**Simulation report – Midterm Exam**

**Linear System Theory – Prof. Kim, Sung Hyun**

**Student: Chung Quang Khanh – 20245360**

* **Finding**

From example 2.12, we obtain set of linear matrix inequalities (LMIs) that ensures the robust stability as follows:

(1)

(2)

From example 2.2, we also have the state-space model of a dc servomotor that can be described as:

where in which has an uncertainty of

So, let’s define (3)

The parameter uncertainty can be expressed as follows:

where

(3)

* **Solving LMIs to find**

Now we have . Let’s find using Robust Control Toolbox in MATLAB.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Define constants

Ra = 5.385;

La = 3.694e-03;

K = 0.0583;

J = 6.88627e-06;

B0 = 3.1346e-05;

% Define state-space model

A = [-B0/J K/J;

-K/La -Ra/La];

E = [-0.2 0]';

H = [B0/J 0];

N = [2 3]';

G = [1 0];

n = size(A, 1);

% gm = 0.21; % for the second step

% Initialize description of LMIs....

setlmis([]);

% Specify matrix variables in LMIs...

vP = lmivar(1, [n,1]); % nxn, symmetric

veps = lmivar(1, [1, 1]); % scalar variable

vgm2 = lmivar(1, [1, 1]); % scalar variable

% Specify term content of LMIs...

%%% LMI #1

lmiterm( [-1, 1, 1, vP], 1, 1); % 0 < P

%%% LMI #2

lmiterm( [2, 1, 1, vP], 1, A, 's'); % P\*A + A'\*P

lmiterm( [2, 1, 1, 0], G'\*G); % G'\*G

lmiterm( [2, 1, 1, veps], H', H); % eps\*H'\*H

lmiterm( [2, 1, 2, vP], 1, N); % P\*N

lmiterm( [2, 2, 2, vgm2], -1, 1); % -gamma^2

% lmiterm( [2, 2, 2, 0], -gm^2); % for the second step

lmiterm( [2, 1, 3, vP], 1, E); % P\*E

lmiterm( [2, 3, 3, veps], -1, 1); % -eps\*I

% Compute solution

lmisys = getlmis;

options = [0, 0, **0.23**, 0, 0];

% options = [0, 0, 0, 0, 0]; % for the second step

target = 0;

[tmin, xfeas] = feasp(lmisys, options, target);

if ~isempty(xfeas) && tmin < 0

disp('This system is Hinf stable!!');

P = dec2mat(lmisys, xfeas, vP);

eps = dec2mat(lmisys, xfeas, veps);

gm2 = dec2mat(lmisys, xfeas, vgm2);

disp('P = ');disp(P);

disp('Epsilon = ');disp(eps);

disp('Gamma = ');disp(sqrt(gm2));

else

disp('unstable!!');

P = NaN;

eps = NaN;

gm2 = NaN;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

* At first, I declare as a variable (**vgm2**) along with 2 other variables (**vP** and **veps**) in LMIs. I also tune the **options** parameters, specifically **options(3)** is set to **0.23** to find the value of as small as possible. That value is **0.3027**.
* The second step is that is defined as a constant in LMIs with the initial value of **0.3027**. By using trial and error method, is decreased to **0.21** while the system is still stable.
* Therefore, the minimum value of is **0.21**