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

Expertise with LMI Methods:

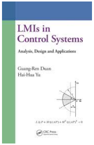
- Control of Delayed Systems
- Control of PDE Systems
- Control of Nonlinear Systems
- Control Software (SOSTOOLS/PIETOOLS)

My Background:

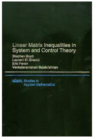
- B.Sc. University of Texas at Austin
- Ph.D. Stanford University
- Postdoc at INRIA Paris
- NSF CAREER Awardee

Office: ERC 253; Lab: GWC 531

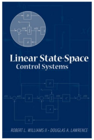
Required: LMIs in Control Systems
by Duan and Yu



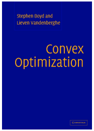
LMIs in Systems and Control Theory
by S. Boyd
Link: Available Online Here



Linear State-Space Control Systems
by Williams and Lawrence

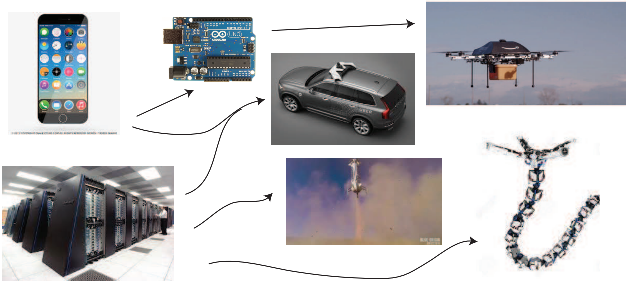


Convex Optimization
by S. Boyd
Link: Available Online Here



Link: Entire Course Online Here

What are the challenges?

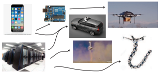


Megatrends:

- 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
- Delay

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

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:

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

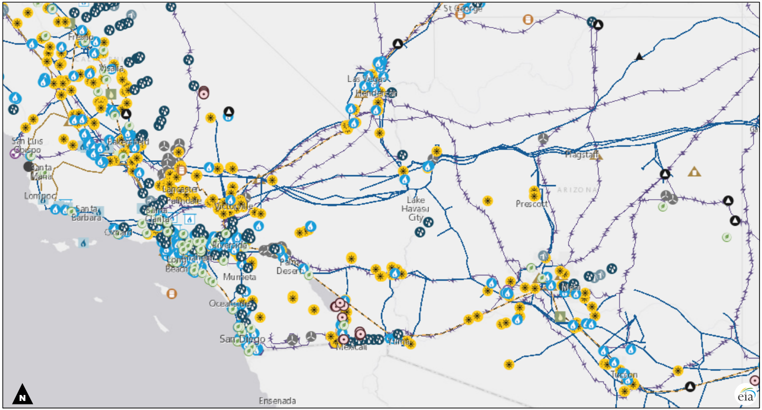
Self-Driving Vehicles

- 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

Challenges for Control in the 21st century

Interconnectivity (Decentralized Control)



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

Challenges for Control in the 21st century

Robotics (Hybrid and Nonlinear Dynamics, PDE systems)

HARD Robots (nonlinear, hybrid)

- 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 (PDEs)

- Infinite Degrees of Freedom
- Material Dynamics

Link:

Robotic Worm

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.
  - Need 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

State-Space Representation of Optimal Control

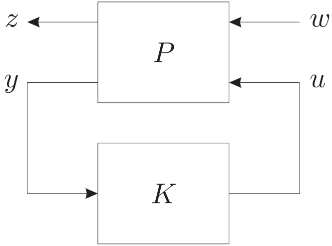
A state-space system has the form (9-matrix representation)

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

ANY optimal control problem can be formulated using 9 matrices.

- Computers love matrices

State-Space Representation of Optimal Control



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

- Can be *static*: $u(t) = Ky(t)$.
- Can be *dynamic*:

$$\begin{bmatrix} \dot{x}_K(t) \\ u(t) \end{bmatrix} = \underbrace{\begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix}}_{K} \begin{bmatrix} x_K(t) \\ y(t) \end{bmatrix}$$

Our job is to find the K which optimizes the closed-loop response $w \mapsto z$.

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: $x \in \mathbb{F}$

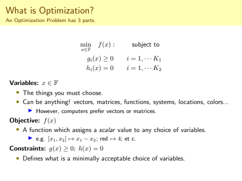
- The things you must choose.
- Can be anything! vectors, matrices, functions, systems, locations, colors...
  - However, computers prefer vectors or matrices.

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.



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 a Linear Matrix Inequality (LMI)?

LMIs are the largest class of optimization problems we can solve.

We have very few general-purpose tools in control

- LMIs, Ricatti Equations for State-space
- Root-locus, Bode plots, Nyquist for Transfer Functions

Definition 2.

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

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

$$APB + Q > 0$$

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?

- LMIs are convex optimization (Computers can solve them)
- LMIs are the most powerful general-purpose tool in modern control
- Makes it easy to teach a lot of material
  - Theory is not complicated (relatively)
  - Don't need a new method for every problem
  - Almost **ALL** control problems can be solved using LMIs.

Illustration of an LMI: The Lyapunov Inequality

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 3 pages (6 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.