Problem 3

# **CODE:**

% PROBLEM 3

% Clear workspace

close all

clear

clc

% 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];

% Lumped system matrices

A = Ap;

B = [Dp Bp];

C = [Cp; Mp];

D = [Dy By; Dz 0]

% State-space system

Sol = ss(A, B, C, D)

% sys\_openloop = ss(Ap, Dp, Cp, 0)

% Define the lumped (disturbance + control) input

t = 0:0.01:10;

w\_amplitude = 0.1;

w\_duration = 1;

w\_pulse = w\_amplitude \* (t >= 0 & t <= w\_duration);

u = zeros(size(w\_pulse));

w = [w\_pulse; w\_pulse; u];

% w = [w\_pulse; w\_pulse];

% Simulate the open-loop system response

[y\_ol, t\_out, x\_ol] = lsim(Sol, w', t);

% [y\_ol, t\_out, x\_ol] = lsim(Sol, w, t);

% Plot the results

figure;

sgtitle('Open-Loop System Response');

subplot(4, 1, 1);

plot(t, w(2, :), t, w(3, :));

legend('W', 'U');

subplot(4, 1, 2);

plot(t, x\_ol(:, 1), t, x\_ol(:, 2));

legend('X1', 'X2');

subplot(4, 1, 3);

plot(t, y\_ol(:, 1), t, y\_ol(:, 2));

legend('Y1', 'Y2');

subplot(4, 1, 4);

plot(t, y\_ol(:, 3));

legend('Z');

% H-infinity LMI

S = ltisys(A, B, C, D);

[gopt, G] = hinflmi(S,[1 1])

% 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);

% Scl = slft(S, G);

% Verification

disp('H∞ norm:')

hinf\_norm = hinfnorm(Scl)

% hinf\_norm = norminf(Scl)

disp('Closed-loop poles:')

eig\_Acl = eig(Acl)

if((abs(gopt-hinf\_norm) <= 0.1) && all(real(eig\_Acl) < 0.0))

disp('Verification of H∞ norm and pole location constraints successful!')

else

disp('Verification of H∞ norm and pole location constraints failed!')

end

% Define the lumped (disturbance + control) input

t = 0:0.01:10;

w\_amplitude = 0.1;

w\_duration = 1;

w\_pulse = w\_amplitude \* (t >= 0 & t <= w\_duration);

u = zeros(size(w\_pulse));

w = [w\_pulse; w\_pulse];

% Simulate the closed-loop system response

[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(2, :));

legend('W');

subplot(3, 1, 2);

plot(t, x\_cl(:, 1), t, x\_cl(:, 2), t, x\_cl(:, 3));

legend('X1', 'X2', 'X3');

subplot(3, 1, 3);

plot(t, y\_cl(:, 1), t, y\_cl(:, 2));

legend('Y1', 'Y2');

# **OUTPUT:**

D = 3×3

0 0 0

0 0 1

0 1 0

Sol =

A =

x1 x2

x1 0 1

x2 0 0

B =

u1 u2 u3

x1 0 0 0

x2 1 0 1

C =

x1 x2

y1 0 1

y2 0 0

y3 1 0

D =

u1 u2 u3

y1 0 0 0

y2 0 0 1

y3 0 1 0

Continuous-time state-space model.

[Model Properties](matlab:disp(char('','%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20A:%20[2×2%20double]%20','%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20B:%20[2×3%20double]%20','%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20C:%20[3×2%20double]%20','%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20D:%20[3×3%20double]%20','%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20E:%20[]%20','%20%20%20%20%20%20%20%20%20%20%20Scaled:%200%20','%20%20%20%20%20%20%20%20StateName:%20%7b2×1%20cell%7d%20','%20%20%20%20%20%20%20%20StatePath:%20%7b2×1%20cell%7d%20','%20%20%20%20%20%20%20%20StateUnit:%20%7b2×1%20cell%7d%20','%20%20%20%20InternalDelay:%20[0×1%20double]%20','%20%20%20%20%20%20%20InputDelay:%20[3×1%20double]%20','%20%20%20%20%20%20OutputDelay:%20[3×1%20double]%20','%20%20%20%20%20%20%20%20InputName:%20%7b3×1%20cell%7d%20','%20%20%20%20%20%20%20%20InputUnit:%20%7b3×1%20cell%7d%20','%20%20%20%20%20%20%20InputGroup:%20[1×1%20struct]%20','%20%20%20%20%20%20%20OutputName:%20%7b3×1%20cell%7d%20','%20%20%20%20%20%20%20OutputUnit:%20%7b3×1%20cell%7d%20','%20%20%20%20%20%20OutputGroup:%20[1×1%20struct]%20','%20%20%20%20%20%20%20%20%20%20%20%20Notes:%20[0×1%20string]%20','%20%20%20%20%20%20%20%20%20UserData:%20[]%20','%20%20%20%20%20%20%20%20%20%20%20%20%20Name:%20''''%20','%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20Ts:%200%20','%20%20%20%20%20%20%20%20%20TimeUnit:%20''seconds''%20','%20%20%20%20%20SamplingGrid:%20[1×1%20struct]%20','%20')))

A diagram of a system response

Description automatically generated

Minimization of gamma:

Solver for linear objective minimization under LMI constraints

Iterations : Best objective value so far

1

2 4.331487

3 2.480378

4 2.094585

5 1.984537

6 1.984537

7 1.891906

8 1.891906

\*\*\* new lower bound: 0.208335

9 1.693528

10 1.693528

\*\*\* new lower bound: 0.773250

11 1.636185

\*\*\* new lower bound: 1.240010

12 1.636185

\*\*\* new lower bound: 1.567929

13 1.622326

14 1.620133

\*\*\* new lower bound: 1.604742

Result: feasible solution of required accuracy

best objective value: 1.620133

guaranteed relative accuracy: 9.50e-03

f-radius saturation: 0.399% of R = 1.00e+08

Optimal Hinf performance: 1.620e+00

gopt = 1.6197

G = 3×3

-2.0572 1.5998 1.0000

2.0785 -1.6165 0

0 0 -Inf

H∞ controller:

Ac = -2.0572

Bc = 1.5998

Cc = 2.0785

Dc = -1.6165

Closed-loop system:

Acl = 3×3

0 1.0000 0

-1.6165 0 2.0785

1.5998 0 -2.0572

Bcl = 3×2

0 0

1.0000 -1.6165

0 1.5998

Ccl = 2×3

0 1.0000 0

-1.6165 0 2.0785

Dcl = 2×2

0 0

0 -1.6165

H∞ norm:

hinf\_norm = 1.6185

Closed-loop poles:

eig\_Acl = 3×1 complex

-1.0285 + 0.7471i

-1.0285 - 0.7471i

-0.0002 + 0.0000i

Verification of H∞ norm and pole location constraints successful!

A diagram of a closed loop system

Description automatically generated

# **SCREENSHOT:**

