Topic #12

16.30/31 Feedback Control Systems

State-Space Systems
• Full-state Feedback Control
• How do we change location of state-space eigenvalues/poles?
• Or, if we can change the pole locations where do we put the poles?
  • Heuristics
  • Linear Quadratic Regulator
• How well does this approach work?

• Reading: FPE 7.4
Pole Placement Approach

- So far we have looked at how to pick $K$ to get the dynamics to have some nice properties (i.e. stabilize $A$)

$$\lambda_i(A) \sim \lambda_i(A - BK)$$

- **Question:** where should we put the closed-loop poles?

- **Approach #1:** use time-domain specifications to locate dominant poles – roots of:

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$$

- Then place rest of the poles so they are “much faster” than the dominant 2nd order behavior.

- Example: could keep the same damped frequency $w_d$ and then move the real part to be 2–3 times faster than the real part of dominant poles $\zeta\omega_n$

  ♦ Just be careful moving the poles too far to the left because it takes a lot of control effort

- Recall ROT for 2nd order response (4–??):

  10-90% rise time
  $$t_r = \frac{1 + 1.1\zeta + 1.4\zeta^2}{\omega_n}$$

  Settling time (5%)
  $$t_s = \frac{3}{\zeta\omega_n}$$

  Time to peak amplitude
  $$t_p = \frac{\pi}{\omega_n\sqrt{1 - \zeta^2}}$$

  Peak overshoot
  $$M_p = e^{-\zeta\omega_nt_p}$$

- **Key difference** in this case: since all poles are being placed, the assumption of dominant 2nd order behavior is pretty much guaranteed to be valid.
Linear Quadratic Regulator

- **Approach #2:** is to place the pole locations so that the closed-loop system optimizes the cost function

\[ J_{LQR} = \int_0^\infty \left[ x(t)^T Q x(t) + u(t)^T R u(t) \right] dt \]

where:

- \( x^T Q x \) is the **State Cost** with weight \( Q \)
- \( u^T R u \) is called the **Control Cost** with weight \( R \)
- Basic form of **Linear Quadratic Regulator** problem.

- MIMO optimal control is a time invariant linear state feedback

\[ u(t) = -K_{lqr} x(t) \]

and \( K_{lqr} \) found by solving **Algebraic Riccati Equation** (ARE)

\[ 0 = A^T P + P A + Q - P B R^{-1} B^T P \]

\[ K_{lqr} = R^{-1} B^T P \]

- Some details to follow, but discussed at length in 16.323

- **Note:** state cost written using output \( x^T Q x \), but could define a system output of interest \( z = C_z x \) that is not based on a physical sensor measurement and use cost ftn:

\[ \Rightarrow J_{LQR} = \int_0^\infty \left[ x^T(t) C_z^T \tilde{Q} C_z x(t) + \rho u(t)^T u(t) \right] dt \]

- Then effectively have state penalty \( Q = (C_z^T \tilde{Q} C_z) \)

- Selection of \( z \) used to isolate system states of particular interest that you would like to be regulated to “zero”.

- \( R = \rho I \) effectively sets the controller bandwidth
Fig. 1: Example #1: \( G(s) = \frac{8\cdot14\cdot20}{(s+8)(s+14)(s+20)} \) with control penalty \( \rho \) and \( 10\rho \)
Fig. 2: Example #2: \( G(s) = \frac{0.94}{s^2 - 0.0297} \) with control penalty \( \rho \) and \( 10\rho \)

Step Response

\[
\begin{align*}
\text{Y output} & \quad \text{time (sec)} \\
0 & \quad 0 \quad 5 \quad 10 \quad 15 \quad 20 \quad 25 \quad 30
\end{align*}
\]

\[ N_{\text{bar}} = 3.1624 \]

\[ K = 1.22523 \quad 12.0096 \]

Closed-loop Freq Response

\[
\begin{align*}
G_c^d & \quad \text{Freq (rad/sec)} \\
10^{-1} & \quad 10^0 \quad 10^1 \quad 10^2
\end{align*}
\]

\[ \rho = 0.1 \quad \rho = 1 \]
Fig. 3: Example #3: $G(s) = \frac{8 \cdot 14 \cdot 20}{(s-8)(s-14)(s-20)}$ with control penalty $\rho$ and $10\rho$
Fig. 4: Example #4: $G(s) = \frac{(s-1)}{(s+1)(s-3)}$ with control penalty $\rho$ and $10\rho$
Fig. 5: Example #5: \( G(s) = \frac{(s-2)(s-4)}{(s-1)(s-3)(s^2+0.8s+4)s^2} \) with control penalty \( \rho \) and \( 10\rho \).
LQR Weight Matrix Selection

- Good ROT (typically called Bryson’s Rules) when selecting the weighting matrices $Q$ and $R$ is to normalize the signals:

$$Q = \begin{bmatrix}
\frac{\alpha_1^2}{(x_1)_{\text{max}}^2} \\
\frac{\alpha_2^2}{(x_2)_{\text{max}}^2} \\
\vdots \\
\frac{\alpha_n^2}{(x_n)_{\text{max}}^2}
\end{bmatrix}$$

$$R = \rho \begin{bmatrix}
\frac{\beta_1^2}{(u_1)_{\text{max}}^2} \\
\frac{\beta_2^2}{(u_2)_{\text{max}}^2} \\
\vdots \\
\frac{\beta_m^2}{(u_m)_{\text{max}}^2}
\end{bmatrix}$$

- The $(x_i)_{\text{max}}$ and $(u_i)_{\text{max}}$ represent the largest desired response or control input for that component of the state/actuator signal.

- $\sum_i \alpha_i^2 = 1$ and $\sum_i \beta_i^2 = 1$ are used to add an additional relative weighting on the various components of the state/control

- $\rho$ is used as the last relative weighting between the control and state penalties ⇒ gives a relatively concrete way to discuss the relative size of $Q$ and $R$ and their ratio $Q/R$
Regulator Summary

- Dominant second order approach places the closed-loop pole locations with no regard to the amount of control effort required.
  - Designer must iterate on the selected bandwidth ($\omega_n$) to ensure that the control effort is reasonable.

- LQR selects closed-loop poles that balance between state errors and control effort.
  - Easy design iteration using $R$
  - Sometimes difficult to relate the desired transient response to the LQR cost function.
  - Key thing is that the designer is focused on system performance issues rather than the mechanics of the design process.
% LQR examples for 16.31
% Fall 2010
% Jonathan How, MIT
%
close all; clear all
set(0, 'DefaultLineLineWidth', 2);
set(0, 'DefaultLineMarkerSize', 8); set(0, 'DefaultLineMarkerFace', 'b')
set(0, 'DefaultAxesFontSize', 12); set(0, 'DefaultTextFontSize', 12)
set(0, 'DefaultAxesFontName', 'ariel'); set(0, 'DefaultTextFontName', 'arial')
set(0, 'DefaultFigureColor', 'w', 'DefaultAxesColor', 'w', ...
'DefaultAxesXColor', 'k', 'DefaultAxesYColor', 'k', 'k', ...
'DefaultAxes2Color', 'k', 'DefaultAxesColor', 'k')

if 1

% system
G = tf(8*14*20, conv([1 8], conv([1 14],[1 20])));
[a,b,c,d] = ssdata(G);
R = 1e-3;
k1 = lqr(a,b,c'*c,R); % nominal controller
k2 = lqr(a,b,c'*c,10*R); % slower control bec of higher control penalty

% find the feedback gains
N = inv(-c*inv(a-b*k)*b);
N2 = inv(-c*inv(a-b*k2)*b);
sys1 = ss(a-b*k,b*N,c,d);
sys2 = ss(a-b*k2,b*N2,c,d);
t = [0:.005:1];
[y,t,x] = step(sys1,t);
[y2,t2,x2] = step(sys2,t);
figure(fig); clf; fig = fig + 1;
plot(t,y,'--',t2,y2,'LineWidth', 2); axis([0 1 0 1.2])
grid;
legend( ['\rho = ', num2str(R)], ['\rho = ', num2str(10*R)], 'Location', 'SouthEast')

% find the feedforward gains
Nbar = inv(-c*inv(a-b*k)*b);
Nbar2 = inv(-c*inv(a-b*k2)*b);
sys1 = ss(a-b*k,b*Nbar,c,d);
sys2 = ss(a-b*k2,b*Nbar2,c,d);
t = [0:.005:1];
[y,t,x] = step(sys1,t);
[y2,t2,x2] = step(sys2,t);
figure(fig); clf; fig = fig + 1;
plot(t,y,'--',t2,y2,'LineWidth', 2); axis([0 1 0 1.2])
grid;
legend( ['\rho = ', num2str(R)], ['\rho = ', num2str(10*R)], 'Location', 'SouthEast')

% system
G = tf(8*14*20, conv([1 8], conv([1 -8],[1 -14],[1 -20])));
[a,b,c,d] = ssdata(G);
R = 1;
k = lqr(a,b,c'*c,R);

if 1

figure(fig); clf;
f = logspace(-1,2,400);
gcl1 = freqresp(sys1,f);
gcl2 = freqresp(sys2,f);
loglog(f, abs(squeeze(gcl1)), f, abs(squeeze(gcl2)),'LineWidth', 2); grid
axis([1 100 0.01 2])
xlabel('Freq [Hz]'); ylabel('G{cl}')
title('Closed-loop Freq Response')
legend( ['\rho = ', num2str(R)], ['\rho = ', num2str(10*R)], 'Location', 'SouthWest')

rm slr13.pdf

export_fig slr13 -pdf
end

if 1
f = logspace(-1,2,400);
gcl1 = freqresp(sys1,f);
gcl2 = freqresp(sys2,f);
loglog(f, abs(squeeze(gcl1)), f, abs(squeeze(gcl2)),'LineWidth', 2); grid
axis([1 100 0.01 2])
xlabel('Freq [rad/sec]'); ylabel('G{cl}')
title('Closed-loop Freq Response')
legend( ['\rho = ', num2str(R)], ['\rho = ', num2str(10*R)], 'Location', 'SouthWest')

rm slr13.pdf

export_fig slr13 -pdf
end
% find the feedforward gains
Nbar=inv(-c*inv(a-b*k)*b);
Nbar2=inv(-c*inv(a-b*k2)*b);
sys1=ss(s-a+b*k,b*Nbar,c,d);
sys2=ss(s-a+b*k2,b*Nbar2,c,d);
t=[0:.005:1];
[y,t,x]=step(sys1,t);
[y2,t2,x2]=step(sys2,t);
figure(fig);clf;fig=fig+1;
plot(t,y,'--',t2,y2,'LineWidth',2);axis([0 1 0 1.2])
grid;
legend(['$\rho =$',num2str(R)],['$\rho =$',num2str(10*R)],'Location','SouthEast')
xlabel('time (sec)');ylabel('Y output');
title('Step Response')
hold on
plot(t2([1 end]),[.1 .1]*y2(end),'r--');
plot(t2([1 end]),[.1 .1]*9*y2(end),'r--');
hold off

text(.4,.6,['$k= $ ',num2str(round(k*1000)/1000)])
text(.4,.8,['Nbar= ',num2str(Nbar)])

% find the feedforward gains
Nbar=inv(-c*inv(a-b*k)*b);
Nbar2=inv(-c*inv(a-b*k2)*b);
sys1=ss(s-a+b*k,b*Nbar,c,d);
sys2=ss(s-a+b*k2,b*Nbar2,c,d);
t=[0:.01:30];
[y,t,x]=step(sys1,t);
[y2,t2,x2]=step(sys2,t);
figure(fig);clf;fig=fig+1;
plot(t,y,'--',t2,y2,'LineWidth',2);axis([0 30 0 1.2])
grid;
legend(['$\rho =$',num2str(R)],['$\rho =$',num2str(10*R)],'Location','SouthEast')
xlabel('time (sec)');ylabel('Y output');
title('Step Response')
hold on
plot(t2([1 end]),[.1 .1]*y2(end),'r--');
plot(t2([1 end]),[.1 .1]*9*y2(end),'r--');
hold off

% find the feedforward gains
Nbar=inv(-c*inv(a-b*k)*b);
Nbar2=inv(-c*inv(a-b*k2)*b);
sys1=ss(s-a+b*k,b*Nbar,c,d);
sys2=ss(s-a+b*k2,b*Nbar2,c,d);
t=[0:.01:30];
[y,t,x]=step(sys1,t);
[y2,t2,x2]=step(sys2,t);
figure(fig);clf;fig=fig+1;
plot(t,y,'--',t2,y2,'LineWidth',2);axis([0 30 0 1.2])
grid;
legend(['$\rho =$',num2str(R)],['$\rho =$',num2str(10*R)],'Location','SouthEast')
xlabel('time (sec)');ylabel('Y output');
title('Step Response')
hold on
plot(t2([1 end]),[.1 .1]*y2(end),'r--');
plot(t2([1 end]),[.1 .1]*9*y2(end),'r--');
hold off

% find the feedforward gains
Nbar=inv(-c*inv(a-b*k)*b);
Nbar2=inv(-c*inv(a-b*k2)*b);
sys1=ss(s-a+b*k,b*Nbar,c,d);
sys2=ss(s-a+b*k2,b*Nbar2,c,d);
t=[0:.01:30];
[y,t,x]=step(sys1,t);
[y2,t2,x2]=step(sys2,t);
figure(fig);clf;fig=fig+1;
plot(t,y,'--',t2,y2,'LineWidth',2);axis([0 30 0 1.2])
grid;
legend(['$\rho =$',num2str(R)],['$\rho =$',num2str(10*R)],'Location','SouthEast')
xlabel('time (sec)');ylabel('Y output');
title('Step Response')
hold on
plot(t2([1 end]),[.1 .1]*y2(end),'r--');
plot(t2([1 end]),[.1 .1]*9*y2(end),'r--');
hold off
if 1

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

f=logspace(-3,3,400);
gcl1=freqresp(sys1,f);
gcl2=freqresp(sys2,f);

loglog(f,abs(squeeze(gcl1)),f,abs(squeeze(gcl2)),'LineWidth',2);%grid
axis([.1 100 .01 2])
xlabel('Freq (rad/sec)');ylabel('G_{cl}');
title('Closed-loop Freq Response')
legend(['\rho = ',num2str(R)],['\rho = ',num2str(10*R)],'Location','SouthWest')

%!rm srl33.pdf;
export fig srl33 −pdf

end

end % if 0

%%%%%%%%%%%%

if 1

clear all

fig=10;

G=tf([1 -1],conv([1 1],([1 -3])]);
[a,b,c,d]=ssdata(G);
R=.1
k=lqr(a,b,c'*c,R);
k2=lqr(a,b,c'*c,10*R);

% find the feedforward gains
Nbar=inv([-c*inv(a−b*k)*b]);
Nbar2=inv([-c*inv(a−b*k2)*b]);
sys1=ss(a−b*k,b*Nbar,c,d);
sys2=ss(a−b*k2,b*Nbar2,c,d);

t=[0:.01:10];
[y,t,x]=step(sys1,t);
[y2,t2,x2]=step(sys2,t);

figure(fig);clf;fig=fig+1;
plot(t,y,'−−',t2,y2,'LineWidth',2);axis([0 10 −1 1.2])
grid;
legend(['\rho = ',num2str(R)],['\rho = ',num2str(10*R)],'Location','SouthEast')
title('Unstable, NMP system Step Response')
hold on
plot(t2([1 end]),[.1 .1]*y2(end),'r−−');
plot(t2([1 end]),[.1 .1]*9*y2(end),r−−);
hold off

text(5,.6,['k= ',num2str(round(k*1000)/1000),' ']')
text(5,.8,['Nbar= ',num2str(Nbar)])

%!rm srl42.pdf;
export_fig srl42 −pdf

if 1

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

f=logspace(−2,2,400);
gcl1=freqresp(sys1,f);
gcl2=freqresp(sys2,f);

loglog(f,abs(squeeze(gcl1)),f,abs(squeeze(gcl2)),'LineWidth',2);%grid
axis([.1 100 .01 2])
xlabel('Freq (rad/sec)');ylabel('G_{cl}');
title('Closed-loop Freq Response')
legend(['\rho = ',num2str(R)],['\rho = ',num2str(10*R)],'Location','SouthWest')

%!rm srl43.pdf;
export fig srl43 −pdf

end

end % if 0

%%%%%%%%%%%%
clear all
fig=13;

G=tf(conv([1 -2],[1 -4]),conv(conv([1 -1],[1 -3]),[1 2*2+2 2^2 0 0]));
[a,b,c,d]=ssdata(G);
R=.1
k1=lqr(a,b,c'*c,R);
k2=lqr(a,b,c'*c,10*R);

% find the feedforward gains
Nbar=inv(-c*inv(a-b*k1)*b);
Nbar2=inv(-c*inv(a-b*k2)*b);
sys1=ss(a-b*k,b*Nbar,c,d);
sys2=ss(a-b*k2,b*Nbar2,c,d);

t=[0:.01:10];
[y,t,x]=step(sys1,t);
[y2,t2,x2]=step(sys2,t);
figure(fig);clf;fig=fig+1;
plot(t,y,'−−',t2,y2,'LineWidth',2);axis([0 10 −1 1.2])
grid;
legend(['\rho = ',num2str(R)],[\rho = ',num2str(10*R)], 'Location', 'SouthEast')
title('Unstable, NMP system Step Response')
hold on
plot(t2([1 end]),[.1 .1]*y2(end),'r−−');
plot(t2([1 end]),[.1 .1]*9*y2(end),'r−−');
hold off

if 1
    figure(fig);clf;fig=fig+1;
    f=logspace(-2,2,400);
    gcl1=freqresp(sys1,f);
    gcl2=freqresp(sys2,f);
    loglog(f,abs(squeeze(gcl1)),f,abs(squeeze(gcl2)),'LineWidth',2);grid
    xlabel('Freq (rad/sec)')
    ylabel('|G_{cl}|')
    title('Closed-loop Freq Response')
    legend(['\rho = ',num2str(R)],[\rho = ',num2str(10*R)], 'Location', 'SouthWest')
end
Example: B747

- Consider the lateral dynamics of a B747 at cruise (40,000ft, M=0.8)
- Variables of interest now include lateral velocity (side slip, $\beta$), yaw $\psi$ and yaw rate $r$, roll $\phi$ and roll rate $p$.
- Actuators are aileron $\delta_a$ and rudder $\delta_r$ (Figure 10.30 from FPE)

- Form nonlinear dynamics as before and linearize about forward cruise flight condition to get equations of motion for **lateral dynamics**

$$
\dot{x} = Ax + Bu, \quad x = \begin{bmatrix} \beta \\ r \\ p \\ \phi \end{bmatrix}, \quad u = \begin{bmatrix} \delta_r \\ \delta_a \end{bmatrix}, \quad \psi = r
$$

$$
A = \begin{bmatrix}
-0.0558 & -0.9968 & 0.0802 & 0.0415 \\
0.598 & -0.115 & -0.0318 & 0 \\
-3.05 & 0.388 & -0.4650 & 0 \\
0 & 0.0805 & 1 & 0
\end{bmatrix}
$$

$$
B = \begin{bmatrix}
0.00729 & 0 \\
-0.475 & 0.00775 \\
0.153 & 0.143 \\
0 & 0
\end{bmatrix}
$$

and we can sense the yaw rate $r$ and bank angle $\phi$.

- Goal: Investigate OL behavior and correct parts that we don’t like.
Open-Loop Dynamics

- Code gives the numerical values for all of the stability derivatives.
  - Solve for the eigenvalues of the matrix $A$ to find the lateral modes.

$$-0.0329 \pm 0.9467i, -0.5627, -0.0073$$

- Stable, but there is one very slow pole.

- There are 3 modes, but they are a **lot more complicated** than the longitudinal case.

  - Slow mode $-0.0073 \Rightarrow$ Spiral Mode
  - Fast real $-0.5627 \Rightarrow$ Roll Damping
  - Oscillatory $-0.0329 \pm 0.9467i \Rightarrow$ Dutch Roll

- Can look at normalized eigenvectors:

<table>
<thead>
<tr>
<th></th>
<th>Spiral</th>
<th>Roll</th>
<th>Dutch Roll</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta = v/U_0$</td>
<td>0.0067</td>
<td>-0.0197</td>
<td>0.3085</td>
</tr>
<tr>
<td>$\dot{p} = p/(2U_0/b)$</td>
<td>-0.0009</td>
<td>-0.0712</td>
<td>0.1131</td>
</tr>
<tr>
<td>$\dot{r} = r/(2U_0/b)$</td>
<td>0.0052</td>
<td>0.0040</td>
<td>0.0348</td>
</tr>
<tr>
<td>$\phi$</td>
<td>1.0000</td>
<td>1.0000</td>
<td>0.9438</td>
</tr>
</tbody>
</table>

- **Not as enlightening as the longitudinal case** – the lateral dynamics tightly couple all of the states in a complex motion

- Note that the Dutch roll is too lightly damped, and we would typically like to increase that damping using closed-loop control
Washout Filter

- A hidden complexity in the design of an autopilot for the lateral dynamics is that there are various flight modes that require a steady yaw rate \( r_{ss} \neq 0 \). For example, steady turning flight.
  - So the control that we implement must not be concerned if the steady state value \( r \) is non-zero

- Typically achieved using a washout filter \( H_w(s) \) – a high pass version of the \( r \) signal.
  - High pass cuts out the low frequency content in the signal

- For now we will penalize that filtered state in the LQR cost function
  - When we get to output feedback, it is only the filtered \( r \) that is available to the controller

Fig. 6: Washout filter with \( \tau = 4.2 \)
• Add the rudder dynamics $H_r(s) = 10/(s + 10)$, i.e. effectively a lag

• New control picture (shows a nominal yaw damper – we will replace that with a state feedback controller)

\[
H_w(s) \quad \text{Yaw Gyro}
\]

\[
H_r(s) \quad \text{Aircraft}
\]

\[
\frac{1}{s} \quad \psi
\]

\[
\frac{1}{s} \quad \phi
\]

\[
r_c \quad K \quad e \quad e_{\delta_r} \quad \delta_r \quad r
\]

\[
\delta_a
\]

Fig. 7: Typical implementation of the LQ servo

• To proceed, must augment the filter and actuator dynamics to the state model

• Algebra of this will be discussed later, but Matlab has easy ways of doing this augmentation because we can multiply systems together (note code using a I/O order different than picture above)

```matlab
1  sys = ss(A,B,C,D);
2  set(sys, 'inputname', {'rudder' 'aileron'},... 
3      'outputname', {'yaw rate' 'bank angle'});
4  set(sys, 'statename', {'beta' 'yaw rate' 'roll rate' 'phi'});
5  % actuator dynamics are a lag at 10
6  actn=10;actd=[1 10]; % H(s) in notes
7  H=tf([actn 0;0 1],[actd 1;1 1]);
8  %
9  % Washout filter
10  tau=3;
11  washn=[1 0];washd=[1 1/tau]; % washout filter on yaw rate
12  WashFilt=tf([washn 0;0 1],[washd 1;1 1]);
13  %
14  Op=WashFilt*sys*H;
```

• Inputs are now $e_{\delta_r}$ and $\delta_a$, and outputs are $e$ and $\phi$ – from this MIMO system, pull out the SISO system from $e_{\delta_r} \rightarrow e$

• Sixth order because of the augmented states

• Easiest control is of course to try gain feedback from $e$ to $e_{\delta_r}$ – will compare that with LQR
B747 Lateral Gain Feedback

- Try gain feedback - see root locus

- Presence of zero on imaginary axis “traps” the Dutch roll pole so that it cannot be moved far into the LHP

- Net result is that we are limited to about $\approx 30\%$ damping in that pole
B747 Lateral LQR

- Using an LQR state cost based on $e^2$, can design an LQR controller for various $R = \rho$.
- But where do the poles go as a fn of $\rho$?
- Follow a pattern defined by a symmetric root locus (SRL) – form pole/zero map from input $e_\delta$, to output $e$, put these on the s-plane, introduce mirror image of these dynamics about the imaginary axis, plot root locus using standard root-locus tools.

Dutch roll mode is strongly influenced by this control approach, but effect of zero still apparent
- Suggests $\rho \approx 0.1$, which gives about 40% damping in that mode

```matlab
1 Gp=WashFilt*sys*H;
2 set(Gp, 'statename', {'xwo' 'beta' 'yaw rate' 'roll' 'phi' 'xa'});
3 set(Gp, 'inputname', {'rudder inputs' 'aileron'},
   'outputname', {'filtered yaw rate' 'bank angle'});
4 [Ap,Bp,Cp,Dp]=ssdata(Gp);
5 [Klqr,S,Elqr]=lqr(Ap,Bp(:,1),Cp(1,:)'*Cp(1,:),0.1)
```
• $K_{lqr} = \begin{bmatrix} 1.0105 & -0.1968 & -2.3864 & 0.1005 & 0.0374 & 0.2721 \end{bmatrix}$ giving

$$\lambda(A_p - B_p(:,1)K_{lqr}) = \begin{bmatrix} -9.8852 \\ -1.1477 \\ -0.2750 \pm 0.6200i \\ -0.4696 \\ -0.0050 \end{bmatrix}$$

• Compare the initial condition response to a perturbation to $\beta$

• Closed-loop response looks pretty good

• But note that the original gain feedback design wasn’t that much worse – is the additional complexity of the full state feedback justified?
Code: B747 LQR, Estimator and DOFB

% B747 lateral dynamics
% Jonathan How, Fall 2010
% working from Matlab Demo called jetdemo
clear all;

% A=[-0.0558 -0.9968 .0802 .0415; .598 -0.115 -0.0318 0; -3.05 .388 -0.4650 0; 0 0.0805 1 0];
B=[ 0.00729 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'});

% (Yo1,Yo2)=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} ,{actd 1;1 1});

tau=3;washn=[1 0];washd=[1 1/tau]; % washout filter on yaw rate
WashFilt=tf( {washn 0;0} ,{washd 1;1 1});

Gp=WashFilt*sys*H;

% gain feedback on the filter yaw rate
figure(1);clf
rlocus(A,B(:,1),-C(1,:),0);
sgrid([.1 .2 .3 .4],[.7 .8 .9 1]);grid on;axis([-1.4 .1 -1.2 1.2])
Kgain=2;

Gbeta=ss(A-B(:,1)*Kgain*C(1,:),B,C(1,:),0);% performance output variable
xp0=[0 1 0 0 0 0]';
[Ygain,Tgain]=initial(Gbeta,xp0,Tol);

% LQR loop
Acl=Ap-B(:,1)*Klqr;
Bcl=Bp(:,1);

Ccl=[Cbeta;Klqr];
Dcl=[0;0];
Glqr=ss(Acl,Bcl,Ccl,Dcl);

% estimator poles made faster by jfactor

October 17, 2010
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);

% choose output to be state beta and control u
acl=[Ap Bp(:,1)*cc;−bc*Cp(1,:) ac];
bcl=[zeros(na,1);bc];
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));
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('β');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