

MEM 355 Performance Enhancement of Dynamical Systems

Root Locus Design

Harry G. Kwatny

Department of Mechanical Engineering &
Mechanics

Drexel University



Outline

The root locus design method is an iterative, graphical procedure for selecting and tuning compensators.

- Basic Compensator Types
- The Root Locus Design process
- Example: BWR Pressure Control
- Hard Control Problems – a brief introduction

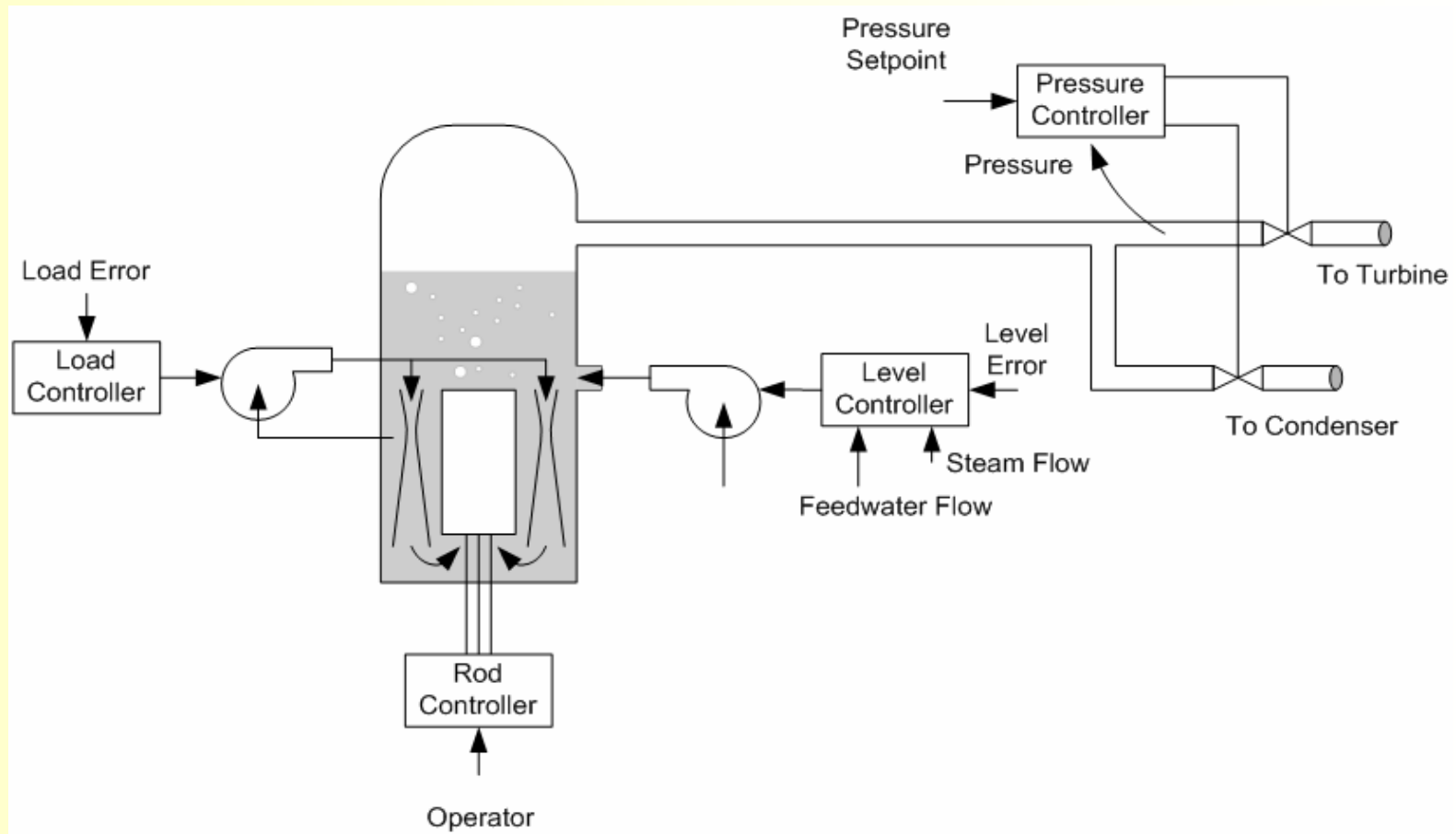
Basic Compensators

$G_c(s)$	Name	Effect on Ultimate State Error	Effect on Stability
K	P (uncompensated)		
$K \frac{s + \alpha}{s}$	PI	Improves	Degrades
$K \frac{s^2 + \alpha_1 s + \alpha_0}{s}$	PID	Improves	Improves somewhat
$K \frac{s + \alpha}{s + \beta}, \alpha > \beta$	lag	Improves somewhat	Degrades somewhat
$K \frac{s + \alpha}{s + \beta}, \alpha < \beta$	lead	Degrades somewhat	Improves somewhat
$K(s + \alpha)$	Rate feedback (PD)	Degrades	Improves
$K \frac{s^2 + 2\rho_1\omega_1s + \omega_1^2}{s^2 + 2\rho_2\omega_2s + \omega_2^2}$	Notch		Neutralizes plant resonance

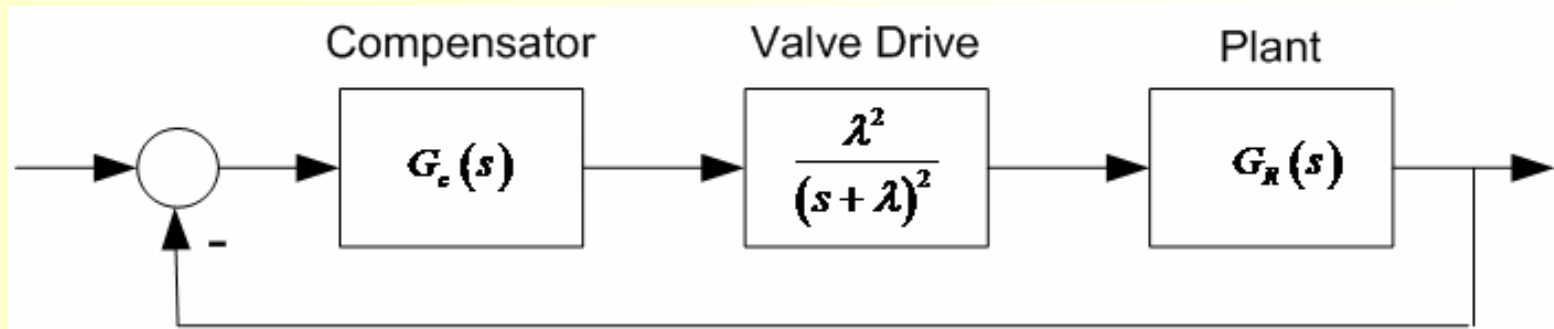
Procedure

- 1) Uncompensated system
 - Root locus
 - Ultimate state error/step response
- 2) Compensated system
 - Choose/modify compensator
 - Root locus
 - Ultimate state error/step
- 3) Repeat step 2)

Example: BWR Pressure Controller



BWR Pressure Control Model



$$G_R(s) = \gamma \underbrace{\frac{(s - z_r)}{(s - p_r)}}_{\substack{\text{Reactor} \\ \text{Pipe} \\ \text{Dipole}}} \underbrace{\frac{(s - z_p)}{(s - p_p)}}_{\substack{\text{Pipe} \\ \text{Dipole}}} \underbrace{\left[\frac{s^2 + 2\xi_1 v_1 s + v_1^2}{s^2 + 2\rho_1 \omega_1 s + \omega_1^2} \right] \dots \left[\frac{s^2 + 2\xi_N v_N s + v_N^2}{s^2 + 2\rho_N \omega_N s + \omega_N^2} \right]}_{\text{Acoustics (first N harmonics)}}$$

BWR Transfer Functions

$$G_p(s) = 15 \left(\frac{25}{(s+5)^2} \right) \left(\frac{s+0.05}{s+0.1} \right) \left(\frac{s+0.025}{s+4} \right) \left(\frac{s^2 + 2(0.05)12s + 144}{s^2 + 2(0.25)8s + 64} \right)$$

$$\text{PI } G_c(s) = K \frac{s+2}{s}$$

$$\text{PI plus Notch } G_c(s) = K \frac{s+2}{s} \frac{s^2 + 2(0.15)8s + 64}{(s+8)^2}$$

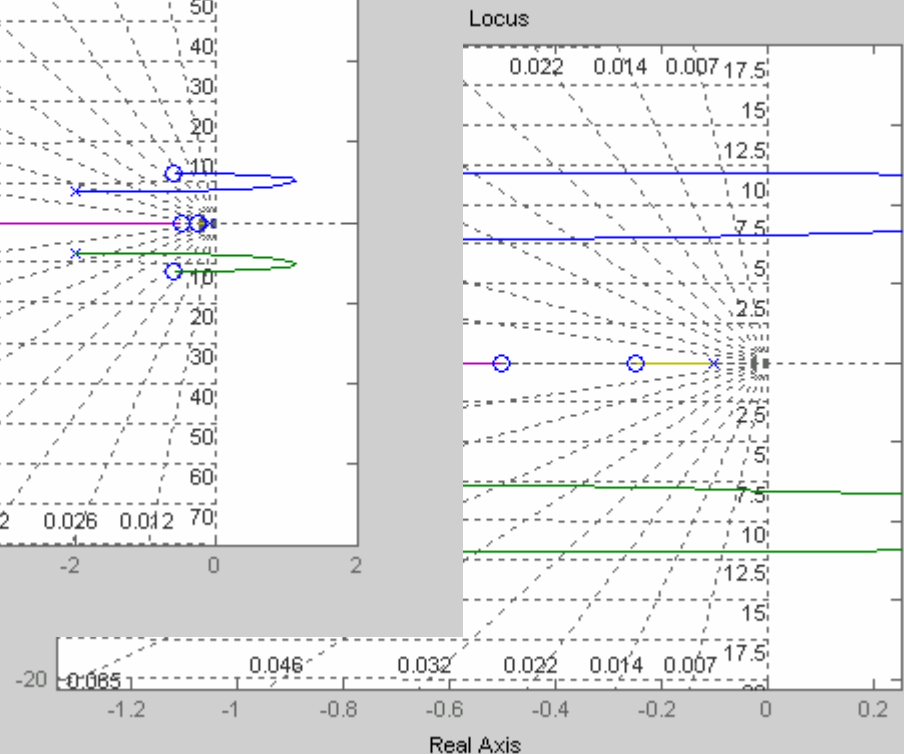
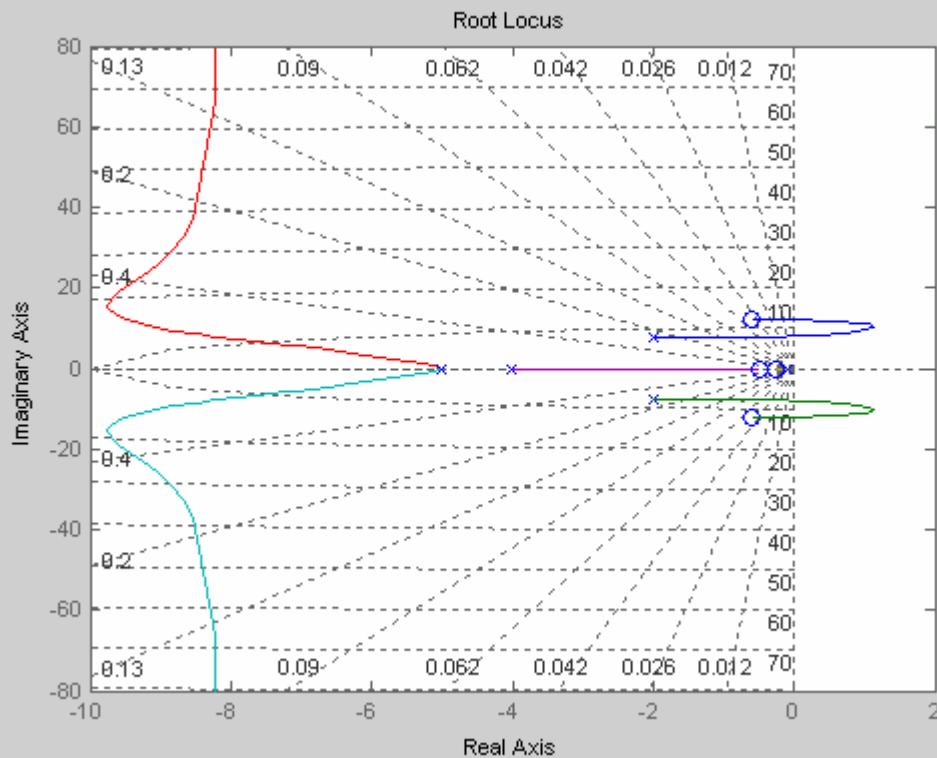
Uncompensated (P)

```
>> s=tf('s');
>> Gp=15*(25/(s+5)^2)*((s+0.5)/(s+0.1))*((s+0.25)/(s+4))*((s^2+2*.05*12*s+144)/(s^2+2*.25*8*s+64));
>> rlocus(Gp)
>> sgrid
>> [K,Poles]=rlocfind(G)
Select a point in the graphics window
sgrid
selected_point =
  -1.1137 + 7.2050i
K =
  0.0324
Poles =
  -7.0808 + 5.5584i
  -7.0808 - 5.5584i
  -1.1219 + 7.5434i
  -1.1219 - 7.5434i
  -1.5793
  -0.1154
>> Gce=1/(1+K*Gp)
```

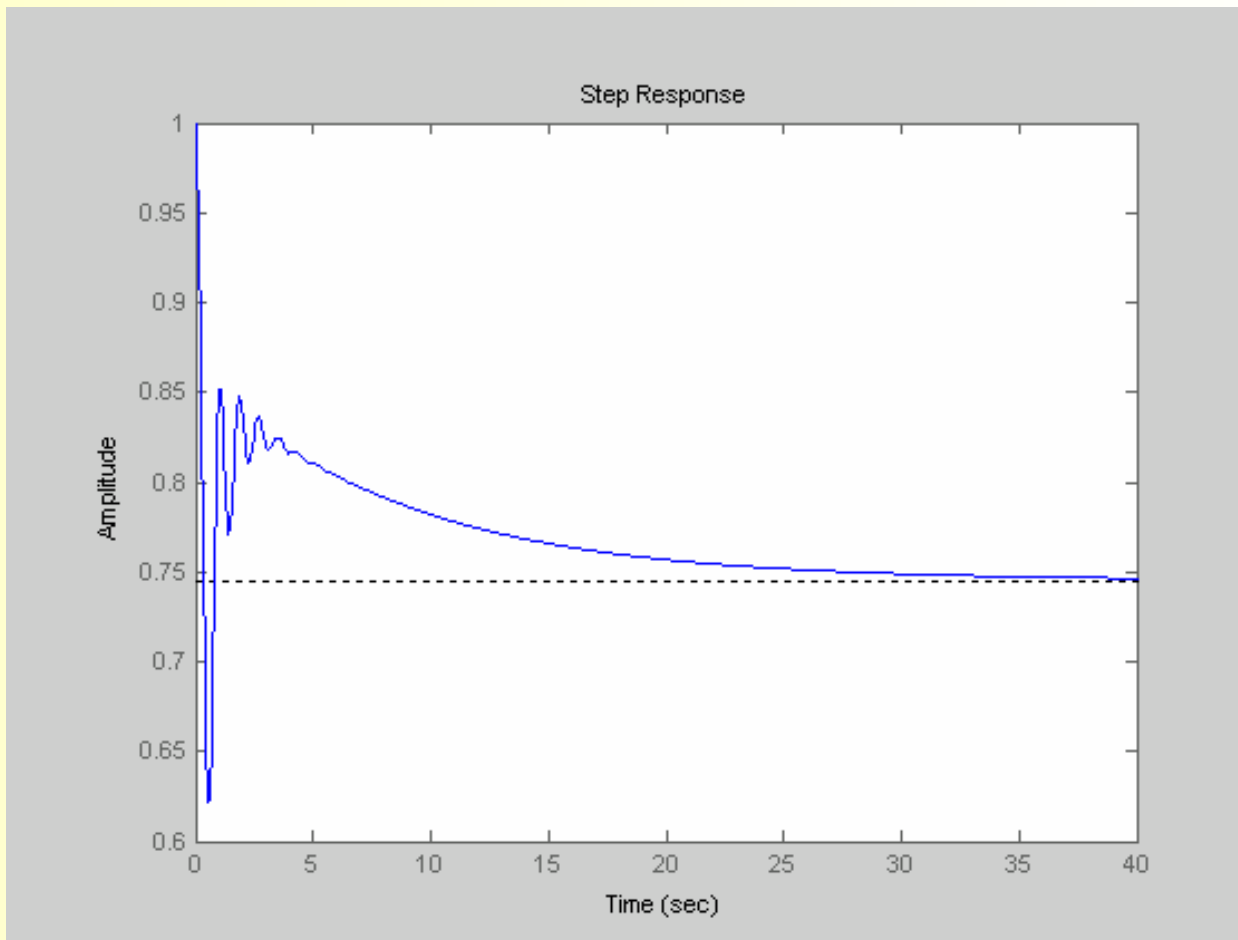


Uncompensated (P)

$$\lim_{s \rightarrow 0} sG_e(s) \frac{1}{s} = 0.7452$$



Uncompensated (P)



PI

Stability limit

```
>> [K,Poles]=rlocfind(G)
```

Select a point in the graphics window
selected_point =

-0.0047 + 7.6087i

K =

0.0861

Poles =

-7.6539 + 7.6078i

-7.6539 - 7.6078i

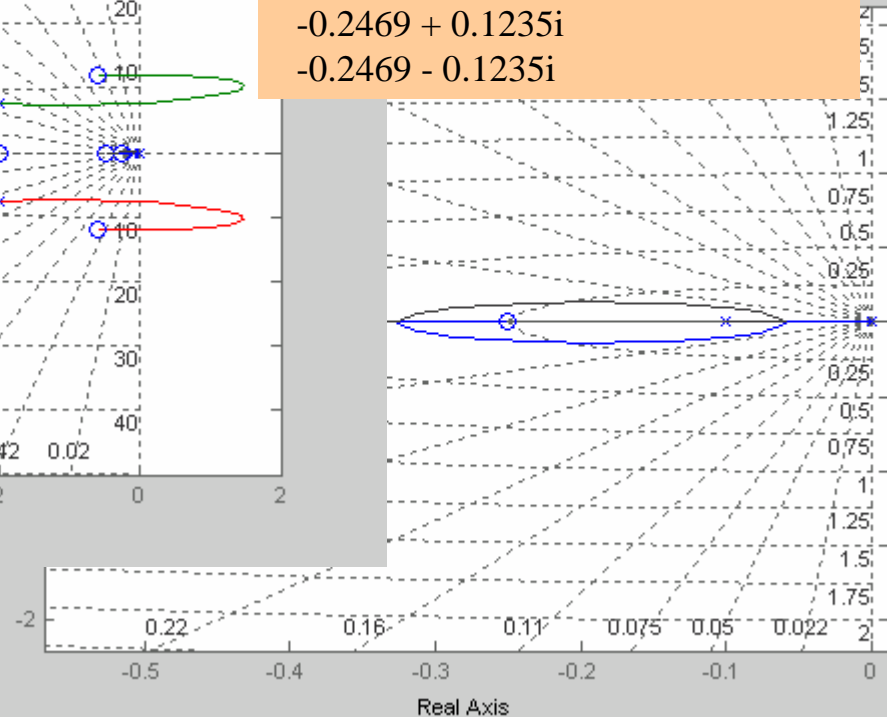
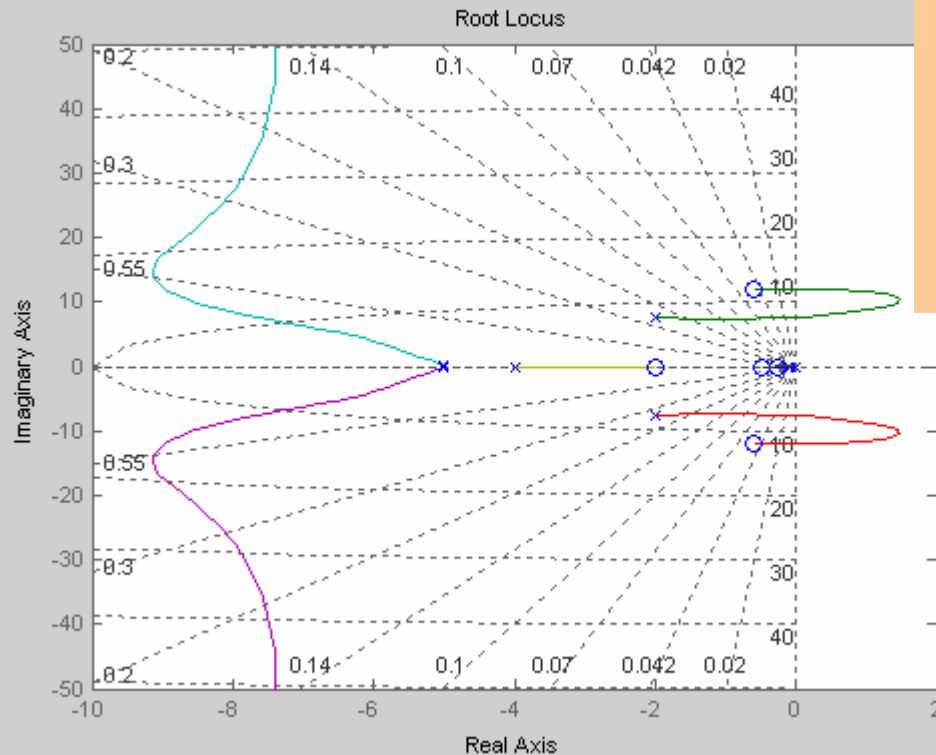
-0.0307 + 7.6511i

-0.0307 - 7.6511i

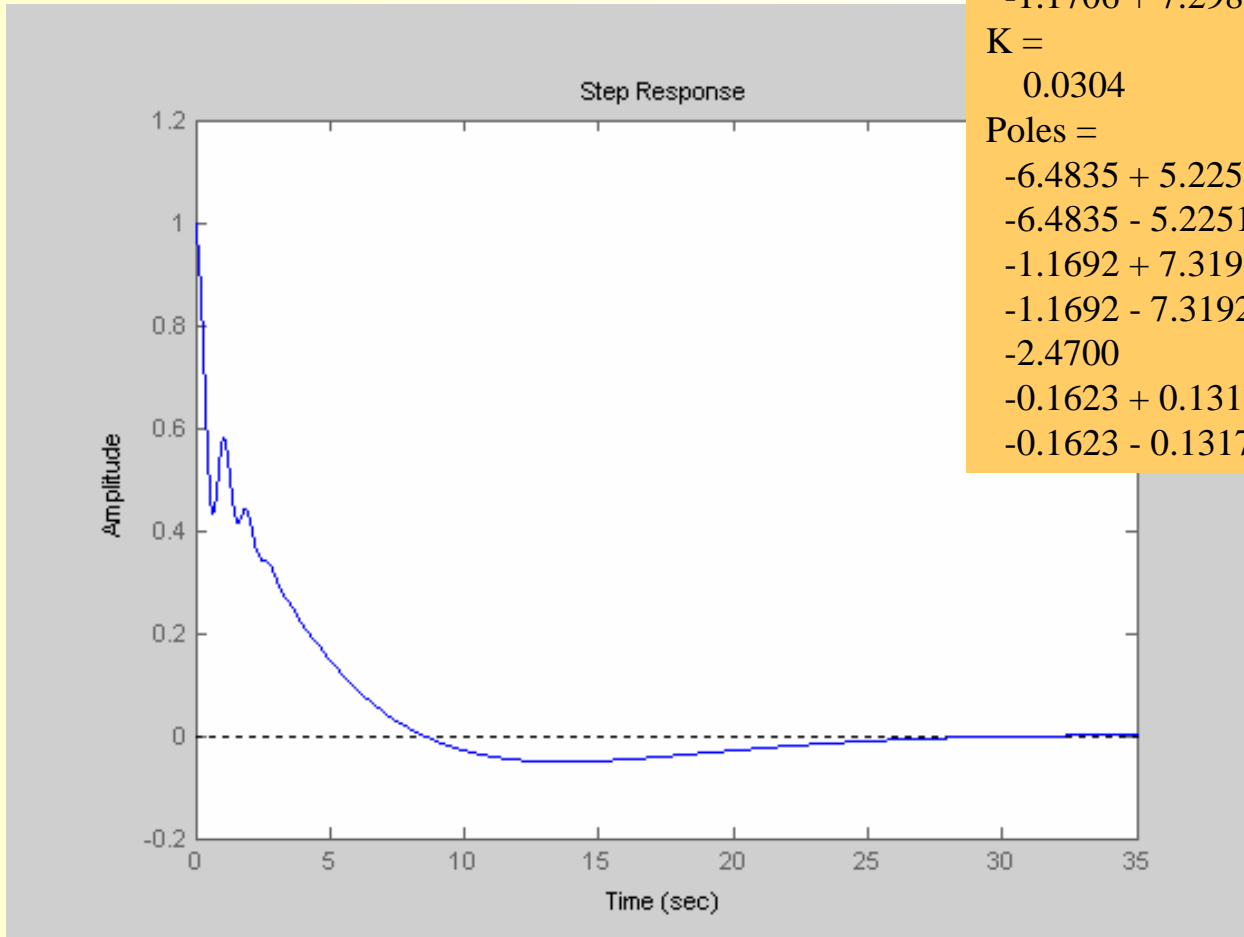
-2.2370

-0.2469 + 0.1235i

-0.2469 - 0.1235i

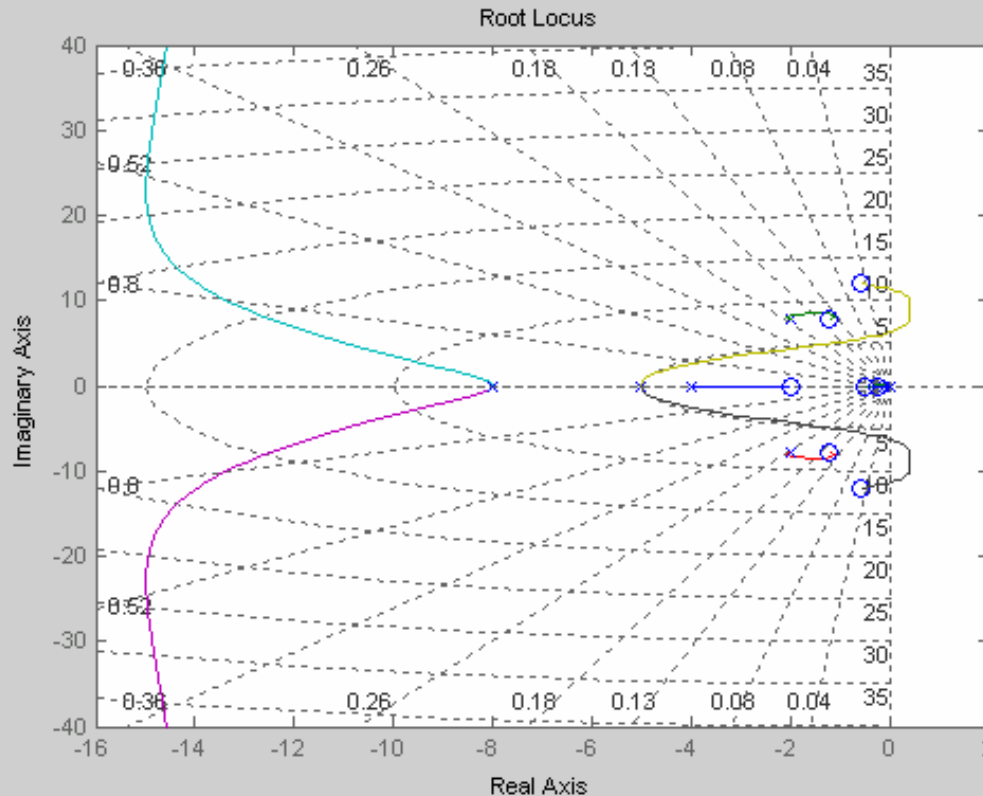


PI



```
>> [K,Poles]=rlocfind(G)
Select a point in the graphics window
selected_point =
    -1.1706 + 7.2981i
K =
    0.0304
Poles =
    -6.4835 + 5.2251i
    -6.4835 - 5.2251i
    -1.1692 + 7.3192i
    -1.1692 - 7.3192i
    -2.4700
    -0.1623 + 0.1317i
    -0.1623 - 0.1317i
```

PI + Notch



```
>> [K,Poles]=rlocfind(G)
```

Select a point in the graphics window

selected_point =

-2.7133 + 3.6025i

K =

0.0149

Poles =

-10.8605 + 4.7906i

-10.8605 - 4.7906i

-2.0317 + 7.7960i

-2.0317 - 7.7960i

-2.8100 + 3.7251i

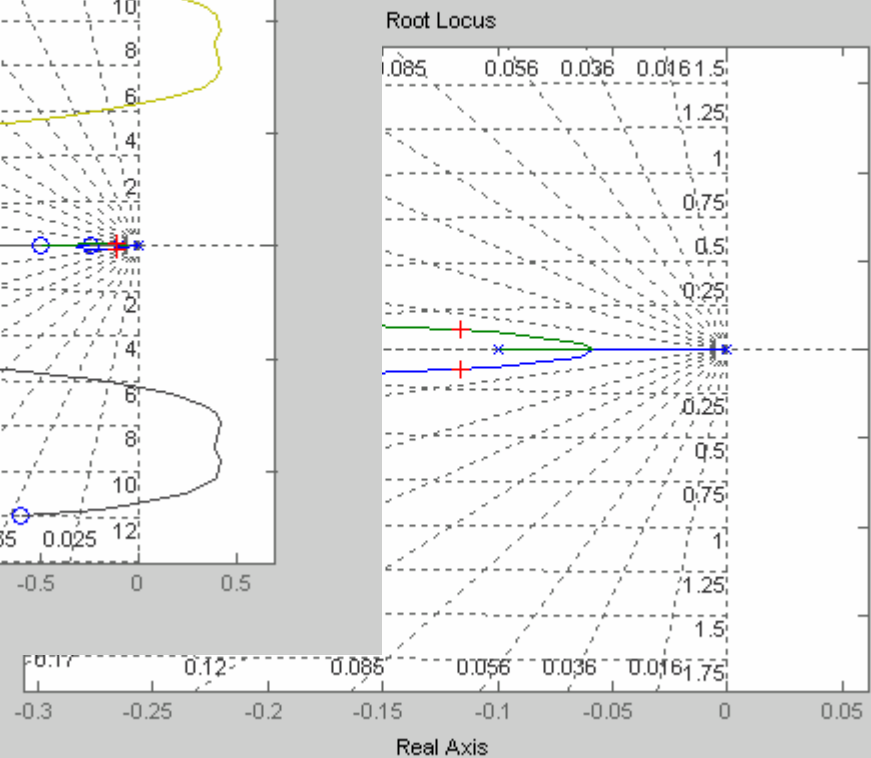
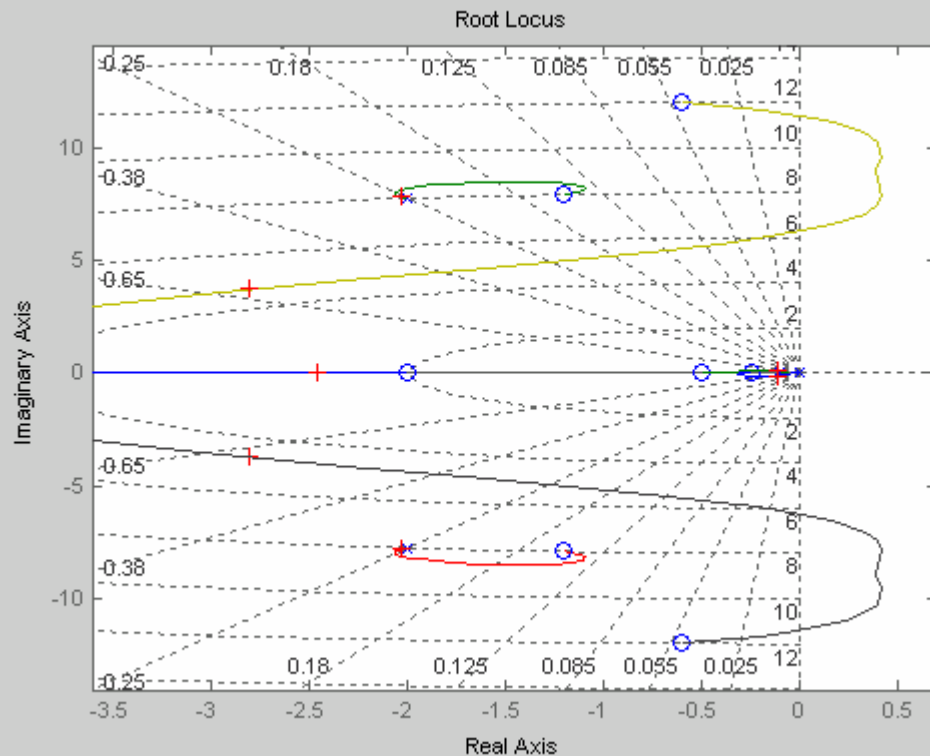
-2.8100 - 3.7251i

-2.4630

-0.1163 + 0.1131i

-0.1163 - 0.1131i

PI + Notch



PI + Notch

```
>> [K,Poles]=rlocfind(G)
```

Select a point in the graphics window

```
selected_point =
```

```
-3.3104 + 3.3540i
```

```
K =
```

```
0.0083
```

```
Poles =
```

```
-10.3532 + 4.0115i
```

```
-10.3532 - 4.0115i
```

```
-2.0187 + 7.7728i
```

```
-2.0187 - 7.7728i
```

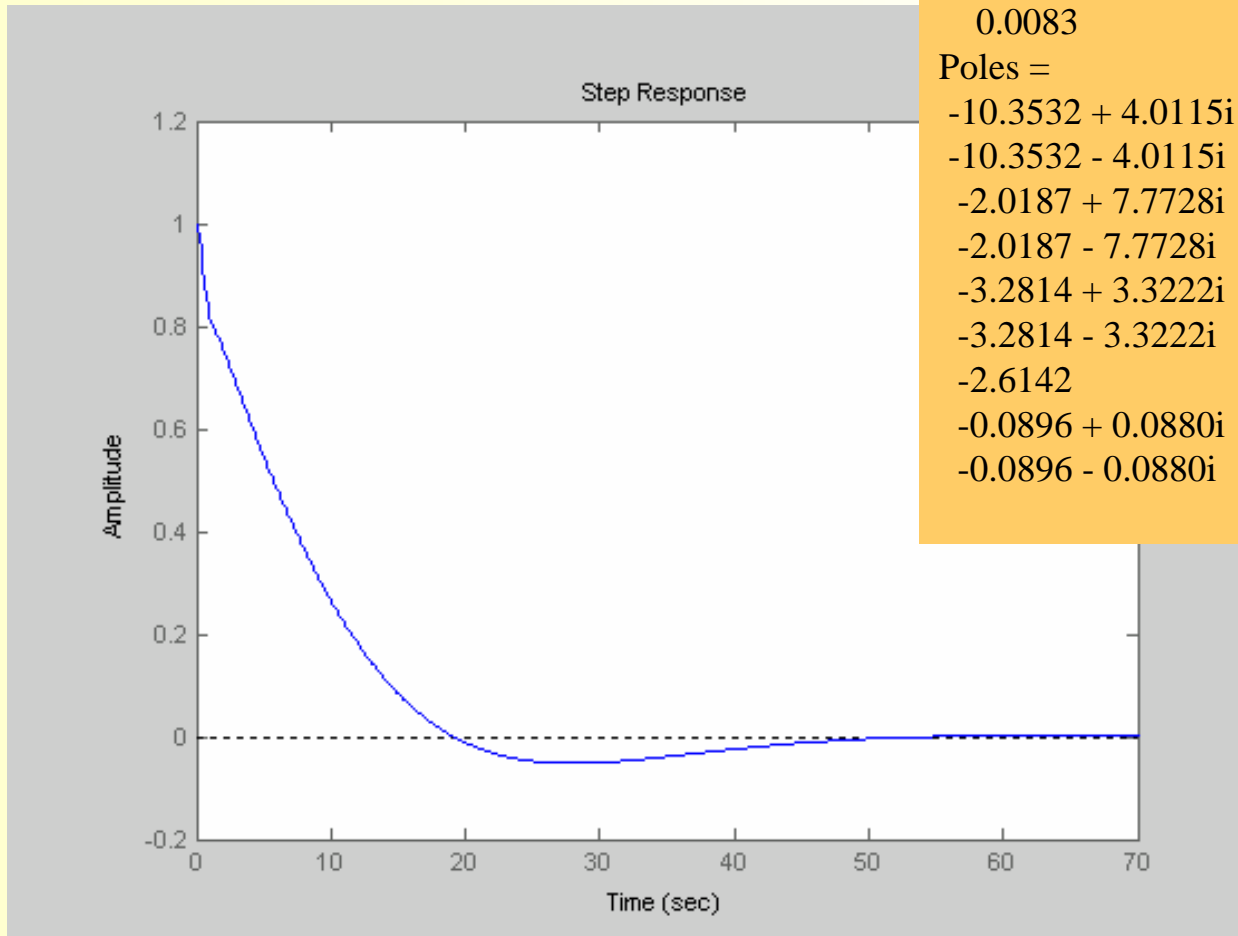
```
-3.2814 + 3.3222i
```

```
-3.2814 - 3.3222i
```

```
-2.6142
```

```
-0.0896 + 0.0880i
```

```
-0.0896 - 0.0880i
```



Hard Control Problems

Control design problems can be difficult for many reasons:

Demanding specifications

Nonlinearity

Actuator/sensor constraints

Coupling

Implementation constraints

to name a few. But here are some deceptively simple issues involving linear SISO systems that can give a designer headaches.

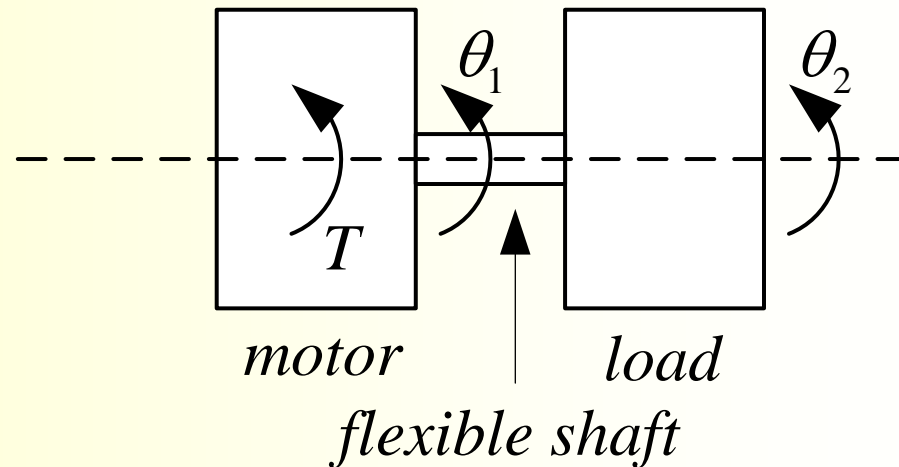
- Non-collocated
 - Nuclear plant pressure control
 - Large spacecraft attitude
 - Flexible rocket attitude
- Right half plane zero
 - Most tail lift aircraft (space shuttle)
 - Steam plant level control
 - Bicycle with front wheel steering
- Right half plane pole
 - Rotorcraft
 - High performance aircraft (F-16)
 - Some high performance ground vehicles
 - Most missiles
- Both RHF pole and zero
 - Some aircraft (X-29)
 - Bicycle with rear wheel steering
 - Inverted pendulum



Example: Collocated vs Non-collocated

The goal is to position the load.

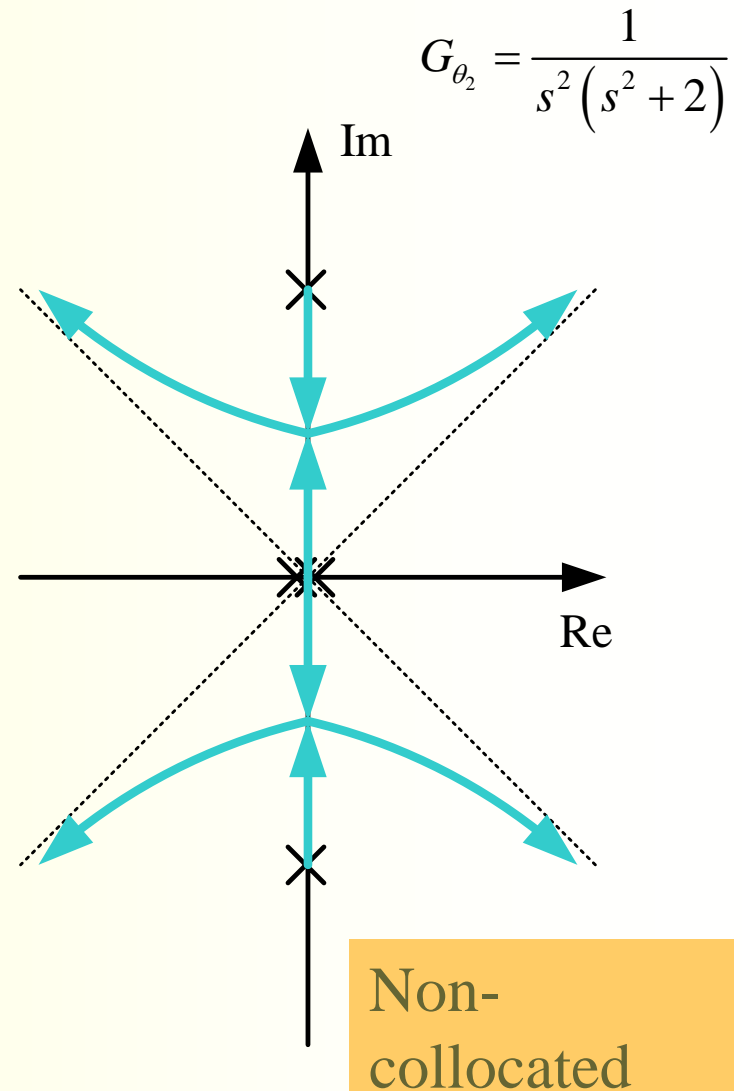
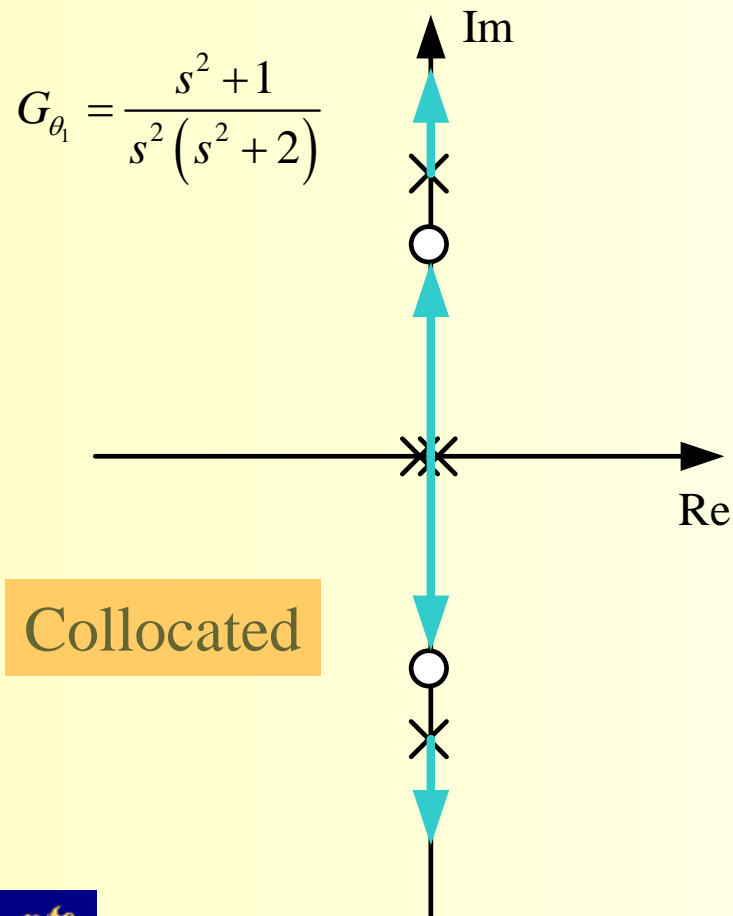
A position sensor can be placed on motor or load.



$$\Theta_1 = \frac{1}{J_1} \frac{s^2 + k/J_2}{s^2 \left(s^2 + \frac{(J_1 + J_2)k}{J_1 J_2} \right)} T$$
$$\Theta_2 = \frac{k}{J_1 J_2} \frac{1}{s^2 \left(s^2 + \frac{(J_1 + J_2)k}{J_1 J_2} \right)} T$$

$$J_1 = J_2 = 1, k = 1$$

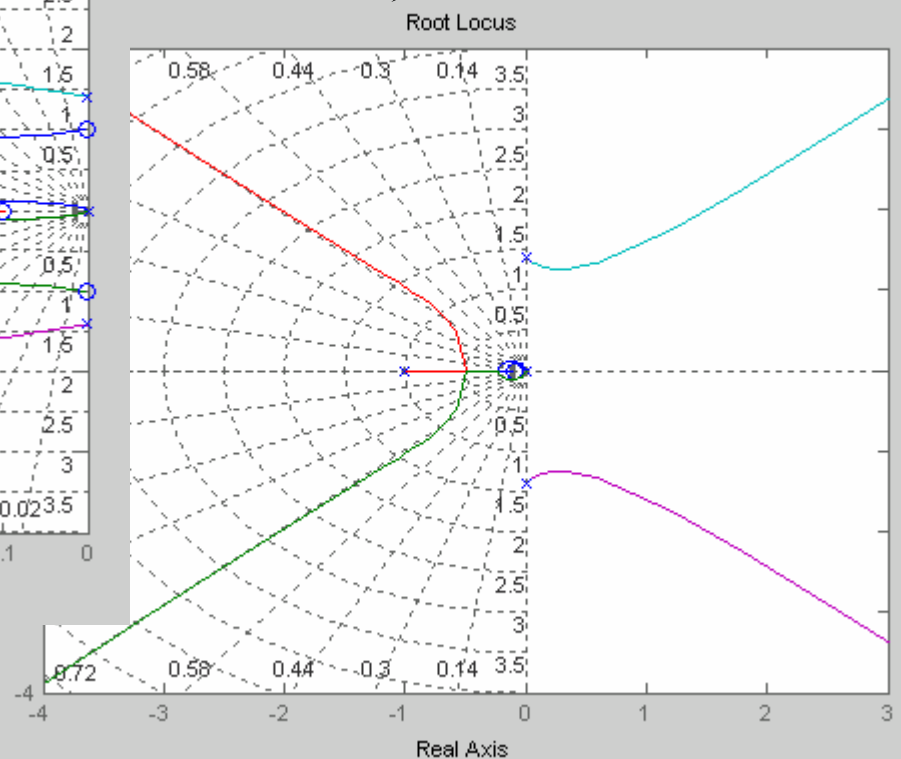
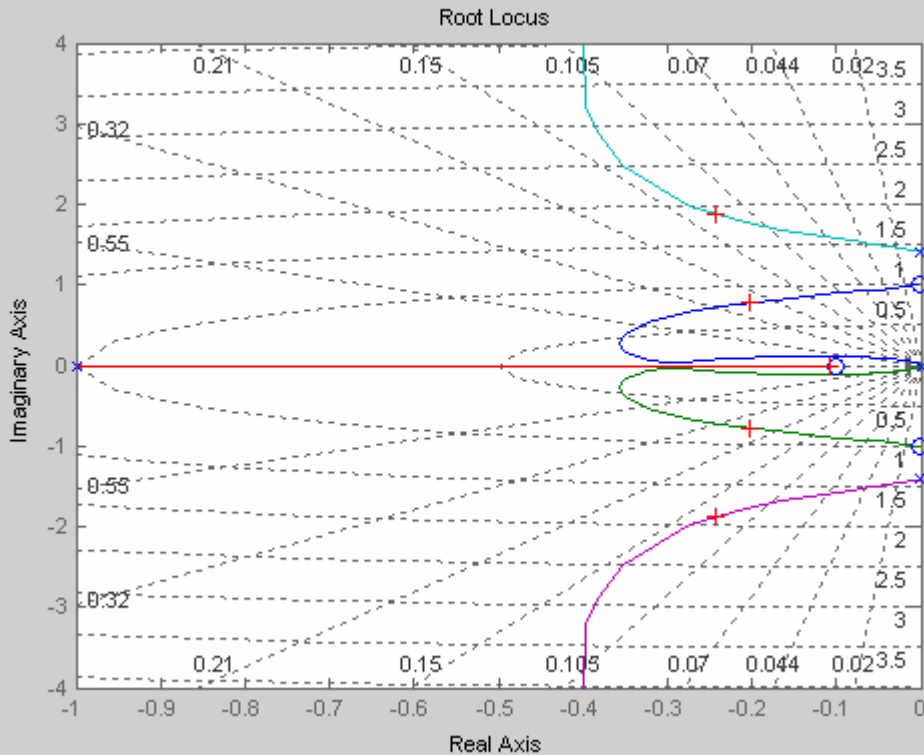
Example Cont'd



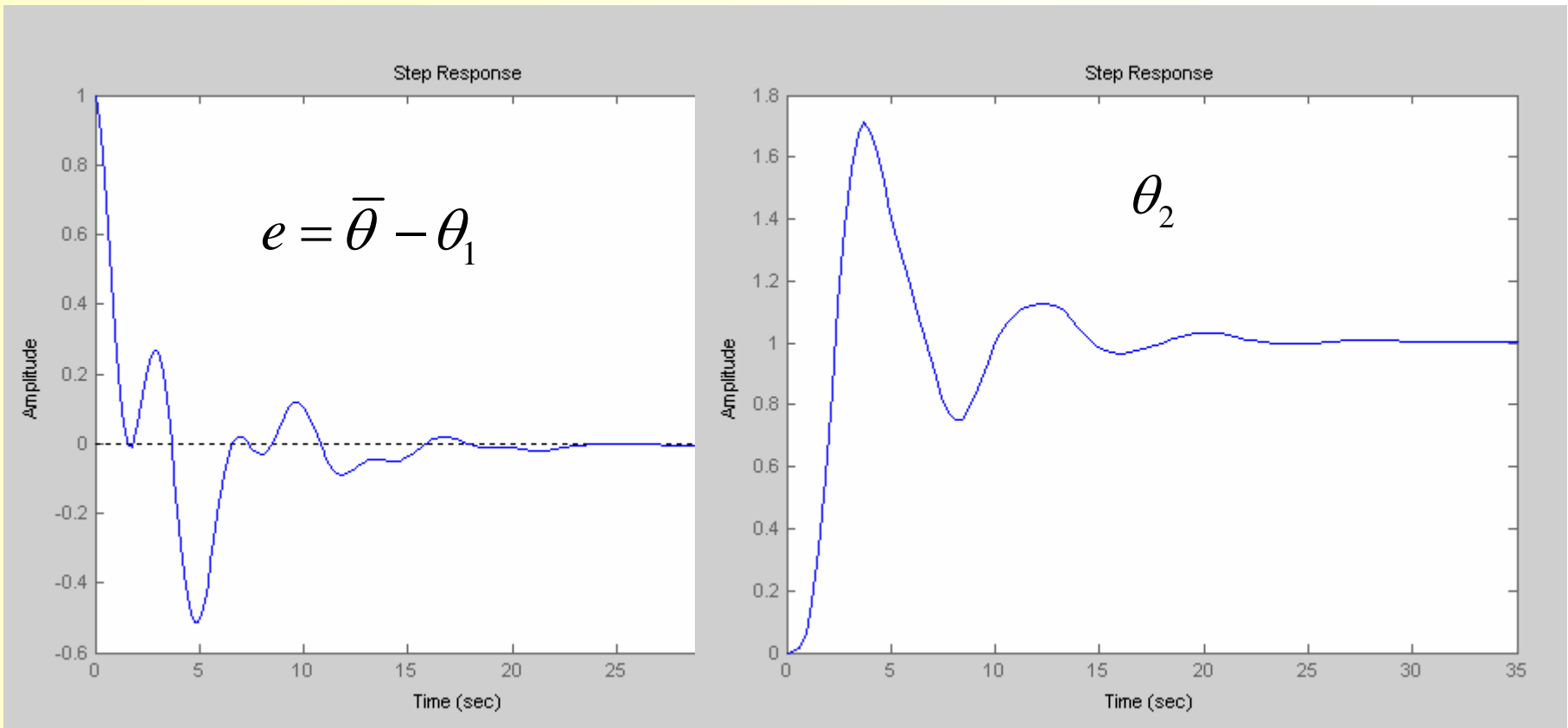
Lead Compensation

Collocated

Non-collocated



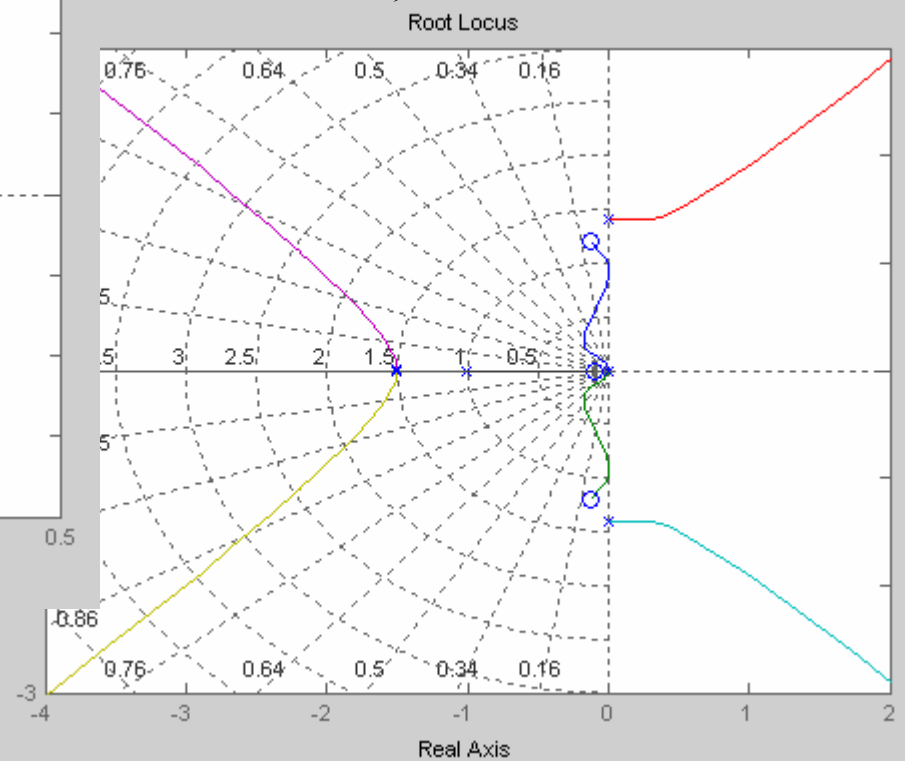
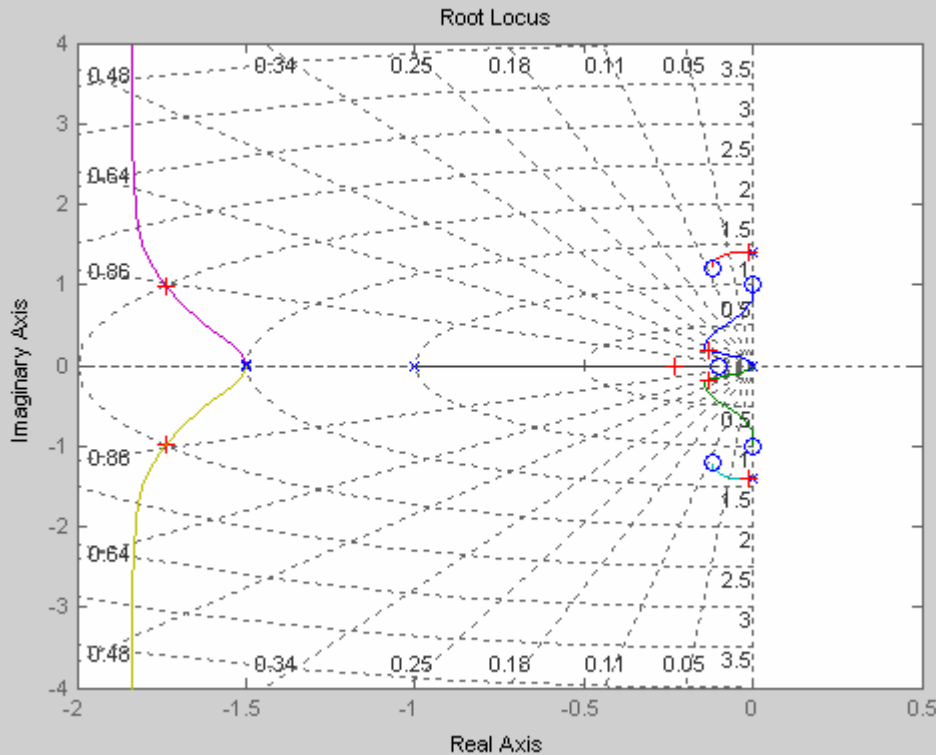
Lead Response (collocated case)



Lead + Notch

Collocated

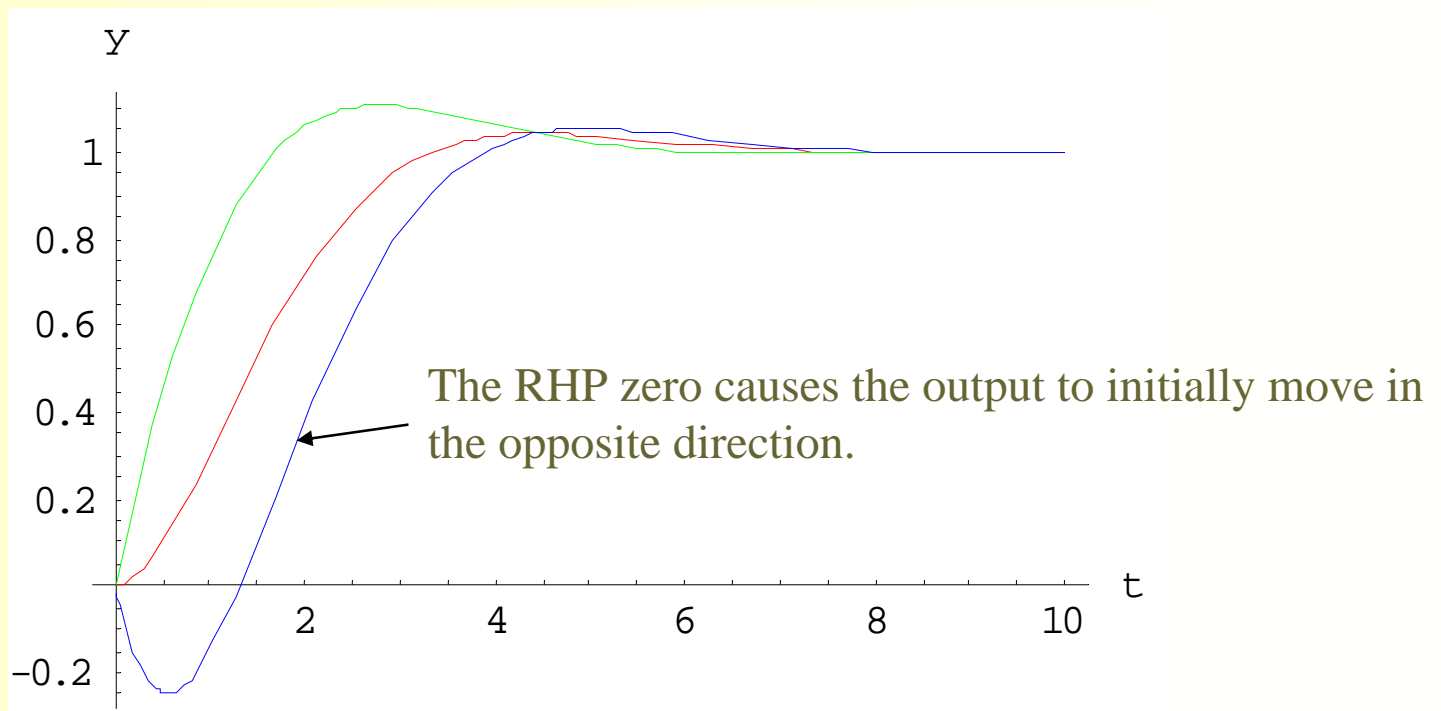
Non-collocated



Right Half Plane Zero

First, let's look at the effect of a zero on the step response:

$$G(s) = \frac{1}{s^2 + \sqrt{2}s + 1} \quad G(s) = \frac{s+1}{s^2 + \sqrt{2}s + 1} \quad G(s) = -\frac{s-1}{s^2 + \sqrt{2}s + 1}$$



Example: Simple Nonminimum Phase System

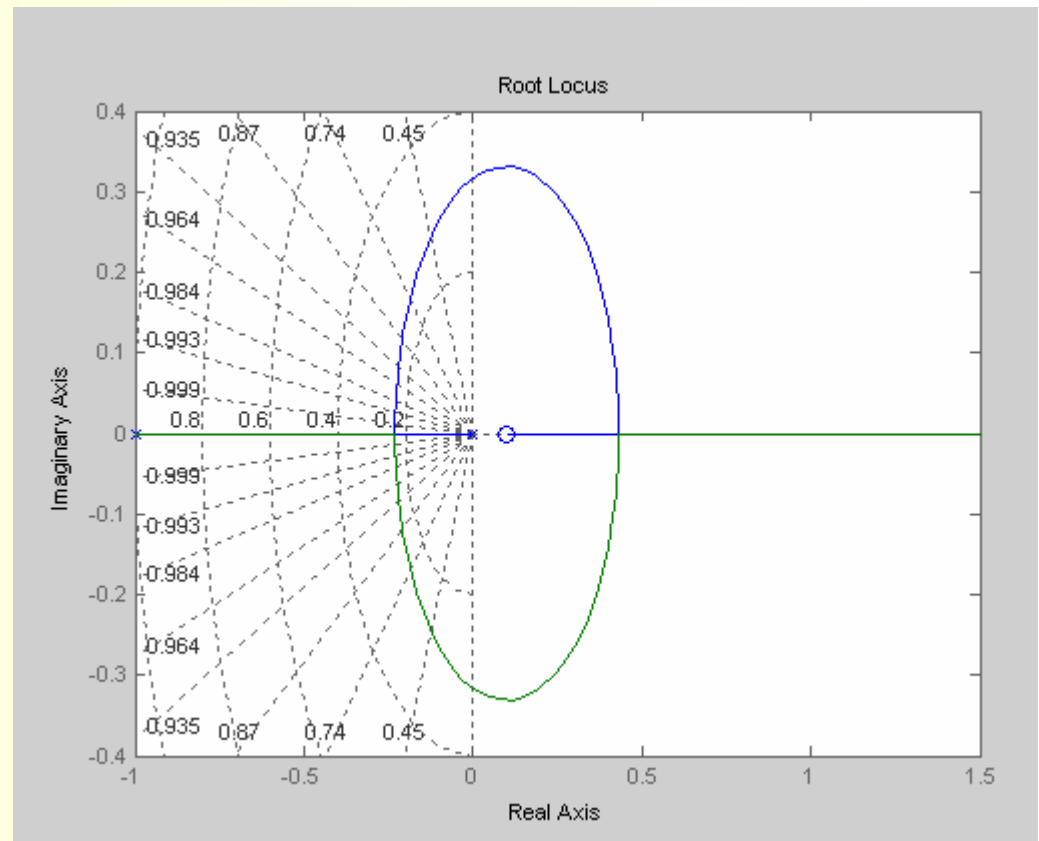
Uncompensated

Consider the process

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

Try a PD compensator

$$G_c(s) = K(s + 1.1)$$



Comments on Positive Feedback

Root Locus

Interpreting the root locus presents a problem because the system involves positive feedback (equivalently, negative K when using a negative feedback loop). This is because of the negative sign in the transfer function. Note that computing the root locus with MATLAB does not present a problem.

Positive feedback (or $K < 0$) root locus rules are slightly different from the negative feedback rules. Here are the two changes.

4. Real-axis segments: For $K < 0$, real axis segments to the left of an **even** number of finite real axis poles and/or zeros are part of the root locus.



Positive Feedback Root Locus, Cont'd

5. Behavior at infinity: The root locus approaches infinity along asymptotes with angles:

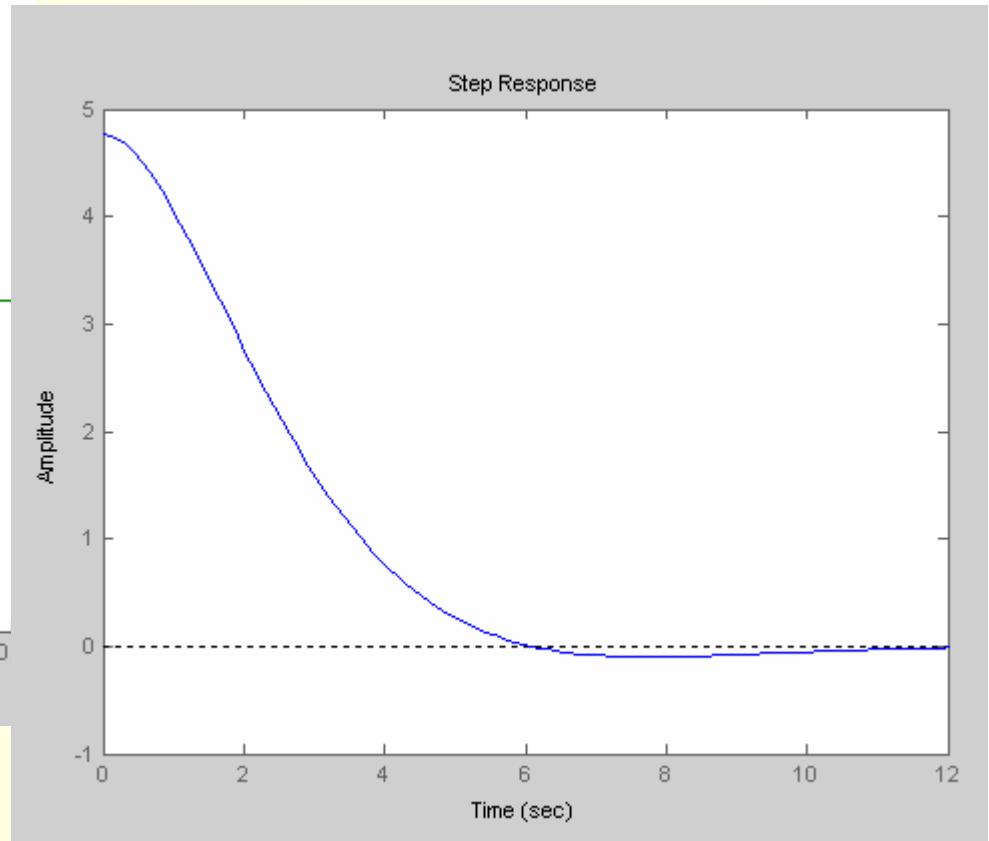
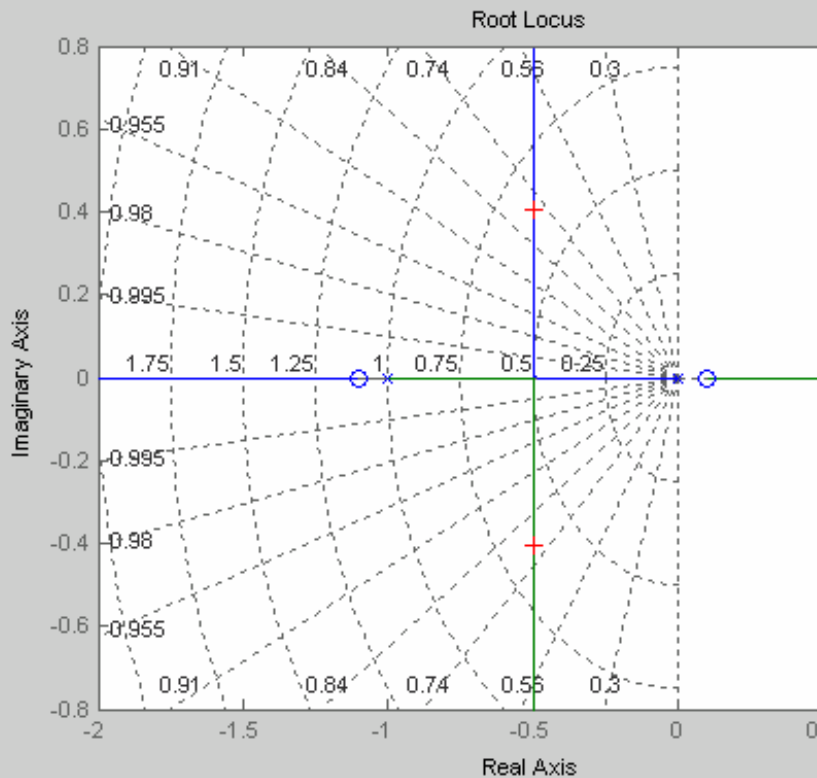
$$\theta = \frac{2k\pi}{\# \text{ finite poles} - \# \text{ finite zeros}}, k = 0, \pm 1, \pm 2, \pm 3, \dots$$

Furthermore, these asymptotes intersect the real axis at a common point given by

$$\sigma = \frac{\sum \text{ finite poles} - \sum \text{ finite zeros}}{\# \text{ finite poles} - \# \text{ finite zeros}}$$

Example with PD

Poles look decent, but look at the initial error.



Example: Check Initial Error

$$G_e = \frac{1}{1+G} \Rightarrow \frac{1}{1+K(s+1.1)\frac{-10s+1}{s(s+1)}}$$

$$G_e = \frac{s(s+1)}{(s^2+s)+K(s+1.1)(-10s+1)} = \frac{s(s+1)}{(1-10K)s^2+(1-10K)s+1.1K}$$

$K = 0.0790$ (from MATLAB)

$$G_e = \frac{s(s+1)}{0.2099s^2+0.2099s+0.08691}$$

$$e(0^+) = \lim_{s \rightarrow \infty} sG_e(s) \frac{1}{s} \rightarrow \frac{1}{0.2099} = 4.7642$$

Initial Value Theorem

F-16, X-29

$$G_p(s) = 1.645 \frac{s(s + 0.0423101)(s + 0.586543)}{(s - 0.730937)(s + 1.7036)(s^2 + 0.0876334s + 0.044546)}$$

Unstable Pole



$$G_p(s) = G(s) \frac{s - 26}{s - 6}$$

RHP pole-zero pair

Minimum phase part