Linear Matrix Inequalities in Control

Carsten Scherer
Delft Center for Systems and Control (DCSC)
Delft University of Technology
The Netherlands

Siep Weiland
Department of Electrical Engineering
Eindhoven University of Technology
The Netherlands

Optimization and Control

Classically optimization and control are highly intertwined:

- Optimal control (Pontryagin/Bellman)
- LQG-control or H_2-control
- H_∞-synthesis and robust control
- Model Predictive Control

Main Theme

View control input signal and/or feedback controller as decision variable of an optimization problem.

Desired specs are imposed as constraints on controlled system.

Sketch of Issues

- How to distinguish easy from difficult optimization problems?
- What are the **consequences of convexity** in optimization?
- What is **robust optimization**?
- Which **performance measures** can be incorporated?
- How can **controller synthesis** be convexified?
- How can we check **robustness** by convex optimization?
- What are the limits for the synthesis of **robust controllers**?
- How can we perform systematic **gain-scheduling**?

Outline

- From Optimization to Convex Semi-Definite Programming
- Convex Sets and Convex Functions
- Linear Matrix Inequalities (LMIs)
- Truss-Topology Design
- LMIs and Stability
- A First Glimpse at Robustness

Optimization

The ingredients of any optimization problem are:

- A universe of **decisions** $x \in \mathcal{X}$
- A subset $\mathcal{S} \subset \mathcal{X}$ of **feasible** decisions
- A **cost function** or **objective function** $f : \mathcal{S} \to \mathbb{R}$

Optimization Problem/Programming Problem

Find a feasible decision $x \in \mathcal{S}$ with the least possible cost $f(x)$.

In short such a problem is denoted as

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathcal{S} \end{array}$$

Questions in Optimization

- What is the least possible cost? Compute the **optimal value**

$$f_{\text{opt}} := \inf_{x \in \mathcal{S}} f(x) \geq -\infty.$$

Convention: If $\mathcal{S} = \emptyset$ then $f_{\mathrm{opt}} = +\infty$.

If $f_{\text{opt}} = -\infty$ the problem is said to be **unbounded from below**.

- How can we compute **almost optimal solutions**? For any chosen positive absolute error ε, determine

$$x_\varepsilon \in \mathcal{S} \ \text{ with } \ f_{\text{opt}} \leq f(x_\varepsilon) \leq f_{\text{opt}} + \varepsilon.$$

By definition of the infimum such an x_ε does exist for all $\varepsilon > 0$.

Questions in Optimization

- Is there an **optimal solution**? Does there exist

$$x_{\text{opt}} \in \mathcal{S} \quad \text{with} \quad f(x_{\text{opt}}) = f_{\text{opt}}?$$

  If yes, the **minimum is attained** and we write

$$f_{\text{opt}} = \min_{x \in \mathcal{S}} f(x).$$

  Set of all optimal solutions is denoted as

$$\arg\min_{x \in \mathcal{S}} f(x) = \{x \in \mathcal{S} : \ f(x) = f_{\text{opt}}\}.$$

- Is optimal solution **unique**?

Recap: Infimum and Minimum of Functions

Any $f : \mathcal{S} \to \mathbb{R}$ has **infimum** $l \in \mathbb{R} \cup \{-\infty\}$ denoted as $\inf_{x \in \mathcal{S}} f(x)$.

The infimum is uniquely defined by the following properties:

- $l \leq f(x)$ for all $x \in \mathcal{S}$.
- l finite: For all $\epsilon > 0$ exists $x \in \mathcal{S}$ with $f(x) \leq l + \epsilon$.
  $l = -\infty$: For all $\epsilon > 0$ exists $x \in \mathcal{S}$ with $f(x) \leq -\epsilon$.

If there exists $x_0 \in \mathcal{S}$ with $f(x_0) = \inf_{x \in \mathcal{S}} f(x)$ we say that f attains its **minimum** on $\mathcal{S}$ and write $l = \min_{x \in \mathcal{S}} f(x)$.

If existing the minimum is uniquely defined through the properties:

- $l \leq f(x)$ for all $x \in \mathcal{S}$.
- There exists some $x_0 \in \mathcal{S}$ with $f(x_0) = l$.

Nonlinear Programming (NLP)

With decision universe $\mathcal{X} = \mathbb{R}^n$, the feasible set $\mathcal{S}$ is often defined by **constraint functions** $g_1 : \mathcal{X} \to \mathbb{R}, \ldots, g_m : \mathcal{X} \to \mathbb{R}$ as

$$\mathcal{S} = \{x \in \mathcal{X} : \; g_1(x) \leq 0, \ldots, g_m(x) \leq 0\}.$$

The **general nonlinear optimization problem** is formulated as

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathcal{X}, \; g_1(x) \leq 0, \ldots, g_m(x) \leq 0. \end{array}$$

By exploiting particular properties of $f, \; g_1, \ldots, g_m$ (e.g. smoothness), optimization algorithms typically allow to iteratively compute locally optimal solutions x_0: There exists an $\varepsilon > 0$ such that x_0 is optimal on

$$\mathcal{S} \cap \{x \in \mathcal{X} : \; \|x - x_0\| \leq \varepsilon\}.$$

Example: Quadratic Program

$f : \mathbb{R}^n \to \mathbb{R}$ is **quadratic** iff there exists a symmetric matrix P with

$$f(x) = \begin{pmatrix} 1 \\ x \end{pmatrix}^T P \begin{pmatrix} 1 \\ x \end{pmatrix}.$$

Quadratically constrained quadratic program

$$\text{minimize} \quad \begin{pmatrix} 1 \\ x \end{pmatrix}^T P_0 \begin{pmatrix} 1 \\ x \end{pmatrix}$$

$$\text{subject to} \quad x \in \mathbb{R}^n, \quad \begin{pmatrix} 1 \\ x \end{pmatrix}^T P_k \begin{pmatrix} 1 \\ x \end{pmatrix} \leq 0, \quad k = 1, \ldots, m$$

where $P_0, P_1, \ldots, P_m \in \mathbb{S}^{n+1}$.

Linear Programming (LP)

With the decision vectors $x = (x_1 \;\cdots\; x_n)^T \in \mathbb{R}^n$ consider the problem

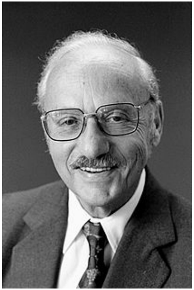
$$\begin{array}{lrcl} \text{minimize} & c_1 x_1 + \cdots + c_n x_n & & \\ \text{subject to} & a_{11} x_1 + \cdots + a_{1n} x_n & \leq & b_1 \\ & & \vdots & \\ & a_{m1} x_1 + \cdots + a_{mn} x_n & \leq & b_m \end{array}$$

The **cost is linear** and the set of feasible decisions is defined by finitely many **affine inequality constraints**.

Simplex algorithms or **interior point methods** allow to efficiently

1. decide whether the constraint set is feasible (value $< \infty$).
2. decide whether problem is bounded from below (value $> -\infty$).
3. compute an (always existing) optimal solution if 1. & 2. are true.

Linear Programming (LP)



Major early contributions in 1940s by

- **George Dantzig** (simplex method)
- **John von Neumann** (duality)
- **Lenoid Kantorovich** (economics applications)

Leonid Khachiyan proved polynomial-time solvability in 1979.
Narendra Karmakar introduce an interior point method in 1984.

Has numerous applications for example in economical planning problems (business management, flight-scheduling, resource allocation, finance).

LPs have spurred the development of optimization theory and appear as subproblem in many optimization algorithms.

Recap

For a real or complex matrix A the inequality $A \preccurlyeq 0$ means that A is **Hermitian** and **negative semi-definite**.

- A is defined to be Hermitian if $A = A^* = \bar{A}^T$. If A is real then this amounts to $A = A^T$ and A is then called symmetric.

  All eigenvalues of Hermitian matrices are real.

- By definition A is negative semi-definite if $A = A^*$ and

  $$x^* A x \leq 0 \ \text{ for all complex vectors } \ x \neq 0.$$

  A is negative semi-definite iff all its eigenvalues are non-positive.

- $A \preccurlyeq B$ means by definition: A, B are Hermitian and $A - B \preccurlyeq 0$.
- $A \prec B$, $A \preccurlyeq B$, $A \succcurlyeq B$, $A \succ B$ defined/characterized analogously.

Recap

Let A be partitioned with square diagonal blocks as

$$A = \begin{pmatrix} A_{11} & \cdots & A_{1m} \\ \vdots & \ddots & \vdots \\ A_{m1} & \cdots & A_{mm} \end{pmatrix}.$$

Then

$$A \prec 0 \text{ implies } A_{11} \prec 0, \ldots, A_{mm} \prec 0.$$

Prototypical Proof. Choose any vector $z_i \neq 0$ of length compatible with the size of A_{ii}. Define $z = (0, \ldots, 0, z_i^T, 0, \ldots, 0)^T \neq 0$ with the zeros blocks compatible in size with $A_{11}, \ldots, A_{mm}$. This construction implies $z^T A z = z_i^T A_{ii} z_i$. Since $A \prec 0$, we infer $z^T A z < 0$. Therefore, $z_i^T A_{ii} z_i < 0$. Since $z_i \neq 0$ was arbitrary we infer $A_{ii} \prec 0$.

Semi-Definite Programming (SDP)

Let us now assume that the constraint functions $G_1, \ldots, G_m$ map $\mathcal{X}$ into the set of **symmetric matrices**, and define the feasible set $\mathcal{S}$ as

$$\mathcal{S} = \{x \in \mathcal{X} : \; G_1(x) \preccurlyeq 0, \ldots, G_m(x) \preccurlyeq 0\}.$$

The **general semi-definite program (SDP)** is formulated as

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathcal{X}, \; G_1(x) \preccurlyeq 0, \ldots, G_m(x) \preccurlyeq 0. \end{array}$$

- This includes NLPs as a special case.
- Is called **convex** if f and $G_1, \ldots, G_m$ are **convex**.
- Is called **linear matrix inequality (LMI) optimization problem** or **linear SDP** if f and $G_1, \ldots, G_m$ are **affine**.

Outline

- From Optimization to Convex Semi-Definite Programming
- Convex Sets and Convex Functions
- Linear Matrix Inequalities (LMIs)
- Truss-Topology Design
- LMIs and Stability
- A First Glimpse at Robustness

Recap: Affine Sets and Functions

The **set** $\mathcal{S}$ in the vector space $\mathcal{X}$ is **affine** if

$$\lambda x^1 + (1-\lambda)x^2 \in \mathcal{S} \ \text{ for all } \ x^1, x^2 \in \mathcal{S},\ \lambda \in \mathbb{R}.$$

The **matrix-valued function** F defined on $\mathcal{S}$ is **affine** if the domain of definition $\mathcal{S}$ is affine and if

$$F(\lambda x^1 + (1-\lambda)x^2) = \lambda F(x^1) + (1-\lambda)F(x^2)$$
$$\text{for all } \ x^1, x^2 \in \mathcal{S},\ \lambda \in \mathbb{R}.$$

For points $x^1, x^2 \in \mathcal{S}$ recall that

- $\{\lambda x^1 + (1-\lambda)x^2 :\ \lambda \in \mathbb{R}\}$ is the **line** through x^1, x^2
- $\{\lambda x^1 + (1-\lambda)x^2 :\ \lambda \in [0,1]\}$ is the **line segment** between x^1, x^2

Recap: Convex Sets and Functions

The **set** $\mathcal{S}$ in the vector space $\mathcal{X}$ is **convex** if

$$\lambda x^1 + (1-\lambda)x^2 \in \mathcal{S} \ \text{ for all } \ x^1, x^2 \in \mathcal{S},\ \lambda \in (0,1).$$

The **symmetric-valued function** F defined on $\mathcal{S}$ is **convex** if the domain of definition $\mathcal{S}$ is convex and if

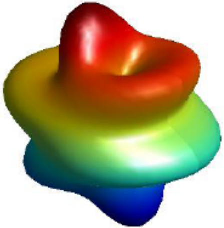
$$F(\lambda x^1 + (1-\lambda)x^2) \preccurlyeq \lambda F(x^1) + (1-\lambda)F(x^2)$$
$$\text{for all } \ x^1, x^2 \in \mathcal{S},\ \lambda \in (0,1).$$

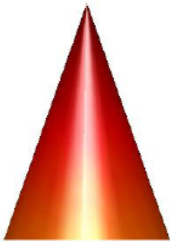
F is **strictly convex** if $\preccurlyeq$ can be replaced by $\prec$.

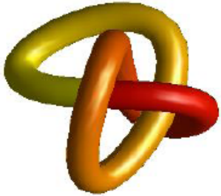
If F is real-valued, the inequalities $\preccurlyeq$ and $\prec$ are the same as the usual inequalities $\leq$ and $<$ between real numbers. Therefore our definition captures the usual one for **real-valued** functions!

Carsten Scherer TU Delft Siep Weiland TU/e technische universiteit eindhoven

Examples of Convex and Non-Convex Sets







Examples of Convex Sets

The intersection of any family of convex sets is convex.

- With $a \in \mathbb{R}^n \setminus \{0\}$ and $b \in \mathbb{R}$, the **hyperplane** $\{x \in \mathbb{R}^n : a^T x = b\}$ is affine while the **half-space** $\{x \in \mathbb{R}^n : \ a^T x \leq b\}$ is convex.

- The intersection of finitely many hyperplanes and half-planes defines a **polyhedron**. Any polyhedron is convex and can be described as

$$\{x \in \mathbb{R}^n : \ Ax \leq b, \quad Dx = e\}$$

  with suitable matrices A, D, vectors b, e. A compact polyhedron is said to be a **polytope**.

- The set of negative semi-definite/negative definite matrices is convex.

Convex Combination and Convex Hull

- $x \in \mathcal{X}$ is **convex combination** of $x^1, \ldots, x^l \in \mathcal{X}$ if

$$x = \sum_{k=1}^{l} \lambda_k x^k \quad \text{with} \quad \lambda_k \geq 0, \; \sum_{k=1}^{l} \lambda_k = 1.$$

Convex combination of a convex combination is a convex combination.

- The **convex hull** $\text{co}(\mathcal{S})$ of any subset $\mathcal{S} \subset \mathcal{X}$ is defined in one of the following equivalent fashions:

   1. Set of all convex combinations of points in $\mathcal{S}$.
   2. Intersection of all convex sets that contain $\mathcal{S}$.

For arbitrary $\mathcal{S}$ the convex hull $\text{co}(\mathcal{S})$ is convex.

Carsten Scherer TUDelft Siep Weiland TU/e technische universiteit eindhoven

Explict Description of Polytopes

Any point in the convex hull of the finite set $\{x^1, \ldots, x^l\}$ is given by a (not necessarily unique) convex combination of generators $x^1, \ldots, x^l$.

The convex hull of finitely many points $\text{co}\{x^1, \ldots, x^l\}$ is a polytope. Any polytope can be represented in this way.

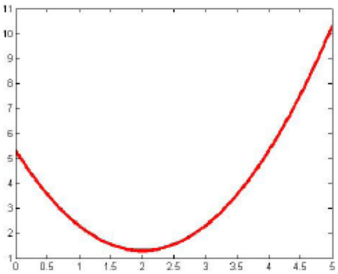
In this fashion one can **explicitly** describe polytopes, in contrast to an implicit description such as $\{x \in \mathbb{R}^n : \ Ax \leq b\}$. Note that the implicit description is often preferable for reasons of computational complexity.

Example: $\{x \in \mathbb{R}^n : a \leq x \leq b\}$ is defined by $2n$ inequalities but requires 2^n generators for its description as a convex hull.

Examples of Convex and Non-Convex Functions

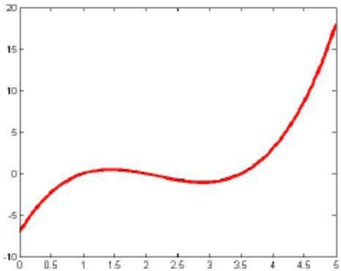
Convex functions

- $f(x) = ax^2 + bx + c$ convex if $a > 0$
- $f(x) = |x|$
- $f(x) = \|x\|$
- $f(x) = \sin x$ on $[\pi, 2\pi]$



Non-convex functions

- $f(x) = x^3$ on $\mathbb{R}$
- $f(x) = -|x|$
- $f(x) = \sqrt{x}$ on $\mathbb{R}_+$
- $f(x) = \sin x$ on $[0, \pi]$



On Checking Convexity of Functions

All real-valued or Hermitian-valued **affine** functions are convex.

It is often not simple to verify whether a non-affine function is convex. The following fact (for $\mathcal{S} \subset \mathbb{R}^n$ with interior points) might help.

The C^2-map $f : \mathcal{S} \to \mathbb{R}$ is convex iff $\partial^2 f(x) \succcurlyeq 0$ for all $x \in \mathcal{S}$.

It is not easy to find convexity tests for Hermitian-valued function in the literature. The following reduction to real-valued functions often helps.

The Hermitian-valued map F defined on $\mathcal{S}$ is convex iff

$$\mathcal{S} \ni x \to z^* F(x) z \in \mathbb{R}$$

is convex for all complex vectors $z \in \mathbb{C}^m$.

Convex Constraints

Suppose that F defined on $\mathcal{S}$ is convex. For all Hermitian H the strict or non-strict **sub-level sets**

$$\{x \in \mathcal{S} : \; F(x) \prec H\} \quad \text{and} \quad \{x \in \mathcal{S} : \; F(x) \preccurlyeq H\}$$

"of level H" are convex.

Note that the converse is in general not true!

- The feasible set of the convex SDP on slide 13 is convex.
- Convex sets are typically described as the finite intersection of such sub-level sets. If the involved functions are even affine this is often called an LMI representation.
- Convexity is necessary for a set to have an LMI representation.

Example: Quadratic Functions

The quadratic function

$$f(x) = \begin{pmatrix} 1 \\ x \end{pmatrix}^T \begin{pmatrix} q & s^T \\ s & R \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix} = q + 2s^T x + x^T R x$$

is convex iff R is positive semidefinite.

The zero sub-level set of a convex quadratic function is a **half-space** ($R = 0$) or an **ellipsoid**. The ellipsoid is compact if $R \succ 0$.

For later use: $f(x) = q + 2s^T x + x^T R x$ with $R = R^T$ is **nonnegative** for all $x \in \mathbb{R}^n$ iff its defining matrix satisfies

$$\begin{pmatrix} q & s^T \\ s & R \end{pmatrix} \succcurlyeq 0.$$

Important Properties

Property 1: Jensen's Inequality.

If F defined on $\mathcal{S}$ is convex then for all $x^1, \ldots, x^l \in \mathcal{S}$ and $\lambda_1, \ldots, \lambda_l \geq 0$ with $\lambda_1 + \cdots + \lambda_l = 1$ we infer $\lambda_1 x^1 + \cdots + \lambda_l x^l \in \mathcal{S}$ and

$$F(\lambda_1 x^1 + \cdots + \lambda_l x^l) \preccurlyeq \lambda_1 F(x^1) + \cdots + \lambda_l F(x^l).$$

Source of many inequalities in mathematics! Proof: cc of cc is cc!

Property 2. If F and G define on $\mathcal{S}$ are convex then $F + G$ and αF for $\alpha \geq 0$ as well as

$$\mathcal{S} \ni x \to \lambda_{\text{max}}(F(x)) \in \mathbb{R}$$

are all convex. If F and G are scalar-valued then $\max\{F, G\}$ is convex.

There are many other operations for functions that preserve convexity.

A Key Consequence

Let F be convex. Then

$$F(x) \prec 0 \ \text{ for all } \ x \in \text{co}\{x^1, \ldots, x^l\}$$

if and only if

$$F(x^k) \prec 0 \ \text{ for all } \ k = 1, \ldots, l.$$

Proof. We only need to show "if". Choose $x \in \text{co}\{x^1, \ldots, x^l\}$. Then there exists $\lambda_1 \geq 0, \ldots, \lambda_l \geq 0$ that sum up to one with

$$x = \lambda_1 x^1 + \cdots + \lambda_l x^l.$$

By convexity and Jensen's inequality we infer

$$F(x) \preccurlyeq \lambda_1 F(x^1) + \cdots + \lambda_l F(x^l) \prec 0$$

since the set of negative definite matrices is convex.

General Remarks on Convex Programs

- Solvers for general nonlinear programs typically determine **local** optimal solutions. There are no guarantees for **global optimality**.
- Main feature of convex programs:

  Locally optimal solutions are globally optimal.

  Convexity alone **neither** guarantees that the optimal value is finite, **nor** that there exists an optimal solution/efficient solution algorithms.
- In general convex programs can be solved with guarantees on accuracy if one can compute (sub)gradients of objective/constraint functions.
- Strictly feasible linear semi-definite programs are convex and **can** be solved very efficiently, with guarantees on accuracy at termination.

Outline

- From Optimization to Convex Semi-Definite Programming
- Convex Sets and Convex Functions
- Linear Matrix Inequalities (LMIs)
- Truss-Topology Design
- LMIs and Stability
- A First Glimpse at Robustness

Linear Matrix Inequalities (LMIs)

With the decision vectors $x = (x_1 \ \cdots \ x_n)^T \in \mathbb{R}^n$ a system of LMIs is

$$A_0^1 + x_1 A_1^1 + \cdots + x_n A_n^1 \preccurlyeq 0$$
$$\vdots$$
$$A_0^m + x_1 A_1^m + \cdots + x_n A_n^m \preccurlyeq 0$$

where A_0^i, $A_1^i, \ldots, A_n^i$, $i = 1, \ldots, m$, are real symmetric data matrices.

LMI feasibility problem: Test whether there exist $x_1, \ldots, x_n$ that render the LMIs satisfied.

LMI optimization problem: Minimize $c_1 x_1 + \cdots + c_n x_n$ over all $x_1, \ldots, x_n$ that satisfy the LMIs.

Only simple cases can be treated analytically $\rightarrow$ Numerical techniques.

LMI Optimization Problems

minimize	$c_1 x_1 + \cdots + c_n x_n$
subject to	$A_0^i + x_1 A_1^i + \cdots + x_n A_n^i \preccurlyeq 0, \quad i = 1, \ldots, m$

- Natural generalization of LPs with inequalities defined by the cone of positive semi-definite matrices. Considerable richer class than LPs.

- The i-th constraint can be equivalently expressed as

$$\lambda_{\max}(A_0^i + x_1 A_1^i + \cdots + x_n A_n^i) \leq 0.$$

- **Interior-point** or **bundle** methods allow to effectively decide about feasibility/boundedness and to determine **almost** optimal solutions.

- **Must be strictly feasible:** There exists some decision x for which the constraint inequalities are strictly satisfied.

Testing Strict Feasbility

Introduce the auxiliary variable $t \in \mathbb{R}$ and consider

$$A_0^1 + x_1 A_1^1 + \cdots + x_n A_n^1 \preccurlyeq tI$$
$$\vdots$$
$$A_0^m + x_1 A_1^m + \cdots + x_n A_n^m \preccurlyeq tI.$$

Find infimal value t_* of t over these LMI constraints.

- This problem is strictly feasible. We can hence compute t_* efficiently.
- If t_* is **negative** then original problem **is** strictly feasible.
- If t_* is **non-negative** then original problem **is not** strictly feasible.

LMI Optimization Problems

Developments

- **Bellman/Fan** initialized derivation of optimality conditions (1963)
- **Jan Willems** coined terminology LMI and revealed relation to dissipative dynamical systems (1971/72)
- **Nesterov/Nemirovski** exhibited essential feature (self-concordance) for existence of polynomial-time solution algorithm (1988)
- Interior-point methods: Alizadeh (1992), Kamath/Karmarkar (1992)

Suggested books: Boyd/El Ghaoui (1994), El Ghaoui/Niculescu (2000), Ben-Tal/Nemiorvski (2001), Boyd/Vandenberghe (2004)

What are LMIs good for?

- Many engineering optimization problem can be (often but not always easily) **translated** into LMI problems.
- Various computationally difficult optimization problems can be effectively **approximated** by LMI problems.
- In practice the description of the data is affected by uncertainty. **Robust optimization** problems can be handled/approximated by standard LMI problems.

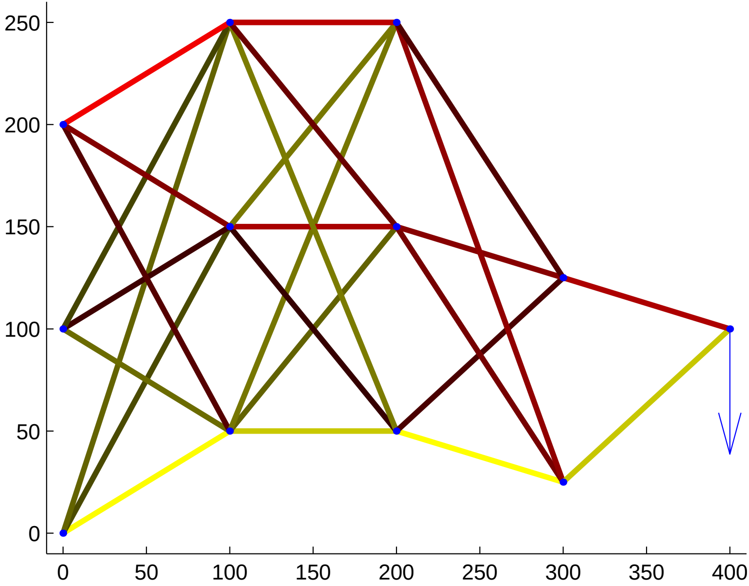
In this course

How can we solve (robust) control problems with LMIs?

Outline

- From Optimization to Convex Semi-Definite Programming
- Convex Sets and Convex Functions
- Linear Matrix Inequalities (LMIs)
- Truss-Topology Design
- LMIs and Stability
- A First Glimpse at Robustness

Truss Topology Design



Example: Truss Topology Design

- Connect nodes with N bars of length $l = \text{col}(l_1, \ldots, l_N)$ (fixed) and cross-sections $x = \text{col}(x_1, \ldots, x_N)$ **(to-be-designed)**.
- Impose **bounds** $a_k \leq x_k \leq b_k$ on cross-section and $l^T x \leq v$ on total volume (weight). Abbreviate $a = \text{col}(a_1, \ldots, a_N)$, $b = \text{col}(b_1, \ldots, b_N)$.
- If applying external forces $f = \text{col}(f_1, \ldots, f_M)$ (fixed) on nodes the construction reacts with the node displacement $d = \text{col}(d_1, \ldots, d_M)$.

  Mechanical model: $A(x)d = f$ where $A(x)$ is the stiffness matrix which depends linearly on x and has to be **positive semi-definite**.
- Goal is to **maximize** stiffness, for example by minimizing the elastic stored energy $f^T d$.

Example: Truss Topology Design

Find $x \in \mathbb{R}^N$ which minimizes $f^T d$ subject to the constraints

$$A(x) \succcurlyeq 0, \quad A(x)d = f, \quad l^T x \leq v, \quad a \leq x \leq b.$$

Features

- **Data:** Scalar v, vectors f, a, b, l, and symmetric matrices $A_1, \ldots, A_N$ which define the linear mapping $A(x) = A_1 x_1 + \cdots + A_N x_N$.
- **Decision variables:** Vectors x and d.
- **Objective function:** $d \to f^T d$ which happens to be linear.
- **Constraints:** Semi-definite constraint $A(x) \succcurlyeq 0$, nonlinear equality constraint $A(x)d = f$, and linear inequality constraints $l^T x \leq v$, $a \leq x \leq b$. Latter interpreted **elementwise**!

From Truss Topology Design to LMI's

Render LMI inequality strict. Equality constraint $A(x)d = f$ allows to **eliminate** d which results in

$$\begin{array}{ll} \text{minimize} & f^T A(x)^{-1} f \\ \text{subject to} & A(x) \succ 0, \;\; l^T x \leq v, \;\; a \leq x \leq b. \end{array}$$

Push objective to constraints with auxiliary variable:

$$\begin{array}{ll} \text{minimize} & \gamma \\ \text{subject to} & \gamma > f^T A(x)^{-1} f, \;\; A(x) \succ 0, \;\; l^T x \leq v, \;\; a \leq x \leq b. \end{array}$$

Trouble: **Nonlinear** inequality constraint $\gamma > f^T A(x)^{-1} f$.

Recap: Congruence Transformations

Given a Hermitian matrix A and a square non-singular matrix T,

$$A \to T^* A T$$

is called a **congruence transformation** of A.

If T is square and non-singular then

$$A \prec 0 \quad \text{if and only if} \quad T^* A T \prec 0.$$

The following more general statement is also easy to remember.

If A is Hermitian and T is nonsingular, the matrices A and $T^* A T$ have the **same number** of negative, zero, positive eigenvalues.

What is true if T is not square? ... if T has full column rank?

Recap: Schur-Complement Lemma

The Hermitian block matrix $\begin{pmatrix} Q & S \\ S^T & R \end{pmatrix}$ is negative definite

if and only if

$$Q \prec 0 \;\; \text{and} \;\; R - S^T Q^{-1} S \prec 0$$

if and only if

$$R \prec 0 \;\; \text{and} \;\; Q - S R^{-1} S^T \prec 0.$$

Proof. First equivalence follows from

$$\begin{pmatrix} I & 0 \\ -S^T Q^{-1} & I \end{pmatrix} \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \begin{pmatrix} I & -Q^{-1} S \\ 0 & I \end{pmatrix} = \begin{pmatrix} Q & 0 \\ 0 & R - S^T Q^{-1} S \end{pmatrix}.$$

The proof reveals a more general relation between the number of negative, zero, positive eigenvalues of the three matrices.

From Truss Topology Design to LMI's

Render LMI inequality strict. Equality constraint $A(x)d = f$ allows to **eliminate** d which results in

$$\begin{array}{ll} \text{minimize} & f^T A(x)^{-1} f \\ \text{subject to} & A(x) \succ 0, \quad l^T x \leq v, \quad a \leq x \leq b. \end{array}$$

Push objective to constraints with auxiliary variable:

$$\begin{array}{ll} \text{minimize} & \gamma \\ \text{subject to} & \gamma > f^T A(x)^{-1} f, \quad A(x) \succ 0, \quad l^T x \leq v, \quad a \leq x \leq b. \end{array}$$

Linearize with Schur lemma to equivalent LMI problem

$$\begin{array}{ll} \text{minimize} & \gamma \\ \text{subject to} & \begin{pmatrix} \gamma & f^T \\ f & A(x) \end{pmatrix} \succ 0, \quad l^T x \leq v, \quad a \leq x \leq b. \end{array}$$

Yalmip-Coding: Truss Toplogy Design

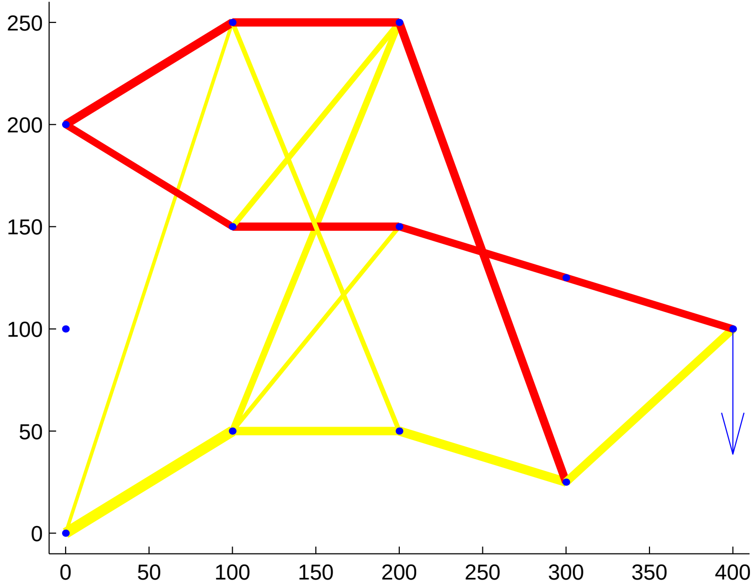
$$\begin{array}{ll} \text{minimize} & \gamma \\ \text{subject to} & \begin{pmatrix} \gamma & f^T \\ f & A(x) \end{pmatrix} \succ 0, \quad l^T x \leq v, \quad a \leq x \leq b. \end{array}$$

Suppose $A(x) = \sum_{k=1}^{N} x_k m_k m_k^T$ with vectors m_k collected in matrix M.

The following code with Yalmip commands solves LMI problem:

```
gamma=sdpvar(1,1); x=sdpvar(N,1,'full');
lmi=set([gamma f';f M*diag(x)*M']);
lmi=lmi+set(l'*x<=v);
lmi=lmi+set(a<=x<=b);
options=sdpsettings('solver','sedumi');
solvesdp(lmi,gamma,options); s=double(x);
```

Result: Truss Toplogy Design



Quickly Accessible Software

General purpose Matlab interface **Yalmip**:

- Free code developed by J. Löfberg and accessible at

  `http://control.ee.ethz.ch/~joloef/yalmip.msql`

- Can use usual Matlab-syntax to define optimization problem.

  Is extremely easy to use and very versatile. Highly recommended!

- Provides access to a whole suite of public and commercial optimization solvers, including fastest available dedicated LMI-solvers.

Matlab **LMI-Toolbox** for dedicated control applications. Has recently been integrated into new Robust Control Toolbox.

Outline

- From Optimization to Convex Semi-Definite Programming
- Convex Sets and Convex Functions
- Linear Matrix Inequalities (LMIs)
- Truss-Topology Design
- LMIs and Stability
- A First Glimpse at Robustness

General Formulation of LMI Problems

Let $\mathcal{X}$ be a finite-dimensional real vector space. Suppose the mappings

$c : \mathcal{X} \to \mathbb{R}, \quad F : \mathcal{X} \to \{\text{symmetric matrices of fixed size}\}$ are **affine.**

LMI feasibility problem: Test existence of $X \in \mathcal{X}$ with $F(X) \prec 0$.

LMI optimization problem: Minimize $f(X)$ over all $X \in \mathcal{X}$ that satisfy the LMI $F(X) \prec 0$.

Translation to standard form: Choose basis $X_1, \ldots, X_n$ of $\mathcal{X}$ and parameterize $X = x_1 X_1 + \cdots + x_n X_n$. For any affine f infer

$$f\Big(\sum_{k=1}^{n} x_k X_k\Big) = f(0) + \sum_{k=1}^{n} x_k [f(X_k) - f(0)].$$

Diverse Remarks

- The **standard basis** of $\mathbb{R}^{p\times q}$ is $X_{(k,l)}$, $k = 1,\ldots,p$, $l = 1,\ldots,q$, where the only nonzero element of $X_{(k,l)}$ is one at position (k,l).

- General **affine equation** constraint can be routinely eliminated - just recall how we can parameterize the solution set of general affine equations. This might be cumbersome and is **not required in Yalmip**.

- Multiple LMI constraints can be collected into one single constraint.

- If $F(X)$ is **linear** in X, then

$$F(X) \prec 0 \;\text{ implies }\; F(\alpha X) \prec 0 \;\text{ for all }\; \alpha > 0.$$

With some solvers this might cause numerical trouble. Avoided by normalization or extra constraints (e.g. by bounding the variables).

Example: Spectral Norm Approximation

For real data matrices A, B, C and some unknown X consider

$$\begin{array}{ll} \text{minimize} & \|AXB - C\| \\ \text{subject to} & X \in \mathcal{S} \end{array}$$

where $\mathcal{S}$ is a matrix subspace reflecting structural constraints.

Key equivalence with Schur:

$$\|M\| < \gamma \iff M^T M \prec \gamma^2 I \iff \begin{pmatrix} \gamma I & M \\ M^T & \gamma I \end{pmatrix} \succ 0.$$

Norm minimization hence equivalent to following LMI problem:

$$\begin{array}{ll} \text{minimize} & \gamma \\ \text{subject to} & X \in \mathcal{S}, \; \begin{pmatrix} \gamma I & AXB - C \\ (AXB - C)^T & \gamma I \end{pmatrix} \succ 0 \end{array}$$

Stability of Dynamical Systems

For dynamical systems one can distinguish many notions of stability.

We will mainly rely on definitions related to the state-space descriptions

$$\dot{x}(t) = Ax(t), \quad \dot{x}(t) = A(t)x(t), \quad \dot{x}(t) = f(t, x(t)), \;\; x(t_0) = x_0$$

which capture the behavior of $x(t)$ for $t \to \infty$ depending on x_0.

Exponential stability means that there exist real constants $a > 0$ (decay rate) and K (peaking constant) such that

$$\|x(t)\| \leq \|x(t_0)\| K e^{-a(t-t_0)} \text{ for all trajectories and } t \geq t_0.$$

K and α are assumed not to depend on t_0 or $x(t_0)$ (uniformity).

Lyapunov theory provides the background for **testing** stability.

Stability of LTI Systems

The linear time-invariant dynamical system

$$\dot{x}(t) = Ax(t)$$

is **exponentially stable** if and only if there exists K with

$$K \succ 0 \quad \text{and} \quad A^T K + KA \prec 0.$$

Two inequalities can be combined as

$$\begin{pmatrix} -K & 0 \\ 0 & A^T K + KA \end{pmatrix} \prec 0.$$

Since the left-hand side depends **affinely** on the matrix variable K, this is indeed a standard strict feasibility test!

Matrix variables are fully supported by Yalmip and LMI-toolbox!

Trajectory-Based Proof of Sufficiency

Choose $\varepsilon > 0$ such that $A^TK + KA + \varepsilon K \prec 0$. Let $x(.)$ be any state-trajectory of the system. Then

$$x(t)^T(A^TK + KA)x(t) + \varepsilon x(t)^T Kx(t) \leq 0 \text{ for all } t \in \mathbb{R}$$

and hence (using $\dot{x}(t) = Ax(t)$)

$$\frac{d}{dt}x(t)^T Kx(t) + \varepsilon x(t)^T Kx(t) \leq 0 \text{ for all } t \in \mathbb{R}$$

and hence (integrating factor $e^{\varepsilon t}$)

$$x(t)^T Kx(t) \leq x(t_0)^T Kx(t_0)e^{-\varepsilon(t-t_0)} \text{ for all } t \in \mathbb{R}, \quad t \geq t_0.$$

Since $\lambda_{\min}(K)\|x\|^2 \leq x^T Kx \leq \lambda_{\max}(K)\|x\|^2$ we can conclude that

$$\|x(t)\| \leq \|x(t_0)\| \sqrt{\frac{\lambda_{\max}(K)}{\lambda_{\min}(K)} e^{-\varepsilon(t-t_0)}} \text{ for } t \geq t_0.$$

Algebraic Proof

Sufficiency. Let $\lambda \in \lambda(A)$. Choose a complex eigenvector $x \neq 0$ with $Ax = \lambda x$. Then the LMI's imply $x^* K x > 0$ and

$$0 > x^*(A^T K + KA)x = \bar{\lambda} x^* K x + x^* K x \lambda = 2\text{Re}(\lambda) x^* K x.$$

This guarantees $\text{Re}(\lambda) < 0$. Therefore all eigenvalues of A are in $\mathbb{C}^-$.

Necessity if A is diagonizable. Suppose all eigenvalues of A are in $\mathbb{C}^-$. Since A is diagonizable there exists a complex nonsingular T with $TAT^{-1} = \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$. Since $\text{Re}(\lambda_k) < 0$ for $k = 1, \ldots, n$ we infer

$$\Lambda^* + \Lambda \prec 0 \ \text{ and hence } \ (T^*)^{-1} A^T T^* + TAT^{-1} \prec 0.$$

If we left-multiply with T^* and right-multiply with T (congruence) we infer

$$A^T (T^* T) + (T^* T) A \prec 0.$$

Hence $K = T^* T \succ 0$ satisfies the LMI's.

Algebraic Proof

Necessity if A is not diagonizable. If A is not diagonizable it can be transformed by similarity into its **Jordan form**: There exists a nonsingular T with $TAT^{-1} = \Lambda + J$ where Λ is diagonal and J has either ones or zeros on the first upper diagonal.

For any $\varepsilon > 0$ one can even choose T_ε with $T_\varepsilon A T_\varepsilon^{-1} = \Lambda + \varepsilon J$. Since Λ has the eigenvalues of A on its diagonal we still infer $\Lambda^* + \Lambda \prec 0$. Therefore it is possible to fix a sufficiently small $\varepsilon > 0$ with

$$0 \succ \Lambda^* + \Lambda + \varepsilon(J^T + J) = (\Lambda + \varepsilon J)^* + (\Lambda + \varepsilon J).$$

As before we can conclude that $K = T_\varepsilon^* T_\varepsilon$ satisfies the LMI's.

Stability of Discrete-Time LTI Systems

The linear time-invariant dynamical system

$$x(t+1) = Ax(t), \quad t = 0, 1, 2, \ldots$$

is **exponentially stable** if and only if there exists K with

$$K \succ 0 \;\; \text{and} \;\; A^T K A - K \prec 0.$$

Recall how "negativity" of $\frac{d}{dt}x(t)^T K x(t)$ in continuous-time leads to

$$A^T K + KA = \begin{pmatrix} I \\ A \end{pmatrix}^T \begin{pmatrix} 0 & K \\ K & 0 \end{pmatrix} \begin{pmatrix} I \\ A \end{pmatrix} \prec 0.$$

Now "negativity" of $x(t+1)^T K x(t+1) - x(t)^T K x(t)$ leads to

$$A^T K A - K = \begin{pmatrix} I \\ A \end{pmatrix}^T \begin{pmatrix} -K & 0 \\ 0 & K \end{pmatrix} \begin{pmatrix} I \\ A \end{pmatrix} \prec 0.$$

Outline

- From Optimization to Convex Semi-Definite Programming
- Convex Sets and Convex Functions
- Linear Matrix Inequalities (LMIs)
- Truss-Topology Design
- LMIs and Stability
- A First Glimpse at Robustness

A First Glimpse at Robustness

With some compact set $\boldsymbol{A} \subset \mathbb{R}^{n \times n}$ consider the family of LTI systems

$$\dot{x}(t) = Ax(t) \text{ with } A \in \boldsymbol{A}.$$

$\boldsymbol{A}$ is said to be **quadratically stable** if there exists K such that

$$K \succ 0 \text{ and } A^T K + KA \prec 0 \text{ for all } A \in \boldsymbol{A}.$$

Why name? $V(x) = x^T K x$ is quadratic Lyapunov function.

Why relevant? Implies that all $A \in \boldsymbol{A}$ are Hurwitz.

Even stronger: Implies, for any piece-wise continuous $A : \mathbb{R} \to \boldsymbol{A}$, exponential stability of the time-varying system

$$\dot{x}(t) = A(t)x(t).$$

Computational Verification

If $\boldsymbol{A}$ has infinitely many elements, testing quadratic stability amounts to verifying the feasibility of an infinite number of LMIs.

Key question: How to reduce to a standard LMI problem?

Let $\boldsymbol{A}$ be the **convex hull** of $\{A_1, \ldots, A_N\}$: For each $A \in \boldsymbol{A}$ there exist coefficients $\lambda_1 \geq 0, \ldots, \lambda_N \geq 0$ with $\lambda_1 + \cdots + \lambda_N = 1$ such that

$$A = \lambda_1 A_1 + \cdots + \lambda_N A_N.$$

If $\boldsymbol{A}$ is the convex hull of $\{A_1, \ldots, A_N\}$ then $\boldsymbol{A}$ is quadratically stable iff there exists some K with

$$K \succ 0 \quad \text{and} \quad A_i^T K + K A_i \prec 0 \quad \text{for all} \quad i = 1, \ldots, N.$$

Proof. Slide 26.

Lessons to be Learnt

- Many interesting engineering problems are LMI problems.
- Variables can live in arbitrary vector space.

  In control: Variables are typically matrices.

  Can involve equation and inequality constraints. Just **check** whether cost function and constraints are affine & verify strict feasibility.
- Translation to input for solution algorithm by parser (e.g. Yalmip).

  Can choose among many efficient LMI solvers (e.g. Sedumi).
- Main trick in removing nonlinearities so far: Schur Lemma.