CONTROL SYSTEMS II (REGELSYSTEME II)

SOLUTIONS TO PROBLEM SET 11

Exercise 1  Robust Performance with full-block/diagonal Input Uncertainty

The $\mu$ curves for diagonal and full-block input uncertainty are shown in Fig. 1. The value of $\mu$ for diagonal uncertainty is much smaller than that of the case discussed in Section 8.11.3.

A block diagram of the system with input uncertainty and weighted sensitivity output is shown in Figure 2.

The exercise can be solved with the Simulink file depicted in Fig. 3, in conjunction with the m-file:

```matlab
1 % CS2 problem set 11; exercise 1
s = tf('s');
tau = 100;
G = 1/(tau*s+1)*[-87.8 1.4; -108.2 -1.4];
K = (0.7/s)/(G);
Wi = (s+0.2)/(0.5*s+1)*eye(2);
Wp = (s/2+0.05)/s*eye(2);
[A,B,C,D] = linmod('prob11_1_simulinkfile');
N = ss(A,B,C,D);
BLKfull = [2 2;2 2];

16 BLKstructured = [1 1; 1 1; 2 2];
f = logspace(-2,2,100);
Nf = frd(N,f);
lb1 = bounds(Nf,BLKfull);
lb2 = bounds(Nf,BLKstructured);
figure;
[Mu,bounds,MUINFO] = mussv(Nf,BLKfull);
return
```

Figure 1: $\mu$ curves for diagonal (−−) and full (−) uncertainty.
Exercise 2  \( (H_\infty) \) Mixed-Sensitivity Problem: Airplane Control

We first define the plant and the weighting functions:

```matlab
A = [-0.0226 -36.6170 -18.8970 -32.0900 3.2509 -0.7626; ... 0.0001 -1.8997 0.9831 -0.0007 -0.1708 -0.0050; ... 0.0123 11.7200 -2.6316 0.0009 -31.6040 22.3960; ... 0 0 1.0000 0 0 0; ... 6 0 0 0 0 -30.0000 0; ... 0 0 0 0 0 -30.0000];
B = [0 0; 0 0; 0 0; 0 0; 30 0; 0 30];
C = [0 1 0 0 0 0; 0 0 1 0 0 0];
D = zeros(2,2);
G_ss = ss(A,B,C,D);

% weights
kS = 1;
r = 0.5;
kT = 0.001;
WS = kS * tf({[1 100] [0]; [0] [1 100]},{[100 1] 1; 1 [100 1]});
WT = kT * tf({[1 0 0] [0]; [0] [r 1 0 0]},{[1000] 1; 1 [1000]});
```

Figure 2: Block diagram of the system with clarification of the generalized plant

Figure 3: Simulink file `prob11_l_simulinkfile.mdl`. Note that the input port dimensions should be set to 2 (not -1 for inherited)
The generalized plant is given as:

\[
P(s) = \begin{bmatrix} W_S & -W_SG \\ 0 & W_TG \\ I & -G \end{bmatrix}(s)
\]

There are different ways of creating the generalized plant in Matlab, here, we chose to use the appropriate function from the robust control toolbox:

```matlab
%% ROBUST CONTROL TOOLBOX

%% 1. generalized plant
GenSys = augtf(sys,WS,[],WT);

%% alternative: generalized plant by hand
sys_tf = tf(ss(A,B,C,D));
zero_tf = tf(0,1);
one_tf = tf(1,1);
GenSys_tf = [WS (-WS*sys_tf); ...
[zero_tf,zero_tf;zero_tf,zero_tf] (WT *sys_tf); ...
[one_tf, zero_tf; zero_tf one_tf] -sys_tf];
\[Agen,Bgen,Cgen,Dgen\] = ssdata(minreal(ss(GenSys_tf)));
GenSys = ss(Agen,Bgen,Cgen,Dgen);

%% alternative: generalized plant in simulink
GenSys = linmod(gensys.mdl);
```

2. A stabilizing controller that minimizes the closed-loop transfer function 
\( z = F_L(P, K)w \) can be found by invoking the appropriate function of the robust control toolbox:

```matlab
%% 2. synthesis
nmeas = 2; % number of measured outputs
ncon = 2; % number of controlled inputs
gmin = 0; % default
gmax = Inf; % default
tol = 1e-2; % default, don't use other tol, otherwise
methd = 'ric'; % default
quiet = 'on'; % default

[SS_K,SS_CL,GAM_OPT,INFO] = hinfsyn(GenSys,nmeas,ncon,...
'DISPLAY',quiet, 'TOLGAM',tol,'METHOD',methd,...
'GMIN',gmin,'GMAX',gmax);
\[\text{eig}(SS_K)\] % check if eigenvals are to big...
```

3.4. The relevant plots are created by:

```matlab
figure(1);
LFT_sys = lft(GenSys,SS_K,2,2);
sigma(LFT_sys)
title('Closed Loop')
```

Figure 4: \( S/T \) mixed-sensitivity optimization in standard form.
6 % Sensitivity function
figure(2);
S = (eye(2) + G_ss * SS_K)ˆ(-1);
sigma(S,'g-','b:');
title('Sensitivity');

11 % Complementary Sensitivity function
figure(3);
T = eye(2) - S;
WT_inv = tf({[1000] 0; 0 [1000]},{[1 0 0] [1];[1] [1 0 0]});
sigma(T,'g-',WT_inv,'b:');
title('Complementary Sensitivity');

16 % Step responses
figure(4);
step(T,0.5)

---

**Figure 5:** Singular value Bode plot of the closed-loop transfer function $F_L(P, K)$.

**Figure 6:** Singular value Bode plot of $S(s)$ together with $W_S(s)^{-1}$. 
Exercise 3  Airplane Lateral Control via $\mu$-Synthesis and DK-Iteration

1. All solutions are basically given in the F-14 Lateral Axis demo.

2. The term handling quality (HQ) model which is used in this demo, stands for reference model and describes the desired dynamic behaviour of the closed-loop system. While the $p$ HQ model describes the desired response of the roll rate $p$ to the input from the lateral stick, the $\beta$ HQ model relates the desired side-slip angle $\beta$ to the rudder pedal.

3. $W_{act}$: constant weight with the inverse of the actuator limitations to guarantee the specifications. $W_{in}$: high-pass filter to model the sensor noise in the roll rate, yaw rate and lateral acceleration channels. $W_{p}$ and $W_{\beta}$: band-pass filters to emphasize the frequency range between 0.06 and 30 rad/sec (due to a right-half plane zero in the model at 0.002 rad/sec, an accurate tracking below 0.002 rad/sec is not possible). $W_{in}$: a high-pass filter which reflects the modelling errors of the F-14 model due to unmodeled dynamics i.e. the model is more uncertain at higher frequencies.

4. The signal names are taken from the MATLAB F-14 demo.

exogenous input $w$: [roll_cmd beta_cmd sn_noise Delta_G]'
exogenous output $w$: [delta_dstab delta_rud W_act W_beta W_p]'
control inputs $y_m$: [roll_cmd beta_cmd antia_filt]'
control signals $u$: [delta_dstab delta_rud]'
The generalized plant of the complete system can be seen in Fig. 7.

5. The final value of $\mu$ is 1.2322. A value $> 1$ means that the specifications could NOT be met completely. Anyway, 1.2322 is not much bigger than 1, thus the violation of the specifications may not be severe. By plotting the structured singular value $\mu$ over the frequency for the nominal plant and the worst-case, more insight into problematic frequencies can be yielded.

6. Figure 8 shows the worst-case structured singular value over frequency for the $\mu$-synthesis controllers computed in iteration 1 - 5. The plot has been produced by the following code:

```matlab
% Uses the Robust Control Toolbox
% Has to be run after the F-14 Lateral Axis demo
close all
clear

% examined frequency
fmu = logspace(-2, 2, 60); % examined frequency

for i = 1:5
    % Compute $\mu$-synthesis controller
    opt = dkitopt('FrequencyVector', fmu, 'NumberofAutoIterations', i);
    [kmu, clpmu, bnd] = dksyn(F14IC, nmeas, nctrls, opt);
    % Frequency Response Data of closed-loop system
    clmu = lft(F14IC, kmu);
    % Compute worst-case gain for kmu
    opt = wcgopt('FreqPtWise', 1); % options for WCGAIN
    [mgmu, wcumu, infomu] = wcgain(clmu, opt);
    mu{i} = mgmu.UpperBound;
end

% plot $\mu$ over frequency
semilogx(mu{1}, 'b', mu{2}, 'r', mu{3}, 'g', mu{4}, 'k', mu{5}, 'y');
title('Worst-case gains for $\mu$-synthesis controllers')
xlabel('Frequency (rad/sec)')
ylabel('Closed-loop gain')
legend('Iter. 1', 'Iter. 2', 'Iter. 3', 'Iter. 4', 'Iter. 5', 'Location', 'Best')
```
7. The $H_\infty$-optimal controller, which was computed by hinfsyn, is minimizing the $H_\infty$-norm of the nominal plant. Thus, it is not considering any robustness against model uncertainties like the $\mu$-synthesis controller computed by dksyn does. While the $H_\infty$-norm of a MIMO system (which equals the maximum of the largest singular value $\hat{\sigma}$ over frequency) does not exploit a given structure of the uncertainty, the structured singular value $\mu$ does, which yields a less conservative controller.
Figure 8: Worst-case gains for $\mu$-synthesis controllers.
Exercise 4  Digital Control

1. The ZOH-equivalent of \( G(s) \) is computed as

\[
G_d(z) = \left( \frac{z - 1}{z} \right) \cdot \mathcal{Z} \left( \frac{G(s)}{s} \right) = \left( \frac{z - 1}{z} \right) \cdot \mathcal{Z} \left( \frac{1}{s^2} \right) = \left( \frac{z - 1}{z} \right) \cdot \frac{\tau z}{(z - 1)^2} = \frac{\tau}{z - 1}
\]

(1)

2. The Tustin approximation without prewarping of \( K(s) \) is

\[
K_\tau(z) = K(\tilde{s}) \quad \text{with} \quad \tilde{s} = \frac{2}{\tau} \cdot \frac{z - 1}{z + 1},
\]

resulting in

\[
K_\tau(z) = 1 + \frac{1}{\tilde{s}} = 1 + \frac{\tau}{2} \cdot \frac{z + 1}{z - 1} = \frac{2(z - 1) + \tau(z + 1)}{2(z - 1)} = \frac{(2 + \tau)z + (\tau - 2)}{2(z - 1)}
\]

(2)

3. The closed loop transfer function in the \( z \)-domain is

\[
T_d(z) = \frac{K_\tau(z)G_d(z)}{1 + K_\tau(z)G_d(z)} = \frac{\tau(2 + \tau)z + \tau(\tau - 2)}{2(z - 1)^2 + \tau(2 + \tau)z + \tau(\tau - 2)}
\]

(4)

The pole polynomial is

\[
\phi(z) = z^2 + \frac{\tau(2 + \tau) - 4}{2} \cdot z + \frac{\tau(\tau - 2) + 2}{2}
\]

(5)

The discriminant of the polynomial \( \phi(z) = 0 \) is

\[
D = \left( \frac{\tau(2 + \tau) - 4}{2} \right)^2 - 4 \left( \frac{\tau(\tau - 2) + 2}{2} \right) = \frac{\tau^2(\tau^2 + 4\tau - 12)}{4},
\]

which satisfies \( D \leq 0 \) for all \( 0 < \tau \leq 2 \). This implies that \( \phi(z) = 0 \) has a complex root \( z_0 = a \pm jb \) for all \( \tau \) in that interval. To check the stability requirement, whether the poles leave the unit circle, i.e. \( a^2 + b^2 = 1 \), observe that the polynomial with complex roots must have the form

\[
\phi(z) = (z - z_0)(z - z_0^*) = z^2 - 2az + a^2 + b^2.
\]

(7)

Hence the stability requirement becomes

\[
\frac{\tau(\tau - 2) + 2}{2} = 1 \quad \iff \quad \tau - 2 = 0,
\]

thereby providing the critical sampling time \( \tau = 2 \), with both roots at \( z_0 = -1 \pm j0 \).

4. The control input applied to the system is shown in Figure 9. It can be seen that the overshoot and the worst case value of the control input becomes bigger with increasing sampling times.

5. The closed loop plant output is shown in Figure 10. It can be seen that the overshoot becomes bigger with increasing sampling times. One can generate the trajectories of the continuous time system \( G(s) \) with the discretized input from \( K_\tau(z) \) using the source code:
Figure 9: Reference step response of the control input using $K(s)$ (blue) and $K_{\tau}(z)$ with $\tau = 0.25$ (red), $\tau = 0.5$ (green), $\tau = 0.75$ (magenta), $\tau = 1$ (cyan),
Figure 10: Reference step response of the plant output using $K(s)$ (blue) and $K_{\tau}(z)$ with $\tau = 0.25$ (red), $\tau = 0.5$ (green), $\tau = 0.75$ (magenta), $\tau = 1$ (cyan).

```matlab
clear
clc
close all
5 %% sampling times
vTd = [.25 .5 .75 1];
cs = {'r' 'g' 'm' 'c'}
%% define plant
10 s = tf([1 0],1);
K = 1 + 1/s
L = G*K
% plot continuous plant output
15 Tcont = minreal(L/(1+L));
figure(1); step(Tcont,'b');hold on;grid on;
% plot continuous plant input
Tu= minreal(K/(1+L));
figure(2); step(Tu,'b');hold on;grid on;
%% plot intervals
20 Ts = 0.001;
for i = 1:length(vTd)
    % construct discrete time closed loop
    25 Td = vTd(i);
    Gz = c2d(G,Td)
    Kz = c2d(K,Td,'tustin')
    Lz = Gz*Kz
    Tz = minreal(Lz/(1+Lz));
    %% simulate discrete time input
    35 [uz, tz] = step(Tz);
    %% simulate discrete time output
    [yz, tz] = step(Tz);
    %% construct stair input
    ufull = kron(uz(1:end-1),ones(Ni,1));
    tfull = 0:Td:(tz(end)-Tt);
    %% simulate continuous plant output
    yfull = lsim(G,ufull,tfull);
    %% plot plant output
    45 figure(1);plot(tfull,yfull, cs{i});hold on;
    figure(1);plot(tz,yz, [cs{i} '-*']);
    %% plot plant input
    figure(2);plot(tfull,ufull, cs{i} '-');hold on;
    figure(2);plot(tz,uz, [cs{i} '*']);
end
50 figure(1);axis auto;
figure(2);axis auto;grid on; a = axis; axis([0 10 a(3:4)]);```

10 of 10