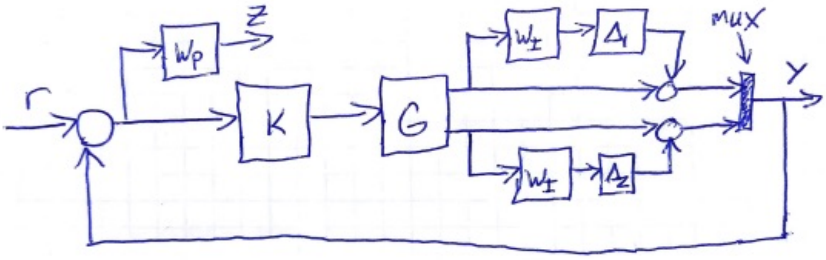
1: 20 points

Spinning Satellite Control

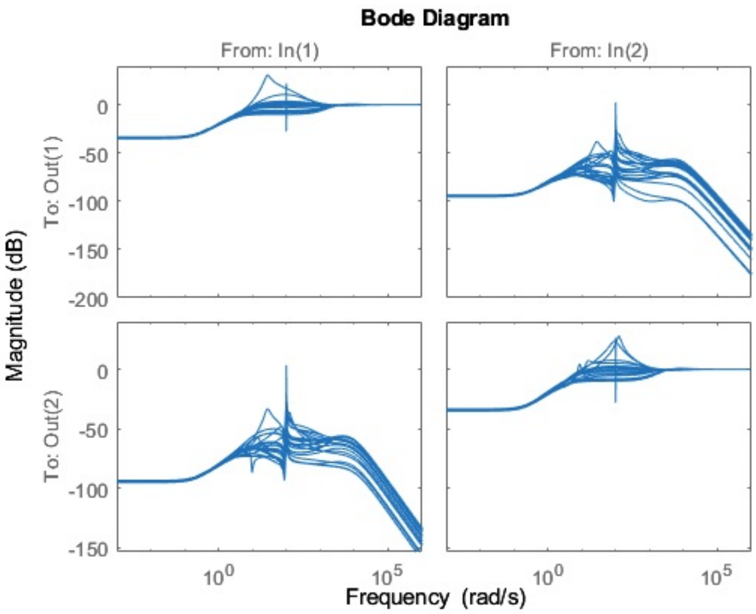
In a prior problem we considered a system with plant model

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

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.



(a) Assuming multiplicative output uncertainty of $W_I(s) = \frac{20s+10}{s+100}$ in the configuration shown and a performance weight of $W_P(s) = \frac{0.5s+20}{s+0.2}I_{2\times2}$, design a controller using the command *mixsyn*. Plot the achieved sensitivity function. Check whether you met your RP / RS goals.



Sensitivity Function

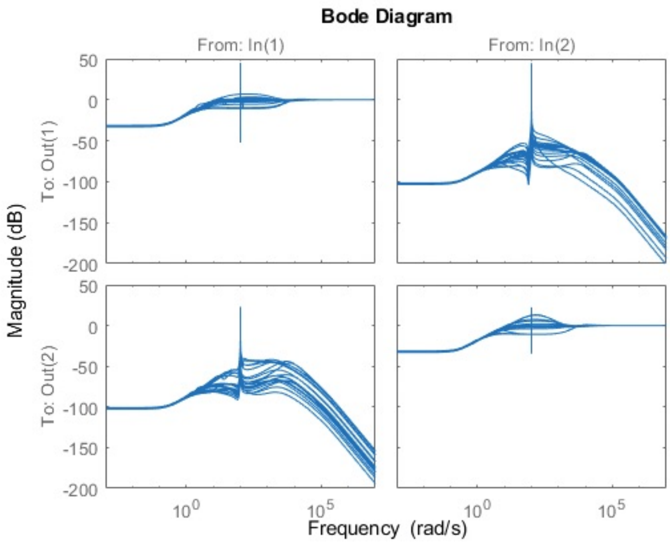
```
stable_test = logical
   1

mu = 1.8599

mu = 2.7252
```

all $\mu > 1$, doesnt meet RS and RP goals

```
%Problem 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];
W_i = (20*s+10)/(s+100) * eye(2);
W_p = (0.5*s+20)/(s+0.2) * eye(2);
[K, CL, GAM] = mixsyn(G, W_p, [], W_i);

del_1 = ultidyn('del_1',[1,1]);
del_2 = ultidyn('del_2',[1,1]);
W_i_unc = [del_1 0; 0 del_2] * W_i;
G_hat = (eye(2)+ W_i_unc) * G;

S = 1/(eye(2) + G_hat*K);
bodemag(S)
T = eye(2) - S;

stable_test = isstable(T.NominalValue)

[STABMARG,~] = robstab(S);
mu = 1/STABMARG.LowerBound
[perfmarg, ~] = robustperf(W_p*S);
mu = 1/perfmarg.LowerBound
```

(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.

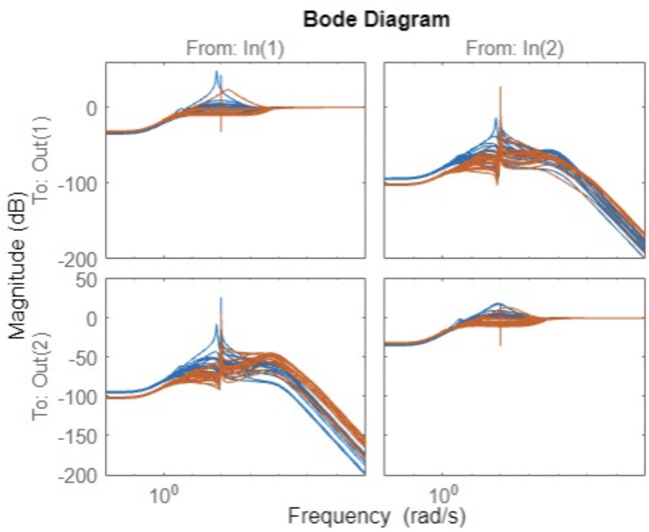
```
D-K ITERATION SUMMARY:
-----------------------------------------------------------------
                       Robust performance                Fit order
-----------------------------------------------------------------
  Iter          K Step       Peak MU       D Fit             D
    1           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
```



Sensitivity Function

```
stable_test = logical
   1
mu_RS = 2.0993
mu_RP = 2.5879
```

doesnt meet RS/RP as $\mu > 1$




```
%Problem 1b
systemnames = 'G_hat W_p';
inputvar = '[r{2};u{2}]';
outputvar = '[W_p;r-G_hat]';
input_to_G_hat = '[u]';
input_to_W_p = '[r-G_hat]';
cleanupsysic = 'yes';
P_mu = sysic;

[K_mu, ~, mu] = musyn(P_mu, 2, 2);

S_mu = eye(2) - feedback(G_hat*K_mu,eye(2));
bodemag(S_mu)
stable_test = isstable(eye(2) - S_mu.NominalValue)

[STABMARG,~] = robstab(S_mu);
mu_RS = 1/STABMARG.LowerBound
[perfmarg,~] = robustperf(W_p * S_mu);
mu_RP = 1/perfmarg.LowerBound

bodemag(S, S_mu)
```

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.

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

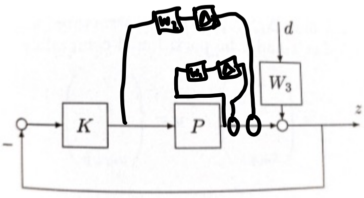
and $\|\Delta_i\|_\infty < 1$. Assume that W_1 and W_2 are stable and show the following.

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

$$\||W_1T| + |W_2KS|\|_\infty \leq 1.$$

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

$$\||W_3S| + |W_1T| + |W_2KS|\|_\infty \leq 1.$$



1.

$$G_p = \begin{bmatrix} z \\ v \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ -PW_1 & W_2 & -P \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ u \end{bmatrix}$$

$$1 - M_\Delta = \begin{vmatrix} 1 + W_1T\Delta_1 & W_1T\Delta_2 \\ W_2KS\Delta_1 & 1 + W_2KS\Delta_2 \end{vmatrix}$$

$$(1 + W_1T\Delta_1)(1 + W_2KS\Delta_2) - W_2KS\Delta_1(W_1T\Delta_2)$$

$$1 + W_2KS\Delta_2 + W_1T\Delta_1 = 0$$

assume $\Delta_2 = \Delta = \Delta_1$

$$1 + W_2KS\Delta + W_1T\Delta = 0$$

$$1 - |W_1T||\Delta| - |W_2KS||\Delta| = 0$$

$$\frac{1}{|\Delta|} = |W_1T| + |W_2KS|$$

$$\therefore \||W_1T| + |W_2KS|\|_\infty \leq 1$$

2.

$$\begin{bmatrix} y_{\Delta 1} \\ y_{\Delta 2} \\ x_{\Delta 3} \\ z \\ v \end{bmatrix} = \left[\begin{array}{ccc|c} 0 & 0 & 0 & W_1 \\ 0 & 0 & 0 & W_2 \\ G & 1 & W_3 & G \\ G & 1 & W_3 & G \\ \hline -G & -1 & -W_3 & -G \end{array}\right] \begin{bmatrix} u_{\Delta 1} \\ v_{\Delta 2} \\ u_{\Delta 3} \\ d \\ u \end{bmatrix}$$

$N = \text{lft}(P, K)$

$$= \begin{bmatrix} -W_1T & -W_1KS & W_1KSW_3 \\ -W_2T & -W_2KS & -W_2KSW_3 \\ GS & S & W_3S \\ GS & S & W_3S \end{bmatrix} \rightarrow \text{symbolab } \det(1 - M\Delta)$$

$$= 1 + W_1T\Delta_1 + W_2KS\Delta_2 - SW_3\Delta_3 = 0$$

all Δ are $|\Delta|$

$$-|W_1T| - |W_2KS| - |W_3S| = \frac{1}{|\Delta|}$$

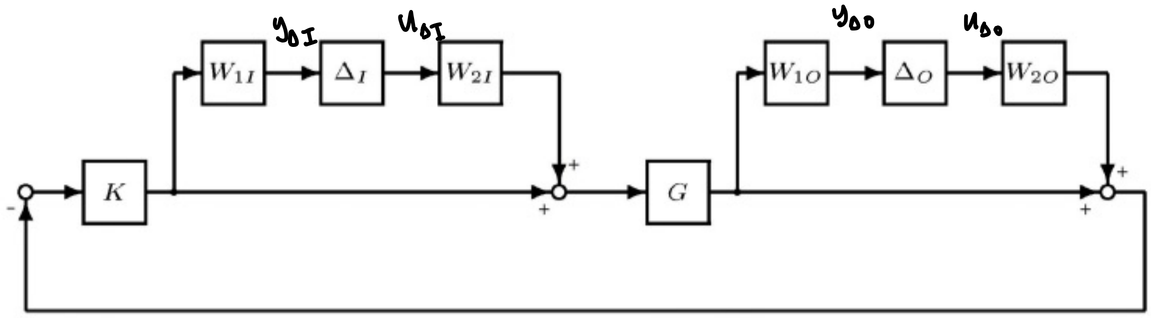
$$\therefore \||W_3S| + |W_1T| + |W_2KS|\|_\infty \leq 1$$

(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

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



$$\begin{bmatrix} z_I \\ z_O \\ v \end{bmatrix} = \begin{bmatrix} 0 & 0 & W_{1I} \\ W_{1O} G W_{2I} & 0 & W_{1O} G \\ -G W_{2I} & -W_{2O} & -G \end{bmatrix} \begin{bmatrix} w_I \\ w_O \\ u \end{bmatrix}$$

$$M = \begin{bmatrix} 0 & 0 \\ W_{1O} G W_{2I} & 0 \end{bmatrix} + \begin{bmatrix} W_{1I} \\ W_{1O} G \end{bmatrix} K(I+GK)^{-1} \begin{bmatrix} -GW_{2I} & -W_{2O} \end{bmatrix}$$

Symbolab

$$= \begin{bmatrix} -W_{1I} K(1+GK)^{-1} G W_{2I} & -W_{1I} K(1+GK)^{-1} W_{2O} \\ W_{1O} G W_{2I} - W_{1O} GK(1+GK)^{-1} G W_{2I} & -W_{1O} GK(1+GK)^{-1} W_{2O} \end{bmatrix}$$

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

(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.

```
%Problem 2c
clear;clc
s = tf('s');
G = 1/(75*s+1)*[-87.8 1.4;-108.2 -1.4];
K = 0.7/s/G;
w_I = (s+0.2)/(0.5*s+1);
W_i = [w_I 0; 0 w_I];
W_p = (s/2 + 0.05)/(s + 1e-3) * eye(2);

%Diagonal
del_1 = ultidyn('del_1',[1,1]);
del_2 = ultidyn('del_2',[1,1]);
d_diag = [del_1 0;0 del_2];
G_p_diag = G*(eye(2)+ d_diag * W_i);
S_diag = eye(2) - feedback(G_p_diag*K,eye(2));
[perfmarg_diag,~] = robustperf(W_p * S_diag);
mu_diag = 1/perfmarg_diag.LowerBound

%Full-Block
del_1 = ultidyn('del_1',[1,1]);
del_2 = ultidyn('del_2',[1,1]);
del_3 = ultidyn('del_3',[1,1]);
del_4 = ultidyn('del_4',[1,1]);
d_full = [del_1 del_2; del_3 del_4];
G_p_full = G * (eye(2)+ d_full * W_i);
S_full = eye(2)-feedback(G_p_full * K,eye(2));
[perfmarg_full,~] = robustperf(W_p * S_full);
mu_full = 1/perfmarg_full.LowerBound
```

```
mu_diag = 0.9666
```

RP because $\mu < 1$

```
mu_full = 4.6265
```

not RP as $\mu > 1$

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

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

```
%Problem 3a
clear;clc
s = tf('s');
G = [1/(s+5) (s-2)/(s+10); 10/(s^2+4*s+15) 3*s/(4*s+3)];
inf_norm_val = infinity_norm(G)
norm(G, 'inf')
```

```
inf_norm_val = 1.2503
ans = 1.2500
```

mine

matlab

```
function norm_inf_G = infinity_norm(G)
    G_ss = ss(G);
    A = G_ss.A;
    B = G_ss.B;
    C = G_ss.C;
    D = G_ss.D;

    %Bisection
    gam_low = 0;
    gam_high = 1000;
    test_val = gam_high;
    split_pt = 1/2 * (gam_high + gam_low);

    while(abs(split_pt - test_val) > 0.001)
        test_val = split_pt;
        R = test_val^2 * eye(size(D)) - D'*D;

        H = [A+B*inv(R)*D'*C B*inv(R)*B'; -C'*(eye(size(D))+D*inv(R)*D')*C -(A+B*inv(R)*D'*C)'];
        eig_check = min(abs(real(eig(H))));

        if eig_check > 0.001
            gam_high = test_val;
        else
            gam_low = test_val;
        end

        split_pt = 1/2 * (gam_high + gam_low);
    end
    norm_inf_G = split_pt;
end
```

(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.

```
%Problem 3b
G = [1/(s+5) 1/(s+10); 10/(s^2+4*s+15) 3/(4*s+3)];
two_norm_val = two_norm(G)
norm(G, 2)
```

```
two_norm_val = 1.1655
ans = 1.1655
```

mine

matlab

```
function norm = two_norm(G)
    G_ss = ss(G);
    norm = sqrt(trace(G_ss.B'*gram(G_ss,"o")*G_ss.B));
end
```