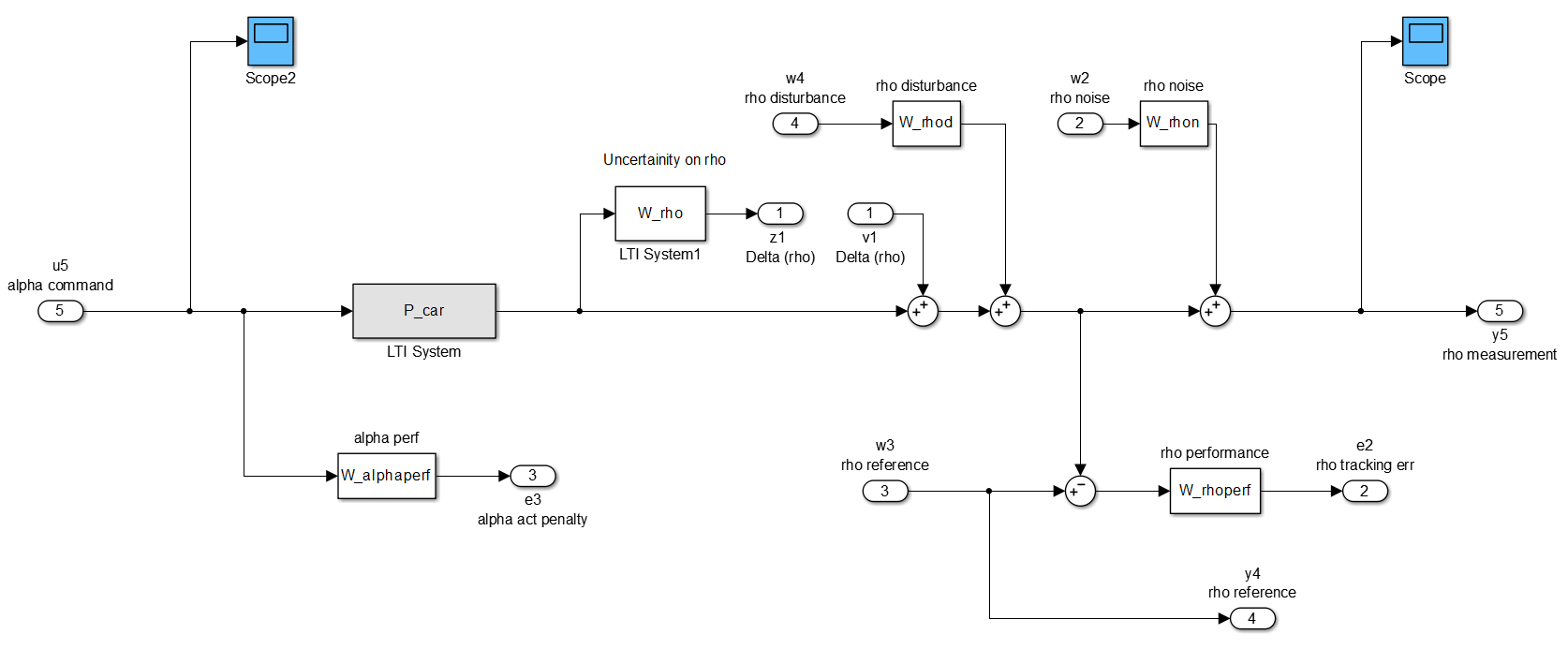
ATIC Exercise 11

a)

i) model\_car:



[A\_P,B\_P,C\_P,D\_P] = linmod('model\_car');

P = ss(A\_P,B\_P,C\_P,D\_P);

P =

a =

x1 x2 x3 x4 x5 x6 x7 x8

x1 0 0 1 0 0 0 0 0

x2 0 0 0 1 0 0 0 0

x3 78.23 -59.38 0 0 0 0 0 0

x4 0 -400 0 -20 0 0 0 0

x5 40.74 0 0 0 -4 0 0 0

x6 -2.546 0 0 0 0 -0.001667 0 -0.02

x7 0 0 0 0 0 0 -1000 0

x8 0 0 0 0 0 0 0 -0.002

b =

u1 u2 u3 u4 u5

x1 0 0 0 0 0

x2 0 0 0 0 0

x3 0 0 0 0 0

x4 0 0 0 0 69.81

x5 0 0 0 0 0

x6 -0.25 0 0.25 0 0

x7 0 0 0 0 128

x8 0 0 0 0.0625 0

c =

x1 x2 x3 x4 x5 x6 x7 x8

y1 20.37 0 0 0 -2 0 0 0

y2 0 0 0 0 0 0.1333 0 0

y3 0 0 0 0 0 0 -125 0

y4 0 0 0 0 0 0 0 0

y5 10.19 0 0 0 0 0 0 0.08

d =

u1 u2 u3 u4 u5

y1 0 0 0 0 0

y2 0 0 0 0 0

y3 0 0 0 0 16

y4 0 0 1 0 0

y5 1 0.01 0 0 0

ii)

% \_\_\_\_\_\_\_\_

% | |

% Delta inputs: z <---| |<--- v: Delta output

% cost/errors: e <---| P |<--- w: exogenous inputs

% measurements: y <---| |<--- u: control actuation

% |\_\_\_\_\_\_\_|

%

% Outputs Inputs

%

% Delta1 (rho) z1 v1

%

% rho tracking err e2 w2 rho noise

% alpha act penalty e3 w3 rho reference

% w4 rho disturbance

%

% rho reference y4 u5 alpha command

% rho measurement y5

Iz = [1:1]';

Ie = [2:3]';

Iy = [4:5]';

Iv = [1:1]';

Iw = [2:4]';

Iu = [5:5]';

A = A\_P;

Bw = B\_P(:,Iw);

Bu = B\_P(:,Iu);

Ce = C\_P(Ie,:);

Cy = C\_P(Iy,:);

Dew = D\_P(Ie,Iw);

Dyw = D\_P(Iy,Iw);

Deu = D\_P(Ie,Iu);

Dyu = D\_P(Iy,Iu);

P\_test = ss(A,[Bw,Bu], [Ce;Cy],[Dew, Deu;Dyw,Dyu]);

P\_test =

a =

x1 x2 x3 x4 x5 x6 x7 x8

x1 0 0 1 0 0 0 0 0

x2 0 0 0 1 0 0 0 0

x3 78.23 -59.38 0 0 0 0 0 0

x4 0 -400 0 -20 0 0 0 0

x5 40.74 0 0 0 -4 0 0 0

x6 -2.546 0 0 0 0 -0.001667 0 -0.02

x7 0 0 0 0 0 0 -1000 0

x8 0 0 0 0 0 0 0 -0.002

b =

u1 u2 u3 u4

x1 0 0 0 0

x2 0 0 0 0

x3 0 0 0 0

x4 0 0 0 69.81

x5 0 0 0 0

x6 0 0.25 0 0

x7 0 0 0 128

x8 0 0 0.0625 0

c =

x1 x2 x3 x4 x5 x6 x7 x8

y1 0 0 0 0 0 0.1333 0 0

y2 0 0 0 0 0 0 -125 0

y3 0 0 0 0 0 0 0 0

y4 10.19 0 0 0 0 0 0 0.08

d =

u1 u2 u3 u4

y1 0 0 0 0

y2 0 0 0 16

y3 0 1 0 0

y4 0.01 0 0 0

iii)

cvx\_begin sdp

cvx\_solver sedumi

variable X(n,n) symmetric;

variable Y(n,n) symmetric;

variable W(length(Ie),length(Ie)) symmetric;

variable Ah(n,n);

variable Bh(n,length(Iy));

variable Ch(length(Iu),n);

variable gamma\_2lmi

minimize gamma\_2lmi;

subject to

trace(W) < gamma\_2lmi;

0.5\*(...

[W, Ce\*X + Deu\*Ch, Ce;

X\*Ce'+Ch'\*Deu', X, eye(n,n);

Ce', eye(n,n),Y]+...

[W, Ce\*X + Deu\*Ch, Ce;

X\*Ce'+Ch'\*Deu', X, eye(n,n);

Ce', eye(n,n),Y]') > 0; % 0.5\*(X+X') to make it symmetric (rounding errors...)

0.5\*(...

[A\*X+Bu\*Ch+X\*A'+Ch'\*Bu', A+Ah', Bw;

A'+Ah, Y\*A'+A'\*Y+Bh\*Cy+Cy'\*Bh', Y\*Bw+Bh\*Dyw;

Bw',Bw'\*Y+Dyw'\*Bh', -eye(length(Iw),length(Iw))]+...

[A\*X+Bu\*Ch+X\*A'+Ch'\*Bu', A+Ah', Bw;

A'+Ah, Y\*A'+A'\*Y+Bh\*Cy+Cy'\*Bh', Y\*Bw+Bh\*Dyw;

Bw',Bw'\*Y+Dyw'\*Bh', -eye(length(Iw),length(Iw))]') < 0;

cvx\_end

iv)

With the LMI, I don’t get a stable closed loop.

|  |  |
| --- | --- |
| **LMI**  open\_loop\_poles =  1.0e+03 \*  -0.0000 + 0.0000i  -0.0100 + 0.0173i  -0.0100 - 0.0173i  0.0088 + 0.0000i  -0.0088 + 0.0000i  -1.0000 + 0.0000i  -0.0000 + 0.0000i  closed\_loop\_poles =  1.0e+03 \*  -0.0000 + 0.0000i  -1.0000 + 0.0000i  -7.3444 + 0.0000i  -0.0042 + 0.0557i  -0.0042 - 0.0557i  -0.0101 + 0.0172i  -0.0101 - 0.0172i  -0.0176 + 0.0000i  -0.0014 + 0.0108i  -0.0014 - 0.0108i  -0.0103 + 0.0000i  -0.0044 + 0.0000i  -0.0040 + 0.0000i  **0.0000 + 0.0000i**  -0.0000 + 0.0000i  closed\_loop\_norm =  Inf | **H2Syn**  open\_loop\_poles =  1.0e+03 \*  -0.0000 + 0.0000i  -0.0100 + 0.0173i  -0.0100 - 0.0173i  0.0088 + 0.0000i  -0.0088 + 0.0000i  -1.0000 + 0.0000i  -0.0000 + 0.0000i  closed\_loop\_poles =  1.0e+03 \*  -0.0000 + 0.0000i  -1.0000 + 0.0000i  -0.0057 + 0.0100i  -0.0057 - 0.0100i  -0.0005 + 0.0000i  -0.0125 + 0.0034i  -0.0125 - 0.0034i  -0.0088 + 0.0000i  -0.0088 + 0.0000i  -0.0100 + 0.0172i  -0.0100 - 0.0172i  -0.0100 + 0.0173i  -0.0100 - 0.0173i  -0.0000 + 0.0000i  -0.0040 + 0.0000i  closed\_loop\_norm =  0.0188 |