16.31 Problem Set 7

Solution

Problem 1
See the attached Matlab output, which has on it the resulting compensator.

The gain margin is the inverse of the loop gain magnitude, when the phase is \( \pm 180^\circ \). The gain margin is then

\[
GM = [0.84, 1.15]
\]

\begin{align*}
\uparrow & \quad \text{upward gain margin} \\
\downarrow & \quad \text{downward gain margin}
\end{align*}

The phase margin is the difference between the phase of the loop gain and \( \pm 180^\circ \) when the magnitude of the loop gain is unity.

\[
PM = [-81^\circ, 19.3^\circ]
\]

\begin{align*}
\uparrow & \quad \text{additional phase lag allowed} \\
\downarrow & \quad \text{additional phase lead allowed}
\end{align*}

Inherently, this is a hard plant to control – it has an unstable pole near a
right half plane zero. It is inevitable that the margins will be low.
cc=f;
dc=0;

% Now put back into state space form
[numc,denc]=ss2tf(ac,bc,cc,dc,1)

numc =
    0    82.5288    247.5690

denc =
    1.0000    9.2304   -47.4282

% Is it stable?
roots(denc)

ans =
    -12.9055
    3.6750

% Now prepare the nyquist plot
num=conv(numg,numc);
den=conv(deng,denc);
w=logspace(-2,2,200);
[mag,phase]=bode(num(3:4),den,w);
[re,im]=nyquist(num,den,w);
hold off
clf
axis([-1.2 0 -0.4 0.4])
plot(re,im)
hold on
plot(re,-im)

% Find largest circle around -1 point and draw, to see the robustness.
re2=re+1;
mag2=abs(re2+j*im);

r=min(mag2)

r =
    0.1300

radius of circle around s = -1 that just touches nyquist plot.

th=(0:.002:1)*2*pi;
circle=exp(j*th);
plot(r*circle-1)
diary off
% P6-1 Solution

% Define j
j=sqrt(-1);

% Express G as a transfer function, then put into statespace form

numg=[0 1 -1];
deng=conv([1 -2],[1 3]);
[a,b,c,d]=tf2ss(numg,deng)

\[ G(s) \text{ from problem statement} \]
\[ \text{tf2ss puts } G(s) \text{ in state-space form, using phase-variable form, but with states in reverse order.} \]

a =
\[
\begin{bmatrix}
-1 & 6 \\
1 & 0
\end{bmatrix}
\]

b =
\[
\begin{bmatrix}
1 \\
0
\end{bmatrix}
\]

c =
\[
\begin{bmatrix}
1 \\
-1
\end{bmatrix}
\]

d =
\[
\begin{bmatrix}
0
\end{bmatrix}
\]

% Find the Kalman filter gain

[k,p]=lqe(a,b,c,1,1);
k =
\[
\begin{bmatrix}
8.1647 \\
4.0495
\end{bmatrix}
\]
Kalman gain

% Find the LQR gain--add a small matrix to c'*c to make positive definite.

[f,p]=lqr(a,b,c'*c+30*eps*eye(2),1);
f =
\[
\begin{bmatrix}
4.1152 \\
12.0828
\end{bmatrix}
\]
LQR gain

% The Compensator is found using the separation principle

ac=a-b*f-k*c;
bc=k;
Nyquist plot

\[ \omega = 0^+ \]
\[ \omega = 0^- \]
\[ \omega = +\infty \]
\[ \omega = -\infty \]
Problem 2
The singular values of $T$ and $T^*$ are the square roots of the eigenvalues of $T^*T$ and $TT^*$, respectively. The maximum singular value is the maximum of these, which will be nonzero, unless $T$ is the zero matrix. So it suffices to show that any nonzero eigenvalue of $T^*T$ is an eigenvalue of $TT^*$, and vice versa.

So suppose

$$T^*T \mathbf{p} = \lambda \mathbf{p}$$

Define $\frac{q}{\mathbf{b}} = T \mathbf{p}$. Then

$$TT^* \frac{q}{\mathbf{b}} = TT^*T \mathbf{p}$$

$$= T (\lambda \mathbf{p}) = \lambda T \mathbf{p}$$

$$= \lambda \frac{q}{\mathbf{b}}$$

So if $\lambda$ is an eigenvalue of $T^*T$, it is also an eigenvalue of $TT^*$. □
Problem 3

See the following two pages for the Matlab code and a test run of the code.
% Solution 7.3

function y=infnorm(A,B,C)

% Initial guess for gamma

gamma=1;
gamma_upper=Inf;
gamma_lower=0;

% Iterate until upper bound and lower bound converge

while (gamma_upper-gamma_lower)/gamma_upper > 10*eps | gamma_upper==Inf

% Form the Hamiltonian matrix

H=[ A B*B'/gamma^2 ;
   -C'*C -A' ];

% Find the eigenvalues of H, and decide if any are on jw axis
% This part is tricky, so you may use a different approach.

d=eig(H);

if min(abs(real(d)))/max(abs(d)) < 1e-14
    jw=1;
else
    jw=0;
end

% based on result, move bounds

if jw
    % gamma is higher, so move up
    if gamma_upper==Inf
        gamma_lower=gamma;
        gamma=gamma^2;
    else gamma_lower=gamma;
        gamma=(gamma_upper+gamma_lower)/2;
    end
else
    % gamma is lower, so split the difference
    gamma_upper=gamma;
    gamma=(gamma_upper+gamma_lower)/2;
end
end
y=gamma;
\begin{verbatim}
clear

A=[-0.1 1 ;
   -2 -0.3];

B=[1 2 ;
   3 4];

C=[1 1 ;
   1 -1];

infnorm(A,B,C)
echo off
diary off

ans =

25.9297 \qquad \| C \|_0
\end{verbatim}
Problem 4

The block diagram to check $\| G \|_{\infty} \leq \sigma$ is

The equivalent state-space dynamics are

$$
\begin{align*}
\dot{x}_1 &= A x_1 + B u \\
y &= C x_1 + D u
\end{align*}
\right\} \text{G}
$$

$$
\begin{align*}
\dot{x}_2 &= -A^T x_2 - C^T y \\
z &= B^T x_2 + D^T y
\end{align*}
\right\} \text{G}^T(-s)
$$

$$
\begin{align*}
u &= \frac{1}{\gamma^2} z
\end{align*}
$$

We need to eliminate $y$, $u$, and $z$ from these equations:

$$
\begin{align*}
y &= C x_1 + D u \\
&= C x_1 + \frac{D}{\gamma^2} z \\
&= C x_1 + \frac{1}{\gamma^2} D (B^T x_2 + D^T y)
\end{align*}
$$
\[
(I - \frac{1}{\sigma^2} DD^T) y = C x_1 + \frac{1}{\sigma^2} DB^T x_2
\]

\[
(\sigma^2 I - DD^T) y = \sigma^2 C x_1 + DB^T x_2
\]

\[
y = (\sigma^2 I - DD^T)^{-1} [\sigma^2 C x_1 + DB^T x_2]
\]

Similarly,

\[
u = (\sigma^2 I - DT D)^{-1} [B^T x_2 + DT C^T x_1]
\]

Therefore, the state dynamics are

\[
\dot{x}_1 = A x_1 + Bu
\]

\[
= \left\{ A + B (\sigma^2 I - DT D)^{-1} D^T C \right\} x_1
\]

\[
+ \left\{ B (\sigma^2 I - DT D)^{-1} B^T \right\} x_2
\]

\[
\dot{x}_2 = -A^T x_2 - c^T y
\]

\[
= -\left\{ A + B D^T (\sigma^2 I - DD^T)^{-1} D^T c \right\} x_2
\]

\[
- c^T (\sigma^2 I - DD^T)^{-1} \sigma^2 C x_1
\]
This can be simplified some.

\[ \hat{\chi}_1 = \{ A + B \hat{D} \hat{D}^T C \} \chi_1 + \{ B \hat{D} \hat{D}^T \} \chi_2 \]

where \( \hat{D} = (\gamma^2 I - D^T D)^{-1} \)

Also, note that

\[ D^T (\gamma^2 I - D D^T)^{-1} = (\gamma^2 I - D^T D)^{-1} \hat{D} D \]

so

\[ \chi_2 = -\{ A + B \hat{D} \hat{D}^T C \} \chi_2 \]

\[-\{ C^T (\gamma^2 I - D D^T)^{-1} \gamma^2 C \} \chi_1 \]

We would like to express \( (\gamma^2 I - D D^T)^{-1} \)
in terms of \( \hat{D} \), although this is not necessary. To do this, note that

\[ I = (\gamma^2 I - D D^T)^{-1} (\gamma^2 I - D D^T) \]

\[ = (\gamma^2 I - D D^T)^{-1} \gamma^2 - (\gamma^2 I - D D^T)^{-1} D D^T \]

\[ \Rightarrow (\gamma^2 I - D D^T)^{-1} \gamma^2 = I + (\gamma^2 I - D D^T)^{-1} D D^T \]

\[ = I + D (\gamma^2 I - D^T D)^{-1} D^T \]

\[ = I + D \hat{D} D^T \]
Therefore,

\[ \chi_2 = -(A + BD^{T}C)^{T}\chi_2 - (CT_C + CT_D \hat{D}D^{T}C)\chi_1 \]

Therefore, the Hamiltonian is

\[
H = \begin{bmatrix}
A + BD^{T}C + \hat{B}D^{T}B^{T} \\
-C^{T}C - CT_D \hat{D}DC - (A + BD^{T}C)^{T}
\end{bmatrix}
\]

\[ \hat{D} = (\gamma^2 I - D^{T}D)^{-1} \]

Having obtained the Hamiltonian, the algorithm is the same, except that we must start with the lower bound

\[ \gamma_{\text{lower}} = \overline{\gamma}(D) \]

See the Matlab function and test script on the following pages.
% % P54 Solution 7.4
% % ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% function y=infnorm2(A,B,C,D)
%
% initial guess for gamma

gamma_upper=Inf;
gamma_lower=max(svd(D));
gamma=gamma_lower*2+1; % guess bigger than gamma_lower and less than gamma_upper
%
% iterate until upper bound and lower bound converge

while (gamma_upper-gamma_lower)/gamma_upper > 10*eps | gamma_upper==Inf
%
% form the Hamiltonian matrix
%
[m,n]=size(D);
Dh=inv(eye(m)*gamma^2-D'*D);
H=[ A+B*Dh*D'*C B*Dh*D' ; -C'*C-Dh*D'*C -(A+B*Dh*D'*C) ];
%
% find the eigenvalues of H, and decide if any are on jω axis
% this part is tricky, so you may use a different approach.

d=eig(H);
if min(abs(real(d)))/max(abs(d)) < 1e-14
    jw=1;
else
    jw=0;
end
%
% based on result, move bounds

if jw % gamma is higher, so move up
    if gamma_upper==Inf
        gamma_lower=gamma;
        gamma=gamma*2;
    elseif gamma_lower=gamma;
        gamma=(gamma_upper+gamma_lower)/2;
    end
else % gamma is lower, so split the difference
    gamma_upper=gamma;
    gamma=(gamma_upper+gamma_lower)/2;
end
end
y=gamma ;
clear
A=[-0.1 1;
   -2 -0.3];
B=[1 2;
   3 4];
C=[1 1;
   1 -1];
D=[20 3;
   4 26];
infnorm2(A,B,C,D)

ans =
   43.6829  \| G \|_\infty