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 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 continuos-time (CT) 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],A,1,'s')
lmiterm([1 1 1 Z],B,1,'s')
lmiterm([1 1 2 S],1,1)
lmiterm([1 1 3 -Z],1,1)
lmiterm([1 2 2 0],-inv(Q))
lmiterm([1 3 3 0],-inv(R))
lmiterm([-2 1 1 S],1,1)
lmi_sys=getlmis;
[tmin,xfeas]=feasp(lmi_sys);
S=dec2mat(lmi_sys,xfeas,S);
Z=dec2mat(lmi_sys,xfeas,Z);
disp('Solution matrix P using MATLAB"s Toolbox')
P1=inv(S)
disp('Eigenvalues of P1')
eig(P1)
%
K1=inv(R)*B'*P1
disp('Closed-loop matrix Acl1=A-B*K1')
Acl1=A-B*K1
disp('Closed-loop poles')
eig(A-B*K1)
```

We obtain

$$P1 = \begin{bmatrix} 34.1030 & 28.2440 & -10.3948 & -10.9346 \\ 28.2440 & 25.8771 & -7.6211 & -10.4619 \\ -10.3948 & -7.6211 & 4.4785 & 2.9400 \\ -10.9346 & -10.4619 & 2.9400 & 5.7829 \end{bmatrix}.$$

This gives the controller gain $K1$ resulting in the following set of closed-loop poles:

$$\{-22.0864,\ -0.6958,\ -2.6810,\ -4.1480\}$$

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

PURDUE UNIVERSITY

```
disp('Solving BIG LMI using CVX')
[n,~]=size(A);
[~,m]=size(B);
cvx_begin sdp quiet
% Variable definition
variable S(n, n) symmetric
variable Z(m,n)
% LMIs
[S*A' + A*S - B*Z - Z'*B', S, Z';...
S, -inv(Q), zeros(n,m);...
Z, zeros(m,n), -inv(R)] <= 0
S >= eps*eye(n)
cvx_end
disp('Gain K using CVX')
K2 = Z/S  % compute K matrix
disp('Solution P using CVX')
P2=inv(S) % compute P matrix
disp('Closed-loop matrix Acl2=A-B*K2')
Acl2=A-B*K2
disp('eig(A-BK2)')
eig(A-B*K2)
disp('eig(P2)')
eig(inv(S))
```

We obtain

$$P2 = \begin{bmatrix} 20.3044 & 17.5381 & -5.9855 & -7.4441 \\ 17.5381 & 18.0662 & -4.1387 & -8.2103 \\ -5.9855 & -4.1387 & 3.0477 & 1.6449 \\ -7.4441 & -8.2103 & 1.6449 & 5.1447 \end{bmatrix}.$$

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

$$\{-11.3913,\ -0.5828,\ -2.5835,\ -3.7095\}.$$