EEE 444 Robust Feedback Theory HW5 Report
Oğuz Altan – 21600966

Problem 1

Here we implement all sthe steps to solve the mixed sensitivity minimization problem

$$\gamma_{\text{opt}} = \inf_{C \in \mathcal{C}(P)} \left\| \begin{bmatrix} W_1 S \\ W_2 T \end{bmatrix} \right\|_\infty \quad \text{where } S(s) = \left(1 + P(s)C(s)\right)^{-1},\ T(s) = 1 - S(s)$$

for the plant $P(s) = \frac{N(s)}{D(s)}$

$$N(s) = \frac{20(s-0.5)(s-1)}{(s+10)(s^2+s+1)}$$

$$D(s) = \frac{(s^2-s+1)}{(s^2+s+1)}$$

and the weights

$$W_1(s) = \frac{1+0.1s}{s+0.1} \qquad W_2(s) = 0.1(s+0.1)$$

a) We perform innter-outer factorizations for N and D, the procedure as we take the zeros of the original transfer function in right half plane (RHP) and multiply it by some polynomial such that the magnitude of the resulting transfer function is 1. This transfer function is called inner function. If we divide the original transfer function by the inner transfer function, we get outer function. Now, we first factorize N such that:

$$N(s) = N_i(s)N_o(s)$$

Then, we get

$$N_i(s) = \frac{s^2 - 1.5s + 0.5}{s^2 + 1.5s + 0.5}$$

Note that the magnitude of $N_i(jw)$ is always 1 for all w's. In addition,

$$N_o(s) = \frac{N(s)}{N_i(s)} = \frac{20s^2 + 30s + 10}{s^3 + 11s^2 + 11s + 10}$$

Now, we do the same operation for D. First, we note that D is its inner function, with magnitude 1. Then, we can say

$$D(s) = D_i(s)D_o(s)$$

$$D_i(s) = D(s) \quad D_o(s) = 1$$

b) Now, we find $X, Y \in H_\infty$ such that the Bezout equation $N(s)X(s) + D(s)Y(s) = 1$ is satisfied, where $X(s)$ has the form

$$X(s) = \frac{x_1 s + x_0}{s+4}, \quad x1, x1 \in \mathrm{R}$$

Using the Bezout equation, we can write the $Y(s)$ as

$$Y(s) = \frac{1 - N(s)X(s)}{D(s)}$$

For $Y(s)$ to be stable, we must have zeros of $D(s)$ appear also as the zeros of $Y(s)$ so that pole-zero cancellation occurs. $D(s)$ have two zeros, which are $0.5 + 0.866j$ and $0.5 - 0.866j$.

We obtain two equations, two interpolation conditions, for $X(s)$:

$$X(10.5 + 0.866j\,) = 1/N(0.5 + 0.866j\,)$$

$$X(0.5 - 0.866j) = 1/N(0.5 - 0.866j)$$

Now, we need to find a stable transfer function $X(s)$ so that $X(s) \in H_\infty$ and satisfies these two interplolation conditions. Solving these two interpolation conditions to find two unknowns x_1 and x_2, we obtain

$x_1 = -4.6 \quad \& \quad x_2 = -1.5999$

$$X_n(s) = \frac{-4.6s - 1.5999}{s+4} \in H_\infty$$

Once we found $X(s)$, we can compute $Y(s)$ by using the equation of $Y(s)$ written above and doing pole-zero cancellations, we obtain

$$Y(s) = \frac{(s + 107.5)(s + 0.521)}{(s+10)(s+4)} \in H_\infty$$

c) We perform following spectral factorization and find satisfying G:

$$G(-s)G(s) = W_1(-s)W_1(s) + W_2(-s)W_2(s), \quad G, G^{-1} \in H_\infty$$

and all the poles and zeros of $G(s)$ are in left half plane (LHP).

Calculating right hand side of this factorization equation, we get

$$G(-s)G(s) = \frac{-0.01s^4 + 0.0102s^2 - 1}{s^2 - 0.01}$$

We know that $G \in H_\infty$, therefore after factorization of $G(-s)G(s)$, therefore, we obtain

$$G(s) = \frac{0.1(s^2 + 4.585s + 10)}{(s + 0.1)}$$

d) In this part, we perform bi-section search for γ_{opt} and C_{opt}. We first define

$$V(s) = \frac{W_1(s)W_2(s)}{G(s)}$$

Then, we compute $V(s)$ as

$$V(s) = \frac{0.1(s + 10)(s + 0.1)}{(s^2 + 4.585s + 10)}$$

Step 0: We let $\gamma_{max} = 100$ and $\gamma_{min} = \left\|V\right\|_\infty$

γ_{min} is the maximum of magnitude of $V(j\omega)$ and calculated as $\gamma_{min} = 0.2291$

Step 1: We let $\gamma = \frac{\gamma_{max} + \gamma_{min}}{2}$ and perform the spectral factorization

$$V\gamma(-s)V\gamma(s) = \gamma^2 - V(-s)V(s), \quad V\gamma, V\gamma$$

Note that, in the first iteration of this bi-section search, namely as we first set up and compute transfer functions, we compute the right hand side of this spectral factorization and get

$$V\gamma(-s)V\gamma(s) = \frac{2511.5(s^2 - 4.585s + 10)(s^2 + 4.585s + 10)}{(s^2 - 4.585s + 10)(s^2 + 4.585s + 10)}$$

Then, we factorize this result and get

$$V\gamma(s) = \frac{50.115(s^2 + 4.585s + 10)}{(s^2 + 4.585s + 10)}$$

As the value of γ changes in the each iteration, we expect this function $V\gamma(s)$ to change in each iteration, too.

Step 2: We define

$$R\gamma = V\gamma^{-1}(s)[N_i(-s)D_i(-s)W_1(-s)G^{-1}(-s)W_1(s) - D_i(-s)G(s)N_o(s)X(s)]$$

and check if there exists $Q_1 \in H_\infty$ such that $\left\|R\gamma - Q_1\right\|_\infty \leq 1$.

For this purpose, we solve the best achievable one-block problem performance level

$$\gamma_1 = inf\left|\left|R\gamma - Q_1\right|\right|_\infty$$

We solve this problem using Nehari approach, as we have worked on Homework #4, and ultimately we compute γ_1.

Then, we check if $\gamma_1 \leq 1$, if this condition is satisfied, we set $\gamma_{max} = \gamma$. Otherwise, we set $\gamma_{min} = \gamma$.

It can be seen that, the value of γ changes for different values of γ_{min} and γ_{max}. The aim is to find the optimal value of γ. To do this, we retake the step 1 and then 2 until we get

$$\gamma_{max} - \gamma_{min} \leq 0.00001$$

If we satisfy this condition, it means that we have found optimal γ, i.e. γ_{opt}.

After these iterations, we find γ_{opt} as

$$\gamma_{opt} = \ 3.4388$$

***Step 3*:** We define $\gamma_{opt} \ = \ \gamma$ and find the optimal solution to $Q_{1,opt} \ \in H_\infty$ of the one block problem for the latest value of γ as follows:

$$Q_{1,opt}(s) = R\gamma(s) - F_{opt}(s)$$

where

$$F_{opt}(s) = \frac{\gamma_1 C(sI - A)^{-1}x_{max}}{B^T(sI + A^T)^{-1}y_{max}} \in H_\infty$$

with $C(sI - A)^{-1}B$ being a state space realization of $R\gamma, u(s)$, the unstable part of $R\gamma(s)$. We have calculated, by MATLAB, the otimal one-block performance level γ_1 and found

$$\gamma_1 = \ 1.0000$$

as it is expected in theory.

We calculate $Q_{1,opt}(s)$ as

$$Q_{1,opt}(s) = \ 1.6755 \frac{(s + 2.824)(s + 1)(s + 0.5)(s + 0.07207)(s^2 \ - \ 0.4646s \ + \ 1.44)(s^2 \ + \ 4.585s \ + \ 10)}{(s + 10)\ (s + 4)\ (s + 0.1)\ (s + 0.04155)\ (s^2 \ + \ 1.455s \ + \ 0.5899)\ (s^2 \ + \ 4.667s \ + \ 10.2)}$$

Note that $Q_{1,opt}(s) \ \in H_\infty$, it is stable.

Step 4: We define $Q_{opt}(s) = V_\gamma(s)Q_{1,opt}(s)$ and

$$Q_c(s) = \frac{Q_{opt}(s)}{N_o(s)D_o(s)G(s)}$$

$$C_{opt}(s) = \frac{X(s) + D(s)Q_c(s)}{Y(s) - N(s)Q_c(s)}$$

After calculations, these functions are:

$$Q_{opt}(s) = 5.7614\frac{(s+2.824)(s+1)(s+0.5)(s+0.07207)(s^2-0.4646s+1.44)(s^2+4.585s+10)(s^2+4.584s+10)}{(s+4)(s+10)(s+0.1)(s+0.04155)(s\hat{}2+1.455s+0.5899)(s\hat{}2+4.585s+10)(s^2+4.667s+10.2)}$$

Note that $Q_{opt}(s) \in H_\infty$, it is stable.

$$Q_c(s) = 2.8807\frac{(s+2.824)(s+0.07207)(s^2+s+1)(s^2-0.4646s+1.44)(s^2+4.585s+10)(s^2+4.584s+10)}{(s+4)(s+0.04155)(s^2+1.455s+0.5899)(s^2+4.585s+10)^2(s^2+4.667s+10.2)}$$

Note that $Q_c(s) \in H_\infty$, it is stable.

$$C_{opt}(s) = -\frac{1.7193\left((s+4)^2(s+10)^2(s+0.09326)(s-0.06921)(s^2+s+1)(s^2+4.585s+10)(s^2+4.584s+10)\right)}{(s+41.39)(s+9.994)(s+4.001)(s+4)(s+0.164)(s+0.02051)(s^2+s+1)(s^2+4.585s+10)(s^2+4.584s+10)}$$

Note that $C_{opt}(s) \in H_\infty$, it is stable.

Step 5: Verification step. We first check that there is no unstable pole zero cancellation in the open loop transfer function $P(s)C_{opt}(s)$.

To do so, we first calculate the product $P(s)C_{opt}(s)$, it is equal to

$$-\frac{34.386\left((s-1)(s-0.5)(s+4)^2(s+10)^2(s-0.06921)(s+0.09326)(s^2+s+1)(s^2+4.585s+10)(s^2+4.584s+10)\right)}{(s+10)(s+9.994)(s+4.001)(s+4)(s+41.39)(s+0.164)(s+0.02051)(s^2+s+1)(s^2-s+1)(s^2+4.585s+10)(s^2+4.584s+10)}$$

Then, we again calculate the product $P(s)C_{opt}(s)$ by applying pole-zero cancellations on MATLAB using “minreal” command. Consequently, the pole-zero cancellation applied version of $P(s)C_{opt}(s)$ turns out to be

$$-\frac{34.386\left((s-1)(s-0.5)(s+4)^2(s+10)(s-0.06921)(s+0.09326)(s^2+s+1)(s^2+4.585s+10)(s^2+4.584s+10)\right)}{(s+9.994)(s+4.001)(s+4)(s+41.39)(s+0.164)(s+0.02051)(s^2+s+1)(s^2-s+1)(s^2+4.585s+10)(s^2+4.584s+10)}$$

We can observe that the cancelled term is (s+10) and the root of this term is on the LHP, therefore it is not unstable. Consequently, we see that there is not any unstable pole-zero cancellation on this tansfer function, $P(s)C_{opt}(s)$.

Then, we check that sensitivity function

$S_{opt}(s) = \left(1 + P(s)C_{opt}(s)\right)^{-1}$ is stable. We calculate $S_{opt}(s)$ as

$$\frac{(s+4.001)(s+9.994)(s+41.39)(s+0.164)(s+0.02051)(s^2+s+1)(s^2-s+1)(s^2+4.585s+10)(s^2+4.584s+10)}{(s+10.04)(s+3.98)(s+0.04155)(s^2+1.455s+0.5899)(s^2+s+1.001)(s^2+4.585s+10)^2(s^2+4.667s+10.2)}$$

Note that $S_{opt}(s) \in H_\infty$, it is stable.

We also verify that the optimal performance level is achieved, i.e.

$$\gamma_{opt} = \left(\left|W_1(j\omega)S_{opt}(j\omega)\right|^2 + \left|W_2(j\omega)(1 - S_{opt}(j\omega))\right|^2\right)^{\frac{1}{2}} \forall\omega$$

To do so, we look at the difference between the optimal γ that we have found and this γ_{opt} calculated as given above, for a range of frequency from $\omega = 0.01$ rad/s to $\omega = 1000$ rad/s

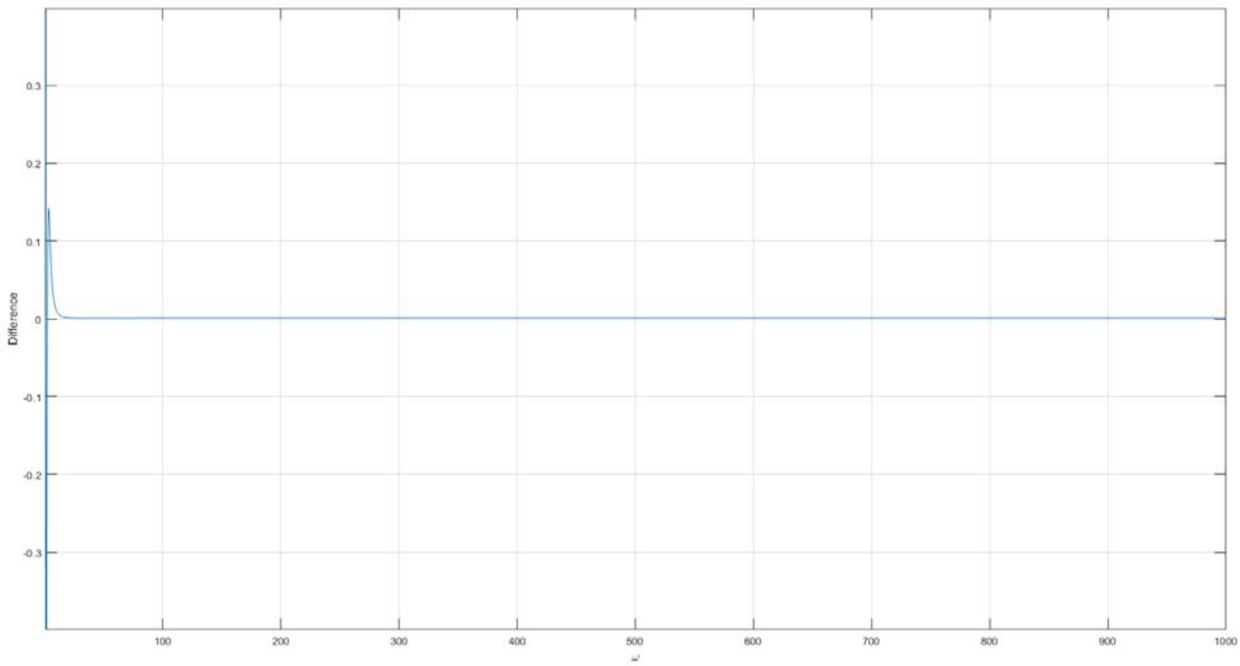

Figure 1: Difference between two γ_{opt}

From the plot, we can observe that the difference is nearly zero, meaning that they are nearly identically same. Therefore, we verify that the optimal performance level is achieved.

APPENDIX

```
clear all;
clc;

N = tf([20 -30 10],[1 11 11 10]);
D = tf([1 -1 1],[1 1 1]);
P = minreal(N/D);

Ni = tf([1 -1.5 0.5],[1 1.5 0.5]);
No = tf([20 30 10],[1 11 11 10]);
Di = D;
Do = 1;

r = roots([1 -1 1]);
a1 = 1/evalfr(N,r(1));
a2 = 1/evalfr(N,r(2));
X = zpk([-0.3478],[-4],[-4.6]);
display("X is stable: " + isstable(X));

Y = zpk([-107.5 -0.521],[-10 -4],[1]);
display("Y is stable: " + isstable(Y));

W1 = tf([0.1 1],[1 0.1]);
W2 = tf([0.1 0.01],[1]);
W1con = tf([0.1 -1],[1 -0.1]);
W2con = tf([-0.1 0.01],[1]);

Gspec = W1*W1con + W2*W2con;
[Gspecnum, Gspecden] = tfdata(Gspec, 'v');
rnum_G = roots(Gspecnum);
rden_G = roots(Gspecden);

left_poles = real(rnum_G) < 0;
left_zeros = real(rden_G) < 0;
G = zpk(rnum_G(left_poles),rden_G(left_zeros),(1));
[~,~,kG] = zpkdata(Gspec);
G = sqrt(-kG)*G;
display("G is stable: " + isstable(G));

V = minreal(W1*W2/G);
Vcon = tf([0.01 -0.0101 0.01],[1 -4.585 10]);

Nicon = tf([1 1.5 0.5],[1 -1.5 0.5]);
Dicon = tf([1 1 1],[1 -1 1]);
W1con = tf([-0.1 1],[-1 0.1]);

Gcon = tf([1 -4.585 10],[-1 0.1]);

%Step 0
gamma_max = 100;
[mag,~] = bode(V);
gamma_min = max(mag);

while(gamma_max - gamma_min > 0.00001)

    %Step 1
```

```
    gamma = (gamma_min + gamma_max)/2;
    Vgamma_spectral = gamma^2 - Vcon*V;
    [Vgamma_spectral_num, Vgamma_spectral_den] = tfdata(Vgamma_spectral,
'v');
    rnum_vgamma = roots(Vgamma_spectral_num);
    rden_vgamma = roots(Vgamma_spectral_den);

    left_poles = real(rnum_vgamma) < 0;
    left_zeros = real(rden_vgamma) < 0;

    [~,~,kgamma] = zpkdata(Vgamma_spectral);

    Vgamma =
zpk(rnum_vgamma(left_poles),rden_vgamma(left_zeros),(sqrt(kgamma)));
    display("Vgamma is stable: " + isstable(Vgamma));
    display("1/Vgamma is stable: " + isstable(1/Vgamma));

    %Step 2
    Rgamma = minreal((1/Vgamma)*(Nicon*Dicon*W1con/Gcon*W1-Dicon*G*No*X));

    %Nehari
    [Rs,Ru] = stabsep(minreal(Rgamma));

    [A,B,C,~] = ssdata(tf(Ru));

    Wc = lyap(A,-B*B');
    Wo = lyap(A',-C'*C);

    gamma1 = sqrt(max(eig(Wc*Wo)));
    [V_eig,D_eig] = eig(Wc*Wo);
    xmax = V_eig(:,1);
    ymax = (1/gamma1)*Wo*xmax;

    if (gamma1 < 1 || gamma1 == 1)
        gamma_max = gamma;
    else
        gamma_min = gamma;
    end
end

%Step 1
gamma = (gamma_min + gamma_max)/2;
Vgamma_spectral = gamma^2 - Vcon*V;
[Vgamma_spectral_num, Vgamma_spectral_den] = tfdata(Vgamma_spectral, 'v');
rnum_vgamma = roots(Vgamma_spectral_num);
rden_vgamma = roots(Vgamma_spectral_den);

left_poles = real(rnum_vgamma) < 0;
left_zeros = real(rden_vgamma) < 0;

[~,~,kgamma] = zpkdata(Vgamma_spectral);

Vgamma =
zpk(rnum_vgamma(left_poles),rden_vgamma(left_zeros),(sqrt(kgamma)));
display("Vgamma is stable: " + isstable(Vgamma));
display("1/Vgamma is stable: " + isstable(1/Vgamma));
```

```
%Step 2
Rgamma = minreal((1/Vgamma)*(Nicon*Dicon*W1con/Gcon*W1-Dicon*G*No*X));

%Nehari
[Rs,Ru] = stabsep(minreal(Rgamma));

[A,B,C,~] = ssdata(tf(Ru));

Wc = lyap(A,-B*B');
Wo = lyap(A',-C'*C);

gamma1 = sqrt(max(eig(Wc*Wo)));
[V_eig,D_eig] = eig(Wc*Wo);
xmax = V_eig(:,1);
ymax = (1/gamma1)*Wo*xmax;

if (gamma1 < 1 || gamma1 == 1)
    gamma_max = gamma;
else
    gamma_min = gamma;
end

%Step 3
gamma_opt = gamma;

Q1opt = Rgamma - gamma1*(ss(A,xmax,C,0)/ss(-A',ymax,B',0));
Q1opt = zpk(minreal(Q1opt));
display("Q1opt is stable: " + isstable(Q1opt));

%Step 4
% Qopt = (Vgamma*Q1opt);
Qopt = minreal(Vgamma*Q1opt);
display("Qopt is stable: " + isstable(Qopt));

Qc = minreal(Qopt/(No*Do*G));
display("Qc is stable: " + isstable(Qc));

Copt = minreal((X+D*Qc)/(Y-N*Qc));

openlooppolezero = P*Copt;
openlooptf = minreal(P*Copt);
Sopt = minreal(1/(1+openlooptf));
display("Sopt is stable: " + isstable(Sopt));

% nyquist(openlooptf);

w = 0.001:0.01:1000;
[mag1,~] = bode(W1*Sopt,w);
[mag2,~] = bode(W2*(1-Sopt),w);
mag1 = squeeze(mag1);
mag2 = squeeze(mag2);

gamma_opt_2 = sqrt(mag1.^2 + mag2.^2);
gamma_opt_2_t = (gamma_opt_2-gamma_opt)';
figure;
plot(w,gamma_opt_2_t);
grid on;
```

```
xlabel("\omega");
ylabel("Difference");
```