Introduction to Modern Control Systems
Convex Optimization, Duality and Linear Matrix Inequalities

Kostas Margellos
University of Oxford
Agenda for today’s lectures

1. 09.30 - 10.30  
   Convex Optimization

2. 10.30 - 10.45  
   Break

3. 10.45 - 11.30  
   Duality Theory

4. 11.30 - 12.15  
   Linear Matrix Inequalities (LMIs)

---

1 Thanks to Prof. Paul Goulart and Dr. James Anderson for slides input!
So far we have discussed about ...

Standard form optimization problems:

- Convexity
- Convex optimization programs
- Optimality conditions
- Duality

What is left to study?

- Linear Matrix Inequalities, LMIs for short
Linear Matrix Inequalities

- What are LMIs?
- Non-obvious LMIs
- LMIs in state feedback control
- LMIs in optimization
  - Semidefinite programming
  - Primal and dual formulation
What are LMIs?

A Linear Matrix Inequality (LMI) is a constraint of the form:

$$x_1A_1 + x_2A_2 + \cdots + x_nA_n \preceq B$$

where the matrices \((A_1, \ldots, A_n, B)\) are all symmetric.

- This is a constraint that imposes matrix

$$B - \sum_{i}^{n} x_i A_i$$

to be positive semidefinite.

- It is equivalent to imposing \(n\) polynomial inequalities
  - *Not* element-wise constraints.
  - All leading principle minors of this matrix are positive.
What are LMIs?

**A Linear Matrix Inequality (LMI)** is a constraint of the form:

\[ x_1 A_1 + x_2 A_2 + \cdots + x_n A_n \preceq B \]

where the matrices \((A_1, \ldots, A_n, B)\) are all symmetric.

- Consider the constraint

\[ Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \succeq 0 \]

- This is equivalent to 2 inequalities:

\[ Q_{11} \geq 0 \quad \text{det}(Q) \geq 0 \iff Q_{11} Q_{22} - Q_{12} Q_{21} \geq 0 \]
The following LMI constraint is convex.

\[ F(x) = B - \sum_{i=1}^{n} x_i A_i \succeq 0 \]

**Proof:** Let \( x, y \) such that \( F(x), F(y) \succeq 0 \), and \( \lambda \in (0, 1) \).

\[
F(\lambda x + (1 - \lambda)y) = B - \sum_{i} (\lambda x_i + (1 - \lambda)y_i) A_i \\
= \lambda B + (1 - \lambda)B - \lambda \sum_{i} x_i A_i - (1 - \lambda) \sum_{i} y_i A_i \\
= \lambda F(x) + (1 - \lambda)F(y) \\
\geq 0
\]
Theorem

The following LMI constraint is convex.

\[ F(x) = B - \sum_{i=1}^{n} x_i A_i \succeq 0 \]

Alternative proof: We want to show that the set \( \{ x : F(x) \succeq 0 \} \) is convex. We have that ...

\[ \{ x : F(x) \succeq 0 \} = \{ x : z^\top F(x) z \geq 0, \text{ for all } z \} \]

\[ = \bigcap_{z} \{ x : z^\top F(x) z \geq 0 \} \]

... but this is an infinite intersection of sets affine in \( x \) ... so it is convex!

- LMI much harder than linear constraints – an infinite number of them!
- Result can be piecewise affine – LMIs nonlinear!
Why are LMIs interesting?

Linear Matrix Inequalities:

- Appear in many common control design problems (more later on)
- Most of the problems presented so far can be written using LMI constraints

Linear constraints

\[ Ax \leq b \iff \text{diag}(Ax) \leq \text{diag}(b) \]

Quadratic constraints (It will be clear in a couple of slides)

\[ x^\top Q x + b^\top x + c \geq 0, \quad Q \succ 0 \iff \begin{pmatrix} c + b^\top x & x^\top \\ x & -Q^{-1} \end{pmatrix} \succeq 0 \]
Non-obvious LMIs

Some cases (like the QP) are harder to write as LMIs.

There are two useful “tricks”:

- Schur complement
- The $S$-procedure

**Schur complement:** Turns a nonlinear constraint into an LMI

Assume that $Q(x) = Q(x)^\top$, $R(x) = R(x)^\top$ : affine functions of $x$

We then have that

$$R(x) \succ 0 \text{ and } Q(x) - S(x)R(x)^{-1}S(x)^\top \succeq 0$$

$$\iff \begin{bmatrix} Q(x) & S(x) \\ S(x)^\top & R(x) \end{bmatrix} \succeq 0$$
Schur complement

**Schur complement**: Turns a nonlinear constraint into an LMI

Assume that $Q(x) = Q(x)^\top$, $Q(x) = Q(x)^\top$: affine functions of $x$

We then have that

$$R(x) \succ 0 \text{ and } Q(x) - S(x)R(x)^{-1}S(x)^\top \succeq 0$$

$$\iff \begin{bmatrix} Q(x) & S(x) \\ S(x)^\top & R(x) \end{bmatrix} \preceq 0$$

**Example 1:**

$$\|A\|_2 \leq t \iff A^\top A \preceq t^2 I, \quad t \geq 0 \iff \begin{pmatrix} tl & A^\top \\ A & tl \end{pmatrix} \succeq 0$$

**Example 2:** The QP

$$x^\top Qx + b^\top x + c \geq 0, \quad Q \succ 0 \iff \begin{pmatrix} c + b^\top x & x^\top \\ x & -Q^{-1} \end{pmatrix} \succeq 0$$
$S$-procedure: Turns implications to LMIs

When is it true that one quadratic inequality implies another? In other words, when does

$$x^\top \hat{F} x \geq 0, \quad x \neq 0 \implies x^\top F x \geq 0$$

It is not difficult to see that the condition is satisfied if there exists $\tau \in \mathbb{R}$ such that $\tau \geq 0$ and $F \succeq \tau \hat{F}$

The only if part also holds, but less obvious and difficult to prove.
Example: Use a quadratic Lyapunov function to show that

\[ \dot{x} = Ax + g(x) \]

is stable if \( ||g(x)||_2 \leq \gamma ||x||^2, \ \gamma > 0. \)

We want to show that there exists a Lyapunov function

\[ V(x) = x^\top Px, \quad P \succeq 0 \] such that

\[ \dot{V}(x) \leq -\alpha V(x), \quad \text{for all } x, \quad [\alpha > 0 \text{ given}] \]

if \( ||g(x)||_2 \leq \gamma ||x||^2 \)

Or, in other words, if \( ||g(x)||_2 \leq \gamma ||x||^2 \), we want

\[ \dot{V}(x) + \alpha V(x) \leq 0 \]
S-procedure: Example (cont’d)

We want to show that there exists a Lyapunov function

\[ V(x) = x^\top Px, \quad P \succeq 0 \] such that

\[ \dot{V}(x) \leq -\alpha V(x), \text{ for all } x, \quad [\alpha > 0 \text{ given}] \]

if \( \|g(x)\|_2 \leq \gamma \|x\|^2 \)

Letting \( z = g(x) \), we have

\[ \dot{V}(x) + \alpha V(x) = x^\top (A^\top P + PA + \alpha P)x + 2z^\top Pz \]

\[ = \begin{bmatrix} x \\ z \end{bmatrix}^\top \begin{bmatrix} A^\top P + PA + \alpha P & P \\ P & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} \]

Hence, if \( \|g(x)\|_2 \leq \gamma \|x\|^2 \), we want

\[ - \begin{bmatrix} A^\top P + PA + \alpha P & P \\ P & 0 \end{bmatrix} \succcurlyeq 0 \]
The condition

$$\|g(x)\|_2 \leq \gamma \|x\|^2 \Rightarrow z^\top z \leq \gamma^2 x^\top x$$

This can be written as

$$\begin{bmatrix} x \\ z \end{bmatrix}^\top \begin{bmatrix} -\gamma^2 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} \leq 0$$

We thus have

$$- \begin{bmatrix} A^\top P + PA + \alpha P & P \\ P & 0 \end{bmatrix} \succeq 0 \quad \text{if} \quad \begin{bmatrix} \gamma^2 & 0 \\ 0 & -1 \end{bmatrix} \succeq 0$$
S-procedure: Example (cont’d)

We thus have

\[- \begin{bmatrix} A^T P + PA + \alpha P & P \\ P & 0 \end{bmatrix} \succeq 0 \quad \text{if} \quad \begin{bmatrix} \gamma^2 & 0 \\ 0 & -I \end{bmatrix} \succeq 0\]

Applying S-procedure (finally!) we have

\[- \begin{bmatrix} A^T P + PA + \alpha P & P \\ P & 0 \end{bmatrix} \succeq \tau \begin{bmatrix} \gamma^2 & 0 \\ 0 & -I \end{bmatrix} \succeq 0\]

Overall,

\[- \begin{bmatrix} A^T P + PA + \alpha P + \tau \gamma^2 I & P \\ P & -\tau I \end{bmatrix} \succeq 0, \quad P \succeq 0, \quad \tau > 0\]
Consider a system \( G: \dot{x} = Ax + Bu \)

Determine a feedback gain matrix \( K \) such that \( u = Kx \) renders the closed loop system stable.

Closed loop system: \( \dot{x} = (A + BK)x \).

Goal: Determine \( K \) such that \( A + BK \) is Hurwitz.
LMIs in state feedback control

Closed loop system: \( \dot{x} = (A + BK)x \).

- Goal: Determine \( K \) such that \( A + BK \) is Hurwitz.

**Lyapunov stability:** A matrix \( A \) is stable if and only if there exists \( P = P^\top \succeq 0 \) such that

\[
A^\top P + PA \preceq 0
\]

**Equivalent representation:** Multiply by \( P^{-1} \) from the left and right:

\[
P^{-1}A^\top PP^{-1} + P^{-1}PAP^{-1} \preceq 0
\]

and set \( Q = P^{-1} \). We then have

\[
QA^\top + AQ \preceq 0
\]
**LMIs in state feedback control**

Closed loop system: \[ \dot{x} = (A + BK)x. \]

- **Goal:** Determine \( K \) such that \( A + BK \) is Hurwitz.

**Lyapunov stability:** A matrix \( A \) is stable **if and only if** there exists \( Q = Q^\top \succeq 0 \) such that

\[
QA^\top + AQ \preceq 0
\]

Enforce this condition with \( A + BK \) in place of \( A \) and determine \( K \) and \( Q \):

\[
Q(A + BK)^\top + (A + BK)Q \preceq 0
\]

which leads to

\[
QA^\top + (QK^\top)B^\top + AQ + B(QK) \preceq 0
\]
LMIs in state feedback control

Closed loop system: \( \dot{x} = (A + BK)x \).

- Goal: Determine \( K \) such that \( A + BK \) is Hurwitz.

**Lyapunov stability:** A matrix \( A \) is stable if and only if there exists \( Q = Q^\top \succeq 0 \) such that

\[
QA^\top + AQ \leq 0
\]

We are left with this condition which is not nice!

\[
QA^\top + (QK^\top)B^\top + AQ + B(QK) \leq 0
\]

Setting \( Z = KQ \) we have

\[
QA^\top + Z^\top B^\top + AQ + BZ \leq 0
\]

Solve this LMI to determine \( Q \) and \( Z \) and then compute \( K = ZQ^{-1} \).
LMIs in optimization

Consider the following optimization program

\[
\text{min } c^\top x \\
\text{(SDP)}: \quad \text{subject to: } x_1A_1 + x_2A_2 + \cdots + x_nA_n \preceq B
\]

where the matrices \((A_1, \ldots, A_n, B)\) are all symmetric.

- We could also have equality constraints
- Optimization over LMI constraints

Why is this class of optimization programs interesting?

- Semidefinite programming (SDP)
- Many control analysis and synthesis problems can be written as SDPs
- Most of the problems presented so far can be written as SDPs
Semidefinite programming

Primal SDP problem:

\[
\begin{align*}
\min & \quad c^\top x \\
\text{subject to:} & \quad x_1 A_1 + x_2 A_2 + \cdots + x_n A_n \preceq B
\end{align*}
\]

where the matrices \((A_1, \ldots, A_n, B)\) are all symmetric.

Lagrangian:

\[
\mathcal{L}(x, \Lambda) = c^\top x + \sum_i \langle \Lambda, A_i \rangle x_i - \langle \Lambda, B \rangle,
\]

where \(\langle X, Y \rangle = \text{trace}(X^\top Y) = \sum_{i,j} X_{ij} Y_{ij}\).

This fact relies on the following property (dual cone for semidefinite matrices):

\[
\left\{ x \in \mathbb{S}^n \mid \text{trace}(X^\top Y) \geq 0, \forall Y \succeq 0 \right\} = \{ X \in \mathbb{S}^n \mid X \succeq 0 \}.
\]
Semidefinite programming

Primal SDP problem:

$$\min \quad c^\top x$$

subject to: $$x_1 A_1 + x_2 A_2 + \cdots + x_n A_n \preceq B$$

where the matrices $$(A_1, \ldots, A_n, B)$$ are all symmetric.

Lagrangian:

$$\mathcal{L}(x, \Lambda) = c^\top x + \sum_i \langle \Lambda, A_i \rangle x_i - \langle \Lambda, B \rangle$$

$$= \sum_i \left( c_i + \langle \Lambda, A_i \rangle \right) x_i - \langle \Lambda, B \rangle$$

Dual function:

$$g(\lambda) = \begin{cases} 
-\langle \Lambda, B \rangle & \text{if } c_i + \langle \Lambda, A_i \rangle = 0 \text{ for } i = 1 \ldots n \\
-\infty & \text{otherwise}
\end{cases}$$
Semidefinite programming

Primal SDP problem:

$$\min c^\top x$$

subject to: $$x_1 A_1 + x_2 A_2 + \cdots x_n A_n \preceq B$$

where the matrices $$(A_1, \ldots, A_n, B)$$ are all symmetric.

Dual function:

$$g(\lambda) = \begin{cases} -\langle \Lambda, B \rangle & \text{if } c_i + \langle \Lambda, A_i \rangle = 0 \text{ for } i = 1 \ldots n \\ -\infty & \text{otherwise} \end{cases}$$

The dual problem:

$$\max -\langle B, \Lambda \rangle$$

subject to: $$\langle A_i, \Lambda \rangle = -c_i, \text{ for all } i$$

$$\Lambda \succeq 0$$
Semidefinite programming

**Primal SDP problem:**

$$\min \quad c^\top x$$

subject to:  
$$x_1 A_1 + x_2 A_2 + \cdots + x_n A_n \preceq B$$

**The dual problem:**

$$\max \quad -\langle B, \Lambda \rangle$$

subject to:  
$$\langle A_i, \Lambda \rangle = -c_i, \text{ for all } i$$

$$\Lambda \succeq 0$$

**Weak duality:**

$$p^* - d^* = c^\top x + \langle B, \Lambda \rangle$$ \hspace{1cm} \text{[primal feasibility]}$$

$$\geq c^\top x + \sum_i \langle A_i, \Lambda \rangle x_i$$ \hspace{1cm} \text{[dual feasibility]}$$

$$= \sum_i c_i x_i - \sum_i c_i x_i = 0$$
Semidefinite programming

Primal SDP problem:
\[
\begin{align*}
\text{min} & \quad c^\top x \\
\text{subject to:} & \quad x_1 A_1 + x_2 A_2 + \cdots + x_n A_n \preceq B
\end{align*}
\]

The dual problem:
\[
\begin{align*}
\text{max} & \quad -\langle B, \Lambda \rangle \\
\text{subject to:} & \quad \langle A_i, \Lambda \rangle = -c_i, \text{ for all } i \\
\Lambda & \succeq 0
\end{align*}
\]

Weak duality: \( p^* - d^* \geq 0 \)

Strong duality:
Also true under Slater’s condition (constraint qualification). Constraints in the primal need to be satisfied with \( \prec \) instead of \( \preceq \).
What was this lecture about?

All about optimization ...

- Convex optimization programs are “nice”
  - Local solution = Global solution
- Recognizing convex functions and sets
  - If you do this life is easier; you can recast seemingly difficult problems to convex ones
- Formulating dual optimization programs
  - Sometimes much easier than the primal ones (think of MILPs, SDPs)
- Linear matrix inequalities
  - More difficult but appear in many stability and control design problems

Don’t forget the “big picture” ...

- Determine controllers via optimization programs
- Allows trading between different objectives
- Allows enforcing constraints
- Efficient alternative to PID and LQR - price to pay is computation!
Thank you! Questions?

Contact at:

kostas.margellos@eng.ox.ac.uk