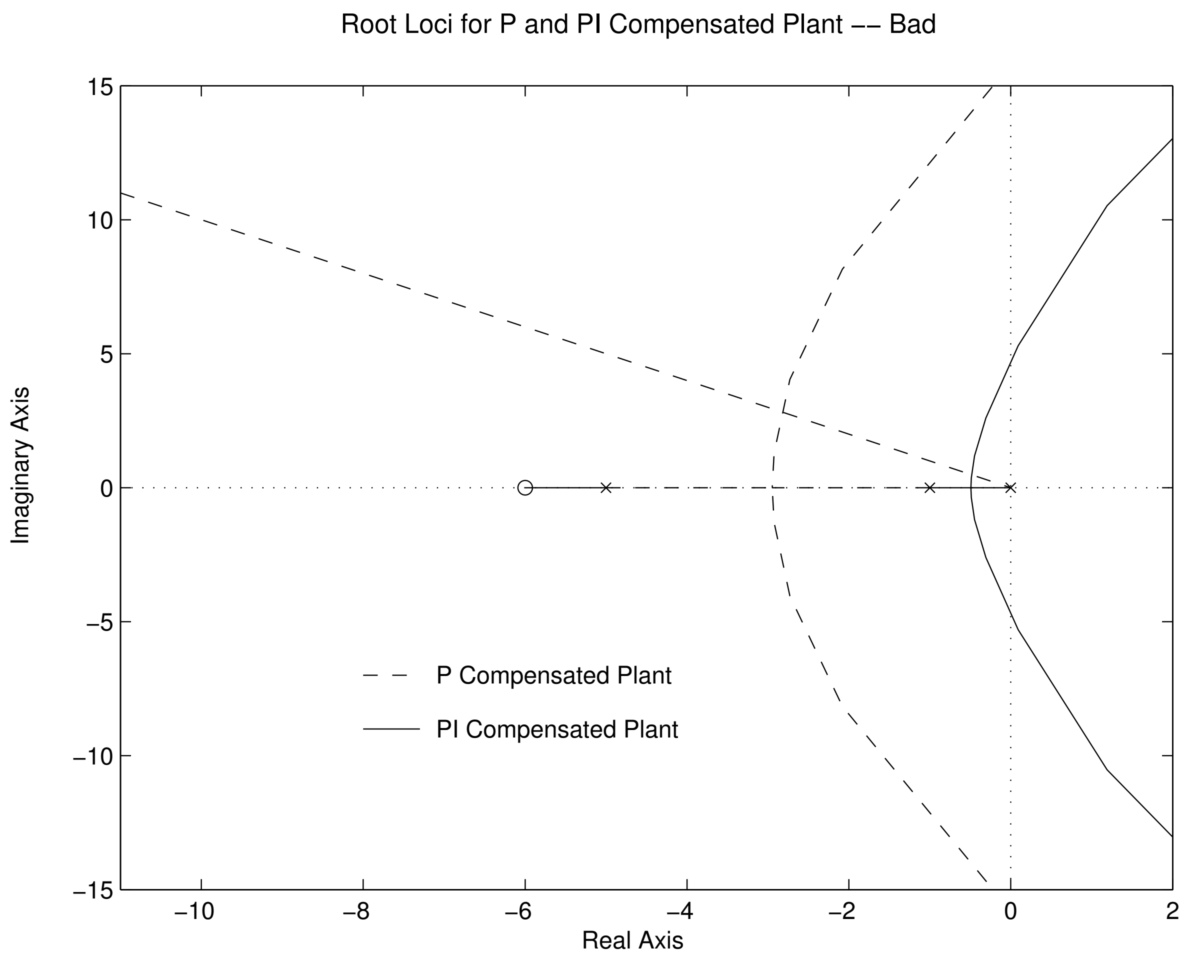


The PI compensator zero at -0.9 pulls the root locus to the right slightly $\rightarrow$ still a reasonably fast response **and** zero steady-state error for a step input!

PI Compensation – P/Z Cancellation (Cont'd)

A **bad choice**: zero between the poles at – 5 and – 40

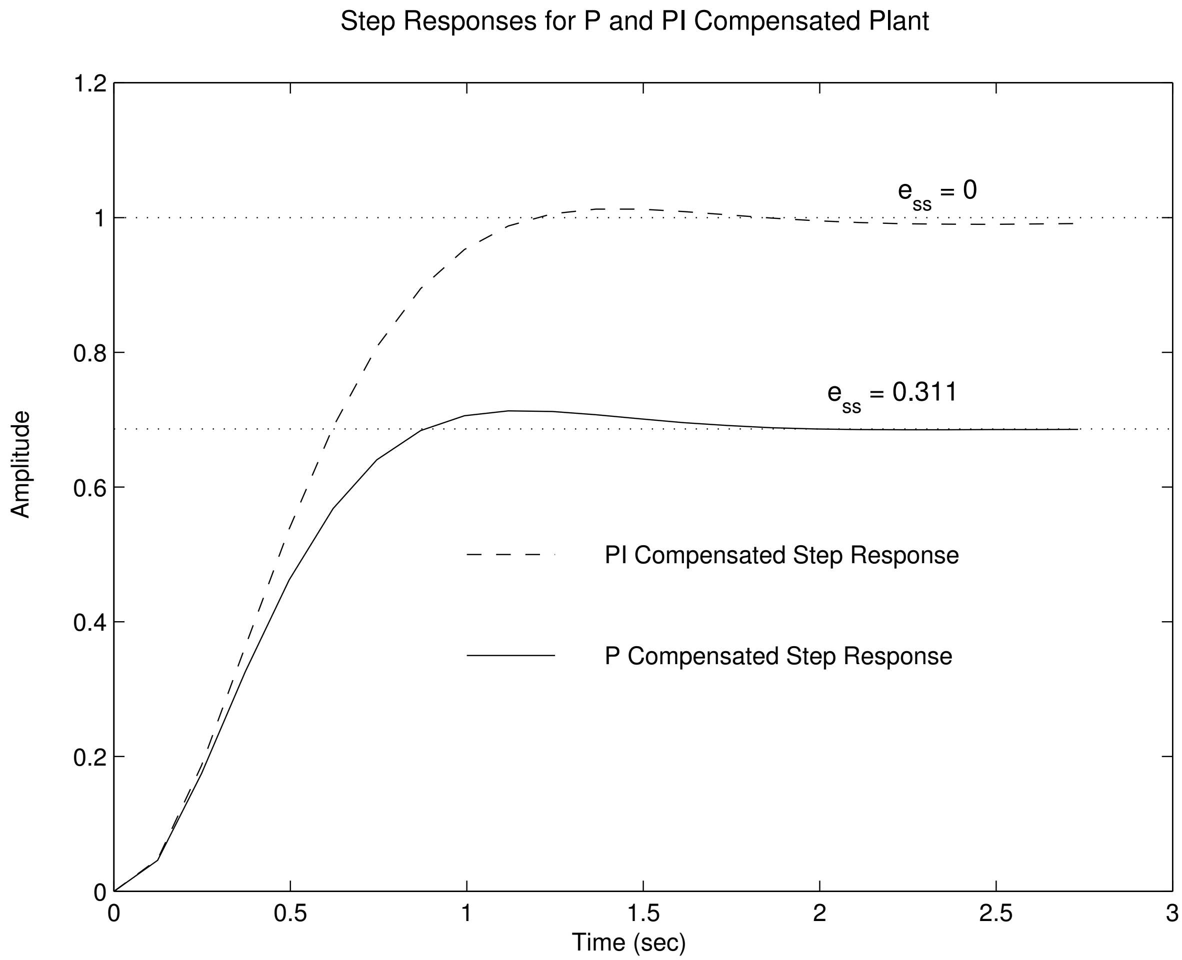
```
denpi = poly([ 0 -1 -5 -40 ]);
numpi2 = 200*[ 1/6 1 ];
figure; rlocus(numpi2,denpi);
```



Now, the root locus has to depart between 0 and – 1 ; the PI compensator zero at – 6 does little to pull the root locus to the left, so the response would be much slower . . .

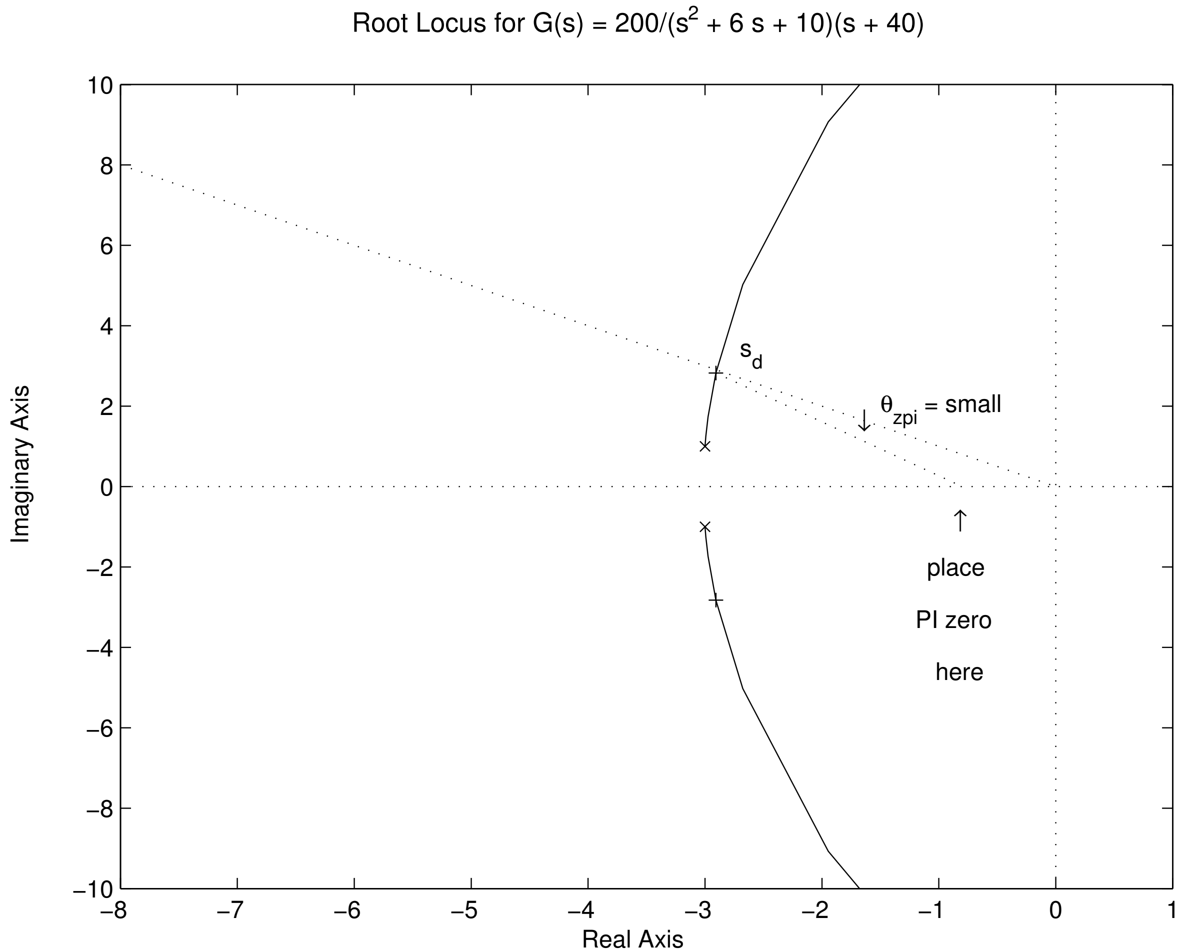
PI Compensation – P/Z Cancellation (Cont'd)

Finally, here's the final result for $z_{PI} = -0.9$:



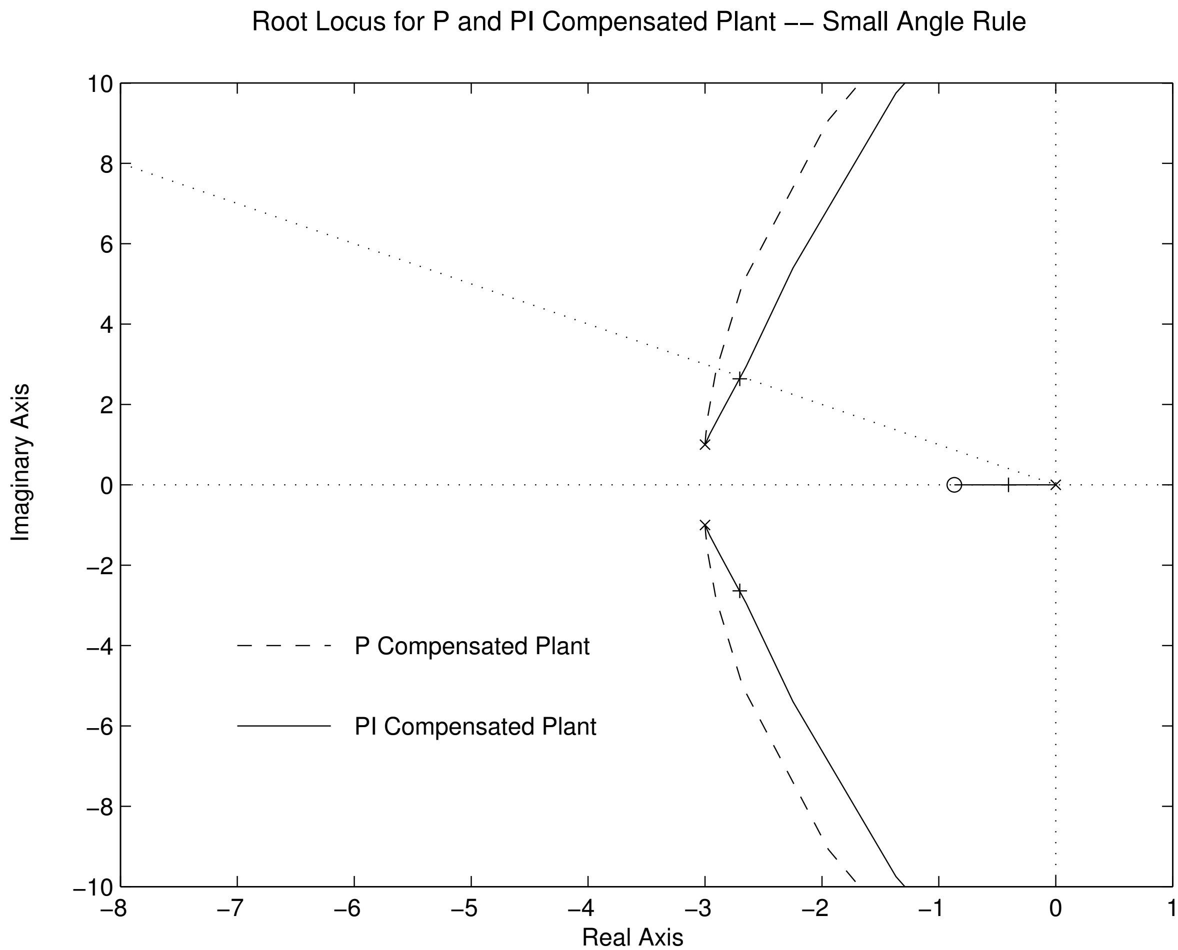
PI Compensation – "Small Influence Angle"

In this case we don't want to sacrifice much speed of response (reduce the ω_n significantly), so we place the PI zero so it only has a small influence on the root locus near the desired pole location s_d (based on the design specifications)



Let's see how this works …

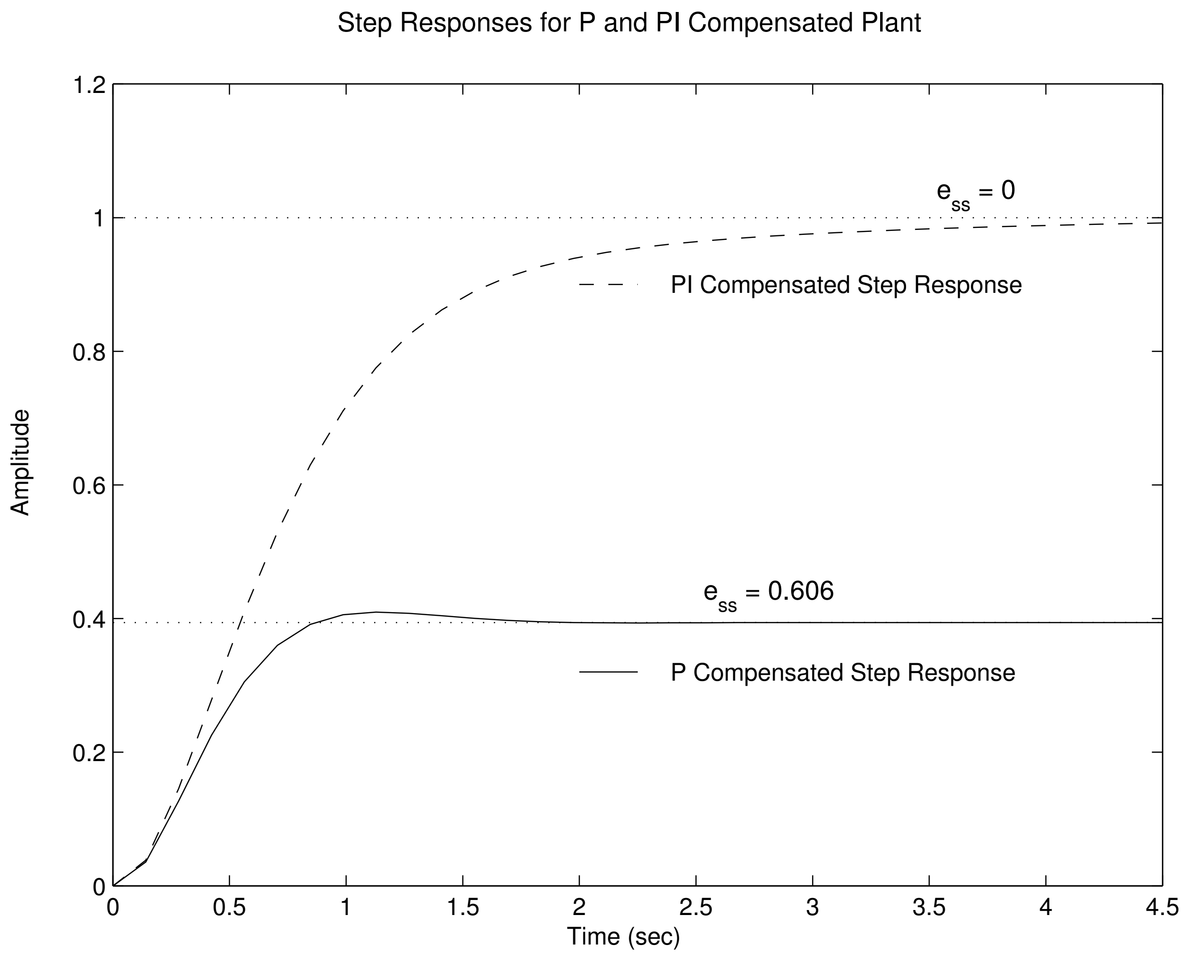
PI Compensation – "Small Influence Angle" (Cont'd)



Even though the angle θ_{zpi} is as large as 10 deg we still obtain good results ... although we have a less desirable slow eigenvalue (small real pole).

PI Compensation – "Small Influence Angle" (Cont'd)

Finally, here's the final result:



Overview of Lag Compensation

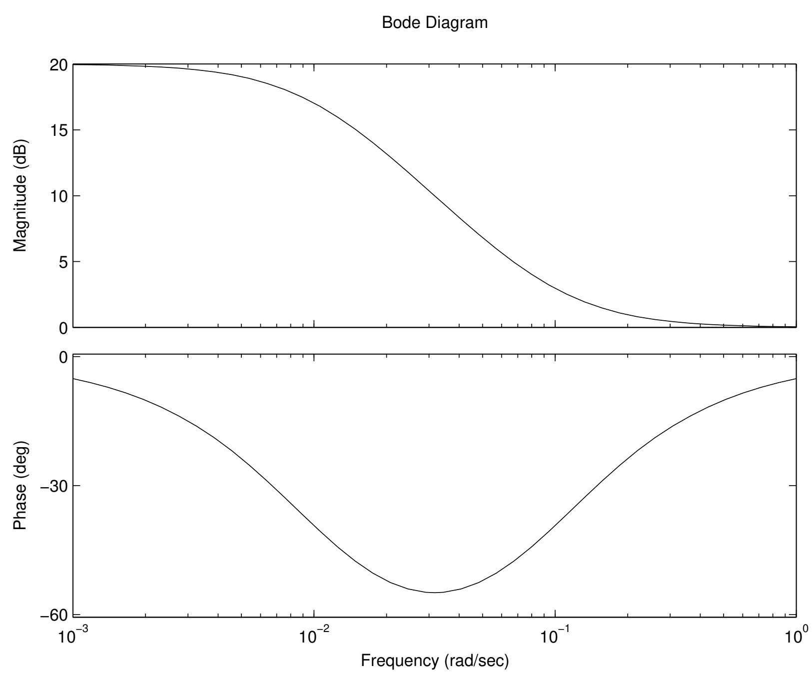
Basic Idea: The attainable closed-loop poles are acceptable but steady-state error is not; use lag compensation to **substantially increase low-frequency gain** (reduce steady-state error); the compensator has the form

$$C_{LAG} = R\frac{1 + s/R\alpha}{1 + s/\alpha} = \frac{R + s/\alpha}{1 + s/\alpha} = \frac{s + \alpha R}{s + \alpha} \quad (7)$$

where R will be the increase in low-frequency gain and α the low-frequency break point.

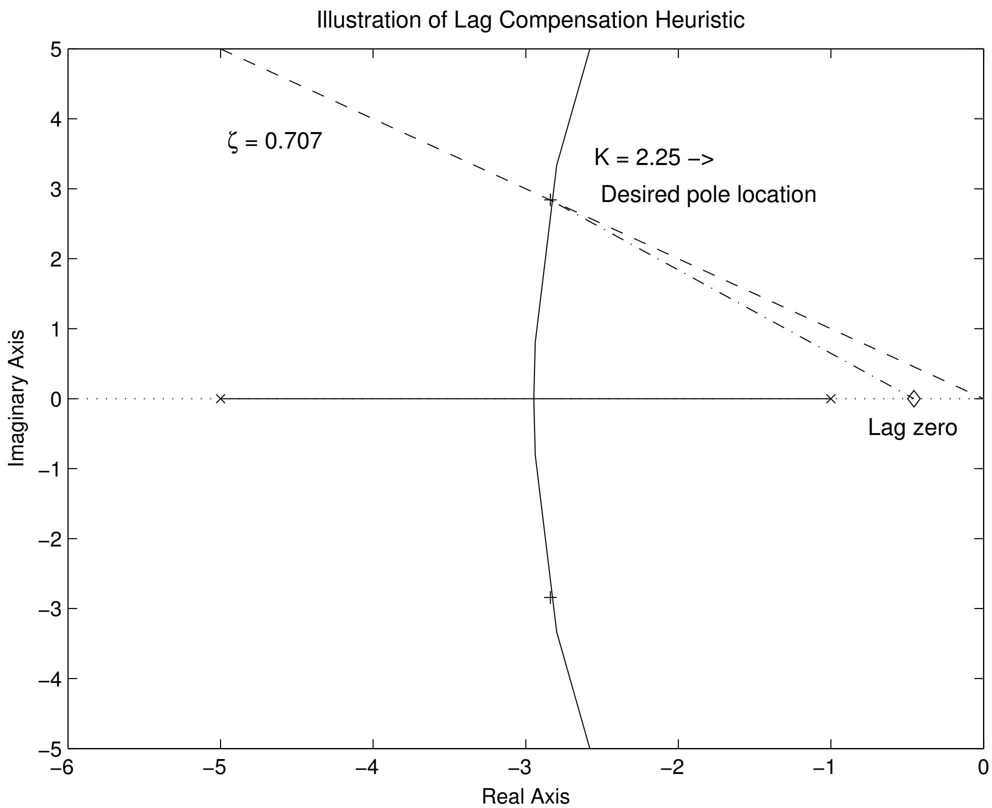
Strategy: Place the pole/zero pair so that the original root locus is "not disturbed very much" near the dominant poles.

```
R = 10; alfa = 0.01;
num = [ 1/alfa R ]; den = [ 1/alfa 1 ];
bode(num,den)
```



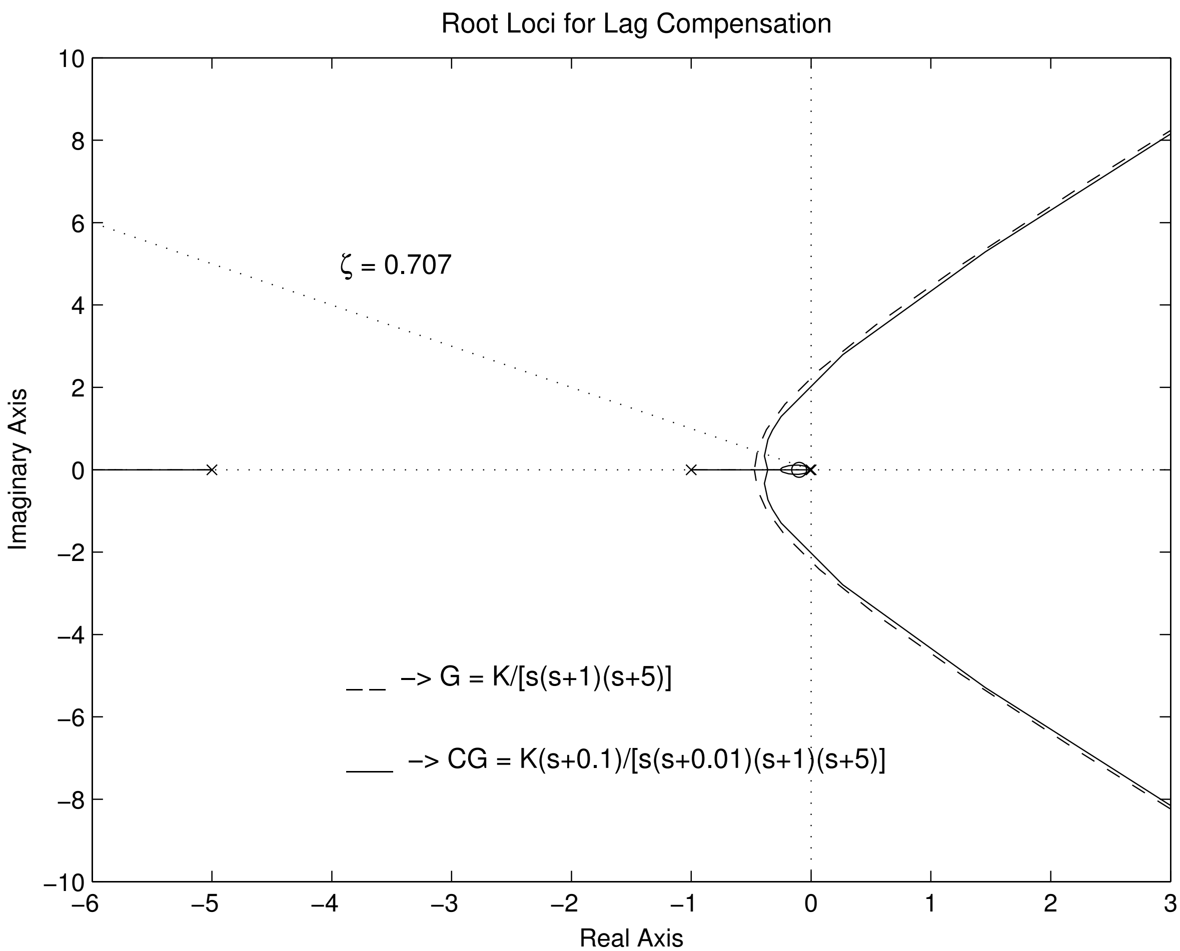
Overview of Lag Compensation (Cont'd)

A lag compensator "rule of thumb": place the lag compensator zero so that the angle from the desirable pole location to (1) the origin and (2) the lag zero is small, say 5 deg



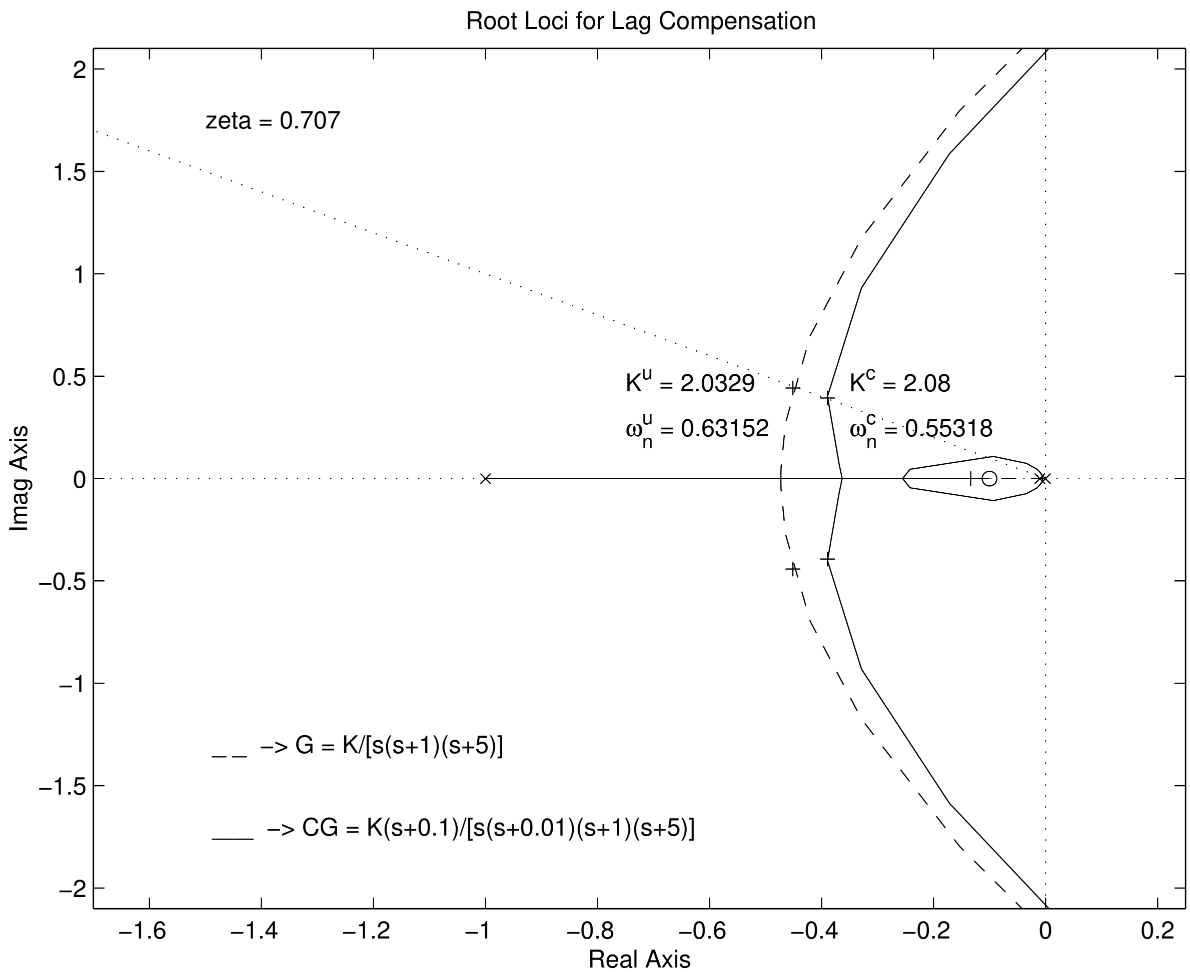
(The lag compensator pole $p_{lg} = z_{lg}/R$ is very small, so in essence it's at the origin.)

Lag Compensation via Root Locus



Big picture: The compensated root locus is not changed much in the vicinity of s_d . . .

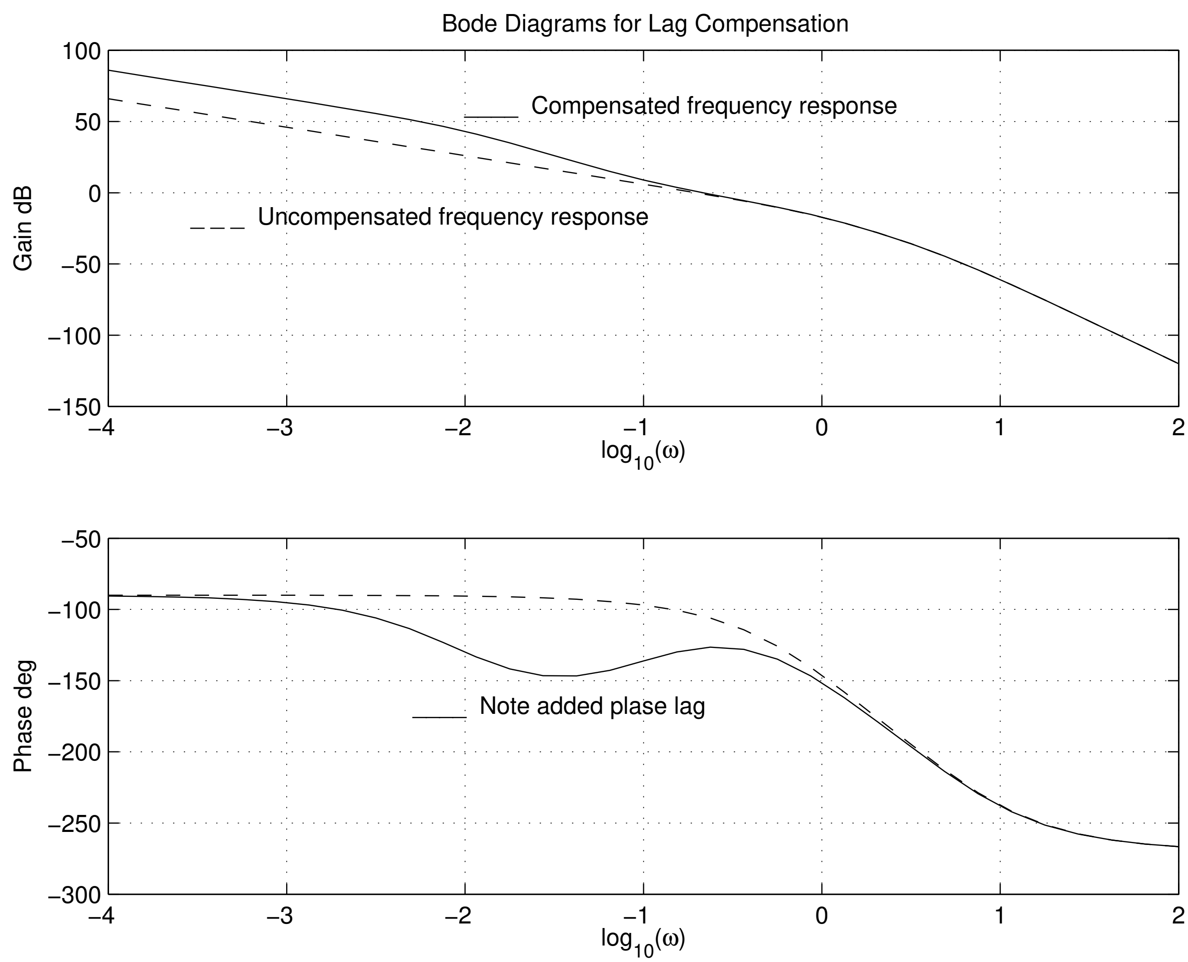
Lag Compensation Root Locus (Cont'd)



Zoomed view: The attainable closed-loop poles were OK with P control but steady-state error for a unit ramp was not – in fact, $K_v = \lim_{s\to\infty} sG_{OL}(s) = 0.407$, so $e_{ss}^{\text{ramp}} = 1/K_v = 2.46$. We can use lag compensation to increase K_v; if we place the zero/pole pair using $R = 10$ (to decrease e_{ss}^{ramp} 90 %) and put the zero at about $\omega_n^u/6$, so the original root locus is "not disturbed very much", we should do well . . .

Frequency-Domain Lag Compensation

The alternative view:



Note: The low-frequency gain is 20 dB higher at low frequencies; the compensator p/z pair is set so that at mid frequency (near the phase crossover) the effect on plant magnitude and phase is slight.
Rule of Thumb: Place the lag compensator zero at $\omega_{co}/10$; that will change the phase at ω_{co} by a little less than 6 deg.

Lag Compensation – Time-domain Response

Finally, here's the performance test: steady-state error for a *ramp* input is reduced 90 %