Homework 4 Spring 2020 AerE331 Due 3/13(F) SOLUTION

PROBLEM 1(45pts) In this problem we address feedback control of the AOA/elevator transfer function $\frac{\alpha(s)}{\delta_e(s)} = G_p(s) = \frac{0.023s^3 + 1.168s^2 + 0.008s + 0.005}{s^4 + 0.75s^3 + 0.94s^2 + 0.01s + 0.004}$. Clearly, this is a challenging problem, since it has three zeros.

The design specifications include: (S1) unit ramp $e_{ss} \cong 0.2^o$; (S2) $PM \cong 70^o$ at $\omega_{gc} \cong 1r/s$

(a)(5pts) It should be clear that the controller must include the term K/s. Show that in order to satisfy (S1), $K = 4.0$.

Solution:

The Type-1 error constant is $K_e = \lim_{s\to 0} s(K/s)G_p(s) = 1.25K = 1/e_{ss} = 5$. Hence, $K = 4.0$.

(b)(9pts) Beginning with data cursor information associated with $(4/s)G_p(s)$, design a unity static gain compensator that will result in open loop phase $\theta(\omega=1) \cong -110^o$. Then verify this via the OL Bode plot. Also, give the value of $M(\omega=1)_{dB}$.

Solution: [See code @ 1(b).]

From the left plot in Fig.1(b) we see that $\theta(\omega=1) \cong -185^o$. To raise it to $\theta(\omega=1) \cong -110^o$ will require a lead compensator with $\phi_{max} = 75^o$ at $\omega_{max} = 1 = \sqrt{\omega_1\omega_2}$. Now,

$\alpha = \frac{\omega_2}{\omega_1} = \frac{1+\sin(75^o)}{1-\sin(75^o)} = 57.7$. Hence,

$\omega_1 = \omega_{max}\sqrt{\alpha} = 0.1317$, and

$\omega_2 = \alpha\omega_1 = 7.596$. $M(1) = $ **33.4**dB

Hence, $G_{lead}(s) = \frac{7.596}{0.1317}\left(\frac{s+0.1317}{s+7.596}\right)$.

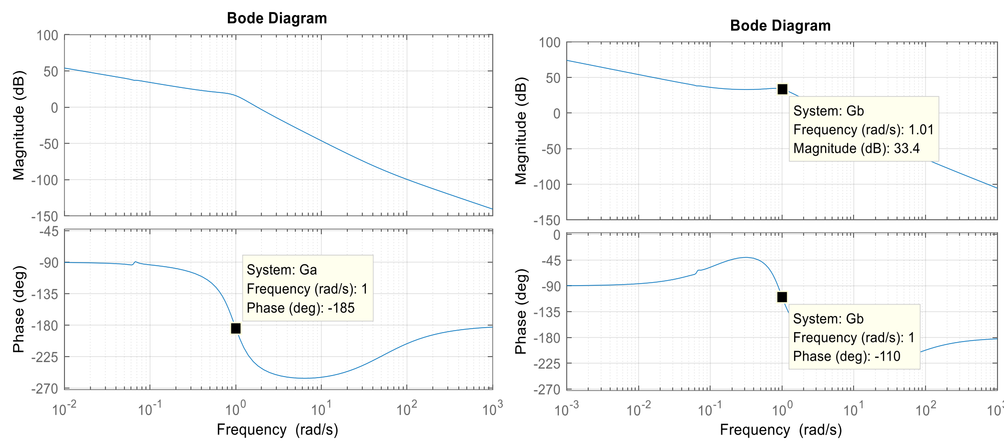

Figure 1(b). OL Bode plot prior to (LEFT) and including (RIGHT) $G_{lead}(s)$.

(c)(9pts) Design a second unity static gain compensator to force $\omega_{gc} \cong 1r/s$. Then give the % error between your PM and (S2).

Solution: [See code @ 1(c).]

A lag compensator with $M(\omega=1)_{dB} = -33.4dB$ and with $\theta(\omega=1) \cong 0^o$ is required. So $\alpha = \frac{\omega_2}{\omega_1} = 10^{-33.4/20} = .0214$. I will set $20\omega_1 = 1$. These give $\omega_1 = 0.05$ and $\omega_2 = 0.0011$. Hence, $G_{lag}(s) = \frac{.0011}{0.05}\left(\frac{s+.05}{s+.0011}\right)$.

I have a $PM = 66^o$ and (S2) is $PM \cong 70^o$. My error is **-5.7%**

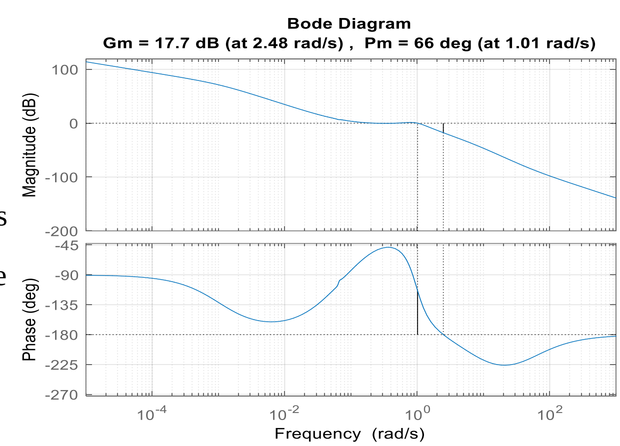

Figure 1(c). Final OL Bode plot.

(d)(5pts) Use your lag compensator in (c) to explain why your PM does not exactly satisfy (S2).
Solution:
$G_{lag}(i1)=\frac{.0011}{0.05}\left(\frac{i1+.05}{i1+.0011}\right)$ has $\phi(1)$ = atand(1/.05)-atand(1/.0011) = -2.8°. Hence, the 70° design value is reduced to 67.2°. The remaining 1.2° is likely due to numerical round-off in the computations.

(e)(5pts) Obtain a plot of the error response to a unit ramp to verify (S1).
Solution: [See code @ 1(e).]

The error TF is $E(s)=1-W(s)$.

From the plot we have $e_{ss}\cong 0.199^{o}$, which verifies (S1)

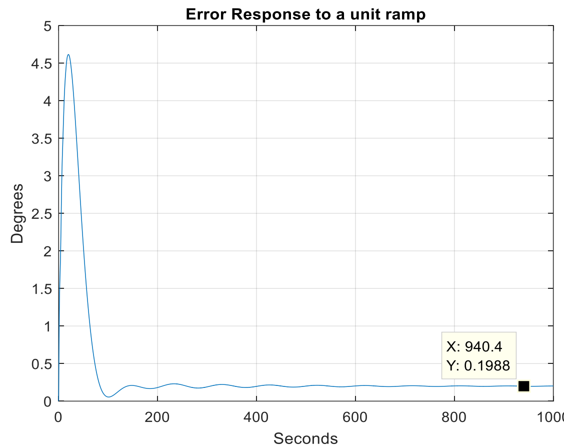

Figure 1(e). Error response to a unit ramp.

(f)(7pts) Overlay the plant and final CL command system step responses. Then discuss the relative advantage(s) of each.
Solution: [See code @ 1(f).]

The advantage of $G_p(s)$ is that the response time is ~15 seconds, compared to $W(s)$ that is ~80seconds. The advantages of $W(s)$ include:
(i) 50% less overshoot.
(ii) unity static gain.
(iii) automated.
(iv) (S1).

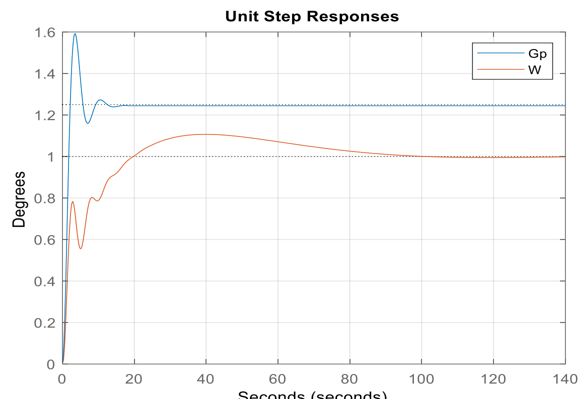

Figure 1(f). CL response to command step.

(g)(5pts) Give the controller (i) TF and (ii) Bode plot. Then (iii) compute the controller power over [10^{-5}, 10^{3}]. Give your Matlab code for the power (in dB) HERE.
Solution: [See code @ 1(g).]

$$G_c(s)=\frac{4.934s^2+0.8963s+0.03248}{s^3+7.597s^2+0.00812s}$$

```
[n,d]=tfdata(Gc,'v');
f=@(w) abs((-n(2)*w.^2+1i*n(3)*w+n(4))./(-1i*w.^3-d(2)*w.^2+1i*d(3)*w)).^2;
PWR=integral(f,10^-5,10^3);
PWRdB=10*log10(PWR)  = 61.98 dB
```

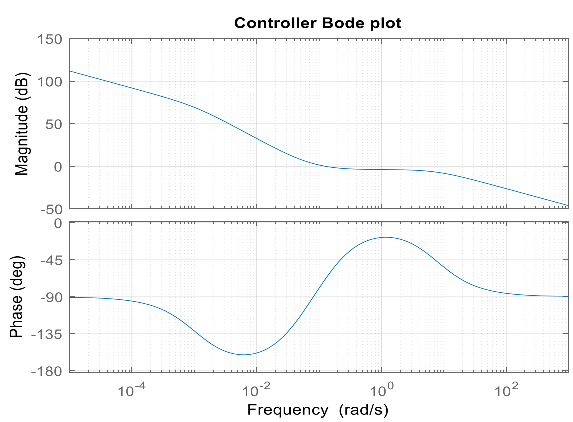

Figure 1(g). Controller Bode plot.

Problem 2(55pts) Consider the lightly damped plant $G_p(s) = \frac{2}{s^2+s+9} \overset{\Delta}{=} \frac{Y(s)}{U(s)}$. The O.D.E. is: $\ddot{y} + \dot{y} + 9y = 2u$. This is a single input-single output (SISO) system. Define the 2-D state $\mathbf{x} = [x_1 \quad x_2]^{tr} = [\dot{y} \quad y]^{tr}$.

(a)(5pts) Beginning with the ODE, show that the defined state gives $\dot{\mathbf{x}} = \begin{bmatrix} -1 & -9 \\ 1 & 0 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 2 \\ 0 \end{bmatrix}u = \mathbf{A}\mathbf{x} + \mathbf{B}u$ and that $y = [0 \quad 1]\mathbf{x} + 0u = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}$.

<u>*Solution*</u>: Since $x_1 = \dot{y}$, we have $\dot{x}_1 = \ddot{y}$, so that $\dot{x}_1 + x_1 + 9x_2 = 2u$. This gives $\dot{x}_1 = -x_1 - 9x_2 + 2u$. (1). Since $x_2 = y$, we have $\dot{x}_2 = \dot{y} = x_1$. (2) Putting (1) and (2) into the matrix form gives the desired state equation. The system output is *y*, which can be written as: $y = [0 \quad 1][\dot{y} \quad y] = [0 \quad 1]\mathbf{x} + 0u$. This is the desired output equation.

(b)(5pts) Use the 'tf2ss' command to put the SISO system into a state space form. [Copy/paste your code/result HERE.]
<u>*Solution*</u>:
[A B C D]=tf2ss(2,[1 1 9]) **A** = [-1 -9 ; 1 0] **B** =[1 ; 0] **C** =[0 2] **D** = 0

(c)(5pts) You should have found that the form in (b) is not the form in (a). Show that the state variable for this representation is $\mathbf{x} = [x_1 \quad x_2]^{tr} = 0.5[\dot{y} \quad y]^{tr}$.

<u>*Solution*</u>: From $\ddot{y} + \dot{y} + 9y = 2u$ we have: $0.5\ddot{y} + 0.5\dot{y} + 9(0.5)y = u$. Let $x_1 = 0.5\dot{y}$ and $x_2 = 0.5y$. We can then proceed exactly as in (a) to obtain the desired representation in (b).

(d)(5pts) Show that the eigenvalues of **A** are the roots of the characteristic polynomial.
<u>*Solution*</u>: [Include your code/answers HERE.]
A=[-1 -9; 1 0]; B=[2;0]; C=[0 1]; D=0;
L = eigs(A) = -0.5000 +/- 2.9580i
roots([1 1 9]) = -0.5000 +/- 2.9580i

(e)(5pts) Use the 'ss2tf' command to recover the transfer function from (**A**,**B**,**C**,**D**) in (a).
<u>*Solution*</u>: [Copy/paste code/results HERE.]
[n,d]=ss2tf(A,B,C,D) n =[0 0 2] d =[1.0000 1.0000 9.0000]
G=tf(n,d) = 2 / (s^2 + s + 9)

(f)(5pts) If in (a) we define the input *u* to be $u = -\mathbf{K}\mathbf{x} = -[K_1 \quad K_2][\dot{y} \quad y]^{tr}$, then the state equation becomes $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{K}\mathbf{x} = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x} \overset{\Delta}{=} \mathbf{A}'\mathbf{x}$, where $\mathbf{A}' = \mathbf{A} - \mathbf{B}\mathbf{K}$ Taking the Laplace transform of $\dot{\mathbf{x}} = \mathbf{A}'\mathbf{x}$ gives: $(s\mathbf{I} - \mathbf{A}')\mathbf{X}(s) = 0$. Hence, we see that the closed loop poles are the eigenvalues of $\mathbf{A}'$. (i) Use the *Matlab* command 'place(A,B)' to arrive at the gain matrix, $\mathbf{K} = [K_1 \; K_2]$ that will place poles at $s_{1,2} = -5 \pm i5$. Then (ii) verify your answer using $\mathbf{A}'$.

<u>*Solution*</u>: [Copy/paste code/results HERE.]
(i): s1=-5+1i*5; s2=conj(s1); K=place(A,B,[s1 s2]) = [**4.5 20.5**]
(ii): AA=A-B*K; eigs(AA) = **-5.0000 +/- 5.0000i**

(g)(5pts) For your $\mathbf{A}'$ in (f), use the 'ss2tf' command to determine whether the controller is in the forward or the feedback loop.

Solution: [Copy/paste code/results HERE.]

[N,D]=ss2tf(AA,B,C,D); N =[0 0 2] D =[1.00 10.00 50.00]. It is in the **feedback loop**.

(h)(5pts) In (g) you should have found that the controller was placed in the feedback loop. To place it in the forward loop it is easiest to use the state space representation in (b) where **B** = [1 ; 0] and **C** = [0 2]. Let the new $\mathbf{C}' = C(2)\mathbf{K}$. Use the ss2tf command to verify that now the controller is in the forward loop.

Solution: [Copy/paste code/results HERE.]

B1=[1;0]; C1=[0 2]; CC=C1(2)*K; [NN,DD]=ss2tf(AA,B1,CC,0) NN =[0 9.0 41.0] DD =[1.0 10.0 50.0].

(i)(5pts) Overlay the CL unit step responses associated with (g) and (h). Then discuss them in relation to command versus disturbance inputs.

Solution: [See code @ 2(i).]

The CL response with the controller in the feedback loop is $W_g(s)$, and the response with it in the forward loop is $W_h(s)$. For a disturbance input we would use $W_g(s)$, and for a command input we would use $W_h(s)$.

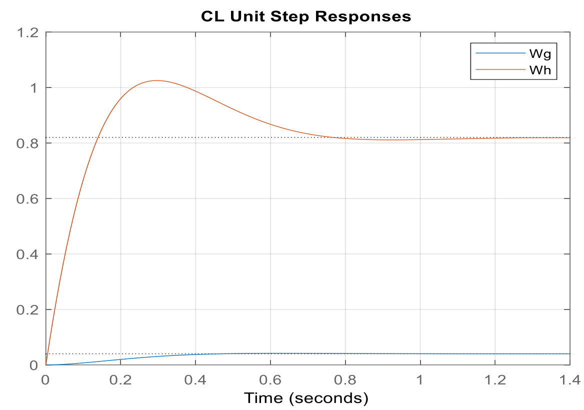

Figure 2(i) CL step responses $W_g(s)$ and $W_h(s)$.

(j)(5pts) Given that the state vector is $\mathbf{x} = [x_1 \quad x_2]^{tr} = [\dot{y} \quad y]^{tr}$, the state feedback pole placement method must result in a PD controller. Even so, the use of PID control is far more popular. Consider the state vector $\mathbf{x} = [x_1\ x_2\ x_3]^{tr} = \left[\dot{y}\ y \int y\right]^{tr}$. For this state, show that

$$\dot{\mathbf{x}} = \begin{bmatrix} -1 & -9 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}u = \mathbf{A}\mathbf{x} + \mathbf{B}u \text{ and } y = [0\,1\,0]\mathbf{x} + 0u = \mathbf{C}\mathbf{x} + \mathbf{D}u.$$

Solution: $x_3 = \int y \Rightarrow \dot{x}_3 = y = x_2$. The third row of **A** follows immediately, as does the form for **C**.

(k)(5pts) Use the 'place' command to find the values for the PID controller $\mathbf{K} = [K_1\ K_2\ K_3]$ that will place CL poles at $s_{1,2} = -5 \pm i5$ and $s_3 = -5$.

Solution: [See code @ 2(k).] $\mathbf{K}$ = [**7.0 45.5 125.0**]

Matlab Code

```
%PROGRAM NAME: hw4.m  2/25/20FROM Etkin (6.2.1) and(7.6.5)
%PROBLEM 1
Gp=tf([.023 1.168 .008 .005],[1 .75 .94 .01 .004]);
figure(10)
bode(Gp)
grid
%(a):
K=5/1.25;
Gca=tf(K,[1 0]);
Ga=Gca*Gp;
figure(11)
bode(Ga)
grid
%(b):
phimax=75;
a=(1+sind(phimax))/(1-sind(phimax));
wgc=1;
w1=wgc/sqrt(a);
w2=a*w1;
%[w1 w2]
Glead=(w2/w1)*tf([1 w1],[1 w2]);
Gcb=Gca*Glead;
Gb=Gcb*Gp;
figure(12)
bode(Gb)
grid
%(c):
aa=10^(-33.4/20);
ww1=wgc/20; %Choose 20*w1 = wgc
ww2=ww1*aa;
[ww1 ww2];
Glag=(ww2/ww1)*tf([1 ww1],[1 ww2]);
Gc=Gcb*Glag;
G=Gc*Gp;
figure(13)
margin(G)
%(e):
W=feedback(G,1);
E=1-W;
t=0:.01:1000;
u=t;
e=lsim(E,u,t);
figure(14)
plot(t,e)
title('Error Response to a unit ramp')
xlabel('Seconds')
ylabel('Degrees')
grid
%(f)
figure(15)
step(Gp)
hold on
step(W)
title('Unit Step Responses')
xlabel('Seconds')
ylabel('Degrees')
grid
legend('Gp','W')
%(g):
figure(16)
bode(Gc)
title('Controller Bode plot')
grid
[n,d]=tfdata(Gc,'v');
```

```
f=@(w) abs((-n(2)*w.^2+1i*n(3)*w+n(4))./(-1i*w.^3-d(2)*w.^2+1i*d(3)*w)).^2;
PWR=integral(f,10^-5,10^3);
PWRdB=10*log10(PWR)
%==========================================================
%PROBLEM 2
%(b):
[A B C D]=tf2ss(2,[1 1 9])
%(c):
A=[-1 -9; 1 0]; B=[2;0]; C=[0 1]; D=0;
L=eigs(A)
r=roots([1 1 9])
G=tf(n,d)
%(f):
s1=-5+1i*5; s2=conj(s1);
K=place(A,B,[s1 s2])
AA=A-B*K;
eigs(AA)
%(g):
[N,D]=ss2tf(AA,B,C,D)
%(h):
B1=[1;0];
C1=[0 2];
CC=C1(2)*K;
[NN,DD]=ss2tf(AA,B1,CC,0)
%(i):
Wg=tf(N,D);
Wh=tf(NN,DD);
figure(20)
step(Wg)
hold
step(Wh)
title('CL Unit Step Responses')
legend('Wg','Wh')
grid
%(k):
A=[-1 -9 0;1 0 0; 0 1 0]; B=[2;0;0]; C=[0 1 0];
s3=-5; svec=[s1 s2 s3];
K=place(A,B,svec)
```