Topic #16

16.30/31 Feedback Control Systems

- Add reference inputs for the DOFB case

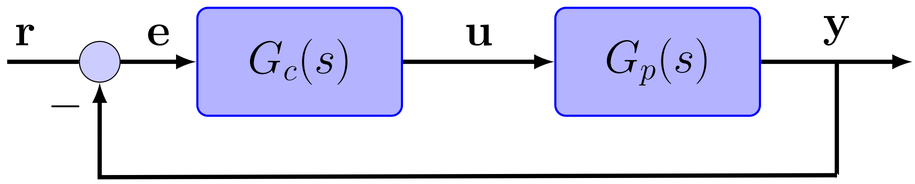
- Reading: FPE 7.8, 7.9

Reference Input - II

- On page 15-6, compensator implemented with reference command by changing to feedback on

$$\mathbf{e}(t) = \mathbf{r}(t) - \mathbf{y}(t)$$

rather than $-\mathbf{y}(t)$



  - So $\mathbf{u}(t) = G_c(s)\mathbf{e}(t) = G_c(s)(\mathbf{r}(t) - \mathbf{y}(t))$
  - Intuitively appealing because **same approach** used for classical control, but it turns out **not** to be the best.

- Can improve the implementation by using a more general form:

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

  - Now explicitly have two inputs to controller ($\mathbf{y}(t)$ and $\mathbf{r}(t)$)
  - $\overline{N}$ performs the same role that we used it for previously.
  - Introduce G as an extra degree of freedom in the problem.

- Turns out that setting $G = B\overline{N}$ is a particularly good choice.
  - Following presents some observations on the impact of G

- **First:** this generalization does not change the closed-loop poles of the system, regardless of the selection of G and $\overline{N}$, since

$$\begin{aligned}\dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + B\mathbf{u}(t) \quad , \quad \mathbf{y}(t) = C\mathbf{x}(t)\\ \dot{\mathbf{x}}_c(t) &= A_c\mathbf{x}_c(t) + B_c\mathbf{y}(t) + G\mathbf{r}(t)\\ \mathbf{u}(t) &= -C_c\mathbf{x}_c(t) + \overline{N}\mathbf{r}(t)\end{aligned}$$

$$\Rightarrow \quad \begin{bmatrix}\dot{\mathbf{x}}(t)\\ \dot{\mathbf{x}}_c(t)\end{bmatrix} = \begin{bmatrix}A & -BC_c\\ B_cC & A_c\end{bmatrix}\begin{bmatrix}\mathbf{x}(t)\\ \mathbf{x}_c(t)\end{bmatrix} + \begin{bmatrix}B\overline{N}\\ G\end{bmatrix}\mathbf{r}(t)$$

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

  - So the closed-loop poles are the eigenvalues of

$$\begin{bmatrix}A & -BC_c\\ B_cC & A_c\end{bmatrix}$$

    (same as 15–7 except "–" in a different place, gives same closed-loop eigenvalues) regardless of the choice of G and $\overline{N}$

  - G and $\overline{N}$ impact the forward path, not the feedback path

- **Second:** if $\overline{N} = 0$ and $G = -L = -B_c$, then we recover the `original implementation`, on 15–6 since the controller reduces to:

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

  - With $G_c(s) = C_c(sI - A_c)^{-1}B_c$, then this compensator can be written as $\mathbf{u}(t) = G_c(s)\mathbf{e}(t)$ as before (since the negative signs cancel).

- **Third:** Given this extra freedom, what is best way to use it?
  - One good objective is to select G and $\overline{N}$ so that the state estimation error is **independent** of $\mathbf{r}(t)$.
  - With this choice, changes in $\mathbf{r}(t)$ do not tend to cause such large transients in $\tilde{\mathbf{x}}(t)$
  - For this analysis, take $\tilde{\mathbf{x}}(t) = \mathbf{x}(t) - \mathbf{x}_c(t)$ since $\mathbf{x}_c(t) \equiv \hat{\mathbf{x}}(t)$

$$
\begin{aligned}
\dot{\tilde{\mathbf{x}}}(t) &= \dot{\mathbf{x}}(t) - \dot{\mathbf{x}}_c(t) \\
&= A\mathbf{x}(t) + B\mathbf{u}(t) - (A_c\mathbf{x}_c(t) + B_c\mathbf{y}(t) + G\mathbf{r}(t)) \\
&= A\mathbf{x}(t) + B(-C_c\mathbf{x}_c(t) + \overline{N}\mathbf{r}(t)) \\
&\quad - (\{A - BC_c - B_cC\}\mathbf{x}_c(t) + B_cC\mathbf{x}(t) + G\mathbf{r}(t))
\end{aligned}
$$

So

$$
\begin{aligned}
\dot{\tilde{\mathbf{x}}}(t) &= A\mathbf{x}(t) + B(\overline{N}\mathbf{r}(t)) - (\{A - B_cC\}\mathbf{x}_c(t) + B_cC\mathbf{x}(t) + G\mathbf{r}(t)) \\
&= (A - B_cC)\mathbf{x}(t) + B\overline{N}\mathbf{r}(t) - (\{A - B_cC\}\mathbf{x}_c(t) + G\mathbf{r}(t)) \\
&= (A - B_cC)\tilde{\mathbf{x}}(t) + B\overline{N}\mathbf{r}(t) - G\mathbf{r}(t) \\
&= (A - B_cC)\tilde{\mathbf{x}}(t) + (B\overline{N} - G)\mathbf{r}(t)
\end{aligned}
$$

- Thus we can eliminate the effect of $\mathbf{r}(t)$ on $\tilde{\mathbf{x}}(t)$ by setting

$$
\boxed{G \equiv B\overline{N}}
$$

- With this choice, the controller:

$$\begin{aligned}\dot{\mathbf{x}}_c(t) &= (A - BK - LC)\mathbf{x}_c(t) + L\mathbf{y}(t) + B\overline{N}\mathbf{r}(t)\\ \mathbf{u}(t) &= -K\mathbf{x}_c(t) + \overline{N}\mathbf{r}(t)\end{aligned}$$

  can be rewritten as:

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

  - So the control is computed using the reference before it is applied, and that control is applied to both system and estimator.

- **Fourth:** if this generalization does not change the closed-loop poles of the system, then what does it change?
  - Recall that zeros of SISO y/r transfer function solve:

$$\texttt{general}\qquad \det\left[\begin{array}{cc|c} sI - A & BC_c & -B\overline{N} \\ -B_cC & sI - A_c & -G \\ \hline C & 0 & 0 \end{array}\right] = 0$$

$$\texttt{original}\qquad \det\left[\begin{array}{cc|c} sI - A & BC_c & 0 \\ -B_cC & sI - A_c & B_c \\ \hline C & 0 & 0 \end{array}\right] = 0$$

$$\texttt{new}\qquad \det\left[\begin{array}{cc|c} sI - A & BC_c & -B\overline{N} \\ -B_cC & sI - A_c & -B\overline{N} \\ \hline C & 0 & 0 \end{array}\right] = 0$$

- Hard to see how this helps, but consider the scalar `new` case [14]:

$$
\begin{aligned}
\Rightarrow \quad C(-BC_cB\overline{N} + (sI - A_c)B\overline{N}) &= 0 \\
-CB\overline{N}(BC_c - (sI - [A - BC_c - B_cC])) &= 0 \\
CB\overline{N}(sI - [A - B_cC]) &= 0
\end{aligned}
$$

  - So zero of the y/r path is the root of $sI - [A - B_cC] = 0$, which is the pole of the estimator.
  - So by setting $G = B\overline{N}$ as in the `new` case, the estimator dynamics are canceled out of the response of the system to a reference command.
  - Cancelation does not occur with `original` implementation.
  - SISO, multi-dimensional state case handled in the appendix.

- So in summary, if the SISO system is $G(s) = b(s)/a(s)$, then with DOFB control, the closed-loop transfer function will be of the form:

$$
T(s) = \frac{Y(s)}{R(s)} = K_g \frac{\gamma(s)b(s)}{\Phi_e(s)\Phi_c(s)}
$$

  - Designer determines $\Phi_c(s)$ and $\Phi_e(s)$
  - Plant zeroes $b(s)$ remain, unless canceled out
  - Many choices for $\gamma(s)$ – the one called `new` sets it to

$$
\gamma(s) \equiv \Phi_e(s)
$$

  to cancel out the estimator dynamics from the response.

[14] no plant zeros in this case

- **Fifth:** select $\overline{N}$ to ensure that the steady-state error is zero.
  - As before, this can be done by selecting $\overline{N}$ so that the DC gain of the closed-loop y/r transfer function is 1.

$$\left.\frac{y}{r}\right|_{DC} \triangleq -\begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} A & -BC_c \\ B_cC & A_c \end{bmatrix}^{-1} \begin{bmatrix} B \\ B \end{bmatrix} \overline{N} = I$$

- **The `new` implementation of the compensator is**

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

  - Which has two separate inputs $\mathbf{y}(t)$ and $\mathbf{r}(t)$
  - Selection of $\overline{N}$ ensures good steady-state performance
  - `new` implementation gives better transient performance.

Appendix: Zero Calculation

- Calculation on 16–5 requires we find (assume r and y scalars)

$$\det \begin{bmatrix} sI - A_{cl} & B_{cl} \\ C_{cl} & 0 \end{bmatrix} = \det(sI - A_{cl}) \det(0 - C_{cl}(sI - A_{cl})^{-1} B_{cl})$$
$$= \Phi_e(s)\Phi_c(s) \det(-C_{cl}(sI - A_{cl})^{-1} B_{cl})$$
$$= -\Phi_e(s)\Phi_c(s)\ C_{cl}(sI - A_{cl})^{-1} B_{cl}$$

- But finding $(sI - A_{cl})^{-1}$ can be tricky – it is simplified by the following steps.
  - First note if $T = \begin{bmatrix} I & 0 \\ I & -I \end{bmatrix}$ so that $T = T^{-1}$, then

$$(sI - A_{cl})^{-1} = TT^{-1}(sI - A_{cl})^{-1}TT^{-1}$$
$$= T(sI - TA_{cl}T)^{-1}T$$

where

$$TA_{cl}T \triangleq \bar{A}_{cl} = \begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix}$$

$$\Rightarrow (sI - A_{cl})^{-1} = T \begin{bmatrix} sI - (A - BK) & -BK \\ 0 & sI - (A - LC) \end{bmatrix}^{-1} T$$

- Now use fact that $\begin{bmatrix} F & H \\ 0 & G \end{bmatrix}^{-1} = \begin{bmatrix} F^{-1} & -F^{-1}HG^{-1} \\ 0 & G^{-1} \end{bmatrix}$
  - To get that

$$(sI - A_{cl})^{-1} = T \begin{bmatrix} (sI - (A - BK))^{-1} & (sI - (A - BK))^{-1}BK(sI - (A - LC))^{-1} \\ 0 & (sI - (A - LC))^{-1} \end{bmatrix} T$$

- Now consider `new` case:

$$
\begin{aligned}
& \begin{bmatrix} C & 0 \end{bmatrix} (sI - A_{cl})^{-1} \begin{bmatrix} -B\overline{N} \\ -B\overline{N} \end{bmatrix} \\
= & \begin{bmatrix} C & 0 \end{bmatrix} T \begin{bmatrix} (sI-(A-BK))^{-1} & (sI-(A-BK))^{-1}BK(sI-(A-LC))^{-1} \\ 0 & (sI-(A-LC))^{-1} \end{bmatrix} T \begin{bmatrix} -B\overline{N} \\ -B\overline{N} \end{bmatrix} \\
= & \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} (sI-(A-BK))^{-1} & (sI-(A-BK))^{-1}BK(sI-(A-LC))^{-1} \\ 0 & (sI-(A-LC))^{-1} \end{bmatrix} \begin{bmatrix} -B\overline{N} \\ 0 \end{bmatrix} \\
= & -C(sI-(A-BK))^{-1}B\overline{N} = -C\texttt{adj}[sI-(A-BK)]B\overline{N}\Phi_c(s)^{-1}
\end{aligned}
$$

thus in the `new` case,

$$
\begin{aligned}
\det \begin{bmatrix} sI - A_{cl} & B_{cl} \\ C_{cl} & 0 \end{bmatrix} &= -\Phi_e(s)\Phi_c(s) \ \det(C_{cl}(sI-A_{cl})^{-1}B_{cl}) \\
&= \Phi_e(s)\Phi_c(s) \ C\texttt{adj}[sI-(A-BK)]B\overline{N}\Phi_c(s)^{-1} \\
&= \Phi_e(s) \ C\texttt{adj}[sI-(A-BK)]B\overline{N}
\end{aligned}
$$

- Whereas in the `original` case, we get:

$$
\begin{aligned}
& \begin{bmatrix} C & 0 \end{bmatrix} (sI - A_{cl})^{-1} \begin{bmatrix} 0 \\ L \end{bmatrix} \\
= & \begin{bmatrix} C & 0 \end{bmatrix} T \begin{bmatrix} (sI-(A-BK))^{-1} & (sI-(A-BK))^{-1}BK(sI-(A-LC))^{-1} \\ 0 & (sI-(A-LC))^{-1} \end{bmatrix} T \begin{bmatrix} 0 \\ L \end{bmatrix} \\
= & \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} (sI-(A-BK))^{-1} & (sI-(A-BK))^{-1}BK(sI-(A-LC))^{-1} \\ 0 & (sI-(A-LC))^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ -L \end{bmatrix} \\
= & -C(sI-(A-BK))^{-1}BK(sI-(A-LC))^{-1}L \\
= & -C\texttt{adj}[sI-(A-BK)] \ BK \ \texttt{adj}[sI-(A-LC)]L\Phi_e(s)^{-1}\Phi_c(s)^{-1}
\end{aligned}
$$

thus in the `original` case,

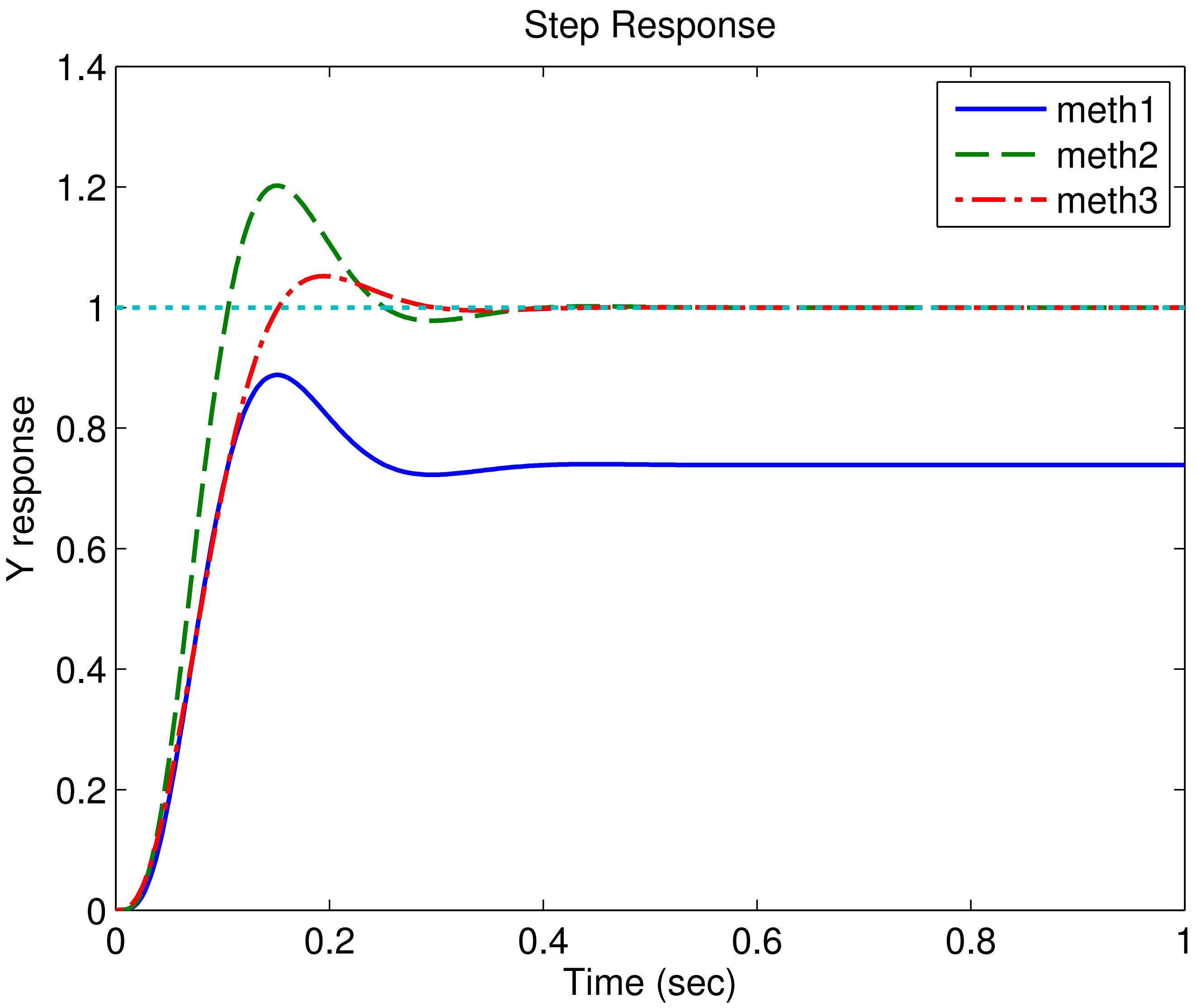
$$
\begin{aligned}
& \det \begin{bmatrix} sI - A_{cl} & B_{cl} \\ C_{cl} & 0 \end{bmatrix} = -\Phi_e(s)\Phi_c(s) \ \det(C_{cl}(sI-A_{cl})^{-1}B_{cl}) \\
= & -\Phi_e(s)\Phi_c(s) \ C\texttt{adj}[sI-(A-BK)] \ BK \ \texttt{adj}[sI-(A-LC)]L\Phi_e(s)^{-1}\Phi_c(s)^{-1} \\
= & C\texttt{adj}[sI-(A-BK)] \ BK \ \texttt{adj}[sI-(A-LC)]L
\end{aligned}
$$

- So `new` case has $\Phi_e(s)$ in numerator, which cancels the zero dynamics out of the closed-loop response, but the `original` case does not.

FF Example 1

$$G(s) = \frac{8 \cdot 14 \cdot 20}{(s+8)(s+14)(s+20)}$$

- Method #1: `original` implementation.
- Method #2: `original`, with the reference input scaled to ensure that the DC gain of $y/r|_{DC} = 1$.
- Method #3: `new` implementation with both $G = B\overline{N}$ and $\overline{N}$ selected.

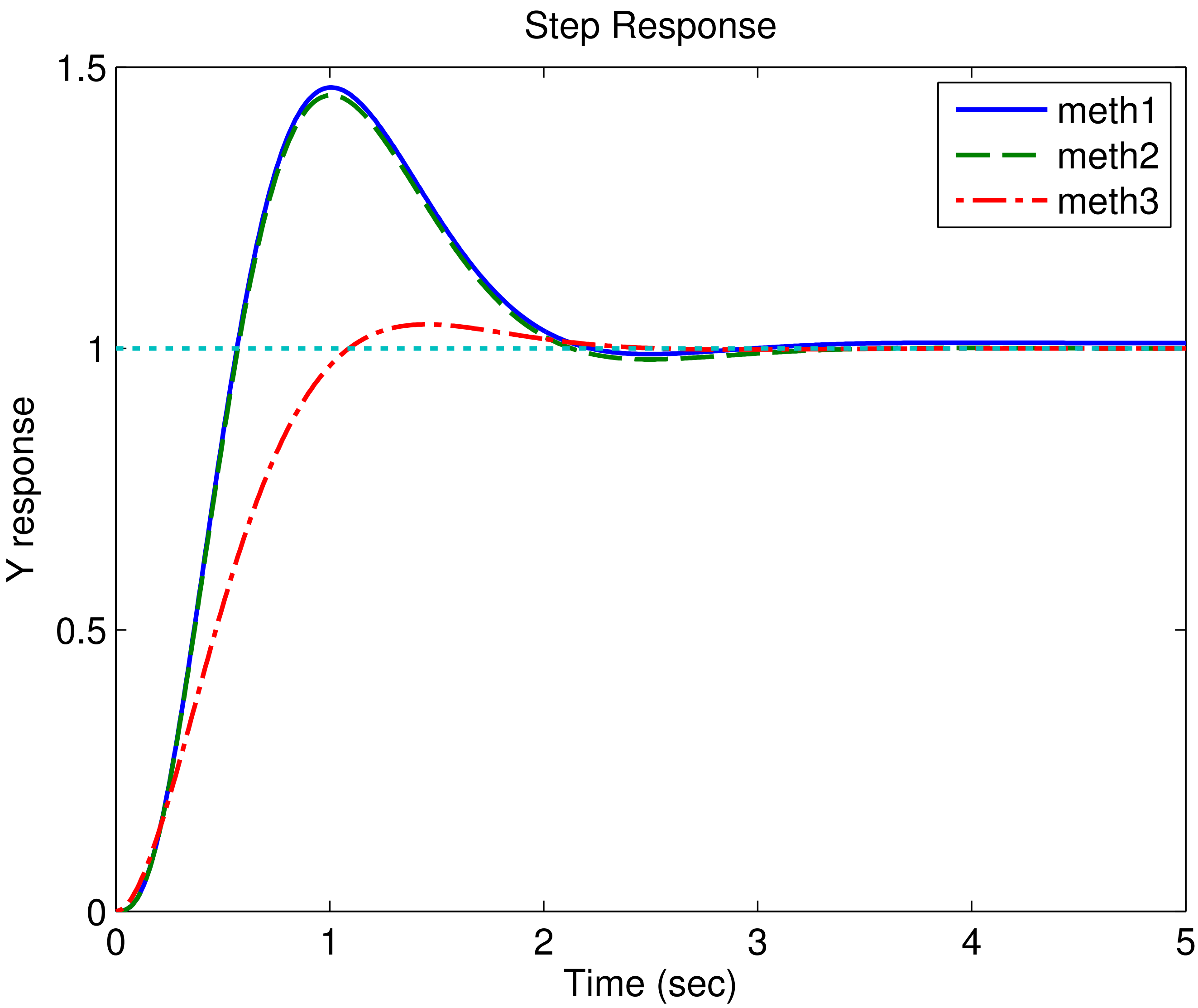


- New method (#3) shows less transient impact of applying $r(t)$.

FF Example 2

$$G(s) = \frac{0.94}{s^2 - 0.0297}$$

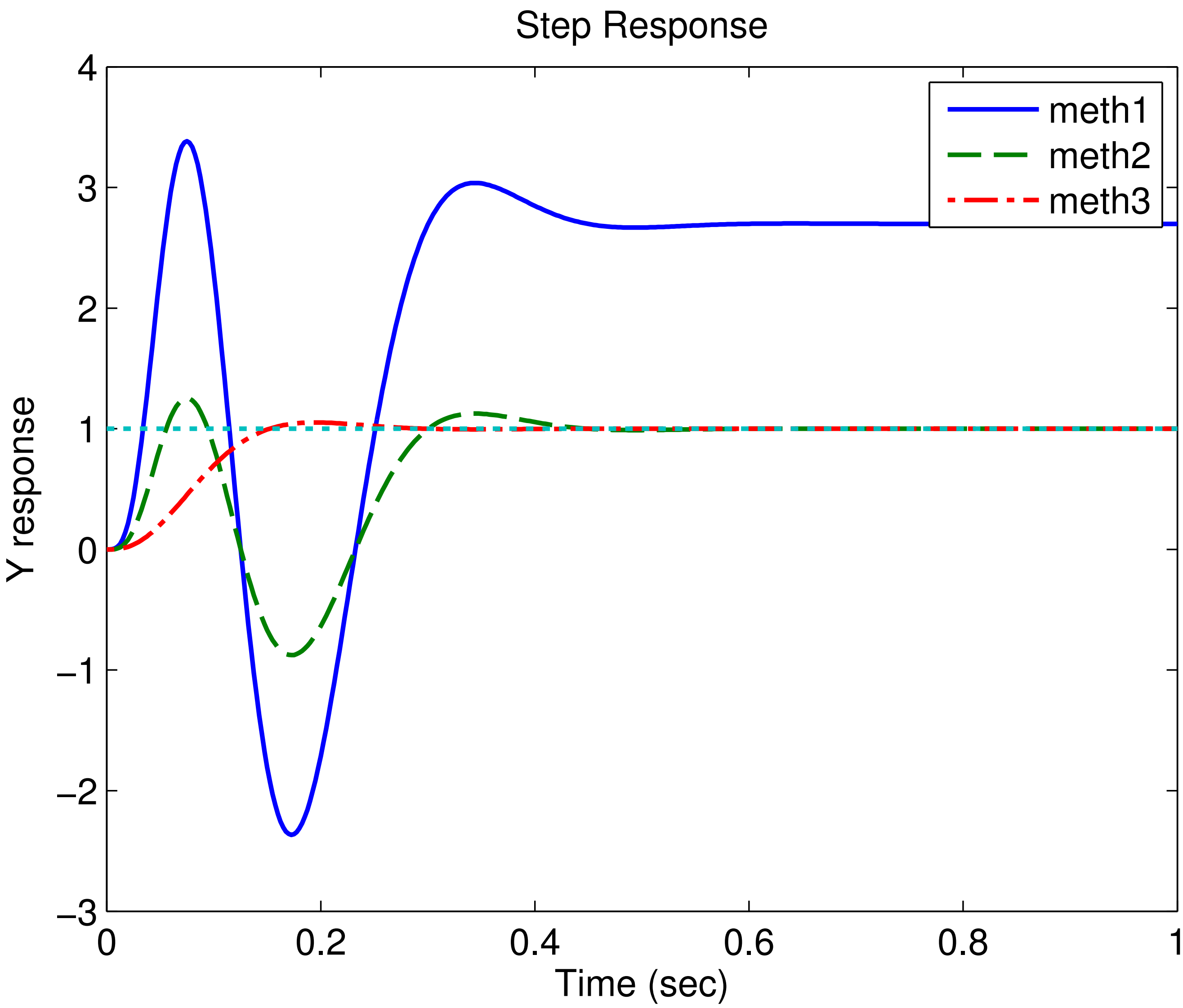
- Method #1: `original` implementation.
- Method #2: `original`, with the reference input scaled to ensure that the DC gain of $y/r|_{DC} = 1$.
- Method #3: `new` implementation with both $G = B\overline{N}$ and $\overline{N}$ selected.



FF Example 3

$$G(s) = \frac{8 \cdot 14 \cdot 20}{(s-8)(s-14)(s-20)}$$

- Method #1: `original` implementation.
- Method #2: `original`, with the reference input scaled to ensure that the DC gain of $y/r|_{DC} = 1$.
- Method #3: `new` implementation with both $G = B\overline{N}$ and $\overline{N}$ selected.

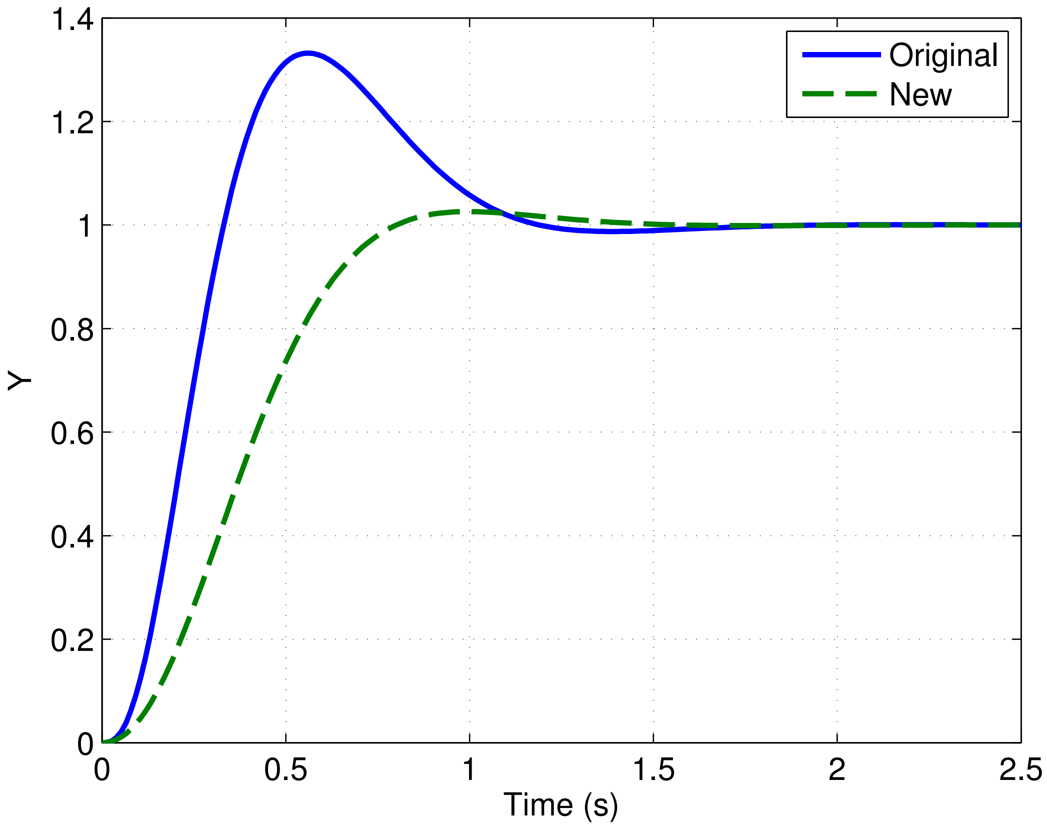


FF Example 4

- Revisit example on 15-9, with $G(s) = 1/(s^2 + s + 1)$ and regulator target poles at $-4 \pm 4j$ and estimator target poles at -10.
  - So dominant poles have $\omega_n \approx 5.5$ and $\zeta = 0.707$
- From these closed-loop dynamics, would expect to see

10-90% rise time	$t_r = \dfrac{1 + 1.1\zeta + 1.4\zeta^2}{\omega_n} = 0.44s$
Settling time (5%)	$t_s = \dfrac{3}{\zeta\omega_n} = 0.75s$
Time to peak amplitude	$t_p = \dfrac{\pi}{\omega_n\sqrt{1-\zeta^2}} = 0.79s$
Peak overshoot	$M_p = e^{-\zeta\omega_n t_p} = 0.04$

- Now compare step responses of `new` and `original` implementations



- Hopefully it is clear that the `new` implementation meets the expected criteria quite well - and certainly **much better** than the `original`

Code: Dynamic Output Feedback FF Examples

```
% Code for Topic 16 in 2010
% Examples of dynamic output feedback
% Jonathan How
%
close all;
% clear all;for model=1:5;dofb_examp2;end
fig=0;
% system
switch model
 case 1
  G=tf(8*14*20,conv([1 8],conv([1 14],[1 20])));name='nexamp1';
  tt=[-20 0 -10 10]*2.5;
  t=[0:.0025:1];
 case 2
   G=tf(8*14*20,conv([1 -8],conv([1 -14],[1 -20])));name='nexamp2';
   tt=[-10 10 -10 10]*2.5;
   t=[0:.0025:1];
 case 3
  G=tf(.94,[1 0 -0.0297]);name='nexamp3';
  tt=[-10 10 -10 10]/5;
  t=[0:.01:5];
 case 4
  G=tf([1 -1],conv([1 1],[1 -3]));name='nexamp4';
  tt=[-10 10 -10 10];
  t=[0:.01:10];
 case 5
  G=tf(conv([1 -2],[1 -4]),conv(conv([1 -1],[1 -3]),[1 2*.2*2 2^2 0 0]));
  name='examp5';
 case 6
  G=tf(8*14*20,conv([1 8],conv([1 14],[1 20])));name='nexamp1h';
  tt=[-20 0 -10 10]*2.5;
  t=[0:.0001:1];
 otherwise
  return
end

%%%
[a,b,c,d]=ssdata(G);na=length(a);
%
if model == 6
    R=.00001;
else
    R=.01;
end
% choose the regulator poles using LQR 12-2
[k,P,reg_poles]=lqr(a,b,c'*c,R);
%hold on
%plot(reg_poles+j*eps,'md','MarkerSize',12,'MarkerFaceColor','m')
%hold off

% design the estimator by doubling the real part of the regulator poles
PP=2*real(reg_poles)+imag(reg_poles)*j;
% see 14-13
ke=place(a',c',PP);l=ke';
% now form compensator see 14-4
ac=a-b*k-l*c;bc=l;cc=k;dc=0;
% see 15-6
acl=[a -b*cc;bc*c ac];ccl=[c,zeros(1,na)];dcl=0;
% standard case
bcl1=[zeros(na,1);-bc];
%scale up the ref path so that at least the Gcl(s) has unity DC gain
bcl2=[zeros(na,1);-bc]; % original implementation on 16-24
scale=ccl*inv(-acl)*bcl2;
bcl2=bcl2/scale;
%
bcl3=[b;b]; % new implementation on 14-23
scale=ccl*inv(-acl)*bcl3;
bcl3=bcl3/scale;
%
Gcl1=ss(acl,bcl1,ccl,dcl); % original implementation (14-5) unscaled
Gcl2=ss(acl,bcl2,ccl,dcl); % original implementation (14-5) scaled
Gcl3=ss(acl,bcl3,ccl,dcl); % new implementation on (14-23) scaled
Gc=ss(ac,bc,cc,dc);

fig=fig+1;figure(fig);clf;
f=logspace(-2,3,400);j=sqrt(-1);
```

```
g=freqresp(G,f*j);g=squeeze(g);
gc=freqresp(Gc,f*j);gc=squeeze(gc);
gcl1=freqresp(Gcl1,f*j);gcl1=squeeze(gcl1);
gcl2=freqresp(Gcl2,f*j);gcl2=squeeze(gcl2);
gcl3=freqresp(Gcl3,f*j);gcl3=squeeze(gcl3);

figure(fig);fig=fig+1;clf
orient tall
subplot(211)
loglog(f,abs(g),f,abs(gc),'--','LineWidth',2);axis([.1 1e3 1e-2 500])
xlabel('Freq (rad/sec)');ylabel('Mag')
legend('G','G_c','Location','Southeast');grid
subplot(212)
semilogx(f,180/pi*unwrap(angle(g)),f,180/pi*unwrap(angle(gc)),'--','LineWidth',2);
axis([.1 1e3 -200 50])
xlabel('Freq (rad/sec)');ylabel('Phase (deg)');grid
legend('G','G_c','Location','SouthWest')

L=g.*gc;

figure(fig);fig=fig+1;clf
orient tall
subplot(211)
loglog(f,abs(L),[.1 1e3],[1 1],'LineWidth',2);axis([.1 1e3 1e-2 10])
xlabel('Freq (rad/sec)');ylabel('Mag')
legend('Loop L');
grid
subplot(212)
semilogx(f,180/pi*phase(L.'),[.1 1e3],-180*[1 1],'LineWidth',2);
axis([.1 1e3 -290 -0])
xlabel('Freq (rad/sec)');ylabel('Phase (deg)');grid
%
% loop dynamics L = G Gc
% see 15-6
al=[a b*cc;zeros(na) ac];bl=[zeros(na,1);bc];cl=[c zeros(1,na)];dl=0;

figure(fig);fig=fig+1;clf
margin(al,bl,cl,dl);
figure(fig);fig=fig+1;clf
rlocus(al,bl,cl,dl)
hold on;
plot(eig(acl)+eps*j,'bd','MarkerFaceColor','b')
plot(eig(a)+eps*j,'mv','MarkerFaceColor','m')
plot(eig(ac)+eps*j,'rs','MarkerSize',9,'MarkerFaceColor','r')
plot(tzero(ac,bc,cc,dc)+eps*j,'ro','MarkerFaceColor','r')
hold off;grid on
%
% closed-loop freq response
%
figure(15);clf
loglog(f,abs(g),f,abs(gcl1),f,abs(gcl2),'--','LineWidth',2);
axis([.1 1e3 .01 1e2])
xlabel('Freq (rad/sec)');ylabel('Mag')
legend('Plant G','Gcl unscaled','Gcl scaled');grid

figure(fig);fig=fig+1;clf
loglog(f,abs(g),f,abs(gcl1),f,abs(gcl2),f,abs(gcl3),'--','LineWidth',2);
axis([.1 1e3 .01 1e2])
xlabel('Freq (rad/sec)');ylabel('Mag')
legend('Plant G','Gcl1','Gcl2','Gcl3');grid

ZZ=-1+.1*exp(j*[0:.01:1]*2*pi);
figure(fig);fig=fig+1;clf
semilogy(unwrap(angle(L))*180/pi,abs(L))
hold on;
semilogy(unwrap(angle(ZZ))*180/pi-360,abs(ZZ),'g-')
hold off
axis([-270 -100 .1 10])
hold on;plot(-180,1,'rx');hold off
if max(real(eig(a))) > 0
    title('Nichols: Unstable Open-loop System')
else
    title('Nichols: Stable Open-loop System')
end
ylabel('Mag');xlabel('Phase (deg)')

figure(fig);fig=fig+1;clf
```

```
plot(unwrap(angle(L))*180/pi,abs(L),'LineWidth',1.5)
hold on
plot(unwrap(angle(L))*180/pi,.95*abs(L),'r-.','LineWidth',1.5)
plot(unwrap(angle(L))*180/pi,1.05*abs(L),'m:','LineWidth',1.5)
plot(unwrap(angle(ZZ))*180/pi,abs(ZZ),'g-')
plot(-180,1,'rx');hold off
hold on;
semilogy(unwrap(angle(ZZ))*180/pi-360,abs(ZZ),'g-')
hold off
legend('1','0.95','1.05')
axis([-195 -165 .5 1.5])
if max(real(eig(a))) > 0
    title('Nichols: Unstable Open-loop System')
else
    title('Nichols: Stable Open-loop System')
end
ylabel('Mag');xlabel('Phase (deg)')

figure(fig);fig=fig+1;clf
loglog(f,abs(1./(1+L)),f,abs(L),'--','LineWidth',1.5);grid
title('Sensitivity Plot')
legend('|S|','|L|')
xlabel('Freq (rad/sec)');ylabel('Mag')
axis([.1 1e3 1e-2 100])

figure(fig);fig=fig+1;clf
[y1,t]=step(Gcl1,t);
[y2,t]=step(Gcl2,t);
[y3,t]=step(Gcl3,t);
plot(t,y1,t,y2,t,y3,t,ones(size(t)),'--','LineWidth',1.5)
setlines(1.5)
legend('meth1','meth2','meth3')
title('Step Response')
xlabel('Time (sec)');ylabel('Y response')

for ii=[1:gcf 15]
  eval(['figure(',num2str(ii),'); export_fig ',name,'_',num2str(ii),' -pdf -a1'])
  if ii==4;
    figure(ii);axis(tt)
    eval(['export_fig ',name,'_',num2str(ii),'a -pdf'])
 end
end

eval(['save ',name,' R G Gc Gcl1 Gcl2 Gcl3 k l P PP'])

return
```

16.30 / 16.31 Feedback Control Systems
Fall 2010