Advanced Model Predictive Control

Recitation 1
Introduction to LMIs and Optimization Toolboxes

Alexandre Didier and Jérôme Sieber

ETH Zurich

Fall 2021

Linear Matrix Inequalities [1]

Linear Matrix Inequalities (LMIs) are convex constraints with respect to symmetric matrices.

- General form (strict, but the following slides also hold for non-strict):

$$F(x) = F_0 + \sum_{i=1}^{m} x_i F_i \succ 0 \tag{1}$$

  where $x \in \mathbb{R}^m$, $F_i \in \mathbb{R}^{n \times n}$.

- $F(x) \succ 0$ implies that $F(x)$ is *positive definite*, $F(x) \prec 0$ implies that it is *negative definite* and

$$F(x) \succ G(x) \Leftrightarrow F(x) - G(x) \succ 0.$$

- (1) implies that $z^T F(x) z > 0, \forall z \neq 0, z \in \mathbb{R}^n$.
- (1) can be rewritten as n polynomial constraints (all n principle minors positive)

Homework (PS1): Show that the set of positive definite matrices is convex.

[1] VanAntwerp and Braatz. A tutorial on linear and bilinear matrix inequalities. Journal of Process Control, 2000.

Linear Matrix Inequalities [1]

- LMIs are not unique $\Rightarrow$ Congruence transformation:

$$\begin{aligned} A \succ 0 &\Leftrightarrow z^T A z > 0, \forall z \neq 0 \\ &\Leftrightarrow y^T M^T A M y > 0, \forall y \neq 0, M \text{ nonsingular} \\ &\Leftrightarrow M^T A M \succ 0 \end{aligned}$$

- Multiple LMIs can be expressed as a single LMI

$$F^1(x) \succ 0, F^2(x) \succ 0, \ldots, F^q(x) \succ 0$$

$$F(x) = F_0 + \sum_{i=1}^{m} x_i F_i = \operatorname{diag}\{F^1(x), F^2(x), \ldots, F^q(x)\} \succ 0$$

where $F_i = \operatorname{diag}\{F_i^1, F_i^2, \ldots, F_i^q\}, \ \forall i = 0, \ldots, m.$

[1] VanAntwerp and Braatz. A tutorial on linear and bilinear matrix inequalities. Journal of Process Control, 2000.

Linear Matrix Inequalities [1]

- Schur complement allows rewriting a class of convex nonlinear inequalities as LMIs

$$R(x) \succ 0, \; Q(x) - S(x)R(x)^{-1}S(x)^T \succ 0$$

$$\Leftrightarrow \begin{bmatrix} Q(x) & S(x) \\ S(x)^T & R(x) \end{bmatrix} \succ 0$$

Exercise. Use Schur's complement to rewrite an ellipsoidal inequality as an LMI.

$\mathcal{E} = \{x \mid (x-x_c)^T P (x-x_c) \le 1\} \quad P \succ 0$

$\underbrace{1}_{Q} - \underbrace{(x-x_c)^T}_{S} \underbrace{P}_{R} \underbrace{(x-x_c)}_{S^T} \succeq 0$

$\Leftrightarrow \begin{bmatrix} 1 & (x-x_c)^T \\ (x-x_c) & P^{-1} \end{bmatrix} \succeq 0$

Define $E = P^{-1}$

$\underbrace{1}_{Q} - \underbrace{(x-x_c)^T P}_{S} \underbrace{P^{-1}}_{R} \underbrace{P(x-x_c)}_{S^T} \succeq 0$

$\Leftrightarrow \begin{bmatrix} 1 & (x-x_c)^T P \\ P(x-x_c) & P \end{bmatrix} \succeq 0$

[1] VanAntwerp and Braatz. A tutorial on linear and bilinear matrix inequalities. Journal of Process Control, 2000.

Optimization Toolboxes

A large amount of toolboxes exist that help you solve mathematical optimization problems.

- YALMIP: high level interface to (mostly) convex optimization solvers (SeduMi, quadprog, MOSEK, . . .).
  $\Rightarrow$ Used for very easy problem formulation (directly implement constraints as (in)equalities).
- CasADi: high level interface to nonlinear optimization solvers (IPOPT, . . .) and automatic differentiation
  $\Rightarrow$ Used to manipulate nonlinear functions and to directly implement nonlinear constraints.
- MPT3: toolbox for parametric optimization and computational geometry.
  $\Rightarrow$ Used for easy polytope manipulation.

Quadratic Programming using YALMIP

Example:

$$\text{QP:} \ \min_{x,u} \sum_{i=0}^{4} ||x_i||_Q^2 + ||u_i||_R^2$$

$$\begin{aligned}
\text{s.t.} \ \forall i = 0, \ldots, 4 \\
x_0 &= x(k) \\
x_i &\in \mathbb{X} = \{x \in \mathbb{R}^n | A_x x \leq b_x\} \\
x_{i+1} &= Ax_i + Bu_i \\
x_5 &= 0
\end{aligned}$$

Quadratic Programming using YALMIP

Example:

$$\text{QP:} \quad \min_{x,u} \sum_{i=0}^{4} \|x_i\|_Q^2 + \|u_i\|_R^2$$

$$\text{s.t.} \ \forall i = 0, \ldots, 4$$

$$x_0 = x(k)$$

$$x_i \in \mathbb{X} = \{x \in \mathbb{R}^n | A_x x \leq b_x\}$$

$$x_{i+1} = Ax_i + Bu_i$$

$$x_5 = 0$$

```matlab
% Define optimization variables
x = sdpvar(2, 6, 'full'); u = sdpvar(1, 5, 'full');
x_k = sdpvar(2, 1, 'full');

% Define objective
objective = 0;
for i=1:5
    objective = objective ...
        +x(:,i)'*Q*x(:,i)+u(:,i)'*R*u(:,i);
end

% Define constraints
constraints = [x(:,1) == x_k];
for i = 1:5
    constraints = [constraints, ...
        x(:,i+1) == A*x(:,i) + B*u(:,i)];
    constraints = [constraints, A_x*x(:,i) <= b_x];
end
constraints = [constraints, x(:,6) == zeros(2,1)];

% Define YALMIP Optimizer
opt_MPC =optimizer(constraints, objective, [], {x_k}, ...
    {u(:,1)});

% Simulate for 30 time steps
x_k=[0; 1.8];
for k = 1:30
    u_k = opt_MPC(x_k(:,k));
    x_k(:,k+1) = A*x_k(:,k) + B*u_k;
end
```

Quadratic Programming using YALMIP

Example:

$$\text{QP:} \quad \min_{x,u} \sum_{i=0}^{4} \|x_i\|_Q^2 + \|u_i\|_R^2$$

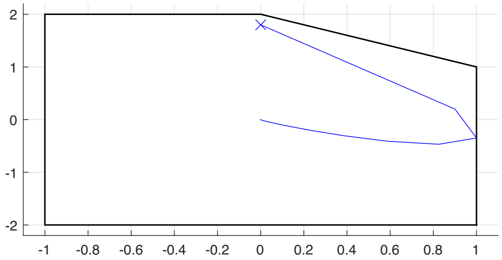
$$\text{s.t.} \ \forall i = 0, \ldots, 4$$

$$x_0 = x(k)$$

$$x_i \in \mathbb{X} = \{x \in \mathbb{R}^n | A_x x \leq b_x\}$$

$$x_{i+1} = Ax_i + Bu_i$$

$$x_5 = 0$$




```
% Define optimization variables
x = sdpvar(2, 6, 'full'); u = sdpvar(1, 5, 'full');
x_k = sdpvar(2, 1, 'full');

% Define objective
objective = 0;
for i=1:5
    objective = objective ...
            +x(:,i)'*Q*x(:,i)+u(:,i)'*R*u(:,i);
end

% Define constraints
constraints = [x(:,1) == x_k];
for i = 1:5
    constraints = [constraints, ...
                   x(:,i+1) == A*x(:,i) + B*u(:,i)];
    constraints = [constraints, A_x*x(:,i) <= b_x];
end
constraints = [constraints, x(:,6) == zeros(2,1)];

% Define YALMIP Optimizer
opt_MPC = optimizer(constraints, objective, [], {x_k}, ...
     {u(:,1)});

% Simulate for 30 time steps
x_k=[0; 1.8];
for k = 1:30
    u_k = opt_MPC(x_k(:,k));
    x_k(:,k+1) = A*x_k(:,k) + B*u_k;
end
```

Semi-definite Programming using YALMIP

Exercise: Compute the largest ellipsoid $\mathcal{E} = \{x \in \mathbb{R}^n \mid x^T P x \leq 1\}$ subset of a polytope $\mathbb{X} = \{x \in \mathbb{R}^n | A_x x \leq b_x\}$ with $A_x \in \mathbb{R}^{n_x \times n}$, $b_x \in \mathbb{R}^{n_x}$. *Hint:* Volume of $\mathcal{E} \propto \operatorname{logdet}(P^{-1})$, which is concave. An ellipsoid is contained within a halfspace $a^T x \leq b$ if $a^T P^{-1} a \leq b^2$.

Semi-definite Programming using YALMIP

Exercise: Compute the largest ellipsoid $\mathcal{E} = \{x \in \mathbb{R}^n \mid x^T P x \leq 1\}$ subset of a polytope $\mathbb{X} = \{x \in \mathbb{R}^n | A_x x \leq b_x\}$ with $A_x \in \mathbb{R}^{n_x \times n}$, $b_x \in \mathbb{R}^{n_x}$. *Hint:* Volume of $\mathcal{E} \propto \text{logdet}(P^{-1})$, which is concave. An ellipsoid is contained within a halfspace $a^T x \leq b$ if $a^T P^{-1} a \leq b^2$.

SDP:
$$
\begin{aligned}
&\min_E \; -\text{logdet } E \\
&\text{s.t. } \forall i = 0, \ldots, n_x \\
&\qquad E \succeq 0 \\
&\qquad [A_x]_i E [A_x]_i^T \leq [b_x]_i^2
\end{aligned}
$$

where $P = E^{-1}$.

Semi-definite Programming using YALMIP

Exercise: Compute the largest ellipsoid $\mathcal{E} = \{x \in \mathbb{R}^n \mid x^T P x \leq 1\}$ subset of a polytope $\mathbb{X} = \{x \in \mathbb{R}^n | A_x x \leq b_x\}$ with $A_x \in \mathbb{R}^{n_x \times n}$, $b_x \in \mathbb{R}^{n_x}$. *Hint:* Volume of $\mathcal{E} \propto \text{logdet}(P^{-1})$, which is concave. An ellipsoid is contained within a halfspace $a^T x \leq b$ if $a^T P^{-1} a \leq b^2$.

SDP:

$$
\begin{aligned}
\min_{E} \ & -\text{logdet } E \\
\text{s.t. } & \forall i = 0, \ldots, n_x \\
& E \succeq 0 \\
& [A_x]_i E [A_x]_i^T \leq [b_x]_i^2
\end{aligned}
$$

where $P = E^{-1}$.

```matlab
% Define A_x, b_x and symmetric variable E
A_x = [1,0; -1,0; 0,1; 0,-1; 1,1];
b_x = [1; 1; 2; 2; 2];
E = sdpvar(2, 2);

% Define objective
objective = -logdet(E);

% Define constraints
% The operator >= denotes matrix inequality
constraints = [E >= 0];
for i=1:size(A_x,1)
    constraints=[constraints, ...
        A_x(i,:)*E*A_x(i,:)' <= b_x(i)^2];
end

% Solve the YALMIP Problem
optimize(constraints, objective)

% Display the result
disp(inv(value(E)))
```

Semi-definite Programming using YALMIP

Exercise: Compute the largest ellipsoid $\mathcal{E} = \{x \in \mathbb{R}^n \mid x^T P x \leq 1\}$ subset of a polytope $\mathbb{X} = \{x \in \mathbb{R}^n | A_x x \leq b_x\}$ with $A_x \in \mathbb{R}^{n_x \times n}$, $b_x \in \mathbb{R}^{n_x}$. *Hint:* Volume of $\mathcal{E} \propto \text{logdet}(P^{-1})$, which is concave. An ellipsoid is contained within a halfspace $a^T x \leq b$ if $a^T P^{-1} a \leq b^2$.

SDP:

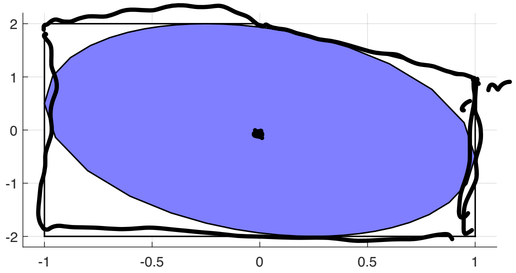
$$\min_E \; -\text{logdet } E$$

$$\text{s.t. } \forall i = 0, \ldots, n_x$$

$$E \succeq 0$$

$$[A_x]_i E [A_x]_i^T \leq [b_x]_i^2$$

where $P = E^{-1}$.




```matlab
% Define A_x, b_x and symmetric variable E
A_x = [1,0; -1,0; 0,1; 0,-1; 1,1];
b_x = [1; 1; 2; 2; 2];
E = sdpvar(2, 2);

% Define objective
objective = -logdet(E);

% Define constraints
% The operator >= denotes matrix inequality
constraints = [E >= 0];
for i=1:size(A_x,1)
    constraints=[constraints, ...
        A_x(i,:)*E*A_x(i,:)' <= b_x(i)^2];
end

% Solve the YALMIP Problem
optimize(constraints, objective)

% Display the result
disp(inv(value(E)))
```