MAE 598 11/13/2020

Mid-Term Examination

Due at 6:00PM Sunday 11/15 on Gradescope

Instructions: Absolutely no collaboration of any kind is allowed during the period of the exam. You may not talk to or communicate with your classmates during the exam in any way. You may not seek or receive assistance with the exam from anyone. Violators will receive an E for MAE598. Books, notes, Matlab, lectures are all open. Use of the internet is not allowed, with the exception of Blackboard. Use a new page for each problem. Return the exam in person. Attach a printout of the exam questions to the front.

Written Copy and Code Submission: For each problem, show all work. If there is a final answer, circle it. A paper version of the homework must also be submitted and this should include a printout of your final answer. A printout of the code is not necessary. For computational problems, email your code to `mpeet@asu.edu`. The subject line should be precisely: `MAE598 CODE: MIDTERM, [LAST NAME]`, where you replace [LAST NAME] with your last name. For each Problem requiring a code, include a .m file entitled `MAE598_MIDTERM_PY_LASTNAME.m`, where you should replace Y with the problem number and LASTNAME with your last name. You may use any of the standard YALMIP solvers, including CPLEX and MOSEK. However, use sdpsettings to choose your solver. Make sure your code executes.

1. **[Decentralized Feedback] (25pts)**

   - In large-scale systems such as national economies or chemical refineries, it is often the case that an actuator must act on strictly local information. In this case, the standard controller , say $u = Kx(t)$ can be broken down into sub-controllers $u_i = K_i x_i(t)$, where the u_i, x_i may be scalar or vector subsets of u and x. Consider the system

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 1 \\ -1 & 0 & -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 & 1 & 1 \\ -1 & 1 & -1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 1 & -1 & -1 \\ 0 & -1 & 0 & 0 & -1 & 0 \end{bmatrix} B_1 = \begin{bmatrix} 0 & -1 & -1 \\ 0 & 0 & 0 \\ -1 & -1 & 1 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 1 & 1 \end{bmatrix}, B_2 = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ -1 & 1 & 0 \\ 1 & -1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 1 \end{bmatrix},$$

$$D_{12} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} C_1 = \begin{bmatrix} 0 & 1 & 0 & -1 & -1 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 \end{bmatrix} D_{11} = \begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

     Where we want a controller of the following form:

$$K = \begin{bmatrix} k_1 & k_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_3 & k_4 & k_5 & k_6 \\ 0 & 0 & k_7 & k_8 & k_9 & k_{10} \end{bmatrix}$$

     [i] For dynamic system given above, ignore the constraints on the controller and design the standard H_∞-optimal full-state-feedback controller and determine the H_∞ norm of the closed-loop system.

     [ii] Now we would like to solve the H_∞-optimal full-state-feedback problem using a controller $u = Kx$ where u_1 only depend on states 1-2, while u_2 and u_3 only depends on states 3-6. Reformulate the full-state-feedback LMI to solve this problem by constraining both the Lyapunov variable $X > 0$ and the variable Z to be block-diagonal. What are the constraints on Z and how do these correspond to constraints on K? Compare optimal gains in both the structured and unstructured cases.

2. **[Switched Systems] (25pts)** For this problem, use the following system matrices

$$A = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 \\ -1 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & 1 & -1 & 1 & 1 \\ -1 & 1 & -1 & -1 & 0 & 0 \\ -1 & -1 & 1 & 1 & 1 & -1 \\ 0 & -1 & 0 & 0 & -1 & -1 \end{bmatrix}, B_1 = \begin{bmatrix} 0 & -1 & -1 \\ 0 & 0 & 0 \\ -1 & 1 & 1 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \\ -1 & 1 & 1 \end{bmatrix},$$

$$D_{12} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ .1 & .2 & .4 \end{bmatrix} C_1 = \begin{bmatrix} 0 & 1 & 0 & -1 & -1 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 \end{bmatrix} D_{11} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Unfortunately, the actuator is malfunctioning and hence B_2 keeps switching back and forth between

$$B_2 = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ -1 & 1 & 0 \\ 1 & -1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 1 \end{bmatrix} \quad \text{and} \quad B_2 = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ -1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & -3 & -1 \end{bmatrix}.$$

[i] Design an LMI which yields a full-state feedback stabilizing controller which works for both input models while minimizing and H_∞-bound which is valid despite the switching back and forth. Provide me with both the controller and the closed-loop H_∞ gain.

[ii] For comparison, provide the H_∞ gain achievable using only the first input model (the one on the left).

3. **[The Tracking Problem] (35 pts)** For this Problem, use the following plant (P) state-space description:

$$A = \begin{bmatrix} -1 & 1 & 0 & 1 & 0 & 1 \\ -1 & -2 & -1 & 0 & 0 & 1 \\ 1 & 0 & -2 & -1 & 1 & 1 \\ -1 & -1 & 1 & -3 & 0 & 0 \\ -1 & 1 & -1 & 1 & -2 & -1 \\ 0 & -1 & 0 & 0 & -1 & -2 \end{bmatrix} B = \begin{bmatrix} 0 & -1 & -1 \\ 0 & 0 & 0 \\ -1 & 1 & 1 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \\ -1 & 1 & 1 \end{bmatrix} C = \begin{bmatrix} 0 & 1 & 0 & -1 & -1 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 \end{bmatrix} D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

- Using the Tracking Framework, construct an optimal control framework 9-matrix representation. Compute the open-loop (set $K = 0$) H_2 and H_∞ norms of the system (if finite).
- Use an LMI to formulate and solve the H_∞-optimal output-feedback problem. What is the predicted H_∞ gain?
- Reconstruct the controller using $Y_2 = 2 \cdot I$.
- Construct and write down the closed-loop system matrices and use the Matlab `norm` command to verify the closed-loop system has the desired H_∞ gain.
- Construct the H_∞-optimal full state feedback controller for this system. What is the predicted H_∞-gain?
- Using the LMI on page 86 in Caverly (295 in Duan and Li, but this uses different notation), construct the H_∞-optimal observer, using the matrices given in that reference. What is the predicted H_∞ gain?
- Combine the optimal observer and optimal state-feedback controller and construct the closed-loop system. What is the resulting H_∞ gain? You may use the Matlab `norm` command to determine the norm of the closed-loop system.

Mid-Term Examination MAE 598

Mason Smith

November 15 2020

1 Decentralized Feedback

1.a For dynamic system given above, ignore the constraints on the controller and design the standard H-optimal full-state-feedback controller and determine the H norm of the closed-loop system.

THE LMI FORMULATION

$min\gamma$ such that there exists a $X > 0$ and Z:

$$\begin{bmatrix} [AX + XA^T + B_2Z + Z^TB2^T & B_1 & (C_1X + D_{12}Z)^T \\ B_1^T & -\gamma I & D_{11}^T \\ C_1X + D_{12}Z & D_{11} & -\gamma I \end{bmatrix} < 0$$

where $K = ZX^{-1}$

Matlab Results

Using MATLAB, the resulting gain is $\gamma = 2.8682$

$$K = 1.0e+06 * \begin{bmatrix} 0.0023 & 0.0076 & -0.0031 & -0.0099 & 0.0009 & -0.0163 \\ 0.5920 & 1.9811 & -0.8053 & -2.5624 & 0.2216 & -4.2498 \\ -0.5921 & -1.9814 & 0.8054 & 2.5628 & -0.2216 & 4.2504 \end{bmatrix}$$

$$H_\infty Norm : \gamma = 2.8682$$

1.b Contraints on controller

THE LMI FORMULATION

For the controller to obtain the specified constraints on K we must specify the form of the matrices involved in its calculation:

$$K = ZX^{-1} = \begin{bmatrix} K & *K_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & K_3 & K_4 & K_5 & K_6 \\ 0 & 0 & K_7 & K_8 & K_9 & K_{10} \end{bmatrix}$$

Therefore, X and Z matrices must have the following block diagonal form for the controller K to be reconstructed properly when the matrix multiplication process is carried out:

$$Z = \begin{bmatrix} Z & Z_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_3 & Z_4 & Z_5 & Z_6 \\ 0 & 0 & Z_7 & Z_8 & Z_9 & Z_{10} \end{bmatrix}$$

$$X = \begin{bmatrix} X_1 & X_2 & 0 & 0 & 0 & 0 \\ X_3 & X_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & X_5 & X_6 & X_7 & X_8 \\ 0 & 0 & X_9 & X_{10} & X_{11} & X_{12} \\ 0 & 0 & X_{13} & X_{14} & X_{15} & X_{16} \\ 0 & 0 & X_{17} & X_{18} & X_{19} & X_{20} \end{bmatrix}$$

To specify this constraint we define the yalmip variable as

Z= blkdiag(sdpvar(1,2),sdpvar(2,4));
X= blkdiag(sdpvar(2,2),sdpvar(4,4));

and carry out the same LMI procedure in part 1.A. The resulting gain is = 4.5175 which is higher than the unconstrained controller. This makes sense since adding extra constraints on the optimization process reduces the possibilities for the minimization of gamma.

MATLAB RESULTS

$$K = 1.0e+03 * \begin{bmatrix} 7.0395 & 3.1076 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.7074 & 1.7764 & -0.1606 & 3.0215 \\ 0 & 0 & 0.8718 & -2.2110 & 0.2002 & -3.7726 \end{bmatrix}$$

$$H_\infty Norm = 4.5175$$

2 Switched Systems

2.A DESIGN AN LMI WHICH YIELDS A FULL-STATE FEEDBACK STABILIZING CONTROLLER WHICH WORKS FOR BOTH INPUT MODELS WHILE MINIMIZING AND H-BOUND WHICH IS VALID DESPITE THE SWITCHING BACK AND FORTH. PROVIDE ME WITH BOTH THE CONTROLLER AND THE CLOSED-LOOP H GAIN

THE LMI FORMULATION

LMI for Quadratic Polytopic H?-OptimalState-Feedback Control(Lecture 13 Theorem 10)
Since we only have the two switching matrices, they will be defined as B2a and B2b

There exists F such that $||\underline{S}(P(\Delta), K(0,0,0,F)||_{H_\infty} \leq \gamma$ for all $\Delta \in C_0(\Delta_1 ... \Delta_k$ if there exists $Y > 0$ and Z such that

$$\begin{bmatrix} Y*A' + A*Y + Z'*(B2a+B2b)' + (B2a+B2b)*Z & B_1 & (C1*Y+D12*Z)' \\ B_1' & -\gamma I & D11' \\ C1*Y+D12*Z & D11 & -\gamma I \end{bmatrix} < 0$$

where $F = ZY^{-1}$

MATLAB RESULTS

$$\gamma = 4.3202$$

$$F = 1.0e+07 * \begin{bmatrix} 0.1690 & 0.0001 & 0.1415 & 0.1414 & 0.1139 & 0.1414 \\ -5.8063 & -0.0010 & -4.8625 & -4.8593 & -3.9176 & -4.8615 \\ 5.9281 & 0.0010 & 4.9644 & 4.9612 & 3.9998 & 4.9634 \end{bmatrix}$$

2.B FOR COMPARISON, PROVIDE THE H GAIN ACHIEVABLE USING ONLY THE FIRST INPUT MODEL (THE ONE ON THE LEFT)

THE LMI FORMULATION

$min\gamma$ such that there exists $Y > 0$ and Z:

$$\begin{bmatrix} Y*A' + A*Y + Z'*(B2a)' + (B2a)*Z & B_1 & (C1*Y+D12*Z)' \\ B_1' & -\gamma I & D11' \\ C1*Y+D12*Z & D11 & -\gamma I \end{bmatrix} < 0$$

MATLAB RESULTS

$$\gamma = 3.7417$$

$$F = \begin{bmatrix} -1.1911 & -5.6429 & -0.1010 & -0.1494 & 2.7307 & 3.4175 \\ -4.6557 & 4.6478 & 0.3274 & -2.6488 & 2.9227 & -2.5860 \\ 4.9574 & -21.3640 & -2.1640 & 11.2957 & 9.0855 & 15.9222 \end{bmatrix}$$

The larger γ for the switching system makes sense since it is unlikely that the optimal gamma for one system will be optimal for the other. Therefore, the optimization compromises to meet the demands of the minimization of the two systems. However, when optimizing for a γ that is just concerned with one system, a more optimal solution can be found.

3 Tracking Problem

3.A USING THE TRACKING FRAMEWORK, CONSTRUCT AN OPTIMAL CONTROL FRAMEWORK 9-MATRIX REPRESENTATION. COMPUTE THE OPEN-LOOP (SET K = 0) H2 AND H NORMS OF THE SYSTEM (IF FINITE).

9-Matrix Representation

(i)Define Equations for Regulated Outputs in terms of Exogenous/Actuator Inputs:

$z1 = r - P_0(n_{proc} + u)$
$z2 = u$

(ii)Define Equations for Sensed Outputs in terms of Exogenous/Actuator Inputs:

$y_1 = r$
$y_2 = q + n_{sensor}$

(iii)Construct Matrices for Regulated/Sensed Outputs:

$$z = \begin{bmatrix} e \\ u \end{bmatrix} = \begin{bmatrix} r - P_0(n_{proc} + u) \\ u \end{bmatrix} = \begin{bmatrix} \omega_1 - P_0(\omega_2 + u) \\ u \end{bmatrix}$$

$$y = \begin{bmatrix} r \\ q + n_{sensor} \end{bmatrix} = \begin{bmatrix} r \\ P_0(n_{proc} + u) + n_{sensor} \end{bmatrix} = \begin{bmatrix} \omega_1 \\ P_0(\omega_2 + u) + \omega_3 \end{bmatrix}$$

(iiv)Construct Aggregate Plant:

$$P = \begin{bmatrix} z_1 \\ z_2 \\ y_1 \\ y_2 \end{bmatrix} = \left[\begin{array}{cccc} I & -P_0 & 0 & | & -P_0 \\ 0 & 0 & 0 & | & I \\ \hline I & 0 & 0 & | & 0 \\ 0 & P_0 & I & | & P_0 \end{array}\right] \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ u \end{bmatrix} = \left[\begin{array}{ccc} P_{11} & | & P_{12} \\ \hline P_{21} & | & P_{22} \end{array}\right] \begin{bmatrix} \omega \\ u \end{bmatrix}$$

(iv)Construct State Space Representation:

where $\omega_2 + u$ is our input signal

$$P = \left[\begin{array}{ccc} P_{11} & | & P_{12} \\ \hline P_{21} & | & P_{22} \end{array}\right] \begin{bmatrix} \omega \\ u \end{bmatrix}$$

$$\dot{x} = Ax + B(\omega_2 + u)$$

$$y = Cx + D(\omega_2 + u)$$

(v)Construct the 9-matrix representation

$$P = \left[\begin{array}{cccccc} A & | & 0 & B & 0 & B \\ \hline C & | & I & -D & 0 & -D \\ 0 & | & 0 & 0 & 0 & I \\ 0 & | & I & 0 & 0 & 0 \\ C & | & 0 & D & I & D \end{array}\right]$$

$$\text{B1} = \begin{bmatrix} 0 & B & 0 \end{bmatrix} \ \text{B2} = \begin{bmatrix} B \end{bmatrix}$$

$$\text{C1} = \begin{bmatrix} C \\ 0 \end{bmatrix} \ \text{C2} = \begin{bmatrix} 0 \\ C \end{bmatrix}$$

$$\text{D11} = \begin{bmatrix} I & -D & 0 \\ 0 & 0 & 0 \end{bmatrix} \ \text{D12} = \begin{bmatrix} -D \\ I \end{bmatrix}$$

$$\text{D21} = \begin{bmatrix} I & 0 & 0 \\ 0 & D & I \end{bmatrix} \ \text{D22} = \begin{bmatrix} 0 \\ D \end{bmatrix}$$

Matlab Results

$$
P = \begin{bmatrix}
-1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & -1 & -1 \\
-1 & -2 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & -2 & -1 & 1 & 1 & 0 & 0 & 0 & -1 & 1 & 1 & 0 & 0 & 0 & -1 & 1 & 1 \\
-1 & -1 & 1 & -3 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
-1 & 1 & -1 & 1 & -2 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & -1 & 0 & 0 & -1 & -2 & 0 & 0 & 0 & -1 & 1 & 1 & 0 & 0 & 0 & -1 & 1 & 1 \\
0 & 1 & 0 & -1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0
\end{bmatrix}'
$$

Defining Open-Loop System

$$
\begin{aligned}
A_{ol} &= A \\
B_{ol} &= \begin{bmatrix} B1 & B2 \end{bmatrix} \\
C_{ol} &= \begin{bmatrix} C1 \\ C2 \end{bmatrix} \\
D_{ol} &= \begin{bmatrix} D11 & D12 \\ D21 & D22 \end{bmatrix}
\end{aligned}
$$

Open-Loop H2 Gain

The H2 norm is infinite because the system has nonzero feed-through.

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

Therefore: H2 OpenLoop Norm = Inf

Open-Loop Hinf Gain

Hinf OpenLoop Norm =8.6374 Matlab was used to calculate the Hinf norm of the open-loop system $sys = ss(A_{ol}, B_{ol}, C_{ol}, D_{ol})$ using the following command:

$$norm(sys, inf)$$

this resulted in a predicted Hinf Gain $\gamma = 1.3930$

3.B Use an LMI to formulate and solve the H-optimal output-feedback problem. What is thepredicted H gain?

THE LMI FORMULATION

$min\gamma$ such that there exists a $X1, Y1 > 0$ and A_n, B_n, C_n, D_n:

$$\begin{bmatrix} X1 & I \\ I & Y1 \end{bmatrix} > 0$$

$$\begin{bmatrix} A*Y1+Y1*A'+B2*Cn+Cn'*B2' & (A'+An+(B2*Dn*C2)')' & B1+B2*Dn*D21 & (C1*Y1+D12*Cn)' \\ A'+An+(B2*Dn*C2)' & X1*A+A'*X1+Bn*C2+C2'*Bn' & X1*B1+Bn*D21 & (C1+D12*Dn*C2)' \\ (B1+B2*Dn*D21)' & (X1*B1+Bn*D21)' & -\gamma*I & (D11+D12*Dn*D21)' \\ C1*Y1+D12*Cn & C1+D12*Dn*C2 & D11+D12*Dn*D21 & -\gamma*I \end{bmatrix} < 0$$

MATLAB RESULTTS

Predicted$_H inf_G ain$:$\gamma = 1.3930$

3.C Reconstruct the controller using Y2 = 2 · I.

H_∞ norm. In addition to this the controller $\hat{K}(A_K, B_K, C_K, D_K)$ can also be recovered.
$D_K = (I + D_{K2}D_{22})^{-1}D_{K2}$
$B_K = B_{K2}(I + D_{22}D_K)$
$C_K = (I + D_K D_{22})C_{K2}$
$A_K = A_{K2} - B_K(I + D_{22}D_K)^{-1}D_{22}C_K$

where, $\begin{bmatrix} A_{K2} & B_{K2} \\ C_{K2} & D_{K2} \end{bmatrix} = \begin{bmatrix} X_2 & X_1B_2 \\ 0 & I \end{bmatrix}^{-1} \left[\begin{bmatrix} A_n & B_n \\ C_n & D_n \end{bmatrix} - \begin{bmatrix} X_1AY_1 & 0 \\ 0 & 0 \end{bmatrix}\right] \begin{bmatrix} Y_2^T & 0 \\ C_2Y_1 & I \end{bmatrix}^{-1}$

for any full-rank X_2 and Y_2 such that

$$\begin{bmatrix} X_1 & X_2 \\ X_2^T & X_3 \end{bmatrix} = \begin{bmatrix} Y_1 & Y_2B_2 \\ Y_2^T & Y_3 \end{bmatrix}^{-1}$$

$\therefore$ we can set $Y_2 = I$ and $X_2 = I - X_1 * Y_1$ and perform the above operations in inverse order to on our feasible solutions to our optimization problem to find A_K, B_K, C_K, D_K and construct the controller:

$$K = \begin{bmatrix} A_K & B_k \\ C_K & D_k \end{bmatrix}$$

since Y2=2I we calculate X2 accordingly

$$X2 = (I - X1 * Y1) * 2$$

we then carry out the following calculations in matlab with the results from the formulated LMI

Matlab Results

$$Ak = \begin{bmatrix} -1.1359e+08 & 1.2135e+08 & 2.6457e+08 & -5.3968e+07 & 3.1408e+07 & 1.0117e+08 \\ 1.1480e+07 & -1.2264e+07 & -2.6737e+07 & 5.4540e+06 & -3.1741e+06 & -1.0224e+07 \\ -2.4290e+07 & 2.5949e+07 & 5.6573e+07 & -1.1540e+07 & 6.7161e+06 & 2.1633e+07 \\ -1.4278e+07 & 1.5253e+07 & 3.3253e+07 & -6.7833e+06 & 3.9477e+06 & 1.2716e+07 \\ 5.7690e+07 & -6.1630e+07 & -1.3436e+08 & 2.7408e+07 & -1.5951e+07 & -5.1379e+07 \\ -6.8128e+07 & 7.2780e+07 & 1.5867e+08 & -3.2367e+07 & 1.8837e+07 & 6.0675e+07 \end{bmatrix}$$

$$Bk = \begin{bmatrix} -6.3700e+06 & 1.1927e+06 & 1.9217e+07 & -1.0574e+07 & -9.2282e+06 & 4.3064e+07 \\ 6.4375e+05 & -1.2053e+05 & -1.9421e+06 & 1.0686e+06 & 9.3260e+05 & -4.3520e+06 \\ -1.3621e+06 & 2.5503e+05 & 4.1092e+06 & -2.2610e+06 & -1.9733e+06 & 9.2085e+06 \\ -8.0065e+05 & 1.4990e+05 & 2.4154e+06 & -1.3290e+06 & -1.1599e+06 & 5.4127e+06 \\ 3.2351e+06 & -6.0570e+05 & -9.7596e+06 & 5.3700e+06 & 4.6867e+06 & -2.1871e+07 \\ -3.8204e+06 & 7.1529e+05 & 1.1525e+07 & -6.3417e+06 & -5.5346e+06 & 2.5828e+07 \end{bmatrix}$$

$$Ck = \begin{bmatrix} 0.1158 & -0.6545 & -0.0448 & 0.4787 & -0.1319 & 0.7074 \\ -0.8634 & -0.8474 & 1.3220 & -1.3362 & -1.4293 & -0.0985 \\ 0.1071 & 0.7518 & 0.9540 & -0.4925 & -0.2955 & -0.0339 \end{bmatrix}$$

$$Dk = \begin{bmatrix} 0.1299 & -0.1056 & -0.1472 & 0.0123 & -0.0040 & 0.0126 \\ -0.2712 & -0.2481 & 0.1210 & -0.0398 & -0.0425 & -0.0042 \\ -0.0803 & -0.0580 & 0.3079 & -0.0362 & -0.0304 & 0.0221 \end{bmatrix}$$

3.D Construct and write down the closed-loop system matrices and use the Matlab norm command to verify the closed-loop system has the desired H gain.

Using the matlab state space system command and the linear fractional transform command:

$$controller = ss(Ak, Bk, Ck, Dk);$$

$$plant = ss(A, [B1B2], [C1; C2t], [D11D12; D21tD22t]);$$

$$sys_{cl} = lft(plant, controller);$$

the closed loop system can be defined

MATLAB RESULTS

$$Acl = \begin{bmatrix} -0.6485 & 1.3061 & -0.4289 & 1.0760 & 0.0729 & 0.9821 & 0.7563 & 0.0956 & -2.2760 & 1.8286 & 1.7248 & 0.1324 \\ -1 & -2 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5186 & -0.2005 & -1.4239 & -1.0884 & 0.9312 & 1.0053 & -0.8722 & 0.5589 & 2.3207 & -2.3074 & -1.5929 & -0.8398 \\ -1.1299 & -0.8944 & 1.1472 & -3.0123 & 0.0040 & -0.0126 & -0.1158 & 0.6545 & 0.0448 & -0.4787 & 0.1319 & -0.7074 \\ -1.0803 & 0.9420 & -0.6921 & 0.9638 & -2.0304 & -0.9779 & 0.1071 & 0.7518 & 0.9540 & -0.4925 & -0.2955 & -0.0339 \\ -0.4814 & -1.2005 & 0.5761 & -0.0884 & -1.0688 & -1.9947 & -0.8722 & 0.5589 & 2.3207 & -2.3074 & -1.5929 & -0.8398 \\ -1.5925e+06 & 2.9816e+05 & 4.8043e+06 & -2.6435e+06 & -2.3071e+06 & 1.0766e+07 & -2.8398e+07 & 3.0338e+07 & 6.6142e+07 & -1.3492e+07 & 7.8521e+06 & 2.5292e+07 \\ 1.6094e+05 & -3.0132e+04 & -4.8552e+05 & 2.6715e+05 & 2.3315e+05 & -1.0880e+06 & 2.8699e+06 & -3.0659e+06 & -6.6843e+06 & 1.3635e+06 & -7.9353e+05 & -2.5560e+06 \\ -3.4053e+05 & 6.3757e+04 & 1.0273e+06 & -5.6526e+05 & -4.9332e+05 & 2.3021e+06 & -6.0725e+06 & 6.4872e+06 & 1.4143e+07 & -2.8850e+06 & 1.6790e+06 & 5.4082e+06 \\ -2.0016e+05 & 3.7476e+04 & 6.0385e+05 & -3.3226e+05 & -2.8997e+05 & 1.3532e+06 & -3.5694e+06 & 3.8131e+06 & 8.3134e+06 & -1.6958e+06 & 9.8692e+05 & 3.1789e+06 \\ 8.0877e+05 & -1.5143e+05 & -2.4399e+06 & 1.3425e+06 & 1.1717e+06 & -5.4676e+06 & 1.4422e+07 & -1.5407e+07 & -3.3591e+07 & 6.8521e+06 & -3.9878e+06 & -1.2845e+07 \\ -9.5511e+05 & 1.7882e+05 & 2.8814e+06 & -1.5854e+06 & -1.3837e+06 & 6.4569e+06 & -1.7032e+07 & 1.8195e+07 & 3.9669e+07 & -8.0918e+06 & 4.7093e+06 & 1.5169e+07 \end{bmatrix}$$

$$Bcl = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$Ccl = \begin{bmatrix} 0 & 1 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.1299 & -0.1056 & -0.1472 & 0.0123 & -0.0040 & 0.0126 & 0.1158 & -0.6545 & -0.0448 & 0.4787 & -0.1319 & 0.7074 \\ -0.2712 & -0.2481 & 0.1210 & -0.0398 & -0.0425 & -0.0042 & -0.8634 & -0.8474 & 1.3220 & -1.3362 & -1.4293 & -0.0985 \\ -0.0803 & -0.0580 & 0.3079 & -0.0362 & -0.0304 & 0.0221 & 0.1071 & 0.7518 & 0.9540 & -0.4925 & -0.2955 & -0.0339 \end{bmatrix}$$

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

3.E CONSTRUCT THE H-OPTIMAL FULL STATE FEEDBACK CONTROLLER FOR THIS SYSTEM. WHAT IS THE PREDICTED H-GAIN?

THE LMI FORMULATION
$min\gamma$ such that there exists a $X > 0$ and Z:

$$\begin{bmatrix} [AX + XA^T + B_2Z + Z^TB2^T & B_1 & (C_1X + D_{12}Z)^T \\ B_1^T & -\gamma I & D_{11}^T \\ C_1X + D_{12}Z & D_{11} & -\gamma I \end{bmatrix} < 0$$

where $K = ZX^{-1}$

Matlab Results

$$K = \begin{bmatrix} 4.3914e+03 & -3.8719e+03 & 1.9796e+03 & 4.3779e+03 & 716.6931 & 5.4267e+03 \\ 1.7240e+03 & 1.8655e+03 & -279.5315 & -1.7268e+03 & -2.8260e+03 & -1.0110e+03 \\ 5.1093e+03 & 3.5015e+03 & -43.4754 & -3.0026e+03 & -6.4600e+03 & -681.8082 \end{bmatrix}$$

Predicted Hinf Gain = 1.0001

3.F USING THE LMI ON PAGE 86 IN CAVERLY (295 IN DUAN AND LI, BUT THIS USES DIFFERENT NOTATION), CONSTRUCT THE H-OPTIMAL OBSERVER, USING THE MATRICES GIVEN IN THAT REFERENCE. WHAT IS THE PREDICTED H GAIN?

LMI Formulation
$min\gamma$ such that $P > 0$ and

$$\begin{bmatrix} PA + A^TP - GC_2 & PB_1 - GD_{21} & C_1^T \\ * & -\gamma I & D_{11}^T \\ * & * & -\gamma I \end{bmatrix} < 0$$

where $L = P^{-1}G$

MATLAB RESULTS
Predicted$_H inf_G ain = 1.2420$

$$L = \begin{bmatrix} 2.2084e+03 & 1.0814e+03 & 4.6612e+03 & 1.5863e+03 & 3.5902e+03 & 4.9307e+03 \\ -858.5717 & -420.4329 & -1.8121e+03 & -616.8680 & -1.3958e+03 & -1.9169e+03 \\ -645.6689 & -316.7675 & -1.3609e+03 & -465.8036 & -1.0515e+03 & -1.4392e+03 \\ -798.4180 & -391.3720 & -1.6841e+03 & -574.7438 & -1.2988e+03 & -1.7815e+03 \\ -2.5250e+03 & -1.2364e+03 & -5.3297e+03 & -1.8135e+03 & -4.1043e+03 & -5.6385e+03 \\ 576.4192 & 281.8158 & 1.2183e+03 & 412.3793 & 935.2885 & 1.2896e+03 \end{bmatrix}$$

3.G COMBINE THE OPTIMAL OBSERVER AND OPTIMAL STATE-FEEDBACK CONTROLLER AND CONSTRUCT THE CLOSEDLOOP SYSTEM. WHAT IS THE RESULTING H GAIN? YOU MAY USE THE MATLAB NORM COMMAND TO DETERMINE THE NORM OF THE CLOSED-LOOP SYSTEM.

To combine the optimal observer with the optimal controller we define:

$$At = \begin{bmatrix} A + B * K & -B * K \\ 0 & A + L * C \end{bmatrix}$$

$$Bt = \begin{bmatrix} B \\ 0 \end{bmatrix}$$

$$Ct = \begin{bmatrix} C & 0 \end{bmatrix}$$

$$Dt = D$$

which can be evauluatedm using the MATLAB state space system and norm command

$$sys = ss(At, Bt, Ct, D);$$

$$gain = norm(sys_c l, inf)$$

MATLAB RESULTS

Observer/Full-State FeedbackH_∞ gain = 3.2066

4 Matlab Main Function: Problem 1

```
% Demo from M. Peet for lecture 10 in MAE 598
%
% % Model
clear all
clc

A=[
    1   1   1   0   0   1;
    -1  0   -1  0   0   1;
    1   0   0   -1  1   1;
    -1  1   -1  0   0   0;
    -1  -1  1   1   -1  -1;
    0   -1  0   0   -1  0
    ];
B1 = [
    0   -1  -1;
    0   0   0;
    -1  -1  1;
    -1  0   1;
    0   0   0;
    -1  1   1
    ];
B2 = [
    0   0   0;
    -1  0   1;
    -1  1   0;
    1   -1  0;
    -1  0   -1;
    0   1   1
    ];
C1 = [
    0   1   0   -1  -1  -1;
    0   0   0   -1  0   0;
    1   0   0   0   -1  0
    ];
C2 = zeros(size(C1));
D12 = [
    0   1   1 ;
    0   0   0;
    1   1   1
    ];
D11 = [
    0    0   1;
    -1  0   0;
    0   0   0
    ];
D21 = zeros(size(D11));
D22 = zeros(size(D11));

% measure numbers of inputs and outputs

```

```
eta=.0001;     % degree of strict positivity
ns=size(A,1);    % number of states
nc=size(B2,2);  % number of actuators
nd=size(B1,2);  % number of external inputs
nr=size(C1,1);  % number of regulated outputs
C2t=eye(ns); D21t=zeros(ns,nd); D22t=zeros(ns,nc);
nm=size(C2,1);  % number of sensors


% H-infinity State Feedback Controller Synthesis


% Declare the variables
gamma=sdpvar(1);                  % represents the bound on the H-infinity norm of the CL system.
X=sdpvar(ns);
Z=sdpvar(nc,ns,'full');
W=sdpvar(nr);

% declare constraints
MAT=[A*X+X*A'+B2*Z+Z'*B2'         B1                  (C1*X+D12*Z)';
      B1'                         -gamma*eye(nd)    D11';
      C1*X+D12*Z                  D11                  -gamma*eye(nr)];
F=[MAT<=0];
F=[F;X>=eta*eye(ns)];

OPTIONS = sdpsettings('solver','mosek','verbose',0);

% Solve the LMI, minimizing gamma
optimize(F,gamma,OPTIONS);
gamman=value(gamma)

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% retrieve decision variables
Xn=value(X);
Zn=value(Z);
K=Zn*inv(Xn)

controller=ss(K);
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Close the loop with Lower LFT
plant=ss(A,[B1 B2],[C1;C2t],[D11 D12; D21t D22t]);
sys_cl=lft(plant,controller);
%[Acl,Bcl,Ccl,Dcl] = ssdata(sys_cl);
%sys_cl=ss(Acl,Bcl,Ccl,0);
Hinf_Norm = norm(sys_cl,inf)
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% compare with Matlab built-in functions
```

```
C2tt=ones(1,ns); D21tt=ones(1,nd);D22tt=zeros(1,nc);
sys=ss(A,[B1 B2],[C1;C2tt],[D11 D12; D21tt D22tt]);nm=size(C2tt,1);  % number of sensors
[Knew,CL,GAM,INFO]=hinfsyn(sys,nm,nc);
Matlab_Norm =  INFO.GAMFI % This assumes controller depnds on disturbances too, which is apples to oranges

%% 1B
disp("################################################################################")
disp("1B")
disp("################################################################################")

clear all
A=[
    1   1   1   0   0   1;
    -1  0   -1  0   0   1;
    1   0   0   -1  1   1;
    -1  1   -1  0   0   0;
    -1  -1  1   1   -1  -1;
    0   -1  0   0   -1  0
    ];
B1 = [
    0   -1  -1;
    0   0   0;
    -1  -1  1;
    -1  0   1;
    0   0   0;
    -1  1   1
    ];
B2 = [
    0   0   0;
    -1  0   1;
    -1  1   0;
    1   -1  0;
    -1  0   -1;
    0   1   1
    ];
C1 = [
    0   1   0   -1  -1  -1;
    0   0   0   -1  0   0;
    1   0   0   0   -1  0
    ];
C2 = zeros(size(C1));
D12 = [
    0   1   1 ;
    0   0   0;
    1   1   1
    ];
D11 = [
    0    0    1;
    -1  0   0;
    0   0   0
    ];
D21 = zeros(size(D11));
```

```
D22 = zeros(size(D11));

% measure numbers of inputs and outputs

eta=.0001;     % degree of strict positivity
ns=size(A,1);    % number of states
nc=size(B2,2);  % number of actuators
nd=size(B1,2);  % number of external inputs
nr=size(C1,1);  % number of regulated outputs
C2t=eye(ns); D21t=zeros(ns,nd); D22t=zeros(ns,nc);
nm=size(C2,1);  % number of sensors


% H-infinity State Feedback Controller Synthesis


% Declare the variables
gamma=sdpvar(1);                % represents the bound on the H-infinity norm
    of the CL system.
X= blkdiag(sdpvar(2,2),sdpvar(4,4));

Z= blkdiag(sdpvar(1,2),sdpvar(2,4));
%Z=sdpvar(nc,ns,'full');
W=sdpvar(nr);

% declare constraints
MAT=[A*X+X*A'+B2*Z+Z'*B2'        B1                 (C1*X+D12*Z)';
      B1'                         -gamma*eye(nd)     D11';
      C1*X+D12*Z                  D11                -gamma*eye(nr)];
F=[MAT<=0];
F=[F;X>=eta*eye(size(ns))];

OPTIONS = sdpsettings('solver','mosek','verbose',0);

% Solve the LMI, minimizing gamma
optimize(F,gamma,OPTIONS);
gamman=value(gamma)

%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% retrieve decision variables
Xn=value(X);
Zn=value(Z);
K=Zn*inv(Xn)
controller=ss(K);
%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Close the loop with Lower LFT
plant=ss(A,[B1 B2],[C1;C2t],[D11 D12; D21t D22t]);
sys_cl=lft(plant,controller);
%[Acl,Bcl,Ccl,Dcl] = ssdata(sys_cl);
%sys_cl=ss(Acl,Bcl,Ccl,0);
```

```
Hinf_norm = norm(sys_cl ,inf)
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% compare with Matlab built-in functions
C2tt=ones(1,ns); D21tt=ones(1,nd);D22tt=zeros(1,nc);
sys=ss(A,[B1 B2],[C1;C2tt],[D11 D12; D21tt D22tt]);nm=size(C2tt,1); % number of sensors
[Knew,CL,GAM,INFO]=hinfsyn(sys,nm,nc);
Matlab_norm = INFO.GAMFI % This assumes controller depnds on disturbances too, which is apples to oranges
%INFO.KFI-K % compare full-state gains.
```

5 Matlab Main Function: Problem 2

```
%%1A
clear all
clc
disp("################################################################################")
disp("1A")
disp("################################################################################")
A=[
    1   1   0   1   0   1;
    -1  -1  -1  0   0   1;
    1   0   1   -1  1   1;
    -1  1   -1  -1  0   0;
    -1  -1  1   1   1   -1;
    0   -1  0   0   -1  -1];
B1=[
    0   -1  -1;
    0   0   0;
    -1  1   1;
    -1  0   0;
    0   0   1;
    -1  1   1];
C1=[
    0   1   0   -1  -1  -1;
    0   0   0   -1  0   0;
    1   0   0   0   -1  0;
    ];
C2= zeros(size(C1));
B2a =[
    0   0   0;
    -1  0   1;
    -1  1   0;
    1   -1  0;
    -1  0   -1;
    0   1   1];
B2b = [
    0   0   0;
    -1  0   1;
    -1  1   0;
    1   1   0;
    1   0   1;
    0   -3  -1];
D11 = [1 2 3; 0 0 0; 0 0 0];
D12 = [0 0 0; 0 0 0; 0 0 0];
D21 = [0 0 0; 0 0 0; 0 0 0];
D22 = [0 0 0; 0 0 0; 0 0 0];


% measure numbers of inputs and outputs

eta=.0001;    % degree of strict positivity
ns=size(A,1);   % number of states
nc=size(B2a,2);  % number of actuators
```

```
nd=size(B1,2);  % number of external inputs
nr=size(C1,1);  % number of regulated outputs
C2t=eye(ns); D21t=zeros(ns,nd); D22t=zeros(ns,nc);
nm=size(C2,1);  % number of sensors


% LMI for Quadratic Polytopic H?-OptimalState-Feedback Control
% Lecture 13 Theorem 10.


% Declare the variables
gamma=sdpvar(1);                % represents the bound on the H-infinity norm
    of the CL system.
Y=sdpvar(ns);
Z=sdpvar(nc,ns,'full');

% declare constraints
m11 = [Y*A'+A*Y+Z'*(B2a+B2b)'+(B2a+B2b)*Z];
m21 = [B1'];
m31 = [C1*Y+D12*Z];
m22 = [-gamma*eye(3)];
m32 = [D11];
m33 = [-gamma*eye(3)];

MAT=[m11 m21' m31';
     m21 m22 m32';
     m31 m32 m33];

F=[MAT<=0];
F=[F;Y>=eta*eye(ns)];

OPTIONS = sdpsettings('solver','mosek','verbose',0);

% Solve the LMI, minimizing gamma
optimize(F,gamma,OPTIONS);
gamman=value(gamma)

%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% retrieve decision variables
Yn=value(Y);
Zn=value(Z);
F=Zn*inv(Yn)

%% 1B
disp("##########################################################################")
disp("1B")
disp("##########################################################################")

% measure numbers of inputs and outputs
B2= B2a;
% measure numbers of inputs and outputs

```

```
eta=.0001;    % degree of strict positivity
ns=size(A,1);   % number of states
nc=size(B2,2);  % number of actuators
nd=size(B1,2);  % number of external inputs
nr=size(C1,1);  % number of regulated outputs
C2t=eye(ns); D21t=zeros(ns,nd); D22t=zeros(ns,nc);
nm=size(C2,1);  % number of sensors


% H-infinity State Feedback Controller Synthesis


% measure numbers of inputs and outputs

eta=.0001;    % degree of strict positivity
ns=size(A,1);   % number of states
nc=size(B2a,2);  % number of actuators
nd=size(B1,2);  % number of external inputs
nr=size(C1,1);  % number of regulated outputs
C2t=eye(ns); D21t=zeros(ns,nd); D22t=zeros(ns,nc);
nm=size(C2,1);  % number of sensors


% LMI for Quadratic Polytopic H?-OptimalState-Feedback Control
% Lecture 13 Theorem 10.


% Declare the variables
gamma=sdpvar(1);                % represents the bound on the H-infinity norm
    of the CL system.
Y=sdpvar(ns);
Z=sdpvar(nc,ns,'full');

% declare constraints
m11 = [Y*A'+A*Y+Z'*(B2a)'+(B2a)*Z];
m21 = [B1'];
m31 = [C1*Y+D12*Z];
m22 = [-gamma*eye(3)];
m32 = [D11];
m33 = [-gamma*eye(3)];

MAT=[m11 m21' m31';
     m21 m22 m32';
     m31 m32 m33];

F=[MAT<=0];
F=[F;Y>=eta*eye(ns)];

OPTIONS = sdpsettings('solver','mosek','verbose',0);

% Solve the LMI, minimizing gamma
optimize(F,gamma,OPTIONS);
gammanA=value(gamma)
```

```
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% retrieve decision variables
Yn=value(Y);
Zn=value(Z);
F=Zn*inv(Yn)
```

6 Matlab Main Function: Problem 3

```
%% 3A
disp("###########################################################################")
disp("3A")
disp("###########################################################################")
% % Model
clear all
clc

A=[
    -1  1   0   1   0   1;
    -1  -2  -1  0   0   1;
    1   0   -2  -1  1   1;
    -1  -1  1   -3  0   0;
    -1  1   -1  1   -2  -1;
    0   -1  0   0   -1  -2;
    ];
B=[
    0   -1  -1;
    0   0   0;
    -1  1   1;
    -1  0   0;
    0   0   1;
    -1  1   1];
C=[
    0   1   0   -1  -1  -1;
    0   0   0   -1  0   0;
    1   0   0   0   -1  0];
D=zeros(3);

disp("9-MATRIX REP###")
I = eye(3);
zB = zeros(size(B));
zC = zeros(size(C));
zD = zeros(size(D));
P = [
    A   zB  B   zB  B;
    C   I   -D  zD  -D;
    zC  zD  zD  zD  I;
    zC  I   zD  zD  zD;
    C   zD  D   I   D
    ];
sympref('FloatingPointOutput',true);
Latex_out = latex(sym(P))
% B1 = [zB B];
% B2 = [zB B];
% C1 = [C; zC];
% C2 = [zC; C];
%
% D11=[I -D; zD zD];
% D12=[I zD; zD D];
% D21=[zD -D; zD I];
```

```
% D22 = [zD zD; I D];


B1 = [zB B zB];
B2 = [B];
C1 = [C; zC];
C2 = [zC; C];

D11=[I    -D   zD;
     zD   zD    zD];
D12=[-D; I];
D21=[I     zD   zD;
     zD   D    I];
D22 = [zD; D];

% Use an LMI or matlab function?
% ############################################################
disp("Hinf and H2 norms of open loop sys")
sys=ss(A,[B1 B2],[C1;C2],[D11 D12; D21 D22]);
disp("The H2 norm is infinite because the system has nonzero feedthrough. ")
disp([D11 D12; D21 D22])
sympref('FloatingPointOutput',true);
Latex_out = latex(sym([D11 D12; D21 D22]))
H2_OpenLoop_Norm = norm(sys,2)
Hinf_OpenLoop_Norm = norm(sys,inf)

%% 3B
disp("##########################################################################")
disp("3B")
disp("##########################################################################")
% measure numbers of inputs and outputs

eps=.000001;    % degree of strict positivity
ns=size(A,1);   % number of states
nc=size(B2,2);  % number of actuators
nm=size(C2,1);  % number of sensors
nd=size(B1,2);  % number of external inputs
no=size(C1,1);  % number of regulated outputs


% H-infinity Dynamic Output Feedback Controller Synthesis


% Declare the variables
gamma=sdpvar(1);                % represents the bound on the H-infinity norm of the CL system.
X1=sdpvar(ns);
Y1=sdpvar(ns);
An=sdpvar(ns,ns,'full');
Cn=sdpvar(nc,ns,'full');
Dn=sdpvar(nc,nm,'full');
Bn=sdpvar(ns,nm,'full');

% declare constraints
```

```
F=[X1>=eps*eye(ns)]
F=[F;Y1>=eps*eye(ns)]
F=[F;[X1 eye(ns); eye(ns) Y1]>=0];
MAT=[A*Y1+Y1*A'+B2*Cn+Cn'*B2'    (A'+An+(B2*Dn*C2)')'        B1+B2*Dn*D21        (C1*Y1+D12*Cn)';
     A'+An+(B2*Dn*C2)'           X1*A+A'*X1+Bn*C2+C2'*Bn'    X1*B1+Bn*D21        (C1+D12*Dn*C2)'    ;
     (B1+B2*Dn*D21)'             (X1*B1+Bn*D21)'             -gamma*eye(nd)      (D11+D12*Dn*D21)'    ;
     C1*Y1+D12*Cn                C1+D12*Dn*C2                D11+D12*Dn*D21      -gamma*eye(no)];

F=[F;MAT<=eps*eye(size(MAT))];
OPTIONS = sdpsettings('solver','mosek','verbose',0);


% Solve the LMI, minimizing gamma
optimize(F,gamma,OPTIONS);
gamman=value(gamma);
Predicted_Hinf_Gain = gamman

disp("##########################################################################")
disp("3C Reconstruct Controller with Y2-2*I")
disp("##########################################################################")
% retrieve decision variables
X1n=value(X1);
Y1n=value(Y1);
Ann=value(An);
Bnn=value(Bn);
Cnn=value(Cn);
Dnn=value(Dn);
temp1=[Ann Bnn; Cnn Dnn]-[X1n*A*Y1n zeros(ns,nm); zeros(nc,ns) zeros(nc,nm)];

% Choose X2, Y2, so that X2*Y2=I-X1*Y1;
Y2n=2.0*eye(ns); %############################################## USE Y2=2*I
X2n=(eye(ns)-X1n*Y1n)*2.0;

% Reverse variable substitution
temp2=inv([X2n X1n*B2;zeros(nc,ns) eye(nc)])*temp1*inv([Y2n' zeros(ns,nm); C2*Y1n eye(nm)]);
Ak2=temp2(1:ns,1:ns);Bk2=temp2(1:ns,(ns+1):(ns+nm));Ck2=temp2((ns+1):(ns+nc),1:ns);Dk2=temp2((ns+1):(ns+nc), (ns+1):(ns+nm));
Dk=inv(eye(nc)-Dk2*D22)*Dk2;
Bk=Bk2*(eye(nm)-D22*Dk);
Ck=(eye(nc)-Dk*D22)*Ck2;
Ak=Ak2-Bk*inv(eye(nm)-D22*Dk)*D22*Ck;
disp("Controller")
sympref('FloatingPointOutput',true);
Latex_out = latex(sym(Ak))
Latex_out = latex(sym(Bk))
Latex_out = latex(sym(Ck))
Latex_out = latex(sym(Dk))
Ak
Bk
```

```matlab
Ck
Dk

%% 3D
disp("###########################################################################")
disp("3D Ckosed Loop System Matricies")
disp("###########################################################################")

C2t=eye(ns); D21t=zeros(ns,nd); D22t=zeros(ns,nc);
controller=ss(Ak,Bk,Ck,Dk);
plant=ss(A,[B1 B2],[C1;C2t],[D11 D12; D21t D22t]);
sys_cl=lft(plant,controller);

%[Acl,Bcl,Ccl,Dcl] = ssdata(sys_cl);
%sys_cl=ss(Acl,Bcl,Ccl,0);
Hinf_norm = norm(sys_cl,inf)
Acl = sys_cl.A
Bcl = sys_cl.B
Ccl = sys_cl.C
Dcl = sys_cl.D
% Acl = A
% Bcl = [B1 B2]
% Ccl = [C1;C2]
% Dcl = [D11 D12; D21 D22]
sympref('FloatingPointOutput',true);
Latex_out = latex(sym(Acl))
Latex_out = latex(sym(Bcl))
Latex_out = latex(sym(Ccl))
Latex_out = latex(sym(Dcl))
%
% % Close the loop with Lower LFT
% plant=ss(A,[B1 B2],[C1;C2],[D11 D12; D21 D22]);
% controller=ss(Ak,Bk,Ck,Dk);
% sys_cl=lft(plant,controller);
% Controller_Hinf_Gain = norm(sys_cl,Inf)
%
% % compare with Matlab built-in functions
% sys=ss(A,[B1 B2],[C1;C2],[D11 D12; D21 D22]);
% [K,CL,GAM,INFO]=hinfsyn(sys,nm,nc,'METHOD','lmi');
% [K,CL,GAM,INFO]=hinfsyn(sys,nm,nc);


%% 3E
disp("###########################################################################")
disp("3E Hinf Optimal Full State Feeback Controller")
disp("###########################################################################")



eta=.0001;    % degree of strict positivity
ns=size(A,1);   % number of states
nc=size(B2,2);  % number of actuators
nd=size(B1,2);  % number of external inputs
nr=size(C1,1);  % number of regulated outputs
```

```
C2t=eye(ns); D21t=zeros(ns,nd); D22t=zeros(ns,nc);
nm=size(C2,1);  % number of sensors


% H-infinity State Feedback Controller Synthesis


% Declare the variables
gamma=sdpvar(1);                % represents the bound on the H-infinity norm
    of the CL system.
X=sdpvar(ns);
Z=sdpvar(nc,ns,'full');
W=sdpvar(nr);

% declare constraints
MAT=[A*X+X*A'+B2*Z+Z'*B2'        B1                 (C1*X+D12*Z)';
      B1'                        -gamma*eye(nd)     D11';
      C1*X+D12*Z                 D11                -gamma*eye(nr)];
F=[MAT<=0];
F=[F;X>=eta*eye(ns)];

%OPTIONS = sdpsettings('solver','sedumi');

% Solve the LMI, minimizing gamma
optimize(F,gamma,OPTIONS);
gamman=value(gamma);
Xn = value(X)
Zn = value(Z)
K=Zn*inv(Xn)
Latex_out = latex(sym(K))
Predicted_Hinf_Gain = gamman

%% 3F
disp("#############################################################################")
disp("3F Hinf Optimal Observer")
disp("#############################################################################")

gamma=sdpvar(1);                % represents the bound on the H-infinity norm
    of the CL system.
P=sdpvar(ns);
G=sdpvar(ns,nr);

MAT = [P*A+A'*P-G*C2-C2'*G' P*B1-G*D21 C1';
        (P*B1-G*D21)' -gamma*eye(9) D11';
        C1 D11 -gamma*eye(6)]
F=[MAT<=0];
F=[F;P>=eta*eye(ns)];

optimize(F,gamma,OPTIONS);

Pn = value(P)
Gn = value(G);
gamman=value(gamma);
Predicted_Hinf_Gain = gamman
```

```
L=inv(Pn)*Gn
Latex_out = latex(sym(L))
%% 3E
disp("################################################################################")
disp("Combine the optimal observer and optimal state-feedback controller and
    construct the closedloop system. What is the resulting H? gain? You may
    use the Matlab norm command to determine the norm of the closed-loop
    system.")
disp("################################################################################")
%K = [Ak Bk; Ck Dk]
% At = [ Acl+Bcl*K -Bcl*K
%        zeros(size(Acl)) Acl+L*Ccl ];
% Bt = [ B;
%        zeros(size(Bcl))];
% Ct = [ C     zeros(size(Ccl)) ];
% A=A;
% B=B*K;
% C=-L*Ccl;
% D=Acl+L*Ccl+Bcl*K;
At = [ A+B2*K -B2*K
       zeros(size(A)) A+L*C2 ];
Bt = [ B2;
       zeros(size(B2))];
Ct = [ C2     zeros(size(C2)) ];
sys = ss(At,Bt,Ct,0);
%sys_cl=ss(A,B,C,D);
ObserverStateFeedback_Hinf_Gain = norm(sys_cl,inf)
```