Case Study

The purpose of this case study is to show that different LMI solvers may give different solutions to the same linear matrix inequality.

We compare the solutions of the discrete Riccati linear matrix inequality generated by the MATLAB's LMI solver and the CVX LMI solver.

We test the above mentioned solvers on a randomly generated A and B matrices of a discrete-time (DT) system, where

$$A = \begin{bmatrix} 1.0667 & -0.1825 & -0.0983 & -0.2323 \\ -0.9337 & 1.5651 & -0.0414 & -0.4264 \\ -0.3503 & 0.0845 & 0.7342 & 0.3728 \\ 0.0290 & -1.6039 & 0.0308 & 0.2365 \end{bmatrix}$$

and

$$B = \begin{bmatrix} 2.0237 & 1.0001 \\ -2.2584 & -1.6642 \\ 2.2294 & -0.5900 \\ 0.3376 & -0.2781 \end{bmatrix}.$$

The weight matrices are $Q = I_4$ and $R = I_2$.

We first test MATLAB's LMI solver uing the following script:

```
disp('Computing optimal gain using BIG LMI with MATLAB"s Toolbox')
[n,~]=size(A);
[~,m]=size(B);
setlmis([])
S=lmivar(1,[n 1]);
Z=lmivar(2,[m n]);
lmiterm([1 1 1 S],-1,1)
lmiterm([1 1 2 S],1,1)
lmiterm([1 1 3 -Z],1,1)
lmiterm([1 1 4 -Z],1,B')
lmiterm([1 1 4 S],-1,A')
lmiterm([1 2 2 0],-inv(Q))
lmiterm([1 3 3 0],-inv(R))
lmiterm([1 4 4 S],-1,1)
lmiterm([-2 1 1 S],1,1)
lmi_sys=getlmis;
[tmin,xfeas]=feasp(lmi_sys);
S=dec2mat(lmi_sys,xfeas,S);
Zb=dec2mat(lmi_sys,xfeas,Z);
P1=inv(S)
disp('Eigenvalues of P1')
eig(P1)
%
disp('Feedback gain using the ARE formula')
Kb=inv(R+B'*P1*B)*B'*P1*A
disp('Feedback gain from the LMIs')
Kb2=Zb*P1
disp('Closed-loop poles')
eig(A-B*Kb2)
disp('Magnitudes of the closed-loop poles')
disp('using the LMI')
abs(eig(A-B*Kb2))
```

We obtain

$$P1 = \begin{bmatrix} 2.1089 & 0.4507 & -0.4806 & -0.5884 \\ 0.4507 & 12.4199 & -0.3526 & -2.9004 \\ -0.4806 & -0.3526 & 1.6868 & 0.4626 \\ -0.5884 & -2.9004 & 0.4626 & 2.3942 \end{bmatrix}.$$

This gives the controller gain $Kb2$ resulting in the follwing set of closed-loop poles:

$$\{0.4551,\ -0.1755,\ 0.0234,\ 0.1466\}$$

We then test the CVX's LMI solver using the following script:

```
disp('Computing optimal gain using CVX to solve the discrete BIG LMI')
[n,~]=size(A);
[~,m]=size(B);
cvx_begin sdp quiet
% Variable definition
variable S(n, n) symmetric
variable Z(m,n)
% LMIs
[-S, S, Z',Z'*B'-S*A';...
S, -inv(Q), zeros(n,m), zeros(n,n);...
Z, zeros(m,n), -inv(R), zeros(m,n);...
B*Z-A*S, zeros(n,n), zeros(n,m), -S] <= 0
S >= eps*eye(n)
cvx_end
K2 = Z/S  % compute the gain matrix
disp('eig(A-BK2)')
eig(A-B*K2)
disp('Closed-loop pole magnitudes')
abs(eig(A-B*K2))
P2=inv(S) % compute P matrix
disp('eig(P2)')
eig(inv(S))
```

We obtain

$$P2 = \begin{bmatrix} 2.2390 & 1.4615 & -0.5780 & -0.9871 \\ 1.4615 & 16.2159 & -0.8868 & -4.5549 \\ -0.5780 & -0.8868 & 1.6111 & 0.6260 \\ -0.9871 & -4.5549 & 0.6260 & 2.9795 \end{bmatrix}.$$

This gives the controller gain $K2$ resulting in the following set of closed-loop poles:

$$\{0.2012,\ -0.1094 + 0.0502i,\ -0.1094 - 0.0502i,\ 0.0645\}.$$