LMI Methods in Optimal and Robust Control

Matthew M. Peet
Arizona State University

Lecture 1: The Big Picture

Who Am I?

Website: http://control.asu.edu

Research Interests: Computation, Optimization and Control

Focus Areas:

- Control of Nuclear Fusion
- Immunology
- Thermostats, Renewable Energy, and Power Distribution

Expertise with LMI Methods:

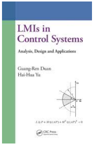
- Optimization of Polynomials
- Parallel Computing for Control
- Control of Delayed Systems
- Control of PDE Systems
- Control of Nonlinear Systems

My Background:

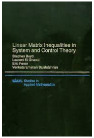
- B.Sc. University of Texas at Austin
- Ph.D. Stanford University
- Postdoc at INRIA Paris
- NSF CAREER Awardee

Office: ERC 253; Lab: GWC 531

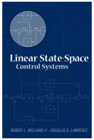
Required: LMIs in Control Systems
by Duan and Yu



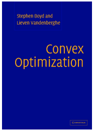
LMIs in Systems and Control Theory
by S. Boyd
Link: Available Online Here



Linear State-Space Control Systems
by Williams and Lawrence



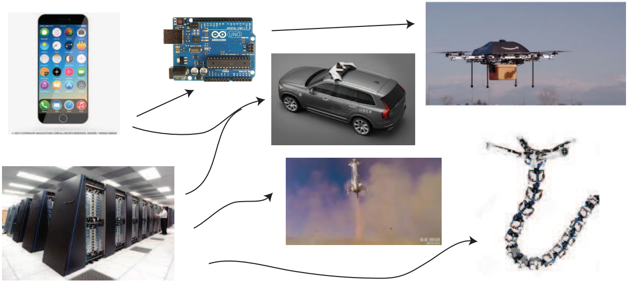
Convex Optimization
by S. Boyd
Link: Available Online Here



Link: Entire Course Online Here

MAE 598: LMI Methods in Optimal and Robust Control

What are the challenges?

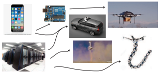


Megatrends:

- Increased Complexity (Embedded Computation and Control)
- Increased Connectivity (Internet of Things)
- Robots, Drones and Self-Driving Cars
- Increased Demands (Higher Standards)
- Mobile Computing (Mobile Apps)

MAE 598: LMI Methods in Optimal and Robust Control

What are the challenges?



Megatrends:

- Increased Complexity (Embedded Computation and Control)
- Increased Connectivity (Internet of Things)
- Robots, Drones and Self-Driving Cars
- Increased Demands (Higher Standards)
- Mobile Computing (Mobile Apps)

- **Sources of Complexity:** Smarter devices have more complicated action spaces; Ubiquitous computation; Cheap sensors and actuators;
- **Sources of Connectivity:** RFID, bluetooth, low-energy bluetooth, LAN, WiFi, WAN, 5G LTE, GPS, satellite broadband, TDRS, integrated circuits
  - **Problems:** delay, lost packets, noise, loss of signal, hacking
- **Sources of Demands:** Improved Efficiency; Expanded Functionality; User Friendliness; Reduced Tolerance for Failure.

Challenges for Control in the 21st century

Privatization of Space Travel

Challenges

- Safety
- Complexity
- Uncertainty

Links:
Blue Origin Successful Landing
Blue Origin Successful Landing: Flight 3
SpaceX Landing, Second Attempt
Proton M launch Failure (FCS was for wrong rocket)
Kepler Space Telescope

Challenges for Control in the 21st century

UAVs and Drones (Delay, Sampled-Data)

Safe Interaction with

- Crowded Airspace
- Real-Time Obstacle Avoidance

Precision Control with

- Delayed Feedback
  $$\dot{x}(t) = Ax(t) + Bu(t-\tau)$$
- Lossy Connections
  $$\dot{x}(t) = Ax(t) + Bu(t_k)$$

Links:
X47 Drone Carrier Landing
Raff's TED talk

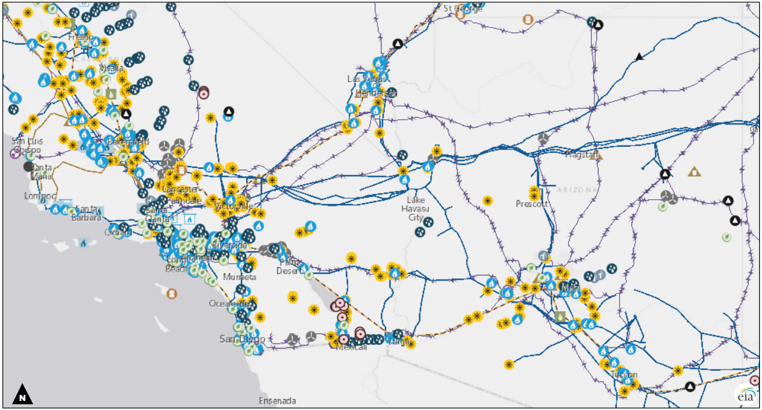
Challenges:

- Safety (Provable)
- Uncertainty (in model, environment)
- Other Drivers (Multi-Agent)
- Obstacles

Self-Driving Vehicles

- Google (Waymo)
- Über
- Tesla, Mobileye
- Toyota, Nutonomy

Links:
Toyota's Research Expansion in Automation
Uber's self-driving Taxis are in Pittsburg
Self-Driving Cars Flood into Arizona



0 15 30 60 Miles

Credits: layer1 : Esri, HERE, DeLorme, MapmyIndia, © OpenStreetMap contributors, and the GIS user community; State Layers : ; layer0 : Esri, HERE, DeLorme, MapmyIndia, © OpenStreetMap contributors, and the GIS user community

- Surface Coal Mine
- Underground Coal Mine
- Biomass Power Plant
- Coal Power Plant
- Geothermal Power Plant
- Hydroelectric Power Plant
- Natural Gas Power Plant
- Nuclear Power Plant
- Other Power Plant
- Petroleum Power Plant
- Pumped Storage Power Plant
- Solar Power Plant
- Wind Power Plant
- Petroleum Refinery
- Biodiesel Plant
- Ethanol Plant
- Natural Gas Processing Plant (z)
- Ethylene Cracker
- HGL Market Hub (z)
- Natural Gas Market Hub (z)
- Electricity Border Crossing
- Natural Gas Pipeline Border Crossing

HARD Robots

- Uncertain Terrain
- Interactions with the environment

`If x(t)>0:`

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

`If` x_1`(t)=0 AND` x_2`(t)<0: Set`

$$x_2(t) = -x_2(t)$$

Link:

Boston Dynamics, Atlas Mark 3

SOFT Robots

- Infinite Degrees of Freedom
- Material Dynamics

Link:

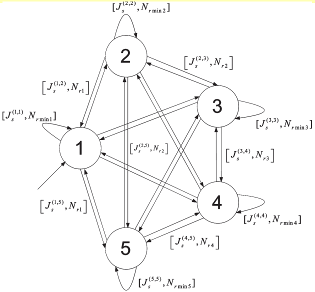
Robotic Worm

Challenges for Control in the 21st century

Arduino and Raspberry Pi

Trends:

- Rapid prototyping
- Internet of Things
- Control is Everywhere



Challenges

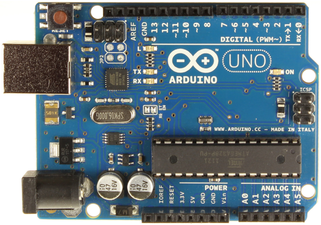
- Noisy Sensors
- Data-Driven Modeling
- Dynamics with logical switching

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

`If Occupied=True :`

$$u(t) = K_1 x(t)$$

`Else :` $u(t) = K_2 x(t)$



MAE 598: LMI Methods in Optimal and Robust Control

This course is on **RECENT** Developments in Control

- Techniques Developed in the Last 20 years
- Computational Methods
  - No Root Locus
  - No Bode Plots
  - No PID (Proportion-Integral-Differential)

We focus on State-Space Methods

- In the time-domain
- We use large state-space matrices

$$\frac{d}{dt}\begin{bmatrix}x_1(t)\\x_2(t)\\x_3(t)\\x_4(t)\end{bmatrix} = \begin{bmatrix}-1 & 1.2 & -1 & .8\\1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}x_1(t)\\x_2(t)\\x_3(t)\\x_4(t)\end{bmatrix} + \begin{bmatrix}1 & 0\\0 & 1\\0 & 0\\0 & 0\end{bmatrix}\begin{bmatrix}u_1(t)\\u_2(t)\end{bmatrix}$$

- We require Matlab
  - Need robust control toolbox.
  - Recommend using YALMIP.
    Link: Installs YALMIP and some other toolboxes

So What is an Automatic Control System???

Well... What is a System?

Definition 1.

A **System** is anything with Inputs and Outputs

There should ALWAYS be Inputs and Outputs!

- **If No Inputs**: You can't change anything.
- **IF No Outputs**: Then it doesn't matter anyway.

So What is an Automatic Control System???

In Controls, we separate internal signals from external signals.

Output Signals:

- **z:** Output to be controlled/minimized
- **y:** Output used by the controller

Input Signals:

- **w:** Disturbance, Tracking Signal, etc.
- **u:** Output from controller
  - Input to actuator

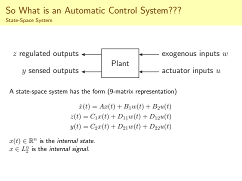
So What is an Automatic Control System???

State-Space System

A state-space system has the form (9-matrix representation)

$$\begin{aligned}\dot{x}(t) &= Ax(t) + B_1 w(t) + B_2 u(t)\\ z(t) &= C_1 x(t) + D_{11} w(t) + D_{12} u(t)\\ y(t) &= C_2 x(t) + D_{21} w(t) + D_{22} u(t)\end{aligned}$$

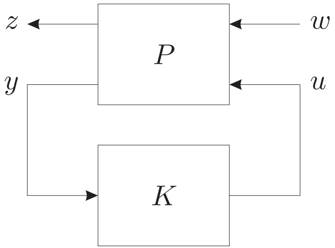
$x(t) \in \mathbb{R}^n$ is the *internal state*.
$x \in L_2^n$ is the *internal signal*.



Notation Matters

- $y \in L_2$ is a function
- $y(t) \in \mathbb{R}^m$ is a real number
- Systems (e.g. K) map signals to signals
  - We can say $y = Ku$
  - We can NOT say $y(t) = Ku(t)$

So What is an Automatic Control System???



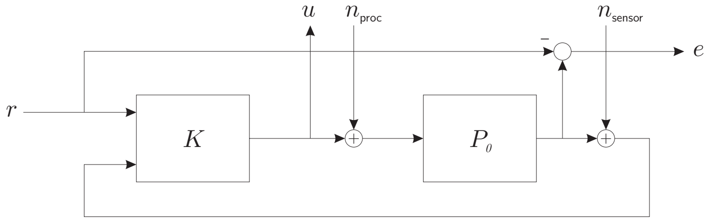
The controller, K, determines how to use the **signal** y to get the **signal** u.

- Can be *dynamic*: $u(t) = F\hat{x}(t)$, $\dot{\hat{x}}(t) = A\hat{x}(t) + L(y(t) - C\hat{x}(t))$
- Can be *static*: $u(t) = Fy(t)$.

Our job is to find the BEST K.

So What is an Automatic Control System???

Consider the Tracking Problem



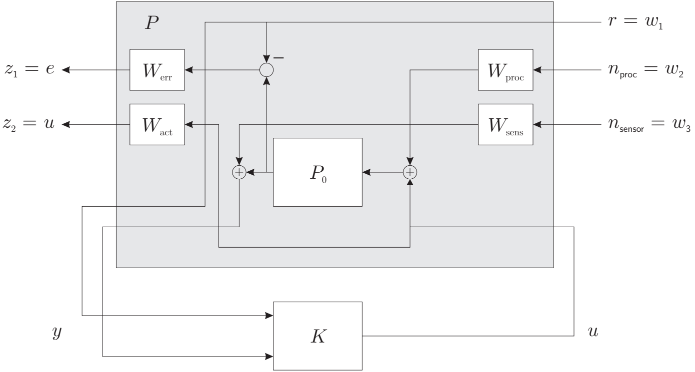
$$r = \text{reference input} \qquad w_2 = n_{proc} \qquad w_1 = r$$

$$e = \text{tracking error} \qquad w_3 = n_{sensor} \qquad u = u$$

$$n_{proc} = \text{process noise} \qquad z_1 = e \qquad y_1 = r$$

$$n_{sensor} = \text{sensor noise} \qquad z_2 = u \qquad y_2 = y_p$$

Tracking Control



$$P = \begin{bmatrix} I & -P_0 & 0 & -P_0 \\ 0 & 0 & 0 & I \\ I & 0 & 0 & 0 \\ 0 & P_0 & I & P_0 \end{bmatrix}$$

$$z_1 = r - P_0(n_{proc} + u)$$

$$z_2 = u$$

$$y_1 = r$$

$$y_2 = w_3 + P_0(n_{proc} + u)$$

What is Optimization?

An Optimization Problem has 3 parts.

$$\min_{x \in \mathbb{F}} \quad f(x): \qquad \text{subject to}$$
$$g_i(x) \geq 0 \qquad i = 1, \cdots K_1$$
$$h_i(x) = 0 \qquad i = 1, \cdots K_2$$

Variables: $x \in \mathbb{F}$

- The things you must choose.
- $\mathbb{F}$ represents the set of possible choices for the variables.
- Can be vectors, matrices, functions, systems, locations, colors...
  - However, computers prefer vectors or matrices.

Objective: $f(x)$

- A function which assigns a *scalar* value to any choice of variables.
  - e.g. $[x_1, x_2] \mapsto x_1 - x_2$; red $\mapsto 4$; et c.

Constraints: $g(x) \geq 0;\ h(x) = 0$

- Defines what is a minimally acceptable choice of variables.
- Equality forces two things to be the same
- Inequalities force one thing to be "better" than another
  - x is OK if $g(x) \geq 0$ and $h(x) = 0$.
- Constraints mean variables are not independent.

What is Optimization?

An Optimization Problem has 3 parts.

$$\min_{x \in \mathbb{F}} \quad f(x): \qquad \text{subject to}$$
$$g_i(x) \geq 0 \qquad i = 1, \cdots K_1$$
$$h_i(x) = 0 \qquad i = 1, \cdots K_2$$

Variables: $x \in \mathbb{F}$

- The things you must choose.
- $\mathbb{F}$ represents the set of possible choices for the variables.
- Can be vectors, matrices, functions, systems, locations, colors...
  - However, computers prefer vectors or matrices.

Objective: $f(x)$

- A function which assigns a *scalar* value to any choice of variables.
  - e.g. $[x_1, x_2] \mapsto x_1 - x_2$; red $\mapsto 4$; et c.

Constraints: $g(x) \geq 0$; $h(x) = 0$

- Defines what is a minimally acceptable choice of variables.
- Equality forces two things to be the same
- Inequalities force one thing to be "better" than another
  - x is OK if $g(x) \geq 0$ and $h(x) = 0$.
- Constraints mean variables are not independent.

The word "better" is defined using a notion of positivity (A Complete or Partial Ordering)

EVERYTHING is an Optimization Problem

- Teaching
- Studying
- Choosing a Class
- Getting Lunch
- Getting to Class
- Doing chores

The Trick is Modeling the Optimization Problem

How Hard is it to Solve Optimization Problems

For Humans:

- Almost always IMPOSSIBLE (or at least tedious)

For Computers:

- Easy if the Problem is CONVEX. (Polynomial Time)
- Otherwise IMPOSSIBLE. (NP-Hard)

We will talk about this a bit more later!

Now What is an LMI?

An LMI is a type of constraint

Definition 2.

A symmetric matrix ($P = P^T$) is **Positive Definite** (denoted $P > 0$) if all of its eigenvalues are positive.

A Linear Matrix Inequality (LMI) is a constraint that looks like

$$A_i P B_i + Q_i > 0$$

where P is the variable and A_i, B_i, Q_i are matrices.

Question: Why do we have a whole controls course devoted to LMIs?

- LMI constraints are convex (Computers can solve them)
- Positive matrices can be used to study systems.
  - This is because we are really optimizing Lyapunov functions.
  - $V(x) = x^T P x \geq 0$ if $P > 0$.

Almost **ALL** computational methods in Control are based on LMIs.

- Or at least be reformulated as an LMI.

Now What is an LMI?

An LMI is a type of constraint

Definition 2.
A symmetric matrix ($P = P^T$) is **Positive Definite** (denoted $P > 0$) if all of its eigenvalues are positive.

A Linear Matrix Inequality (LMI) is a constraint that looks like

$$A_i P B_i + Q_i > 0$$

where P is the variable and A_i, B_i, Q_i are matrices.

Question: Why do we have a whole controls course devoted to LMIs?

- LMI constraints are convex (Computers can solve them)
- Positive matrices can be used to study systems.
  - This is because we are really optimizing Lyapunov functions.
  - $V(x) = x^T P x \geq 0$ if $P > 0$.

Almost **ALL** computational methods in Control are based on LMIs.

- Or at least be reformulated as an LMI.

LMIs define a *Partial Ordering*

- One matrix may not be better or worse than another
- The LMI means the LHS must be better in EVERY way.

The system

$$\dot{x} = Ax$$

is stable (eigenvalues have negative real part) if and only if there exists a $P > 0$ such that

$$A^T P + PA < 0$$

YALMIP Code for Stability Analysis:

```
> A = [-1 2 0; -3 -4 1; 0 0 -2];
> P = sdpvar(3,3);
> F = [P >= eye(3)];
> F = [F, A'*P+P*A <= 0];
> optimize(F);
```

If Feasible, YALMIP Code to Retrieve the Solution:

```
> Pfeasible = value(P);
```

Class Project

In lieu of a final exam, we will have two class projects (Alone or in pairs).

1. **Write a Wikibook Chapter**
   - Include a minimum of 10 pages (20 for pairs)
2. **Do Research/Solve a Problem**
   - Can be based on existing research.

Some Project Ideas:

- Gain Scheduling for Missile Attitude Control (Switched Systems)
- Control of Robots over the internet (Sampled-Data Systems)
- Spacecraft Attitude Control with delayed communication (Delay Systems)
- Social Cognitive Therapy using Discrete Inputs (Mixed-Integer Control)
- Self-Driving Vehicles (Decentralized Control)
- Soft Robotics (Decentralized Control)
- Thermostat Programming (Dynamic Programming)
- Flow Control (PDEs)
- Controller/Estimator Design using Arduino and Simulink (Robust Control)
- System Identification using LMIs
- Mobile App for solving an optimization or control problem.

For those who dislike Projects, we can arrange to take a Final Exam instead.