Lecture 18 Some State Space Examples in Relation to Flight Dynamics   (3/5/20)

***Example* 1.** Consider a general aviation aircraft constrained to **pure** yaw motion (e.g. in a wind tunnel), described by: $\Delta\ddot{\psi} + 0.76\Delta\dot{\psi} + 4.55\Delta\psi = -4.6\Delta\delta_r$. Recall that in this setting the yaw rate is: $\Delta\dot{\psi}(t) = \Delta r(t)$.

Define the state $\mathbf{x}(t) = [x_1(t)\ x_2(t)]^{tr} = [\Delta r(t)\ \Delta\psi(t)]^{tr}$.

(a) Develop the state space form of (1): $\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$.

Solution: $\Delta\dot{r} + 0.76\Delta r + 4.55\Delta\psi = -4.6\Delta\delta_r$ gives $\dot{\mathbf{x}} = \begin{bmatrix}\Delta\dot{r}\\ \Delta\dot{\psi}\end{bmatrix} = \begin{bmatrix}-0.76 & -4.55\\ 1 & 0\end{bmatrix}\begin{bmatrix}\Delta r\\ \Delta\psi\end{bmatrix} + \begin{bmatrix}-4.6\\ 0\end{bmatrix}\Delta\delta_r = \mathbf{Ax} + \mathbf{Bu}$.

(b) Compute the eigenvalues for the system in (a) using (i) ‘eig(A)’. Then (ii) compute the roots of the system characteristic polynomial, to show that these roots are, indeed, the eigenvalues of **A**.

Solution: eigs(Ap) = -0.3800 +/- 2.0990i. Because the system is represented in the controller canonical form, the coefficients of $p(s)$ are a=[1 -Ap(1,:)] = s^2 + 0.76s + 4.55. Hence, roots(a) = -0.3800 +/- 2.0990i. Verified. ☺

(c) Determine the system time constant τ, damping ratio ζ, and undamped natural frequency ω_n.

Solution: $\zeta\omega_n = 0.38 \Rightarrow \tau = 1/\zeta\omega_n = 2.63\,\text{sec.}$

$\omega_d^2 = \omega_n^2(1-\zeta^2) = \omega_n^2 - (\zeta\omega_n)^2 \Rightarrow \omega_n = \sqrt{\omega_d^2 + (\zeta\omega_n)^2} = \sqrt{2.09^2 + 0.38^2} = 2.124\,rad/\text{sec}$. So $\zeta = 0.38 \ \ \&\ \ \omega_n = 0.179$

(d) Obtain the full state feedback control matrix $\mathbf{K} = [K_1 \quad K_2]$ that will result in $\tau = 1.3\,\text{sec.}$ and $\zeta = 0.75$.

Solution: For $\tau = 1.3 = 1/\zeta\omega_n \Rightarrow \zeta\omega_n = 0.769 \Rightarrow \omega_n = 0.769/\zeta = 1.025\,rad/\text{sec.}$. So, $s_{1,2} = -\zeta\omega_n \pm i\,\omega_n\sqrt{1-\zeta^2} = -0.769 \pm i0.678$.

p=[-.769+1i*.678; -.769-1i*.678]; K=acker(Ap,Bp,p) = [-0.1691   0.7606];

(e) Develop the closed loop transfer functions for this 1-input/2-output ***command*** system.

Solution: $\dot{\mathbf{x}} = \breve{\mathbf{A}}\mathbf{x} + \mathbf{Bu} \Rightarrow \mathbf{X}(s) = (s\mathbf{I} - \breve{\mathbf{A}})^{-1}\mathbf{B}\,\mathbf{U}(s)$.

$$\breve{\mathbf{A}} = \mathbf{A} - \mathbf{BK} = \begin{bmatrix}-0.76 & -4.55\\ 1 & 0\end{bmatrix} - \begin{bmatrix}0.78 & -3.50\\ 0 & 0\end{bmatrix} = \begin{bmatrix}-1.54 & -1.05\\ 1 & 0\end{bmatrix}.$$ Hence, $p(s) = s^2 + 1.54s + 1.05$.

$$(s\mathbf{I} - \breve{\mathbf{A}})^{-1}\mathbf{B} = \begin{bmatrix}s+1.54 & 1.05\\ -1 & s\end{bmatrix}^{-1}\begin{bmatrix}-4.6\\ 0\end{bmatrix} = \begin{bmatrix}s/p(s) & -1.05/p(s)\\ 1/p(s) & (s+1.54)/p(s)\end{bmatrix}\begin{bmatrix}-4.6\\ 0\end{bmatrix} = \begin{bmatrix}-4.6s/p(s)\\ -4.6/p(s)\end{bmatrix} = \mathbf{W}(s).$$

Yes- “Develop” means develop. It does not mean use Matlab commands. If you are unsure, then ASK.

Specifically, $\dfrac{\Delta r(s)}{\Delta\delta_r(s)} = W_{rr}(s) = \dfrac{-4.46s}{s^2 + 1.54s + 1.05}$ and $\dfrac{\Delta\psi(s)}{\Delta\delta_r(s)} = W_{\psi r}(s) = \dfrac{-4.46}{s^2 + 1.54s + 1.05}$.

Remark 1. The similarity of the TFs is to be expected, since $\Delta\dot{\psi}(t) = \Delta r(t) \Rightarrow s\,\Delta\psi(s) = \Delta r(s)$.

(f) The PD controller $G_c(s) = -0.1691s + 0,7606$ was placed in the feedback loop. What would the CL transfer functions be, had it been placed in the forward loop? [IN-CLASS ANSWERS]

Solution:

The plant TF is: $G_p(s) = \dfrac{-4.6}{s^2 + 0.76s + 4.55}$. The controller TF is: $G_c(s) = -0.1691s + 0.7606$.

Hence, the command CL TF is: $W_c(s) = \dfrac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)} = \dfrac{-4.6G_c(s)}{G_c(s) + s^2 + 0.76s + 4.55}$.

(g) Compute the command CL TF and obtain the response to a unit step rudder angle. Then discuss the performance of the CL system.

Solution: $W_c(s) = \dfrac{0.78s - 3.5}{s^2 + 1.54s + 1.05}$.

The system has minimal overshoot and settling time ~7 seconds. However, it has static gain ~ -3.3. The desired gain is -1. We could modify it to 1.0 by multiplying the input by 1/3.3. This is sometimes called a fudge factor. □

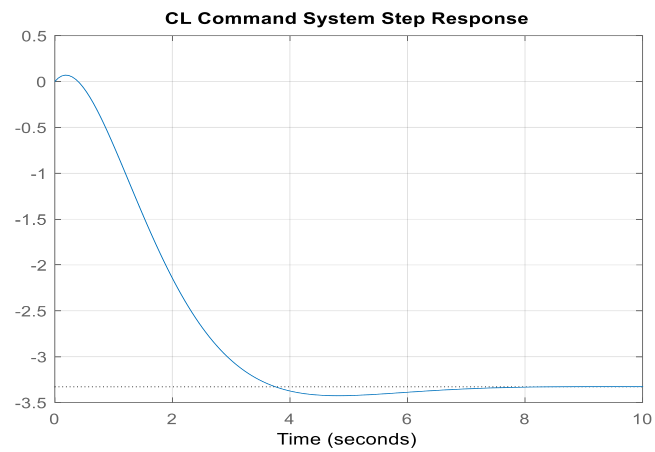


Figure E1. CL command system unit step response.

***Example* 2.** Aircraft small perturbation lateral dynamics are described by [see Nelson (5.35) p.195]:

$$\begin{bmatrix} \Delta\dot{\beta} \\ \Delta\dot{p} \\ \Delta\dot{r} \\ \Delta\dot{\phi} \end{bmatrix} = \begin{bmatrix} Y_\beta/u_0 & Y_p/u_0 & Y_r/u_0 - 1 & g\cos\theta_0/u_0 \\ L_\beta & L_p & L_r & 0 \\ N_\beta & N_p & N_r & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\beta \\ \Delta p \\ \Delta r \\ \Delta\phi \end{bmatrix} + \begin{bmatrix} 0 & Y_{\delta_r}/u_0 \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\delta_a \\ \Delta\delta_r \end{bmatrix} \tag{1a}$$

Consider a general aviation plane whose lateral dynamics described by (1a) is:

$$\mathbf{A} = \begin{bmatrix} -.25 & 0 & -1 & .18 \\ -16.02 & -8.40 & 2.19 & 0 \\ 4.49 & -.35 & -.76 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} .07 & 0 \\ 23.16 & -29.01 \\ 4.55 & .22 \\ 0 & 0 \end{bmatrix} \tag{1b}$$

The general state space equations are $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$. In this problem, choose $\mathbf{y}(t) = \mathbf{x}(t)$

(a) Use *Matlab* to find the eigenvalues of three lateral modes.

<u>*Solution*</u>: 'eig(A)' gives: s_1= -8.43 (roll) ; s_2= -0.48 +/- 2.33i (Dutch roll) ; s_4=-0.0088 (spiral)

(b) Use the *Matlab* command 'ss2tf' to arrive at the coefficients of the 4 transfer functions for input $\Delta\delta_r$.

<u>*Solution*</u>: `C=eye(4);D=zeros(4,2); [n1,d1]=ss2tf(A,Bp,C,D,2);`

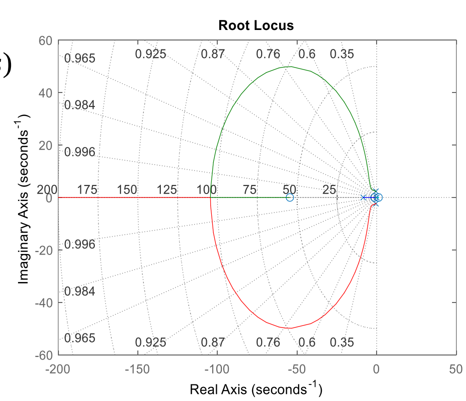
```
n1 =

        0         0   -0.2200  -17.2233   -3.8818
        0  -29.0100  -28.8183 -132.1219         0
        0    0.2200   12.0565    3.0004  -22.8115
        0         0  -29.0100  -28.8183 -132.1219

d1 =

   1.0000    9.4100   13.9305   47.9942    0.4216
```

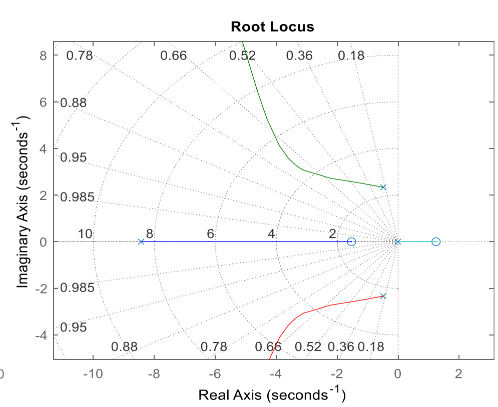
(d) Give the transfer function $\dfrac{\Delta r(s)}{\Delta\delta_r(s)} = G_{rr}(s)$. Then evaluate its poles and zeros.

<u>*Solution*</u>: Grr=tf(n1(3,:),d1) = (0.22 s^3 + 12.06 s^2 + 3 s - 22.81) / (s^4 + 9.41 s^3 + 13.93 s^2 + 47.99 s + 0.4216)

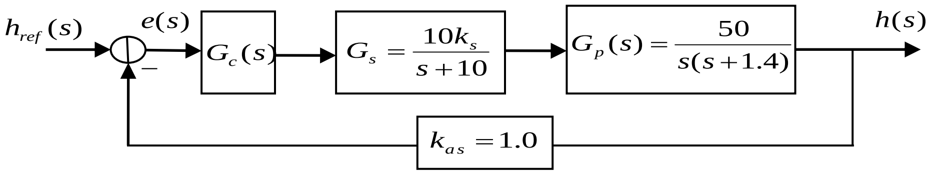
GrrPOLES=roots(d1) = -8.4322 ; -0.4845 +/- 2.3329i ; -0.0088
GrrZEROS=roots(n1(3,:)) = -54.5172 ; -1.5290 ; 1.2439

(e) Give a plot of the roots of $1 + KG_{rr}(s)$ as a function of K. Then draw a conclusion regarding the associated CL stability as a function of K.
<u>*Solution*</u>: rlocus(Grr). The zoomed plot shows a locus beginning at zero and continuing into the RHP. Hence, no P-control will stabilize the system. □





***Example* 3.**



Attitude indicator with integrated localizer and glideslope and split-cue flight director command bar indicators, indicating brown earth below and sky above, wings level with horizon, in a slight nose-down attitude.

A STOL transport has been modified to include direct-lift control surfaces, integrated with altitude indicator feedback, a servo drive, and a controller, as shown in the block diagram above. Assume that the forward loop controller $G_c(s) = 1.0$.

(a) Recover the differential equation that relates the input $e(t)$ to the output $h(t)$.

<u>Solution</u>: $\quad \dfrac{h(s)}{e(s)} = \dfrac{500k_s}{s^3 + 11.4s^2 + 14s}$. Hence, $\dddot{h} + 11.4\ddot{h} + 14\dot{h} = 500k_s e(t)$

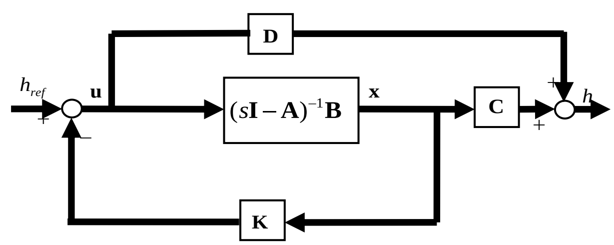
(b) Develop the *controller* canonical form for $(\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D})$ that relates the input $e(t)$ to the output $h(t)$.

<u>Solution</u>: The controller form uses: $\mathbf{x}(t) = (1/500k_s)\begin{bmatrix}\ddot{h}(t) & \dot{h}(t) & h(t)\end{bmatrix}^{tr}$ and $\mathbf{C} = \begin{bmatrix}0 & 0 & 500k_s\end{bmatrix}$

From (a) we have $\dot{x}_1 = -11.4x_1 + 14x_2 = e(t)$. We also have $\dot{x}_2 = x_1$ and $\dot{x}_3 = x_2$. Hence:

$$\dot{\mathbf{x}} = \begin{bmatrix} -11.4 & -14 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} e(t) = \mathbf{Ax} + \mathbf{B}e \text{ and } h(t) = \begin{bmatrix} 0 & 0 & 500k_s \end{bmatrix} x + 0e(t) = \mathbf{Cx} + \mathbf{D}e(t).$$

(c) Suppose that we incorporate *full state <u>feedback</u>* as shown at the right. We then have: $\mathbf{X}(s) = (s\mathbf{I} - \mathbf{A}_{\mathbf{BK}})^{-1}\mathbf{Bu}$ with $\mathbf{A}_{\mathbf{BK}} \overset{\Delta}{=} \mathbf{A} - \mathbf{BK}$.
It is desired to find the values of **K** such that the closed loop poles will be $s_1 = -2$; $s_{2,3} = -2 \pm 1.5i$. Use the *Matlab* command 'place' to find **K**.



<u>Solution</u>: K=place(Ap,Bp,C,D,p) = [-5.4 0.25 12.5]

(d) Plot the closed loop step response for $k_s = 1.0$. Then find the value of k_s that will result in unity static gain, and plot the closed loop step response to verify your answer.

<u>Solution</u>: [n1,d1]=ss2tf(A,Bp,C,D); W1=tf(n1,d1) = 500 / (s^3 + 6 s^2 + 14.25 s + 12.5), so W(0)= 40. Hence, unity static gain requires $k_s = .025$.

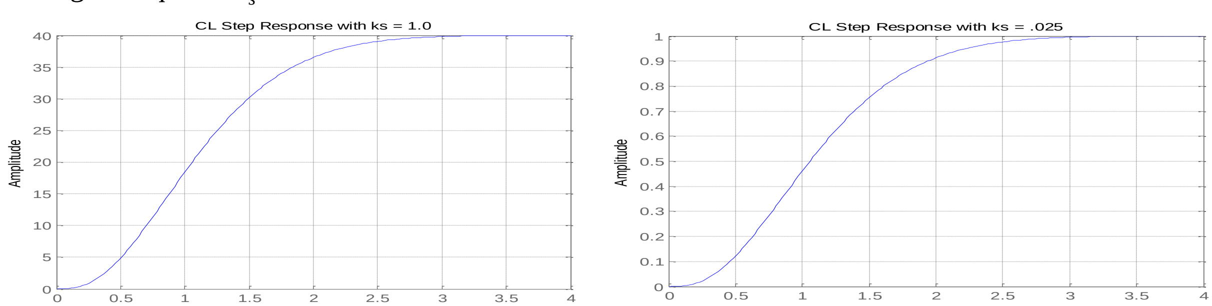


Figure 3(d) Closed loop step response for $k_s = 1.0$ (left) and for $k_s = .025$ (right).

(e) The controller output is $[K_1 \quad K_2 \quad K_3]\mathbf{x}(t) = (1/500k_s)[K_1\ddot{h} + K_2\dot{h} + K_3 h]$. Hence, the controller transfer function is

$$\frac{U(s)}{H(s)} = G_c(s) = (1/500k_s)[K_1 s^2 + K_2 s + K_3].$$

Place the controller into the forward loop to arrive at the command closed loop TF. Then obtain a plot of the system step response.

<u>*Solution*</u>:

Wc = -2700 s^2 + 125 s + 6250

----

500 s^3 + 3000 s^2 + 7125 s + 6250

Equivalently: $W_c(s) = \dfrac{-5.4s^2 + 0.25s + 12.5}{s^3 + 6s^2 + 14.25s + 12.5}$

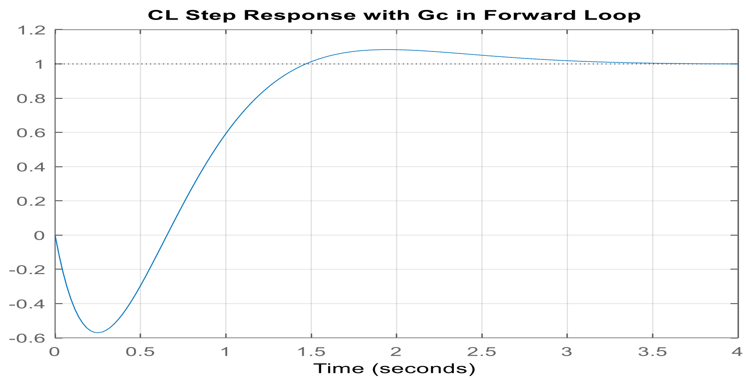


Figure 3(e) CL command system step response.

(f) Discuss the nature of the step response in (e), as well as the nature of the controller.

<u>*Discussion*</u>:

The step response is quite sensitive to the step in the initial response region. This is because the controller is not just a PD controller. It is a proportional-plus-derivative-plus second derivative controller. The higher the derivative, the more sensitive the response will be in the initial stages. If such a controller has a step input, it will respond not only to the step, but also to its derivative that is an impulse, but also to its second derivative that is a pair of impulses!

Remark. This example demonstrates the issue associated with state feedback that can be 'complicated'. Using the Matlab commands `[n1,d1]=ss2tf(A,Bp,C,D); W1=tf(n1,d1);` places the controller in the <u>state</u> <u>feedback</u> loop. Hence, the CL system becomes a regulator, not a command system. It takes some thought as to how to incorporate the controller into the forward loop, so as to have a command system. In this example it was necessary to realize that when using the controller canonical form, the open loop numerator coefficient $500k_s$ is placed in the **C** matrix. Hence, to obtain the controller transfer required that it be properly scaled. □

Matlab Code

```
% PROGRAM NAME: lec18.m  (3/4/20)
%EXAMPLE 1:
s=tf('s');
Gp=tf(-4.6,[1 .76 4.55]);
Ap=[-.76,-4.55;1,0];
Bp=[-4.6;0];
C=eye(2);
D=zeros(2,1);
%--------------------
[np,dp]=ss2tf(Ap,Bp,C,D);
Gp1=tf(np(1,:),dp);
Gp2=tf(np(2,:),dp);
%--------------------
%(b):
eigs(Ap)
a=[1 -Ap(1,:)];
roots(a)
%(d)
%-------------------
p=[-.769+1i*.678; -.769-1i*.678];
K=acker(Ap,Bp,p)
A=Ap-Bp*K;
ps=s^2-A(1,1)*s-A(1,2)
%(f):
Gc1=tf(K,1);
Wc=feedback(Gc1*Gp,1);
figure(1)
step(Wc)
title('CL Command System Step Response')
grid
%==========================
%EXAMPLE 2
A=[-.25,0,-1,.18;-16.02,-8.4,2.19,0;4.49,-.35,-.76,0;0,1,0,0];
Bp=[.07,0;23.16,-29.01;4.55,.22;0,0];
C=eye(4);
D=zeros(4,2);
eigs(A)
[n1,d1]=ss2tf(A,Bp,C,D,2);
Gp31=tf(n1(3,:),d1);
figure(20)
rlocus(Gp31)
figure(5)
step(Gp31)
title('Step Response for T.F. PHI(s)/DELr(s)')
grid
%==========================================
%EXAMPLE 3:
Ap=[-11.4,-14,0;1,0,0;0,1,0];
Bp=[1;0;0];
s1=-2; s2=-2+1i*1.5; s3=conj(s2);
svec=[s1;s2;s3];
K=place(Ap,Bp,svec);
A=Ap-Bp*K;
ks=1.0;
C=[0,0,500*ks];
D=0;
[n1,d1]=ss2tf(A,Bp,C,D); %Places Gc in feedback loop
W1=tf(n1,d1);
figure(6)
step(W1)
title('C.L. Step Response for ks=1')
grid
ks=1/40;
```

```
W=ks*W1;
figure(7)
step(W)
title('C.L. Step Response for ks=1/40')
grid
%Place Gc in forward loop:
s=tf('s');
Gc=K*[s^2;s;1]/500;
Gs=10/(s+10);
Gp=50/(s^2+1.4*s);
G=Gc*Gs*Gp;
Wc=feedback(G,1)
figure(8)
step(Wc)
title('CL Step Response with Gc in Forward Loop')
grid
```