1: 20 points

Spinning Satellite Control

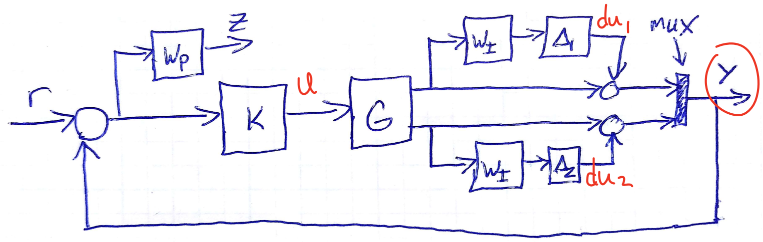
In a prior problem we considered a system with plant model

$$G(s) = \frac{1}{s^2 + \frac{a}{50}s + a^2} \begin{bmatrix} s - a^2 & a(s+1) \\ -a(s+1) & s - a^2 \end{bmatrix}$$

These dynamics model the motion of a cylindrical satellite spinning at constant rate Ω about the z-axis with $a = (1 - I_{zz}/I_{xx})\Omega$. The controls affect the spin rates about x and y; as a becomes large, the system becomes increasingly sensitive to uncertainty. Assume $a = 100$ throughout.

2X2 2X2

2x2 2x2



ud1
ud2

(a) Assuming multiplicative output uncertainty of $W_I(s) = \frac{20s+10}{s+100}$ in the configuration shown and a performance weight of $W_P(s) = \frac{0.5s+20}{s+0.2} I_{2\times 2}$, design a controller using the command *mixsyn*. Plot the achieved sensitivity function. Check whether you met your RP / RS goals.

(b) Assuming the same uncertainty model, design a controller using μ synthesis. Plot the achieved sensitivity function, check whether RP / RS are met, and compare the results with the H_∞ design.

2: 30 points

Robust Stability and Performance

(a) Consider the feedback system shown with a scalar plant having both multiplicative and additive uncertainty, i.e.

$$P_p = P(1 + W_1\Delta_1) + W_2\Delta_2$$

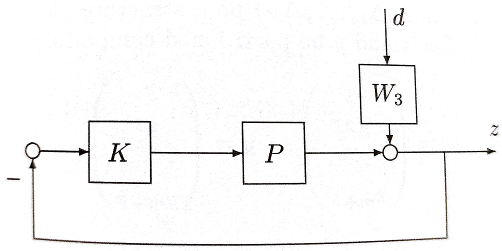
and $\|\Delta_i\|_\infty < 1$. Assume that W_1 and W_2 are stable and show the following.

1. The feedback system is robustly stable $\iff$ K stabilizes P and

$$\| |W_1 T| + |W_2 KS| \|_\infty \le 1.$$

2. The feedback system has robust performance $\iff$ K stabilizes P and

$$\| |W_3 S| + |W_1 T| + |W_2 KS| \|_\infty \le 1.$$

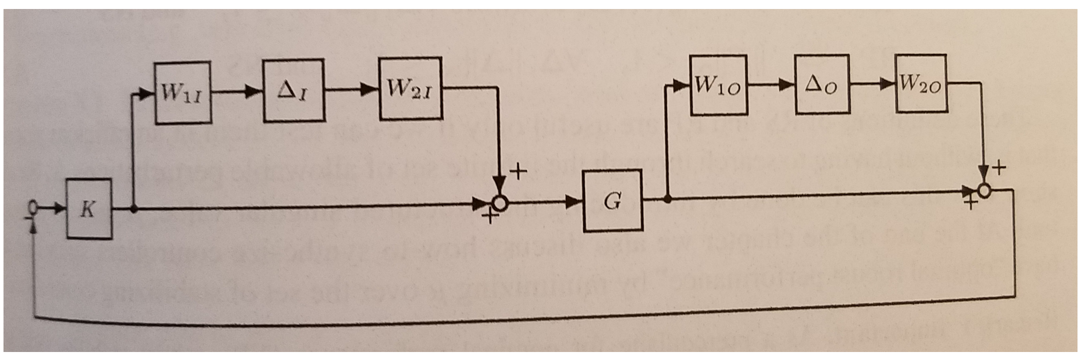


(b) Consider the block diagram shown where we have both input and output multiplicative uncertainty blocks. The set of possible plants is given by

$$G_p = (I + W_{2O}\Delta_O W_{1O})G(I + W_{2I}\Delta_I W_{1I}),$$

where $\|\Delta_I\|_\infty \leq 1$ and $\|\Delta_O\|_\infty \leq 1$. Collect the perturbations into $\Delta = \text{diag}\{\Delta_I, \Delta_O\}$ and rearrange the system into the $M - \Delta$ structure. Show that

$$M = \begin{bmatrix} W_{1I} & 0 \\ 0 & W_{1O} \end{bmatrix} \begin{bmatrix} -T_I & -KS \\ SG & -T \end{bmatrix} \begin{bmatrix} W_{2I} & 0 \\ 0 & W_{2O} \end{bmatrix}$$



(c) Consider the plant model

$$G(s) = \frac{1}{75s + 1} \begin{bmatrix} -87.8 & 1.4 \\ -108.2 & -1.4 \end{bmatrix},$$

which is ill-conditioned with $\gamma(G) = 70.8$ at all frequencies. With an inverse-based controller $K(s) = \frac{0.7}{s} G^{-1}(s)$ and uncertainty and performance weights $w_I = \frac{s+0.2}{0.5s+1}$ and $w_P = \frac{s/2+0.05}{s}$, compute μ for RP with both diagonal and full-block input uncertainty.

3: 20 points

Norm Computations

(a) Write your own function to compute the ∞-norm of an arbitrary system (assumed in transfer function matrix form). Test your performance on the system

$$G(s) = \begin{bmatrix} \frac{1}{s+5} & \frac{s-2}{s+10} \\ \frac{10}{s^2+4s+15} & \frac{3s}{4s+3} \end{bmatrix}$$

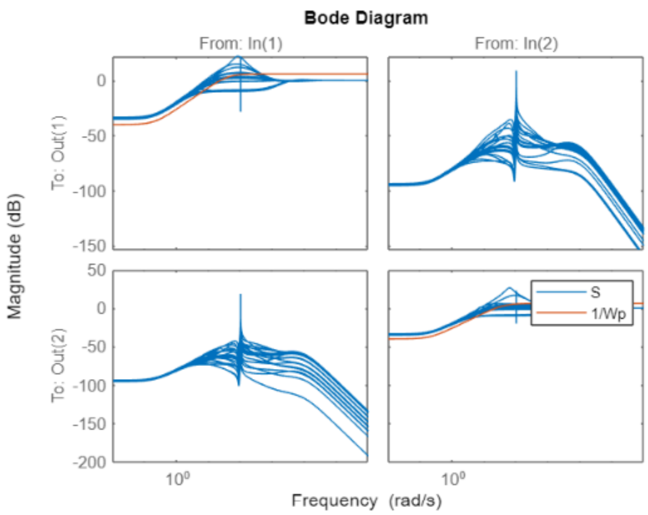
(b) Repeat Part b for the 2-norm. Because the 2-norm is finite only for strictly proper systems, test it on

$$G(s) = \begin{bmatrix} \frac{1}{s+5} & \frac{1}{s+10} \\ \frac{10}{s^2+4s+15} & \frac{3}{4s+3} \end{bmatrix}$$

or some other suitably modified version of the plant from Part b.

1a

```
s = tf('s');
a = 100;
G = 1/(s^2+(a/50)*s+a^2) * [s-a^2 a*(s+1); -a*(s+1) s-a^2];
WI = (20*s+10)/(s+100) * eye(2);
Wp = (0.5*s+20)/(s+0.2) * eye(2);
[K,CL,GAM] = mixsyn(G,Wp,[],WI);
delta1 = ultidyn('delta1',[1,1]);
delta2 = ultidyn('delta2',[1,1]);
delta = [delta1 0; 0 delta2];
G_hat = (eye(2)+delta*WI) * G; %structured uncertainty
S = 1/(eye(2)+G_hat*K);
%T = 1-S;
bodemag(S, 1/Wp)
legend("S","1/Wp")
```



Check RS

```
[STABMARG,WCU] = robstab(S);
mu = 1/STABMARG.LowerBound
```

```
mu = 1.8599
```

Check RP

```
[perfmarg,WCU] = robustperf(Wp*S);
mu = 1/perfmarg.LowerBound
```

```
mu = 2.7252
```

1b

Musyn design

```
%build musyn system
delta1 = ultidyn('delta1',[1,1]);
delta2 = ultidyn('delta2',[1,1]);
G_hat = (eye(2)+[delta1 0;0 delta2]*WI) * G; %structured uncertainty
%need include uncertainty
systemnames = 'G_hat Wp'; %Block name only
```

Now we have to define the inputs and outputs to the system by giving values to *inputvar* and *outputvar*.

```
inputvar = '[r{2};u{2}]';
outputvar = '[Wp;r-G_hat]'; %Strangely, the system outputs are just the name
```

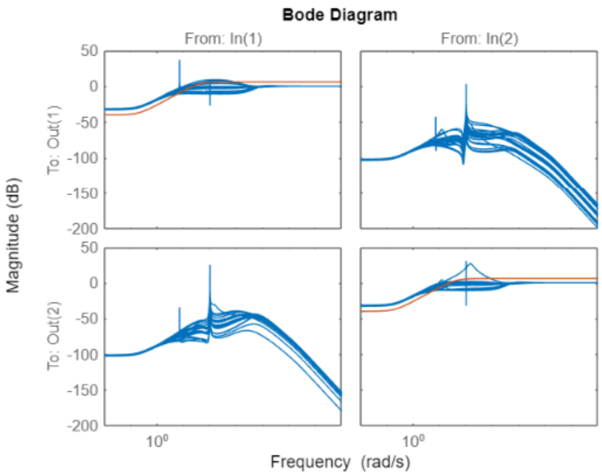
Here the variables are partitioned as before, i.e. the input variables are listed as $[w, u]^T$. Now we need to give the inputs to each system.

```
input_to_G_hat = '[u]';
input_to_Wp = '[r-G_hat]';
cleanupsysic = 'yes';  %This drops all the useless variables from workspace
Pmu = sysic;
[Kmu,CLP,mu_mu] = musyn(Pmu,2,2);  %Same usage as Hinfsyn
```

```
D-K ITERATION SUMMARY:
-----------------------------------------------------------------
                   Robust performance                Fit order
-----------------------------------------------------------------
  Iter        K Step        Peak MU       D Fit             D
    1          2.717         2.711        2.734            12
    2          2.589         2.589        2.614            20
    3          2.589         2.588        2.588            20
    4           2.59         2.589        2.592            20

Best achieved robust performance: 2.59
```

```
Smu =inv(eye(2)+G_hat*Kmu);
bodemag(Smu,1/Wp)
```




```
%Robust stabilty
[STABMARG,WCU] = robstab(Smu);
mu_mu_stab = 1/STABMARG.LowerBound
```

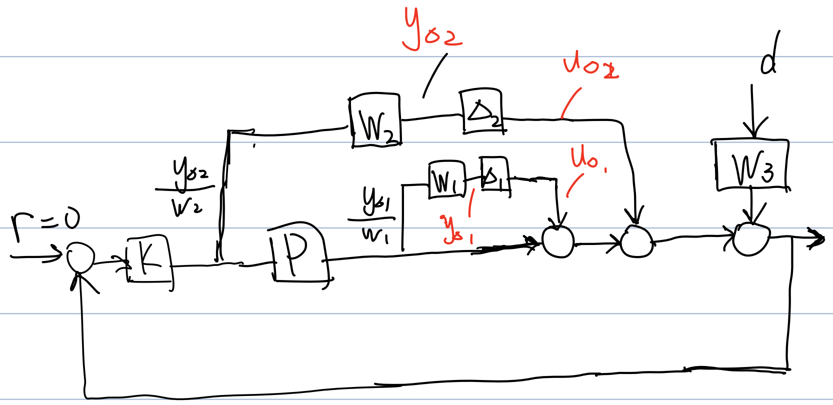
```
mu_mu_stab = 2.0993
```

```
%robust performance
N_mu = lft(Pmu,Kmu);
perfmarg_mu = robustperf(N_mu);
mu_mu_perf = 1/perfmarg_mu.LowerBound
```

```
mu_mu_perf = 2.5879
```

2a.

(1)



$$y_{\Delta 2} = W_2 K\left[-u_{\Delta 2} - u_{\Delta 1} - P(W_2^{-1} y_{\Delta 2})\right]$$

$$y_{\Delta 2} = -W_2 K u_{\Delta 2} - W_2 K u_{\Delta 1} - W_2 K P(W_2^{-1} y_{\Delta 2})$$

$$y_{\Delta 2} = -W_2 K u_{\Delta 2} - W_2 K u_{\Delta 1} - KP y_{\Delta 2}$$

$$y_{\Delta 2} + KP y_{\Delta 2} = -W_2 K u_{\Delta 2} - W_2 K u_{\Delta 1}$$

$$(1+KP) y_{\Delta 2} = -W_2 K (u_{\Delta 1} + u_{\Delta 2})$$

$$y_{\Delta 2} = -W_2 K S (u_{\Delta 1} + u_{\Delta 2})$$

$$y_{\delta_1} = W_1 P K \left(0 - u_{\delta_1} - u_{\delta_2} - \frac{y_{\delta_1}}{W_1}\right)$$

$$y_{\delta_1} = -W_1 P K u_{\delta_1} - W_1 P K u_{\delta_2} - P K y_{\delta_1}$$

$$(1+PK) y_{\delta_1} = -W_1 P K (u_{\delta_1} + u_{\delta_2})$$

$$y_{\delta_1} = -W_1 T (u_{\delta_1} + u_{\delta_2})$$

$$\begin{bmatrix} y_{\delta_1} \\ y_{\delta_2} \end{bmatrix} = \begin{bmatrix} -W_1 T & -W_1 T \\ -W_2 K S & -W_2 K S \end{bmatrix} \begin{bmatrix} u_{\delta_1} \\ u_{\delta_2} \end{bmatrix}$$

$$\left| I - M_\delta \right| = \begin{vmatrix} 1 + W_1 T \delta_1 & W_1 T \delta_2 \\ W_2 K S \delta_1 & 1 + W_2 K S \delta_2 \end{vmatrix}$$

$$(1 + W_1 T \delta_1)(1 + W_2 K S \delta_2) - (W_1 T \delta_2)(W_2 K S \delta_1)$$

$$1 + W_2 K S \delta_2 + W_1 T \delta_1 + W_1 W_2 T K S \delta_1 \delta_2 - W_1 W_2 T K S \delta_1 \delta_2 = 0$$

$$1 + W_2 K S \delta_2 + W_1 T \delta_1 = 0$$

$$|\delta_1| = |\delta_2| = \delta \qquad 1 - |W_1 T||\delta| - |W_2 K S||\delta| = 0$$

$$\frac{1}{|\delta|} = |W_1 T| + |W_2 K S| \quad , \quad u = \left\| \frac{1}{|\delta|} \right\|_\infty$$

if K stablized P, and $\| |W_1 T| + |W_2 KS| \|_\infty \leq 1$

$\therefore \mu < 1 \therefore$ System is robustly stable.

(2)

$$\begin{array}{ccc} & P_{11} & P_{12} \end{array}$$

$$\begin{bmatrix} y_{\delta_1} \\ y_{\delta_2} \\ y_{\delta_3} \\ z \\ v \end{bmatrix} = \left[\begin{array}{ccc|c} 0 & 0 & 0 & W_1 \\ 0 & 0 & 0 & W_2 \\ G & 1 & W_2 & G \\ G & 1 & W_3 & G \\ \hline -G & -1 & -W_3 & -G \end{array}\right] \begin{bmatrix} u_{\delta_1} \\ u_{\delta_2} \\ u_{\delta_3} \\ d \\ u \end{bmatrix}$$

$$\begin{array}{ccc} & P_{21} & P_{22} \end{array}$$

$$N = F_l(P, K) = P_{11} + P_{12} K (I + GK)^{-1} P_{21} = \begin{bmatrix} -W_1 T & -W_1 KS & W_1 KS W_3 \\ -W_2 T & -W_2 KS & -W_2 KS W_3 \\ GS & S & W_3 S \\ GS & S & W_3 S \end{bmatrix}$$

$$\det(I - M\Delta)$$

$$= \begin{bmatrix} 1 + W_1 T \Delta_1 & W_1 KS \Delta_2 & W_1 KS W_3 \Delta_3 \\ W_2 T \Delta_1 & 1 + W_2 KS \Delta_2 & W_2 KS W_3 \Delta \\ -GS\Delta_1 & -S\Delta_2 & 1 - W_3 S \Delta_3 \end{bmatrix}$$

$$= 1 + W_1 T \Delta_1 + W_2 KS \delta_2 - S W_3 \Delta_3 = 0$$

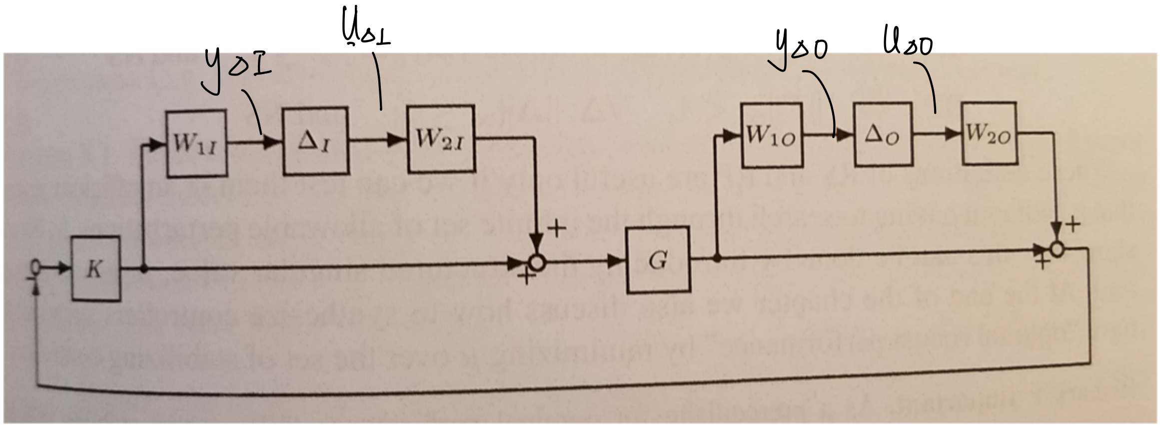
$$|\delta_1| = |\delta_2| = |\delta_3| = \Delta$$

$$\therefore -|W_1 T| - |W_2 KS| - |W_3 S| = \frac{1}{|\delta|}$$

$$u = \left\| \frac{1}{|\delta|} \right\|_\infty \leq 1$$

$$\therefore \left\| \, |W_3 S| + |W_1 T| + |W_2 K S| \, \right\|_\infty \leq 1$$

2b.



$$G_p = (I + W_{2O} \Delta_O W_{1O}) G (I + W_{2I} \Delta_I W_{1I})$$

$$y_{\Delta I} = W_{1I} K \left[0 - u_{\Delta O} W_{2O} - G\left(u_{\Delta I} W_{2I} + \frac{y_{\Delta I}}{W_{1I}} \right) \right]$$

$$y_{\Delta I} = -W_{1I} K \left(u_{\Delta O} W_{2O} + G u_{\Delta I} W_{2I} + \frac{G y_{\Delta I}}{W_{1I}} \right)$$

$$y_{\Delta I} = -W_{1I} W_{2O} K u_{\Delta O} - W_{1I} W_{2I} K G u_{\Delta I} - G y_{\Delta I} K$$

$$(1 + KG) y_{\Delta I} = -W_{1I} W_{2O} K u_{\Delta O} - W_{1I} W_{2I} K G u_{\Delta I}$$

$$(1 + KG) y_{\Delta I} = -K W_{1I} (W_{2O} u_{\Delta O} + W_{2I} G u_{\Delta I})$$

$$\Rightarrow \quad y_{\Delta I} = -K S W_{1I} (W_{2O} u_{\Delta O} + W_{2I} G u_{\Delta I})$$

$$\Rightarrow \quad y_{\Delta O} = W_{1O} G S (W_{2I} u_{\Delta I} - K W_{2O} u_{\Delta O})$$

$$
\underbrace{\begin{bmatrix} -KS\,W_{1I}W_{2I}\,G & -KS\,W_{1I}W_{2O} \\ W_{1O}W_{2I}\,GS & -W_{1O}W_{2O}\,T \end{bmatrix}}_{M} \begin{bmatrix} u_{\Delta I} \\ u_{\Delta O} \end{bmatrix}
$$

$$
= \begin{bmatrix} W_{1I} & 0 \\ 0 & W_{1O} \end{bmatrix} \begin{bmatrix} -T_I & -KS \\ SG & -T \end{bmatrix} \begin{bmatrix} W_{2I} & 0 \\ 0 & W_{2O} \end{bmatrix}
$$

(c) Full Block Uncertainty :

```
s = tf("s");
G = (1/(75*s+1)) * [-87.8 1.4; -108.2 -1.4];
WI = (s+0.2)/(0.5*s+1) * eye(2);
epsilon = 1e-6;
WP = (s/2+0.05)/(s+epsilon) * eye(2);
K = (0.7/s)*inv(G);
delta1 = ultidyn('delta1',[1,1]);
delta2 = ultidyn('delta2',[1,1]);
delta_diag = [delta1 0; 0 delta2];
delta_full = ultidyn('delta_full',[2,2]);
G_hat_full = G*(eye(2)+delta_full*WI); %unstructured uncertainty
G_hat_diag = G*(eye(2)+delta_diag*WI); %structured uncertainty
%need include uncertainty
S_full = inv(eye(2) + G_hat_full * K);
S_diag = inv(eye(2) + G_hat_diag * K);
% bodemag(Smu,1/Wp)

systemnames = 'G_hat_full WP'; %Block name only
inputvar = '[r{2};u{2}]';
outputvar = '[WP;r-G_hat_full]'; %Strangely, the system outputs are just the name
input_to_G_hat_full = '[u]';
input_to_WP = '[r-G_hat_full]';
cleanupsysic = 'yes';  %This drops all the useless variables from workspace
Pmu_full = sysic;
N_full = lft(Pmu_full, K);
perfmarg_full = robustperf(N_full);
mu_full = 1/perfmarg_full.LowerBound
```

```
mu_full = 4.1097
```

Diagonal Uncertainty:

```
systemnames = 'G_hat_diag WP'; %Block name only
inputvar = '[r{2};u{2}]';
outputvar = '[WP;r-G_hat_diag]'; %Strangely, the system outputs are just the name
input_to_G_hat_diag = '[u]';
input_to_WP = '[r-G_hat_diag]';
cleanupsysic = 'yes';  %This drops all the useless variables from workspace
Pmu_diag = sysic;
N_diag = lft(Pmu_diag, K);
perfmarg_diag = robustperf(N_diag);
mu_diag = 1/perfmarg_diag.LowerBound
```

```
mu_diag = 0.9669
```

3a.

Method 1: Bi-section

```
s = tf("s");
G = [1/(s+5) (s-2)/(s+10);10/(s^2+4*s+15) 3*s/(4*s+3)];
state_space = ss(G);
A = state_space.A;
B = state_space.B;
C = state_space.C;
D = state_space.D;
I = eye(size(D));
R = gam^2 * I - D'*D;
g_low = 0;
g_high = 1000;
g_try = g_high;
g_new = 1/2 * (g_high + g_low);
while(abs(g_new - g_try)>0.001)
    g_try = g_new;
    R = g_try^2 * eye(size(D)) - D'*D;
    H = [A+B*inv(R)*D'*C B*inv(R)*B';-C'*(I+D*inv(R)*D')*C -(A+B*inv(R)*D'*C)'];
    r = min(abs(real(eig(H))));
    if r>0.001
        g_high = g_try;
    else
        g_low = g_try;
    end
    g_new = 1/2 * (g_high + g_low);
end
```

Produce norm

```
norm_inf = g_new
```

```
norm_inf = 1.2503
```

```
norm_inf = norm(G,"inf")
```

```
norm_inf = 1.2500
```

Very similiar to norm function.

3b

```
G = [1/(s+5) 1/(s+10);10/(s^2+4*s+15) 3/(4*s+3)];
state_space = ss(G);
A = state_space.A;
B = state_space.B;
C = state_space.C;
Q = gram(state_space,'o');
norm_2 = sqrt(trace(B'*Q*B))
```

```
norm_2 = 1.1655
```

```
norm_2 = norm(G, 2)
```

```
norm_2 = 1.1655
```

Same as norm function