GENERAL

an introduction to linear matrix inequalities github pages

Published: September 19, 2026 | Index ID: #-

Introduction

Linear Matrix Inequalities (LMIs) have become a cornerstone in modern control theory, optimization, and systems engineering. They offer a unified framework to express a wide range of engineering constraints—stability, performance, robustness, and resource allocation—in a convex, tractable form. Yet, for newcomers, the terminology and the mathematical machinery can feel intimidating. This article presents a gentle, yet comprehensive, introduction to LMIs, explains why they matter, and shows how to leverage GitHub Pages to create interactive, shareable tutorials, code repositories, and documentation that bring LMIs to life for students, researchers, and practitioners alike.

What Are Linear Matrix Inequalities?

At its core, a Linear Matrix Inequality is an expression of the form

F(x) = F € + x •F • + x ,F , + … + x ™F ™ *° 0,

where each **F** is a symmetric matrix of fixed size, **x = [x •, x ,, …, x** is a ™ vector of decision variables, and the symbol ***° 0** denotes that the resulting matrix is positive semidefinite. The linearity lies in the fact that the matrix depends linearly on the variables, while the inequality is defined in the cone of positive semidefinite matrices.

LMIs generalize many familiar scalar inequalities. For instance, the scalar condition **ax + b "c 0** can be represented as a 1×1 LMI. In higher dimensions, LMIs capture relationships between multiple variables and matrices, making them especially powerful for system design problems where matrix-valued constraints naturally arise.

Historical Context and Evolution

Linear Matrix Inequalities first emerged in the 1980s as part of the development of robust control theory. Pioneering works by researchers such as **Roberts, Glover, and Doyle** introduced LMIs to express H" control conditions and stability margins in a convex form. Over the past three decades, LMIs have permeated fields beyond control, including signal processing, power systems, finance, and machine learning.

The rise of interior-point methods and efficient semidefinite programming solvers—like SeDuMi, SDPT3, and MOSEK—transformed LMIs from theoretical curiosities into practical tools. Today, engineers routinely formulate design problems as LMIs and solve them in seconds on a laptop.

Key Advantages of LMIs

- Convexity: LMIs define convex feasibility regions, guaranteeing that any local optimum is a global optimum.
- Scalability: Modern solvers handle matrices with thousands of variables efficiently.
- Flexibility: A wide range of constraints—stability, performance, actuator limits—can be expressed as LMIs.
- Software Ecosystem: Toolboxes like MATLAB's Robust Control Toolbox, CVX, and YALMIP provide high-level interfaces.
- Modularity: LMIs can be combined, decomposed, and embedded in larger optimization frameworks.

Typical LMI Applications

1. Stability Analysis

For a continuous-time linear system **‹ = Ax**, the Lyapunov stability condition can be expressed as **P "{ 0** and **A @P + PA "z 0**, an LMI in the variable **P**. Solving this LMI certifies asymptotic stability.

2. H" Control Design

Designing a controller that achieves a specified H" norm bound reduces to solving a set of coupled LMIs involving the plant matrices and controller gains.

3. Robust Control with Parameter Uncertainty

Uncertain systems can be modeled using polytopic or ellipsoidal uncertainty sets. LMIs provide a systematic way to guarantee performance for all admissible uncertainties.

4. Model Predictive Control (MPC)

Quadratic cost functions and linear constraints in MPC naturally lead to LMIs when formulating the dual problem or ensuring recursive feasibility.

5. Power System Optimization

Optimal power flow problems with voltage magnitude limits, generator constraints, and network flow equations can be relaxed into LMIs.

Mathematical Foundations

Positive Semidefinite Matrices

A symmetric matrix **M** is positive semidefinite if **z @Mz "e 0** for all vectors **z**. Equivalently, all eigenvalues of **M** are nonnegative. The notation **M *° 0** denotes this property.

Schur Complement

The Schur complement provides a powerful tool to transform block LMIs into simpler forms. For a symmetric block matrix

M = [[A, B]; [B @, C]]

with **A "{ 0**, the condition **M *° 0** is equivalent to **C - B @A {¹B *° 0**. This equivalence is frequently used to eliminate variables and reduce problem size.

Duality in Semidefinite Programming

Every LMI feasibility problem has a dual that provides bounds on the optimal value. Understanding the dual is essential for sensitivity analysis and for proving optimality.

Formulating an LMI Problem

1. Define Decision Variables: Decide which matrices or scalars will be optimized. Common choices include Lyapunov matrices, controller gains, or weighting matrices.
2. Express Constraints: Translate system requirements into matrix inequalities. Use linear combinations of the decision variables.
3. Set Objective (Optional): If you seek an optimum (e.g., minimize control effort), formulate a linear or quadratic objective. For feasibility problems, set a dummy objective.

4. Implement in a Solver: Use a modeling language (CVX, YALMIP) or a dedicated toolbox to encode the LMIs and call a semidefinite solver.
5. Interpret Results: Verify that the solution satisfies all constraints and extract meaningful parameters.

Popular Software Tools

- MATLAB + Robust Control Toolbox: Provides lyap, hinfsyn, and lmi functions for LMI manipulation.
- CVX: A convex optimization modeling system that accepts LMI constraints in a natural syntax.
- YALMIP: A MATLAB-based modeling language with extensive support for LMIs.
- Python + CVXPY: Offers a Pythonic interface to define and solve LMIs using solvers like SCS or MOSEK.
- SeDuMi, SDPT3, MOSEK: High-performance semidefinite programming solvers.

Why Use GitHub Pages for LMI Resources?

GitHub Pages provides a free, static website hosting service that is perfect for sharing tutorials, example code, and documentation. When combined with Jekyll or static site generators, you can create interactive, responsive pages that showcase LMI concepts to a global audience.

- Version Control: All code, data, and documentation are tracked in Git, enabling reproducible research.
- Collaboration: Contributors can fork, pull-request, and review changes, fostering community-driven learning.
- CI/CD Integration: Tools like GitHub Actions can automatically run tests or build documentation whenever the repository changes.
- SEO Friendly: Static pages load quickly and can be optimized for search engines.
- Open Access: Anyone can view or clone the repository, promoting transparency.

Building an LMI Tutorial Site on GitHub Pages

1. Create a Repository

Start by creating a new repository on GitHub. Name it something descriptive like **lmi-tutorials**. Enable the **GitHub Pages** feature in the repository settings, selecting the **main** branch and **/docs** folder as the source.

2. Structure Your Content

Organize the repository into logical sections:

- docs/ – Static HTML files.
- examples/ – MATLAB or Python scripts.
- data/ – Sample datasets.
- assets/ – Images, diagrams.
- README.md – Project overview.

3. Write Clean HTML Articles

Use semantic tags (**h2**, **h3**, **p**, **ul**, **ol**, **li**, **strong**) to structure each page. Keep the markup minimal to ensure fast loading times.

4. Embed Interactive Code

Include code snippets in **pre** blocks. For deeper interactivity, use **Python notebooks** converted to HTML via **nbconvert** or **JupyterLite** for browser-based execution.

5. Automate Documentation Build

Set up a GitHub Actions workflow that triggers on pushes to **main**. The workflow can run **mkdocs** or **Jekyll**to

rebuild the site, ensuring that the latest content is always live.

6. Optimize for SEO

Use descriptive titles, meta descriptions, and keyword-rich headings. The target keyword—“an introduction to linear matrix inequalities github pages”—should appear naturally in the title, first paragraph, and a few subheadings. Avoid overusing it.

7. Publish and Promote

Once the site is live, share the link on academic forums, social media, and mailing lists. Encourage feedback and contributions via pull requests.

Sample LMI Tutorial Outline

- Chapter 1: Basics of LMIs – Definitions, properties, and simple examples.
- Chapter 2: Stability via LMIs – Lyapunov theory and computational methods.
- Chapter 3: H" Control – Design procedure and case studies.
- Chapter 4: Robust Control – Modeling uncertainties and LMI formulations.
- Chapter 5: MPC with LMIs – Predictive control and recursive feasibility.
- Chapter 6: Advanced Topics – Parameter-dependent LMIs, distributed control, and networked systems.
- Appendix: MATLAB & Python Code – Full scripts for each example.

Case Study: A Simple Stabilization Problem

Consider the continuous-time system

‹ = Ax + Bu,

Baca Juga (Artikel Terkait)

- **advanced engineering electromagnetics solution manual edition hardcover by balanis constantine a published by wiley**
  URL: http://blueesystems.com/advanced-engineering-electromagnetics-solution-manual-edition-hardcover-by-balanis-constantine-a-published-by-wiley.pdf
- **advanced engineering electromagnetics solutions manual**
  URL: http://blueesystems.com/advanced-engineering-electromagnetics-solutions-manual.pdf
- **advanced engineering electromagnetics wiley 1989 grading**
  URL: http://blueesystems.com/advanced-engineering-electromagnetics-wiley-1989-grading.pdf
- **advanced engineering fluid mechanics by biswas**
  URL: http://blueesystems.com/advanced-engineering-fluid-mechanics-by-biswas.pdf

Tags: #introduction #linear #matrix