8 A First Glimpse on Design with LMIs

8.1 Conceptual Design Problem

Given a linear time invariant system design a linear time invariant controller or filter so as to guarantee some closed loop indices of robustness and performance. Desired Properties:

a) Structural constraints: decentralization, design of plant and controller parameters, robustness to structured uncertainty.

b) Controllers and filters with fixed order (assigned by the designer).

c) The design procedure (algorithm) should have *low computational complexity*.

Available Methods for Linear Filtering and Control Design:

a) Linear quadratic

b) Youla parametrization (frequency-domain)

c) LMIs (time-domain)

8.2 Linear quadratic (infinite-horizon) and LMIs

a) Relatively flexible design,

b) Controller/filter is a function of some auxiliary variables and the Lyapunov matrix P,

c) Controller/filter is obtained as a nonlinear function of P,

d) Structure on auxiliary variables does not translate into structure on controller or filter,

e) However, some structural constraints are possible,

f) Controller/filter with full order,

g) Contains linear quadratic methods,

h) Robustness

8.3 Design with Linear Matrix Inequalities

Steps:

1. Impose the controller/filter structure:
   a) State feedback control,
   b) Dynamic filtering,
   c) Dynamic output feedback control.

2. Obtain appropriate analysis conditions in closed loop.
   WARNING: these conditions will not be LMIs!

3. Apply non linear transformations to obtain an LMI design problem:
   a) Congruence transformation + linearizing change-of-variables,
   b) Existence conditions: Finsler's Lemma.

4. Recover controller/filter as a nonlinear function of the synthesis variables.

8.4 An LMI Design Problem: stabilization by state feedback

Plant
(CTLTI system)

$$\dot{x} = Ax + B_u u, \qquad x(0) = x_0, \qquad x \in \mathbb{R}^n$$

Controller
(*state-feedback* controller)

$$u = Kx$$

Closed Loop System

$$\dot{x} = \underbrace{(A + B_u K)}_{A_{\mathrm{cl}}} x, \qquad x(0) = x_0$$

Problem: Find a stabilizing controller gain K

8.5 Stabilization by state feedback: solution by congruence + change-of-variables

First formulate the analysis conditions, in this case, stability.

A_{cl} is stable iff there exists $K \in \mathbb{R}^{m\times n}$ and $P \in \mathbb{S}^n$ such that

$$P \succ 0, \qquad \underbrace{(A^T + K^T B_u^T)}_{A_{\mathrm{cl}}^T} P + P \underbrace{(A + B_u K)}_{A_{\mathrm{cl}}} \prec 0$$

Both K and P are variables $\implies$ NOT an LMI!
Then apply nonlinear transformation: two recipes are available.

Recipe I (congruence + change-of-variables):
Apply congruence transformations:

$$P \succ 0 \qquad \iff \qquad 0 \prec P^{-1}(P)P^{-1} = P^{-1}$$

$$\begin{aligned}(A^T + K^T B_u^T)P + P(A + B_u K) \prec 0 \\ \iff \qquad 0 \succ P^{-1}\left[(A^T + K^T B_u^T)P + P(A + B_u K)\right]P^{-1} \\ = (A + B_u K)P^{-1} + P^{-1}(A^T + K^T B_u^T)\end{aligned}$$

That is

$$P^{-1} \succ 0 \qquad (A + B_u K)P^{-1} + P^{-1}(A^T + K^T B_u^T) \prec 0$$

Still not an LMI!

Apply change-of-variables

$$X := P^{-1}, \qquad L := KP^{-1}$$

to obtain the LMI

$$X \succ 0 \qquad AX + B_u L + XA^T + L^T B_u^T \prec 0$$

Note that

$$
\begin{array}{ll}
P \succ 0, \quad (A^T + K^T B_u^T)P + P(A + B_u K) \prec 0, & \\
\Updownarrow & \text{(by congruence)}, \\
P^{-1} \succ 0, \quad (A + B_u K)P^{-1} + P^{-1}(A^T + K^T B_u^T) \prec 0, & \\
\Downarrow & (X := P^{-1}, \quad L := KP^{-1}), \\
X \succ 0, \quad AX + XA^T + B_u L + L^T B_u^T \prec 0, & \\
\Downarrow & (P := X^{-1}, \quad K = LX^{-1}), \\
P^{-1} \succ 0, \quad (A + B_u K)P^{-1} + P^{-1}(A^T + K^T B_u^T) \prec 0 &
\end{array}
$$

and the fact that $X \succ 0$ is what guarantees sufficiency!

8.5.1 Summary: Stabilization by State Feedback (first form)

The CTLTI system

$$\dot{x} = Ax + B_u u, \quad x(0) = x_0,$$

is stabilizable by the state feedback control

$$u = Kx,$$

if, and only if, $\exists X \in \mathbb{S}^n$ and $L \in \mathbb{R}^{m \times n}$ such that

$$X \succ 0, \quad AX + XA^T + B_u L + L^T B_u^T \prec 0.$$

If so, a stabilizing control gain is $K = LX^{-1}$.

8.6 Finsler's Lemma

Lemma [Finsler]: Let $x \in \mathbb{R}^n$, $\mathcal{Q} \in \mathbb{S}^n$ and $\mathcal{B} \in \mathbb{R}^{m \times n}$ such that $\operatorname{rank}(\mathcal{B}) < n$. The following statements are equivalent:

i) $x^T \mathcal{Q} x < 0, \quad$ for all $\mathcal{B}x = 0, \quad x \neq 0$.

ii) $\mathcal{B}^{\perp T} \mathcal{Q} \mathcal{B}^{\perp} \prec 0$.

iii) $\exists\, \mu \in \mathbb{R} : \mathcal{Q} - \mu \mathcal{B}^T \mathcal{B} \prec 0$.

iv) $\exists\, \mathcal{X} \in \mathbb{R}^{n \times m} : \mathcal{Q} + \mathcal{X}\mathcal{B} + \mathcal{B}^T \mathcal{X}^T \prec 0$.

Remarks:

a) $\mathcal{B}^{\perp}$ is a *basis* for the null space of $\mathcal{B}$. That is, all $x \neq 0$ such that $\mathcal{B}x = 0$ is generated by some $z \neq 0$ in the form $x = \mathcal{B}^{\perp} z$.

b) $\mathcal{X} = -\frac{\mu}{2} \mathcal{B}^T$ is feasible.

Proof:

$i) \Rightarrow ii)$: All x such that $\mathcal{B}x = 0$ can be written as $x = \mathcal{B}^{\perp}y$. Consequently

$$x^T \mathcal{Q} x = y^T {\mathcal{B}^{\perp}}^T \mathcal{Q} \mathcal{B}^{\perp} y < 0$$

for all $y \neq 0$ which implies ${\mathcal{B}^{\perp}}^T \mathcal{Q} \mathcal{B}^{\perp} \prec 0$.

$ii) \Rightarrow i)$: Assuming that the first part of $ii)$ holds, multiply ${\mathcal{B}^{\perp}}^T \mathcal{Q} \mathcal{B}^{\perp}$ on the right by any $y \neq 0$ and on the left by y^T to obtain $y^T {\mathcal{B}^{\perp}}^T \mathcal{Q} \mathcal{B}^{\perp} y = x^T \mathcal{Q} x < 0$.

$iii)$, $iv) \Rightarrow ii)$: Multiply $ii)$ or $iii)$ on the right by $\mathcal{B}^{\perp}$ and on the left by ${\mathcal{B}^{\perp}}^T$ so as to obtain $ii)$.

$iii) \Rightarrow iv)$: Choose $\mathcal{X} = -(\mu/2)\mathcal{B}^T$.

$ii) \Rightarrow iii)$: Assume that $ii)$ holds. Partition $\mathcal{B}$ in the full rank factors $\mathcal{B} = \mathcal{B}_l \mathcal{B}_r$, define $\mathcal{D} := \mathcal{B}_r^T \left(\mathcal{B}_r \mathcal{B}_r^T\right)^{-1} \left(\mathcal{B}_l^T \mathcal{B}_l\right)^{1/2}$ and apply the congruence transformation

$$\begin{bmatrix} \mathcal{D}^T \\ {\mathcal{B}^{\perp}}^T \end{bmatrix} \left(\mathcal{Q} - \mu \mathcal{B}^T \mathcal{B}\right) \begin{bmatrix} \mathcal{D} & \mathcal{B}^{\perp} \end{bmatrix} = \begin{bmatrix} \mathcal{D}^T \mathcal{Q} \mathcal{D} - \mu I & \mathcal{D}^T \mathcal{Q} \mathcal{B}^{\perp} \\ {\mathcal{B}^{\perp}}^T \mathcal{Q} \mathcal{D} & {\mathcal{B}^{\perp}}^T \mathcal{Q} \mathcal{B}^{\perp} \end{bmatrix} \prec 0.$$

Since the second diagonal block is negative definite by assumption, a sufficiently large μ exists so that the whole matrix is negative definite.

8.7 Stabilization by state feedback: solution by Finsler's Lemma

Recipe II (existence conditions):
$A_{\rm cl}$ is stable iff there exists $K \in \mathbb{R}^{m\times n}$ and $P \in \mathbb{S}^n$ such that

$$P \succ 0, \qquad \underbrace{A^T P + PA}_{\mathcal{Q}} + \underbrace{K^T}_{\mathcal{X}} \underbrace{B_u^T P}_{\mathcal{B}} + \underbrace{PB_u}_{\mathcal{B}^T} \underbrace{K}_{\mathcal{X}^T} \prec 0.$$

In this form the equivalence between items $iv)$ and $ii)$ of Finsler's Lemma can be used to *eliminate* variable K.

A first step is to compute $\mathcal{B}^\perp$. Since $P \succ 0$ is nonsingular

$$\underbrace{B_u^T P}_{\mathcal{B}} x = 0, \qquad x \neq 0 \qquad \Longleftrightarrow \qquad B_u^T y = 0, \qquad y = Px \neq 0.$$

Therefore

$$y = B_u^{T\perp} z, \quad \Rightarrow \quad x = P^{-1} y = \underbrace{P^{-1} B_u^{T\perp}}_{\mathcal{B}^\perp} z \quad \Rightarrow \quad \mathcal{B}^\perp = P^{-1} B_u^{T\perp}$$

From Finsler's Lemma,

$$iv) \quad \exists \underbrace{K^T}_{\mathcal{X}} : \underbrace{A^T P + PA}_{\mathcal{Q}} + \underbrace{K^T}_{\mathcal{X}} \underbrace{B_u^T P}_{\mathcal{B}} + \underbrace{PB_u}_{\mathcal{B}^T} \underbrace{K}_{\mathcal{X}^T} \prec 0,$$

$$\Updownarrow$$

$$ii) \quad \underbrace{B_u^{T\perp^T} P^{-1}}_{\mathcal{B}^{\perp^T}} \underbrace{(A^T P + PA)}_{\mathcal{Q}} \underbrace{P^{-1} B_u^{T\perp}}_{\mathcal{B}^\perp} \prec 0.$$

Using $X := P^{-1}$ this last inequality can becomes

$$B_u^{T\perp^T} (AX + XA^T) B_u^{T\perp} \prec 0.$$

8.7.1 Summary: Stabilization by State Feedback (second form)

The CTLTI system

$$\dot{x} = Ax + B_u u, \quad x(0) = x_0,$$

is stabilizable by the state feedback control

$$u = Kx,$$

if, and only if, $\exists X \in \mathbb{S}^n$ such that

$$X \succ 0, \quad {B_u^{T\perp}}^T (AX + XA^T) B_u^{T\perp} \prec 0.$$

The item $iii)$ of Finsler's Lemma provide yet another equivalent condition that does not require the explicit computation of $\mathcal{B}^{\perp}$

$$iv) \quad \exists \underbrace{K^T}_{\mathcal{X}} : \underbrace{A^T P + PA}_{\mathcal{Q}} + \underbrace{K^T}_{\mathcal{X}} \underbrace{B_u^T P}_{\mathcal{B}} + \underbrace{PB_u}_{\mathcal{B}^T} \underbrace{K}_{\mathcal{X}^T} \prec 0,$$

$$\Updownarrow$$

$$iii) \quad \underbrace{(A^T P + PA)}_{\mathcal{Q}} - \mu \underbrace{PB_u}_{\mathcal{B}^T} \underbrace{B_u^T P}_{\mathcal{B}} \prec 0.$$

Multiplying this inequality by μ and using $Y := \mu P$

$$A^T(\mu P) + (\mu P)A - (\mu P)B_u B_u^T(\mu P) = A^T Y + YA - YB_u B_u^T Y \prec 0.$$

Hence, there exists $Q \succ 0$ such that the *Riccati equation*

$$A^T Y + YA - YB_u B_u^T Y + Q = 0,$$

holds. This can be explored to provide a full equivalence between stabilizability and the solution to the LQR problem.
However, from the point of view of inequality conditions, the *Riccati inequality*

$$A^T Y + YA - YB_u B_u^T Y \prec 0$$

is not convex (it is indeed concave!).
Once again the congruence transformation

$$Y^{-1}(A^T Y + YA - YB_u B_u^T Y)Y^{-1} \prec 0$$

with $X := Y^{-1}$ provides the LMI

$$AX + XA^T - B_u B_u^T \prec 0.$$

8.7.2 Summary: Stabilization by State Feedback (third form)

The CTLTI system

$$\dot{x} = Ax + B_u u, \quad x(0) = x_0,$$

is stabilizable by the state feedback control

$$u = Kx,$$

if, and only if, $\exists X \in \mathbb{S}^n$ such that

$$X \succ 0, \quad AX + XA^T - B_u B_u^T \prec 0.$$

8.8 Recovering the Controller

Given a feasible $X \succ 0$, a stabilizing controller is given by

$$\underbrace{K}_{\mathcal{X}^T} = -\underbrace{1/2}_{\mu/2}\underbrace{B_u^T P}_{\mathcal{B}} = -\frac{1}{2} B_u^T X^{-1}$$

8.9 Relation with stabilizability (PBH test)

The pair (A, B_u) is stabilizable if, and only if, there exists no $x \in \mathbb{C}^n$ and such that $x^* A = \lambda x^*$ and $x^* B_u = 0$ with $\lambda + \lambda^* \geq 0$.

Note that the conditions

$$X \succ 0, \quad AX + XA^T - B_u B_u^T \prec 0,$$

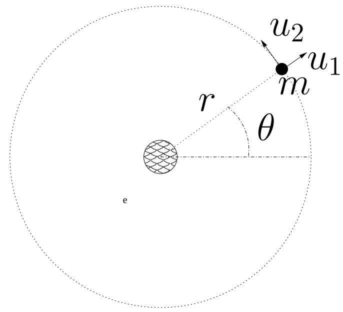
imply that for all $x \in \mathbb{C}^n$ such that $x^* A = \lambda x^*$, $\lambda + \lambda^* \geq 0$

$$x^* X x > 0, \quad x^*(AX + XA^T - B_u B_u^T)x < 0.$$

Hence

$$\begin{aligned} (\lambda + \lambda^*) x^* X x - x^* B_u B_u^T x \leq 0 \quad &\Rightarrow \quad x^* B_u B_u^T x > (\lambda + \lambda^*) x^* X x > 0 \\ &\Rightarrow \quad x^* B_u \neq 0. \end{aligned}$$

8.10 Example: satellite in circular orbit



Satellite of mass m with thrust in the radial direction u_1 and in the tangential direction u_2. Continuing...

$$m(\ddot{r} - r\dot{\theta}^2) = u_1 - \frac{km}{r^2} + w_1,$$
$$m(2\dot{r}\dot{\theta} + r\ddot{\theta}) = u_2 + w_2,$$

where w_1 and w_2 are independent white noise disturbances with variances δ_1 and δ_2.
As before, putting in state space and linearizing around

$$\bar{x}(t) = \begin{pmatrix} \bar{r} \\ \bar{\omega}t \\ 0 \\ \bar{\omega} \end{pmatrix}, \qquad \bar{u} = \begin{pmatrix} \bar{u}_1 \\ \bar{u}_2 \end{pmatrix} = 0, \qquad \bar{w} = \begin{pmatrix} \bar{w}_1 \\ \bar{w}_2 \end{pmatrix} = 0, \qquad \bar{\omega} = \sqrt{k/\bar{r}^3}$$

one obtains the linearized system

$$\dot{x} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3\,\bar{\omega}^2 & 0 & 0 & 2\,\bar{r}\bar{\omega} \\ 0 & 0 & -2\,\bar{\omega}/\bar{r} & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1/m & 0 \\ 0 & 1/(m\bar{r}) \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1/m & 0 \\ 0 & 1/(m\bar{r}) \end{bmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}.$$

Problem: Given

$$m = 100 \text{ kg}, \qquad \bar{r} = R + 300 \text{ km}, \qquad \bar{k} = GM$$

where $G \approx 6.673 \times 10^{-11}$ N m^2/kg^2 is the universal gravitational constant, and $M \approx 5.98 \times 10^{24}$ kg and $R \approx 6.37 \times 10^3$ km are the mass and radius of the earth. If the variances δ_1 and δ_2 are $0.1N$ find stabilizing controls using u_2 only first, then using u_1 and u_2.

```
% MAE 280 B - Linear Control Design
% Mauricio de Oliveira
%
% LMI Stabilizing Control
%
m = 100;               % 100 kg
r = 300E3;             % 300 km
R = 6.37E6;            % 6.37 10^3 km
G = 6.673E-11;         % 6.673 N m^2/kg^2
M = 5.98E24;           % 5.98 10^24 kg
k = G * M;             % gravitational force constant
w = sqrt(k/((R+r)^3)); % angular velocity (rad/s)
v = w * (R + r);       % "ground" velocity (m/s)

% linearized system matrices
A = [0 0 1 0; 0 0 0 1; 3*w^2 0 0 2*(r+R)*w; 0 0 -2*w/(r+R) 0];
Bu = [0 0; 0 0; 1/m 0; 0 1/(m*r)];

% scale
T = diag([1 r 1 r])
T =
           1           0           0           0
           0      300000           0           0
           0           0           1           0
           0           0           0      300000

% similarity transformation
At = T * A / T
At =
            0            0   1.0000e+00            0
            0            0            0   1.0000e+00
   4.0343e-06            0            0   5.1565e-02
            0            0  -1.0432e-04            0
But = T * Bu
But =
            0            0
            0            0
   1.0000e-02            0
            0   1.0000e-02

% stabilizing state feedback control (using u2 only)
n = size(At,1);
m = 1;

% declare variables
X = sdpvar(n,n,'symmetric');
L = sdpvar(m,n);

% declare LMIs
LMI1 = At*X+X*At'+But(:,2)*L+L'*But(:,2)';
LMI2 = X;

LMI = set(LMI1 < 0) + set(LMI2 > 0);
```

```
options = sdpsettings('solver','sedumi');
solution =solvesdp(LMI,[],options)
SeDuMi 1.1R3 by AdvOL, 2006 and Jos F. Sturm, 1998-2003.
Alg = 2: xz-corrector, theta = 0.250, beta = 0.500
eqs m = 14, order n = 9, dim = 33, blocks = 3
nnz(A) = 34 + 0, nnz(ADA) = 188, nnz(L) = 101
 it :     b*y       gap    delta  rate   t/tP*  t/tD*   feas cg cg  prec
  0 :            5.44E+00 0.000
  1 :   0.00E+00 3.97E-01 0.000 0.0729 0.9900 0.9900   1.00  1  0  4.4E-01
  2 :   0.00E+00 2.04E-02 0.000 0.0513 0.9900 0.9900   1.00  1  1  2.2E-02
  3 :   0.00E+00 1.32E-03 0.000 0.0647 0.9900 0.9900   1.00  1  1  1.5E-03
  4 :   0.00E+00 6.66E-05 0.022 0.0506 0.9900 0.9900   1.00  1  1  7.3E-05
  5 :   0.00E+00 3.00E-06 0.107 0.0450 0.9900 0.9900   1.00  1  1  3.3E-06
  6 :   0.00E+00 9.10E-07 0.000 0.3037 0.9000 0.9000   1.00  1  1  1.0E-06
  7 :   0.00E+00 2.11E-08 0.000 0.0232 0.9900 0.9900   1.00  1  1  2.3E-08
  8 :   0.00E+00 1.83E-13 0.000 0.0000 1.0000 1.0000   1.00  1  1  1.6E-12

iter seconds digits       c*x               b*y
  8      0.2   Inf  0.0000000000e+00  0.0000000000e+00
|Ax-b| =   1.2e-13, [Ay-c]_+ =   0.0E+00, |x|=  1.3e-08, |y|=  1.3e+02

Detailed timing (sec)
   Pre          IPM          Post
1.700E-01    2.300E-01    5.000E-02
Max-norms: ||b||=0, ||c|| = 0,
Cholesky |add|=0, |skip| = 0, ||L.L|| = 2615.73.
solution =
    yalmiptime: 1.0549e+00
    solvertime: 6.9148e-01
          info: 'No problems detected (SeDuMi-1.1)'
       problem: 0
        dimacs: [1.1622e-13 0 0 0 0 2.6908e-13]

eig(double(LMI1))
ans =
  -1.0000e+00
  -7.3756e-01
  -1.0763e-02
  -2.8261e-07
eig(double(LMI2))
ans =
   1.3536e-03
   1.0389e+00
   1.9441e+00
   3.2569e+00

% Construct K and check closed loop stability
K = double(L) / double(X)
K =
  -7.5648e+00   3.7632e+00  -2.8232e+03  -8.9542e+01
Acl = At + But(:,2)*K;
eig(Acl)
ans =
  -4.4633e-01 + 1.1030e+00i
```

```
  -4.4633e-01 - 1.1030e+00i
  -2.7132e-03
  -3.9524e-05

% stabilizing state feedback control (using u1 and u2)
n = size(At,1);
m = 2;

% declare variables
X = sdpvar(n,n,'symmetric');
L = sdpvar(m,n);

% declare LMIs
LMI1 = At*X+X*At'+But*L+L'*But';
LMI2 = X;

LMI = set(LMI1 < 0) + set(LMI2 > 0);
options = sdpsettings('solver','sedumi');
solution =solvesdp(LMI,[],options)
SeDuMi 1.1R3 by AdvOL, 2006 and Jos F. Sturm, 1998-2003.
Alg = 2: xz-corrector, theta = 0.250, beta = 0.500
eqs m = 18, order n = 9, dim = 33, blocks = 3
nnz(A) = 38 + 0, nnz(ADA) = 308, nnz(L) = 163
 it :     b*y       gap    delta  rate   t/tP*  t/tD*   feas cg cg  prec
  0 :            5.44E+00 0.000
  1 :   0.00E+00 2.47E-01 0.000 0.0454 0.9900 0.9900   1.00  1  0  2.7E-01
  2 :   0.00E+00 2.41E-05 0.000 0.0001 1.0000 1.0000   1.00  1  1  2.7E-05
  3 :   0.00E+00 2.41E-12 0.000 0.0000 1.0000 1.0000   1.00  1  1  2.7E-12

iter seconds digits       c*x               b*y
  3      0.0   Inf  0.0000000000e+00  0.0000000000e+00
|Ax-b| =   1.5e-12, [Ay-c]_+ =   0.0E+00, |x|=  1.3e-12, |y|=  1.7e+02

Detailed timing (sec)
   Pre          IPM          Post
1.000E-02    2.000E-02    0.000E+00
Max-norms: ||b||=0, ||c|| = 0,
Cholesky |add|=0, |skip| = 1, ||L.L|| = 1.
solution =
    yalmiptime: 3.8992e-02
    solvertime: 4.0474e-02
          info: 'No problems detected (SeDuMi-1.1)'
       problem: 0
        dimacs: [1.5333e-12 0 0 0 0 3.5395e-12]

eig(double(LMI1))
ans =
  -1.0000e+00
  -1.0000e+00
  -7.5000e-01
  -7.5000e-01
eig(double(LMI2))
ans =
   7.5000e-01
```

```
   7.5000e-01
   1.5000e+00
   1.5000e+00

% Construct K and check closed loop stability
K = double(L) / double(X)
K =
  -1.2917e+02   1.9337e+00  -8.7500e+01   6.4457e-01
  -1.9337e+00  -1.2917e+02  -5.7907e+00  -8.7500e+01
Acl = At + But*K;
eig(Acl)
ans =
  -4.4038e-01 + 1.0783e+00i
  -4.4038e-01 - 1.0783e+00i
  -4.3462e-01 + 1.0203e+00i
  -4.3462e-01 - 1.0203e+00i

diary off
```

8.11 A simple example gone bad

```
% MAE 280 B - Linear Control Design
% Mauricio de Oliveira
%
% A simple example gone bad
%
% A simple stabilizable example
A = [0 1; 0 0];
Bu = [0; 1];

n = size(A,1);
m = size(Bu,2);

X = sdpvar(n,n,'symmetric');
L = sdpvar(m,n);

LMI1 = A*X+X*A'+Bu*L+L'*Bu'
Linear matrix variable 2x2 (symmetric, real, 4 variables)
LMI2 = X
Linear matrix variable 2x2 (symmetric, real, 3 variables)

LMI = set(LMI1 < 0) + set(LMI2 > 0);
options = sdpsettings('solver','sedumi');
solution =solvesdp(LMI,[],options)
SeDuMi 1.1R3 by AdvOL, 2006 and Jos F. Sturm, 1998-2003.
Alg = 2: xz-corrector, theta = 0.250, beta = 0.500
eqs m = 5, order n = 5, dim = 9, blocks = 3
nnz(A) = 7 + 0, nnz(ADA) = 21, nnz(L) = 13
 it :     b*y       gap    delta  rate   t/tP*  t/tD*   feas cg cg  prec
  0 :            9.80E+00 0.000
  1 :   0.00E+00 4.45E-01 0.000 0.0454 0.9900 0.9900   1.00  1  0  2.7E-01
  2 :   0.00E+00 4.33E-05 0.000 0.0001 1.0000 1.0000   1.00  1  1  2.7E-05
  3 :   0.00E+00 4.33E-12 0.000 0.0000 1.0000 1.0000   1.00  1  1  2.7E-12

iter seconds digits       c*x               b*y
  3      0.0   Inf  0.0000000000e+00  0.0000000000e+00
|Ax-b| =   1.4e-12, [Ay-c]_+ =   0.0E+00, |x|=  9.2e-13, |y|=  2.0e+00

Detailed timing (sec)
   Pre          IPM          Post
0.000E+00    1.000E-02    1.000E-02
Max-norms: ||b||=0, ||c|| = 0,
Cholesky |add|=0, |skip| = 0, ||L.L|| = 1.
solution =
    yalmiptime: 6.1837e-02
    solvertime: 2.1595e-02
          info: 'No problems detected (SeDuMi-1.1)'
       problem: 0
        dimacs: [1.3977e-12 0 0 0 0 1.7680e-12]

eig(double(LMI1))
ans =
  -1.0000e+00
```

```
  -7.5000e-01
eig(double(LMI2))
ans =
   7.5000e-01
   1.5000e+00

K = double(L) / double(X)
K =
  -1.2917e+00  -8.7500e-01
Acl = A + Bu*K;
eig(Acl)
ans =
  -4.3750e-01 + 1.0489e+00i
  -4.3750e-01 - 1.0489e+00i

% A simple non-stabilizable example
A = [0 1; 0 0];
Bu = [0; 0];

n = size(A,1);
m = size(Bu,2);

X = sdpvar(n,n,'symmetric');
L = sdpvar(m,n);

LMI1 = A*X+X*A'+Bu*L+L'*Bu'
Linear matrix variable 2x2 (symmetric, real, 2 variables)
LMI2 = X
Linear matrix variable 2x2 (symmetric, real, 3 variables)

LMI = set(LMI1 < 0) + set(LMI2 > 0);
options = sdpsettings('solver','sedumi');
solution =solvesdp(LMI,[],options)
SeDuMi 1.1R3 by AdvOL, 2006 and Jos F. Sturm, 1998-2003.
Alg = 2: xz-corrector, theta = 0.250, beta = 0.500
eqs m = 3, order n = 5, dim = 9, blocks = 3
nnz(A) = 5 + 0, nnz(ADA) = 9, nnz(L) = 6
 it :     b*y       gap    delta  rate   t/tP*  t/tD*   feas cg cg  prec
  0 :            9.80E+00 0.000
  1 :   0.00E+00 7.10E-01 0.000 0.0724 0.9900 0.9900   1.00  1  0  4.3E-01
  2 :   0.00E+00 3.43E-02 0.000 0.0484 0.9900 0.9900   1.00  1  1  2.1E-02
  3 :   0.00E+00 2.11E-03 0.053 0.0613 0.9900 0.9900   1.00  1  1  1.3E-03
  4 :   0.00E+00 8.16E-05 0.399 0.0387 0.9900 0.9900   1.00  1  1  5.0E-05
  5 :   0.00E+00 5.02E-07 0.489 0.0062 0.9990 0.9990   1.00  1  1  3.1E-07
  6 :   0.00E+00 1.97E-09 0.163 0.0039 0.9990 0.9990   1.00  1  1  1.2E-09
  7 :   0.00E+00 1.24E-11 0.199 0.0063 0.9990 0.9990   1.00  2  2  7.6E-12

iter seconds digits       c*x               b*y
  7      0.1   Inf  0.0000000000e+00  0.0000000000e+00
|Ax-b| =   3.1e-12, [Ay-c]_+ =   1.1E-12, |x|=  2.0e+00, |y|=  2.0e+00

Detailed timing (sec)
   Pre          IPM          Post
1.000E-02    8.000E-02    0.000E+00
```

```
Max-norms: ||b||=0, ||c|| = 0,
Cholesky |add|=1, |skip| = 0, ||L.L|| = 500000.
solution =
    yalmiptime: 6.2647e-02
    solvertime: 9.6693e-02
          info: 'No problems detected (SeDuMi-1.1)'
       problem: 0
        dimacs: [3.1223e-12 0 0 1.1229e-12 0 2.5606e-12]

eig(double(LMI1))
ans =
  -6.1822e-05
   1.1229e-12
eig(double(LMI2))
ans =
   7.8561e-09
   2.0086e+00

K = double(L) / double(X)
K =
     0     0
Acl = A + Bu*K;
eig(Acl)
ans =
     0
     0

% A simple non-stabilizable example (better)
A = [0 1; 0 0];
Bu = [0; 0];

n = size(A,1);
m = size(Bu,2);

X = sdpvar(n,n,'symmetric');
L = sdpvar(m,n);

LMI1 = A*X+X*A'+Bu*L+L'*Bu'+eye(n)
Linear matrix variable 2x2 (symmetric, real, 2 variables)
LMI2 = X
Linear matrix variable 2x2 (symmetric, real, 3 variables)

LMI = set(LMI1 < 0) + set(LMI2 > 0);
options = sdpsettings('solver','sedumi');
solution =solvesdp(LMI,[],options)
SeDuMi 1.1R3 by AdvOL, 2006 and Jos F. Sturm, 1998-2003.
Alg = 2: xz-corrector, theta = 0.250, beta = 0.500
eqs m = 3, order n = 5, dim = 9, blocks = 3
nnz(A) = 5 + 0, nnz(ADA) = 9, nnz(L) = 6
 it :     b*y       gap    delta  rate   t/tP*  t/tD*   feas cg cg  prec
  0 :            4.90E+00 0.000
  1 :   0.00E+00 4.18E-01 0.000 0.0852 0.9900 0.9900  -0.50  1  1  4.2E+00
  2 :   0.00E+00 3.58E-02 0.000 0.0857 0.9900 0.9900  -1.27  1  1  3.6E+01
  3 :   0.00E+00 2.12E-03 0.000 0.0594 0.9900 0.9900  -1.01  1  1  4.9E+00
```

```
  4 :   0.00E+00 1.24E-04 0.174 0.0584 0.9900 0.9900  -1.11  1  1  6.4E+00
  5 :   0.00E+00 4.36E-06 0.232 0.0351 0.9900 0.9900  -1.11  1  1  7.7E+00
  6 :   0.00E+00 9.67E-07 0.384 0.2219 0.9000 0.9000  -0.96  1  1  7.2E+00
  7 :   0.00E+00 2.45E-07 0.145 0.2531 0.9000 0.9000  -0.88  1  1  5.8E+00
  8 :   0.00E+00 7.41E-08 0.035 0.3029 0.9000 0.9000  -0.58  1  1  4.0E+00
  9 :   0.00E+00 1.37E-08 0.000 0.1848 0.9000 0.9000  -0.60  1  1  7.4E+00
 10 :   0.00E+00 2.96E-09 0.000 0.2157 0.9000 0.9000  -0.85  1  1  6.8E+00
 11 :   0.00E+00 6.02E-10 0.000 0.2036 0.9000 0.9000  -1.01  2  2  8.7E+00

Dual infeasible, primal improving direction found.
iter seconds  |Ax|    [Ay]_+     |x|       |y|
 11      0.1   2.4e-10   2.2e-10   1.0e+00   6.3e+00

Detailed timing (sec)
   Pre          IPM          Post
0.000E+00    7.000E-02    0.000E+00
Max-norms: ||b||=0, ||c|| = 1,
Cholesky |add|=1, |skip| = 0, ||L.L|| = 500000.
solution =
    yalmiptime: 3.2790e-02
    solvertime: 7.1155e-02
          info: 'Infeasible problem (SeDuMi-1.1)'
       problem: 1
        dimacs: [2.4234e-10 0 0 5.0000e-01 -5.0000e-01 -5.0000e-01]

eig(double(LMI1))
ans =
   9.9710e-01
   1.0000e+00
eig(double(LMI2))
ans =
   4.7230e-07
   6.2720e+00

% A simple non-stabilizable example (better/alternative)
A = [0 1; 0 0];
Bu = [0; 0];

n = size(A,1);
m = size(Bu,2);

X = sdpvar(n,n,'symmetric');
L = sdpvar(m,n);

LMI1 = A*X+X*A'+Bu*L+L'*Bu'
Linear matrix variable 2x2 (symmetric, real, 2 variables)
LMI2 = X-eye(n)
Linear matrix variable 2x2 (symmetric, real, 3 variables)

LMI = set(LMI1 < 0) + set(LMI2 > 0);
options = sdpsettings('solver','sedumi');
solution =solvesdp(LMI,[],options)
SeDuMi 1.1R3 by AdvOL, 2006 and Jos F. Sturm, 1998-2003.
Alg = 2: xz-corrector, theta = 0.250, beta = 0.500
```

```
eqs m = 3, order n = 5, dim = 9, blocks = 3
nnz(A) = 5 + 0, nnz(ADA) = 9, nnz(L) = 6
 it :     b*y       gap    delta  rate   t/tP*  t/tD*   feas cg cg  prec
  0 :            4.90E+00 0.000
  1 :   0.00E+00 9.15E-01 0.000 0.1867 0.9000 0.9000   0.14  1  1  2.9E+00
  2 :   0.00E+00 1.82E-01 0.000 0.1992 0.9000 0.9000  -0.45  1  1  1.8E+00
  3 :   0.00E+00 3.67E-02 0.000 0.2012 0.9000 0.9000  -0.44  1  1  1.1E+00
  4 :   0.00E+00 7.02E-03 0.000 0.1913 0.9000 0.9000  -0.37  1  1  5.6E-01
  5 :   0.00E+00 1.32E-03 0.000 0.1886 0.9000 0.9000  -0.34  1  1  2.9E-01
  6 :   0.00E+00 2.63E-04 0.000 0.1988 0.9000 0.9000  -0.39  1  1  1.7E-01
  7 :   0.00E+00 5.83E-05 0.000 0.2215 0.9000 0.9000  -0.43  1  1  1.1E-01
  8 :   0.00E+00 1.62E-05 0.000 0.2786 0.9000 0.9000  -0.39  1  1  6.9E-02
  9 :   0.00E+00 4.50E-06 0.000 0.2771 0.9000 0.9000  -0.30  1  1  4.5E-02
 10 :   0.00E+00 9.89E-07 0.000 0.2198 0.9000 0.9000  -0.30  1  1  2.8E-02
 11 :   0.00E+00 2.96E-07 0.000 0.2994 0.9000 0.9000  -0.24  1  1  1.8E-02
 12 :   0.00E+00 5.74E-08 0.000 0.1938 0.9000 0.9000  -0.29  1  1  1.2E-02
 13 :   0.00E+00 1.68E-08 0.000 0.2929 0.9000 0.9000  -0.25  2  2  7.2E-03
 14 :   0.00E+00 3.57E-09 0.000 0.2126 0.9000 0.9000  -0.28  2  2  4.4E-03
 15 :   0.00E+00 9.14E-10 0.000 0.2557 0.9000 0.9000  -0.29  2  2  2.7E-03
 16 :   0.00E+00 2.27E-10 0.000 0.2479 0.9000 0.9000  -0.31  2  2  1.7E-03
 17 :   0.00E+00 5.61E-11 0.000 0.2476 0.9000 0.9000  -0.34  2  2  1.0E-03
 18 :   0.00E+00 1.34E-11 0.000 0.2388 0.9000 0.9000  -0.35  2  2  6.5E-04
 19 :   0.00E+00 3.26E-12 0.000 0.2432 0.9000 0.9000  -0.36  2  2  4.1E-04
 20 :   0.00E+00 7.57E-13 0.000 0.2325 0.9000 0.9000  -0.35  3  3  2.4E-04
 21 :   0.00E+00 1.86E-13 0.000 0.2453 0.9000 0.9000  -0.34  3  3  1.5E-04

Dual infeasible, primal improving direction found.
iter seconds  |Ax|    [Ay]_+     |x|       |y|
 21      0.1   5.7e-10   6.7e-14   1.9e+04   3.7e+00

Detailed timing (sec)
   Pre          IPM          Post
0.000E+00    1.200E-01    0.000E+00
Max-norms: ||b||=0, ||c|| = 1,
Cholesky |add|=1, |skip| = 0, ||L.L|| = 500000.
solution =
    yalmiptime: 3.2394e-02
    solvertime: 1.3172e-01
          info: 'Infeasible problem (SeDuMi-1.1)'
       problem: 1
        dimacs: [5.6947e-10 0 0 5.0000e-01 -5.0000e-01 -5.0000e-01]

eig(double(LMI1))
ans =
  -1.2811e-04
   6.6629e-14
eig(double(LMI2))
ans =
  -1.0000e+00
   2.7471e+00

diary off
```