1: 10 points

Model Order Reduction

Note that a time delay can be approximated as

$$e^{-\tau s} \approx \left(\frac{1 - \frac{\tau}{2n}s}{1 + \frac{\tau}{2n}s}\right)^n$$

for a sufficiently large n. Let a process model $\frac{e^{-s}}{1+Ts}$ be approximated by

$$G(s) = \left(\frac{1 - 0.05s}{1 + 0.05s}\right)^{10} \frac{1}{1 + sT}.$$

For each $T = 0,\ 0.01,\ 0.1,\ 1,\ 10$, find the smallest order model using balanced truncation such that the approximation error is no greater than 0.1. Use Matlab to solve.

```
%Problem 1
s = tf('s');
T = 0;
G = ((1 - 0.05*s)/(1 + 0.05*s))^(10) * (1/(1 + s * T));
Gr = balancmr(G,'MaxError', 0.1);
modelReducer(G)
size(Gr)
```

Image 1

Smallest order: 10

State-space model with 1 outputs, 1 inputs, and 10 states.

```
s = tf('s');
T = 0.01;
G = ((1 - 0.05*s)/(1 + 0.05*s))^(10) * (1/(1 + s * T));
Gr = balancmr(G,'MaxError', 0.1);
modelReducer(G)
size(Gr)
```

Image 2

smallest order: 11

State-space model with 1 outputs, 1 inputs, and 11 states.

```
s = tf('s');
T = 0.1;
G = ((1 - 0.05*s)/(1 + 0.05*s))^(10) * (1/(1 + s * T));
Gr = balancmr(G,'MaxError', 0.1);
modelReducer(G)
size(Gr)
```

Image 3

smallest order: 10

State-space model with 1 outputs, 1 inputs, and 10 states.

```
s = tf('s');
T = 1;
G = ((1 - 0.05*s)/(1 + 0.05*s))^(10) * (1/(1 + s * T));
Gr = balancmr(G,'MaxError', 0.1);
modelReducer(G)
size(Gr)
```

Image 4

smallest order: 8

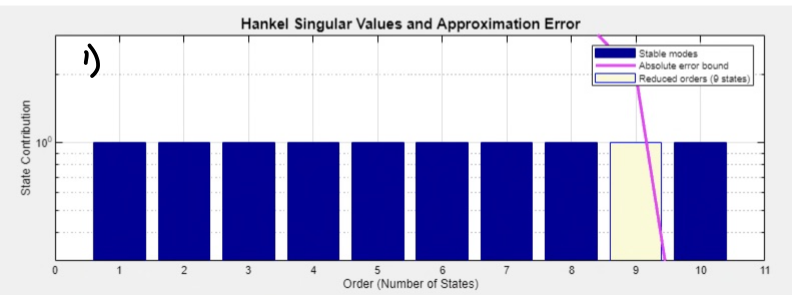
State-space model with 1 outputs, 1 inputs, and 8 states.

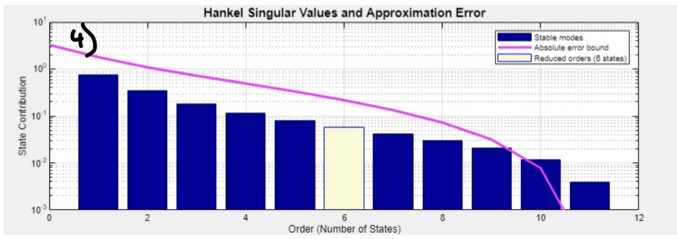
```
s = tf('s');
T = 10;
G = ((1 - 0.05*s)/(1 + 0.05*s))^(10) * (1/(1 + s * T));
Gr = balancmr(G,'MaxError', 0.1);
modelReducer(G)
size(Gr)
```

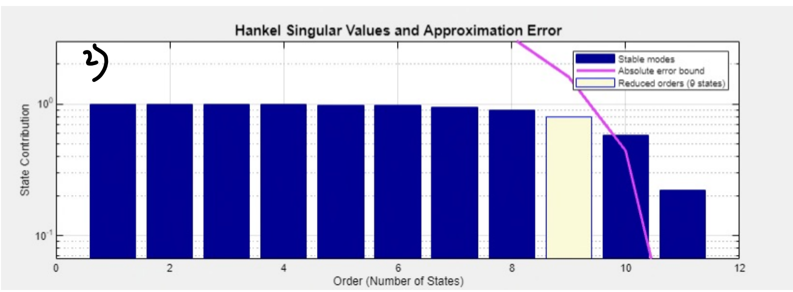
Image 5

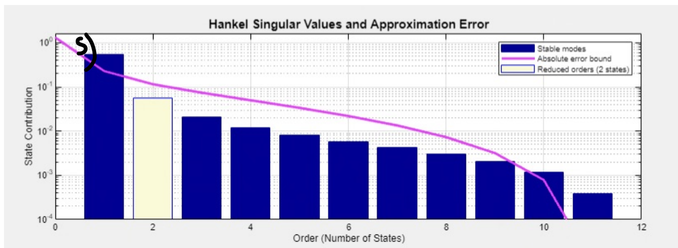
smallest order: 3

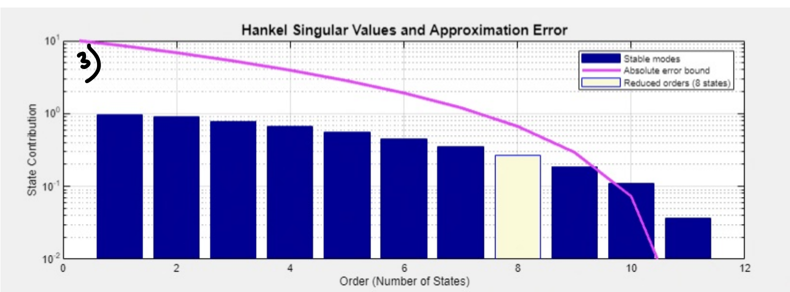
State-space model with 1 outputs, 1 inputs, and 3 states.











2: 30 points

LMI Problems

(a) Consider the system

$$A = \begin{bmatrix} -5 & 1 & 2 \\ 1 & -9 & 1 \\ -1 & -10 & -3 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Compute the H_2 norm of the system using LMIs in Matlab.

(b) Consider the system

$$A = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & -2 & 10 \\ 0 & 1 & -2 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0 \\ -2 & 2 \\ 0 & 1 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Determine whether this system is stabilizable and detectable using LMIs in Matlab.

(c) Consider the system of form

$$\dot{x} = Ax + B_1 w + B_2 u$$

ghin

$$z = Cx + Du$$

viego

zac

rummus

with

$$A = \begin{bmatrix} -5 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad B_1 = \begin{bmatrix} 0.05 \\ 0 \\ 0.03 \end{bmatrix} \quad B_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 1 \end{bmatrix} \quad D = \begin{bmatrix} 1 & 3 \\ 1 & 0 \end{bmatrix}.$$

Design a state feedback control law $u = Kx$ that solves the H_2 optimal control problem using LMIs in Matlab.

a)

```
%Part A
A = [-5 1 2; 1 -9 1; -1 -10 -3];
B = [0; 1; 0];
C = [1 0 0; 0 0 1];
D = [0; 0];

G = ss(A, B, C, D);
h_2_norm = h2Norm(G)
```

```
function norm = h2Norm(G)
    [A, B, C, D] = ssdata(G);

    gam = sdpvar(1);
    X = sdpvar(size(A, 1));

    eq1 = A*X + X*A' + B*B';
    eq2 = trace(C*X*C');

    limit = 1e-10;
    F = [X>=limit*eye(size(X)), eq1 <= -limit, eq2 <= gam];
    optimize(F,gam);

    norm = sqrt(value(gam));
end
```

```
SeDuMi 1.3.7 by AdvOL, 2005-2008 and Jos F. Sturm, 1998-2003.
Alg = 2: xz-corrector, theta = 0.250, beta = 0.500
eqs m = 7, order n = 8, dim = 20, blocks = 3
nnz(A) = 33 + 0, nnz(ADA) = 41, nnz(L) = 24
 it :     b*y       gap    delta  rate   t/tP*  t/tD*   feas cg cg  prec
  0 :            5.78E+01 0.000
  1 :  -3.57E-01 1.36E+01 0.000 0.2361 0.9000 0.9000   1.78  1  1  6.3E+00
  2 :  -1.30E-01 4.02E+00 0.000 0.2947 0.9000 0.9000   2.35  1  1  1.0E+00
  3 :  -1.14E-01 8.46E-01 0.000 0.2105 0.9000 0.9000   1.64  1  1  1.6E-01
  4 :  -1.12E-01 6.64E-02 0.000 0.0785 0.9900 0.9900   1.25  1  1  1.1E-02
  5 :  -1.12E-01 2.74E-03 0.000 0.0413 0.9900 0.9900   1.09  1  1  4.5E-04
  6 :  -1.12E-01 2.83E-05 0.000 0.0103 0.9990 0.9990   1.00  1  1  4.6E-06
  7 :  -1.12E-01 2.89E-07 0.000 0.0102 0.9990 0.9990   1.00  1  1  4.7E-08
  8 :  -1.12E-01 2.94E-09 0.000 0.0102 0.9990 0.9990   1.00  1  1  4.8E-10

iter seconds digits       c*x               b*y
  8      0.2  10.0 -1.1187117381e-01 -1.1187117383e-01
|Ax-b| =   6.5e-10, [Ay-c]_+ =   1.5E-11, |x|=  1.0e+00, |y|=  1.7e-01

Detailed timing (sec)
   Pre          IPM          Post
4.220E-01    5.050E-01    1.300E-02
Max-norms: ||b||=1, ||c|| = 1.000000e+00,
Cholesky |add|=0, |skip| = 0, ||L.L|| = 1.

h_2_norm = 0.3345
```

b)

```
A = [0.5 0 0; 0 -2 10; 0 1 -2];
B = [1 0;-2 2;0 1];
C = [1 0 0;0 0 1];
D = [0 0;0 0];
[stability, detectability] = LMIStable(ss(A,B,C,D))
```

```
function [stable, detect] = LMIStable(sys)
    [A, B, C, D] = ssdata(ss(sys));
    gam = sdpvar(1);
    P = sdpvar(size(A,1));

    limit = 1e-10;
    F = [P >= limit*eye(size(P)), A'*P + P*A <= gam*B*B', gam >= limit];

    optimize(F);

    isStable = check(F);
    if(min(isStable) < 0)
        stable = 0;
    else
        stable = 1;
    end

    w = sdpvar(size(B, 2), size(A, 1), 'full');

    F = [P >=  limit*eye(size(P)), A'*P + P*A + w'*C + C'*w <= limit];
    optimize(F)

    isStable = check(F);
    if(min(isStable) < 0)
        detect = 0;
    else
        detect = 1;
    end
end
```

```
The coefficient matrix is not full row rank, numerical problems may occur.
SeDuMi 1.3.7 by AdvOL, 2005-2008 and Jos F. Sturm, 1998-2003.
Alg = 2: xz-corrector, theta = 0.250, beta = 0.500
eqs m = 12, order n = 7, dim = 19, blocks = 3
nnz(A) = 24 + 0, nnz(ADA) = 144, nnz(L) = 78
 it :     b*y       gap    delta  rate   t/tP*  t/tD*   feas cg cg  prec
  0 :            3.16E+02 0.000
  1 :  0.00E+00 3.11E+01 0.000 0.0987 0.9900 0.9900   1.00  1  1  4.5E+00
  2 :  0.00E+00 7.34E-01 0.000 0.0236 0.9900 0.9900   1.00  1  1  1.1E-01
  3 :  0.00E+00 3.32E-03 0.000 0.0045 0.9990 0.9990   1.00  1  1  4.8E-04
  4 :  0.00E+00 1.49E-05 0.000 0.0045 0.9990 0.9990   1.00  1  1  2.2E-06
  5 :  0.00E+00 6.66E-08 0.000 0.0045 0.9990 0.9990   1.00  1  1  9.7E-09
  6 :  0.00E+00 2.99E-10 0.000 0.0045 0.9990 0.9990   1.00  1  1  4.4E-11

iter seconds digits       c*x               b*y
  6      0.0   Inf -2.8154776391e-22  0.0000000000e+00
|Ax-b| =   2.2e-11, [Ay-c]_+ =   0.0E+00, |x|=  3.8e-12, |y|=  5.3e+00

Detailed timing (sec)
   Pre          IPM          Post
6.799E-02    2.500E-02    2.002E-03
Max-norms: ||b||=0, ||c|| = 2.000000e-10,
Cholesky |add|=0, |skip| = 1, ||L.L|| = 1.57228.
ans = struct with fields:
     yalmipversion: '20210331'
     matlabversion: '9.13.0.2126072 (R2022b) Update 3'
        yalmiptime: 0.0517
        solvertime: 0.0963
              info: 'Successfully solved (SeDuMi-1.3)'
           problem: 0
stability = 1
detectability = 1
```

c)

```
A = [-5 1 0; 0 1 1; 1 1 1];
B1 = [0.05 0 0.03]';
B2 = [0 0; 0 1; 1 0];
C = [1 0 0; 0 2 1];
D = [1 3; 1 0];
[K, ~] = h2Optimal(A, B2, B1, C, D)
```

```
function [K, gam] = h2Optimal(A, B1, B2, C, D)
    z = sdpvar(size(C*A, 1));
    w = sdpvar(size(B1, 2), size(A, 1), 'full');
    X = sdpvar(size(A, 1));

    gam = sdpvar(1);
    limit = 1e-10;

    eq1 = A*X + B1*w + (A*X + B1*w)' + B2*B2';
    eq2 = [-z C*X + D*w; (C*X + D*w)' -X];

    F = [eq1 <= limit, eq2 <= limit, trace(z)<= gam];
    optimize(F, gam);

    gam = sqrt(value(gam));
    K = value(w)*inv(value(X));
end
```

```
SeDuMi 1.3.7 by AdvOL, 2005-2008 and Jos F. Sturm, 1998-2003.
Alg = 2: xz-corrector, theta = 0.250, beta = 0.500
eqs m = 16, order n = 10, dim = 36, blocks = 3
nnz(A) = 54 + 0, nnz(ADA) = 230, nnz(L) = 123
 it :     b*y       gap    delta  rate   t/tP*  t/tD*   feas cg cg  prec
  0 :            3.37E+01 0.000
  1 :  -7.19E-02 6.21E+00 0.000 0.1841 0.9000 0.9000   1.86  1  1  3.5E+00
  2 :  -9.27E-03 6.18E-01 0.000 0.0995 0.9900 0.9900   1.42  1  1  6.9E-01
  3 :  -1.55E-03 4.85E-02 0.000 0.0786 0.9900 0.9900   1.06  1  1  1.9E-01
  4 :  -2.00E-03 1.83E-02 0.000 0.3764 0.9000 0.9000   0.64  1  1  3.9E-02
  5 :  -3.51E-03 6.21E-03 0.000 0.3396 0.9000 0.9000   0.12  1  1  5.4E-03
  6 :  -4.63E-03 1.57E-03 0.000 0.2525 0.9000 0.9000   0.22  1  1  2.0E-03
  7 :  -5.58E-03 4.30E-04 0.000 0.2747 0.9000 0.9000   0.41  1  1  7.6E-04
  8 :  -6.00E-03 1.13E-04 0.000 0.2619 0.9000 0.9000   0.73  1  1  2.2E-04
  9 :  -6.12E-03 2.30E-05 0.000 0.2040 0.9000 0.9000   0.94  1  1  4.6E-05
 10 :  -6.14E-03 4.81E-06 0.000 0.2091 0.9000 0.9000   0.99  1  2  9.7E-06
 11 :  -6.15E-03 1.02E-06 0.000 0.2127 0.9000 0.9000   1.00  1  2  2.1E-06
 12 :  -6.15E-03 2.14E-07 0.000 0.2089 0.9000 0.9000   1.00  2  3  4.3E-07
 13 :  -6.15E-03 4.32E-08 0.000 0.2020 0.9000 0.9000   1.00  3  6  8.7E-08
 14 :  -6.15E-03 8.46E-09 0.000 0.1960 0.9000 0.9000   1.00  5  8  1.7E-08
 15 :  -6.15E-03 4.57E-09 0.000 0.5400 0.9000 0.9000   1.00  9 10  9.2E-09
 16 :  -6.15E-03 1.17E-09 0.000 0.2558 0.9000 0.9000   1.00  9  6  2.4E-09
Run into numerical problems.

iter seconds digits       c*x               b*y
 16      0.1   Inf -6.1492510559e-03 -6.1492505812e-03
|Ax-b| =   3.8e-09, [Ay-c]_+ =   7.6E-11, |x|=  3.1e+01, |y|=  5.2e-01

Detailed timing (sec)
   Pre          IPM          Post
6.991E-03    8.800E-02    4.003E-03
Max-norms: ||b||=1, ||c|| = 3.000000e-03,
Cholesky |add|=0, |skip| = 1, ||L.L|| = 46.2887.
K = 2x3
   -0.3606   -4.8935   -4.3189
   -0.2516    0.8286    0.8506
```

3: 30 points

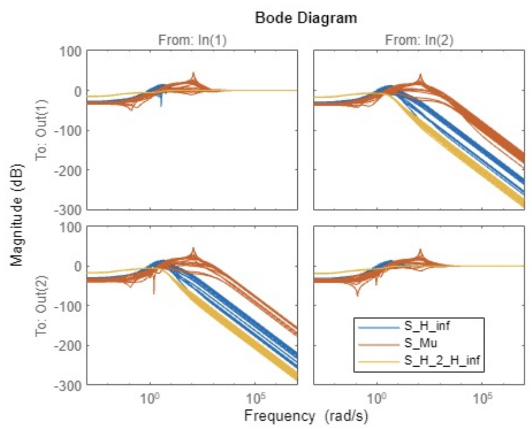
Design Problem: LMI Synthesis

Consider the distillation column with plant model

$$G(s) = \frac{1}{75s+1}\begin{bmatrix} 87.8 & -86.4 \\ 108.2 & -109.6 \end{bmatrix}.$$

Assuming multiplicative input uncertainty of the form $W_I(s) = \frac{10s+10}{s+100} I_{2\times 2}$ and performance weight $W_P(s) = \frac{0.5s+10}{s+0.1} I_{2\times 2}$, design a 1.) an H_∞ optimal controller, 2.) a μ-synthesis controller, and 3.) a mixed H_2 / H_∞ controller that minimizes the H_2 norm such that robust stability is maintained. Plot the uncertain sensitivity functions that are obtained for all three design approaches. Which approach works "best"?

```
%Problem 3
s = tf("s");
G = (1/(75*s + 1)) * [87.8 -86.4; 108.2 -109.6];
W_i = (10*s + 10) / (s + 100) * eye(2);
W_p = (((0.5*s + 10) / (s + 0.1)) / (s/1000+1)) * eye(2);

systemnames = 'G W_i W_p';
inputvar = '[du{2}; r{2}; u{2}]';
outputvar = '[W_i; W_p; r-G]';
input_to_G = '[u + du]';
input_to_W_p = '[r - G]';
input_to_W_i = '[u]';
cleanupsysic = 'yes';
P = sysic;

%H_inf
[K_hinf, ~, GAM] = hinfsyn(P, 2, 2);
fprintf('H_inf Optimal Controller Gamma: %0.2f', GAM);

%Mu
del_1 = ultidyn('del_1',[1, 1]);
del_2 = ultidyn('del_2',[1, 1]);
G_h = G *(eye(2) + [del_1 0; 0 del_2] * W_i);
systemnames = 'G_h W_p';
inputvar = '[r{2}; u{2}]';
outputvar = '[W_p; G_h;r - G_h]';
input_to_G_h = '[u]';
input_to_W_p = '[r-G_h]';
cleanupsysic = 'yes';
P_mu = sysic;
fprintf('Mu Synthesis Performance');
[K_mu, ~, GAM] = musyn(P_mu, 2, 2);

%H_2_H_inf
[K_h2_H_inf, ~, GAM, ~] = h2hinfsyn(prescale(ss(P)),2, 2, 2, [0,1], 'HINFMAX', 1);
fprintf('H_2_H_inf Controller Gamma: %0.2f\n', GAM);

bodemag(inv(eye(2) + G_h * K_hinf), inv(eye(2) + G_h * K_mu), inv(eye(2) + G_h * K_h2_H_inf));
legend("S_H_inf","S_Mu","S_H_2_H_inf", Location="best");
```




```
H_inf Optimal Controller Gamma: 10.89
Mu Synthesis Performance
D-K ITERATION SUMMARY:
-----------------------------------------------------------------
                      Robust performance              Fit order
-----------------------------------------------------------------
  Iter        K Step      Peak MU      D Fit              D
    1         10.84       10.84       10.94              20
    2         7.997       7.997           8              12
    3         6.091       6.091       6.104              12
    4         5.231        5.23        5.24              12
    5         4.841       4.841       4.846              12
    6         4.618       4.618       4.633              16
    7         4.493       4.494       4.515              16
    8         4.427       4.427       4.455              16
    9         4.375       4.376       4.402              16
   10         4.356       4.356       4.386              16

Best achieved robust performance: 4.36
H_2_H_inf Controller Gamma: 1.00
H_2_H_inf Controller Gamma: 25.18
```

best approach: musyn