Topic #15

16.30/31 Feedback Control Systems

State-Space Systems

- Closed-loop control using estimators and regulators.
- Dynamics output feedback

- "Back to reality"

- Reading: FPE 7.6

Combined Estimators and Regulators

- Now evaluate stability and/or performance of a controller when K **designed** assuming

$$\mathbf{u}(t) = -K\mathbf{x}(t)$$

but **implemented** as

$$\mathbf{u}(t) = -K\hat{\mathbf{x}}(t)$$

- Assume we have designed a closed-loop estimator with gain L

$$\begin{aligned}\dot{\hat{\mathbf{x}}}(t) &= (A - LC)\hat{\mathbf{x}}(t) + B\mathbf{u}(t) + L\mathbf{y}(t) \\ \hat{\mathbf{y}}(t) &= C\hat{\mathbf{x}}(t)\end{aligned}$$

- The closed-loop system dynamics are given by:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + B\mathbf{u}(t) \\ \dot{\hat{\mathbf{x}}}(t) &= (A - LC)\hat{\mathbf{x}}(t) + B\mathbf{u}(t) + L\mathbf{y}(t) \\ \mathbf{y}(t) &= C\mathbf{x}(t) \\ \hat{\mathbf{y}}(t) &= C\hat{\mathbf{x}}(t) \\ \mathbf{u}(t) &= -K\hat{\mathbf{x}}(t)\end{aligned}$$

- Which can be compactly written as:

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\hat{\mathbf{x}}}(t) \end{bmatrix} = \begin{bmatrix} A & -BK \\ LC & A - BK - LC \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \hat{\mathbf{x}}(t) \end{bmatrix} \quad \Rightarrow \quad \dot{\mathbf{x}}_{cl}(t) = A_{cl}\mathbf{x}_{cl}(t)$$

- This does not look too good at this point – not even obvious that the closed-system is stable.

$$\lambda_i(A_{cl}) = ??$$

- Can fix this problem by introducing a new variable $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$ and then converting the closed-loop system dynamics using the *similarity transformation* T

$$\tilde{\mathbf{x}}_{cl}(t) \triangleq \begin{bmatrix} \mathbf{x}(t) \\ \tilde{\mathbf{x}}(t) \end{bmatrix} = \begin{bmatrix} I & 0 \\ I & -I \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \hat{\mathbf{x}}(t) \end{bmatrix} = T\mathbf{x}_{cl}(t)$$

  - Note that $T = T^{-1}$

- Now rewrite the system dynamics in terms of the state $\tilde{\mathbf{x}}_{cl}(t)$

$$A_{cl} \quad \Rightarrow \quad TA_{cl}T^{-1} \triangleq \bar{A}_{cl}$$

  - Since similarity transformations preserve the eigenvalues we are guaranteed that
$$\lambda_i(A_{cl}) \equiv \lambda_i(\bar{A}_{cl})$$

- Work through the math:

$$\begin{aligned} \bar{A}_{cl} &= \begin{bmatrix} I & 0 \\ I & -I \end{bmatrix} \begin{bmatrix} A & -BK \\ LC & A - BK - LC \end{bmatrix} \begin{bmatrix} I & 0 \\ I & -I \end{bmatrix} \\ &= \begin{bmatrix} A & -BK \\ A - LC & -A + LC \end{bmatrix} \begin{bmatrix} I & 0 \\ I & -I \end{bmatrix} \\ &= \begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix} \end{aligned}$$

- Because $\bar{A}_{cl}$ is block upper triangular, we know that the closed-loop poles of the system are given by

$$\det(\mathbf{sI} - \bar{\mathbf{A}}_{\mathbf{cl}}) \triangleq \det(\mathbf{sI} - (\mathbf{A} - \mathbf{BK})) \cdot \det(\mathbf{sI} - (\mathbf{A} - \mathbf{LC})) = \mathbf{0}$$

- **Observation: Closed-loop poles for this system consist of the union of the regulator poles and estimator poles.**

- So we can just design the estimator/regulator **separately** and combine them at the end.
  - Called the **Separation Principle**.
  - Keep in mind that the pole locations you are picking for these two sub-problems will also be the closed-loop pole locations.

- **Note:** Separation principle means that there will be **no** ambiguity or uncertainty about the stability and/or performance of the closed-loop system.
  - The closed-loop poles will be exactly where you put them!!
  - And we have not even said what compensator does this amazing accomplishment!!!

The Compensator

- **Dynamic Output Feedback Compensator** is the combination of the regulator and estimator using $\mathbf{u}(t) = -K\hat{\mathbf{x}}(t)$

$$\begin{aligned} \dot{\hat{\mathbf{x}}}(t) &= (A - LC)\hat{\mathbf{x}}(t) + B\mathbf{u}(t) + L\mathbf{y}(t) \\ \Rightarrow \quad \dot{\hat{\mathbf{x}}}(t) &= (A - BK - LC)\hat{\mathbf{x}}(t) + L\mathbf{y}(t) \\ \mathbf{u}(t) &= -K\hat{\mathbf{x}}(t) \end{aligned}$$

- Rewrite with new state $\mathbf{x}_c(t) \equiv \hat{\mathbf{x}}(t)$

$$\begin{aligned} \dot{\mathbf{x}}_c(t) &= A_c\mathbf{x}_c(t) + B_c\mathbf{y}(t) \\ \mathbf{u}(t) &= -C_c\mathbf{x}_c(t) \end{aligned}$$

where the **compensator dynamics** are given by:

$$A_c \triangleq A - BK - LC \ , \quad B_c \triangleq L \ , \quad C_c \triangleq K$$

  - Note that the compensator maps *sensor measurements* to *actuator commands*, as expected.

- Closed-loop system stable if regulator/estimator poles placed in the LHP, but compensator dynamics do not need to be stable.

$$\lambda_i(A - BK - LC) = ??$$

- For consistency with the implementation of classical controllers, define the **compensator transfer function** as

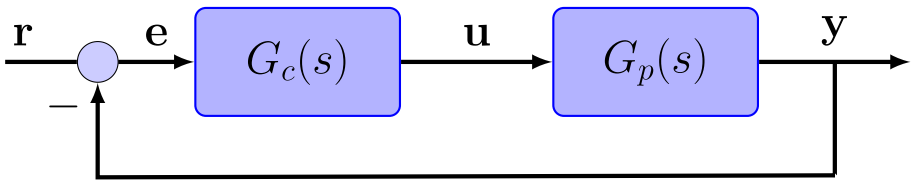
$$G_c(s) = C_c(sI - A_c)^{-1}B_c$$

so that

$$\mathbf{u}(t) = -G_c(s)\mathbf{y}(t)$$

- Note that it is often very easy to provide classical interpretations (such as lead/lag) for the compensator $G_c(s)$.

- Can implement this compensator with a reference command $\mathbf{r}(t)$ using feedback on $\mathbf{e}(t) = \mathbf{r}(t) - \mathbf{y}(t)$ rather than just $-\mathbf{y}(t)$



$$\Rightarrow \quad \mathbf{u}(t) = G_c(s)\mathbf{e}(t) = G_c(s)(\mathbf{r}(t) - \mathbf{y}(t))$$

  - Still have $\mathbf{u}(t) = -G_c(s)\mathbf{y}(t)$ if $\mathbf{r}(t) = 0$.
  - So now write state space model of the compensator as

$$\begin{aligned} \dot{\mathbf{x}}_c(t) &= A_c\mathbf{x}_c(t) + B_c\mathbf{e}(t) \\ \mathbf{u}(t) &= C_c\mathbf{x}_c(t) \end{aligned}$$

- Intuitively appealing setup because it is the **same approach** used for the classical control, but it turns out not to be the best approach $\Rightarrow$ More on this later.

Mechanics

- Basics:

$$\mathbf{e}(t) = \mathbf{r}(t) - \mathbf{y}(t), \qquad \mathbf{u}(t) = G_c\mathbf{e}(t), \qquad \mathbf{y}(t) = G\mathbf{u}(t)$$

$$G_c(s): \qquad \dot{\mathbf{x}}_c(t) = A_c\mathbf{x}_c(t) + B_c\mathbf{e}(t) \quad , \quad \mathbf{u}(t) = C_c\mathbf{x}_c(t)$$

$$G(s): \qquad \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) \quad , \quad \mathbf{y}(t) = C\mathbf{x}(t)$$

- Loop dynamics $L(s) = G(s)G_c(s) \quad \Rightarrow \quad \mathbf{y}(t) = L(s)\mathbf{e}(t)$

$$\begin{aligned} \dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + BC_c\,\mathbf{x}_c(t) \\ \dot{\mathbf{x}}_c(t) &= +A_c\,\mathbf{x}_c(t) + B_c\mathbf{e}(t) \end{aligned}$$

$$L(s) :\Rightarrow \begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\mathbf{x}}_c(t) \end{bmatrix} = \begin{bmatrix} A & BC_c \\ 0 & A_c \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{x}_c(t) \end{bmatrix} + \begin{bmatrix} 0 \\ B_c \end{bmatrix} \mathbf{e}(t)$$

$$\mathbf{y}(t) = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{x}_c(t) \end{bmatrix}$$

- To "close the loop", note that $\mathbf{e}(t) = \mathbf{r}(t) - \mathbf{y}(t)$, then

$$\begin{aligned} \begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\mathbf{x}}_c(t) \end{bmatrix} &= \begin{bmatrix} A & BC_c \\ 0 & A_c \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{x}_c(t) \end{bmatrix} + \begin{bmatrix} 0 \\ B_c \end{bmatrix} \left(\mathbf{r}(t) - \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{x}_c(t) \end{bmatrix} \right) \\ &= \begin{bmatrix} A & BC_c \\ -B_cC & A_c \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{x}_c(t) \end{bmatrix} + \begin{bmatrix} 0 \\ B_c \end{bmatrix} \mathbf{r}(t) \\ \mathbf{y}(t) &= \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{x}_c(t) \end{bmatrix} \end{aligned}$$

  - A_{cl} is not exactly the same as on page 15-2 because we have re-arranged where the negative sign enters into the problem. Same result though.

Scaling DOFB Compensator

- DOFB = Dynamic Output Feedback

- Assume that the closed-loop system has been written for the system on 15-6, then it is still possible that the DC gain of the response from r to y is not unity
  - Scalar input/output assumed for now

- Need to scale the closed-loop to fix this using $r \Rightarrow \overline{N}r$

- Closed-loop dynamics are

$$\begin{aligned}\dot{\mathbf{x}}_{cl}(t) &= A_{cl}\mathbf{x}_{cl}(t) + B_{cl}\overline{N}r(t)\\ y(t) &= C_{cl}\mathbf{x}_{cl}(t)\end{aligned}$$

- So to have the TF of y/u have unity gain at DC, must have

$$\overline{N} = -\left(C_{cl}(A_{cl})^{-1}B_{cl}\right)^{-1}$$

Simple Compensator Example

- Let $G(s) = 1/(s^2 + s + 1)$ with state-space model given by:

$$A = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad D = 0$$

- Design the regulator to place the poles at $s = -4 \pm 4j$

$$\lambda_i(A - BK) = -4 \pm 4j \quad \Rightarrow K = \begin{bmatrix} 31 & 7 \end{bmatrix}$$

  - Regulator pole time constant $\tau_c = 1/\zeta\omega_n \approx 1/4 = 0.25$ sec

- Put estimator poles with faster time constant $\tau_e \approx 1/10$
  - Use real poles, so $\Phi_e(s) = (s + 10)^2$

$$\begin{aligned} L &= \Phi_e(A) \begin{bmatrix} C \\ CA \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ &= \left(\begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}^2 + 20 \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix} + \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix} \right) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 99 & 19 \\ -19 & 80 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 19 \\ 80 \end{bmatrix} \end{aligned}$$

- Compensator:

$$
\begin{aligned}
A_c &= A - BK - LC \\
&= \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 31 & 7 \end{bmatrix} - \begin{bmatrix} 19 \\ 80 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} \\
&= \begin{bmatrix} -19 & 1 \\ -112 & -8 \end{bmatrix}
\end{aligned}
$$

$$
B_c = L = \begin{bmatrix} 19 \\ 80 \end{bmatrix}
$$

$$
C_c = K = \begin{bmatrix} 31 & 7 \end{bmatrix}
$$

```
a=[0 1;-1 -1];b=[0 1]';c=[1 0];d=0;
k=acker(a,b,[-4+4*j;-4-4*j]);
l=acker(a',c',[-10 -10]')';
%
% For state space for G_c(s)
%
ac=a-b*k-l*c;bc=l;cc=k;dc=0;
```

- Compensator transfer function:

$$
\begin{aligned}
G_c(s) &= C_c(sI - A_c)^{-1}B_c \triangleq \frac{U}{E} \\
&= 1149\frac{s + 2.553}{s^2 + 27s + 264}
\end{aligned}
$$

- Note that the compensator has a low frequency real zero and two higher frequency poles.
  - Thus it looks like a "lead" compensator.

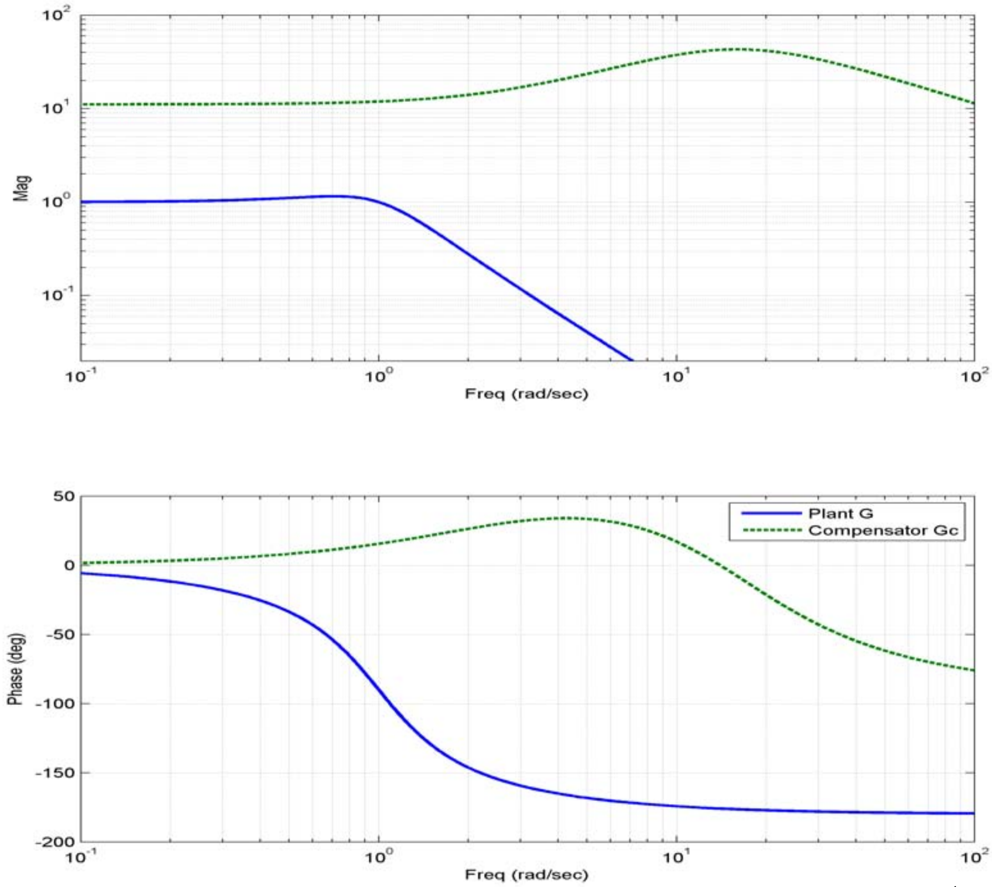


Fig. 1: Simple plant – compensator $\sim$ lead 2–10 rads/sec.

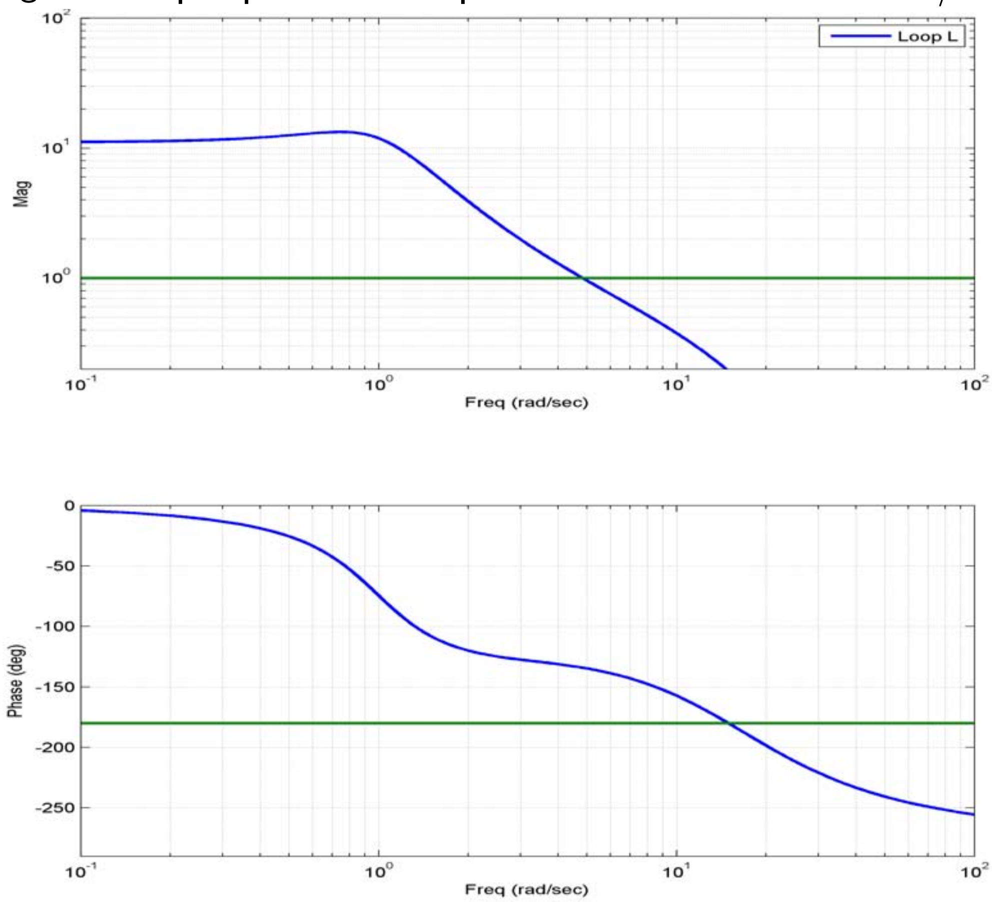


Fig. 2: Loop transfer function $L(s)$ shows the slope change near $\omega_c = 5$ rad/sec. Note that we have a large PM and GM.

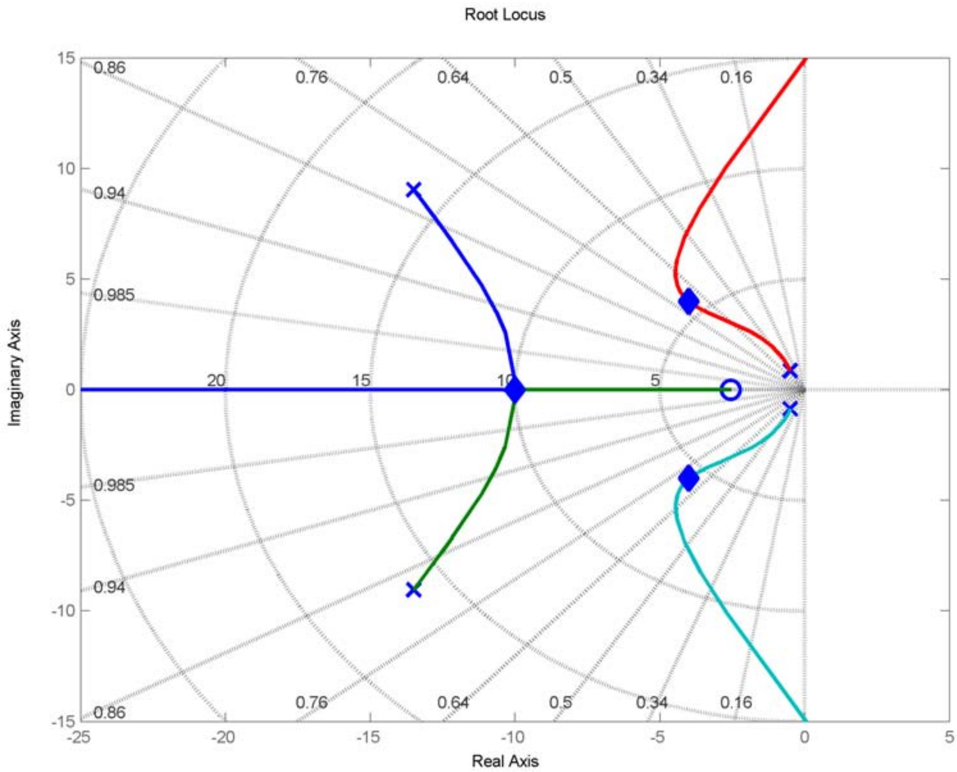


Fig. 3: Freeze compensator poles and zeros; look at root locus of closed-loop poles versus an additional loop gain (nominally $\alpha = 1.$)

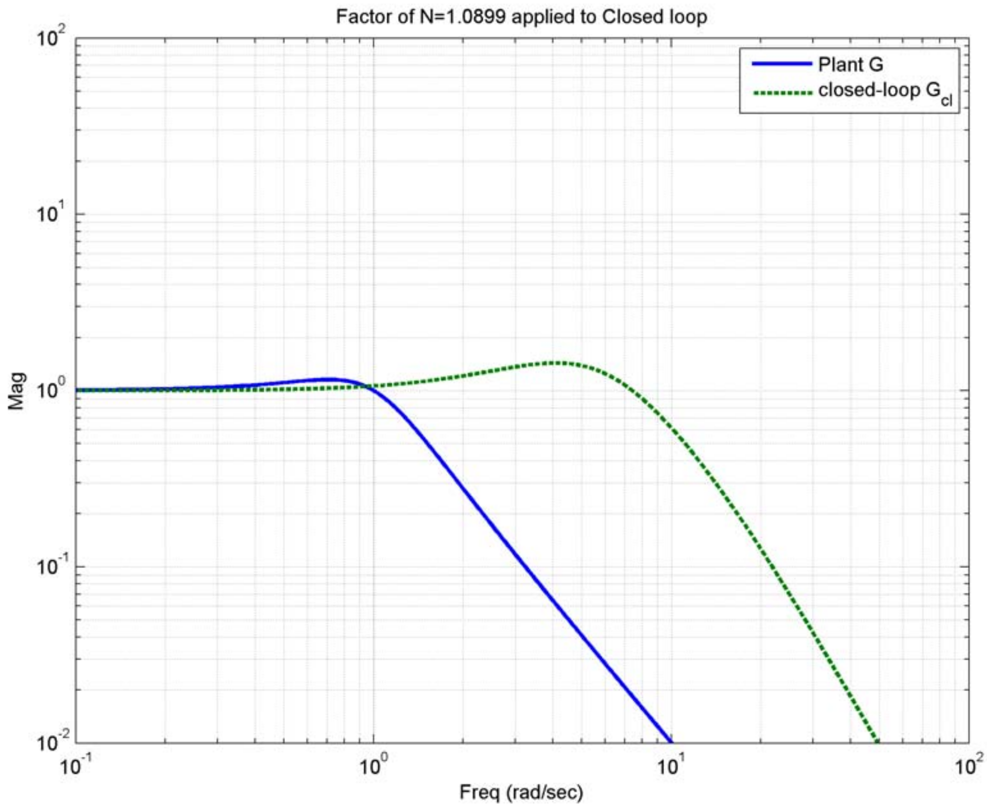


Fig. 4: Closed-loop transfer function.

Code: Dynamic Output Feedback Example

```
% Combined estimator/regulator design for a simple system
% G= 1/(s^2+s+1)
% Jonathan How
% Fall 2010
%
%close all;clear all
set(0,'DefaultLineLineWidth',2)
set(0,'DefaultFigureColor','None')
set(0,'DefaultlineMarkerSize',8);set(0,'DefaultlineMarkerFace','b')
set(0, 'DefaultAxesFontSize', 12);set(0, 'DefaultTextFontSize', 12);
%
a=[0 1;-1 -1];b=[0 1]';c=[1 0];d=0;
k=acker(a,b,[-4+4*j;-4-4*j]);
l=acker(a',c',[-10 -11]')';
%
% For state space for G_c(s)
%
ac=a-b*k-l*c;bc=l;cc=k;dc=0;

G=ss(a,b,c,d);
Gc=ss(ac,bc,cc,dc);

f=logspace(-1,2,400);
g=freqresp(G,f*j);g=squeeze(g);
gc=freqresp(Gc,f*j);gc=squeeze(gc);

figure(1);clf;orient tall
subplot(211)
loglog(f,abs(g),f,abs(gc),'--');axis([.1 1e2 .02 1e2])
xlabel('Freq (rad/sec)');ylabel('Mag')
legend('Plant G','Compensator Gc','Location','East');
grid
subplot(212)
semilogx(f,180/pi*angle(g),f,180/pi*angle(gc),'--');
axis([.1 1e2 -200 50])
xlabel('Freq (rad/sec)');ylabel('Phase (deg)');grid

L=g.*gc;

figure(2);clf;orient tall
subplot(211)
loglog(f,abs(L),[.1 1e2],[1 1]);axis([.1 1e2 .2 1e2])
xlabel('Freq (rad/sec)');ylabel('Mag')
legend('Loop L');
grid
subplot(212)
semilogx(f,180/pi*phase(L.'),[.1 1e2],-180*[1 1]);
axis([.1 1e2 -290 0])
xlabel('Freq (rad/sec)');ylabel('Phase (deg)');grid
%
% loop dynamics L = G Gc
%
al=[a b*cc;zeros(2) ac];
bl=[zeros(2,1);bc];
cl=[c zeros(1,2)];
dl=0;
figure(3)
rlocus(al,bl,cl,dl)
%
% closed-loop dynamics
% unity gain wrapped around loop L
%
acl=al-bl*cl;bcl=bl;ccl=cl;dcl=d;
%
% scale closed-loop to get zero SS error
N=inv(ccl*inv(-acl)*bcl)

hold on;plot(eig(acl),'d','MarkerFaceColor','b');hold off
grid on
axis([-25 5 -15 15])
%
% closed-loop freq response
%
Gcl=ss(acl,bcl*N,ccl,dcl*N);
gcl=freqresp(Gcl,f*j);gcl=squeeze(gcl);

```

```
figure(4);clf
loglog(f,abs(g),f,abs(gcl),'--');
axis([.1 1e2 .01 1e2])
xlabel('Freq (rad/sec)');ylabel('Mag')
legend('Plant G','closed-loop G_{cl}');grid
title(['Factor of N=',num2str(round(1000*N)/1000),' applied to Closed loop'])

figure(5);clf
margin(al,bl,cl,dl)

figure(1);export_fig reg_est1 -pdf;
figure(2);export_fig reg_est2 -pdf;
figure(3);export_fig reg_est3 -pdf;
figure(4);export_fig reg_est4 -pdf;
figure(5);export_fig reg_est5 -pdf;

figure(6);clf
% new form on 16-7
bcl=[-b;-b];
N=inv(ccl*inv(-acl)*bcl)
Gnew=ss(acl,bcl*N,ccl,dcl*N);
[yorig,t]=step(Gcl,2.5);
[ynew]=step(Gnew,t);
plot(t,yorig,t,ynew);setlines(2);
legend('Original','New');grid on
ylabel('Y');xlabel('Time (s)')
figure(6);export_fig reg_est6 -pdf;
```

Fig. 5: Example #1: $G(s) = \frac{8 \cdot 14 \cdot 20}{(s+8)(s+14)(s+20)}$

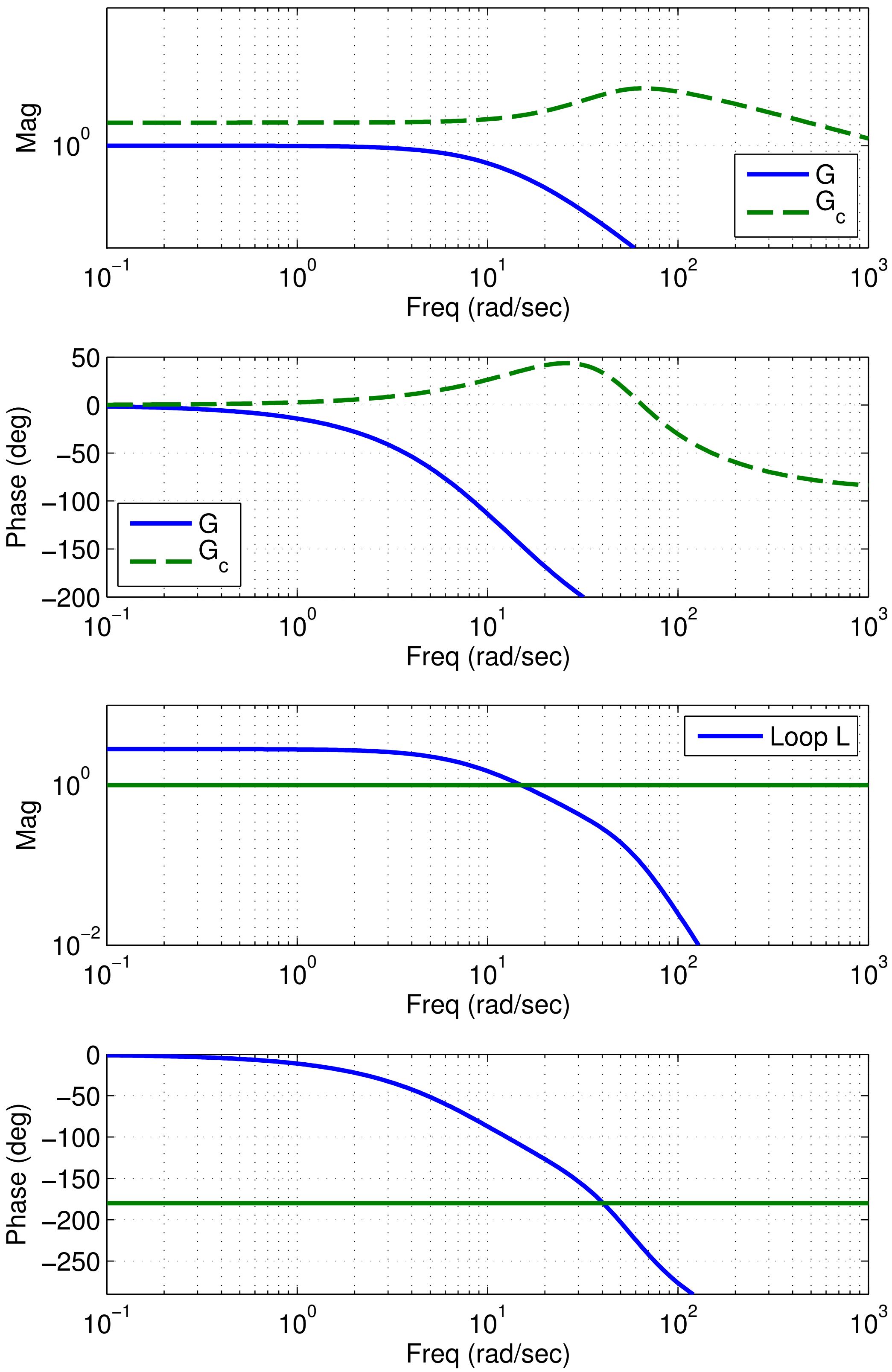
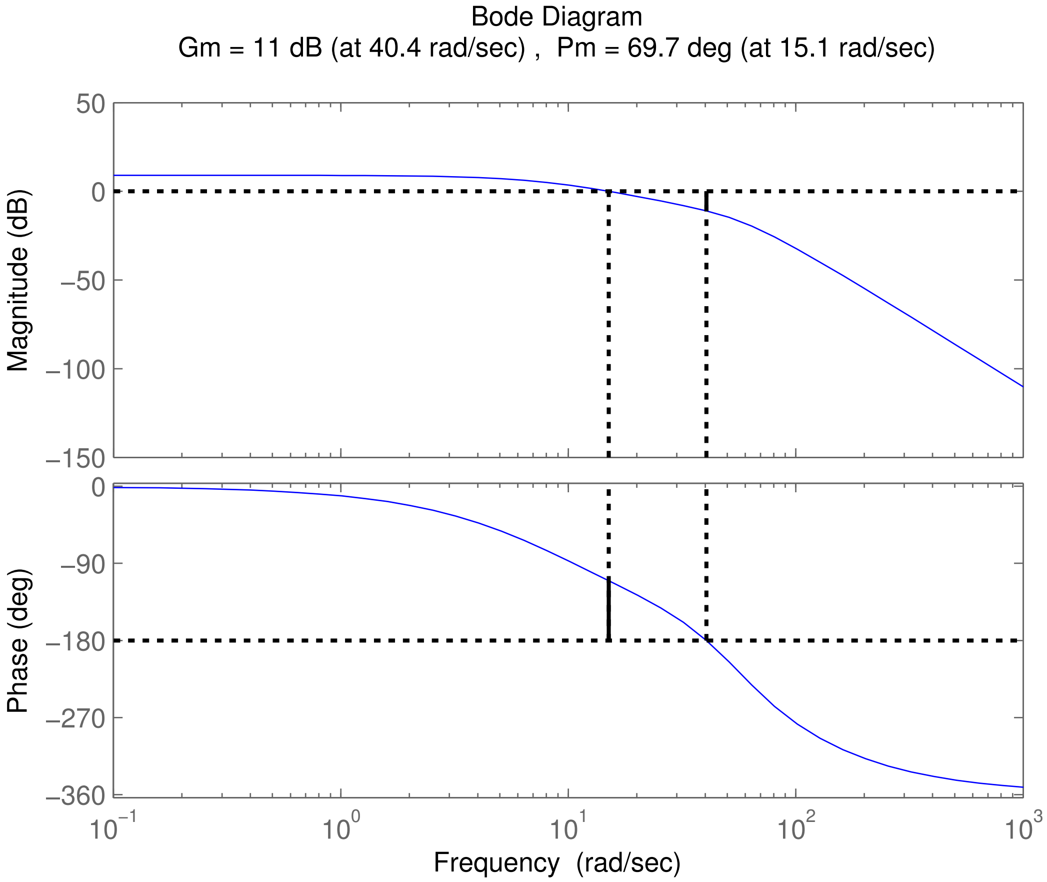


Fig. 6: Example #1: $G(s) = \frac{8 \cdot 14 \cdot 20}{(s+8)(s+14)(s+20)}$



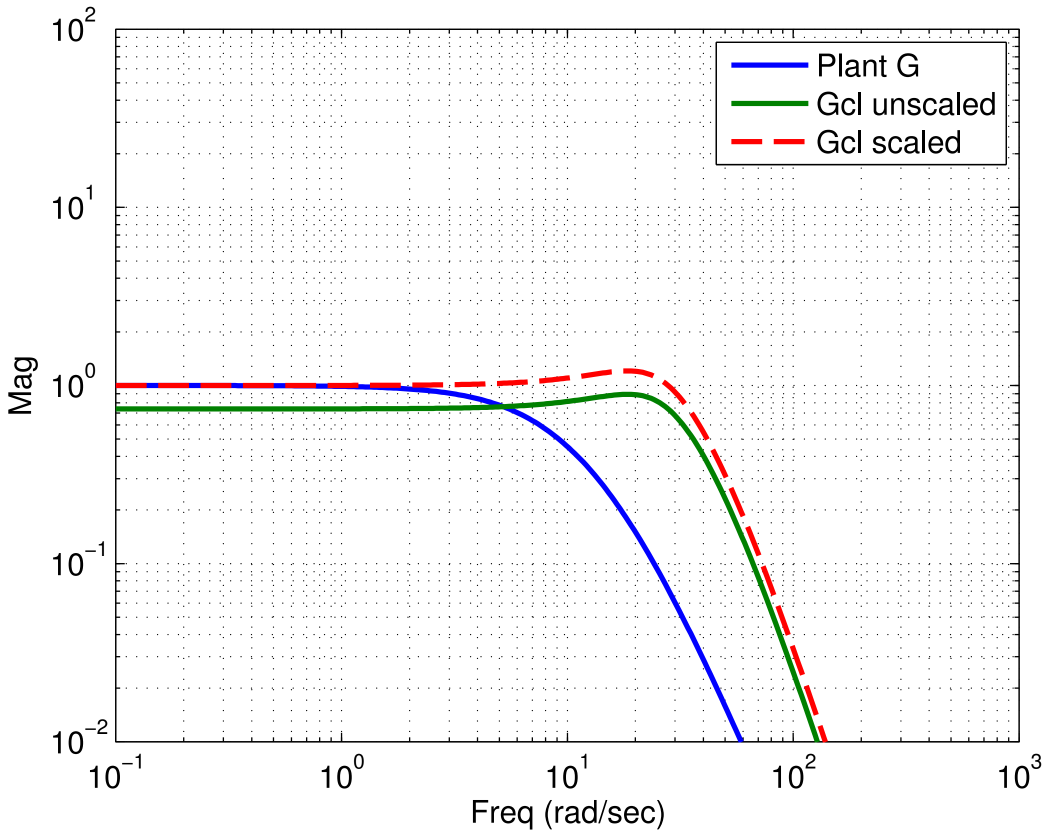
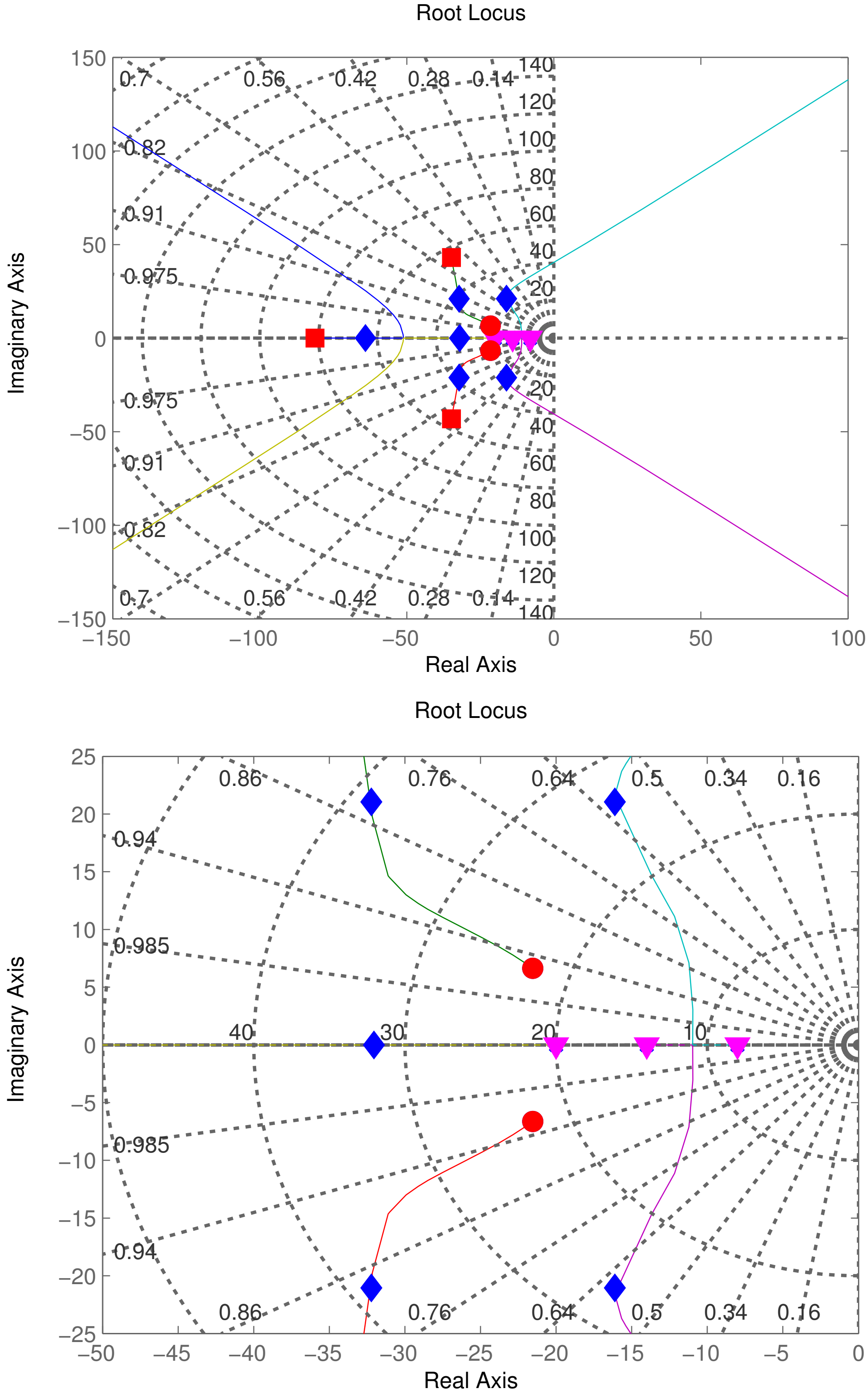


Fig. 7: Example #1: $G(s) = \frac{8 \cdot 14 \cdot 20}{(s+8)(s+14)(s+20)}$



◆ – closed-loop poles, ▼ -- Plant poles, ● – Plant zeros, ■ – Compensator poles, ● – Compensator zeros

- Two compensator zeros at -21.54±6.63j draw the two lower frequency plant poles further into the LHP.
- Compensator poles are at much higher frequency.
- Looks like a lead compensator.

Fig. 8: Example #2: $G(s) = \frac{0.94}{s^2 - 0.0297}$

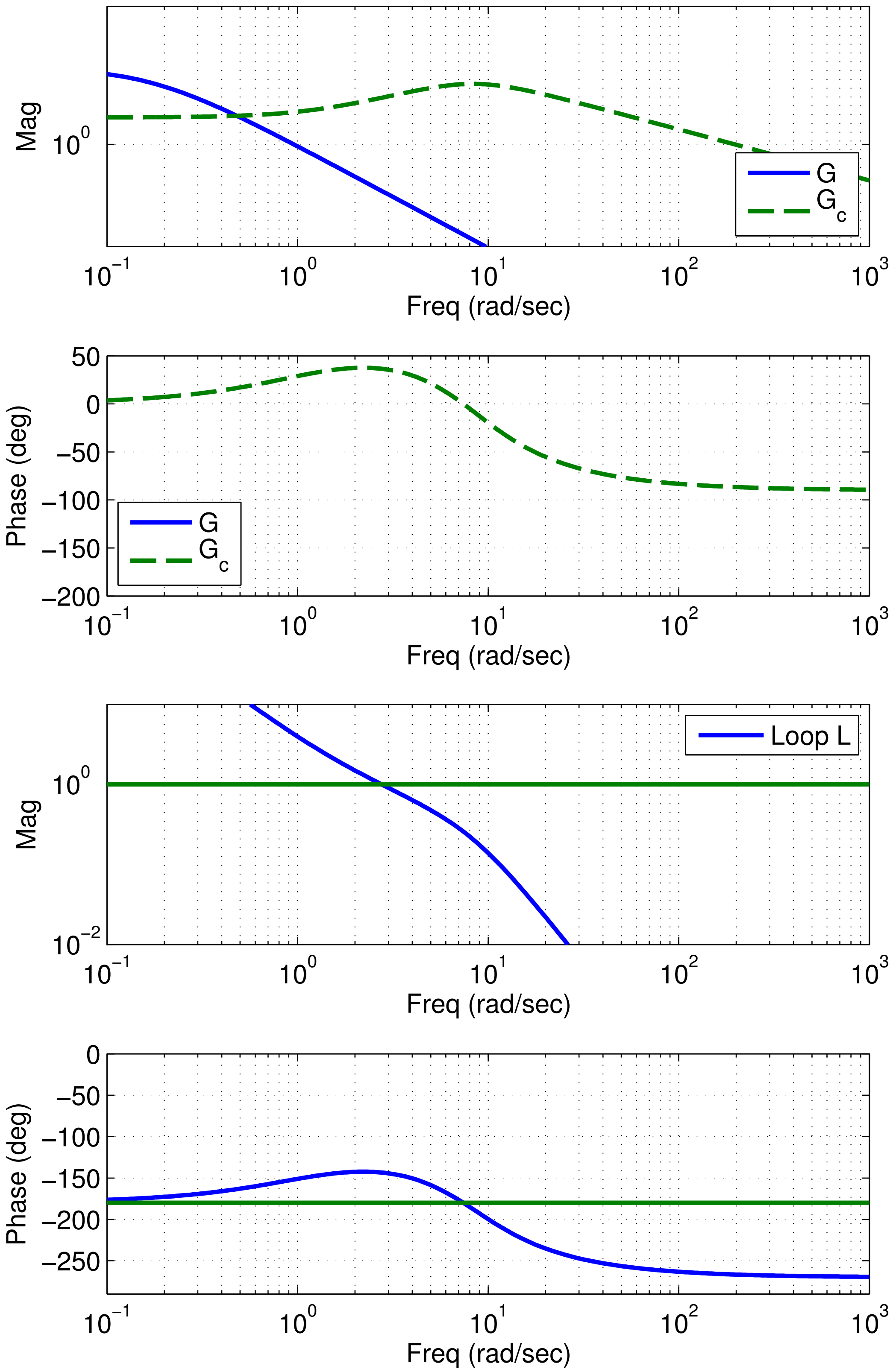


Fig. 9: Example #2: $G(s) = \frac{0.94}{s^2 - 0.0297}$

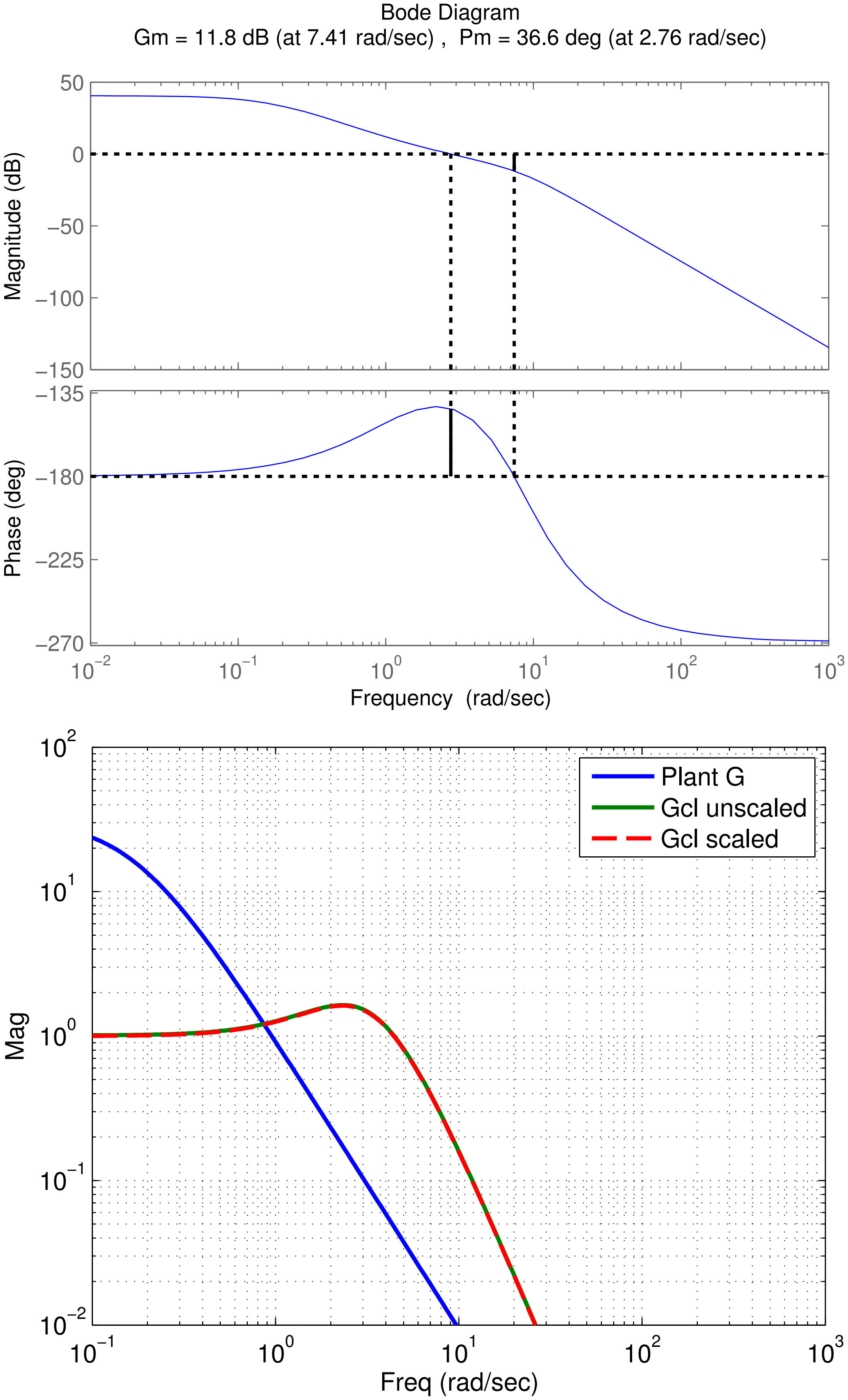
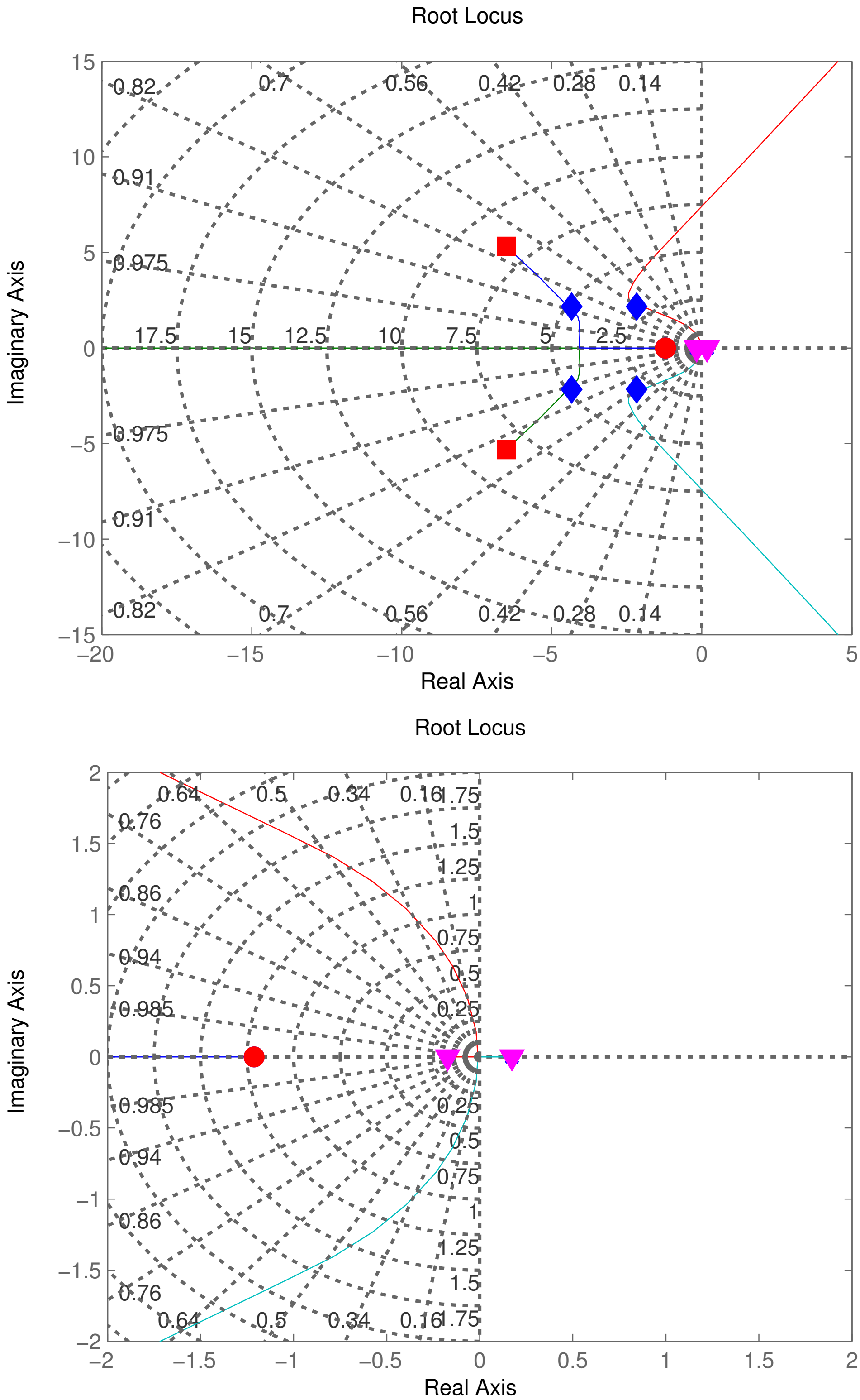


Fig. 10: Example #2: $G(s) = \frac{0.94}{s^2 - 0.0297}$



◆ – closed-loop poles, ▼ -- Plant poles, ● – Plant zeros, ■ – Compensator poles, ● – Compensator zeros

- Compensator zero at -1.21 draws the two lower frequency plant poles further into the LHP.
- Compensator poles are at much higher frequency.
- Looks like a lead compensator.

Fig. 11: Example #3: $G(s) = \frac{8 \cdot 14 \cdot 20}{(s-8)(s-14)(s-20)}$

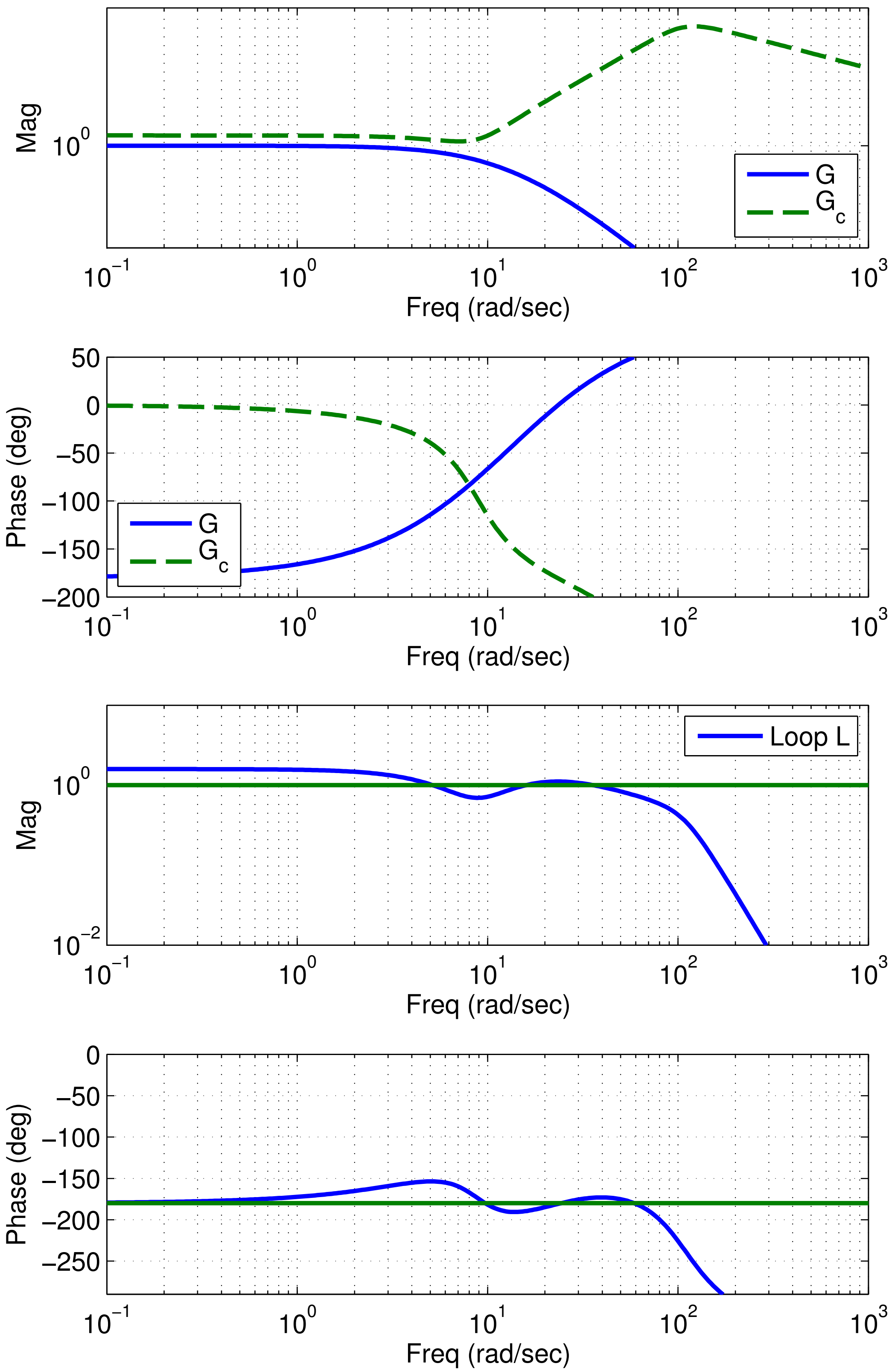
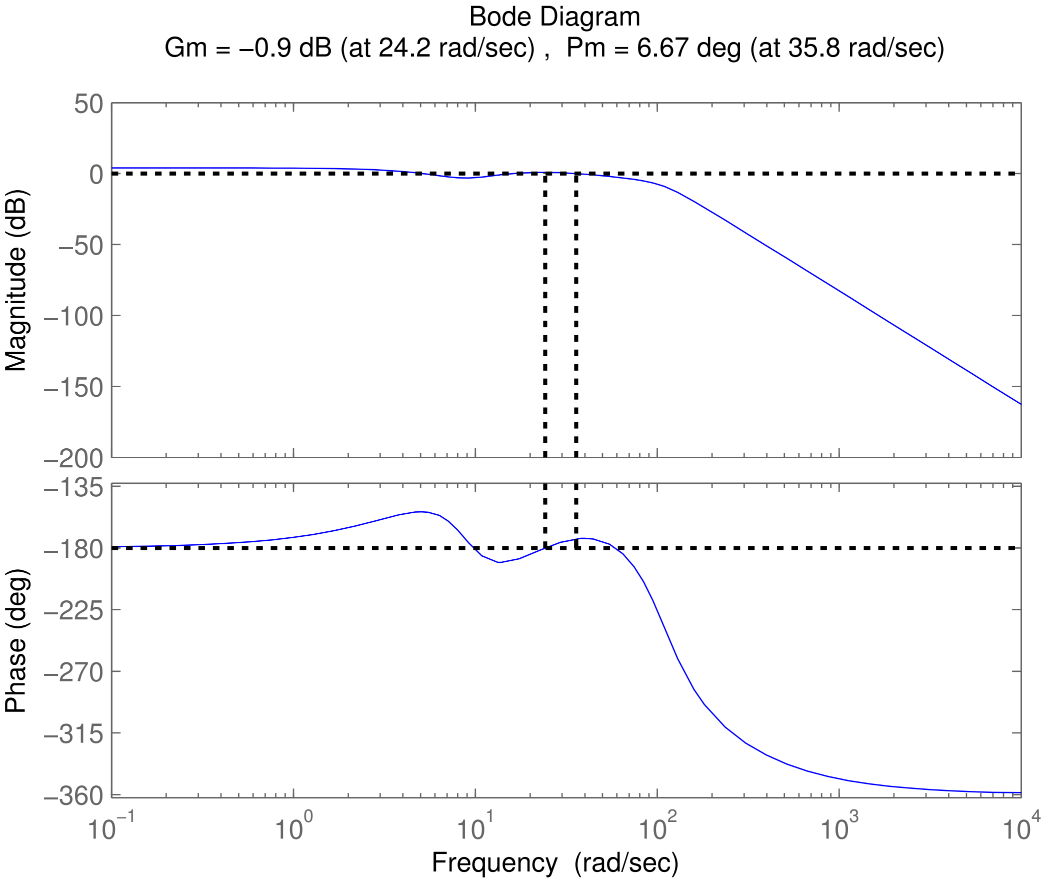


Fig. 12: Example #3: $G(s) = \frac{8 \cdot 14 \cdot 20}{(s-8)(s-14)(s-20)}$



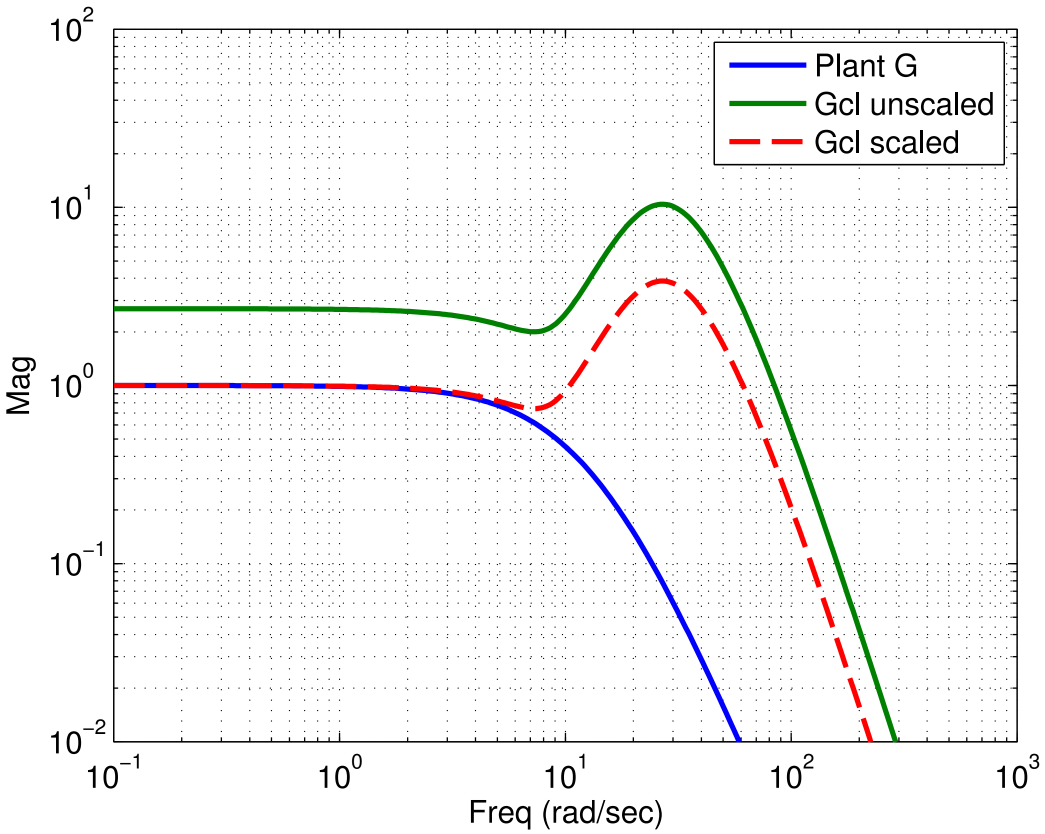
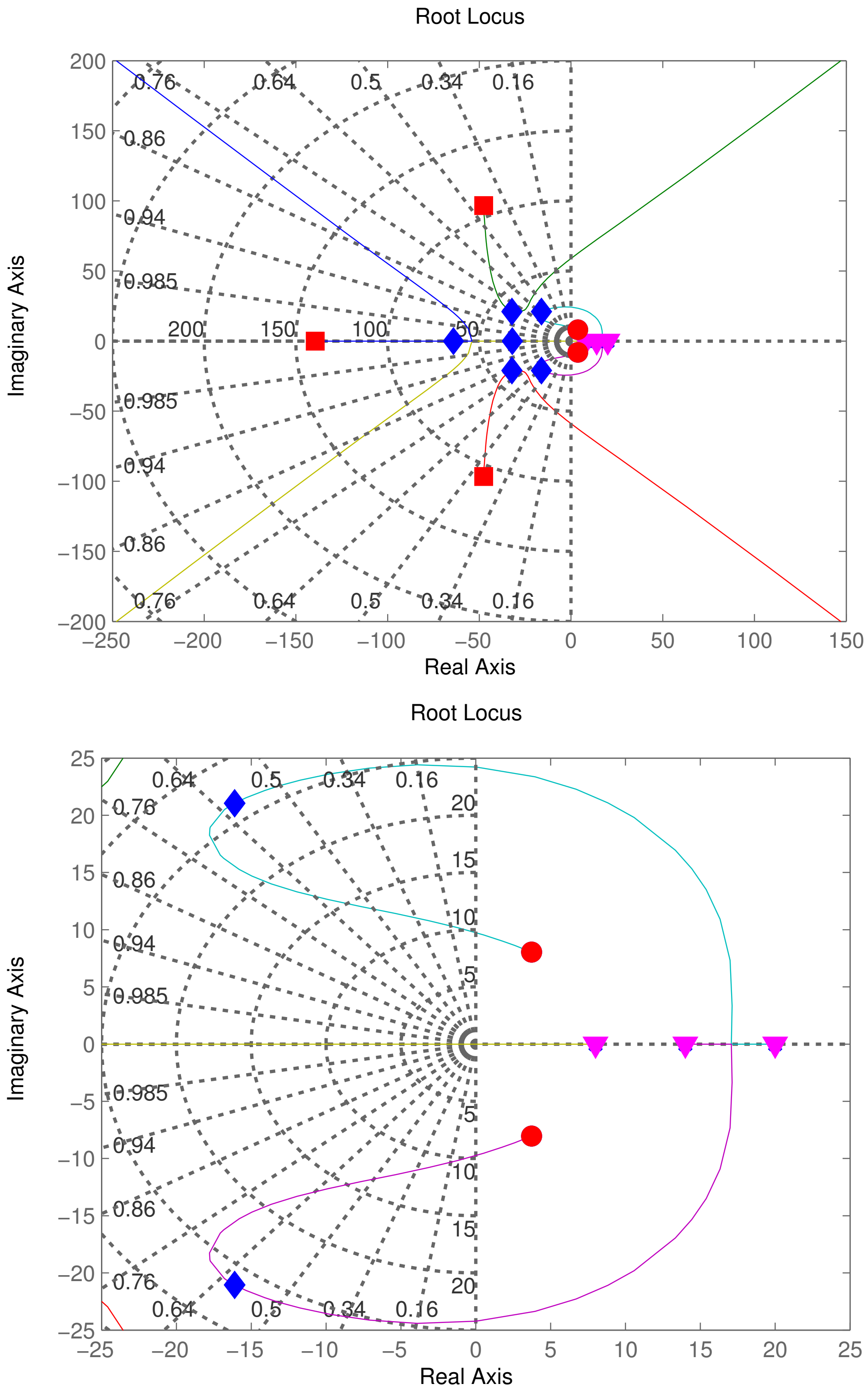


Fig. 13: Example #3: $G(s) = \frac{8 \cdot 14 \cdot 20}{(s-8)(s-14)(s-20)}$



◆ – closed-loop poles, ▼ -- Plant poles, ● – Plant zeros, ■ – Compensator poles, ● – Compensator zeros

- Compensator zeros at 3.72±8.03j draw the two higher frequency plant poles further into the LHP. Lowest frequency one heads into the LHP on its own.
- Compensator poles are at much higher frequency.
- Not sure what this looks like.

Fig. 14: Example #4: $G(s) = \frac{(s-1)}{(s+1)(s-3)}$

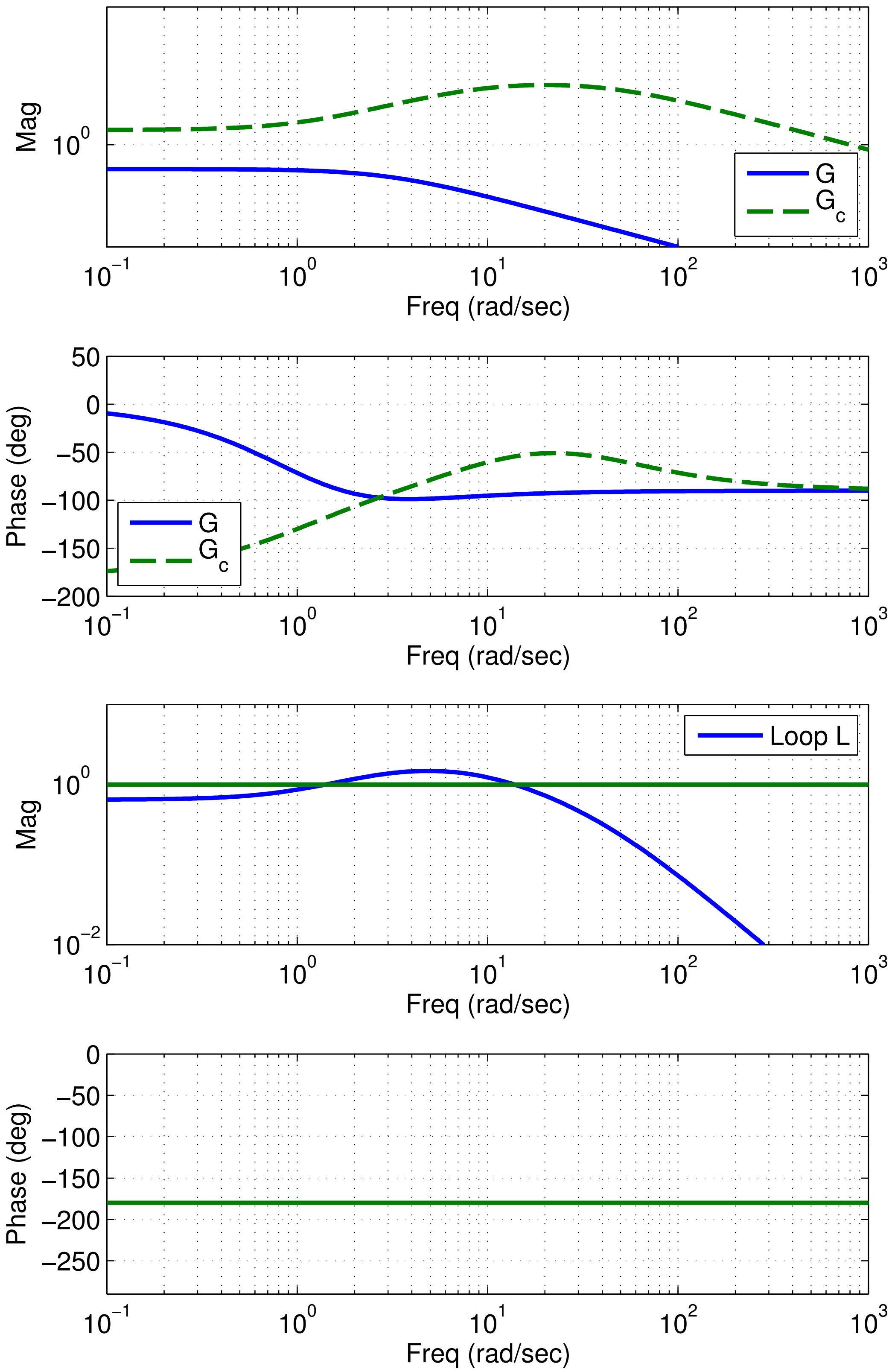
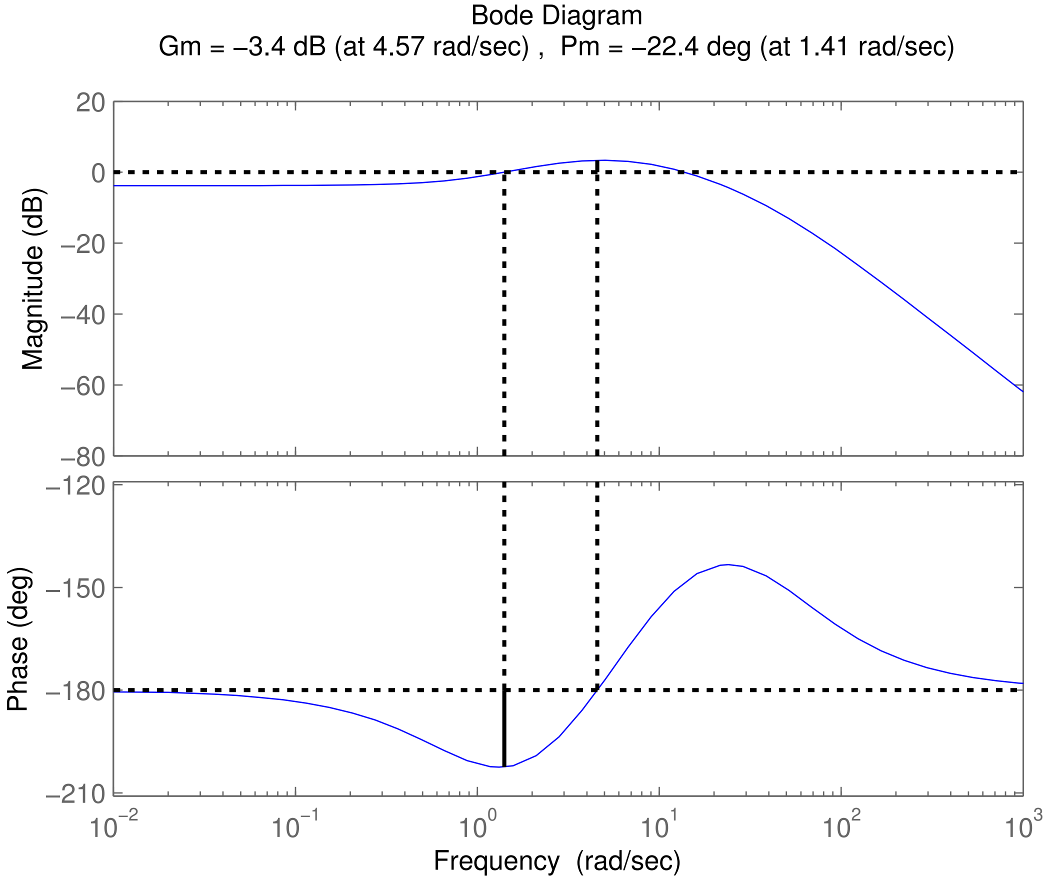


Fig. 15: Example #4: $G(s) = \frac{(s-1)}{(s+1)(s-3)}$



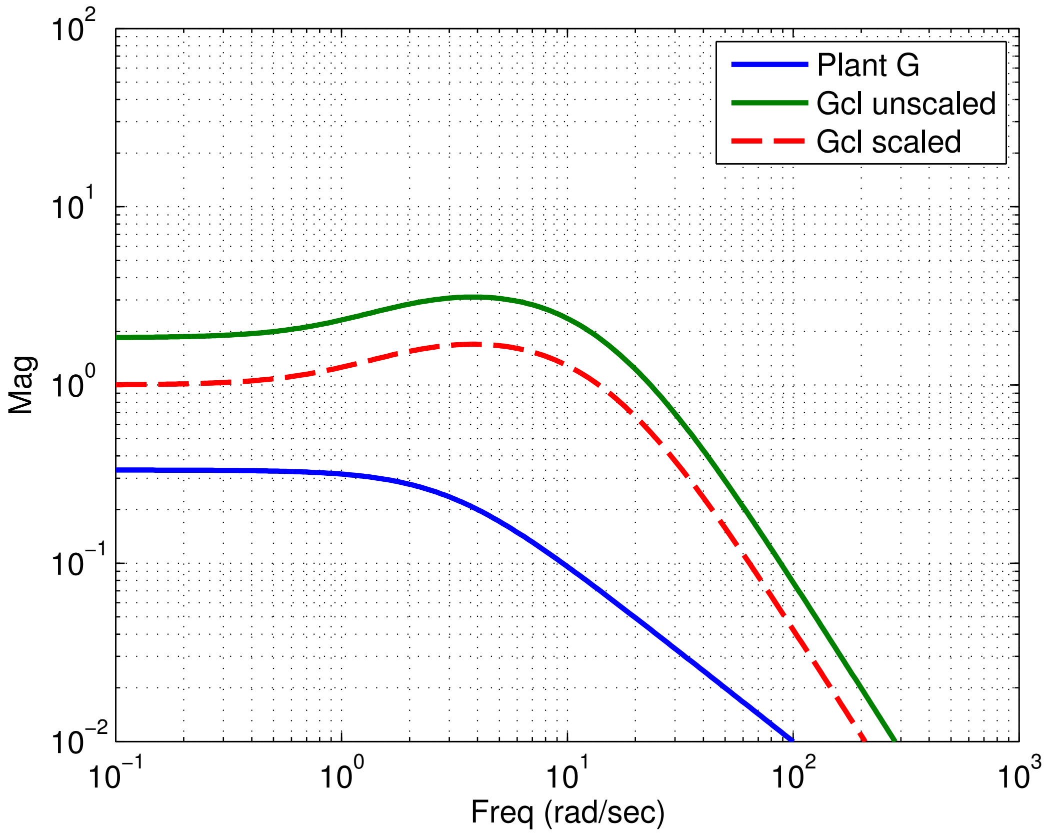
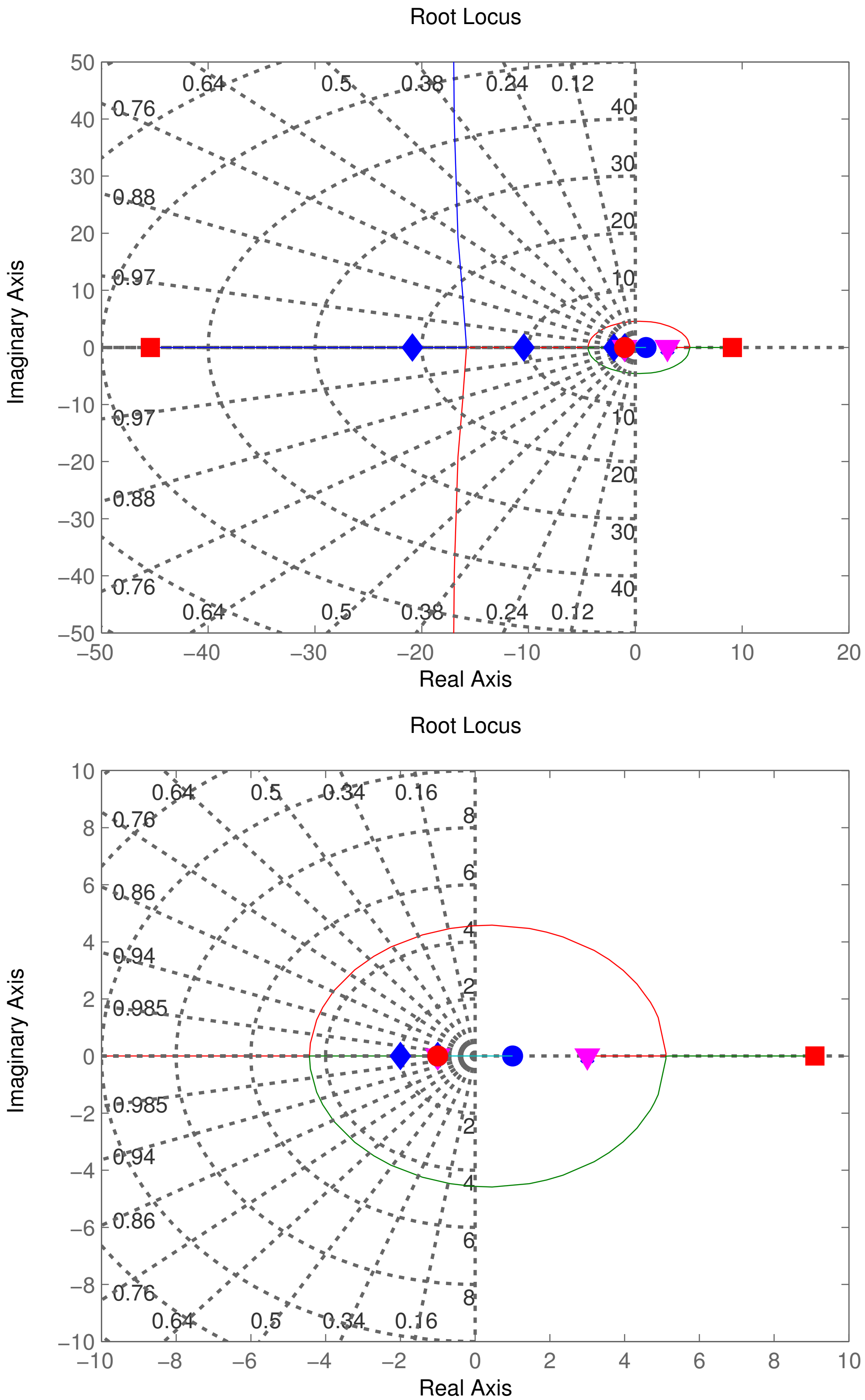


Fig. 16: Example #4: $G(s) = \frac{(s-1)}{(s+1)(s-3)}$



◆ – closed-loop poles, ▼ -- Plant poles, ● – Plant zeros, ■ – Compensator poles, ● – Compensator zeros

- Compensator zero at -1 cancels the plant pole. Note the very unstable compensator pole at $s = 9$!!
  - Needed to get the RHP plant pole to branch off the real line and head into the LHP.
- Other compensator pole is at much higher frequency.
- Note sure what this looks like.

B747 DOFB

- The previous B747 controller assumed full state feedback - would like to extend that now to consider output feedback using the output of the washout filtered yaw rate – e

- To place the estimator poles, take the approach of just increasing the frequency of the LQR closed-loop poles by a factor of 2:

$$\lambda^d(A_{est}) = 2 \times \lambda(A_{lqr})$$

- Then use pole placement to determine the estimator gains `place.m`

```
Gp=WashFilt*sys*H;
set(Gp, 'statename', {'xwo' 'beta' 'yaw rate' 'roll' 'phi' 'xa'});
set(Gp, 'inputname', {'rudder inputs' 'aileron'},...
          'outputname', {'filtered yaw rate' 'bank angle'});
[Ap,Bp,Cp,Dp]=ssdata(Gp);
[Klqr,S,Elqr]=lqr(Ap,Bp(:,1),Cp(1,:)'*Cp(1,:),0.1)

Eest=real(Elqr)*2+2*imag(Elqr)*sqrt(-1);
Kest=place(Ap',Cp(1,:)',Eest);Kest=Kest';
```

- Note the transposes on the A and C here in the `place` command

- Yields the gains

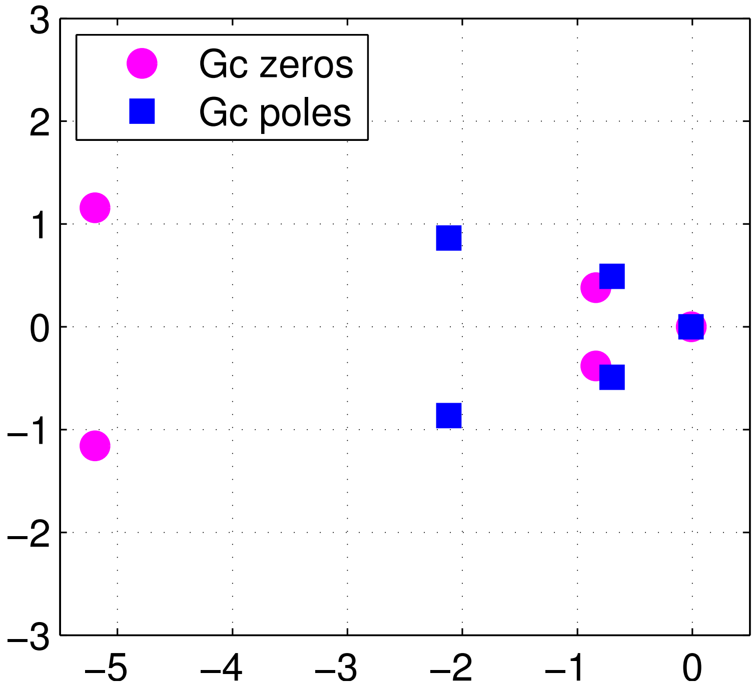
$$L_{est} = \begin{bmatrix} -31.500 & 5.2985 & -7.854 & -122.6895 & 121.7798 & 54.858 \end{bmatrix}$$

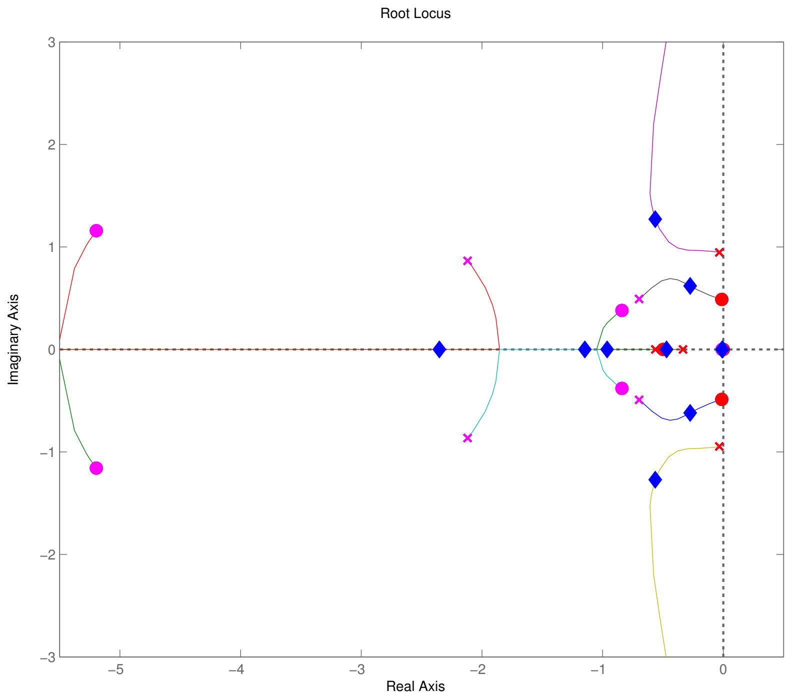
  - Much larger that LQR gains because the poles are moving further

- Now form DOFB compensator as before:

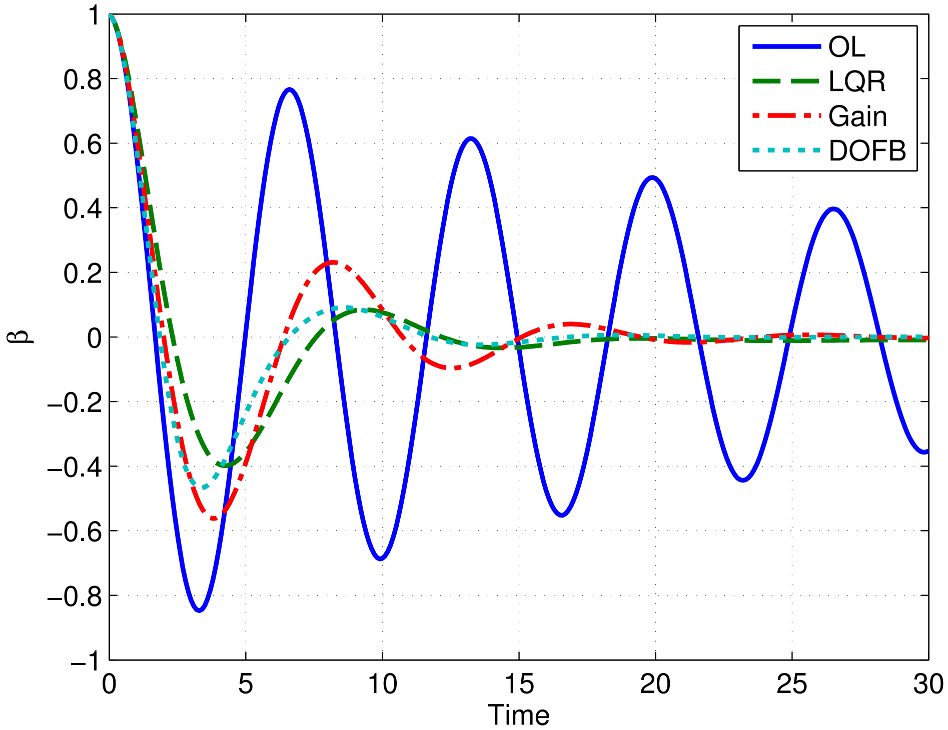
```
na=size(Ap,1);
% Form compensator and closed−loop
% see 15−5
ac=Ap−Bp(:,1)*Klqr−Kest*Cp(1,:);bc=Kest;cc=Klqr;dc=0;
Gc=ss(ac,bc,cc,dc);
% see 15−7
acl=[Ap Bp(:,1)*cc;−bc*Cp(1,:) ac];bcl=[zeros(na,1);bc];ccl=[Cpbeta,zeros(1,na)];dcl=0;
Gcl=ss(acl,bcl,ccl,dcl);
```

- Consider the DOFB compensator for this system – it is 6th order because the augmented plant is (4+1+1) – G_c pole-zero map and closed-loop poles shown

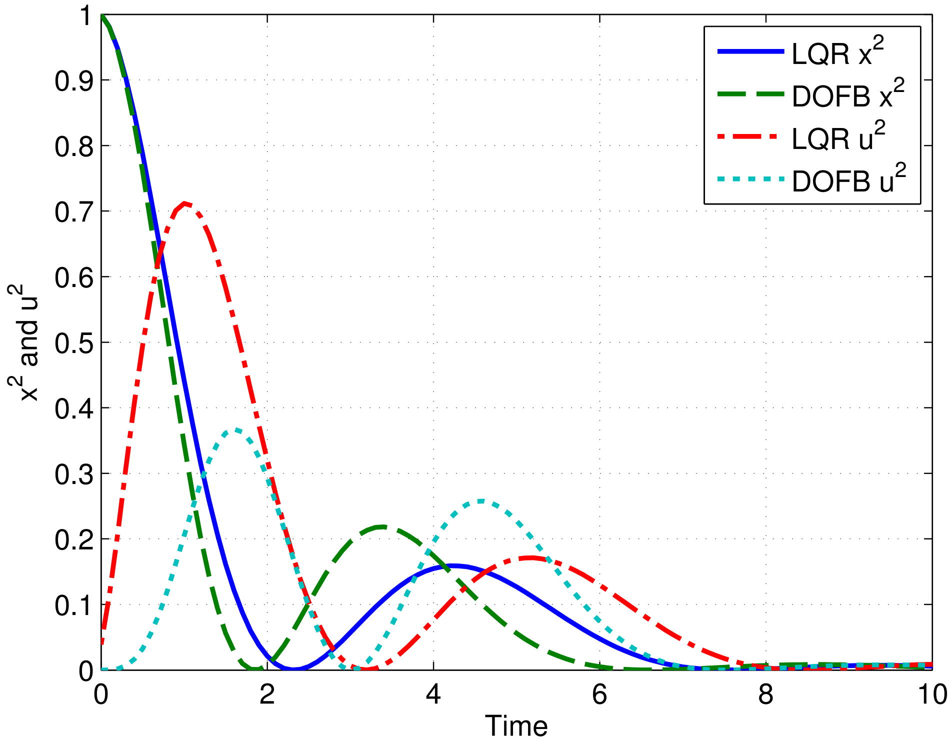




- Now consider the time response again to an initial perturbation in β:



  - Time response looks good – perhaps even better than LQR, but note that is not inconsistent as this DOFB might be using more control effort, and LQR balances that effort with the state cost



Code: B747 LQR, Estimator and DOFB

```
% B747 lateral dynamics
% Jonathan How, Fall 2010
% working from Matlab Demo called jetdemo
clear all;%close all
%
A=[-.0558 -.9968 .0802 .0415;
     .598 -.115 -.0318 0;
    -3.05 .388 -.4650 0;
        0 0.0805 1 0];
B=[ .00729  0;
   -0.475   0.00775;
    0.153   0.143;
     0      0];
C=[0 1 0 0; 0 0 0 1];D=[0 0; 0 0];
sys = ss(A,B,C,D);
set(sys, 'inputname', {'rudder' 'aileron'},...
         'outputname', {'yaw rate' 'bank angle'});
set(sys, 'statename', {'beta' 'yaw rate' 'roll rate' 'phi'});
[Yol,Tol]=initial(ss(A,B,[1 0 0 0],zeros(1,2)),[1 0 0 0]',[0:.1:30]);
[V,E]=eig(A)
%
% CONTROL - actuator dynamics are a lag at 10
actn=10;actd=[1 10]; % H(s) in notes
H=tf({actn 0;0 1},{actd 1;1 1});
%
tau=3;washn=[1 0];washd=[1 1/tau]; % washout filter on yaw rate
WashFilt=tf({washn 0;0 1},{washd 1;1 1});
%
Gp=WashFilt*sys*H;
set(Gp, 'statename', {'xwo' 'beta' 'yaw rate' 'roll' 'phi' 'xa'});
set(Gp, 'inputname', {'rudder inputs' 'aileron'},...
         'outputname', {'filtered yaw rate' 'bank angle'});
[Ap,Bp,Cp,Dp]=ssdata(Gp);
[Klqr,S,Elqr]=lqr(Ap,Bp(:,1),Cp(1,:)'*Cp(1,:),0.1);rifd(Elqr)
%
% gain feedback on the filter yaw rate
figure(1);clf
rlocus(Ap,Bp(:,1),-Cp(1,:),0);
sgrid([.1 .2 .3 .4],[.7 .8 .9 1]);grid on;axis([-1.4 .1 -1.2 1.2])
Kgain=-2;
Egain=eig(Ap-Bp(:,1)*Kgain*Cp(1,:));rifd(Egain)
hold on;plot(Egain+eps*sqrt(-1),'bd');hold off
export_fig b747_lqr1 -pdf

Cpbeta=[0 1 0 0 0 0]; % performance output variable
Ggain=ss(Ap-Bp(:,1)*Kgain*Cp(1,:),Bp,Cpbeta,0);
xp0=[0 1 0 0 0 0]';
[Ygain,Tgain]=initial(Ggain,xp0,Tol); %'

figure(2);clf
srl(ss(Ap,Bp(:,1),Cp(1,:),0));
sgrid([.1 .2 .3 .4],[.7 .8 .9 1])
axis([-1.25 1.25 -1.25 1.25]);grid on
hold on;plot(Elqr+eps*sqrt(-1),'bd');hold off
export_fig b747_lqr2 -pdf

% close the LQR loop
Acl=Ap-Bp(:,1)*Klqr;
Bcl=Bp(:,1);
% choose output to just state beta and control u
Ccl=[Cpbeta;Klqr];
Dcl=[0;0];
Glqr=ss(Acl,Bcl,Ccl,Dcl);
[Y,T]=initial(Glqr,xp0,Tol); %'

figure(3);clf
plot(Tol,Yol,T,Y(:,1),Tgain,Ygain);axis([0 30 -1 1]);
setlines(2)
legend('OL','LQR','Gain')
ylabel('\beta');xlabel('Time')
grid on
export_fig b747_lqr3 -pdf

% estimator poles made faster by jfactor
```

```
jfactor=2.05;
Eest=jfactor*Elqr;
Kest=place(Ap',Cp(1,:)',Eest);Kest=Kest';

na=size(Ap,1);
% Form compensator and closed-loop
% see 15-5
ac=Ap-Bp(:,1)*Klqr-Kest*Cp(1,:);bc=Kest;cc=Klqr;dc=0;
Gc=ss(ac,bc,cc,dc);
% see 15-7
acl=[Ap Bp(:,1)*cc;-bc*Cp(1,:) ac];
bcl=[zeros(na,1);bc];
% choose output to be state beta and control u
ccl=[Cpbeta,zeros(1,na);zeros(1,na) cc];
dcl=[0;0];
Gcl=ss(acl,bcl,ccl,dcl);

figure(5);clf
[p,z]=pzmap(Gc(1,1));
plot(z+sqrt(-1)*eps,'mo','MarkerFace','m')
hold on
plot(p+sqrt(-1)*eps,'bs','MarkerFace','b')
hold off
legend('Gc zeros','Gc poles','Location','NorthWest')
grid on;axis([-5.5 .5 -3 3])
export_fig b747_lqr5 -pdf

figure(4);clf
rlocus(Gp(1,1)*Gc);axis([-5.5 .5 -3 3])
hold on
[p,z]=pzmap(Gp(1,1));
plot(p+sqrt(-1)*eps,'rx','MarkerFace','r')
plot(z+sqrt(-1)*eps,'ro','MarkerFace','r')
[p,z]=pzmap(Gc(1,1));
plot(p+sqrt(-1)*eps,'mx','MarkerFace','m')
plot(z+sqrt(-1)*eps,'mo','MarkerFace','m')
p=eig(acl);
plot(p+sqrt(-1)*eps,'bd')
hold off
export_fig b747_lqr4 -pdf

% initial comp at zero
[Ycl,Tcl]=initial(Gcl,[xp0;xp0*0],Tol); %'

figure(3);clf
plot(Tol,Yol,T,Y(:,1),Tgain,Ygain,Tcl,Ycl(:,1));axis([0 30 -1 1]);
setlines(2)
legend('OL','LQR','Gain','DOFB')
ylabel('\beta');xlabel('Time')
grid on
export_fig b747_lqr3a -pdf

figure(6);clf
plot(T,Y(:,1).^2,Tcl,Ycl(:,1).^2,T,Y(:,2).^2,Tcl,Ycl(:,2).^2);
axis([0 10 0 1]);
hold off
setlines(2)
legend('LQR x^2','DOFB x^2','LQR u^2','DOFB u^2')
ylabel('x^2 and u^2');xlabel('Time')
grid on
export_fig b747_lqr3b -pdf
```

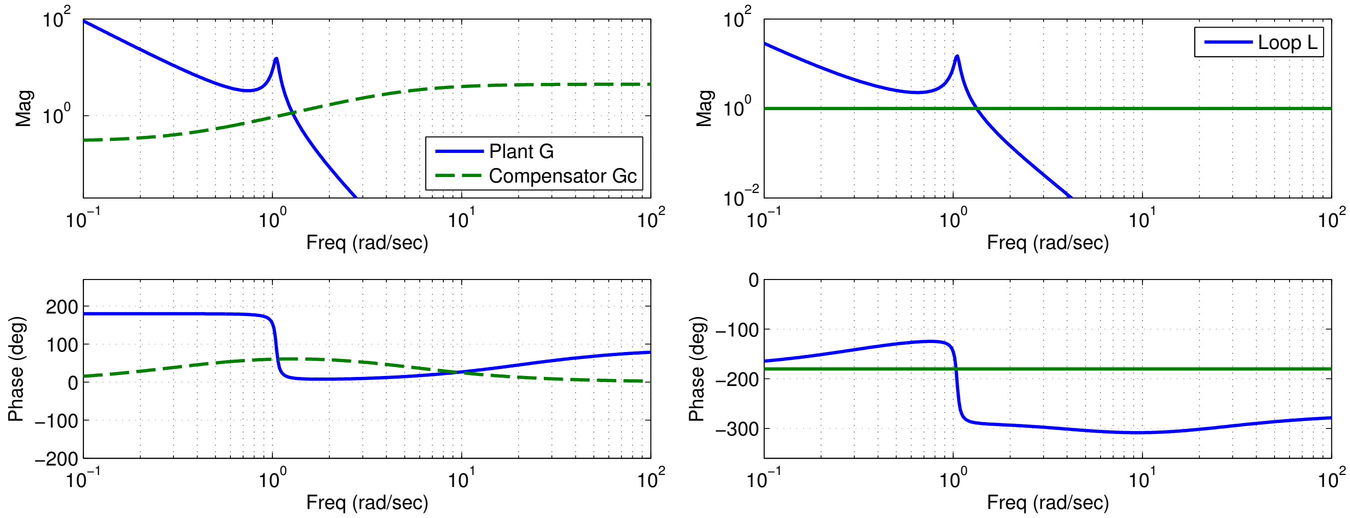
Satellite Control Design

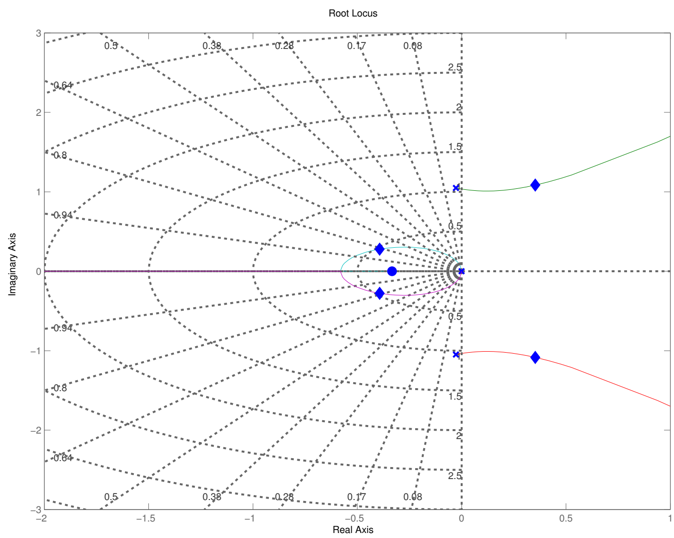
- Simple system model – goal is to point payload device ($J_2 = 0.1$) using torque inputs on the larger spacecraft ($J_1 = 1$)
  - Sensor is star tracker on payload (angle), and actuator is torque applied to spacecraft

- Can form the state space model using the state $\mathbf{x} = [\ \theta_2 \ \ \dot{\theta}_2 \ \ \theta_1 \ \ \dot{\theta}_1\ ]^T$
  - Plant: $G_p(s) = \dfrac{10bs + 10k}{s^2(s^2 + 11bs + 11k)}$, $k = 0.1$ and $b = 0.005$

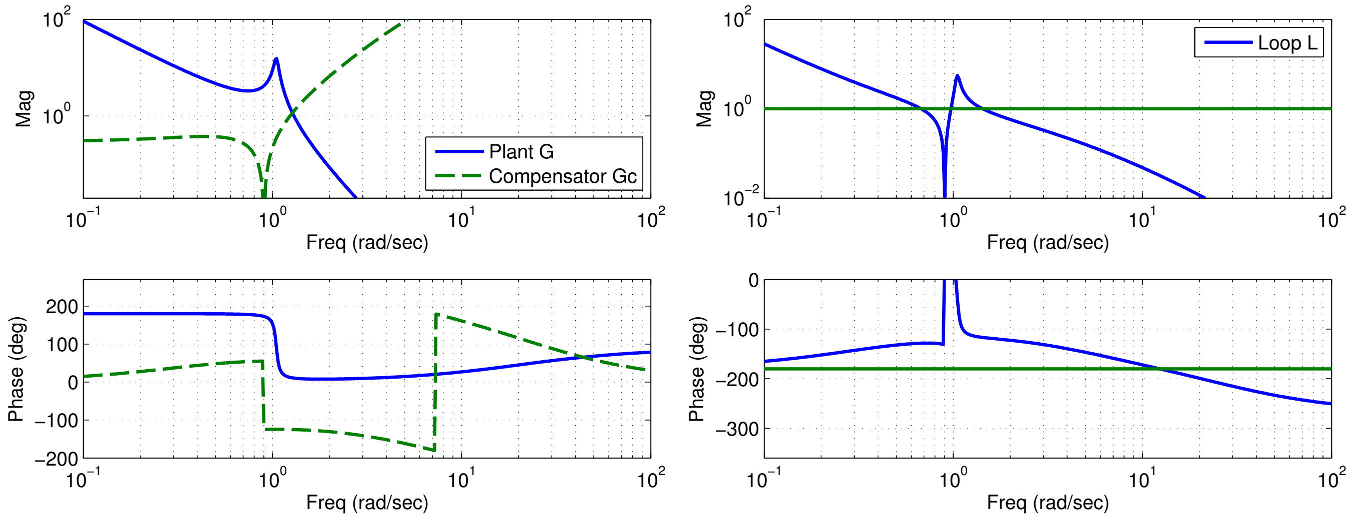
- Rigid body part easily controlled using a lead, such as

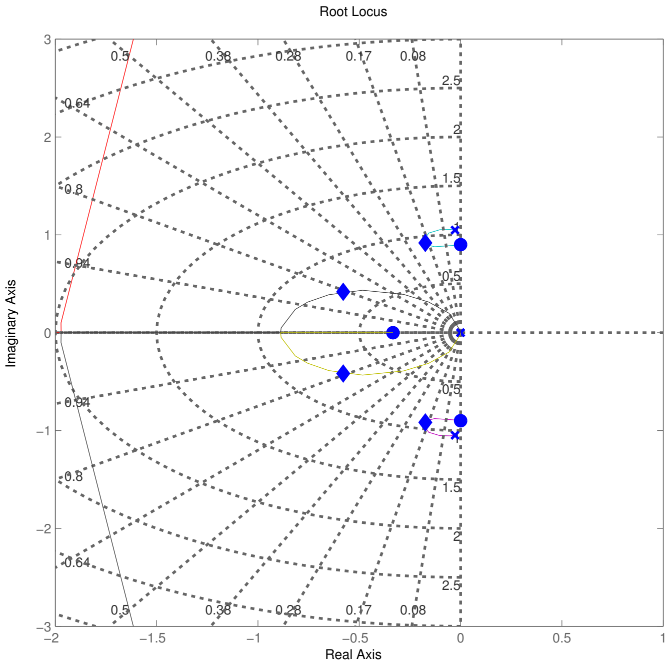
$$K_{lead}(s) = 1.5\frac{3s+1}{s+5}$$





- So problems with this "non-collocated" control is apparent – the flexible modes are driven unstable by the lead controller
- The problem with the lightly damped poles is that the loop phase changes very rapidly – get wrong departure angle for the root locus.
  - Can improve the design using notch filtering
- The notch has a lightly damped zero pair at a frequency just below the resonant frequency of the system, resulting in a rapid phase change, but in the opposite direction.
  - Result is that the departure angle now to the left, not the right.
  - Fixes instability problem - but damping of the resonant pole is quite limited – need to find a better location for the notch zeros.

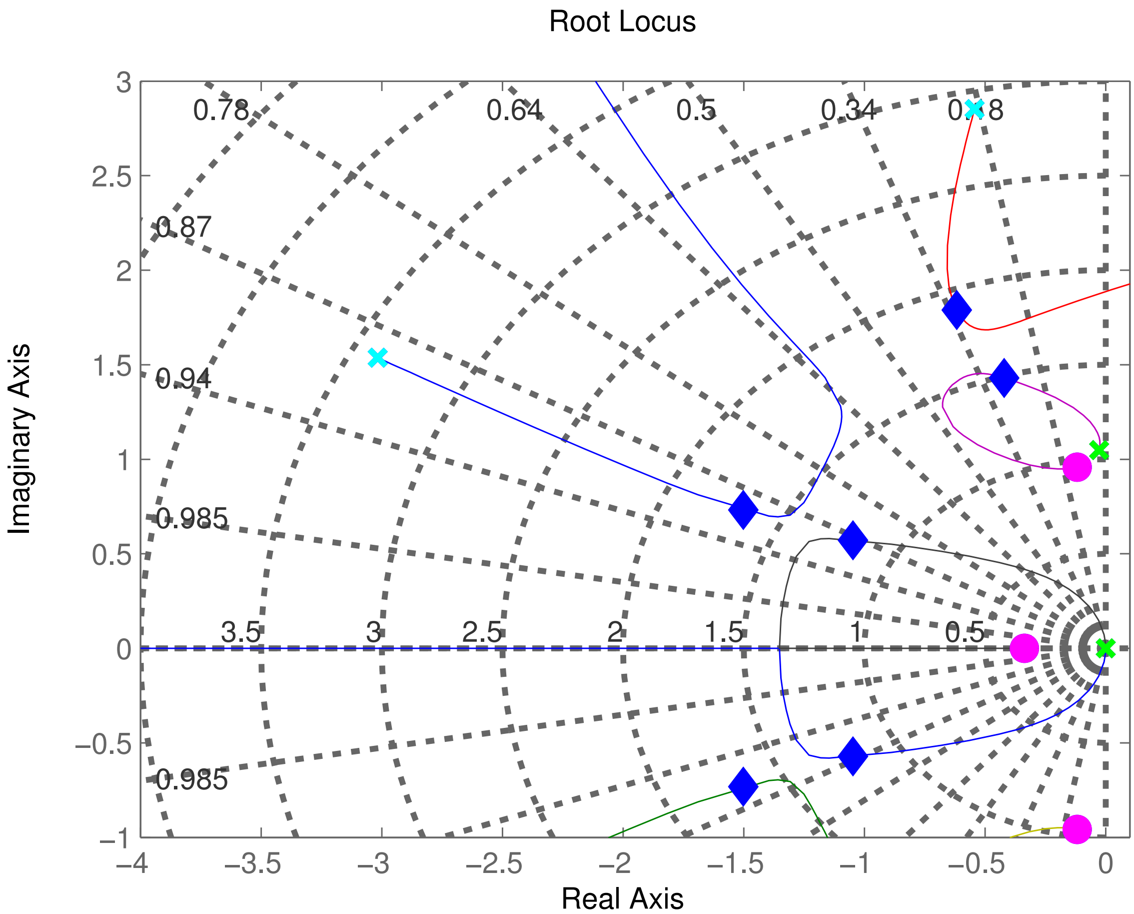




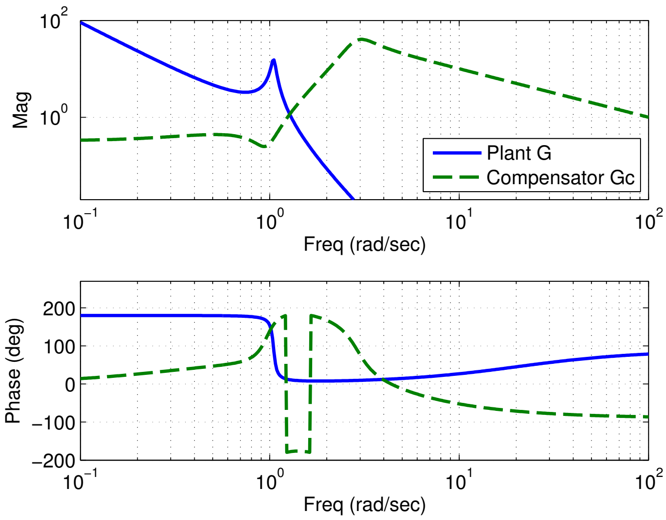
- Lets try to do it using DOFB – nothing particularly complicated here in the weights

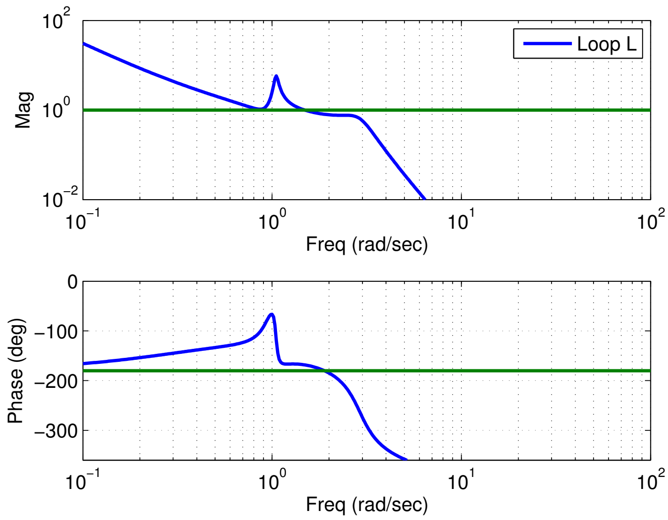
```
kk=0.1;bb=0.05*sqrt(kk/10);J1=1;J2=0.1;
a=[0 1 0 0;-kk/J2 -bb/J2 kk/J2 bb/J2;0 0 0 1;kk/J1 bb/J1 -kk/J1 -bb/J1];na=size(a,1);
b=[0 0 0 1/J1]';c=[1 0 0 0];d=0;
k=lqr(a,b,c'*c,.1);
l=lqr(a',c',b*b',.01);l=l'
%
% For state space for G_c(s)
%
ac=a-b*k-l*c;bc=l;cc=k;dc=0;
```

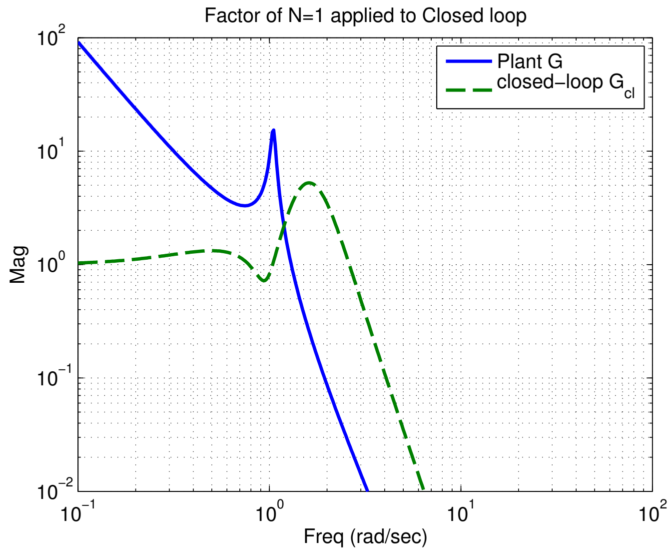
- Look at the resulting DOFB controller - × are the plant poles; ● are the compensator zeros; × are the compensator poles

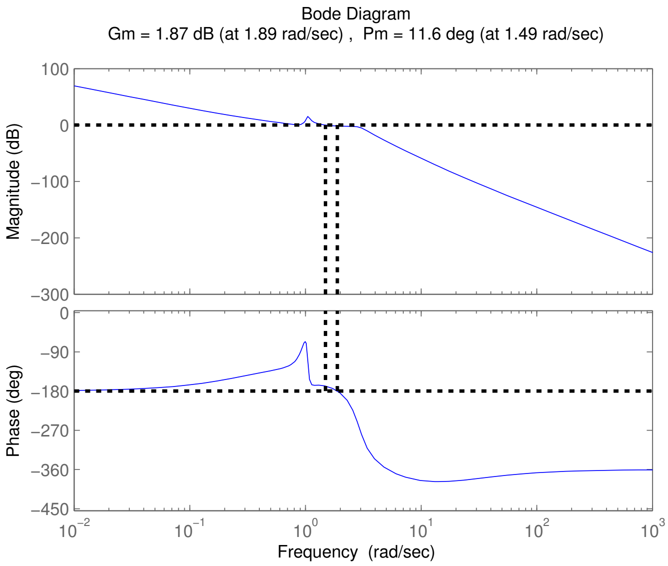


- So the DOFB controller puts zeros right near the plant poles – similar to the notch filter – ensures that the departure angle of the poles is to the left, not the right, so all the poles are stabilized.
  - Appears to achieve much higher levels of damping in that pole than the classical/notch filter design did.


Mag
Freq (rad/sec)
Plant G
Compensator Gc
Phase (deg)
Freq (rad/sec)


Mag
Loop L
Freq (rad/sec)
Phase (deg)
Freq (rad/sec)


Factor of N=1 applied to Closed loop
Mag
Plant G
closed–loop G_cl
Freq (rad/sec)


Bode Diagram
Gm = 1.87 dB (at 1.89 rad/sec) , Pm = 11.6 deg (at 1.49 rad/sec)
Magnitude (dB)
Phase (deg)
Frequency (rad/sec)

DOFB Summary

- Separation principle gives a very powerful and simple way to develop a dynamic output feedback controller

- Note that the designer now focuses on selecting the appropriate regulator and estimator pole locations.
  - Once those and the implementation architecture (i.e., how reference input is applied, as on 15–5) are set, the closed-loop response is specified.
  - Can almost consider the compensator to be a by-product.

- The examples show that the design process is extremely simple.