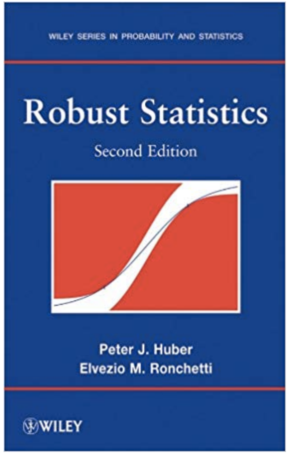
(ROBUST) CONTROL THEORY FUNDAMENTALS

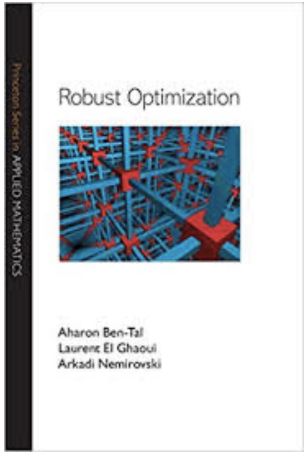
Pablo A. Parrilo - MIT EECS

6.S979 Topics in Deployable ML
Fall 2019

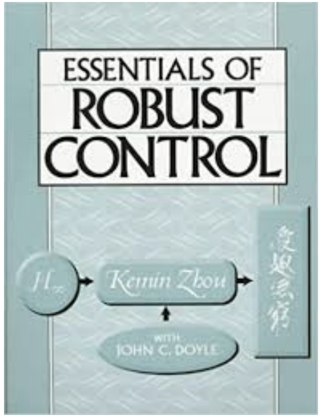
Around Robustness…

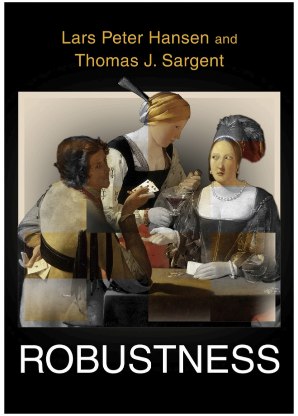
- Robust Statistics (1960s--)
- Robust Control (1980s--)
- Robust Optimization (1990s--)
- Robust Economics (2000s--)
- Robust Machine Learning (2015s--)





RobustML





History repeats itself, first as tragedy, second as farce.

Karl Marx

This lecture

- What is “Control Theory”?
- What is “Robust Control”?
- (Some) insights and results
- A few technical nuggets
- Models of uncertainty, and assessment of robustness
- Many questions that I don’t know the answer to

Dynamics and Control

Two key notions:

Dynamical systems:

- Systems that evolve over *"time"*

Control/Decisions:

- Values of some of the variables can *chosen* by us.

Many different flavors…

- Continuous/Discrete Time
- Differential/Difference Equations
- Differential/Algebraic Equations (DAEs)
- Stochastic Differential Equations (SDEs)
- Markov Decision Processes (MDPs)
- Discrete Event Systems (DES)
- (more…)

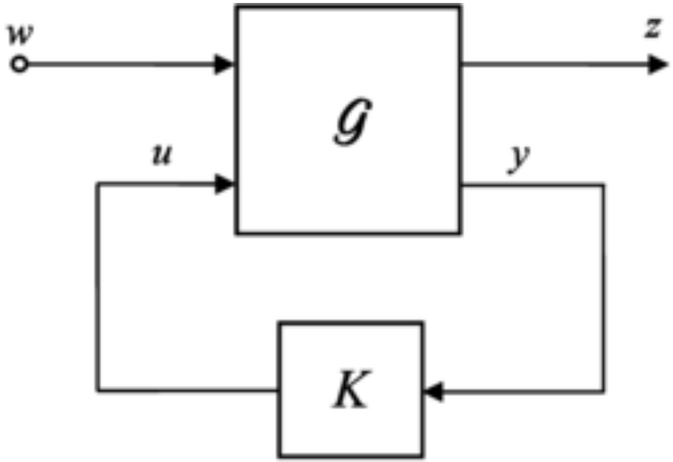
Also:

- Single/Multiple Decision Makers (e.g., games)
- “Classical” vs. “Adaptive” vs. “Robust”

A rough classification

	One Decision Maker	Several Decision Makers
Static (no time/sequential)	Optimization	Game theory Robust Optimization
Dynamic (sequential decisions)	Optimal Control	Dynamic Games Robust Control

Key notions



Analysis: Given a *closed-loop* system (G,K) and specifications, determine if the specifications are satisfied.

Synthesis: Given an *open-loop* system G and specifications, find a controller K such that the specifications are satisfied.

But…

Mathematical/Computational models (regardless of detail) are **always** merely approximations to the "real" phenomenon.

Robustness: Simultaneously model the system, ***and*** the uncertainty associated with it

E.g., unknown parameters, uncertain dynamics, environment…

Optimality does not suffice

J.C. Doyle (1978),
IEEE Transactions in Automatic Control

State-of-the-art design methods (optimal quadratic control, LQG)

Arbitrarily small perturbations can destabilize system!

Guaranteed Margins for LQG Regulators

JOHN C. DOYLE

Abstract—There are none.

Introduction

Considerable attention has been given lately to the issue of robustness of linear–quadratic (LQ) regulators. The recent work by Safonov and Athans [1] has extended to the multivariable case the now well-known guarantee of 60° phase and 6 dB gain margin for such controllers. However, for even the single-input, single-output case there has remained the question of whether there exist any guaranteed margins for the full LQG (Kalman filter in the loop) regulator. By counterexample, this note answers that question; there are none.

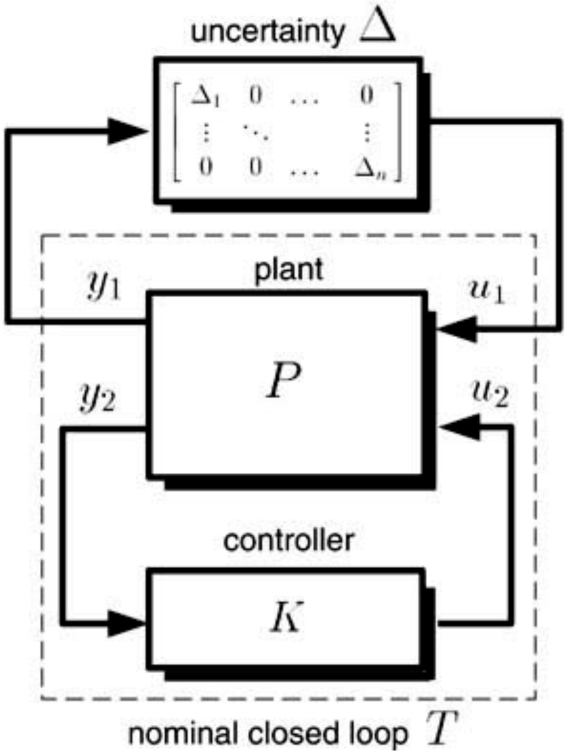
A standard two-state single-input single-output LQG control problem is posed for which the resulting closed-loop regulator has arbitrarily small gain margin.

Example

Model system *and* uncertainty

Many descriptions of uncertainty:

- **Set-based:** unknown elements in a given set (worst-case)
- **Probabilistic:** Set of allowable perturbations, plus a distribution



What can we efficiently do?

Models/theorems are great, but when the rubber meets the road, what really matters is what you can *compute effectively*
(and this may mean different things to different people)

Formulating vs. solving

Optimal Control: Hamilton-Jacobi-Bellman (HJB) equation

Nonlinear filtering: Kushner, Zakai equations

Dynamic Programming: “in principle” can solve any well-defined game, but..

(On the board…)

Linear systems – why do like them so much, and what we can do with them

Going nonlinear – analysis is great, synthesis not so much

Uncertainty – IQC models

Bottom line: in certain cases, reliable, algorithmic analysis/design methods

A Few Lessons

"Optimality" does not guarantee robustness – in fact, may hurt it

Valuable structural insights (e.g., controllability/observability)

Clever reformulations can help a lot, even if in the end we still resort to computational methods

Valuable toolkit to exploit structure

Certifying performance

- *Semidefinite relaxations* are central in today's robust control/ optimization
- Why? Because we don't know many ways of rigorously <u>certifying</u> properties of functions in many variables.
- Speculative: links between robustness and verifiability

Billion Dollar question

What will be the *epistemology* of deployable ML?

How will we convince ourselves that things “work”?

What will this mean? Theorems? Extensive simulations?

Distinction between cat/no-cat vs. safety-critical?
(e.g., 90000 daily flights in US. 99.9% success rate -> 90 daily plane crashes)

How to bridge across all these different cultures?