Class: Robust Multivariate Control
Professor: Dr. Sean Humbert
TAs: Santosh Chaganti
Student: Steve Gillet
Date: February 21, 2025
Assignment: Homework 3

1. D-Stability of a System

"In lecture it was shown that the LMI to verify stability of the system $\dot{x} = Ax$ is given by:

$$\begin{cases} P > 0 \\ A^T P + PA < 0, \end{cases}$$

which verifies that the eigenvalues of A are in the left half plane. Modifications can be made to this LMI that provide conditions to verify the eigenvalues of A are in a subset $\mathbb{D}$ of the left half plane, which is called $\mathbb{D}$-stability. The simplest case is to verify $\lambda_i(A)$ are in region that lies to the left of some value of α in the left half plane. In this case the resulting LMI conditions are:

$$\begin{cases} P > 0 \\ A^T P + PA + 2\alpha P < 0. \end{cases}$$

Code up this new LMI as a ***feasp*** *problem using the MATLAB LMI Toolbox and use it to estimate the largest region of $\mathbb{D}$-stability for the following systems $\dot{x} = Ax$ where*

$$A = \begin{pmatrix} 0 & 1 \\ -1 & -2 \end{pmatrix}, \quad A = \begin{pmatrix} -1.5 & 1 & 0.1 \\ -4 & -1 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

"

I used the following code to set up the feasability problem.

```
A = [-1.5 1 0.1; -4 -1 0; 0 1 0];
% A = [0 1; -1 -2];
nStates = size(A, 1);
disp(eig(A));

```

```
setlmis([])

P = lmivar(1, [nStates 1]);
alpha = 0.05;

lmiterm([-1 1 1 P],1,1);
lmiterm([2 1 1 P],1,A,'s');
lmiterm([2 1 1 P],2*alpha,1);

lmiDstab = getlmis;

[tmin, pfeas] = feasp(lmiDstab);
P = dec2mat(lmiDstab,pfeas,P);
disp(P);
```

I set alpha to .99 here. I confirmed alpha by printing out the Eigenvalues of A using the 'eig' function and then kind of played around with different values of alpha around the Eigenvalues. The output for the first A matrix is:

Eigenvalues: -1 -1
Solver for LMI feasibility problems $L(x) < R(x)$
This solver minimizes t subject to $L(x) < R(x) + t * I$
The best value of t should be negative for feasibility
Iteration : Best value of t so far
1 0.067076
2 0.010110
3 0.010110
4 2.180764e-03
5 1.491047e-03
6 1.491047e-03
7 3.249250e-04
8 -9.440735e-04
Result: best value of t: -9.440735e-04
f-radius saturation: 0.000% of R = 1.00e+09
1.0e+03 *

The output for the second A matrix with alpha set to 0.05 is:
Eigenvalues:
-1.2124 + 1.9616i
-1.2124 - 1.9616i
-0.0752 + 0.0000i
Solver for LMI feasibility problems $L(x) < R(x)$
This solver minimizes t subject to $L(x) < R(x) + t * I$
The best value of t should be negative for feasibility
Iteration : Best value of t so far
1 0.060635
2 -0.027178
Result: best value of t: -0.027178

f-radius saturation: 0.000% of R = 1.00e+09

2. Spectral Norm of a Matrix

"In lecture we derived the following LMI representing the inequality $\|A\|_2 = \sigma(A) < \gamma$,

$$\begin{pmatrix} \gamma^2 I & A^T \\ A & I \end{pmatrix} > 0.$$

Code up this LMI as a ***mincx*** *problem using the MATLAB LMI Toolbox. Assume the minimization variable is* $\rho = \gamma^2$*:*

$$\begin{cases} \min \rho \\ \begin{pmatrix} \rho I & A^T \\ A & I \end{pmatrix} > 0. \end{cases}$$

For the following matrices, use the code above to estimate $\sigma(A)$*. Remember to back solve for* $\gamma = \sqrt{\rho}$*.*

$$A_1 = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}, \quad A_2 = \begin{pmatrix} -1 & 1 & 0 \\ 1 & -1 & 1 \\ 0 & 1 & 1 \end{pmatrix}.$$

"

I used the following Matlab code:

```
% A = [2 -1; -1 2];
A = [-1 1 0; 1 -1 1; 0 1 1];
disp(svd(A));

nStates = size(A,1);

setlmis([])

rho = lmivar(1, [1 0]);

lmiterm([-1 1 1 rho],1,1);
lmiterm([-1 1 2 0],A');
lmiterm([-1 2 2 0],1);

spectralLmi = getlmis;

c = mat2dec(spectralLmi,rho);

[copt, xopt] = mincx(spectralLmi, c);
rhoVal = dec2mat(spectralLmi, xopt, 1);
gamma = sqrt(rhoVal);
disp(gamma);
```

I print out the singular values of the A matrix first so I can confirm my answers. Output for first A matrix:

Singular Values:
3.0000
1.0000
Solver for linear objective minimization under LMI constraints
Iterations : Best objective value so far
1
*** new lower bound: 0.472136
2 9.346101
*** new lower bound: 2.184006
3 9.132359
*** new lower bound: 6.560658
4 9.014443
*** new lower bound: 7.678249
5 9.014443
*** new lower bound: 8.294599
6 9.014443
*** new lower bound: 8.754266
7 9.000200
*** new lower bound: 8.914698
Result: feasible solution of required accuracy
best objective value: 9.000200
guaranteed relative accuracy: 9.50e-03
f-radius saturation: 0.000% of R = 1.00e+09
Gamma:
3.0000

Second A matrix output:
Singular Values:
2.1701
1.4812
0.3111
Solver for linear objective minimization under LMI constraints
Iterations : Best objective value so far
1
*** new lower bound: -0.208712
2 4.883681
*** new lower bound: 2.636568
3 4.730055
4 4.730055
*** new lower bound: 3.484965
5 4.730055
*** new lower bound: 4.359399
6 4.713832
*** new lower bound: 4.613672
7 4.713832

*** new lower bound: 4.693051
Result: feasible solution of required accuracy
best objective value: 4.713832
guaranteed relative accuracy: 4.41e-03
f-radius saturation: 0.000% of R = 1.00e+09
Gamma:
2.1711

3. H_∞ Norm of a System

"In this problem, you will use an alternate LMI form for the Bounded Real Lemma and implement it as a generalized eigenvalue problem (gevp) using the MATLAB LMI Toolbox. If we start with part (c) from the Bounded Real Lemma theorem from lecture and move the γ^2 term to the right side, we have

$$\begin{pmatrix} A^TP + PA + C^TC & PB + C^TD \\ B^TP + D^TC & D^TD \end{pmatrix} < \gamma^2 \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

In this form with P as the matrix variable, it can be cast as a generalized eigenvalue problem

$$\begin{cases} \min \lambda \\ A(x) < \lambda B(x) \\ B(x) > 0 \\ C(x) > 0, \end{cases}$$

where $\lambda = \gamma^2$, $A(x)$ is the left hand side of the above expression, $C(x) = P$, and

$$B(x) = \begin{pmatrix} \epsilon & 0 \\ 0 & 1 \end{pmatrix},$$

Here we need to add a small number $\epsilon \approx 0.00001$ to the matrix so that it is positive definite for numerical stability.

(a) Implement the above LMI as a function that accepts matrices A, B, C, and D from a general state space representation $\dot{x} = Ax + Bu$, $y = Cx + Du$ of a system G and uses the function `gevp` *to compute the infinity norm for the system, $\|G\|_\infty$. Be sure to back solve for $\gamma = \sqrt{\lambda}$ in your function.*

(b) Use your code from (a) to compute $\|G\|_\infty$ for the following stable (from Problem 1) dynamical systems. You can check your results using the MATLAB function `hinfsyn(sys)` *where* `sys` *is a state space object constructed using the command* `ss`*.*

$$\dot{x} = \begin{pmatrix} 0 & 1 \\ -1 & -2 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u, \quad \dot{x} = \begin{pmatrix} -1.5 & 1 & 0.1 \\ -4 & -1 & 0 \\ 1 & 0 & 0 \end{pmatrix} x + \begin{pmatrix} -0.2 \\ -1.8 \\ 0 \end{pmatrix} u$$

$$y = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} x, \qquad y = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} x + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} u$$

”

This is how I set the problem up in Matlab:

```
A = [-1.5 1 0.1; -4 -1 0; 1 0 0];
B = [-0.2; -1.8; 0];
C = [1 0 0; 0 1 0; 0 0 1];
D = [1; 0; 0];

% A = [0 1; -1 -2];
% B = [0; 1];
% C = [1 0; 0 2];
% D = [0];

nStates = size(A,1);
nOutputs = size(C,1);
nInputs = size(B,1);

setlmis([]);

P = lmivar(1, [nStates 1]);

epsilon = 1e-5;

lmiterm([-1 1 1 P], 1, 1);

lmiterm([-2 1 1 0], epsilon);
lmiterm([-2 2 2 0], 1);

lmiterm([3 1 1 P], A', 1, 's');
lmiterm([3 1 1 0], C'*C);
lmiterm([3 1 2 P], 1, B);
lmiterm([3 1 2 0], C'*D);
lmiterm([3 2 2 0], D'*D);
lmiterm([-3 1 1 0], epsilon);
lmiterm([-3 2 2 0], 1);

lmisys = getlmis;

[aa, xopt] = gevp(lmisys, 1);

disp(sqrt(aa));

sys = ss(A,B,C,D);
disp(hinfnorm(sys));
```

I put the system into a state space model using the 'ss' Matlab function and then print out the 'hinfnorm' results to confirm my answers at the end.

And this was the result for the first system:

Solver for generalized eigenvalue minimization
Iterations : Best objective value so far
1 309.375000
2 146.808105
3 101.297593
4 69.895339
5 48.227784
6 33.277171
7 22.961248
8 15.843261
9 10.931850
10 7.542977
11 5.204654
12 2.246454
13 2.246454
14 2.246454
15 2.223989
16 2.201749
17 1.498196
*** new lower bound: -0.333358
18 1.440960
*** new lower bound: -0.276122
19 1.440960
20 1.387301
*** new lower bound: 0.608383
21 1.387301
22 1.343377
*** new lower bound: 1.005225
23 1.338094
24 1.338094
*** new lower bound: 1.171659
25 1.335493
*** new lower bound: 1.297743
26 1.333724
*** new lower bound: 1.298904
27 1.333724
Result: feasible solution
best value of t: 1.333724
guaranteed relative accuracy: 2.61e-02
f-radius saturation: 0.000% of R = 1.00e+08
Termination due to SLOW PROGRESS:
the gen. eigenvalue t decreased by less than

1.000% during the last 5 iterations.
Gamma: 1.1549
'hinfnorm' 1.1547

And this was the result for the second system:
Solver for generalized eigenvalue minimization
Iterations : Best objective value so far
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
* switching to QR
* upper bound on optimal value set to 1.00e+08
26
* upper bound on optimal value set to 1.00e+11
27
28
Result: infeasibility
Gamma:
'hinfnorm':
Inf