ME-8930

Convex Optimization Methods for Robust and Optimal Control Design

HW05

Group:

Chinmay Samak

Tanmay Samak

$$m_1 \ddot{q}_1 + k_1 q_1 - k_1 q_2 = f(t) + w(t)$$
$$m_2 \ddot{q}_2 - k_1 q_1 + (k_1+k_2) q_2 - k_2 q_3 = 0$$
$$m_3 \ddot{q}_3 - k_2 q_2 + k_2 q_3 = 0$$

$$x_p = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \qquad y = q_3$$

$$\dot{x}_p = A_p x_p + B_p u + D_p w$$
$$y = C_p x_p$$

$z = x_p$ (full state feedback)

$u = G x_p$ (Static state feedback control)

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -k_1/m_1 & k_1/m_1 & 0 & 0 & 0 & 0 \\ k_1/m_2 & \frac{(-k_1-k_2)}{m_2} & k_2/m_2 & 0 & 0 & 0 \\ 0 & k_2/m_3 & -k_2/m_3 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/m_1 \\ 0 \\ 0 \end{bmatrix} f(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/m_1 \\ 0 \\ 0 \end{bmatrix} w(t)$$

$$q_3 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

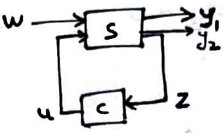
$$A_p = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{-k_1}{m_1} & \frac{k_1}{m_1} & 0 & 0 & 0 & 0 \\ \frac{k_1}{m_2} & \frac{(-k_1-k_2)}{m_2} & \frac{k_2}{m_2} & 0 & 0 & 0 \\ 0 & \frac{k_2}{m_3} & \frac{-k_2}{m_3} & 0 & 0 & 0 \end{bmatrix} \quad B_p = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{m_1} \\ 0 \\ 0 \end{bmatrix} \quad D_p = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{m_1} \\ 0 \\ 0 \end{bmatrix}$$

$$C_p = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

General form:

$$\dot{x}_p = A_p x_p + B_p u + D_p w$$

$$y_1 = C_{p_1} x_p + B_{y_1} u + D_y w$$
$$y_2 = C_{p_2} x_p + B_{y_2} u$$
$$z = M_p x_p + D_z w$$
$$z = x_p \text{ (full state feedback)}$$



Comparing with earlier notation $\begin{cases} \dot{x}_p = A_p x_p + B_p u + D_p w \\ y = C_p x_p \end{cases}$

general form / earlier notation

$A_p = A_p$
$B_p = B_p$
$D_p = D_p$
$C_{p_2} = C_p$
$B_{y_2} = 0$
$D_y = 0$

we only consider Γ_{ep} from w to y_2 to design an optimal H_2 controller so as to minimize Γ_{ep} and ensure CL system stability by placing the CL poles in LHP such that $Re(z) < -1$.

desired region for CL-Poles used in "lmireg".

Lumped system representation

$$\left.\begin{aligned} A &= A_p \\ B &= [D_p \;\; B_p] \\ C &= [C_{p_1}, C_{p_2}]^T \\ D &= \begin{bmatrix} D_y & B_{y_1} \\ 0 & B_{y_2} \end{bmatrix} \end{aligned}\right\} S$$

$r = \text{size}(B_{y_2}) = [\text{size}(y_2) \;\; \text{size}(u)]$

$$\text{obj} = \begin{array}{cccc} [0 & 0 & 0 & 1] \\ \gamma_1 & \gamma_2 & \alpha & \beta \end{array} \Rightarrow \min \Gamma_{ep} \Rightarrow \text{optimal } H_2 \text{ control design}$$

$\min (\alpha \Gamma_{ee}^2 + \beta \Gamma_{ep}^2)$
s.t. $\Gamma_{ee} < \gamma_1$
$\Gamma_{ep} < \gamma_2$

$$[\gamma_1^*, \gamma_2^*, G_1, S_{CL}, X] = \text{msfsyn}(S, r, \text{obj}, \text{region})$$

γ_1^*: Γ_{ee} of CL system
γ_2^*: Γ_{ep} of CL system
G_1: static state feedback control gain ($u = G_1 x_p$)
S_{CL}: CL system representation
X: Lyapunov parameter matrix

$A_{CL} = A_p + B_p \cdot G_1$ ... $\begin{cases} u = kx \\ \Rightarrow A_{CL} = A + Bk \end{cases}$

Design validation:

- performance

→ h2_norm = norm2 (S_{CL})
(Γ_{ep})

↳ $\Gamma_{ep} = \|CPC^T\|^{1/2} = (tr(CPC^T))^{1/2}$

where $AP + PA^T + BB^T = 0$ (Lyapunov equation)
$P > 0$

$\Gamma_{ep} < \gamma_2^*$

- stability

→ CL poles (z) : $Re(\lambda_i(A_{CL})) = Re(z)$ < 0 ← stability (CL) guaranteed
$Re(z) < -1$ ← pole placement (CL) in desired region.

System simulation:

$$w(t) = \begin{cases} 100 & 0 \leq t \leq 2 \\ 0 & t > 2 \end{cases}$$

$u(t) = 0 \quad \forall t$

simulate open (S_{OL}) and closed (S_{CL}) loop systems using "lsim" for t = 0 to t = 10 seconds with $\Delta t = 0.01$ sec.

Uncertain system:

10% uncertainty in k_2 (nominal k_2 = 4000 N/m)

i.e. $3600 \leq k_2 \leq 4400$ N/m

for k2 = 3600:1:4400
h2_norm = norm2 (Scl) ← using G_1

plot (k2, h2_norm) ← Γ_{ep} vs. k_2 with G_1

Robust state-feedback controller design:

Affine system representation

$A = A_0 + k_2 A_1$

$$A_0 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{-k_1}{m_1} & \frac{k_1}{m_1} & 0 & 0 & 0 & 0 \\ \frac{k_1}{m_2} & \frac{-k_1}{m_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad A_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{-1}{m_2} & \frac{1}{m_2} & 0 & 0 & 0 \\ 0 & \frac{1}{m_3} & \frac{-1}{m_3} & 0 & 0 & 0 \end{bmatrix}$$

$B_0 = B$, $C_0 = C$, $D_0 = D$ | $P = [k_2]$; $k_2 \in [3600, 4400]$
= pvec ('box', [3600,4400])

So = ltisys (A_0 , B_0 , C_0 , D_0 , 1)

S_1 = ltisys (A_1 , 0, 0, 0 , 0)

affsys = psys (P, [So , S_1])

r = size (B_{y_2}) = [size (y_2) size (u)]

obj = [0 0 0 1] ← optimal H_2 control design (min Γ_{ep})

region = lmireg ← LHP with Re(z) < -1

[γ_1^*, γ_2^*, G_2, S_{cl}, X] = msfsyn (affsys, r, obj, region)

Γ_{ep} of CL system (guaranteed H_2 performance)

Robust static state feedback control gain ($u = G_2 x_p$)

CL system representation obtained

$A_{cl} = A_p + B_p G_2$

($A_{cl} = A + BK$, $u = kx$)

for k_2 = 3600 : 1 : 4400

h2_norm = norm2 (S_{cl}) ← using G_2

plot (k_2, h2_norm) ← Γ_{ep} vs. k_2 with G_2

Problem 1

CODE:

```
% PROBLEM 1


close all
clear
clc


% Define the system parameters
m1 = 20; % kg
m2 = 40; % kg
m3 = 30; % kg
k1 = 2000; % N/m
k2 = 4000; % N/m


% Define the system matrices
Ap = [0 0 0 1 0 0;
      0 0 0 0 1 0;
      0 0 0 0 0 1;
      -k1/m1 k1/m1 0 0 0 0;
      k1/m2 (-k1-k2)/m2 k2/m2 0 0 0;
      0 k2/m3 -k2/m3 0 0 0]
Bp = [0; 0; 0; 1/m1; 0; 0]
Dp = [0; 0; 0; 1/m1; 0; 0]
Cp2 = [0 0 1 0 0 0]
By2 = 0
Dy = 0
```

```
% Lumped system representation
A = Ap;
B = [Dp Bp];
C = [zeros(size(Cp2)); Cp2];
D = [Dy zeros(size(By2)); zeros(size(Dy)) By2];
S = ltisys(A, B, C, D)


% State-feedback controller G1 to minimize the energy-to-peak gain Γep
r = size(By2);
obj = [0 0 0 1]; % Optimal H2 control design objective
region = lmireg; % h (half-plane) --> l (LHP, i.e., Re(z) < x0) --> -1 (x0 = -1)
[g1opt, g2opt, G1, Scl, X] = msfsyn(S, r, obj, region)
```

```
% Closed-loop system
Acl = Ap + Bp*G1
h2_norm = norm2(Scl)
% Scl = ss(Acl, [Dp zeros(size(Bp))], C, D)
% h2_norm = norm(Scl, 2)


% Verification
if((h2_norm < g2opt) && all(real(eig(Acl)) < -1.0))
    disp('Verification of H2 norm and pole location constraints successful!')
else
    disp('Verification of H2 norm and pole location constraints failed!')
end
```

```
% Define the lumped (disturbance + control) input
t = 0:0.01:10;
w_amplitude = 100.0;
w_duration = 2;
w_pulse = w_amplitude * (t >= 0 & t <= w_duration);
u = zeros(size(w_pulse));
w = [w_pulse; u];


% Simulate the open-loop system response
Sol = ss(A, B, C, D)
[y_ol, t_out, x_ol] = lsim(Sol, w', t);


% Plot the results
figure;
sgtitle('Open-Loop System Response');
subplot(3, 1, 1);
plot(t, w(1, :), t, w(2, :));
legend('W', 'U');
subplot(3, 1, 2);
plot(t, x_ol(:, 1), ...
     t, x_ol(:, 2), ...
     t, x_ol(:, 3), ...
     t, x_ol(:, 4), ...
     t, x_ol(:, 5), ...
     t, x_ol(:, 6) ...
     );
legend('X1', 'X2', 'X3', 'X4', 'X5', 'X6');
subplot(3, 1, 3);
plot(t, y_ol(:, 2));
```

```
legend('Y1');


% Simulate the closed-loop system response
Scl = ss(Acl, [Dp zeros(size(Bp))], C, D)
[y_cl, t_out, x_cl] = lsim(Scl, w', t);


% Plot the results
figure;
sgtitle('Closed-Loop System Response');
subplot(3, 1, 1);
plot(t, w(1, :), t, w(2, :));
legend('W', 'U');
subplot(3, 1, 2);
plot(t, x_cl(:, 1), ...
     t, x_cl(:, 2), ...
     t, x_cl(:, 3), ...
     t, x_cl(:, 4), ...
     t, x_cl(:, 5), ...
     t, x_cl(:, 6) ...
     );
legend('X1', 'X2', 'X3', 'X4', 'X5', 'X6');
subplot(3, 1, 3);
plot(t, y_cl(:, 2));
legend('Y1');
```

```
% Define uncertain system parameters
m1 = 20; % kg
m2 = 40; % kg
m3 = 30; % kg
k1 = 2000; % N/m
i = 1;
for k2 = 3600:1:4400 % N/m
    % Define the system matrices
    Ap = [0 0 0 1 0 0;
          0 0 0 0 1 0;
          0 0 0 0 0 1;
          -k1/m1 k1/m1 0 0 0 0;
          k1/m2 (-k1-k2)/m2 k2/m2 0 0 0;
          0 k2/m3 -k2/m3 0 0 0];
    Bp = [0; 0; 0; 1/m1; 0; 0];
    Dp = [0; 0; 0; 1/m1; 0; 0];
    Cp2 = [0 0 1 0 0 0];
    By2 = 0;
    Dy = 0;
```

```
    % Lumped system representation
    A = Ap;
    B = [Dp Bp];
    C = [zeros(size(Cp2)); Cp2];
    D = [Dy zeros(size(By2)); zeros(size(Dy)) By2];
    % Closed-loop system
    Acl = Ap + Bp*G1;
    Scl = ltisys(Acl, [Dp zeros(size(Bp))], C, D);
    h2_norm(i) = norm2(Scl);
    % Scl = ss(Acl, [Dp zeros(size(Bp))], C, D);
    % h2_norm(i) = norm(Scl, 2);
    i = i+1;
end


figure;
sgtitle('Uncertain System with Controller G_1');
plot(3600:1:4400, h2_norm)
xlabel('Spring Constant k_2')
ylabel('Energy to Peak Gain Γ_ep')
```

```
% Define the system parameters
m1 = 20; % kg
m2 = 40; % kg
m3 = 30; % kg
k1 = 2000; % N/m


% System matrices in affine form
A0 = [0 0 0 1 0 0;
      0 0 0 0 1 0;
      0 0 0 0 0 1;
      -k1/m1 k1/m1 0 0 0 0;
      k1/m2 -k1/m2 0 0 0 0;
      0 0 0 0 0 0]
A1 = [0 0 0 0 0 0;
      0 0 0 0 0 0;
      0 0 0 0 0 0;
      0 0 0 0 0 0;
      0 -1/m2 1/m2 0 0 0;
      0 1/m3 -1/m3 0 0 0]
B0 = B
C0 = C
D0 = D


% Uncertain LTI system in affine form
```

```
S0 = ltisys(A0, B0, C0, D0, 1)
S1 = ltisys(A1, zeros(size(B0)), zeros(size(C0)), zeros(size(D0)), 0)


% Parameter vector
P = pvec('box', [3600, 4400]) % k2


% Affine system
affsys = psys(P, [S0, S1])


% State-feedback controller G2 to minimize the energy-to-peak gain Γep
r = size(By2);
obj = [0 0 0 1]; % Optimal H2 control design objective
region = lmireg; % h (half-plane) --> l (LHP, i.e., Re(z) < x0) --> -1 (x0 = -1)
[g1opt, g2opt, G2, Scl, X] = msfsyn(affsys, r, obj, region)


disp('Optimal guaranteed energy-to-peak gain for the uncertain system is:')
disp(g2opt)
```

```
% Define uncertain system parameters
m1 = 20; % kg
m2 = 40; % kg
m3 = 30; % kg
k1 = 2000; % N/m
i = 1;
for k2 = 3600:1:4400 % N/m
    % Define the system matrices
    Ap = [0 0 0 1 0 0;
          0 0 0 0 1 0;
          0 0 0 0 0 1;
          -k1/m1 k1/m1 0 0 0 0;
          k1/m2 (-k1-k2)/m2 k2/m2 0 0 0;
          0 k2/m3 -k2/m3 0 0 0];
    Bp = [0; 0; 0; 1/m1; 0; 0];
    Dp = [0; 0; 0; 1/m1; 0; 0];
    Cp2 = [0 0 1 0 0 0];
    By2 = 0;
    Dy = 0;
    % Lumped system representation
    A = Ap;
    B = [Dp Bp];
    C = [zeros(size(Cp2)); Cp2];
    D = [Dy zeros(size(By2)); zeros(size(Dy)) By2];
    % Closed-loop system
```

```
    Acl = Ap + Bp*G2;
    Scl = ltisys(Acl, [Dp zeros(size(Bp))], C, D);
    h2_norm(i) = norm2(Scl);
    % Scl = ss(Acl, [Dp zeros(size(Bp))], C, D);
    % h2_norm(i) = norm(Scl, 2);
    i = i+1;
end


figure;
sgtitle('Uncertain System with Controller G_2');
plot(3600:1:4400, h2_norm)
xlabel('Spring Constant k_2')
ylabel('Energy to Peak Gain Γ_ep')
```

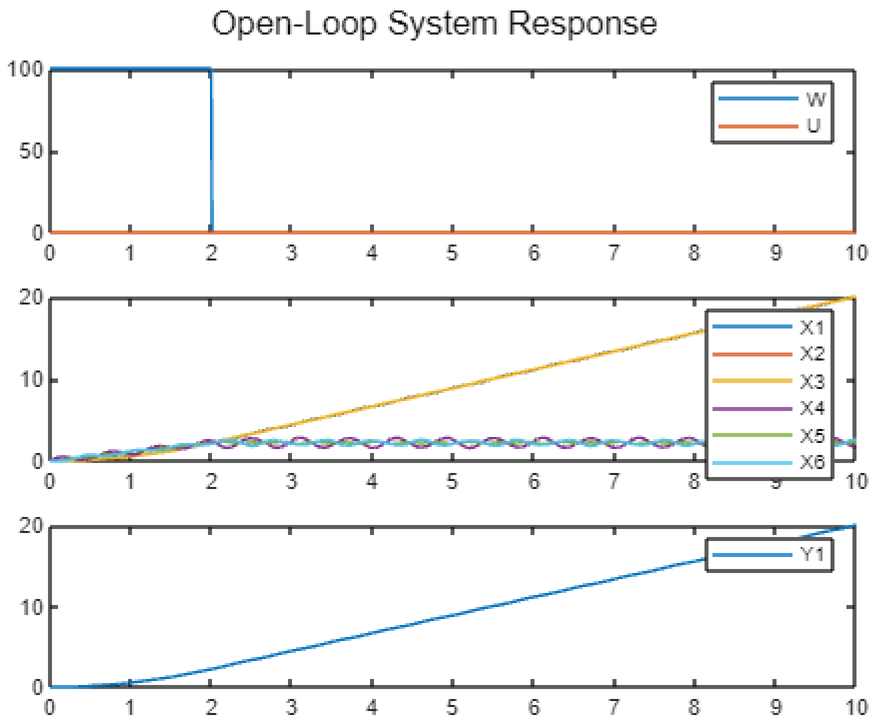
OUTPUT:

```
Ap = 6×6
         0         0         0    1.0000         0         0
         0         0         0         0    1.0000         0
         0         0         0         0         0    1.0000
 -100.0000  100.0000         0         0         0         0
   50.0000 -150.0000  100.0000         0         0         0
         0  133.3333 -133.3333         0         0         0
Bp = 6×1
         0
         0
         0
    0.0500
         0
         0
Dp = 6×1
         0
         0
         0
    0.0500
         0
         0
Cp2 = 1×6
     0     0     1     0     0     0
By2 = 0
Dy = 0
S = 9×9
         0         0         0    1.0000         0         0         0
0    6.0000
         0         0         0         0    1.0000         0         0
0         0
         0         0         0         0         0    1.0000         0
0         0
 -100.0000  100.0000         0         0         0         0    0.0500
0.0500         0
   50.0000 -150.0000  100.0000         0         0         0         0
0         0
```

```
         0  133.3333 -133.3333         0         0         0         0
0         0
         0         0         0         0         0         0         0
0         0
         0         0    1.0000         0         0         0         0
0         0
         0         0         0         0         0         0         0
0      -Inf
Select a region among the following:

h)   Half-plane
d)   Disk
c)   Conic sector
e)   Ellipsoid
p)   Parabola
s)   Horizontal strip
m)   Matrix description of the LMI region
q)   Quit


Select a region among the following:

h)   Half-plane
d)   Disk
c)   Conic sector
e)   Ellipsoid
p)   Parabola
s)   Horizontal strip
m)   Matrix description of the LMI region
q)   Quit
 Optimization of   0.000 * G^2 +  1.000 * H^2 :
 ----------------------------------------------

 Solver for linear objective minimization under LMI constraints

 Iterations   :    Best objective value so far

     1
     2
     3
     4
     5
     6
     7
     8                   0.123781
     9                   0.101980
    10                   0.101980
    11                   0.028857
    12                   0.028857
    13                   0.028857
    14                   0.012000
    15                   0.012000
    16                   0.010372
    17                   0.010372
    18                   0.010372
    19               6.093835e-03
    20               6.093835e-03
```

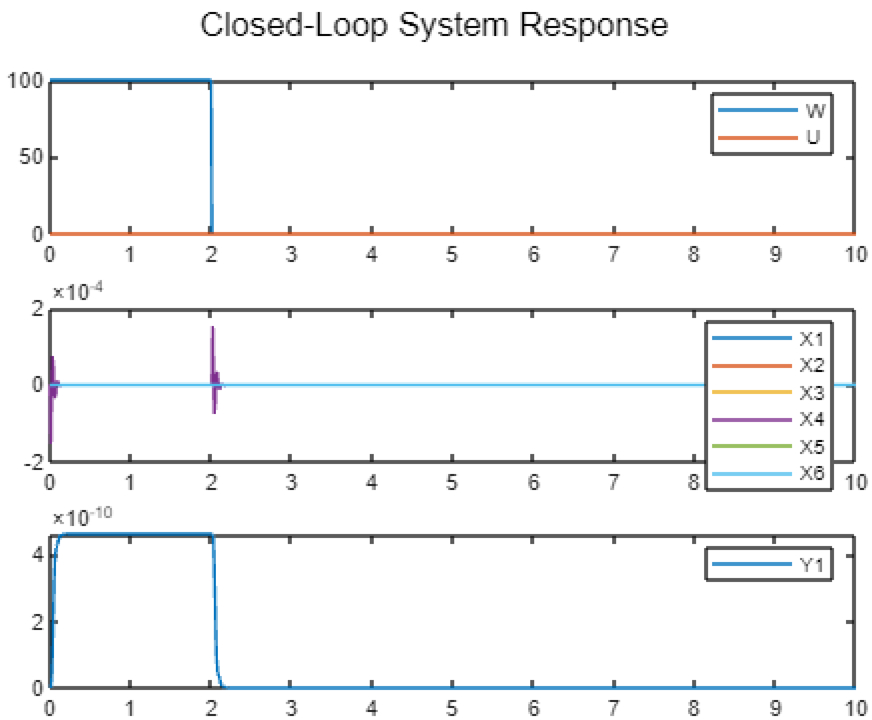
```
    21                6.093835e-03
    22                3.043563e-03
    23                3.043563e-03
    24                3.043563e-03
    25                1.240950e-03
    26                1.240950e-03
    27                1.231455e-03
    28                3.550594e-04
    29                2.305720e-04
    30                2.305720e-04
    31                1.078822e-04
    32                1.078822e-04
    33                1.078822e-04
    34                5.183917e-05
    35                5.183917e-05
    36                5.183917e-05
    37                2.614067e-05
    38                2.614067e-05
    39                2.614067e-05
    40                1.606369e-05
    41                1.606369e-05
    42                1.596526e-05
    43                3.082070e-06
    44                3.082070e-06
    45                2.599298e-06
    46                2.599298e-06
    47                2.599298e-06
    48                9.165308e-07

 Result:  reached the target for the objective value
          best objective value: 9.165308e-07
          f-radius saturation:  0.520% of R =  1.00e+10

 Guaranteed   H2 performance: 9.57e-04

g1opt =

     []
g2opt = 9.5736e-04
G1 = 1×6
10^11 ×
   -0.0008   -0.2312   -1.9312   -0.0000   -0.0024   -0.0996
Scl = 9×8
10^9 ×
         0         0         0    0.0000         0         0         0
0.0000
         0         0         0         0    0.0000         0         0
0
         0         0         0         0         0    0.0000         0
0
   -0.0038   -1.1562   -9.6558   -0.0000   -0.0120   -0.4980    0.0000
0
    0.0000   -0.0000    0.0000         0         0         0         0
0
         0    0.0000   -0.0000         0         0         0         0
0
```

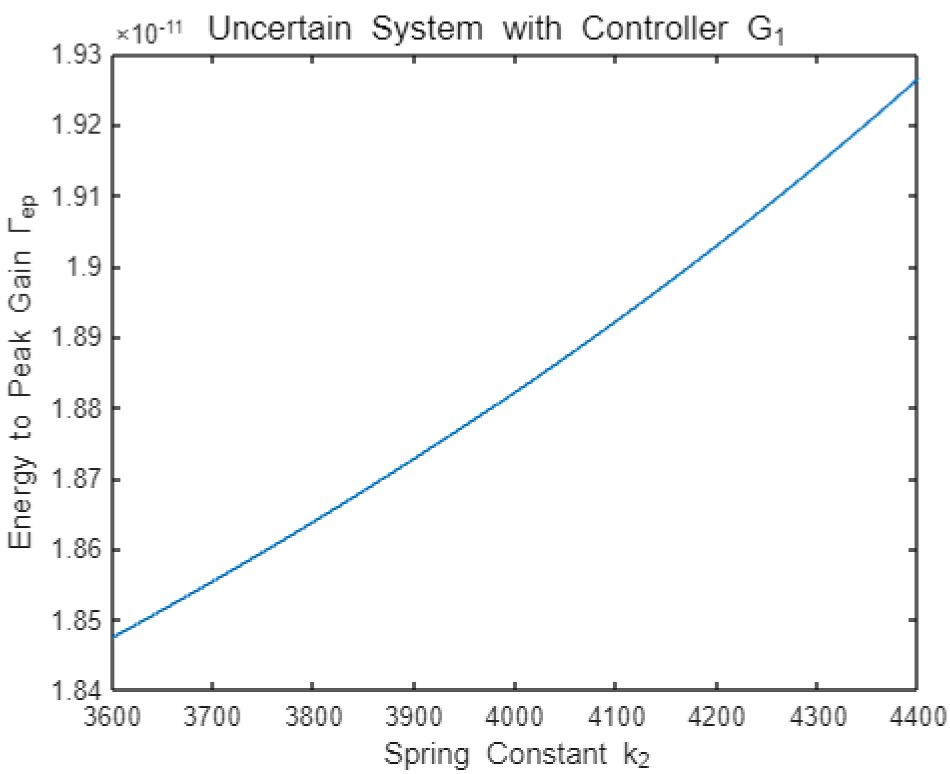
```
         0         0         0         0         0         0         0
0
         0         0    0.0000         0         0         0         0
0
         0         0         0         0         0         0         0      -
Inf
X = 6×6
10⁶ ×
    0.0003    0.0000   -0.0000   -0.0080   -0.0002   -0.0000
    0.0000    0.0000    0.0000    0.0000   -0.0000   -0.0000
   -0.0000    0.0000    0.0000   -0.0000    0.0000   -0.0000
   -0.0080    0.0000   -0.0000    1.3121   -0.0003   -0.0000
   -0.0002   -0.0000    0.0000   -0.0003    0.0002    0.0000
   -0.0000   -0.0000   -0.0000   -0.0000    0.0000    0.0000
Acl = 6×6
10⁹ ×
         0         0         0    0.0000         0         0
         0         0         0         0    0.0000         0
         0         0         0         0         0    0.0000
   -0.0038   -1.1562   -9.6558   -0.0000   -0.0120   -0.4980
    0.0000   -0.0000    0.0000         0         0         0
         0    0.0000   -0.0000         0         0         0
h2_norm = 1.8822e-11
Verification of H2 norm and pole location constraints successful!
Sol =

  A = 
           x1      x2      x3      x4      x5      x6
   x1       0       0       0       1       0       0
   x2       0       0       0       0       1       0
   x3       0       0       0       0       0       1
   x4    -100     100       0       0       0       0
   x5      50    -150     100       0       0       0
   x6       0   133.3  -133.3       0       0       0

  B = 
         u1    u2
   x1     0     0
   x2     0     0
   x3     0     0
   x4  0.05  0.05
   x5     0     0
   x6     0     0

  C = 
       x1  x2  x3  x4  x5  x6
   y1   0   0   0   0   0   0
   y2   0   0   1   0   0   0

  D = 
       u1  u2
   y1   0   0
   y2   0   0

Continuous-time state-space model.
```




```
Scl =

  A = 
                 x1          x2          x3          x4          x5          x6
   x1             0           0           0           1           0           0
   x2             0           0           0           0           1           0
   x3             0           0           0           0           0           1
   x4    -3.805e+06  -1.156e+09  -9.656e+09   -1.15e+04  -1.205e+07   -4.98e+08
   x5            50        -150         100           0           0           0
   x6             0       133.3      -133.3           0           0           0

  B = 
           u1    u2
   x1       0     0
   x2       0     0
   x3       0     0
   x4    0.05     0
   x5       0     0
   x6       0     0

  C = 
       x1  x2  x3  x4  x5  x6
   y1   0   0   0   0   0   0
   y2   0   0   1   0   0   0

  D = 
       u1  u2
   y1   0   0
   y2   0   0

Continuous-time state-space model.
```

[Model Properties]






```
A0 = 6×6
        0         0         0         1         0         0
        0         0         0         0         1         0
        0         0         0         0         0         1
```

```
  -100   100     0     0     0     0
    50   -50     0     0     0     0
     0     0     0     0     0     0
A1 = 6×6
         0         0         0         0         0         0
         0         0         0         0         0         0
         0         0         0         0         0         0
         0         0         0         0         0         0
         0   -0.0250    0.0250         0         0         0
         0    0.0333   -0.0333         0         0         0
B0 = 6×2
         0         0
         0         0
         0         0
    0.0500    0.0500
         0         0
         0         0
C0 = 2×6
     0     0     0     0     0     0
     0     0     1     0     0     0
D0 = 2×2
     0     0
     0     0
S0 = 9×9
         0         0         0    1.0000         0         0         0
0    6.0000
         0         0         0         0    1.0000         0         0
0         0
         0         0         0         0         0    1.0000         0
0         0
 -100.0000  100.0000         0         0         0         0    0.0500
0.0500         0
   50.0000  -50.0000         0         0         0         0         0
0         0
         0         0         0         0         0         0         0
0         0
         0         0         0         0         0         0         0
0         0
         0         0    1.0000         0         0         0         0
0         0
         0         0         0         0         0         0         0
0      -Inf
S1 = 9×9 complex
   0.0000 - 1.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
6.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 - 1.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 - 1.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 - 1.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i
```

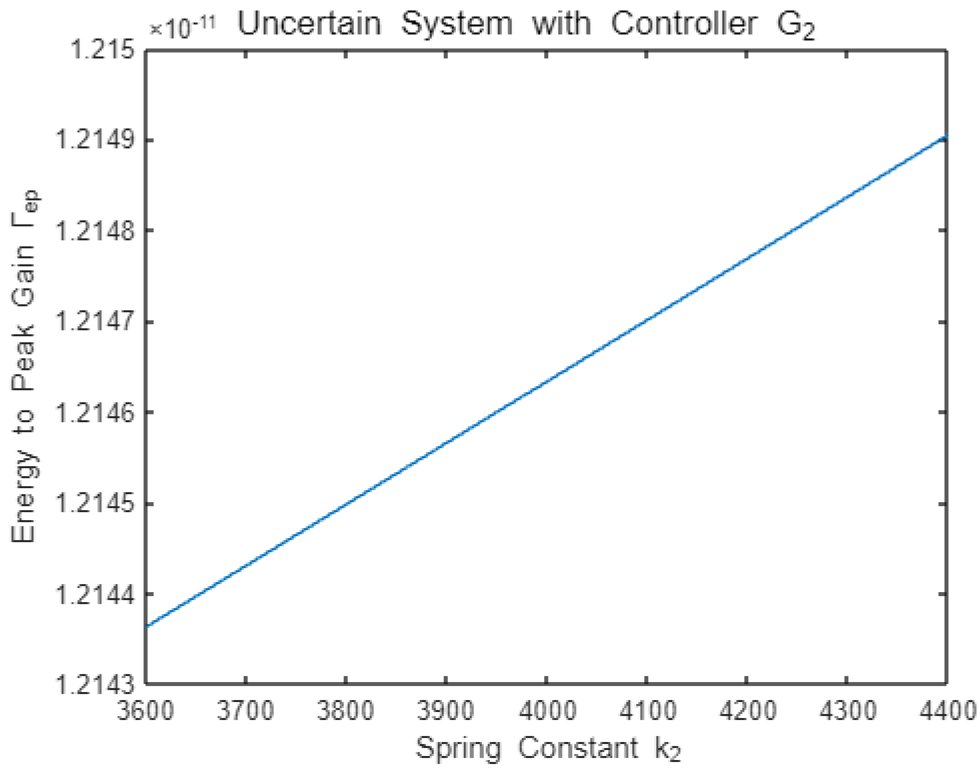
```
   0.0000 + 0.0000i  -0.0250 + 0.0000i   0.0250 + 0.0000i   0.0000 + 0.0000i
0.0000 - 1.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0333 + 0.0000i  -0.0333 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 - 1.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
-Inf + 0.0000i
P = 2×5
           1        3600        4400           0           0
           1           0           0           0           0
affsys = 9×26 complex
10³ ×
     -Inf + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0010 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0060 + 0.0000i   0.0000 + 0.0000i
0.0000 - 0.0010i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0060 + 0.0000i   0.0010 + 0.0000i   3.6000 + 0.0000i   4.4000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0020 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0010 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 - 0.0010i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0010 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0020 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0010 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 - 0.0010i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0060 + 0.0000i   0.0000 + 0.0000i  -0.1000 + 0.0000i   0.1000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0001 + 0.0000i   0.0001 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 - 0.0010i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0020 + 0.0000i   0.0000 + 0.0000i   0.0500 + 0.0000i  -0.0500 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i  -0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 - 0.0010i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0020 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
```

```
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i  -0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 - 0.0010i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0100 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0010 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i     -Inf + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
-Inf + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i
Select a region among the following:

h)   Half-plane
d)   Disk
c)   Conic sector
e)   Ellipsoid
p)   Parabola
s)   Horizontal strip
m)   Matrix description of the LMI region
q)   Quit


Select a region among the following:

h)   Half-plane
d)   Disk
c)   Conic sector
e)   Ellipsoid
p)   Parabola
s)   Horizontal strip
m)   Matrix description of the LMI region
q)   Quit
 Optimization of   0.000 * G^2 +  1.000 * H^2 :
 ----------------------------------------------

 Solver for linear objective minimization under LMI constraints

 Iterations   :    Best objective value so far

     1
     2
```

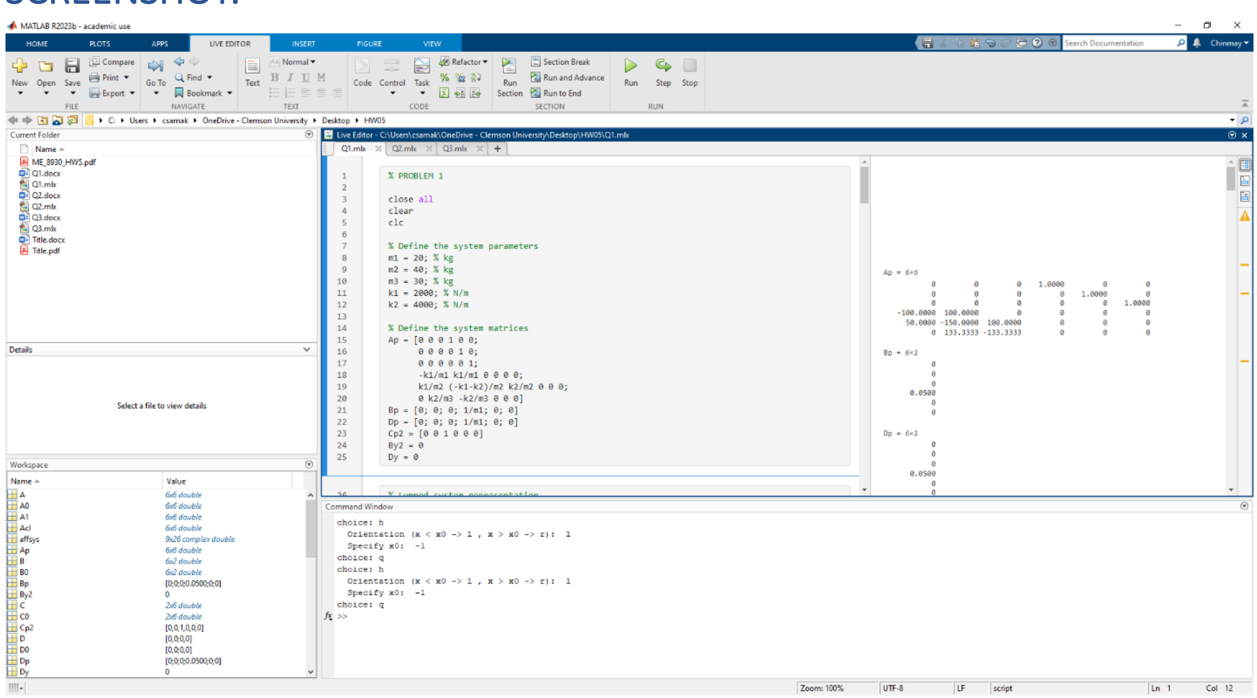
3	
4	
5	
6	
7	
8	
9	
10	
11	0.174617
12	0.084142
13	0.084142
14	0.084142
15	0.035606
16	0.035606
17	0.035606
18	0.019613
19	0.019613
20	0.019613
21	0.011597
22	0.011597
23	0.011597
24	7.440620e-03
25	7.440620e-03
26	7.440620e-03
27	4.761071e-03
28	4.761071e-03
29	4.761071e-03
30	3.057379e-03
31	3.057379e-03
32	3.057379e-03
33	2.094639e-03
34	2.094639e-03
35	2.094639e-03
36	1.728979e-03
37	1.728979e-03
38	1.728979e-03
39	1.401557e-03
40	1.401557e-03
41	1.401557e-03
42	7.898234e-04
43	7.898234e-04
44	7.898234e-04
45	3.528895e-04
46	3.528895e-04
47	3.528895e-04
48	1.357855e-04
49	1.357855e-04
50	1.357855e-04
51	5.390543e-05
52	5.390543e-05
53	5.390543e-05
54	2.140225e-05
55	2.140225e-05
56	2.140225e-05
57	1.012459e-05
58	1.012459e-05
59	1.012459e-05

```
    60                5.006723e-06
    61                5.006723e-06
    62                5.006723e-06
    63                2.216772e-06
    64                2.216772e-06
    65                2.216772e-06
    66                9.611413e-07

 Result:  reached the target for the objective value
          best objective value: 9.611413e-07
          f-radius saturation:  0.243% of R =  1.00e+10

 Guaranteed   H2 performance: 9.80e-04

g1opt =

     []
g2opt = 9.8038e-04
G2 = 1×6
10^11 ×
   -0.0009   -0.2480   -1.9988   -0.0000   -0.0028   -0.0997
Scl = 9×19
10^9 ×
      -Inf         0         0         0         0    0.0000         0
0         0    0.0000         0         0         0         0    0.0000
0         0         0    0.0000
    0.0000         0         0         0         0         0    0.0000
0         0         0         0         0         0         0         0
0.0000         0         0         0
    0.0000         0         0         0         0         0         0
0.0000         0         0         0         0         0         0         0
0    0.0000         0         0
    0.0000         0   -0.0046   -1.2400   -9.9938   -0.0000   -0.0138   -
0.4987    0.0000         0         0   -0.0046   -1.2400   -9.9938   -0.0000
-0.0138   -0.4987    0.0000         0
    0.0000         0    0.0000   -0.0000    0.0000         0         0
0         0         0         0    0.0000   -0.0000    0.0000         0
0         0         0         0
    0.0000         0         0    0.0000   -0.0000         0         0
0         0         0         0         0    0.0000   -0.0000         0
0         0         0         0
    0.0000         0         0         0         0         0         0
0         0         0         0         0         0         0         0
0         0         0         0
         0         0         0         0    0.0000         0         0
0         0         0         0         0         0    0.0000         0
0         0         0         0
         0         0         0         0         0         0         0
0         0      -Inf         0         0         0         0         0
0         0         0      -Inf
X = 6×6
10^5 ×
    0.0013    0.0000    0.0000   -0.0411   -0.0008   -0.0000
    0.0000    0.0000    0.0000    0.0000   -0.0000   -0.0000
    0.0000    0.0000    0.0000   -0.0000    0.0000   -0.0000
   -0.0411    0.0000   -0.0000    5.9055    0.0058    0.0001
   -0.0008   -0.0000    0.0000    0.0058    0.0007    0.0000
```

```
   -0.0000   -0.0000   -0.0000    0.0001    0.0000    0.0000
Optimal guaranteed energy-to-peak gain for the uncertain system is:
   9.8038e-04
```



SCREENSHOT:



$$\dot{x}_p = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x_p + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} w \qquad \Rightarrow n_p = 2$$

$$y = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x_p + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$z = \begin{bmatrix} 1 & 0 \end{bmatrix} x_p + \begin{bmatrix} 0 & 1 \end{bmatrix} w$$

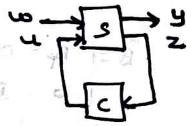
Optimal H_∞ controller design $(n_c = 1)$ $\hookrightarrow$ reduced order $(n_c < n_p)$

$\Gamma_{ee} < \gamma$

$\hookrightarrow \gamma = 3, 4, 5$

General form

$$\dot{x}_p = A_p x_p + B_p u + D_p w$$
$$y = C_p x_p + B_y u + D_y w$$
$$z = M_p x_p + D_z w$$



Comparing given system equations with general form

$$A_p = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad B_p = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad D_p = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$$

$$C_p = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad B_y = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad D_y = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$M_p = \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad D_z = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

* Use above system matrices to obtain X and Y using Standard LMI method and/or Alternating Projection Method (discussed later).

Confirm that $\text{rank}\begin{bmatrix} R \end{bmatrix} = \text{rank}\begin{bmatrix} X & \gamma I \\ \gamma I & Y \end{bmatrix} = \underbrace{3}_{n_p + n_c}$

$\hookrightarrow$ This guarantees existance of first order controller

Perform SVD of $\underbrace{Y - \gamma^2 X^{-1}}_{n_p \times n_p}$

$$\text{SVD}(Y - \gamma^2 X^{-1}) = U \Sigma V^T = {}_{n_p}\Big\{ \begin{bmatrix} \overbrace{U_1}^{n_c} & \overbrace{U_2}^{n_p - n_c} \end{bmatrix} \begin{bmatrix} \overbrace{\Sigma_1}^{n_c \times n_c} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}$$

$$\text{Let} \begin{cases} Y_{11} \in \mathbb{R}^{2\times 2} = Y \\ Y_{22} \in \mathbb{R}^{1\times 1} = I \\ Y_{12} \in \mathbb{R}^{1\times 2} = U_1 \Sigma_1^{1/2} \end{cases} \Rightarrow Y - \gamma^2 X^{-1} = Y_{12} Y_{22}^{-1} Y_{12}^T$$

Construct the augmented matrix

$$P = \begin{bmatrix} Y & Y_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix}$$

Find unknown controller parameters by solving the General Matrix Inequality

$$\Gamma G \Lambda + (\Gamma G \Lambda)^T + Q < 0$$

where $\Gamma = \begin{bmatrix} PB \\ 0 \\ H \end{bmatrix}$, $\Lambda = \begin{bmatrix} M & E & 0 \end{bmatrix}$, $Q = \begin{bmatrix} PA + A^T P & PD & C^T \\ D^T P & -\gamma^2 I & F^T \\ C & F & -I \end{bmatrix}$

where:

$A = \begin{bmatrix} A_p & 0 \\ 0 & 0 \end{bmatrix}$ $\quad B = \begin{bmatrix} B_p & 0 \\ 0 & I \end{bmatrix}$

$M = \begin{bmatrix} M_p & 0 \\ 0 & I \end{bmatrix}$ $\quad E = \begin{bmatrix} D_z \\ 0 \end{bmatrix}$

$H = \begin{bmatrix} B_y & 0 \end{bmatrix}$ $\quad D = \begin{bmatrix} D_p \\ 0 \end{bmatrix}$

$C = \begin{bmatrix} C_p & 0 \end{bmatrix}$ $\quad F = D_y$

$$G = \begin{array}{c} \\ \\ \\ \end{array}\begin{array}{cc|c} & n_z & n_c \\ \hline \end{array}$$

$$G = \begin{array}{cc} \begin{bmatrix} D_c & C_c \\ B_c & A_c \end{bmatrix} & \begin{array}{l} \}\, n_u \\ \}\, n_c \end{array} \\ \begin{array}{cc} \underbrace{\quad}_{n_z} & \underbrace{\quad}_{n_c} \end{array} & \end{array}$$

unknown

i.e. Use $\underbrace{P \text{ and } A, B, C, D, E, F, H, M}_{\Gamma,\ \Lambda,\ Q}$ to solve "basiclmi"

$$G = \text{basiclmi}\,(Q, \Gamma^T, \Lambda)$$

Use partitioned form of controller matrix G to obtain A_c, B_c, C_c and D_c.

Standard LMI Method:

$$\min \ \text{trace}(X+Y)$$

X, Y, γ ← usually unknown/variable but known in this problem $[\gamma = 3,4,5]$

$$\text{s.t.}\ \begin{bmatrix} B_p \\ B_y \end{bmatrix}^{\perp} \begin{bmatrix} A_pX + XA_p^T + D_pD_p^T & XC_p^T + D_pD_y^T \\ C_pX + D_yD_p^T & D_yD_y^T - \gamma^2 I \end{bmatrix} \begin{bmatrix} B_p \\ B_y \end{bmatrix}^{\perp T} < 0 \qquad (C_1)$$

$$\begin{bmatrix} M_p^T \\ D_z^T \end{bmatrix}^{\perp} \begin{bmatrix} YA_p + A_p^TY + C_p^TC_p & YD_p + C_p^TD_y \\ D_p^TY + D_y^TC_p & D_y^TD_y - \gamma^2 I \end{bmatrix} \begin{bmatrix} M_p^T \\ D_z^T \end{bmatrix}^{\perp T} < 0 \qquad (C_2)$$

$$\begin{bmatrix} X & \gamma I \\ \gamma I & Y \end{bmatrix} \geq 0 \qquad (C_3)$$

Alternating Projection Method:

non-convex constraint →

$$\text{rank}\begin{bmatrix} X & \gamma I \\ \gamma I & Y \end{bmatrix} \leq n_p + n_c \qquad (C_4)$$

Idea: Iterate by projecting on $\overbrace{\text{LMIs } (C_1 - C_3)}^{\text{first space}}$ and the $\underbrace{\text{rank constraint } (C_4)}_{\text{second space}}$ back & forth until convergence is met.

Approach:

Working in $(C_1)-(C_3)$ space

Solve $(C_1)-(C_3)$ as an LMI problem to obtain X and Y

Let $R = \begin{bmatrix} X & \gamma I \\ \gamma I & Y \end{bmatrix}$

for iteration = 1: max_iterations (say 100)

- Enforce the rank constraint (if not satisfied)

  $U, \Sigma, V^T = \text{svd}(R)$; $\Sigma = \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_k \end{bmatrix}$ with $\sigma_1 > \sigma_2 \ldots > \sigma_{k-1} > \sigma_k$

  If we replace smallest σ_i of R (i.e. σ_k) by zero we get

  $R' = U \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \sigma_{k-1} & \vdots \\ 0 & \cdots & 0 \end{bmatrix} V^T$ whose rank is one less than original R.

  Instead of replacing one σ_i with 0, we can replace multiple (m) so that rank constraint is satisfied.

- Else (if rank constraint is satisfied)

  $R' = R$

- $\min_{R'} \| R - R' \|$ ← If $\|R - R'\| <$ tolerance ⇒ break

  s.t. $\begin{cases} R' \text{ is solution to } (C_1) - (C_3) & (\leftarrow X, Y) \\ \text{rank}(R') = k - m & \cdots \text{(using SVD)} \end{cases}$

Projection onto C_4 space & working in C_4 space

• $\min_{R_1} \|R_1 - R'\|$ ← If $\|R_1 - R'\| <$ tolerance ⇒ break

s.t. $R_1 \in$ LMIs (C_1)-(C_3)

R_1 is the closest approximation to R' used for projection

Now;

$\because \|R_1 - R'\| < \bar{\gamma} \Rightarrow \begin{bmatrix} -\bar{\gamma}^2 I & R_1 - R' \\ R_1 - R' & -I \end{bmatrix} < 0$

$\therefore \min_{R_1, \gamma} \hat{\gamma}$ ..... $(\hat{\gamma} = \bar{\gamma}^2)$

s.t. $R_1 \in$ LMIs (C_1)-(C_3)

$$\begin{bmatrix} -\hat{\gamma} I & R - R' \\ R - R' & -I \end{bmatrix} < 0$$

↓

Get R_1

Projection (back) onto (C_1)-(C_3) space

↓

Set $R = R_1$

↓

Continue to go back and forth by iterating while projecting on LMI, (C_1)-(C_3) space and rank constraint (C_4) space until convergence is met (i.e. termination criteria based on tolerance or max-iterations is reached).

Problem 2

CODE:

```
% PROBLEM 2


% Clear workspace
close all
clear
clc


% Add parser and solver to path
addpath(genpath('C:\Users\csamak\Downloads\MathWorks\Toolboxes\archives\required\
YALMIP'))
addpath(genpath('C:\Users\csamak\Downloads\MathWorks\Toolboxes\archives\required\
SeDuMi'))


for g = 3:1:5 % Gamma (γ)
    disp('-----------------------------------')
    if g == 3
        disp('Standard LMI Method | CASE 1: γ = 3')
    elseif g == 4
        disp('Standard LMI Method | CASE 2: γ = 4')
    elseif g == 5
        disp('Standard LMI Method | CASE 3: γ = 5')
    end
    disp('-----------------------------------')


    % Define the system matrices
    Ap = [0 1; 0 0];
    Bp = [0; 1];
    Dp = [0 0; 1 0];
    Cp = [0 1; 0 0];
    By = [0; 1];
    Dy = [0 0; 0 0];
    Mp = [1 0];
    Dz = [0 1];

    % Define the LMI variables
    X = sdpvar(2, 2);
    Y = sdpvar(2, 2);

    % Define the LMI constraints
```

```
    C1 = null([Bp; By]')' * [(Ap*X + X*Ap' + Dp*Dp') (X*Cp' + Dp*Dy'); (Cp*X +
Dy*Dp') (Dy*Dy' - g^2*eye(size(Dy)))] * null([Bp; By]') <= 0;
    C2 = null([Mp'; Dz']')' * [(Y*Ap + Ap'*Y + Cp'*Cp) (Y*Dp + Cp'*Dy); (Dp'*Y +
Dy'*Cp) (Dy'*Dy - g^2*eye(size(Dy)))] * null([Mp'; Dz']') <= 0;
    C3 = [X g*eye(size(Dy)); g*eye(size(Dy)) Y] >= 0;

    % Set up the objective
    Objective = trace(X + Y);

    % Define the solver settings (use an LMI solver like YALMIP with a solver of
your choice)
    options = sdpsettings('verbose', 1, 'solver', 'sedumi');

    % Solve the LMI problem
    solution = optimize([C1, C2, C3], Objective, options);

    % 1. Obtain X and Y by solving above optimization problem
    if solution.problem == 0
        % Extract the optimal solutions
        X = value(X);
        Y = value(Y);
        % Display the results
        disp('X*:');
        disp(X);
        disp('Y*:');
        disp(Y);
    else
        fprintf('LMI problem could not be solved.\n');
    end

    % 2. Confirm that rank([X γ*I; γ*I Y]) = 3 and compute terms Y12 and Y22
    if rank([X g*eye(size(Dy)); g*eye(size(Dy)) Y]) == 3
        disp('rank([X γ*I; γ*I Y]) = 3')
        disp('Solving LMI problem resulted in reduced order controller with nc =
1')
    elseif rank([X g*eye(size(Dy)); g*eye(size(Dy)) Y]) == 4
        disp('rank([X γ*I; γ*I Y]) = 4')
        disp('Solving LMI problem resulted in full order controller with nc = 2')
    else
        disp('rank([X γ*I; γ*I Y]) < 3')
        disp('Solving LMI problem resulted in zero order controller with nc = 0')
    end
    [U, S, V] = svd(Y - g^2*inv(X));
    u1 = U(:,1); % Extract u1 so that u1 = np x nc
    S1 = S(1,1); % Extract S1 so that S1 = nc x nc
    disp('Extracted u1 & Σ1 so as to obtain reduced order controller with nc =
1')
    Y12 = u1*(S1^0.5)
```

```
    Y22 = eye(size(Y12', 1), size(Y12, 2))

    % 3. Define augmented matrix P & find controller by solving general matrix
inequality
    P = [Y Y12; Y12' Y22]
    A = [Ap [0; 0]; [0 0] 0];
    B = [Bp [0; 0]; 0 1];
    C = [Cp [0; 0]];
    D = [Dp; [0 0]];
    E = [Dz; [0 0]];
    F = Dy;
    H = [By [0; 0]];
    M = [Mp 0; [0 0] 1];
    Gamma = [P*B; [0 0]; [0 0]; H] % Γ
    Lambda = [M E [0; 0] [0; 0]] % Λ
    Q = [P*A+A'*P P*D C'; D'*P -g^2*eye(2) F'; C F -eye(2)] % Q
    G = basiclmi(Q, Gamma', Lambda); % Solve general matrix inequality
    if size(G) ~= 0
        disp('Controller matrix (G):')
        disp(G)
        fprintf('Ac = %f', value(G(2,2)))
        fprintf('Bc = %f', value(G(2,1)))
        fprintf('Cc = %f', value(G(1,2)))
        fprintf('Dc = %f', value(G(1,1)))
    end
end
```

```
% Clear workspace
close all
clear
clc


% Add parser and solver to path
addpath(genpath('C:\Users\csamak\Downloads\MathWorks\Toolboxes\archives\required\
YALMIP'))
addpath(genpath('C:\Users\csamak\Downloads\MathWorks\Toolboxes\archives\required\
SeDuMi'))


for g = 3:1:5 % Gamma (γ)
    disp('----------------------------------------------')
    if g == 3
        disp('Alternating Projection Method | CASE 1: γ = 3')
    elseif g == 4
        disp('Alternating Projection Method | CASE 2: γ = 4')
```

```
    elseif g == 5
        disp('Alternating Projection Method | CASE 3: γ = 5')
    end
    disp('---------------------------------------------')


    % Define the system matrices
    Ap = [0 1; 0 0];
    Bp = [0; 1];
    Dp = [0 0; 1 0];
    Cp = [0 1; 0 0];
    By = [0; 1];
    Dy = [0 0; 0 0];
    Mp = [1 0];
    Dz = [0 1];

    % Define the LMI variables
    X = sdpvar(2, 2);
    Y = sdpvar(2, 2);

    % Define the LMI constraints
    C1 = null([Bp; By]')' * [(Ap*X + X*Ap' + Dp*Dp') (X*Cp' + Dp*Dy'); (Cp*X +
Dy*Dp') (Dy*Dy' - g^2*eye(size(Dy)))] * null([Bp; By]') <= 0;
    C2 = null([Mp'; Dz']')' * [(Y*Ap + Ap'*Y + Cp'*Cp) (Y*Dp + Cp'*Dy); (Dp'*Y +
Dy'*Cp) (Dy'*Dy - g^2*eye(size(Dy)))] * null([Mp'; Dz']') <= 0;
    C3 = [X g*eye(size(Dy)); g*eye(size(Dy)) Y] >= 0;

    % Set up the objective
    Objective = trace(X + Y);

    % Define the solver settings (use an LMI solver like YALMIP with a solver of
your choice)
    options = sdpsettings('verbose', 1, 'solver', 'sedumi');

    % Alternating projection method
    max_iterations = 100;
    tolerance = 1e-22;

    % Working in C1-C3 space
    solution = optimize([C1, C2, C3], Objective, options);
    if solution.problem == 0
        X = value(X);
        Y = value(Y);
    end
    R = [X, g*eye(size(Dy)); g*eye(size(Dy)), Y];

    for iteration = 1:max_iterations
        [U, S, V] = svd(R);
```

```
        if not(rank(R) == 3)
            S(size(S,1), size(S,2)) = 0;
            R_prime = U*S*V';
        else
            R_prime = R;
        end
        % Check for convergence between R and R_prime
        if norm(R - R_prime) < tolerance
            disp(['Converged at iteration ', num2str(iteration)]);
            break;
        end
        % Projection onto C4 space & working in C4 space
        R1 = sdpvar(size(R,1), size(R,2)); % Define the LMI variable
        g_sqr = sdpvar(1, 1); % Define the LMI variable
        LMI_R1 = [-g_sqr*eye(size(R)) R1-R_prime; R1-R_prime -eye(size(R))] <= 0;
% Define the LMI constraint
        Objective_R1 = g_sqr; % Set up the objective
        solution_R1 = optimize(LMI_R1, Objective_R1, options); % Solve the LMI
problem
        if solution_R1.problem == 0
            R1 = value(R1); % Extract the optimal solution
        end
        % Check for convergence between R1 and R_prime
        if norm(R1 - R_prime) < tolerance
            disp(['Converged at iteration ', num2str(iteration)]);
            break;
        end
        % Projection onto C1-C3 space
        R = R1;
    end

    % Display the results
    disp('X*:');
    disp(X);
    disp('Y*:');
    disp(Y);

    % Display rank information
    if rank(R_prime) == 3
        disp('Solving LMI problem resulted in a reduced-order controller with nc
= 1');
    else
        disp('Rank constraint not satisfied.');
        return; % Exit if the rank constraint is not satisfied
    end

    % Extract u1 & Σ1 to obtain reduced order controller with nc = 1
    [U, S, V] = svd(Y - g^2*inv(X));
```

```
    u1 = U(:, 1);
    S1 = S(1, 1);
    Y12 = u1 * (S1^0.5);
    Y22 = eye(size(Y12', 1), size(Y12, 2));

    % Define augmented matrix P for the controller
    P = [Y, Y12; Y12', Y22];

    % Obtain the controller matrix G
    A = [Ap, [0; 0]; [0 0], 0];
    B = [Bp, [0; 0]; 0, 1];
    C = [Cp, [0; 0]];
    D = [Dp; [0 0]];
    E = [Dz; [0 0]];
    F = Dy;
    H = [By, [0; 0]];
    M = [Mp, 0; [0 0], 1];
    Gamma = [P * B; [0 0]; [0 0]; H]; % Γ
    Lambda = [M, E, [0; 0], [0; 0]]; % Λ
    Q = [P * A + A' * P, P * D, C'; D' * P, -g^2 * eye(2), F'; C, F, -eye(2)]; %
Q
    G = basiclmi(Q, Gamma', Lambda); % Solve general matrix inequality
    if size(G) ~= 0
        disp('Controller matrix (G):')
        disp(G)
        fprintf('Ac = %f', value(G(2,2)))
        fprintf('Bc = %f', value(G(2,1)))
        fprintf('Cc = %f', value(G(1,2)))
        fprintf('Dc = %f', value(G(1,1)))
    end
end
```

OUTPUT:

```
------------------------------------
Standard LMI Method | CASE 1: γ = 3
------------------------------------
SeDuMi 1.3 by AdvOL, 2005-2008 and Jos F. Sturm, 1998-2003.
Alg = 2: xz-corrector, theta = 0.250, beta = 0.500
eqs m = 6, order n = 11, dim = 35, blocks = 4
nnz(A) = 20 + 0, nnz(ADA) = 36, nnz(L) = 21
 it :     b*y       gap    delta  rate   t/tP*  t/tD*   feas cg cg  prec
  0 :            2.71E+01 0.000
  1 :  -5.65E+00 9.11E+00 0.000 0.3364 0.9000 0.9000   1.41  1  1  3.2E+00
  2 :  -1.25E+01 2.87E+00 0.000 0.3153 0.9000 0.9000   0.91  1  1  9.6E-01
  3 :  -1.70E+01 6.35E-01 0.000 0.2209 0.9000 0.9000   0.66  1  1  2.5E-01
  4 :  -1.98E+01 5.13E-02 0.000 0.0808 0.9900 0.9900   0.78  1  1  2.1E-02
  5 :  -2.01E+01 1.90E-03 0.000 0.0369 0.9900 0.9900   0.99  1  1  8.3E-04
  6 :  -2.01E+01 1.03E-04 0.407 0.0545 0.9900 0.9900   1.00  1  1  4.5E-05
  7 :  -2.01E+01 7.99E-06 0.254 0.0773 0.9900 0.9900   1.00  1  1  3.5E-06
```

```
  8 :  -2.01E+01 7.20E-07 0.438 0.0902 0.9900 0.9900   1.00  1  1  3.2E-07
  9 :  -2.01E+01 1.51E-07 0.000 0.2100 0.9000 0.9000   1.00  2  2  6.7E-08
 10 :  -2.01E+01 1.86E-08 0.017 0.1233 0.9450 0.9450   1.00  2  2  8.3E-09
 11 :  -2.01E+01 1.61E-09 0.060 0.0865 0.9900 0.9900   1.00  2  2  7.2E-10

iter seconds digits       c*x               b*y
 11      0.2   9.8 -2.0062574492e+01 -2.0062574496e+01
|Ax-b| =   4.8e-10, [Ay-c]_+ =   9.5E-10, |x|=  1.1e+01, |y|=  1.1e+01

Detailed timing (sec)
   Pre          IPM          Post
3.090E-01    5.140E-01    3.300E-02
Max-norms: ||b||=1, ||c|| = 9,
Cholesky |add|=0, |skip| = 0, ||L.L|| = 6.52152.
X*:
    6.9382   -1.0201
   -1.0201    3.9239
Y*:
    4.2969   -2.3744
   -2.3744    4.9036
rank([X γ*I; γ*I Y]) = 4
Solving LMI problem resulted in full order controller with nc = 2
Extracted u1 & Σ1 so as to obtain reduced order controller with nc = 1
Y12 = 2×1
   -1.7170
    1.5871
Y22 = 1
P = 3×3
    4.2969   -2.3744   -1.7170
   -2.3744    4.9036    1.5871
   -1.7170    1.5871    1.0000
Gamma = 7×2
   -2.3744   -1.7170
    4.9036    1.5871
    1.5871    1.0000
         0         0
         0         0
         0         0
    1.0000         0
Lambda = 2×7
     1     0     0     0     1     0     0
     0     0     1     0     0     0     0
Q = 7×7
         0    4.2969         0   -2.3744         0         0         0
    4.2969   -4.7488   -1.7170    4.9036         0    1.0000         0
         0   -1.7170         0    1.5871         0         0         0
   -2.3744    4.9036    1.5871   -9.0000         0         0         0
         0         0         0         0   -9.0000         0         0
         0    1.0000         0         0         0   -1.0000         0
         0         0         0         0         0         0   -1.0000
   Warning in BASICLMI: the solvability conditions are not satisfied
-----------------------------------
Standard LMI Method | CASE 2: γ = 4
-----------------------------------
SeDuMi 1.3 by AdvOL, 2005-2008 and Jos F. Sturm, 1998-2003.
Alg = 2: xz-corrector, theta = 0.250, beta = 0.500
eqs m = 6, order n = 11, dim = 35, blocks = 4
```

```
nnz(A) = 20 + 0, nnz(ADA) = 36, nnz(L) = 21
 it :     b*y       gap    delta  rate   t/tP*  t/tD*   feas cg cg  prec
  0 :            4.74E+01 0.000
  1 :  -7.71E+00 1.53E+01 0.000 0.3239 0.9000 0.9000   1.46  1  1  2.8E+00
  2 :  -1.67E+01 4.33E+00 0.000 0.2820 0.9000 0.9000   1.04  1  1  8.1E-01
  3 :  -2.02E+01 8.97E-01 0.000 0.2072 0.9000 0.9000   0.83  1  1  1.8E-01
  4 :  -2.23E+01 4.34E-02 0.000 0.0484 0.9900 0.9900   0.85  1  1  8.9E-03
  5 :  -2.25E+01 2.43E-03 0.160 0.0560 0.9675 0.9675   1.00  1  1  5.0E-04
  6 :  -2.25E+01 2.68E-04 0.198 0.1101 0.9450 0.9450   1.00  1  1  5.5E-05
  7 :  -2.25E+01 2.09E-05 0.319 0.0779 0.9900 0.9900   1.00  1  1  4.3E-06
  8 :  -2.25E+01 9.96E-06 0.205 0.4772 0.9000 0.9000   1.00  1  1  2.1E-06
  9 :  -2.25E+01 2.48E-06 0.000 0.2490 0.9000 0.9000   1.00  1  1  5.2E-07
 10 :  -2.25E+01 2.29E-07 0.369 0.0922 0.9900 0.9900   1.00  2  2  4.9E-08
 11 :  -2.25E+01 8.19E-08 0.171 0.3580 0.9000 0.9000   1.00  2  2  1.8E-08
 12 :  -2.25E+01 6.92E-09 0.149 0.0845 0.9900 0.9900   1.00  2  2  1.5E-09
 13 :  -2.25E+01 6.74E-10 0.139 0.0974 0.9900 0.9900   1.00  2  2  1.5E-10

iter seconds digits       c*x               b*y
 13      0.0  10.2 -2.2456285268e+01 -2.2456285269e+01
|Ax-b| =   8.7e-11, [Ay-c]_+ =   2.7E-10, |x|=  7.9e+00, |y|=  1.2e+01

Detailed timing (sec)
   Pre          IPM          Post
1.200E-02    3.201E-02    2.997E-03
Max-norms: ||b||=1, ||c|| = 16,
Cholesky |add|=0, |skip| = 0, ||L.L|| = 4.35694.
X*:
    6.9145   -0.8214
   -0.8214    4.9001
Y*:
    4.9551   -2.0759
   -2.0759    5.6866
rank([X γ*I; γ*I Y]) = 4
Solving LMI problem resulted in full order controller with nc = 2
Extracted u1 & Σ1 so as to obtain reduced order controller with nc = 1
Y12 = 2×1
   -1.6106
    1.5346
Y22 = 1
P = 3×3
    4.9551   -2.0759   -1.6106
   -2.0759    5.6866    1.5346
   -1.6106    1.5346    1.0000
Gamma = 7×2
   -2.0759   -1.6106
    5.6866    1.5346
    1.5346    1.0000
         0         0
         0         0
         0         0
    1.0000         0
Lambda = 2×7
     1     0     0     0     1     0     0
     0     0     1     0     0     0     0
Q = 7×7
         0    4.9551         0   -2.0759         0         0         0
    4.9551   -4.1518   -1.6106    5.6866         0    1.0000         0
```

```
         0   -1.6106         0    1.5346         0         0         0
   -2.0759    5.6866    1.5346  -16.0000         0         0         0
         0         0         0         0  -16.0000         0         0
         0    1.0000         0         0         0   -1.0000         0
         0         0         0         0         0         0   -1.0000
Controller matrix (G):
   -2.6758    1.3083
    8.5579   -5.0717
Ac = -5.071654
Bc = 8.557912
Cc = 1.308284
Dc = -2.675828
-----------------------------------
Standard LMI Method | CASE 3: γ = 5
-----------------------------------
SeDuMi 1.3 by AdvOL, 2005-2008 and Jos F. Sturm, 1998-2003.
Alg = 2: xz-corrector, theta = 0.250, beta = 0.500
eqs m = 6, order n = 11, dim = 35, blocks = 4
nnz(A) = 20 + 0, nnz(ADA) = 36, nnz(L) = 21
 it :     b*y       gap    delta  rate   t/tP*  t/tD*   feas cg cg  prec
  0 :            7.35E+01 0.000
  1 :  -1.03E+01 2.28E+01 0.000 0.3097 0.9000 0.9000   1.48  1  1  2.6E+00
  2 :  -2.10E+01 5.93E+00 0.000 0.2602 0.9000 0.9000   1.11  1  1  7.3E-01
  3 :  -2.36E+01 1.27E+00 0.000 0.2149 0.9000 0.9000   0.92  1  1  1.6E-01
  4 :  -2.57E+01 5.44E-02 0.000 0.0427 0.9900 0.9900   0.87  1  1  7.1E-03
  5 :  -2.58E+01 2.00E-03 0.024 0.0368 0.9900 0.9900   1.00  1  1  2.6E-04
  6 :  -2.58E+01 2.24E-04 0.206 0.1119 0.9450 0.9450   1.00  1  1  2.9E-05
  7 :  -2.58E+01 1.85E-05 0.480 0.0825 0.9900 0.9900   1.00  1  1  2.4E-06
  8 :  -2.58E+01 8.75E-06 0.164 0.4733 0.9000 0.9000   1.00  1  1  1.2E-06
  9 :  -2.58E+01 1.63E-06 0.000 0.1859 0.9000 0.9000   1.00  1  1  2.2E-07
 10 :  -2.58E+01 1.36E-07 0.471 0.0837 0.9900 0.9900   1.00  1  2  1.8E-08
 11 :  -2.58E+01 2.69E-08 0.000 0.1976 0.9000 0.9000   1.00  2  2  3.6E-09
 12 :  -2.58E+01 1.17E-09 0.359 0.0434 0.9900 0.9900   1.00  2  2  1.6E-10

iter seconds digits       c*x               b*y
 12      0.0  10.1 -2.5756604760e+01 -2.5756604762e+01
|Ax-b| =   8.3e-11, [Ay-c]_+ =   4.2E-10, |x|=  6.6e+00, |y|=  1.3e+01

Detailed timing (sec)
   Pre          IPM          Post
3.300E-02    2.400E-02    2.002E-03
Max-norms: ||b||=1, ||c|| = 25,
Cholesky |add|=0, |skip| = 0, ||L.L|| = 4.93398.
X*:
    7.4856   -0.7233
   -0.7233    5.8495
Y*:
    5.8149   -1.9387
   -1.9387    6.6066
rank([X γ*I; γ*I Y]) = 4
Solving LMI problem resulted in full order controller with nc = 2
Extracted u1 & Σ1 so as to obtain reduced order controller with nc = 1
Y12 = 2×1
   -1.5604
    1.5103
Y22 = 1
P = 3×3
```

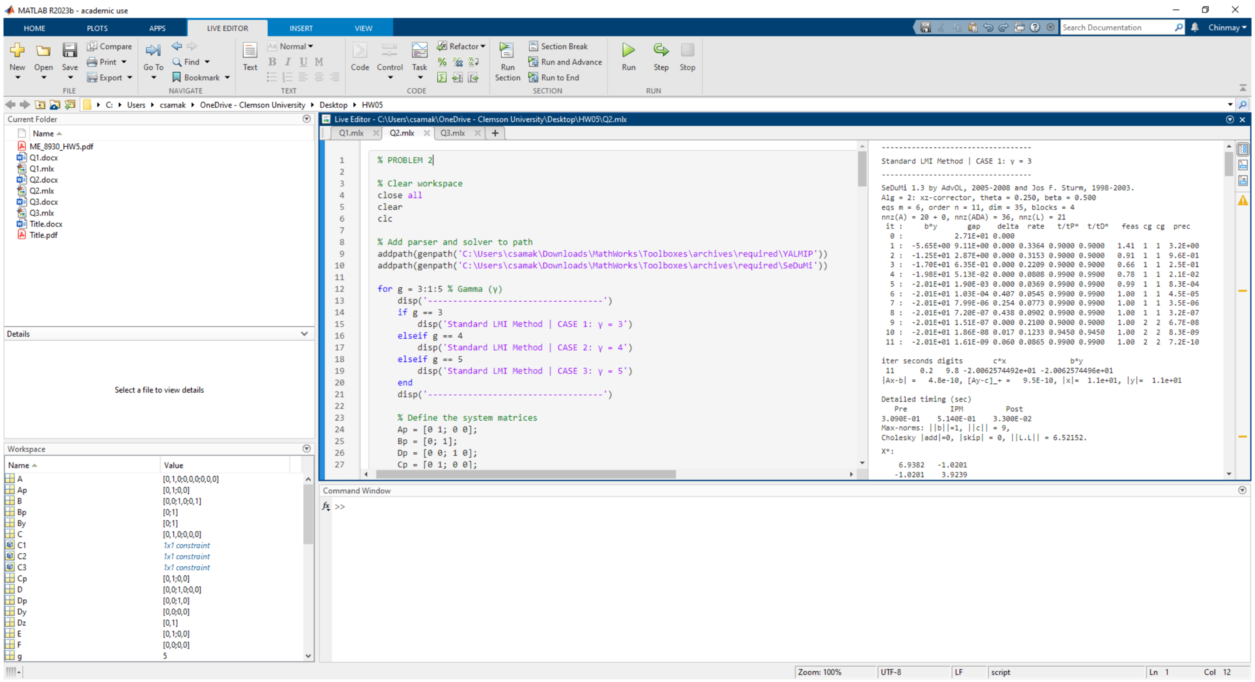
```
    5.8149   -1.9387   -1.5604
   -1.9387    6.6066    1.5103
   -1.5604    1.5103    1.0000
Gamma = 7×2
   -1.9387   -1.5604
    6.6066    1.5103
    1.5103    1.0000
         0         0
         0         0
         0         0
    1.0000         0
Lambda = 2×7
     1     0     0     0     1     0     0
     0     0     1     0     0     0     0
Q = 7×7
         0    5.8149         0   -1.9387         0         0         0
    5.8149   -3.8774   -1.5604    6.6066         0    1.0000         0
         0   -1.5604         0    1.5103         0         0         0
   -1.9387    6.6066    1.5103  -25.0000         0         0         0
         0         0         0         0  -25.0000         0         0
         0    1.0000         0         0         0   -1.0000         0
         0         0         0         0         0         0   -1.0000
Controller matrix (G):
   -3.1340    1.6176
   11.9541   -7.5012
Ac = -7.501234
Bc = 11.954090
Cc = 1.617626
Dc = -3.134028
---------------------------------------------
Alternating Projection Method | CASE 1: γ = 3
---------------------------------------------
SeDuMi 1.3 by AdvOL, 2005-2008 and Jos F. Sturm, 1998-2003.
Alg = 2: xz-corrector, theta = 0.250, beta = 0.500
eqs m = 6, order n = 11, dim = 35, blocks = 4
nnz(A) = 20 + 0, nnz(ADA) = 36, nnz(L) = 21
 it :     b*y       gap    delta  rate   t/tP*  t/tD*   feas cg cg  prec
  0 :            2.71E+01 0.000
  1 :  -5.65E+00 9.11E+00 0.000 0.3364 0.9000 0.9000   1.41  1  1  3.2E+00
  2 :  -1.25E+01 2.87E+00 0.000 0.3153 0.9000 0.9000   0.91  1  1  9.6E-01
  3 :  -1.70E+01 6.35E-01 0.000 0.2209 0.9000 0.9000   0.66  1  1  2.5E-01
  4 :  -1.98E+01 5.13E-02 0.000 0.0808 0.9900 0.9900   0.78  1  1  2.1E-02
  5 :  -2.01E+01 1.90E-03 0.000 0.0369 0.9900 0.9900   0.99  1  1  8.3E-04
  6 :  -2.01E+01 1.03E-04 0.407 0.0545 0.9900 0.9900   1.00  1  1  4.5E-05
  7 :  -2.01E+01 7.99E-06 0.254 0.0773 0.9900 0.9900   1.00  1  1  3.5E-06
  8 :  -2.01E+01 7.20E-07 0.438 0.0902 0.9900 0.9900   1.00  1  1  3.2E-07
  9 :  -2.01E+01 1.51E-07 0.000 0.2100 0.9000 0.9000   1.00  2  2  6.7E-08
 10 :  -2.01E+01 1.86E-08 0.017 0.1233 0.9450 0.9450   1.00  2  2  8.3E-09
 11 :  -2.01E+01 1.61E-09 0.060 0.0865 0.9900 0.9900   1.00  2  2  7.2E-10

iter seconds digits       c*x               b*y
 11      0.0   9.8 -2.0062574492e+01 -2.0062574496e+01
|Ax-b| =   4.8e-10, [Ay-c]_+ =   9.5E-10, |x|=  1.1e+01, |y|=  1.1e+01

Detailed timing (sec)
   Pre          IPM          Post
1.100E-02    3.900E-02    2.997E-03
```

```
Max-norms: ||b||=1, ||c|| = 9,
Cholesky |add|=0, |skip| = 0, ||L.L|| = 6.52152.
SeDuMi 1.3 by AdvOL, 2005-2008 and Jos F. Sturm, 1998-2003.
Alg = 2: xz-corrector, theta = 0.250, beta = 0.500
eqs m = 11, order n = 9, dim = 65, blocks = 2
nnz(A) = 20 + 0, nnz(ADA) = 121, nnz(L) = 66
 it :     b*y       gap    delta  rate   t/tP*  t/tD*   feas cg cg  prec
  0 :            8.16E+00 0.000
  1 :  -2.50E+00 3.13E+00 0.000 0.3836 0.9000 0.9000   2.27  1  1  2.7E+00
  2 :   1.31E-01 8.96E-01 0.000 0.2861 0.9000 0.9000   3.42  1  1  3.8E-01
  3 :   5.07E-03 1.99E-02 0.000 0.0222 0.9900 0.9900   1.25  1  1  1.2E-01
  4 :   1.19E-07 3.91E-07 0.000 0.0000 1.0000 1.0000   1.01  1  1  2.3E-05
  5 :   5.71E-14 1.85E-13 0.442 0.0000 1.0000 1.0000   1.00  1  1  1.1E-11

iter seconds digits       c*x               b*y
  5      0.0   8.9  1.1776667857e-13  5.7097292825e-14
|Ax-b| =   3.0e-14, [Ay-c]_+ =   5.7E-14, |x|=  5.0e-01, |y|=  1.1e+01

Detailed timing (sec)
   Pre          IPM          Post
5.501E-02    2.299E-02    9.002E-03
Max-norms: ||b||=1, ||c|| = 1.387637e+01,
Cholesky |add|=0, |skip| = 0, ||L.L|| = 1.
Converged at iteration 2
X*:
    6.9382   -1.0201
   -1.0201    3.9239
Y*:
    4.2969   -2.3744
   -2.3744    4.9036
Solving LMI problem resulted in a reduced-order controller with nc = 1
   Warning in BASICLMI: the solvability conditions are not satisfied
---------------------------------------------
Alternating Projection Method | CASE 2: γ = 4
---------------------------------------------
SeDuMi 1.3 by AdvOL, 2005-2008 and Jos F. Sturm, 1998-2003.
Alg = 2: xz-corrector, theta = 0.250, beta = 0.500
eqs m = 6, order n = 11, dim = 35, blocks = 4
nnz(A) = 20 + 0, nnz(ADA) = 36, nnz(L) = 21
 it :     b*y       gap    delta  rate   t/tP*  t/tD*   feas cg cg  prec
  0 :            4.74E+01 0.000
  1 :  -7.71E+00 1.53E+01 0.000 0.3239 0.9000 0.9000   1.46  1  1  2.8E+00
  2 :  -1.67E+01 4.33E+00 0.000 0.2820 0.9000 0.9000   1.04  1  1  8.1E-01
  3 :  -2.02E+01 8.97E-01 0.000 0.2072 0.9000 0.9000   0.83  1  1  1.8E-01
  4 :  -2.23E+01 4.34E-02 0.000 0.0484 0.9900 0.9900   0.85  1  1  8.9E-03
  5 :  -2.25E+01 2.43E-03 0.160 0.0560 0.9675 0.9675   1.00  1  1  5.0E-04
  6 :  -2.25E+01 2.68E-04 0.198 0.1101 0.9450 0.9450   1.00  1  1  5.5E-05
  7 :  -2.25E+01 2.09E-05 0.319 0.0779 0.9900 0.9900   1.00  1  1  4.3E-06
  8 :  -2.25E+01 9.96E-06 0.205 0.4772 0.9000 0.9000   1.00  1  1  2.1E-06
  9 :  -2.25E+01 2.48E-06 0.000 0.2490 0.9000 0.9000   1.00  1  1  5.2E-07
 10 :  -2.25E+01 2.29E-07 0.369 0.0922 0.9900 0.9900   1.00  2  2  4.9E-08
 11 :  -2.25E+01 8.19E-08 0.171 0.3580 0.9000 0.9000   1.00  2  2  1.8E-08
 12 :  -2.25E+01 6.92E-09 0.149 0.0845 0.9900 0.9900   1.00  2  2  1.5E-09
 13 :  -2.25E+01 6.74E-10 0.139 0.0974 0.9900 0.9900   1.00  2  2  1.5E-10

iter seconds digits       c*x               b*y
 13      0.0  10.2 -2.2456285268e+01 -2.2456285269e+01
```

```
|Ax-b| =   8.7e-11, [Ay-c]_+ =   2.7E-10, |x|=  7.9e+00, |y|=  1.2e+01

Detailed timing (sec)
   Pre          IPM          Post
2.100E-02    2.800E-02    2.002E-03
Max-norms: ||b||=1, ||c|| = 16,
Cholesky |add|=0, |skip| = 0, ||L.L|| = 4.35694.
SeDuMi 1.3 by AdvOL, 2005-2008 and Jos F. Sturm, 1998-2003.
Alg = 2: xz-corrector, theta = 0.250, beta = 0.500
eqs m = 11, order n = 9, dim = 65, blocks = 2
nnz(A) = 20 + 0, nnz(ADA) = 121, nnz(L) = 66
 it :     b*y       gap    delta  rate   t/tP*  t/tD*   feas cg cg  prec
  0 :            8.16E+00 0.000
  1 :  -2.49E+00 3.13E+00 0.000 0.3838 0.9000 0.9000   2.28  1  1  2.7E+00
  2 :   1.30E-01 8.94E-01 0.000 0.2857 0.9000 0.9000   3.42  1  1  3.8E-01
  3 :   5.01E-03 1.97E-02 0.000 0.0221 0.9900 0.9900   1.25  1  1  1.2E-01
  4 :   1.17E-07 3.85E-07 0.000 0.0000 1.0000 1.0000   1.01  1  1  2.3E-05
  5 :   3.01E-14 9.08E-14 0.202 0.0000 1.0000 1.0000   1.00  1  1  5.5E-12

iter seconds digits       c*x               b*y
  5      0.0   9.4  5.0899527759e-14  3.0144842876e-14
|Ax-b| =   1.7e-14, [Ay-c]_+ =   3.0E-14, |x|=  5.0e-01, |y|=  1.3e+01

Detailed timing (sec)
   Pre          IPM          Post
2.997E-03    1.001E-02    9.958E-04
Max-norms: ||b||=1, ||c|| = 1.382892e+01,
Cholesky |add|=0, |skip| = 0, ||L.L|| = 1.
Converged at iteration 2
X*:
    6.9145   -0.8214
   -0.8214    4.9001
Y*:
    4.9551   -2.0759
   -2.0759    5.6866
Solving LMI problem resulted in a reduced-order controller with nc = 1
Controller matrix (G):
   -2.6758    1.3083
    8.5579   -5.0717
Ac = -5.071654
Bc = 8.557912
Cc = 1.308284
Dc = -2.675828
---------------------------------------------
Alternating Projection Method | CASE 3: γ = 5
---------------------------------------------
SeDuMi 1.3 by AdvOL, 2005-2008 and Jos F. Sturm, 1998-2003.
Alg = 2: xz-corrector, theta = 0.250, beta = 0.500
eqs m = 6, order n = 11, dim = 35, blocks = 4
nnz(A) = 20 + 0, nnz(ADA) = 36, nnz(L) = 21
 it :     b*y       gap    delta  rate   t/tP*  t/tD*   feas cg cg  prec
  0 :            7.35E+01 0.000
  1 :  -1.03E+01 2.28E+01 0.000 0.3097 0.9000 0.9000   1.48  1  1  2.6E+00
  2 :  -2.10E+01 5.93E+00 0.000 0.2602 0.9000 0.9000   1.11  1  1  7.3E-01
  3 :  -2.36E+01 1.27E+00 0.000 0.2149 0.9000 0.9000   0.92  1  1  1.6E-01
  4 :  -2.57E+01 5.44E-02 0.000 0.0427 0.9900 0.9900   0.87  1  1  7.1E-03
  5 :  -2.58E+01 2.00E-03 0.024 0.0368 0.9900 0.9900   1.00  1  1  2.6E-04
```

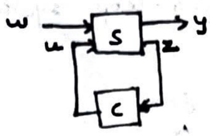
```
  6 :  -2.58E+01 2.24E-04 0.206 0.1119 0.9450 0.9450   1.00  1  1  2.9E-05
  7 :  -2.58E+01 1.85E-05 0.480 0.0825 0.9900 0.9900   1.00  1  1  2.4E-06
  8 :  -2.58E+01 8.75E-06 0.164 0.4733 0.9000 0.9000   1.00  1  1  1.2E-06
  9 :  -2.58E+01 1.63E-06 0.000 0.1859 0.9000 0.9000   1.00  1  1  2.2E-07
 10 :  -2.58E+01 1.36E-07 0.471 0.0837 0.9900 0.9900   1.00  1  2  1.8E-08
 11 :  -2.58E+01 2.69E-08 0.000 0.1976 0.9000 0.9000   1.00  2  2  3.6E-09
 12 :  -2.58E+01 1.17E-09 0.359 0.0434 0.9900 0.9900   1.00  2  2  1.6E-10

iter seconds digits       c*x               b*y
 12      0.0  10.1 -2.5756604760e+01 -2.5756604762e+01
|Ax-b| =   8.3e-11, [Ay-c]_+ =   4.2E-10, |x|=  6.6e+00, |y|=  1.3e+01

Detailed timing (sec)
   Pre          IPM          Post
2.002E-03    2.700E-02    2.002E-03
Max-norms: ||b||=1, ||c|| = 25,
Cholesky |add|=0, |skip| = 0, ||L.L|| = 4.93398.
SeDuMi 1.3 by AdvOL, 2005-2008 and Jos F. Sturm, 1998-2003.
Alg = 2: xz-corrector, theta = 0.250, beta = 0.500
eqs m = 11, order n = 9, dim = 65, blocks = 2
nnz(A) = 20 + 0, nnz(ADA) = 121, nnz(L) = 66
 it :     b*y       gap    delta  rate   t/tP*  t/tD*   feas cg cg  prec
  0 :            8.33E+00 0.000
  1 :  -2.75E+00 3.18E+00 0.000 0.3812 0.9000 0.9000   2.25  1  1  2.7E+00
  2 :   1.48E-01 9.32E-01 0.000 0.2935 0.9000 0.9000   3.43  1  1  3.9E-01
  3 :   6.51E-03 2.33E-02 0.000 0.0250 0.9900 0.9900   1.28  1  1  1.2E-01
  4 :   1.86E-07 5.57E-07 0.000 0.0000 1.0000 1.0000   1.01  1  1  3.1E-05
  5 :   4.75E-14 1.27E-13 0.301 0.0000 1.0000 1.0000   1.00  1  1  6.9E-12

iter seconds digits       c*x               b*y
  5      0.0   9.2  7.8883843350e-14  4.7484765648e-14
|Ax-b| =   2.1e-14, [Ay-c]_+ =   4.8E-14, |x|=  5.0e-01, |y|=  1.5e+01

Detailed timing (sec)
   Pre          IPM          Post
2.299E-02    9.998E-03    1.006E-03
Max-norms: ||b||=1, ||c|| = 1.497125e+01,
Cholesky |add|=0, |skip| = 0, ||L.L|| = 1.
Converged at iteration 2
X*:
    7.4856   -0.7233
   -0.7233    5.8495
Y*:
    5.8149   -1.9387
   -1.9387    6.6066
Solving LMI problem resulted in a reduced-order controller with nc = 1
Controller matrix (G):
   -3.1340    1.6176
   11.9541   -7.5012
Ac = -7.501234
Bc = 11.954090
Cc = 1.617626
Dc = -3.134028
```

SCREENSHOT:



MATLAB R2023b - academic use
Live Editor - C:\Users\csamak\OneDrive - Clemson University\Desktop\HW05\Q2.mlx
Q1.mlx
Q2.mlx
Q3.mlx
% PROBLEM 2
% Clear workspace
close all
clear
clc
% Add parser and solver to path
addpath(genpath('C:\Users\csamak\Downloads\MathWorks\Toolboxes\archives\required\YALMIP'))
addpath(genpath('C:\Users\csamak\Downloads\MathWorks\Toolboxes\archives\required\SeDuMi'))
for g = 3:1:5 % Gamma (γ)
disp('----------------------------------')
if g == 3
disp('Standard LMI Method | CASE 1: γ = 3')
elseif g == 4
disp('Standard LMI Method | CASE 2: γ = 4')
elseif g == 5
disp('Standard LMI Method | CASE 3: γ = 5')
end
disp('----------------------------------')
% Define the system matrices
Ap = [0 1; 0 0];
Bp = [0; 1];
Dp = [0 0; 1 0];
Cp = [0 1; 0 0];
Standard LMI Method | CASE 1: γ = 3
SeDuMi 1.3 by AdvOL, 2005-2008 and Jos F. Sturm, 1998-2003.
Alg = 2: xz-corrector, theta = 0.250, beta = 0.500
eqs m = 6, order n = 11, dim = 35, blocks = 4
nnz(A) = 20 + 0, nnz(ADA) = 36, nnz(L) = 21
it : b*y gap delta rate t/tP* t/tD* feas cg cg prec
0 : 2.71E+01 0.000
1 : -5.65E+00 9.11E+00 0.000 0.3364 0.9000 0.9000 1.41 1 1 3.2E+00
2 : -1.25E+01 2.87E+00 0.000 0.3153 0.9000 0.9000 0.91 1 1 9.6E-01
3 : -1.70E+01 6.35E-01 0.000 0.2209 0.9000 0.9000 0.66 1 1 2.5E-01
4 : -1.98E+01 5.13E-02 0.000 0.0808 0.9900 0.9900 0.78 1 1 2.1E-02
5 : -2.01E+01 1.90E-03 0.000 0.0369 0.9900 0.9900 0.99 1 1 8.3E-04
6 : -2.01E+01 1.03E-04 0.407 0.0545 0.9900 0.9900 1.00 1 1 4.5E-05
7 : -2.01E+01 7.99E-06 0.254 0.0773 0.9900 0.9900 1.00 1 1 3.5E-06
8 : -2.01E+01 7.20E-07 0.438 0.0902 0.9900 0.9900 1.00 1 1 3.2E-07
9 : -2.01E+01 1.51E-07 0.000 0.2100 0.9000 0.9000 1.00 2 2 6.7E-08
10 : -2.01E+01 1.86E-08 0.017 0.1233 0.9450 0.9450 1.00 2 2 8.3E-09
11 : -2.01E+01 1.61E-09 0.060 0.0865 0.9900 0.9900 1.00 2 2 7.2E-10
iter seconds digits c*x b*y
11 0.2 9.8 -2.0062574492e+01 -2.0062574496e+01
|Ax-b| = 4.8e-10, [Ay-c]_+ = 9.5E-10, |x|= 1.1e+01, |y|= 1.1e+01
Detailed timing (sec)
Pre IPM Post
3.090E-01 5.140E-01 3.300E-02
Max-norms: ||b||=1, ||c|| = 9,
Cholesky |add|=0, |skip| = 0, ||L.L|| = 6.52152.
X*:
6.9382 -1.0201
-1.0201 3.9239
Command Window

$$\dot{x}_p = A_p x_p + B_p u + D_p w$$
$$y = C_p x_p + B_y u + D_y w$$
$$z = M_p x_p + D_z w$$



Comparing with given $\overbrace{\text{system}}^{\text{F8 Aircraft}}$ equations:

$$A_p = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1.5 & -1.5 & 0 & 0.0057 & 1.5 & 0 & 0 & 0 \\ -12 & 12 & -0.6 & -0.0344 & -12 & 0 & 0 & 0 \\ -0.852 & 0.29 & 0 & -0.014 & -0.29 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.73 & 2.82940625 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1.25 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1000 \end{bmatrix}$$

$$B_p = \begin{bmatrix} 0 & 0 \\ 0.16 & 0.8 \\ -19 & -3 \\ -0.0115 & -0.0087 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad D_p = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.1149 & 0 & 0 \\ 4 & 0 & 0 \\ 0 & 1024 & 0 \\ 0 & 0 & 1024 \end{bmatrix}$$

$$C_p = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad B_y = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.01 & 0 \\ 0 & 0.01 \end{bmatrix} \qquad D_y = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$M_p = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & -139.020647321 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -139.020647321 \end{bmatrix} \qquad D_z = \begin{bmatrix} 0 & 142.857142857 & 0 \\ 0 & 0 & 142.857142857 \end{bmatrix}$$

Lumped system matrices:

$$A = A_p$$
$$B = \begin{bmatrix} D_p & B_p \end{bmatrix}$$
$$C = \begin{bmatrix} C_p & M_p \end{bmatrix}$$
$$D = \begin{bmatrix} D_y & B_y \\ D_z & 0 \end{bmatrix}$$

Open-loop system:

$$S_{OL} = ss(A, B, C, D)$$

Input to open-loop system:

- $w(t) = \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix} \delta(t)$ ← unit impulse disturbance

where $\delta(t) = \begin{cases} 1 \; ; & 0 \leq t \leq \underbrace{0.01}_{\Delta t} \\ 0 \; ; & t > 0.01 \end{cases}$

- $u(t) = 0 \quad \forall t$

Open-loop system simulation:

- Simulate open-loop system S_{OL} using "lsim" for $t=0$ to $t=10$ sec. with $\Delta t = 0.01$ sec.
- Plot $w(t)$ vs. t ← wind gust disturbance (input)
  Plot $y_1(t)$ vs. t ← aircraft pitch angle (output)

Full order optimal H_∞ dynamic controller design

$S = \text{ltisys}(A, B, C, D)$

$n_z = \text{size}(M_p, 1)$

$n_u = \text{size}(B_p, 2)$

$r = [n_z \;\; n_u]$

$[\text{gopt}, G] = \text{hinflmi}(S, r)$

$\Gamma_{ee} < \gamma^*$ controller

$[A_c, B_c, C_c, D_c] = \text{ltiss}(G)$

Closed-loop system:

$$A_{CL} = \begin{bmatrix} A_p + B_p D_c M_p & B_p C_c \\ B_c M_p & A_c \end{bmatrix} \qquad B_{CL} = \begin{bmatrix} D_p + B_p D_c D_z \\ B_c D_z \end{bmatrix}$$

$$C_{CL} = \begin{bmatrix} C_p + B_y D_c M_p & B_y C_c \end{bmatrix} \qquad D_{CL} = D_y + B_y D_c D_z$$

$$S_{CL} = ss(A_{CL}, B_{CL}, C_{CL}, D_{CL})$$

Design Validation:

- Stability : Eigenvalues (Poles) : $Re(\lambda_i(A_{cl})) < 0$ ← closed loop system stability guaranteed

- Performance: H_∞ norm (Γ_{ee}) : Γ_{ee} = hinfnorm (S_{cl})

  $\underset{\text{hinf\_norm}}{\Gamma_{ee}} < \underset{\text{gopt}}{\gamma^*}$ ← optimal (guaranteed) H_∞ performance

Input to closed-loop system:

- $w(t) = \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix} \delta(t)$ ← unit impulse signal

  where $\delta(t) = \begin{cases} 1; & 0 \le t \le 0.01 \\ 0; & t > \underbrace{0.01}_{\Delta t} \end{cases}$

Closed-loop system simulation:

- Simulate closed-loop system S_{cl} using "lsim" for $t = 0$ to $t = 10$ sec. with $\Delta t = 0.01$ sec.

- Plot $w(t)$ vs. t ← wind gust disturbance (input)

  Plot $y_1(t)$ vs. t ← aircraft pitch angle (output)

Problem 3

CODE:

```
% PROBLEM 3


% Clear workspace
close all
clear
clc


% Define the system matrices
Ap = [0 0 1 0 0 0 0 0;
      1.5 -1.5 0 0.0057 1.5 0 0 0;
      -12 12 -0.6 -0.0344 -12 0 0 0;
      -0.825 0.29 0 -0.014 -0.29 0 0 0;
      0 0 0 0 -0.73 2.82940625 0 0;
      0 0 0 0 0 -1.25 0 0;
      0 0 0 0 0 0 -1000 0;
      0 0 0 0 0 0 0 -1000];
Bp = [0 0;
      0.16 0.8;
      -19 -3;
      -0.0115 -0.0087;
      0 0;
      0 0;
      0 0;
      0 0];
Dp = [0 0 0;
      0 0 0;
      0 0 0;
      0 0 0;
      0.1149 0 0;
      4 0 0;
      0 1024 0
      0 0 1024];
Cp = [1 0 0 0 0 0 0 0;
      0 1 0 0 0 0 0 0;
      0 0 0 0 0 0 0 0;
      0 0 0 0 0 0 0 0];
By = [0 0;
      0 0;
      0.01 0;
      0 0.01];
Dy = [0 0 0;
```

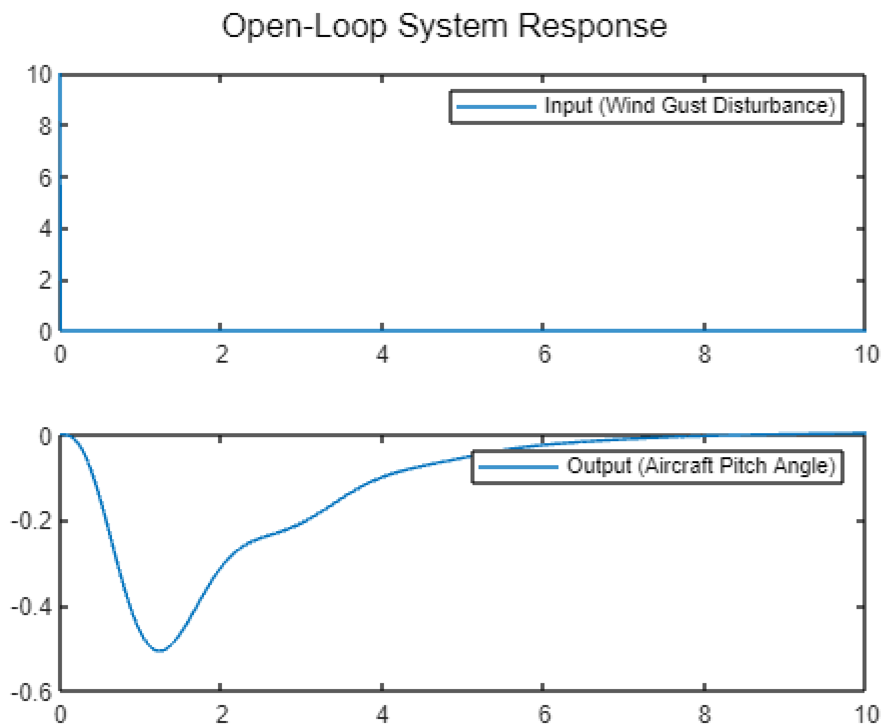
```
      0 0 0;
      0 0 0;
      0 0 0];
Mp = [1 0 0 0 0 0 -139.020647321 0
      0 1 0 0 0 0 0 -139.020647321];
Dz = [0 142.857142857 0
      0 0 142.857142857];


% Lumped system matrices
A = Ap;
B = [Dp Bp];
C = [Cp; Mp];
D = [Dy By; Dz zeros(size(Dz,1), size(By,2))];


% State-space system
Sol = ss(A, B, C, D)


% Define the lumped (disturbance + control) input
t = 0:0.01:10;
w_amplitude = [10; 0; 0];
w_duration = 0.01;
w_impulse = w_amplitude * (t >= 0 & t < w_duration);
u = zeros(size(Bp,2), size(w_impulse,2));
w = [w_impulse; u];


% Simulate the open-loop system response
[y_ol, t_out, x_ol] = lsim(Sol, w', t);


% Plot the results
figure;
sgtitle('Open-Loop System Response');
subplot(2, 1, 1);
plot(t, w(1, :));
legend('Input (Wind Gust Disturbance)');
subplot(2, 1, 2);
plot(t, y_ol(:, 1));
legend('Output (Aircraft Pitch Angle)');


% H∞ controller design
S = ltisys(A, B, C, D);
nz = size(Mp, 1);
nu = size(Bp, 2);
```

```
r = [nz nu];
[gopt, G] = hinflmi(S, r)


% H∞ controller matrices
disp('H∞ controller:')
[Ac, Bc, Cc, Dc] = ltiss(G)


% Closed-loop system matrices
disp('Closed-loop system:')
Acl = [Ap+Bp*Dc*Mp, Bp*Cc; Bc*Mp, Ac]
Bcl = [Dp+Bp*Dc*Dz; Bc*Dz]
Ccl = [Cp+By*Dc*Mp, By*Cc]
Dcl = Dy+By*Dc*Dz


% Closed-loop system
Scl = ss(Acl, Bcl, Ccl, Dcl);


% Verification of designed H∞ controller
disp('H∞ norm:')
hinf_norm = hinfnorm(Scl)
disp('Closed-loop poles:')
eig_Acl = eig(Acl)
if((hinf_norm < gopt) && all(real(eig_Acl) < 0.0))
    disp('Verification of H∞ norm and stability constraints successful!')
else
    disp('Verification of H∞ norm and stability constraints failed!')
end


% Define the lumped (disturbance + control) input
t = 0:0.01:10;
w_amplitude = [10; 0; 0];
w_duration = 0.01;
w_impulse = w_amplitude * (t >= 0 & t < w_duration);
w = w_impulse;


% Simulate the open-loop system response
[y_cl, t_out, x_cl] = lsim(Scl, w', t);


% Plot the results
figure;
sgtitle('Closed-Loop System Response');
subplot(2, 1, 1);
```

```
plot(t, w(1, :));
legend('Input (Wind Gust Disturbance)');
subplot(2, 1, 2);
plot(t, y_cl(:, 1));
legend('Output (Aircraft Pitch Angle)');
```

OUTPUT:

```
Sol =

  A = 
            x1       x2       x3       x4       x5       x6       x7       x8
   x1        0        0        1        0        0        0        0        0
   x2      1.5     -1.5        0   0.0057      1.5        0        0        0
   x3      -12       12     -0.6  -0.0344      -12        0        0        0
   x4   -0.825     0.29        0   -0.014    -0.29        0        0        0
   x5        0        0        0        0    -0.73    2.829        0        0
   x6        0        0        0        0        0    -1.25        0        0
   x7        0        0        0        0        0        0    -1000        0
   x8        0        0        0        0        0        0        0    -1000

  B = 
            u1       u2       u3       u4       u5
   x1        0        0        0        0        0
   x2        0        0        0     0.16      0.8
   x3        0        0        0      -19       -3
   x4        0        0        0  -0.0115  -0.0087
   x5   0.1149        0        0        0        0
   x6        4        0        0        0        0
   x7        0     1024        0        0        0
   x8        0        0     1024        0        0

  C = 
         x1    x2    x3    x4    x5    x6    x7    x8
   y1     1     0     0     0     0     0     0     0
   y2     0     1     0     0     0     0     0     0
   y3     0     0     0     0     0     0     0     0
   y4     0     0     0     0     0     0     0     0
   y5     1     0     0     0     0     0  -139     0
   y6     0     1     0     0     0     0     0  -139

  D = 
          u1     u2     u3     u4     u5
   y1      0      0      0      0      0
   y2      0      0      0      0      0
   y3      0      0      0   0.01      0
   y4      0      0      0      0   0.01
   y5      0  142.9      0      0      0
   y6      0      0  142.9      0      0

Continuous-time state-space model.
```

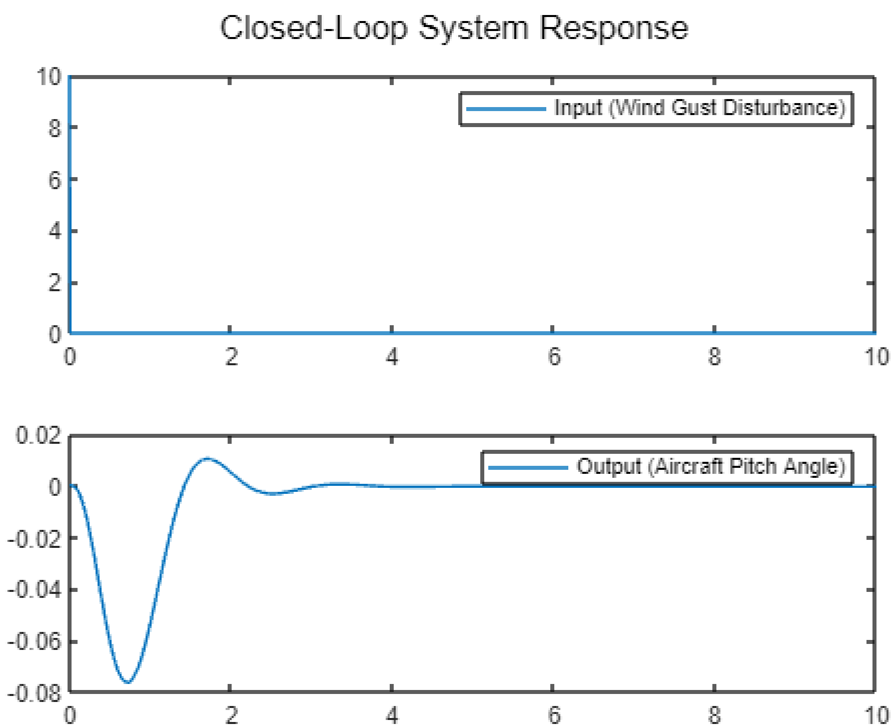



```
Minimization of gamma:

Solver for linear objective minimization under LMI constraints

Iterations   :    Best objective value so far

    1
    2
    3               3650.305369
    4                679.874263
    5                415.718365
    6                196.917091
    7                124.993308
    8                124.993308
    9                 39.263120
   10                 32.197975
   11                 21.790789
   12                 21.790789
   13                  4.993262
   14                  4.993262
   15                  2.508027
   16                  1.610124
   17                  1.610124
   18                  1.224387
   19                  1.224387
   20                  1.224387
   21                  1.003425
   22                  1.003425
   23                  1.003425
   24                  0.941879
```

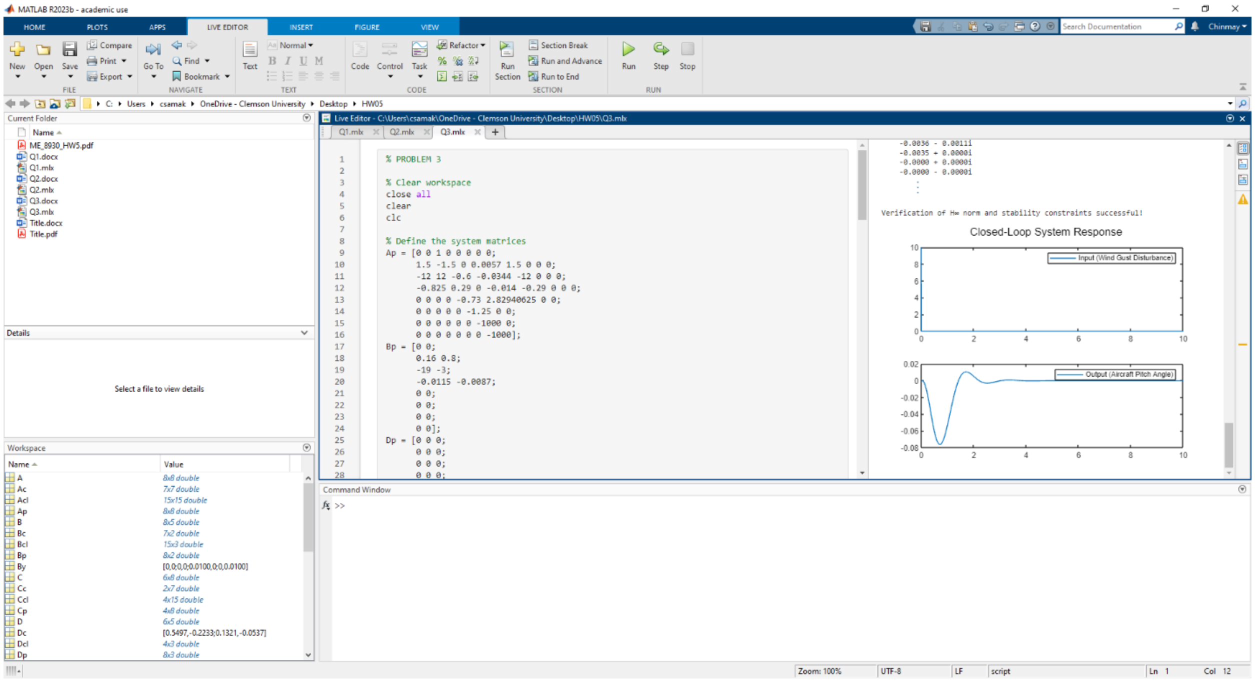
```
    25                   0.941879
    26                   0.941879
    27                   0.941879
    28                   0.912216
    29                   0.912216
    30                   0.912216
    31                   0.896765
    32                   0.896765
    33                   0.896765
***                 new lower bound:     0.820170
    34                   0.896765
    35                   0.887844
    36                   0.887844
    37                   0.887844
***                 new lower bound:     0.867568
    38                   0.884997
    39                   0.883333
* switching to QR
    40                   0.883333
***                 new lower bound:     0.877811

 Result:  feasible solution of required accuracy
          best objective value:     0.883333
          guaranteed absolute accuracy:  5.52e-03
          f-radius saturation:  0.458% of R =  1.00e+08

 Optimal Hinf performance:  8.823e-01
gopt = 0.8823
G = 10×10
   77.0684 -206.4493  -45.4703  -89.5611  -35.5194  -12.4998  106.3172   -
2.3242    3.8637    7.0000
 -220.1100  183.7132    2.3812  116.5260   57.5111    1.5305 -226.7402
4.7605   -1.7312         0
  397.2066 -510.2170  -63.3345 -269.2750 -121.1621  -18.7382  432.1665   -
5.0789   -3.3582         0
    5.7829   -3.5433    1.5761   -4.9621    1.2446    0.7772    7.5619
1.3169    3.6942         0
  293.4949 -341.9114  -33.2695 -188.0955  -88.9407  -10.9885  313.4063   -
1.8555    1.3533         0
   -7.7820    8.1529    0.3504    4.6494    2.2216    0.1100   -8.8267   -
0.2847   -0.0942         0
 -263.0885  307.1698   29.7766  169.0752   79.4155    9.7083 -282.4749
0.6617   -0.6630         0
  -60.3310  101.1714   16.6276   48.7081   21.1076    4.7857  -70.3890
0.5497   -0.2233         0
 -102.4696  -63.6786  -48.4705    1.8408   11.4945  -12.2981  -80.1751
0.1321   -0.0537         0
         0         0         0         0         0         0         0
0         0      -Inf
H∞ controller:
Ac = 7×7
   77.0684 -206.4493  -45.4703  -89.5611  -35.5194  -12.4998  106.3172
 -220.1100  183.7132    2.3812  116.5260   57.5111    1.5305 -226.7402
  397.2066 -510.2170  -63.3345 -269.2750 -121.1621  -18.7382  432.1665
    5.7829   -3.5433    1.5761   -4.9621    1.2446    0.7772    7.5619
  293.4949 -341.9114  -33.2695 -188.0955  -88.9407  -10.9885  313.4063
   -7.7820    8.1529    0.3504    4.6494    2.2216    0.1100   -8.8267
```

```
 -263.0885  307.1698   29.7766  169.0752   79.4155    9.7083 -282.4749
Bc = 7×2
   -2.3242    3.8637
    4.7605   -1.7312
   -5.0789   -3.3582
    1.3169    3.6942
   -1.8555    1.3533
   -0.2847   -0.0942
    0.6617   -0.6630
Cc = 2×7
  -60.3310  101.1714   16.6276   48.7081   21.1076    4.7857  -70.3890
 -102.4696  -63.6786  -48.4705    1.8408   11.4945  -12.2981  -80.1751
Dc = 2×2
    0.5497   -0.2233
    0.1321   -0.0537
Closed-loop system:
Acl = 15×15
10^3 ×
         0         0    0.0010         0         0         0         0
0         0         0         0         0         0         0         0
    0.0017   -0.0016         0    0.0000    0.0015         0   -0.0269
0.0109   -0.0916   -0.0348   -0.0361    0.0093    0.0126   -0.0091   -0.0754
   -0.0228    0.0164   -0.0006   -0.0000   -0.0120         0    1.5070   -
0.6121    1.4537   -1.7312   -0.1705   -0.9310   -0.4355   -0.0540    1.5779
   -0.0008    0.0003         0   -0.0000   -0.0003         0    0.0010   -
0.0004    0.0016   -0.0006    0.0002   -0.0006   -0.0003    0.0001    0.0015
         0         0         0         0   -0.0007    0.0028         0
0         0         0         0         0         0         0         0
         0         0         0         0         0   -0.0013         0
0         0         0         0         0         0         0         0
         0         0         0         0         0         0   -1.0000
0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0   -
1.0000         0         0         0         0         0         0         0
   -0.0023    0.0039         0         0         0         0    0.3231   -
0.5371    0.0771   -0.2064   -0.0455   -0.0896   -0.0355   -0.0125    0.1063
    0.0048   -0.0017         0         0         0         0   -0.6618
0.2407   -0.2201    0.1837    0.0024    0.1165    0.0575    0.0015   -0.2267
Bcl = 15×3
10^3 ×
         0         0         0
         0    0.0277   -0.0112
         0   -1.5486    0.6290
         0   -0.0011    0.0004
    0.0001         0         0
    0.0040         0         0
         0    1.0240         0
         0         0    1.0240
         0   -0.3320    0.5520
         0    0.6801   -0.2473
Ccl = 4×15
    1.0000         0         0         0         0         0         0
0         0         0         0         0         0         0         0
         0    1.0000         0         0         0         0         0
0         0         0         0         0         0         0         0
    0.0055   -0.0022         0         0         0         0   -0.7642
0.3104   -0.6033    1.0117    0.1663    0.4871    0.2111    0.0479   -0.7039
```

```
    0.0013   -0.0005         0         0         0         0   -0.1836
0.0746   -1.0247   -0.6368   -0.4847    0.0184    0.1149   -0.1230   -0.8018
Dcl = 4×3
         0         0         0
         0         0         0
         0    0.7852   -0.3190
         0    0.1887   -0.0766
H∞ norm:
hinf_norm = 0.8813
Closed-loop poles:
eig_Acl = 15×1 complex
10^3 ×
  -0.1048 + 0.0000i
  -0.0311 + 0.0301i
  -0.0311 - 0.0301i
  -0.0016 + 0.0038i
  -0.0016 - 0.0038i
  -0.0036 + 0.0011i
  -0.0036 - 0.0011i
  -0.0035 + 0.0000i
  -0.0000 + 0.0000i
  -0.0000 - 0.0000i
Verification of H∞ norm and stability constraints successful!
```



SCREENSHOT:


MATLAB R2023b - academic use
HOME
PLOTS
APPS
LIVE EDITOR
INSERT
FIGURE
VIEW
Search Documentation
Chinmay
New
Open
Save
Compare
Print
Export
FILE
Go To
Find
Bookmark
NAVIGATE
Text
Normal
TEXT
Code
Control
Task
Refactor
CODE
Run Section
Section Break
Run and Advance
Run to End
SECTION
Run
Step
Stop
RUN
C: ▸ Users ▸ csamak ▸ OneDrive - Clemson University ▸ Desktop ▸ HW05
Current Folder
Name
ME_8930_HW5.pdf
Q1.docx
Q1.mlx
Q2.docx
Q2.mlx
Q3.docx
Q3.mlx
Title.docx
Title.pdf
Details
Select a file to view details
Workspace
Name
Value
A 8x8 double
Ac 7x7 double
Acl 15x15 double
Ap 8x8 double
B 8x5 double
Bc 7x2 double
Bcl 15x3 double
Bp 8x2 double
By [0,0;0,0;0.0100,0;0,0.0100]
C 6x8 double
Cc 2x7 double
Ccl 4x15 double
Cp 4x8 double
D 6x5 double
Dc [0.5497,-0.2233;0.1321,-0.0537]
Dcl 4x3 double
Dp 8x3 double
Live Editor - C:\Users\csamak\OneDrive - Clemson University\Desktop\HW05\Q3.mlx
Q1.mlx
Q2.mlx
Q3.mlx
% PROBLEM 3
% Clear workspace
close all
clear
clc
% Define the system matrices
Ap = [0 0 1 0 0 0 0 0;
1.5 -1.5 0 0.0057 1.5 0 0 0;
-12 12 -0.6 -0.0344 -12 0 0 0;
-0.825 0.29 0 -0.014 -0.29 0 0 0;
0 0 0 0 -0.73 2.82940625 0 0;
0 0 0 0 0 -1.25 0 0;
0 0 0 0 0 0 -1000 0;
0 0 0 0 0 0 0 -1000];
Bp = [0 0;
0.16 0.8;
-19 -3;
-0.0115 -0.0087;
0 0;
0 0;
0 0;
0 0];
Dp = [0 0 0;
0 0 0;
0 0 0;
0 0 0;
-0.0036 - 0.0011i
-0.0035 + 0.0000i
-0.0000 + 0.0000i
-0.0000 - 0.0000i
Verification of H∞ norm and stability constraints successful!
Closed-Loop System Response
Input (Wind Gust Disturbance)
Output (Aircraft Pitch Angle)
Command Window
>>
Zoom: 100%
UTF-8
LF
script
Ln 1
Col 12