Escuela Superior de Ingenieros, Universidad de Sevilla

Randomized Algorithms for Analysis and Control of Uncertain Systems

Roberto Tempo
IEIIT-CNR, Politecnico di Torino
roberto.tempo@polito.it

Course Schedule

- Monday, March 14, 11:00 - 14:00
- Tuesday, March 15, 9:30 - 14:00 and 15:30 - 18:30
- Wednesday, March 16, 9:30 - 14:00 and 15:30 - 18:30
- Thursday, March 17, 9:30 - 14:00

- Additional documents, papers, etc, please consult
  http://staff.polito.it/tempo/
- Questions after the course can be sent to
  roberto.tempo@polito.it
- Some MATLAB™ codes are available at
  http://www.polito.it/~dabbene

Acknowledgments

- This research is supported by IEIIT-CNR of Italy
- Thanks to

Giuseppe Calafiore, Politecnico di Torino, Italy
Fabrizio Dabbene, IEIIT-CNR, Italy
Boris Polyak, Russian Academy of Sciences, Russia

Overview

1. Preliminaries
2. Robustness Analysis and Design
3. Randomized Algorithms
4. Random Vector Generation
5. Random Matrix Generation
6. Randomized Algorithms for Robust Design
7. Probabilistic LMIs

Overview

8. Probabilistic LPV Systems
9. Randomized Algorithms for Switched Systems
10. Miscellaneous Topics
11. Mixed Deterministic/Randomized Methods
12. Applications

Conclusions and Discussions
References

CHAPTER 1

Preliminaries

Keywords: *probability, robustness, uncertainty, good and bad sets*

Randomized Algorithms (RAs)

- Randomized algorithms are frequently used in many areas of engineering, computer science, physics, finance, optimization,…

Randomized Algorithms (RAs)

- Combinatorial optimization, computational geometry
- Examples: Data structuring, search trees, graph algorithms, sorting, …
- Motion and path planning problems
- Mathematics of finance: Computation of path integrals
- Bioinformatics (string matching problems)

RAs and Uncertain Systems

- Main point of the course: Rigorous study and development of RAs for uncertain systems and control, and related areas

Uncertainty

- Uncertainty has been always a critical issue in control theory and applications
- First methods to deal with uncertainty were based on a stochastic approach
- Optimal control: LQG and Kalman filter
- Since early 80's alternative deterministic approach (worst-case or robust) has been proposed

Robustness

- Major stepping stone in 1981: Formulation of the $\mathcal{H}_\infty$ problem by George Zames
- Various "robust" methods to handle uncertainty now exist: Structured singular values, Kharitonov, optimization-based (LMI), *l*-one optimal control, quantitative feedback theory (QFT)

Robustness

- Late 80's and early 90's: Robust control theory became a well-assessed area
- Successful industrial applications in aerospace, chemical, electrical, mechanical engineering, …
- However, …

Limitations of Robust Control

- Researchers realized some drawbacks of robust control
- Computational Complexity: Worst case robustness is often $\mathcal{NP}$-hard (not solvable in polynomial time unless $\mathcal{P} = \mathcal{NP}$)
- Various robustness problems are $\mathcal{NP}$-hard

Limitations of Robust Control

- Consider uncertainty Δ bounded in a set $\mathcal{B}$ of radius ρ. Largest value of ρ such that the systems is stable for all $\Delta \in \mathcal{B}$ is called (worst-case) robustness margin
- Conservatism: Worst case robustness margin may be small
- Discontinuity: Worst case robustness margin may be

  discontinuous wrt problem data

New Paradigm Proposed

- New paradigm proposed is based on uncertainty randomization and leads to randomized algorithms for analysis and synthesis
- Within this setting a different notion of problem tractability is needed

Brief History of RAs

- 1980: New notion of probability of instability by Stengel in the context of flight control[1]
- 1986: Stengel's book on stochastic optimal control[2]
- 1989: CDC paper[3] titled "Probabilistic Robust Controller Design"
- Basic idea: Use of Monte Carlo methods
- Major criticism: Absence of new mathematical tools

[1] R.F. Stengel (1980)

[2] R.F. Stengel (1986)

[3] P. Djavdan and H.J.A.F. Tulleken and M.H. Voetter and H.B. Verbruggen and G.J. Olsder (1989)

Brief History of RAs

- 1996: CDC (held in Kobe) two papers[1,2] on explicit bounds on the sample size
- 1998: Development of methods for "average" design based on statistical learning theory by Vidyasagar[3]

[1] P.P. Khargonekar and A. Tikku (1996)
[2] R. Tempo, E. W. Bai and F. Dabbene (1996)
[3] M. Vidyasagar (1998)

Brief History of RAs

- More recently, various research directions:
- Sample generation in various sets
- Bounds on the sample complexity
- Stochastic gradient algorithms for control design
- RAs for specific applications
- Combination of deterministic/randomized techniques
- RAs for nonconvex problems

New Paradigm Proposed

- New paradigm proposed is based on uncertainty randomization and leads to randomized algorithms for analysis and synthesis
- Main motivation: Limitations of robust control
- Computational complexity ($\mathcal{NP}$-hardness), conservatism and discontinuity of the robustness margin

Probability *and* Robustness

- The interplay of Probability and Robustness for control of uncertain systems
- Robustness: Deterministic uncertainty bounded
- Probability: Random uncertainty (pdf is known)
- Computation of the probability of performance
- Controller which stabilizes *most* uncertain systems

Key Features

- We obtain larger robustness margins at the expense of a small risk
- We study the probability degradation *beyond* the robustness margins
- Computational complexity is generally not an issue: Randomized algorithms are low complexity

Uncertain Systems

$M(s)$ System

Δ Uncertainty

- Δ belongs to a structured set $\mathcal{B}_D$
  - Parametric uncertainty q
  - Nonparametric uncertainty Δ_i
  - Mixed uncertainty

Worst Case Model

- Worst case model: Set membership uncertainty
- The uncertainty Δ is bounded in a set $\mathcal{B}_D$

$$\Delta \in \mathcal{B}_D$$

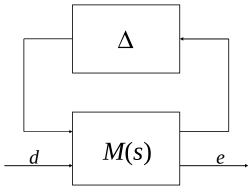
- Real parametric uncertainty $q=[q_1,\ldots,q_\ell] \in \mathbb{R}^\ell$

$$q_i \in [q_i^-, q_i^+]$$

- Nonparametric uncertainty

$$\Delta_i \in \{\Delta_i \in \mathbb{R}^{n,n} : \|\Delta_i\| \leq 1\}$$

Robustness



- Robustness is to guarantee stability and minimize performance for all Δ in $\mathcal{B}_D$

Objective of Robustness

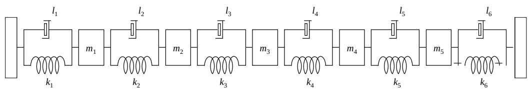
- Objective of robustness: To guarantee stability and performance for all

$$\Delta \in \mathcal{B}_D$$

- Different probabilistic paradigm based on uncertainty randomization of Δ within $\mathcal{B}_D$

Example: Flexible Structure

- Mass spring damper model
- Real parametric uncertainty affecting stiffness and damping
- Complex unmodeled dynamics (nonparametric)



Flexible Structure

- *M*-Δ configuration for controlled systems and study stability of

$$M(s)=C(sI-A)^{-1}B$$

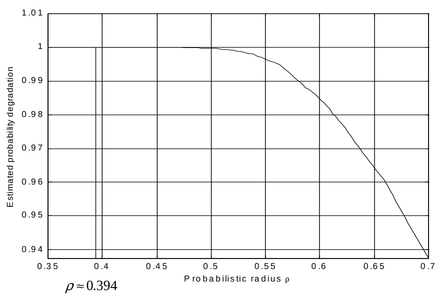
$$\Delta = \begin{bmatrix} q_1 I_5 & 0 & 0 \\ 0 & q_2 I_5 & 0 \\ 0 & 0 & \Delta_1 \end{bmatrix}$$

$q_1, q_2 \in \mathbb{R}$

$\Delta_1 \in \mathbb{C}^{4,4}$

$\Delta \in \mathcal{B}_D = \{\Delta \in \mathcal{D} : \bar{\sigma}(\Delta) < \rho\}$

Probability Degradation Function



Probabilistic Model

- Probability density function associated to $\mathcal{B}_D$
- We now assume that Δ is a random matrix with given density function $f_\Delta(\Delta)$ and support $\mathcal{B}_D$
- Example: Δ is uniform in $\mathcal{B}_D$

Uniform Density

- We take $f_\Delta(\Delta)=\mathcal{U}[\mathcal{B}_D]$ (uniform density within $\mathcal{B}_D$). That is

$$\mathcal{U}[\mathcal{B}_D] = \begin{cases} \dfrac{1}{vol(\mathcal{B}_D)} & if\ \Delta \in \mathcal{B}_D \\ 0 & otherwise \end{cases}$$

- In this case, for a subset $\mathcal{S} \subseteq \mathcal{B}_D$

$$\Pr\{\Delta \in \mathcal{S}\} = \frac{\int_{\mathcal{S}} d\Delta}{vol(\mathcal{B}_D)} = \frac{vol(\mathcal{S})}{vol(\mathcal{B}_D)}$$

Performance Function

- In classical robustness we guarantee that a certain performance requirement is attained for all $\Delta \in \mathcal{B}_D$
- This can be stated in terms of a performance function

$$J = J(\Delta)$$

Example 1: $\mathcal{H}_\infty$ Performance

- Compute the $\mathcal{H}_\infty$ norm of the transfer function $\mathcal{F}_u(M,\Delta)$ between the error e and the disturbance d

$$J(\Delta) = \| \mathcal{F}_u(M, \Delta)\|_\infty$$

$\mathcal{F}_u(M,\Delta)$ is called the upper LFT (Linear Fractional Transformation)

- For given $\gamma>0$, check if

$$J(\Delta) < \gamma$$

for all Δ in $\mathcal{B}_D$

Example 1: $\mathcal{H}_\infty$ Performance - 2

- Continuous time SISO systems with real parametric uncertainty q. Consider the transfer function $P(s,q)$ =

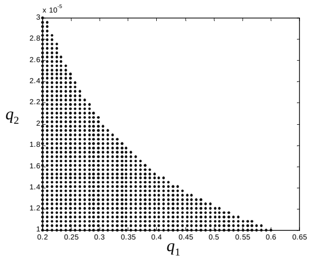
$$\frac{0.5 q_1 q_2 s + 10^{-5} q_1}{(10^{-5} + 0.05 q_2)s^2 + (0.00102 + 0.5 q_2)s + (2 \cdot 10^{-5} + 0.5 q_1^2)}$$

where $q_1 \in [0.2, 0.6]$ and $q_2 \in [10^{-5}, 3 \cdot 10^{-5}]$

- Letting $J(q) = \| P(s,q)\|_\infty$, we choose $\gamma=0.003$
- Check if $J(q)<\gamma$ for all q in these intervals

Example 1: $\mathcal{H}_\infty$ Performance - 3

- The set of q_1, q_2 for which $J(q)<\gamma$ is shown below



Example 2[1] : Robust Stability

- Robust stability of continuous time systems with real parametric uncertainty q. Consider the closed loop polynomial

$$p(s,q) = a_0(q) + a_1(q)s + \cdots + a_\ell(q)s^\ell$$

Then,

$$J(q) = \max\{\mathrm{Re}\ \lambda_1(q), \ldots, \mathrm{Re}\ \lambda_n(q)\}$$

where $\lambda_1(q),\ldots,\lambda_n(q)$ are the roots of $p(s,q)$

[1] G. Truxal (1961)

Example 2: Robust Stability - 2

- Consider the closed loop uncertain polynomial $p(s,q)$ =

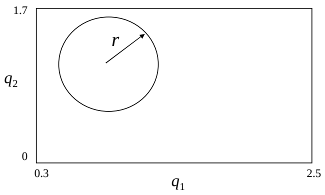
$$\left(1+r^2+6q_1+6q_2+2q_1q_2\right)+\left(q_1+q_2+3\right)s+\left(q_1+q_2+1\right)s^2+s^3$$

where $q_1 \in$ [0.3, 2.5], $q_2 \in$ [0,1.7] and r=0.5

- Check stability for all q in these intervals

Example 2: Robust Stability - 3

- Set of unstable polynomials



- Taking r=0 the unstable set reduces to a singleton

P1: Performance Verification

- For given performance level γ, check whether

$$J(\Delta) \leq \gamma$$

for all Δ in $\mathcal{B}_D$

P2: Worst-Case Performance

- Find $J_{\max}$ such that

$$J_{\max} = \max_{\Delta \in \mathcal{B}_D} J(\Delta)$$

Drawbacks: Complexity

- Complexity issue: Worst-case robustness is often $\mathcal{NP}$-hard (not solvable in polynomial time unless $\mathcal{P}=\mathcal{NP}$)[1]
- In classical robustness a number of problems are $\mathcal{NP}$-hard
  - stability of interval matrices
  - structured singular value
  - static output feedback
  - …

[1] V. Blondel and J.N. Tsitsiklis (2000)

Conservatism and Complexity: Trade-off

- Uncertain parameters often enter into the system in a nonlinear fashion
- To avoid complexity issues (or just to find a solution of the problem) overbounding techniques are used
- Worst-case robustness margins may be very conservative
- Another issue: Discontinuity of the robustness margin

Good and Bad Sets

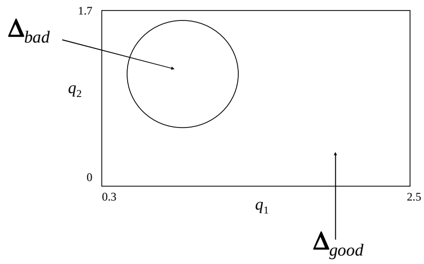
- We define two subsets of $\mathcal{B}_D$

$$\Delta_{good} = \{\Delta : J(\Delta) \le \gamma\} \subseteq \mathcal{B}_D$$
$$\Delta_{bad} = \{\Delta : J(\Delta) > \gamma\} \subseteq \mathcal{B}_D$$

- Δ_{good} is the set of Δ's satisfying performance
- Measure of robustness is

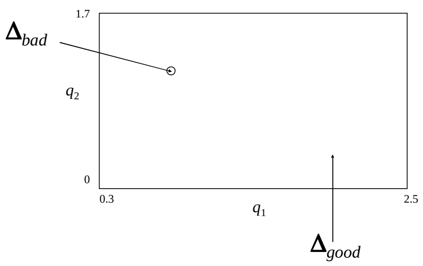
$$vol(\Delta_{good}) = \int_{\Delta_{good}} d\Delta$$

Example of Good and Bad Sets



Example of Good and Bad Sets - 2

Taking small r



Probabilistic Robustness Measure

- In worst-case analysis we compute γ such that all Δ satisfy performance. Equivalently, we evaluate γ such that

$$\Delta_{good} = \mathcal{B}_D$$

- In a probabilistic setting, we are satisfied if the ratio

$$\frac{vol(\Delta_{good})}{vol(\mathcal{B}_D)}$$

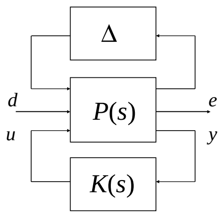
is close to one

CHAPTER 2

Nuts and Bolts of Robustness Analysis and Design

Keywords: *M-Δ configuration, structured singular value, stability radii, Kharitonov Theorem*

General Robust Control Framework



$$M(s) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} = \mathcal{F}_\ell(P,K)$$

$M(s)$ strictly proper

- Performance is measured by a characteristic of e.
- Closed loop: $e = \mathcal{F}_u(M,\Delta)d$

[1] K. Zhou and J.C. Doyle (1998)

Uncertainty

Δ	Uncertainty

- Parametric uncertainty q
- Nonparametric uncertainty Δ_i
- Mixed uncertainty

Structured Uncertainty

- Subspace defining uncertainty structure

$$\mathcal{D} = \{\Delta : \Delta = \text{blockdiag}[q_1 I_{r_1}, \cdots, q_\ell I_{r_\ell}, \Delta_1, \cdots, \Delta_b]\}$$

- Norm-bounded structured uncertainty

$$\mathcal{B}_D(\rho) = \{\Delta : \Delta \in \mathcal{D}, \|\Delta\| \le \rho\}$$

Robust Stability

- Let A,B,C be a realization of $M_{11}(s)$. Define the set

$$\mathcal{A}_\rho = \{A_\Delta : A_\Delta = A + B\Delta C, \Delta \in \mathcal{B}_D(\rho)\}$$

- The feedback connection is robustly stable if and only if every element in $\mathcal{A}_\rho$ is stable.
- The stability radius $\bar{\rho}$

$$\bar{\rho} = \sup\{\rho : \mathcal{A}_\rho \text{ is robustly stable}\}$$

Examples of Structures

- Full complex block: $\mathcal{D} = \mathbb{C}^{n,m}$

$$\bar{\rho} = \frac{1}{\sup_\omega \bar{\sigma}(M_{11}(j\omega))} = \frac{1}{\|M_{11}(s)\|_\infty}$$ Complex stability radius

- Full real block: $\mathcal{D} = \mathbb{R}^{n,m}$

$$\bar{\rho} = \left\{\sup_\omega \inf_{\gamma \in (0,1]} \sigma_2 \begin{bmatrix} \text{Re}(M_{11}(j\omega)) & -\gamma \text{Im}(M_{11}(j\omega)) \\ \gamma^{-1} \text{Im}(M_{11}(j\omega)) & \text{Re}(M_{11}(j\omega)) \end{bmatrix}\right\}^{-1}$$

Real stability radius

Mixed Structured Uncertainty

- Mixed uncertainty: $\mathcal{D} = \{\Delta : \Delta = \text{blockdiag}[q_1 I_{r_1}, \cdots, q_\ell I_{r_\ell}, \Delta_1, \cdots, \Delta_b]\}$

$$\bar{\rho} = \frac{1}{\sup_\omega \mu(M(j\omega))}$$ Structured stability radius

- μ is the structured singular value

$$\mu(M(j\omega)) = \frac{1}{\inf_{\Delta \in \mathcal{D}} \{\bar{\sigma}(\Delta) : \det(I - M_{11}(j\omega)\Delta) = 0\}}$$

Stability Radii

- We deal with stability radii in state space

$$\mathcal{A}_\rho = \{A_\Delta : A_\Delta = A + B\Delta C, \Delta \in \mathcal{B}_D(\rho)\}$$

$$\bar{\rho} = \sup\{\rho : \mathcal{A}_\rho \text{ is robustly stable}\}$$

Quadratic Stability of Interval Matrices

- Consider $A_\Delta = A + \Delta$, where $\|\Delta\|_\infty \leq \rho, A \in \mathbb{R}^{n,n}$
- A_Δ is quadratically stable if and only if $\exists\, P > 0$ such that

$$A_i^T P + P A_i < 0$$

  for all vertex matrices A_i

- Simultaneous solution of Lyapunov inequalities is a convex problem, but number of matrix inequalities grows as 2^{n^2}

Rank One μ Problem

- Consider stable t.f. $M_{11}(s)=u(s)v^T(s)$ where $u(s)$ and $v(s)$ are ℓ-dimensional vectors of rational functions
- Let $\Delta=\text{diag}(q_1,\ldots,q_\ell)$, then

$$\det(I - M_{11}(s)\Delta) = 1 + \sum_{i=1}^{\ell} q_i u_i(s) v_i(s)$$

- This leads to a *polytope of polynomials*
- Edge Theorem: The polytope is stable if and only if the one-dimensional edges are stable

Interval Polynomials

- An interval polynomial is of the form

$$p(s,q) = q_0 + q_1 s + q_2 s^2 + \cdots + s^\ell$$

  where $q_i \in [q_i^-, q_i^+]$

- Kharitonov Theorem: $p(s,q)$ is stable if and only if four particular vertex polynomials (Kharitonov polynomials) are stable

Convex Optimization in Control

- Many robust analysis and design problems may be formulated as convex optimization problems[1]
- A unifying framework is that based on Semidefinite Programming (SDP)

  H_2/H_∞ control

  Computable bounds for μ analysis/synthesis

  Multi-model robust design

[1] S. P. Boyd L. El Ghaoui, E. Feron and V. Balakrishnan (1994)

Robust Optimization

- Many control problems in the presence of uncertainty may be cast as *robust SDP*

$$\min c^T x, \text{ subject to}$$
$$F(x,\Delta) > 0$$
$$F(x,\Delta) = F_0(\Delta) + x_1 F_1(\Delta) + \cdots + x_m F_m(\Delta)$$
$$F_i = F_i^T, \Delta \in \mathcal{B}_D(\rho)$$

- For generic uncertainty structures, we can only compute relaxations of original robust SDP

CHAPTER 3

Randomized Algorithms

Keywords: *Monte Carlo methods, law of large numbers, Chernoff bound, worst-case bound*

Recall: Probabilistic Robustness Measure

- In worst-case analysis we compute γ such that all Δ satisfy performance. Equivalently, we evaluate γ such that

$$\Delta_{good} = \mathcal{B}_D$$

- In a probabilistic setting, we are satisfied if the ratio

$$\frac{vol(\Delta_{good})}{vol(\mathcal{B}_D)}$$

is close to one

Probability of Performance[1]

- We define the probability of performance as

$$p_\gamma = \Pr\{J(\Delta) \le \gamma\}$$

- Notice that, if $f_\Delta(\Delta)$ is uniform, then

$$p_\gamma = \frac{vol(\Delta_{good})}{vol(\mathcal{B}_D)}$$

[1] R.F. Stengel (1980)

Volume of Stable Polynomials Continuous Time[1]

- Let

$$p(s,a) = a_0 + a_1 s + \cdots + a_n s^n$$

- Define the set of stable polynomials

$$\mathcal{S}_n = \{a \in \mathbb{R}^{n+1} : 0 \le a_i \le 1, p(s,a) \ne 0 \forall s : \mathrm{Re}(s) \ge 0\}$$

- The volume of stable polynomials $V_n = vol(\mathcal{S}_n)$

$$V_n \le \frac{1}{n!}$$

$$V_n \approx e^{-n^2} \to 0 \text{ as } n \to \infty$$

[1] A.S. Nemirovskii and B.T. Polyak (1994)

Volume of Stable Polynomials Discrete Time[1]

- Let

$$p(z,a) = a_0 + a_1 z + \cdots + z^n$$

- Define the set of stable polynomials

$$\mathcal{S}_n = \{a \in \mathbb{R}^n : p(z,a) \ne 0 \forall |z| > 1\}$$

- The volume of stable polynomials $V_n = vol(\mathcal{S}_n)$

$$V_{n+1} = \frac{V_n^2}{V_{n-1}} \; n \text{ odd} \qquad V_{n+1} = \frac{nV_nV_{n-1}}{(n+1)V_{n-2}} \; n \text{ even}$$

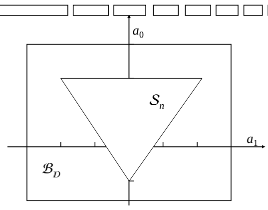
V_1=2, V_2=4, V_3=16/3

$$V_n \approx 2^{2n} n^{-\frac{n}{2}} \to 0 \text{ as } n \to \infty$$

[1] A.T. Fam (1989)

Volume of Stable Polynomials Discrete Time - 2

$$p(z,a) = a_0 + a_1 z + z^2$$



- If $\mathcal{S}_n \subseteq \mathcal{B}_D$ we can compute in closed form the probability of stability

$$p_\gamma = \frac{vol(\mathcal{S}_n \equiv \Delta_{good})}{vol(\mathcal{B}_D)}$$

Randomized Algorithms

- Volume estimate and numerical integration with Monte Carlo methods
- Monte Carlo and Quasi-Monte Carlo methods[1]
- Breaking the curse of dimensionality

[1] H. Niederreider (1992)

Ingredients for RAs

- Assume that Δ is random with pdf $f_\Delta(\Delta)$ with support $\mathcal{B}_D$
- Accuracy $\varepsilon \in (0,1)$ and confidence $\delta \in (0,1)$ be assigned
- Performance function for analysis and level

↓ ↓

$J = J(\Delta)$ γ

Randomized Algorithms for Analysis

- Two classes of randomized algorithms for probabilistic robust performance analysis
- Probabilistic performance verification (compute p_γ)
- Probabilistic worst case performance (compute worst-case performance)
- Both are based on Monte Carlo methods, i.e. on uncertainty randomization

Randomized Algorithms - 2

- We estimate p_γ by means of a randomized algorithm
- First, we generate N i.i.d. samples

$$\Delta^1, \Delta^2, \ldots, \Delta^N \in \mathcal{B}_D$$

according to the density f_Δ

- We evaluate $J(\Delta^1), J(\Delta^2), \ldots, J(\Delta^N)$

Empirical Probability

- Construct an indicator function

$$I(\Delta^i) = \begin{cases} 1 & if\ J(\Delta^i) \le \gamma \\ 0 & otherwise \end{cases}$$

- An estimate of p_γ is the empirical probability

$$\hat{p}_N = \frac{1}{N}\sum_{i=1}^{N} I(\Delta^i) = \frac{N_{good}}{N}$$

where N_{good} is the number of samples such that $J(\Delta^i) \le \gamma$

A Reliable Estimate

- The empirical probability is a reliable estimate if

$$|p_\gamma - \hat{p}_N| = |\Pr\{J(\Delta) \le \gamma\} - \hat{p}_N| \le \varepsilon$$

- Find the minimum N such that

$$\Pr\{|p_\gamma - \hat{p}_N| \le \varepsilon\} \ge 1 - \delta$$

where $\varepsilon \in (0,1)$ and $\delta \in (0,1)$

- ε and δ are called accuracy and confidence

Law of Large Numbers[1]

- For any $\varepsilon \in (0,1)$ and $\delta \in (0,1)$, if

$$N \ge \frac{1}{4\varepsilon^2\delta}$$

then

$$\Pr\{|p_\gamma - \hat{p}_N| \le \varepsilon\} \ge 1 - \delta$$

[1] J. Bernoulli (1713)

Remarks

- The number of samples computed with the Law of Large Numbers is independent of the number of blocks in Δ, the density function f_Δ and the size of $\mathcal{B}_D$
- The number of samples N is very large

ε	0.1%	0.1%	0.5%	0.5%
1-δ	99.9%	99.5%	99.9%	99.5%
N	$2.5\cdot10^{11}$	$5.0\cdot10^{10}$	$1.0\cdot10^{10}$	$2.0\cdot10^{9}$

Other Bounds

- The Bernoulli bound is based on the Chebyshev inequality
- Other bounds are also available, such as those based on the Bienaymé inequality
- A bound that largely improves the previous ones, for small values of δ and ε, is the Chernoff bound

Chernoff Bound[1]

- For any $\varepsilon \in (0,1)$ and $\delta \in (0,1)$, if

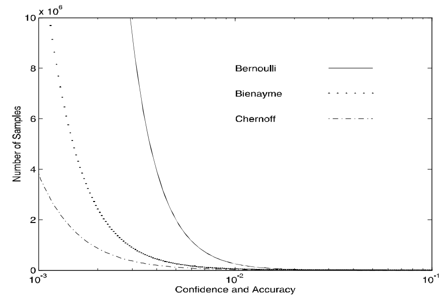
$$N \geq \frac{\log \frac{2}{\delta}}{2\varepsilon^2}$$

then

$$\Pr\left\{\left|p_\gamma - \hat{p}_N\right| \leq \varepsilon\right\} \geq 1-\delta$$

[1] H. Chernoff (1952)

Comparison Between Bounds



Chernoff Bound - 2

- We remark that the Chernoff bound greatly improves upon Bernoulli. The dependence on $1/\delta$ is logarithmic
- The dependence on $1/\varepsilon$ is still quadratic

ε	0.1%	0.1%	0.5%	0.5%
1-δ	99.9%	99.5%	99.9%	99.5%
N	$3.9\cdot10^{6}$	$3.0\cdot10^{6}$	$1.6\cdot10^{6}$	$1.2\cdot10^{5}$

Computational Complexity of RAs

- RAs are efficient (polynomial-time) because

1. Random sample generation of Δ^i can be performed in polynomial-time
2. Cost of checking stability for fixed Δ^i is polynomial-time
3. Sample size is polynomial in the problem size and probabilistic levels ε and δ

Polynomial-Time

- The cost associated with the evaluation of $J(\Delta^i)$ for fixed Δ^i is polynomial-time in many cases. For example, when checking stability or $\mathcal{H}_\infty$ performance

Cost of Checking Stability

- Consider a polynomial

$$p(s,a) = a_0 + a_1 s + \cdots + a_n s^n$$

- To check left half plane stability we can use the Routh test. The number of multiplications needed is

$$\frac{n^2}{4} \text{ for } n \text{ even} \qquad \frac{n^2-1}{4} \text{ for } n \text{ odd}$$

- The number of divisions and additions is equal to this number
- We conclude that checking stability is $O(n^2)$

Worst-Case Performance

- Recall that

$$J_{\max} = \max_{\Delta \in \mathcal{B}_D} J(\Delta)$$

- Generate N i.i.d. samples

$$\Delta^1, \Delta^2, \ldots, \Delta^N \in \mathcal{B}_D$$

according to the density f_Δ

- Compute

$$\hat{J}_N = \max_{i=1\ldots N} J(\Delta^i)$$

Worst-Case Bound[1,2]

- For any $\varepsilon \in (0,1)$ and $\delta \in (0,1)$, if

$$N \geq \frac{\log \frac{1}{\delta}}{\log \frac{1}{1-\varepsilon}}$$

then

$$\Pr\left\{\Pr\left\{J(\Delta) > \hat{J}_N\right\} \leq \varepsilon\right\} \geq 1-\delta$$

[1] P.P. Khargonekar and A. Tikku (1996)

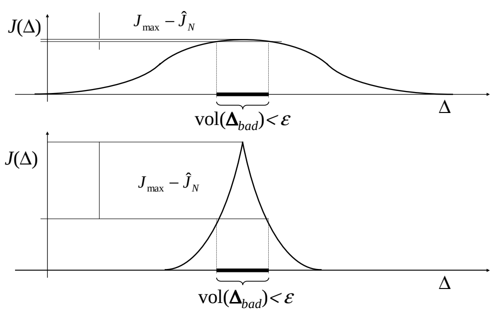
[2] R. Tempo, E. W. Bai and F. Dabbene (1996)

Comparison

- The number of samples required is much smaller than that needed with the Chernoff Bound.
- The dependence on $1/\varepsilon$ is basically linear $\left(\log \frac{1}{1-\varepsilon} \approx \varepsilon\right)$

ε	0.1%	0.1%	0.5%	0.5%	0.01%	0.001%
1-δ	99.9%	99.5%	99.9%	99.5%	99.99%	99.999%
N	$6.91 \cdot 10^3$	$5.30 \cdot 10^3$	$1.38 \cdot 10^3$	$1.06 \cdot 10^3$	$9.21 \cdot 10^4$	$1.16 \cdot 10^6$

Interpretation of Worst Case Bound



Volumetric Interpretation

- In the case of $f_\Delta(\Delta)$ uniform, we have

$$\Pr\{J(\Delta) > \hat{J}_N\} = \frac{vol(\Delta_{bad})}{vol(\mathcal{B}_D)}$$

- Therefore

$$\Pr\{\Pr\{J(\Delta) > \hat{J}_N\} \le \varepsilon\} \ge 1-\delta$$

is equivalent to

$$\Pr\{vol(\Delta_{bad}) \le \varepsilon\, vol(\mathcal{B}_D)\} \ge 1-\delta$$

Confidence Intervals

- The Chernoff and worst-case bounds can be computed *a-priori* and are explicit
- The sample size obtained with the confidence intervals is not explicit
- Given $\delta \in (0,1)$, upper and lower confidence intervals p_L and p_U are such that

$$\Pr\{p_L \le p_\gamma \le p_U\} = 1-\delta$$

Confidence Intervals - 2

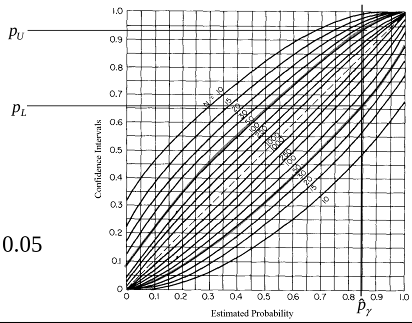
- The probabilities p_L and p_U can be computed *a posteriori* when the value of N_{good} is known, solving equations of the type

$$\sum_{k=N_{good}}^{N} \binom{N}{k} p_L^k (1-p_L)^{N-k} = \delta_L$$

$$\sum_{k=0}^{N_{good}} \binom{N}{k} p_U^k (1-p_U)^{N-k} = \delta_U$$

with $\delta_L + \delta_U = \delta$

Confidence Intervals - 3



$\delta = 0.05$

Example

- Let $J(q)$ be the maximum real part of the roots of the closed loop polynomial

$$p(s,q) = a_0(q) + a_1(q)s + \cdots + a_n(q)s^n$$

with q varying in the box $\mathcal{B}_q = \{q \in \mathbb{R}^n, \|q\|_\infty \le 1\}$

- Suppose that $p(s,0)$ is stable. Then, a box

$$\mathcal{B}_q(\rho) = \{q \in \mathbb{R}^n, \|q\|_\infty \le \rho\}$$

of radius $\rho > 0$ can be determined so that

$$J(q) < 0 \text{ for all } q \in \mathcal{B}_q(\rho)$$

Example - 2

- Can we estimate a box bigger than $\mathcal{B}_q(\rho)$ so that only a small number of plants in this box are unstable?
- We solve such a problem by applying the previous results

Example -3

- Consider the polynomial

$$p(s,q) = a_0(q) + a_1(q)s + a_3(q)s^3 + a_4(q)s^4$$

where

$$\begin{aligned} a_0(q) &= (100q_1 - 1)^2(50q_2 + 0.5)^2(q_3 + 1)(q_4 + 2) \\ a_1(q) &= \left[(100q_1 - 1)^2 + (50q_2 + 0.5)^2\right](q_3 + 1)(q_4 + 2) \\ &\quad + (100q_1 - 1)^2(50q_2 + 0.5)^2(q_3 + q_4 + 3) \\ a_2(q) &= (q_3 + 1)(q_4 + 2) + (100q_1 - 1)^2(50q_2 + 0.5)^2 \\ &\quad + \left[(100q_1 - 1)^2 + (50q_2 + 0.5)^2\right](q_3 + q_4 + 3) \\ a_3(q) &= (100q_1 - 1)^2(50q_2 + 0.5)^2 + (q_3 + q_4 + 3) \end{aligned}$$

Example - 4

- The parameters $q=[q_1,q_2,q_3,q_4]$ vary in the box $[-1/2,1/2]^4$
- Since $a_0(q)=0$ for $q_1=0.01$, the box containing only stable plant is surely smaller than $[-0.01,0.01]^4$
- Taking ε=0.001 and δ=0.01 we compute

$$N = 4603 > \frac{\log \frac{1}{\delta}}{\log \frac{1}{1-\varepsilon}}$$

- We generate $q^1,q^2,\ldots,q^{4603}$ i.i.d. samples using the uniform density function

Example - 5

- We define the performance function $J(q)$ as the maximum real part of the roots of $p(s,q)$ and compute

$$\hat{J}_N = \max_{i=1,\ldots,4603} J(\Delta^i) = -0.0000042395 < 0$$

- With probability at least 1-δ=0.99, the volume of the bad set Δ_{bad} is not greater than $\varepsilon\, vol(\mathcal{B}_D) = \varepsilon = 0.001$
- We obtain an increase in size which is at least

  6,250,000

Computational Learning Theory

- The Law of Large Numbers studies the problem

$$\Pr\left\{\left|p_\gamma - \hat{p}_N\right| \le \varepsilon\right\} \ge 1 - \delta$$

where $p_\gamma = \Pr\{J(\Delta) \le \gamma\}$
- The function $J(\Delta)$ is fixed
- Computational Learning Theory computes bounds on the sample size for the problem

$$\Pr\left\{\left|\Pr(J(\Delta) \le \gamma) - \hat{p}_N\right| \le \varepsilon, \forall J \in \mathcal{J}\right\} \ge 1 - \delta$$

where $\mathcal{J}$ is a class of functions

VC and P-dimension[1,2]

- The bounds obtained depend on quantities called VC-dimension (if J is a binary valued function), or P-dimension (if J is a continuous valued function)
- VC and P-dimension are measures of the problem complexity
- The bounds obtained are very conservative

[1] M. Vidyasagar (1997)
[2] E.D. Sontag (1998)

Systems and Control Application

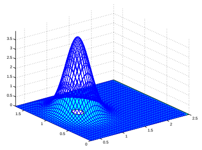
- The class of functions can take into account control parameters
- We replace $J(\Delta)$ with $J(\Delta,\kappa)$ where κ are the controller parameters
- With this approach, we generate random samples for both Δ and κ, i.e.

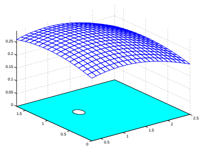
$$\Delta^1, \Delta^2, \ldots, \Delta^N \quad \text{and} \quad \kappa^1, \kappa^2, \ldots, \kappa^N$$

- We will discuss this approach in the broader contest of control system design with randomized algorithms

Choice of the Distribution

- The probability $\Pr\{\Delta \in \mathcal{S}\}$ depends on $f_\Delta(\Delta)$
- It may vary between 0 and 1 depending on the pdf $f_\Delta(\Delta)$





Choice of the Distribution

- The bounds discussed are independent on the choice of the distribution. To compute the estimates we need to know the distribution
- Some research has been done in order to find the worst-case distribution in a certain class[1]
- Uniform distribution is the worst-case if a certain target is convex and centrally symmetric

[1] B. R. Barmish and C. M. Lagoa (1997)

Choice of the Distribution

- Minimax properties of the uniform distribution have been shown in[1]

[1] E. W. Bai, R. Tempo and M. Fu (1998)

CHAPTER 4

Random Vector Generation

Keywords: *radial distributions, inversion method, generalized Gamma density, uniform distribution in norm balls*

RN and Univariate Generation

- Random number generation (RNG): Various methods available for uniform generation in the interval [0,1)
- Linear and nonlinear RNGs, Fibonacci, feedback shift register, BBS, MT, …
- Non-uniform univariate random variables: Suitable functional transformations (e.g., inversion method)

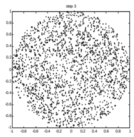
Multivariate Sample Generation

- Emphasis on finite sample size (Chernoff bound)
- Development of techniques not based on asymptotic methods such as Metropolis (random walk) or Hit-or-Miss
- Multivariate random variables: Rejection and conditional density methods
- Conditional density method requires computation of multiple integrals and can be used in some special cases

Uniform Sample Generation in ℓ_p Balls

- We study parametric uncertainty q in ℓ_p norm balls
- Objective: Uniform sample generation in the ball

$$\mathcal{B}_{\mathbb{R}} = \{ q \in \mathbb{R}^n : \|q\|_p \leq 1 \}$$



Rejection Method

- Rejection method: Curse of dimensionality
- Rejection rate for generation of uniform samples in the Euclidean sphere using an hypercube as bounding set

$$\eta(n) = (2/\sqrt{\pi})^n \, \Gamma(n/2+1)$$

n=1	n=2	n=3	n=4	n=10	n=20	n=30
1	1.2732	1.9099	3.2423	401.54	$4.\ 10^{7}$	$5.\ 10^{13}$

Gamma Density

- Need to introduce a technical tool
- A random variable x has (unilateral) Gamma density with parameters (a,b) if

$$f_x(x) = \frac{1}{\Gamma(a) b^a} x^{a-1} e^{-x/b}, x \geq 0$$

- We write $x \sim G(a,b)$
- There exist standard and efficient methods for random generation according to $G(a,b)$

Non-uniform Distributions: Inversion Method

- A standard tool for univariate random variable generation is the *inversion method*
- Let $w \in \mathbb{R}$ be a r.v. with uniform distribution in [0, 1]. Let F be a continuous distribution function on $\mathbb{R}$ with inverse

$$F^{-1}(y) = \inf \{ x : F(x) = y, 0 \leq y \leq 1 \}$$

- Then, the r.v. $z = F^{-1}(w)$ has distribution F
- The generation is obtained by passing a uniform r.v. through an appropriate transformation F^{-1}

Change of Variables

- Let x be a r.v. with pdf $f_x(x)$
- Let $x = g(y)$, g invertible
- The pdf of y is

$$f_y(y) = f_x(g(y)) \left| \frac{\partial g(y)}{\partial y} \right|$$

- This method also has multivariate extensions

Gamma Density

- A random variable $x \in \mathbb{R}$ has Gamma density with parameters (a,b) if

$$f_x(x) = \frac{1}{\Gamma(a) b^a} x^{a-1} e^{-x/b}, x \geq 0$$

- We denote $x \sim G(a,b)$
- There exist standard and efficient methods for random generation according to $G(a,b)$

Generalized Gamma Density

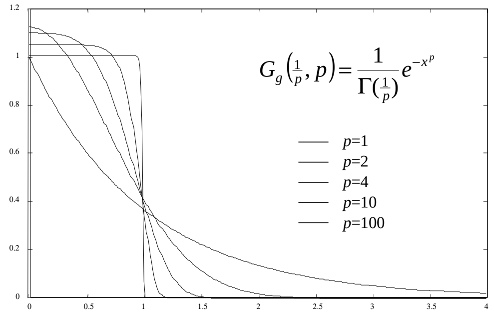
- A random variable $x \in \mathbb{R}$ has Generalized Gamma density with parameters (a,c) if

$$f_x(x) = \frac{c}{\Gamma(a)} x^{ca-1} e^{-x^c}, x \geq 0$$

- We denote $x \sim G_g(a,c)$

Course on Randomized Algorithms, Seville © IEIIT – CNR 2005 109

Generalized Gamma Density - 2



Course on Randomized Algorithms, Seville © IEIIT – CNR 2005 110

Comments

- If $z \sim G(a,1)$, taking $x = z^{1/c}$ we have $x \sim G_g(a,c)$
- Samples distributed according to a bilateral density $x \sim f_x(x)$ can be obtained from a unilateral density $z \sim f_z(z)$ taking

$$x = sz$$

where s is an independent random sign taking values $+1$ and -1 with equal probability

Course on Randomized Algorithms, Seville © IEIIT – CNR 2005 111

Joint Density

- Let $x=[x_1,\ldots,x_n]$ be a r.v. with components x_i independently distributed according to the (bilateral) Generalized Gamma density with parameters $1/p,p$
- The joint density of x is

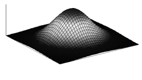
$$f_x(x) = \prod_{i=1}^{n} \frac{p}{2\Gamma(1/p)} e^{-|x_i|^p} = \frac{p^n}{2^n \Gamma^n(1/p)} e^{-\|x\|_p^p}$$

Course on Randomized Algorithms, Seville © IEIIT – CNR 2005 112

Examples

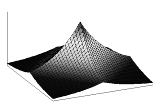
- Multivariate normal $N(0,I)$

$$f_x(x) = \frac{1}{\sqrt{(2\pi)^n}} e^{-\frac{1}{2}x^T x}$$



is $G_g(1/2,2)$
- Multivariate Laplace density

$$f_x(x) = \frac{1}{2^n} e^{-\sum_i |x_i|}$$



is $G_g(1,1)$

Course on Randomized Algorithms, Seville © IEIIT – CNR 2005 113

Radial Densities

- A random vector $x \in \mathbb{F}^n$ has ℓ_p-radial density if

$$f_x(x) = g(r), r = \|x\|_p$$

where $\|x\|_p = \left(\sum_i |x_i|^p\right)^{1/p}$

- $g(r)$ is called the *defining function* of x

Course on Randomized Algorithms, Seville © IEIIT – CNR 2005 114

Examples of Radial Densities

- The multivariate Normal $N(0,I)$

$$f_x(x) = \frac{1}{\sqrt{(2\pi)^n}} e^{-\frac{1}{2}x^T x}$$

is ℓ_2-radial

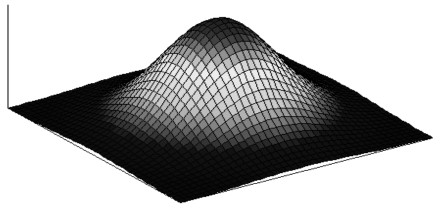
- The multivariate Laplace density

$$f_x(x) = \frac{1}{2^n} e^{-\sum_i |x_i|}$$

is ℓ_1-radial

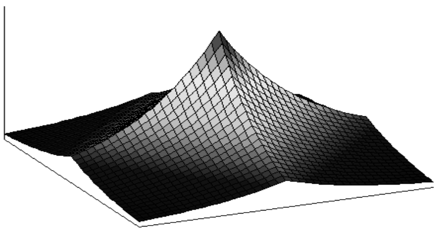
Radial Density, p=2

$$f_x(x) = \kappa e^{-\alpha \|x\|_2^2}$$



Radial Density, p=1

$$f_x(x) = \kappa e^{-\alpha \|x\|_1}$$



Radial Densities and Uniformity

- The key connection between ℓ_p-radial densities and uniform distributions in ℓ_p norm balls is given by the following fact
- Let $x \in \mathbb{R}^n$ be ℓ_p-radial, and let w be uniform in [0, 1], then the random vector

$$y = \frac{x}{\|x\|_p} w^{1/n}$$

has uniform distribution on the norm ball

$$\{x : \|x\|_p \leq 1\}$$

Real Random Vectors

- Theorem[1]: Let $\mathbb{F} \equiv \mathbb{R}$. If ξ_i are real r.v. distributed according to the Generalized Gamma density

$$\xi_i \sim G_g\left(\tfrac{1}{p}, p\right)$$

and $w \in [0,1]$ is a uniformly distributed r.v., then the vector

$$y = w^{\frac{1}{n}} \frac{x}{\|x\|_p}, \quad x = [\xi_1, \ldots, \xi_n]^T$$

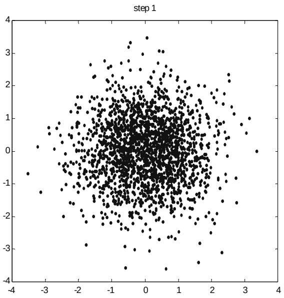
is uniformly distributed in $\mathcal{B}_{\mathbb{R}}$

[1] G. Calafiore, F. Dabbene and R. Tempo (1998)

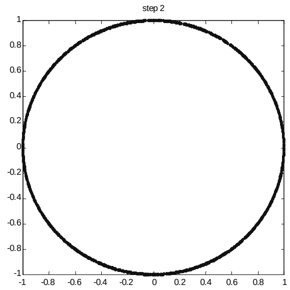
Algorithm for Real Uniform Generation

- Generate n independent real scalars $\xi_i \sim G_g(1/p, p)$
- Construct vector x of components $x_i = s_i\, \xi_i$, where s_i are random signs
- Generate $z = w^{1/n}$, where w is uniform in [0, 1]
- Return $y = \frac{x}{\|x\|_p} z$
- Similar algorithm exists for complex vectors

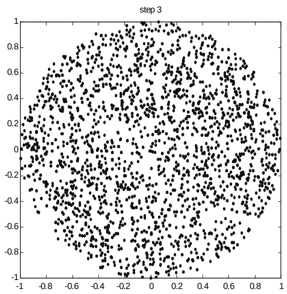
Real RandomVectors – *Step 1*



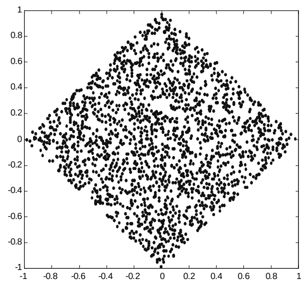
Real RandomVectors – *Step 2*



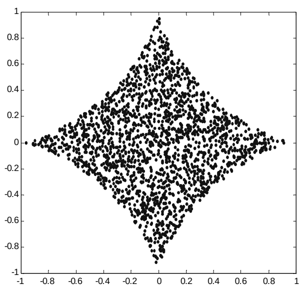
Real RandomVectors – *Step 3*



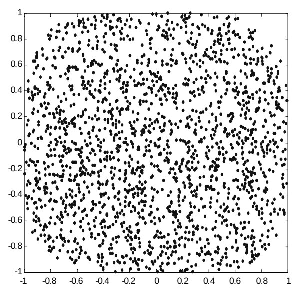
Real Random Vectors p=1



Real Random Vectors p=0.7



Real Random Vectors p=4



Complex Random Vectors

- Theorem: Let $\mathbb{F}\equiv\mathbb{C}$. If η_i are complex r.v. uniformly distributed on the unit circle, ξ_i are real r.v. distributed according to the Generalized Gamma density

$$\xi_i \sim G_g\left(\tfrac{1}{p}, p\right)$$

and $w\in[0,1]$ is a uniformly distributed r.v., then the vector

$$y = w^{\frac{1}{2n}} \frac{x}{\|x\|_p}, \quad x = [\xi_1\eta_1, \ldots, \xi_n\eta_n]^T$$

is uniformly distributed in $\mathcal{B}_{\mathbb{C}}$

Non-uniform Distributions

- ℓ_p-radial densities may be used to obtain generic ℓ_p-radial distributions (other than uniform) on the ℓ_p-norm ball
- Let $x\in\mathbb{R}^n$ be ℓ_p-radial, and let $z\sim f_z$ a scalar r.v., then the random vector $y = \frac{x}{\|x\|_p} z$ has density

$$f_y(y) = g(r) = \frac{1}{\nu n r^{n-1}} f_z(r), \; r = \|y\|_p$$

where ν is a normalization constant

Matrix Extensions

- Matrix Hilbert-Schmidt norms
- For 1-induced and ∞-induced matrix norms, the problem reduces to vector case
- Matrix spectral (max singular value) norm does not reduce to vector case

CHAPTER 5

Matrix Sample Generation

Keywords: *singular value decomposition, spectral norm, Haar density, conditional density method, Selberg integral, examples*

Matrix Sample Generation

- How to generate efficiently uniform matrix samples?
- Vector case is solved for the real and complex case, for any ℓ_p norm ball
- Matrix case is solved for the real and complex case, for any Hilbert-Schmidt p-norm (reduces to the vector case)
- For 1 and ∞ -induced matrix norms, the problem reduces to the vector case
- Hard problem for the spectral norm

A First Attempt: Rejection

- Methods based on rejection of samples generated from an outer-bounding set fail, due to dimensionality issues
- Let

$$\mathcal{B}^F(\sqrt{n}) = \left\{\Delta\in\mathbb{C}^{n,n} : \|\Delta\|_F \le \sqrt{n}\right\}$$
$$\mathcal{B}^\infty(1) = \left\{\Delta\in\mathbb{C}^{n,n} : \|\Delta\|_\infty \le 1\right\}$$
$$\mathcal{B}(1) = \left\{\Delta\in\mathbb{C}^{n,n} : \bar{\sigma}(\Delta) \le 1\right\}$$

then

$$\mathcal{B}(1)\subset\mathcal{B}^F(\sqrt{n});\ \mathcal{B}(1)\subset\mathcal{B}^\infty(1)$$

- Uniform generation in $\mathcal{B}^F$ and $\mathcal{B}^\infty$ is easy

Rejection Rates

- Let η be the average number of samples that one needs to generate in the outer set, to find one sample in the good set

$n =$	2	3	4	5	6	8	10
η_∞	12	8,640	8.7e8	2e16	2e26	5e54	1e95
η_F	8	468	1.8e5	4e8	6e12	2e23	1e37

Singular Value Decomposition

- Consider $\Delta \in \mathbb{F}^{n,m}$, $m \geq n$.
- Singular Value Decomposition

$$\Delta = U \Sigma V^*$$

where $U \in \mathbb{F}^{n,n}$ and $V \in \mathbb{F}^{m,n}$ have orthonormal columns, and

$$\Sigma = \mathrm{diag}\{\sigma_1, \sigma_2, \ldots, \sigma_n\}$$

where $\sigma_1 > \sigma_2 > \ldots > \sigma_n > 0$

A Class of Matrix pdfs

- Unitarily Invariant densities: Depend only on s.v. of Δ

$$\mathcal{F}_{\mathcal{I}} = \{ f_\Delta(\Delta) = f_\Delta(\Sigma) \}$$

- Radial Symmetric densities: Depend only on norm of Δ

$$\mathcal{F}_{\mathcal{R}} = \{ f_\Delta(\Delta) = f_\Delta(\sigma(\Delta)) \}$$

- Uniform distribution in $\mathcal{B}$

$$f_\Delta(\Delta) = \mathcal{U}[\mathcal{B}]$$

is a special case of radial density

The pdf of the Singular Values – Real Case

- Theorem[1]: Let $\Delta \in \mathbb{R}^{n,m}$. The following statements are equivalent
- The pdf $f_\Delta(\Delta)$ is unitarily invariant
- The joint pdf of U, Σ and V is $f_{U,\Sigma,V}(U,\Sigma,V) = f_U(U) f_\Sigma(\Sigma) f_V(V)$

$$f_U(U) = \mathcal{U}[\{U : UU^T = I\}]$$
$$f_V(V) = \mathcal{U}[\{V : V^T V = I, [V]_{1,i} > 0\}]$$

$$f_\Sigma(\Sigma) = \Upsilon_{\mathbb{R}} f_\Delta(\Sigma) \prod_{k=1}^{n} \sigma_k^{m-n} \prod_{1 \le i < k \le n} \left(\sigma_i^2 - \sigma_k^2\right)$$

[1] G. Calafiore, F. Dabbene and R. Tempo (2001)

Real Matrices

- $\Upsilon_{\mathbb{R}}$ is a normalization constant

$$\Upsilon_{\mathbb{R}} = \frac{(8\pi)^{n(m-1)/2}}{2^{n(m+1)/2}} \prod_{k=1}^{n} \frac{\Gamma((k-1)/2)}{\Gamma(k-1)} \prod_{k=m-n+1}^{m} \frac{\Gamma((k-1)/2)}{\Gamma(k-1)}$$

- Proof of previous theorem is based on the determination of the Jacobian of the mapping between Δ and its SVD factors U,Σ,V. Details are highly technical

Uniform Matrices – Real Case

- For particular case of uniform matrices

$$f_\Sigma(\Sigma) = K_{\mathbb{R}} \prod_{k=1}^{n} \sigma_k^{m-n} \prod_{1 \le i < k \le n} \left(\sigma_i^2 - \sigma_k^2\right)$$

with $\sigma_1 > \sigma_2 > \ldots > \sigma_n > 0$

- The value of $K_{\mathbb{R}}$ is given by the Selberg Integral

$$K_{\mathbb{R}} = n! \pi^{n/2} \prod_{i=0}^{n-1} \frac{\Gamma((m+i+1)/2)}{\Gamma(3/2 + i/2)\Gamma(i/2+1)\Gamma((i+m-n+1)/2)}$$

The pdf of the Singular Values – Complex Case

- Theorem[1]: Let $\Delta \in \mathbb{C}^{n,m}$. The following statements are equivalent
- The pdf $f_\Delta(\Delta)$ is unitarily invariant
- The joint pdf of U, Σ and V is $f_{U,\Sigma,V}(U,\Sigma,V) = f_U(U)\, f_\Sigma(\Sigma)\, f_V(V)$

$$f_U(U) = \mathcal{U}[\{U : UU^* = I\}]$$
$$f_V(V) = \mathcal{U}[\{V : V^*V = I, [V]_{1,i} > 0\}]$$
$$f_\Sigma(\Sigma) = \Upsilon_{\mathbb{C}}\, f_\Delta(\Sigma) \prod_{k=1}^{n} \sigma_k^{2(m-n)+1} \prod_{1 \le i < k \le n} (\sigma_i^2 - \sigma_k^2)^2$$

[1] G. Calafiore, F. Dabbene and R. Tempo (2001)

Complex Matrices

- $\Upsilon_{\mathbb{C}}$ is a normalization constant

$$\Upsilon_{\mathbb{C}} = \frac{2^n \pi^{mn}}{\prod_{k=1}^{n} (n-k)!(m-k)!}$$

- Proof of previous theorem is based on the determination of the Jacobian of the mapping between Δ and its s.v.d. factors U,Σ,V

Uniform Matrices – Complex Case

- Consider particular case of uniform matrices, and change of variables $x_i=\sigma_i^2$, with ordering condition removed

$$f_x(x) = K_x \prod_{i=1}^{n} x_i^{m-n} \prod_{1 \le i < k \le n} (x_i - x_k)^2$$

- The value of K_x is given by the Selberg Integral

$$K_x = \frac{1}{n!} \prod_{i=0}^{n-1} \frac{\Gamma(m+i+1)}{\Gamma^2(i+1)\Gamma(i+m-n+1)}$$

Outline of Sample Generation Method

- For uniform Δ, its SVD factors are independently distributed
  1. Generate the samples of U and V (easy problem)
  2. Generate the samples of Σ (hard problem)
  3. Build matrix sample $\Delta=U\Sigma V^T$

Generation of Haar Samples

- Uniform distribution over orthogonal (or unitary) group is known as the Haar invariant distribution
- Fundamental property: If U is Haar, then QU has same distribution as U, for any fixed orthogonal (unitary) matrix Q

Generation of Haar Samples

- Haar matrix $U \in \mathbb{R}^{n,n}$ may be generated by means of QR decomposition as follows

```
1. X=randn(n,n);
2. [Q,R]=QR(X);
3. U=Q;
```

- Complex case works similarly
- Rectangular Haar matrices work similarly

Generation of the Singular Values for Complex Matrices

Conditional Density Method

- Is a general method that reduces generation according to one n-dimensional distribution to n one-dimensional sample generation problems
- Drawback: Requires computation of marginal densities
- The previous is a very hard problem in general: Computing multiple integrals

Conditional Densities Method - 2

- Write $f_x(x_1,\ldots,x_n)$ as

$$f_x(x_1,\ldots,x_n) = f_1(x_1) f_2(x_2 \mid x_1) \cdots f_n(x_n \mid x_1 \cdots x_{n-1})$$

where

$$f_i(x_i \mid x_1 \cdots x_{i-1}) = \frac{f_i(x_1 \cdots x_i)}{f_{i-1}(x_1 \cdots x_{i-1})}$$

and

$$f_i(x_1 \cdots x_i) = \iint \ldots \int f_x(x_1,\ldots,x_n) dx_{i+1} \cdots dx_n$$

Conditional Density Method - 3

- A vector $x \in \mathbb{R}^n$ with density $f_x(x)$ can be obtained generating sequentially the x_i, $i=1,\ldots,n$
- Each x_i is generated independently according to the *univariate* distribution

$$f_i(x_i \mid x_1 \cdots x_{i-1})$$

Computing the Marginal Densities: Complex Matrices

$$f_i(x_1 \cdots x_i) = \iint \ldots \int f_x(x_1,\ldots,x_n) dx_{i+1} \cdots dx_n$$

Let

$$\mathcal{V}_i = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_i \\ x_1^2 & x_2^2 & \cdots & x_i^2 \\ \vdots & \vdots & \cdots & \vdots \\ x_1^{n-1} & x_2^{n-1} & \cdots & x_i^{n-1} \end{bmatrix} = [\mathcal{X}(x_1)\ \mathcal{X}(x_2) \cdots \mathcal{X}(x_i)]$$

be a partial Vandermonde matrix

Key Result

- The marginal density $f_x(x_1,\ldots,x_n)$ is equal to

$$f_x(x_1,\ldots,x_n) = K_x \frac{(n-i)!}{|M|} \left| \mathcal{V}_i^T M \mathcal{V}_i \right| \prod_{k=1}^{i} x_k^{m-n}$$

- Where $M=R^{-1}$,

$$[R]_{r,l} = \frac{1}{r+l+m-n-1}; r,l = 1,\ldots,n$$

- Proof of result based on Dyson-Mehta Theorem

Dyson-Mehta Theorem

- Let $Z_n \in \mathbb{R}^{n,n}$ be a symmetric matrix such that

1. $[Z_n]_{i,j} = \psi(x_i, x_j)$
2. $\int \psi(x,x)\, d\mu(x) = c$
3. $\int \psi(x,y)\psi(y,z)\, d\mu(y) = \psi(x,z)$

where $d\mu$ is a suitable measure, and c is a constant. Then

$$\int \det(Z_n)\, d\mu(x_n) = (c-n+1)\det(Z_{n-1})$$

where Z_{n-1} is the submatrix obtained from Z_n removing the row and column containing x_n

Key Result - 2

- Given $x_1, x_2, \ldots, x_{i-1}$, the marginal density is expressed as a polynomial in x_i

$$p_i(x_i) = C_{i-1} x_i^{m-n} \sum_{k=0}^{2(n-1)} b_k x_i^k$$

- The constants C_i and the coefficients b_i are computed by means of appropriate recursions
- We have efficient way to compute conditional densities

$$f_i(x_i \mid x_1, \ldots, x_{i-1}) = K_i x_i^{m-n} \sum_{k=0}^{2(n-1)} b_k x_i^k$$

Generation of the Singular Values for Real Matrices

Computing the Marginal Densities: Real Matrices

- Use again the conditional method
- Mathematical details are different from the complex case
- We again obtain marginal densities in "closed form"

Key Result

- The marginal density $f_i(\sigma_1, \ldots, \sigma_i)$ may be computed as

$$f_i(\sigma) = \frac{K_R}{2^{n-1}} \Psi(\sigma_1^2, \ldots, \sigma_i^2) \prod_{k=1}^{i} \sigma_k^{m-n}$$

where

$$\Psi_i(x_1, \ldots, x_i) \doteq \iint \ldots \int_{D_i} |\mathcal{V}_n(x)|\, d\mu(x_n) \cdots d\mu(x_{i+1})$$

being

$$D_i \doteq \{0 < x_n < \cdots < x_1 < 1\}$$

$$d\mu(x_k) = x_k^{\upsilon} dx_k$$

$$\upsilon = \tfrac{1}{2}(m-n-1)$$

Computation of ψ

- Theorem: $\Psi_i(x_1, \ldots, x_i)$ is equal to

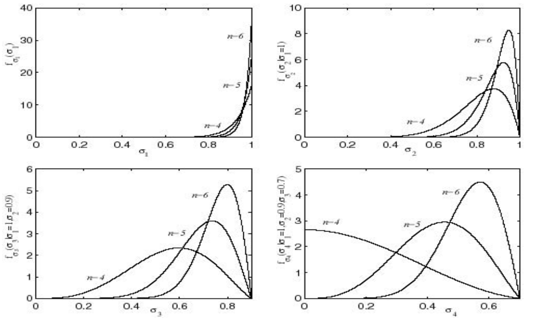
$$x_i^{\alpha_i} \frac{K_R}{2^n} \det{}^{1/2} \left[\begin{array}{cc|c} \multicolumn{2}{c|}{M(x_i)} & \begin{array}{c}\mathcal{V}_{i-1}(x_1,\ldots,x_{i-1}) \\ 0\end{array} \\ \hline -\mathcal{V}_i^T(x_1,\ldots,x_{i-1}) & 0 & 0 \end{array} \right]$$

where $\alpha_i = (\upsilon+1)(n-1)$

$$M(x_i) = \begin{cases} \begin{bmatrix} S(x_i) & \mathcal{X}(x_i) \\ -\mathcal{X}^T(x_i) & 0 \end{bmatrix} & \text{for } n-i \text{ even} \\ \begin{bmatrix} S(x_i) & \mathcal{X}(x_i) & \overline{F}(x_i) \\ -\mathcal{X}^T(x_i) & 0 & 0 \\ \overline{F}^T(x_i) & 0 & 0 \end{bmatrix} & \text{for } n-i \text{ odd,} \end{cases}$$

$$S_{jk}(x_i) = \frac{(j-k)x_i^{j+k-1}}{(j-\frac{1}{2})(k-\frac{1}{2})(j+k-1)}; \quad \overline{F}_j(x_i) = \frac{x_i^{j-\frac{1}{2}}}{j-\frac{1}{2}}$$

Marginal Densities



Sample Generation: Summary

- Details are highly technical
- Computation of the pdf of the singular values
- Computation of the pdf of *U*,*V* (Haar distribution)
- Conditional density method
- Closed-form solution of a multiple integral
- Dyson-Mehta Theorem

Polynomial-Time Algorithms

- Polynomial-time algorithms for the recursive generation of the singular values have been developed
- Algorithms require at each step only additions and multiplications of polynomial matrices
- Technical details and MATLAB™ software can be found at the URL http://www.polito.it/~dabbene
- Method becomes ill-conditioned for large n (n>20)
- Uniform matrices concentrate on the boundary of the norm-ball

Quasi-Monte Carlo Methods

Quasi-Monte Carlo Method[1]

- RAs presented are based on Monte Carlo (MC)
- Monte Carlo used for integration and optimization
- Quasi-Monte Carlo (QMC) is a deterministic version of MC

[1] H. Niederreiter (1992)

Quasi-Monte Carlo Method

- For integration, discrepancy is a measure of how the sample set is “evenly distributed” within the integration domain (unit cube)
- For optimization, the measure is the dispersion

Discrepancy

- Let $\underline{S}$ be a non-empty family of subsets of $[0,1]^n$ and let $\Delta^1,\ldots,\Delta^N \in [0,1]^n$ be a (deterministic) point set
- The discrepancy is defined as

$$D_N(\Delta^1,\ldots,\Delta^N) = \sup \left| \frac{\sum_i^N I(\Delta^i)}{N} - Vol(S) \right|$$

where supremum is taken wrt $S \in \underline{S}$, $Vol(S)$ is the volume of S and I is the indicator function of the set S

Integration Error

- Integration error is given by the Koksma-Hlawka inequality

$$\left| \int g(\Delta)\, d\Delta - (1/N)\sum_i^N g(\Delta^i) \right| \le \mathrm{V}(g)\; D_N(\Delta^1,\ldots,\Delta^N)$$

where $\mathrm{V}(g)$ is the variation of g

- To minimize integration error, we need to minimize discrepancy

Optimal Sequences

- Need to find low discrepancy (optimal) sequences which minimize discrepancy
- There are various low discrepancy sequences including van der Corput (for $n=1$), Halton, Sobol, Nieterreiter (for $n>1$)
- For Halton sequence we have

$$D_N(\Delta^1, \Delta^2, \ldots, \Delta^N) = \mathrm{O}(N^{-1} (\log N)^n)$$

MC and QMC

- The average absolute error of MC is $\mathrm{O}(N^{-1/2})$
- Main result of QMC: The (deterministic) error is

$$\mathrm{O}(N^{-1} (\log N)^n)$$

where n is the problem dimension

- This error is asymptotically smaller than $\mathrm{O}(N^{-1/2})$ but it depends on n
- Huge sample size is required before QMC is superior to MC

CHAPTER 6

Randomized Algorithms for Robust Design

Keywords: *LQ regulators, probabilistic quadratic stabilization, gradient based algorithms, probability-one controller*

Analysis vs Design with Uncertainty

- Starting point: worst-case analysis versus design
- Consider an interval family $p(s,q)$, $q \in \mathcal{B}_q = \{q \in \mathbb{R}^n, \|q\|_\infty \le 1\}$
- Analysis problem:
  - Check if $p(s,q)$ is stable for all $q \in \mathcal{B}_q$
    Answer: Kharitonov Theorem
- Design Problem:
  - Does there exist a $q \in \mathcal{B}_q$ such that $p(s,q)$ is stable?
    Answer: *Unknown* in general

Example

- Example:

$$p(s,q) = (s - q_0)(s - q_1)\cdots(s - q_{n-1})$$

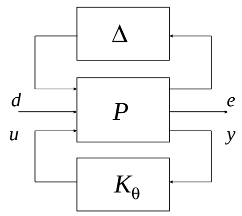
with $q_i \in [-0.5, 0.5]$

- Probability of stability is 0.5^n
- For n large the probability goes to zero
- Nevertheless, it is easy to design a stable polynomial in the family, e.g. $p(s,q^*)$ for

$$q_0^* = q_1^* = \cdots = q_{n-1}^* = -0.25$$

- Notice that this is not a fragile solution

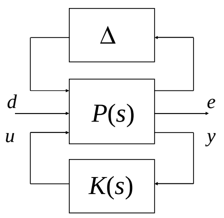
Synthesis Paradigm



- Design the parameterized controller K_θ to guarantee stability and performance

Probabilistic Synthesis Paradigm

- Uncertainty randomization of Δ in $\mathcal{B}_\mathcal{D}$
- Convex optimization to design the controller $K(s)$



Synthesis Performance Function

- Recall that the parameterized controller is K_θ
- We replace $J(\Delta)$ with a synthesis performance function

$$J = J(\Delta,\theta)$$

where $\theta \in \Theta$ represents the controller parameters to be determined and its bounding set

Randomized Algorithms for Synthesis

- Two classes of RAs for probabilistic synthesis
- Average performance synthesis[1]
- Robust performance synthesis[2]

[1] M. Vidyasagar (1998)

[2] B. Polyak and R. Tempo (2001)

Average Performance Synthesis

Statistical Learning Theory for Design

- Statistical learning theory for probabilistic design[1]
- Controller randomization and uncertainty randomization
- Bounds on the sample size: issue of conservatism[2]

[1] M. Vidyasagar (1997)
[2] V. Kolchinskii, C.T. Abdallah, M. Ariola, P. Dorato and D. Panchenko (2000)

Average Performance Synthesis

- Denote the average performance of the plant by

$$\varphi(\theta) = E_\Delta \{J(\Delta,\theta)\}$$

where E_Δ is expectation wrt Δ and optimal average performance is

$$\varphi^* = \inf_{\theta\in\Theta} \varphi(\theta)$$

- RA returns with probability $1-\delta$ a design vector $\boldsymbol{\theta}_N$ such that

$$\varphi(\boldsymbol{\theta}_N) - \varphi^* \leq \varepsilon$$

RA for Average Performance Synthesis

1. Determine finite sample size $N=N(\varepsilon,\delta)$
2. Draw N samples $\Delta^1,\ldots,\Delta^N$ according to the given pdf
3. Return the controller parameter

$$\boldsymbol{\theta}_N = \arg\inf_{\theta\in\Theta} N_{good}(\boldsymbol{\theta})/N$$

where N_{good} $(\boldsymbol{\theta})$ is the number of samples such that $J(\Delta^i,\boldsymbol{\theta})\leq\gamma$

Comments

- This RA is based upon Statistical Learning Theory[1]
- Sample complexity may be determined using the bound

$$N \geq (128/\varepsilon^2)(\log(8/\delta) + d\,(\log(32e/\varepsilon) + \log\log(32e/\varepsilon)))$$

for any $\varepsilon\in(0,1)$ and $\delta\in(0,1)$ and where d is an upper bound on the P-dimension of the function family

$$\mathcal{J} = \{J(\cdot,\boldsymbol{\theta}), \text{ for all } \boldsymbol{\theta}\}$$

[1] M. Vidyasagar (2002)

Comments

- An upper bound for d can be computed for several control problems. However, this introduces an extra level of conservatism.
- First issue: The sample complexity is huge
- Second issue: How to compute “inf”
- Randomization may be used also in this case, but this implies to generate controllers at random
- Advantage: Convexity of $J(\Delta, \boldsymbol{\theta})$ is not required

Robust Performance Synthesis

Robust Performance Synthesis

- RA returns with probability $1-\delta$ a design vector $\boldsymbol{\theta}_N$ such that

$$\Pr\{J(\Delta, \boldsymbol{\theta}_N) \leq \gamma\} \geq 1 - \varepsilon$$

- Controller parameter $\boldsymbol{\theta}_N$ is constructed using a finite number N of random samples of Δ

RA for Robust Performance Synthesis

1. Initialization
2. Determine sample size function $N(k)=N(\varepsilon,\delta,k)$
3. Set k=0, i=0 and choose $\boldsymbol{\theta}^0$
4. Feasibility loop
5. Set l=0 and feas=true
6. While l <N(k) and feas=true
7. Set i=i+1, l=l+1
8. Draw Δ^i according to f_Δ
9. If $J(\Delta^i, \boldsymbol{\theta}^k) > \gamma$ set feas = false
10. End While
11. Exit condition
12. If feas = true
13. Set N=I
14. Return $\boldsymbol{\theta}_N = \boldsymbol{\theta}^{k+1}$ and Exit
15. End If
16. Update
17. Update $\boldsymbol{\theta}^{k+1} = \psi_{upd}(\Delta^i, \boldsymbol{\theta}^k)$
18. Set k=k+1 and goto 4

Comments

- This RA is based on convexity of the performance function $J(\Delta,\theta)$ in θ
- Key point: Update rule $\psi_{upd}(\Delta^i, \boldsymbol{\theta}^k)$ in the update step
- Specific implementations of the update rule are a gradient step or a step of the ellipsoidal method
- For any ε, δ, number of steps $N(k)$ of the algorithm is given by[1]

$$N(k) \leq \frac{2\log \pi(k+1) - \log(6\delta)}{\log \frac{1}{1-\varepsilon}}$$

[1] Y. Oishi (2003)

RAs for Optimal Control (LQR)

Uncertain Systems in State Space

- We consider a state space description of the uncertain system

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

with $x(0)=x_0$; $x \in \mathbb{R}^n$; $u \in \mathbb{R}^m$, $\Delta \in \mathcal{B}_D$

- For example, $A(\Delta)$ is an interval matrix with bounded entries $a_{ij}^- \leq a_{ij} \leq a_{ij}^+$

LQ regulator

- The performance index is the quadratic cost function

$$J = \int_0^\infty \left(x^T S x + u^T R u\right) dt$$

where S >0 and R >0 are given weights

- The state feedback controller is

$$u = -R^{-1} B^T Q^{-1} x$$

where $Q=Q^T$ >0 is solution of a QMI

Quadratic Matrix Inequality

- Find $Q=Q^T>0$ such that, for given $0 \le \gamma \le 1$,

$$A(\Delta)Q + QA^T(\Delta) - 2BR^{-1}B^T + \gamma\left(QSQ + BR^{-1}B^T\right) \le 0$$

for all $\Delta \in \mathcal{B}_D$

- Then, the cost

$$J \le \frac{1}{\gamma} x_0^T Q^{-1} x_0$$

is guaranteed for all $\Delta \in \mathcal{B}_D$

Robust Quadratic Stabilization

- Quadratic stabilization and guaranteed cost can be reduced to checking a finite number of matrix inequalities
- Example: If $A(\Delta)$ is an interval matrix and $R=I$, for quadratic stabilization we take $\gamma = 0$ and we need to find a solution $Q=Q^T>0$ of

$$A^iQ + Q(A^i)^T - 2\,BB^T \le 0$$

for all vertex matrices A^i

Probabilistic Quadratic Stabilization

- Critical problem: The number of LMIs may be too large and not tractable with classical interior point methods
- Example: If $A(\Delta)$ is an interval matrix the number of LMIs is equal to the number of vertex matrices

$$N_V = 2^{n^2}$$

- Probabilistic version of quadratic stabilization and LQ regulator

Probabilistic Solution

- Randomly generate $\Delta^1,\ldots,\Delta^N \in \mathcal{B}_D$. Then, check if the LMI

$$A^iQ + Q(A^i)^T - 2\,BB^T \le 0$$

is feasible for $i=1,\ldots,N$ and find a common solution $Q=Q^T>0$

- Critical problem: Even if N is relatively small, this is a hard computational problem

Sequential Algorithm

- Key point: Sequential algorithm which deals with one constraint at each step
- At step k we have

  Phase 1: Uncertainty randomization of Δ

  Phase 2: Gradient algorithm and projection
- Final result: Find a solution $Q=Q^T>0$ with probability one in a finite number of steps

Definitions

- Let $\mathcal{E}_n$ be an Euclidean space

$$\mathcal{E}_n = \left\{ A = A^T \in \mathbb{R}^n, \|A\| = \sqrt{\sum_{i,k=1}^{n} a_{1k}^2} \right\}$$

and C be the cone of positive semi-definite matrices

$$C = \{A \in \mathcal{E}_n : A \ge 0\}$$

Projection on a Cone

- For any real symmetric matrix A we define the projection $[A]^{+}\in C$ as

$$[A]^{+} = \arg\min_{X\in C}\|A - X\|$$

- The projection can be computed through the eigenvalue decomposition $A=T\Lambda T^{T}$
- Then

$$[A]^{+} = T\Lambda^{+}T^{T}$$

where $\lambda_i^{+}=\max\{\lambda_i,0\}$

Phase 1: Uncertainty Randomization

- Uncertainty randomization: Generate $\Delta^{k}\in \mathcal{B}_D$
- Then, for guaranteed cost we obtain the QMI

$$A(\Delta^{k})Q + QA^{T}(\Delta^{k}) - 2BR^{-1}B^{T} + \gamma\left(QSQ + BR^{-1}B^{T}\right) \le 0$$

Matrix Valued Function

- Define a matrix valued function

$$V(Q,\Delta^{k}) = A(\Delta^{k})Q + QA^{T}(\Delta^{k}) - 2BR^{-1}B^{T} + \gamma\left(QSQ + BR^{-1}B^{T}\right)$$

and a scalar function

$$v(Q,\Delta^{k}) = \left\| \left[V(Q,\Delta^{k})\right]^{+} \right\|$$

where $\|\cdot\|$ is the Frobenius norm

- We can also take the maximum eigenvalue of $V(Q,\Delta^{k})$

Phase 2: Gradient Algorithm

- We write

$$Q^{k+1} = \begin{cases} \left[Q^{k} - \mu^{k}\partial_Q\left\{v\left(Q^{k},\Delta^{k}\right)\right\}\right]^{+} & \text{if } v\left(Q^{k},\Delta^{k}\right) > 0 \\ Q^{k} & \text{otherwise} \end{cases}$$

where ∂_Q is the subgradient and the stepsize μ^{k} is

$$\mu^{k} = \frac{v\left(Q^{k},\Delta^{k}\right) + r\left\|\partial_Q\left\{v\left(Q^{k},\Delta^{k}\right)\right\}\right\|}{\left\|\partial_Q\left\{v\left(Q^{k},\Delta^{k}\right)\right\}\right\|^{2}}$$

and r>0 is a parameter

Closed-form Gradient Computation

- The function $v(Q,\Delta^{k})$ is convex in Q and its subgradient is given by

$$\partial_Q\left\{v\left(Q,\Delta^{k}\right)\right\} = \frac{\left[V\left(Q,\Delta^{k}\right)\right]^{+}\left(A(\Delta^{k}) + \gamma QS\right) + \left(A(\Delta^{k}) + \gamma QS\right)^{T}\left[V\left(Q,\Delta^{k}\right)\right]^{+}}{v\left(Q,\Delta^{k}\right)}$$

if $v(Q,\Delta^{k})\neq 0$, and it is zero otherwise

Theorem[1]

- Assumption: Every open subset of $\mathcal{B}_D$ has positive measure
- Theorem: A solution Q, if it exists, is found in a finite number of steps with probability one
- Idea of proof: The distance of Q^{k} from the solution set decreases at each correction step

[1] B.T. Polyak and R. Tempo (2001)

Worst-Case Solution

- The sequential algorithm provides a candidate solution for the set of QMIs
- We can check if this candidate solution satisfies all QMIs and it is a worst-case solution, otherwise we run the algorithm again

Extensions

- Minimization of a measure of violation for problems that are not strictly feasible[1,2]
- Uncertainty in the control matrix, $B=B(\Delta)$, $\Delta\in \mathcal{B}_D$

  We take the feedback law

$$u = YQ^{-1}x$$

  where Y and $Q=Q^T>0$ are design variables

[1] B.R. Barmish and P. Shcherbakov (1999)

[2] G. Calafiore and B.T. Polyak (2001)

Further Extensions

- Nonlinear constrained systems
- Disturbance rejection and relations with game theory[1]
- Improvements from the numerical point of view

[1] T. Basar and P. Bernhard (1995)

Related Literature

- Related literature on optimization and adaptive control with linear constraints[1,2,3,4]
- Stochastic approximation algorithms have been widely studied in the stochastic control and optimization literature[6,7]

[1] S. Agmon (1954)
[2] T.S. Motzkin and I.J. Schoenberg (1954)
[3] B.T. Polyak (1964)
[4] V.A. Bondarko and V.A. Yakubovich (1992)

[6] H.J. Kushner and G.G. Yin (2003)
[7] J.C. Spall (2003)

Subsequent Research

- Design of common Lyapunov functions for switched system[1]
- From common to piecewise Lyapunov functions[2]
- Ellipsoidal algorithm instead of gradient algorithm[3]
- Stopping rule which provides the number of steps[4]
- Nonlinear constrained systems

[1] D. Liberzon and R. Tempo (2004)
[2] H. Ishii, T. Basar and R. Tempo (2004)
[3] S. Kanev, B. De Schutter and M. Verhaegen (2002)
[4] Y. Oishi and H. Kimura (2003)

Example[1]

- We study a multivariable example for the design of a controller for the lateral motion of an aircraft.
- The model consists of four states and two inputs

$$\dot{x}(t)=\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & L_p & L_\beta & L_r \\ g/V & 0 & Y_\beta & -1 \\ N_\beta(g/V) & N_p & N_\beta + N_\beta Y_\beta & N_r - N_\beta \end{bmatrix} x(t) + \begin{bmatrix} 0 & 0 \\ 0 & -3.91 \\ 0.035 & 0 \\ -2.53 & 0.31 \end{bmatrix} u(t)$$

[1] B.D.O. Anderson and J.B. Moore (1971)

Example - 2

- The state variables are
  - x_1 bank angle
  - x_2 derivative of bank angle
  - x_3 sideslip angle
  - x_4 jaw rate
- The control inputs are
  - u_1 rudder deflection
  - u_2 aileron deflection

Example - 3

- Nominal values: L_p=-2.93, L_β=-4.75, L_r=0.78, g/V=0.086, Y_β=-0.11, N_β=0.1, N_p=-0.042, N_β=2.601, N_r=-0.29
- Perturbed matrix $A(\Delta)$: each parameter can take values in a range of ±15% of the nominal value
- Quadratic stability (γ=0): take $R=I$ and $S=0.01I$
- Remark: $A(\Delta)$ is multiaffine in the uncertain parameters: quadratic stability can be ascertained solving simultaneously 2^9=512 LMIs

Example - 4

- Sequential algorithm:
  - Initial point Q_0 randomly selected
  - 800 random matrices Δ^k
  - The algorithm converged to

$$Q = \begin{bmatrix} 0.7560 & -0.0843 & 0.1645 & 0.7338 \\ -0.0843 & 1.0927 & 0.7020 & 0.4452 \\ 0.1645 & 0.7020 & 0.7798 & 0.7382 \\ 0.7338 & 0.4452 & 0.7382 & 1.2162 \end{bmatrix}$$

Example - 5

- The corresponding controller

$$K = B^T Q^{-1} = \begin{bmatrix} 38.6191 & -4.3731 & 43.1284 & -49.9587 \\ -2.8814 & -10.1758 & 10.2370 & -0.4954 \end{bmatrix}$$

  satisfies all the 512 vertex LMIs and therefore it is also a quadratic stabilizing controller in a deterministic sense
- The optimal LQ controller computed on the nominal plant satisfies only 240 vertex LMIs

CHAPTER 7

Probabilistic Algorithms for Robust LMIs

Keywords: *robust LMI, robust and approximate feasibility, stochastic gradient methods, quadratic stability*

Robust LMIs

- We discuss stochastic optimization methods for determining admissible solutions to robust Linear Matrix Inequalities (LMIs)
- Robust LMI

$$F(x,\Delta) < 0; \forall \Delta \in \mathcal{B}_D, x \in \mathbb{R}^m$$

where

$$F(x,\Delta) = F_0(\Delta) + \sum_{i=1}^{m} x_i F_i(\Delta);\ F_i = F_i^T \in \mathbb{R}^{n,n}$$

Robust LMIs - 2

- The set $\mathcal{B}_D$ indicates a structured norm bounded uncertainty set

$$\mathcal{D} = \{\Delta : \Delta = \text{blockdiag}[q_1 I_{r_1}, \cdots, q_\ell I_{r_\ell}, \Delta_1, \cdots, \Delta_b]\}$$

$$\mathcal{B}_D(\rho) = \{\Delta : \Delta \in \mathcal{D}, \|\Delta\| \le \rho\}$$

Robust Feasibility

- Let $\mathcal{X} \subseteq \mathbb{R}^m$ be a non-empty convex and closed set
- A point $\tilde{x}$ is a robustly feasible solution if and only if

$$\tilde{x} \in \mathcal{X}, \text{and } F(\tilde{x}, \Delta) < 0; \forall \Delta \in \mathcal{B}_D$$

- Computing robust solution to LMIs is in general numerically difficult
- Robust SDP techniques can handle relaxations of the problem

Feasibility Indicator Function

- Introduce a scalar function

$$\varphi(x,\Delta) = \|F^+(x,\Delta)\|$$

where F^+ indicates the projection of F onto the cone of positive semi-definite matrices

$$F^+ = \arg\min_{W \ge 0} \|W - F\|$$

- The function $\varphi(x,\Delta)$ is convex in x

Robust and Approximate Feasibility

- A point $\tilde{x}$ is a robustly feasible solution if and only if

$$\tilde{x} \in \mathcal{X}, \text{and } \varphi(\tilde{x}, \Delta) < 0; \forall \Delta \in \mathcal{B}_D$$

- In case no robustly feasible solution exists, we assume a probability density on Δ, and say that $\underline{x}$ is an approximately feasible solution if

$$\underline{x} = \arg\min_{x \in X} E_\Delta\{\varphi(x,\Delta)\}$$

Recall: Projections

- Let F^+ be the projection of a symmetric matrix F onto the positive semi-definite cone, then
- $F^+=RD^+R^T$, where $F=RDR^T$ is the eigen-decomposition of F, with $D=\text{diag}(d_1,d_2,\ldots,d_n)$, and $D^+= \text{diag}(d_1^+,d_2^+,\ldots,d_n^+)$, where $d_i^+=\max(d_i,0)$
- The (matrix) function $F^+(x)$ is convex in x
- The (scalar) function $\varphi(x,\Delta) = \|F^+(x,\Delta)\|$ is convex in x

Subgradients of φ

- A subgradient of $\varphi(x,\Delta)$ at x is computed as follows. Let

$$\nabla\varphi(x,\Delta) = \frac{1}{\varphi(x,\Delta)} \begin{bmatrix} \text{Tr}\, F_1 F^+(x,\Delta) \\ \vdots \\ \text{Tr}\, F_m F^+(x,\Delta) \end{bmatrix}$$

Then

$$\partial_x \varphi(x,\Delta) = \begin{cases} \nabla\varphi(x,\Delta) \text{ if } \varphi(x,\Delta) \ne 0 \\ 0 \text{ otherwise} \end{cases}$$

Algorithm for Approximate Feasibility

- Assume Δ is a random matrix with pdf f_Δ
- Let $g(x)=E\{\varphi(x,\Delta)\}$
- Let $\underline{x}$ be the minimum of $g(x)$ over $\mathcal{X}$
- Let $\|\partial_x\{\varphi(x,\Delta)\}\| \le \mu;\ \forall x\in \mathcal{X},\Delta$
- Given an initial vector x_0 consider the recursion

$$x_{k+1} = \lfloor x_k - \lambda_k \partial_x \{\varphi(x_k,\Delta^k)\}\rfloor_{\mathcal{X}}$$

where $[\cdot]_{\mathcal{X}}$ denotes the projection onto the set $\mathcal{X}$, and Δ^k are i.i.d. samples drawn from f_Δ

Approximate Feasibility - 2

- Let further

$$\underline{x}_k = \frac{m_{k-1}}{m_k}\underline{x}_{k-1} + \frac{\lambda_k}{m_k}x_k;\ \underline{x}_0 = m_0 = 0, m_k = m_{k-1}+\lambda_k$$

- Theorem[1]: $E\{g(\underline{x}_k)\} - g(\underline{x}) \le C(k)$

$$C(k) = \frac{\|x_0 - \underline{x}\|^2 + \mu^2\sum_{i=0}^{k-1}\lambda_i^2}{2\sum_{i=0}^{k-1}\lambda_i}$$

[1] G. Calafiore and B. Polyak (2001)

Approximate Feasibility - 3

- Moreover, if stepsizes are chosen so that

$$\{\lambda_i\} \to 0$$
$$\sum_{i=0}^{\infty}\lambda_i = \infty$$

Then

$$\lim_{k\to\infty} E\{g(\underline{x}_k)\} = g(\underline{x})$$

- The proof of this result is based on convexity (Jensen inequality) and properties of projections

Algorithm for Robust Feasibility

- Assume that a strong feasibility condition holds: There exist $x^*\in \mathcal{X}, \varepsilon > 0,$ such that

$$\varphi(x,\Delta)\le 0,\ \forall x\in \mathcal{X} : \|x - x^*\| \le \varepsilon,\ \forall \Delta \in \mathcal{B}_D$$

- Assume that, if x is not a robustly feasible solution, then it must hold that

$$\Pr\{F(x,\Delta) > 0\} > 0$$

Robust Feasibility - 2

- Consider the recursion

$$x_{k+1} = \lfloor x_k - \lambda_k \partial_x \{\varphi(x_k,\Delta^k)\}\rfloor_{\mathcal{X}}$$

where Δ^k are i.i.d. random samples drawn from f_Δ

- Define the stepsizes

$$\lambda_k = \begin{cases} \eta \dfrac{\varphi(x_k,\Delta_k) + \varepsilon\|\partial_x\{\varphi(x_k,\Delta_k)\}\|}{\|\partial_x\{\varphi(x_k,\Delta_k)\}\|^2}, & \text{if } \varphi(x_k,\Delta_k)\ne 0 \\ 0, & \text{otherwise} \end{cases}$$

- For any $x_0 \in \mathcal{X}$ the recursion finds a robustly feasible solution in a finite number of steps, w.p. 1

Example: Quadratic Stability[1]

- Consider a linear system with

$$A(\Delta) = A_0 + \Delta,\ |\Delta_{ij}| \le rS_{ij},\ r>0$$

$$A_0 = \begin{bmatrix} -2 & -2 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & -2 \end{bmatrix},\ S = \begin{bmatrix} .1651 & .9394 & .5691 \\ .2451 & .4727 & .1457 \\ .7004 & .4014 & .3141 \end{bmatrix}$$

- System is quadratically stable if and only if there exists P>0 that satisfies *simultaneously* 512 Lyapunov inequalities for the vertex matrices

[1] B. R. Barmish and P. S. Shcherbakov (2002)

Example - 2

- Let r=0.5. The stochastic algorithm (robust feasibility) converged after N=50 iterations to the solution

$$P = \begin{bmatrix} 1.2487 & 0.8155 & 0.3177 \\ 0.8155 & 2.0443 & 0.2425 \\ 0.3177 & 0.2425 & 0.5371 \end{bmatrix}$$

- This solution may be checked to actually satisfy all Lyapunov inequalities

Example - 3

- For r=1, we know that no actual robustly feasible solution exists
- After N=250 iterations of the approximate feasibility algorithm, we found the solution

$$P = \begin{bmatrix} 1.2042 & 0.9899 & -0.2649 \\ 0.9899 & 1.7455 & -0.0967 \\ -0.2649 & -0.0967 & 0.5577 \end{bmatrix}$$

- A posteriori probability of feasibility (computed with Monte Carlo using 100,000 samples) is p_{feas}=0.996

Example - 4

- We recall that to assess quadratic stability form a deterministic viewpoint one needs 2^{n^2} to solve simultaneous Lyapunov equations
- For n=4, 65,536 vertex matrices
- For n=10, 1,26 10^{30} vertex matrices
- n=4 is already beyond the capabilities of many current SDP solvers

Example - 5

- Consider an example with n=10

$$A_0 = \begin{bmatrix} -55 & -23 & 90 & 37 & -16 & -130 & 44 & -49 & 17 & 31 \\ -11 & 40 & -272 & -241 & 250 & 249 & -293 & 103 & -8 & 54 \\ 56 & 56 & -140 & -56 & 58 & 106 & -80 & 82 & -32 & -28 \\ 1 & -54 & 113 & -24 & -111 & -174 & 198 & -46 & -68 & -85 \\ 61 & 61 & -56 & -52 & -12 & 44 & -49 & 76 & -38 & -22 \\ 44 & 107 & -191 & -176 & 197 & 164 & -276 & 111 & -58 & 26 \\ -18 & 96 & -278 & -188 & 283 & 292 & -440 & 107 & 29 & 101 \\ -58 & -134 & 82 & 173 & -77 & -89 & 181 & -135 & 76 & -20 \\ 39 & 133 & -438 & -215 & 273 & 422 & -376 & 165 & -8 & 78 \\ -133 & -113 & 217 & 89 & -80 & -221 & 151 & -108 & 17 & -36 \end{bmatrix}$$

- Nominally stable

Example - 6

- Check quadratic stability for element-wise independent uncertainty with r =0.5
- N =2000 iterations yielded the solution

$$P_N = \begin{bmatrix} 0.0898 & 0.0133 & 0.0708 & 0.0288 & -0.0148 & -0.0745 & 0.0516 & 0.0050 & -0.0226 & -0.0481 \\ 0.0133 & 0.0843 & -0.0890 & -0.1109 & 0.1252 & 0.1242 & -0.1433 & 0.0603 & -0.0071 & 0.0336 \\ 0.0708 & -0.0890 & 0.4107 & 0.2060 & -0.3488 & -0.3592 & 0.3319 & -0.1474 & -0.1014 & -0.1467 \\ 0.0288 & -0.1109 & 0.2060 & 0.1998 & -0.2236 & -0.2538 & 0.2653 & -0.0988 & -0.0168 & -0.0908 \\ -0.0148 & 0.1252 & -0.3488 & -0.2236 & 0.4252 & 0.3490 & -0.3325 & 0.1988 & 0.0553 & 0.1282 \\ -0.0745 & 0.1242 & -0.3592 & -0.2538 & 0.3490 & 0.4237 & -0.3797 & 0.1688 & 0.0404 & 0.1337 \\ 0.0516 & -0.1433 & 0.3319 & 0.2653 & -0.3325 & -0.3797 & 0.3840 & -0.1493 & -0.0458 & -0.1370 \\ 0.0050 & 0.0603 & -0.1474 & -0.0988 & 0.1988 & 0.1688 & -0.1493 & 0.1123 & 0.0081 & 0.0427 \\ -0.0226 & -0.0071 & -0.1014 & -0.0168 & 0.0553 & 0.0404 & -0.0458 & 0.0081 & 0.0693 & 0.0483 \\ -0.0481 & 0.0336 & -0.1467 & -0.0908 & 0.1282 & 0.1337 & -0.1370 & 0.0427 & 0.0483 & 0.0871 \end{bmatrix}$$

- The estimated a posteriori probability of stability is p_{feas}=1.0

Comments

- Two randomized algorithms for determining *robustly feasible* or *approximately feasible* solutions to robust LMIs
- For the latter algorithms convergence is only asymptotic
- These techniques are useful for problems which are intractable by means of standard (exact) LMI methods
- These solutions may serve as a good initial guess for a deterministic algorithm

Optimization Problems[1]

- Extensions to optimization problems
- Consider convex function $f(x)$ and function $g(x,\Delta)$ convex in x for fixed Δ
- Semi-infinite (nonlinear) programming problem

$$\min f(x)$$
$$g(x,\Delta) \text{ for all } \Delta \in \mathcal{B}$$

- Reformulation as stochastic optimization
- Drawback: Convergence results are only asymptotic

[1] V. B. Tadic, S. P. Meyn and R. Tempo (2003)

Scenario Approach[1]

- The scenario approach for convex problems
- Non-sequential method which provides a one-shot solution for general convex problems
- Randomization of $\Delta \in \mathcal{B}$ and solution of a single convex optimization problem
- Derivation of a new bound on the sample size
- Applications to control systems design

[1] G. Calafiore and M. Campi (2004)

CHAPTER 8

Probabilistic LPV Systems

Keywords: *gain scheduling, Linear-Parameter Varying systems, parameter discretization*

LPV

- LPV applications: Aircraft control, automated lane guidance, communication networks
- Parameters $q=q(t)$ are unknown but bounded in set $\mathcal{B}_q$
- They can be measured on-line by the controller
- Critical issue: Parameter discretization (complexity)

Quadratic LPV

- LPV plant

$$\begin{bmatrix} \dot{x}(t) \\ e(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} A(q(t)) & B_1(q(t)) & B_2(q(t)) \\ C_1(q(t)) & 0 & D_{12}(q(t)) \\ C_2(q(t)) & D_{21}(q(t)) & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ d(t) \\ u(t) \end{bmatrix}$$

with $q \in \mathcal{B}_q$

- Assumption: Orthogonality conditions are satisfied

Quadratic LPV - 2

- The main goal is to design an LPV controller of the type

$$\begin{bmatrix} \dot{x}_c(t) \\ u(t) \end{bmatrix} = \begin{bmatrix} A_c(q(t)) & B_c(q(t)) \\ C_c(q(t)) & 0 \end{bmatrix} \begin{bmatrix} x_c(t) \\ y(t) \end{bmatrix}$$

such that quadratic performance of the closed-loop system is guaranteed and

$$\sup_d \frac{\left(\int_0^\infty e^T(t)e(t)dt\right)^{\frac{1}{2}}}{\left(\int_0^\infty d^T(t)d(t)dt\right)^{\frac{1}{2}}} < \gamma \quad \forall q \in \mathcal{B}_q$$

Structured QMI Solution[1]

- The LPV problem is solvable if and only if there exist $X=X^T > 0$ and $Y=Y^T > 0$ such that for $\varepsilon > 0$

$$P(X,q) = A(q)X + XA^T(q) + XC_1^T(q)C_1(q)X + \gamma^{-2}B_1(q)B_1^T(q) - B_2(q)B_2^T(q) + \varepsilon I \le 0$$

$$Q(Y,q) = A^T(q)Y + YA(q) + YB_1(q)B_1^T(q)Y + \gamma^{-2}C_1^T(q)C_1(q) - C_2^T(q)C_2(q) + \varepsilon I \le 0$$

$$R(X,Y) = -\begin{bmatrix} X & \gamma^{-1}I \\ \gamma^{-1}I & Y \end{bmatrix} \le 0$$

[1] G. Becker and A. Packard (1994)

Matrix Valued Function

- Define a matrix valued function

$$V(X,Y,q) = \begin{bmatrix} P(X,q) & 0 & 0 \\ 0 & Q(Y,q) & 0 \\ 0 & 0 & R(X,Y) \end{bmatrix}$$

and a scalar function

$$v(X,Y,q) = \left\| [V(X,Y,q)]^+ \right\|$$

- Remark: The gradients $\partial_X\{v(X,Y,q)\}$ and $\partial_Y\{v(X,Y,q)\}$ can be computed in closed form

Gradient-based Algorithm

- We write

$$X^{k+1} = \left[X^k - \frac{\mu^k \partial_X \{v(X^k,Y^k,q^k)\}}{w(X^k,Y^k,q^k)} \right]^+$$

$$Y^{k+1} = \left[Y^k - \frac{\mu^k \partial_Y \{v(X^k,Y^k,q^k)\}}{w(X^k,Y^k,q^k)} \right]^+$$

if $v(X^k,Y^k,q^k) > 0$, or $X^{k+1} = X^k$ and $Y^{k+1} = Y^k$ otherwise

- Here μ^k is a stepsize and

$$w(X^k,Y^k,q^k) = \left(\left\|\partial_X\{v(X^k,Y^k,q^k)\}\right\|^2 + \left\|\partial_Y\{v(X^k,Y^k,q^k)\}\right\|^2 \right)^{\frac{1}{2}}$$

Probabilistic LPV[1]

- Assume that q is random vector with support $\mathcal{B}_q$
- Consider positive measure within $\mathcal{B}_q$
- Theorem: The algorithm converges with probability one in a finite number of iterations
- Remark: No assumption on the dependence on q of matrices A, B_1, B_2, C_1, C_2, D_{12}, D_{21}

[1] Y. Fujisaki, F. Dabbene and R. Tempo (2003)

Application Example

- Multivariable example for the design of a controller for the lateral motion of an aircraft (2 inputs, 3 outputs, 4 state variables)
- Nine uncertain parameters entering multiaffinely into the state matrix
- Each parameter is perturbed by relative uncertainty ρ

Numerical Results

- Sequential algorithm
  - Initial conditions randomly selected
  - 800 iterations
  - Computation of X^k and Y^k
  - Construction of the controller
- We set γ=3
- With ρ = 5% X^{800} and Y^{800} satisfy all 2^9=512 vertex QMIs

Numerical Results

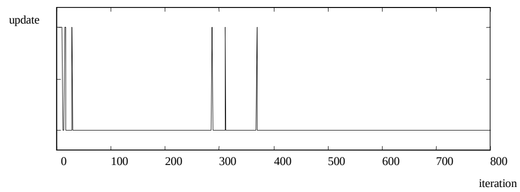
- With ρ = 8% X^{800} and Y^{800} still satisfy all 512 vertex QMIs

$$X^{800} = \begin{bmatrix} 0.6874 & -0.0715 & -0.0035 & 0.1461 \\ -0.0715 & 0.4841 & 0.0687 & 0.0299 \\ -0.0035 & 0.0687 & 0.3395 & 0.1408 \\ 0.1461 & 0.0299 & 0.1408 & 0.5002 \end{bmatrix}$$

$$Y^{800} = \begin{bmatrix} 0.4814 & 0.1546 & 0.2158 & 0.1296 \\ 0.1546 & 0.4166 & 0.1335 & 0.1246 \\ 0.2158 & 0.1335 & 0.6073 & 0.0187 \\ 0.1296 & 0.1246 & 0.0187 & 0.4611 \end{bmatrix}$$

Algorithm Updates

- The plot shows the algorithm updates



Fault Tolerant Control (FTC)

- Definition: Fault is an unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable/usual/standard condition[1]
- Example (dramatic): Chernobyl plant[2]

[1] R. Isermann and P. Ballè (1997)
[2] G. Stein, Bode Lecture "Respect the Unstable" (1989)

Robust Fault Tolerant Control[1]

- Discrete-time plant with time-varying faults f

$$x(k+1) = A(f)\,x(k) + B_1(f)\,d(k) + B_2(f)\,u(k)$$
$$e(k) = C_1(f)\,x(k) + D_{11}(f)\,d(k) + D_{12}(f)\,u(k)$$
$$y(k) = C_2(f)\,x(k) + D_{21}(f)\,d(k) + D_{22}(f)\,u(k)$$

The vector of faults is $f = [f_1,\ldots,f_n]$

$$f_i(k) = (1+w_{f,i}(k)\,\delta_i)\,g_i(k)$$

where $g_i(k)$ is the nominal value and the fault estimate uncertainty δ_i is bounded by

$$|\,\delta_i\,| \leq 1$$

[1] S. Kanev and M. Verhaegen (2004)

Robust Fault Tolerant Control

- Reformulation of the problem as LPV
- Fault estimate uncertainty δ is taken as a random variable with given pdf
- Development of randomized algorithms for computing controller parameters[1,2]
- Random generation of δ and use of ellipsoidal method
- Application: Brushless DC motor

[1] S. Kanev and M. Verhaegen (2004)
[2] S. Kanev (2004)

Model Predictive Control (MPC)[1]

- Development of randomized algorithms for MPC is an open area
- The separation principle does not hold when parametric uncertainty is present
- Difficulty of deriving output feedback controller
- BMI solution is $\mathcal{NP}$-hard
- Approaches based on finite-horizon output feedback MPC do not always guarantee robust stability of the closed-loop system

[1] E. F. Camacho and F. Bordons (2004)

CHAPTER 9

Randomized Algorithms for Switched Systems

Keywords: *common Lyapunov functions, piecewise quadratic function, switching rules*

Common Lyapunov Functions for Finite Families

Switched Systems Problem

We deal with a finite set of matrices

$$\mathcal{A} = \{A_i : i \in \mathcal{I}\}$$

Given Hurwitz matrices $A_i,\ i \in \mathcal{I}$ and matrix $Q > 0$, find matrix $P > 0$

$$PA_i + A_i^T P \leq -Q \ \ \forall i$$

Common Lyapunov Function

- If $P > 0$ exists, then the quadratic function

$$V(x) = x^T P x$$

is a common Lyapunov function for the family of asymptotically stable linear systems

Special Case[1]

In the special case when matrices commute

$$A_i A_j = A_j A_i \quad \forall i, j$$

$\exists$ quadratic common Lyapunov function

$$P_1 A_1 + A_1^T P_1 = -I$$
$$\vdots$$
$$P_m A_m + A_m^T P_m = -P_{m-1}$$

[1] K. S. Narendra and J. Balakrishnan (1994)

Function *f*

- Function *f* is convex differentiable real-valued functional on the space of symmetric matrices with the property

$$f(R) \leq 0 \quad \text{if and only if} \quad R \leq 0$$

Two Examples of Functions f

1. If λ_{max} is a simple eigenvalue we take

$$f(R) = \lambda_{max}(R)$$

2. $$f(R) = \| R^+ \|^2$$

where $\| \cdot \|$ is Frobenius norm, $^+$ is projection into the cone of non-negative definite matrices

Gradient Algorithms: Preliminaries

$v(P,A) := f(PA + A^T P + Q)$ – convex in P (given: Q)

1. $f(R) := \lambda_{\max}(R)$

Gradient: $\partial_P v(P,A) = Axx^T + xx^T A^T$

(x is unit eigenvector of R with eigenvalue $\lambda_{\max}(R)$)

2. $f(R) := \|R^+\|^2$

$\partial_P v = 2(AS + SA^T), \;\; S := (PA + A^T P + Q)^+$

Gradient Iteration

$A_1, \ldots, A_m$ – finite family of Hurwitz matrices

$h : \mathbb{Z}_+ \to \{1, \ldots, m\}$ – visits each index ∞ times

P_0 – arbitrary symmetric matrix

Gradient iteration: (μ_k – suitably chosen stepsize)

$$P_{k+1} = \begin{cases} P_k - \mu_k \partial_P v(P_k, A_{h(k)}), & v(P_k, A_{h(k)}) > 0 \\ P_k, & \text{otherwise} \end{cases}$$

Theorem[1]

- Theorem: Solution $P > 0$, if it exists, is found in finite number of steps
- Idea of proof: Distance of P_k from solution set decreases at each correction step

[1] D. Liberzon and R. Tempo (2004)

Example: Interval Family

- We study interval family of triangular Hurwitz matrices

$$A_k = \begin{pmatrix} a^i_{11} & \cdots & * \\ \vdots & \ddots & \vdots \\ 0 & \cdots & a^i_{nn} \end{pmatrix}, \; L_{ij} \le a^k_{ij} \le U_{ij}$$

- We have $2^{n(n+1)/2}$ vertices

Numerical Results

- Results with deterministic gradient
- $n = 4$ (2^{10} inequalities): 10,000 iterations (a few seconds)
- $n = 5$ (2^{15} inequalities): 10,000,000 iterations (a few hours)
- Randomized gradients gives faster convergence
- For comparison: quadstab MATLAB™ stacks when $n = 5$

Comments

- Contribution
- Gradient iteration algorithms for solving simultaneously Lyapunov matrix inequalities
- Deterministic convergence for finite families

Open Computational Issues

- Performance comparison of different methods
- Comparison with advanced LMI algorithms
- Addressing existence of solutions
- Optimal schedule of iterations
- Additional requirements on solution

Piecewise Quadratic Lyapunov Functions

Switched Systems: Another Approach

- Consider the switched system

$$dx(t)/dt = A_{\sigma(t)}\, x(t)$$

where $x(t)$ is the state and $\sigma(t) = 1, 2$ is the switch

- Beyond common quadratic Lyapunov functions: Piecewise quadratic Lyapunov functions
- Less conservative but harder to construct

Piecewise Lyapunov Functions[1,2]

- Given $\gamma > 0$ and $P_1, P_2 > 0$, suppose that for any x: $||x||=1$

$$x^T(P_1A_1 + A_1^T P_1)\, x \le -\gamma \quad \text{if} \quad x^T(P_1-P_2)x \ge 0$$

and

$$x^T(P_2A_2 + A_2^T P_2)\, x \le -\gamma \quad \text{if} \quad x^T(P_1-P_2)x \le 0$$

[1] M. Wicks and R. DeCarlo (1997)
[2] J. Malmborg, B. Bernhardsson and K. J. Astrom (1996)

Switching Rule

- Then, we construct a switching rule

$$\sigma(t) = \arg\max_{\{i=1,2\}}\ x(t)^T P_i\, x(t)$$

- The piecewise quadratic Lyapunov function

$$V(x) = \max_{\{i=1,2\}}\ x(t)^T P_i\, x(t)$$

decreases along trajectories of the system

- GAS is achieved

IEIIT-CNR

Solution via RA [1]

- This switched system problem is not convex
  Solution: Combination of randomized algorithms
- with branch and bound methods

- Design a piecewise quadratic Lyapunov function
- with probability one in a finite number of steps

- Specific RA and proof techniques are technical

[1] H. Ishii, T. Basar and R. Tempo (2005)

IEIIT-CNR

CHAPTER 10

Miscellaneous Topics

Keywords: *robustness in statistics, value set predictors*

IEIIT-CNR

Robustness in Statistics

IEIIT-CNR

Robustness in Statistics

- Parameter estimation

$$y_i = \theta^* + \xi_i \qquad i = 1,\ldots,\ell$$

$$\hat{\theta} = \arg\min_{\theta} \|y_i - \theta\| = \frac{1}{\ell}\sum_{i=1}^{\ell} y_i$$

- If $\xi \sim N(0,I)$ i.i.d., then $\hat{\theta}$ is the best estimate
- However, if there are outliers, that can be modeled as

$$\xi \sim (1-\varepsilon)N(0,I)+\varepsilon N(0,\sigma^2 I) \text{ with } \sigma^2 >> 1$$

then $\hat{\theta}$ may be very bad

IEIIT-CNR

Robust Estimates

- Consider

$$y_i = \theta^* + \xi_i \qquad i = 1,\ldots,\ell$$

- ξ is i.i.d. with some density $f_\xi(\xi)$ belonging to a class $\mathcal{P}$
- Then

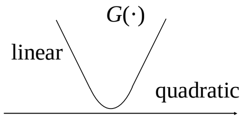
$$\theta_G = \arg\min_{\theta} \sum_{i=1}^{\ell} G(y_i - \theta)$$

- where $G(\cdot)$ is a given function, may be better than $\hat{\theta}$

IEIIT-CNR

Robust Estimates – Choice of *G*

- For instance



is good for normal contaminated distributions
- θ_G is an intermediate between $\hat{\theta}$ and the median

Robust Estimates – Optimal *G*

- Fisher information

$$I(f_\xi) = \int \frac{\left(f_\xi^{'}(t)\right)^2}{f_\xi(t)} dt$$

- The least favorable distribution is given by

$$f_\xi^* = \arg\min_{f_\xi \in \mathcal{P}} I(f_\xi)$$

- The best G in the sense of asymptotic variance of θ_G is

$$G^*(t) = -\ln f_\xi^*(t)$$

Value Set Predictors

Robust Stability – Value Set

- Consider the uncertain closed-loop polynomial

$$p(s,q) = a_0(q) + a_1(q)s + \cdots + a_n(q)s^n$$

with q varying in the box $\mathcal{B}_q = \{q \in \mathbb{R}^n, \|q\|_\infty \leq 1\}$

- Zero exclusion principle: define the value set as

$$V(\omega) \doteq \{p(j\omega, q) : q \in \mathcal{B}_q\}$$

- If $p(s,0)$ is stable and

$$0 \notin V(\omega) \ \text{ for all } 0 \leq \omega \leq \infty$$

then robust stability holds

Probabilistic Predictors of Value Set

- Let q be a random vector uniformly distributed on $\mathcal{B}_q$
- Define the risk-adjusted value set

$$V_\gamma(\omega) : \Pr\{p(j\omega, q) \in V(\omega)\} \geq 1 - \gamma$$

- $V_\gamma(\omega)$ may be much smaller than $V(\omega)$

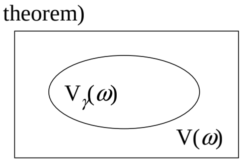
Example – Interval Polynomial

- Consider an interval polynomial

$$p(s,a) = a_0 + a_1 s + a_2 s^2 + \cdots + s^n$$

where $|a_i - a_i^0| \leq r\alpha_i$, $i=1,\ldots,n-1$

- For fixed frequency, the value set is a rectangle (Kharitonov theorem)



Example – Spherical Polynomial - 2

- Consider a spherical polynomial

$$p(s,a) = a_0 + a_1 s + a_2 s^2 + \cdots + s^n$$

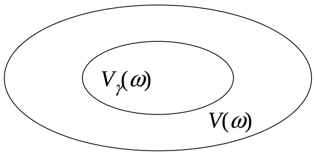
where

$$\sum_{i=0}^{n-1} \frac{(a_i - a_i^0)^2}{\alpha_i^2} \leq r$$

- $V(\omega)$ and $V_\gamma(\omega)$ are ellipses, $V_\gamma(\omega)$ can be rigorously calculated

Example – Spherical Polynomial - 2

- Then



Linear Functions of Uniform Vectors

- Let q be uniformly distributed on a ball

$$q \sim \mathcal{U}[\mathcal{B}_q] \quad q \in \mathbb{R}^n$$

- Then, for $A \in \mathbb{R}^{m,n}$

$$\tau \doteq \left((AA^T)^{-1} Aq, Aq \right)$$

has beta distribution with density

$$f_\tau(\tau) = \frac{\Gamma\left(\frac{n}{2}+1\right)}{\Gamma\left(\frac{m}{2}\right)\Gamma\left(\frac{n-m}{2}+1\right)} \tau^{\frac{m}{2}-1}(1-\tau)^{\frac{n-m}{2}} \quad 0 \le \tau \le 1$$

CHAPTER 11

Mixed Deterministic/Randomized Methods

Keywords: *deterministically tractable and intractable parameters, stability and performance with fixed order controllers*

Randomized vs Deterministic Solutions

- Development of RAs motivated by intractability of various problems in systems and control
- RAs provide (efficiently) a probabilistic solution
- The drawback is that this solution is given with a certain accuracy ε and confidence δ

- Deterministic algorithms provide a strong solution, but can be used for a narrow class of problems
- Randomized algorithms give a weaker solution, but can be utilized for a broader set of problems

Mixed Randomized and Deterministic Methods

- Key idea: Development of algorithms which combine randomized and deterministic techniques[1,2]
- Problem $P(\eta,\theta)$ with parameters η and θ
- We divide these parameters in two sets consisting of (deterministically) tractable and intractable

- η are tractable parameters → deterministic methods
- θ are intractable parameters → randomized techniques

[1] Y. Fujisaki, Y. Oishi and R. Tempo (2004)
[2] Y. Fujisaki, Y. Oishi and R. Tempo (2005)

Stabilization with Fixed Order Controller

- Specific problem $\Pi(\theta,\eta)$
- Strictly proper SISO plant $P(s) = N_P(s)/D_P(s)$
- Fixed order controller

$$C(s) = \frac{N_C(s)}{D_C(s)} = \frac{X(s^2) + sY(s^2)}{Z(s^2) + sV(s^2)}$$

where $X(s^2)$, $Y(s^2)$, $Z(s^2)$ and $V(s^2)$ are even polynomials in s

- Objective: Find a stabilizing controller placing the roots of the closed-loop polynomial in the open LHP

Tractable vs Intractable Parameters

- Definition: The (deterministically) tractable parameters are the coefficients θ of $X(s^2)$ and the intractable parameters are the coefficients η of $Y(s^2)$, $Z(s^2)$ and $V(s^2)$

Step 1: We use RAs to compute the coefficients η
Step 2: We use deterministic methods to determine the coefficients θ

The set of all tractable parameters θ is denoted by $\boldsymbol{\theta}$

Characterization of Tractable Parameter Set

- Lemma: Suppose that the parameters η are selected according to a RA. Then, the set $\boldsymbol{\theta}$ of all (tractable) stabilizing controller parameters is either empty or is a union of a finite number of polyhedral sets
- Exploitation of this result to determine a stabilizing controller

Polynomial-Time Algorithm

- We construct an algorithm which proceeds in two phases:

1. Computation of a marginal stabilizing controller by means of a method based on matrix inversion
2. Computation of a stabilizing controller using a sensitivity approach

The resulting algorithm is polynomial-time

Extensions to $\mathcal{H}_\infty$ Performance

- Consider a performance problem with sensitivity function

$$S(s) \doteq \frac{1}{1+P(s)C(s)}$$

and a (stable) weighting function $W(s)$
- The objective is to find a controller $C(s)$ satisfying

$$\|W(s)S(s)\|_\infty \leq 1$$

Characterization for $\mathcal{H}_\infty$ Performance

- Theorem: Suppose that the parameters η are chosen with a RA. Then, the set $\boldsymbol{\theta}$ of all tractable controller parameters satisfying

$$\|W(s)S(s)\|_\infty \leq 1$$

is either empty or is given by

$$\boldsymbol{\Theta} = \bigcup_{i=1,2,\ldots<\infty} \bigcap_{\phi\in[0,2\pi]} \Gamma_i(\phi)$$

where $\Gamma_i(\phi)$ is a (possibly unbounded) polyhedral set for fixed ϕ and i

Further Results and Extensions

- Development of polynomial-time algorithms for $\mathcal{H}_\infty$ performance based on matrix inversion and sensitivity methods, adding an extra parameter $\phi \in [0, 2\pi]$
- Extensions to uncertain plants where $P(s)$ is replaced by an interval plant $P(s,q,r)$
- This extension can be carried on invoking some existing results of parametric stability and by means of a additional parameter $\lambda \in [0, 1]$

CHAPTER 12

Applications of Randomized Algorithms

Keywords: *flexible structures, high-speed networks, brushless DC motors, quantized systems*

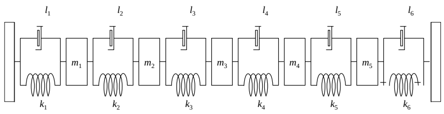
Application of RAs

- Randomized algorithms have been developed for various specific applications
- Control of flexible structures
- Stability and robustness of high speed networks
- Stability of quantized sampled-data systems
- Brushless DC motors
- …

Application 1

Control of Flexible Structures

Flexible Structure



- Mass spring damper model
- Real parametric uncertainty affecting stiffness and damping
- Complex unmodeled dynamics (nonparametric)

Flexible Structure

- *M*-Δ configuration for controlled systems

$$M(s)=C(sI-A)^{-1}B$$

$$\Delta = \begin{bmatrix} q_1 I_5 & 0 & 0 \\ 0 & q_2 I_5 & 0 \\ 0 & 0 & \Delta_1 \end{bmatrix}$$

$q_1, q_2 \in \mathbb{R}$

$\Delta_1 \in \mathbb{C}^{4,4}$

Probabilistic Stability Margin

- For fixed ρ, we let

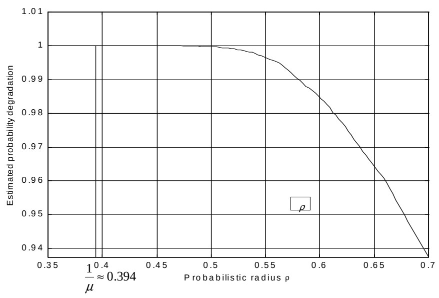
$$p(\rho) \doteq \Pr\{A + B\Delta C \text{ is stable}\}$$

- For given $p^* \in [0,1]$ we define the probabilistic stability margin

$$\rho(p^*) \doteq \sup\{\rho : p(\rho) \geq p^*\}$$

- Clearly $$\rho(p^*) \geq \frac{1}{\mu}$$

Probability Degradation Function

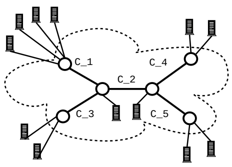


Application 2

Stability and Performance of High-Speed Networks

Stability and Robustness

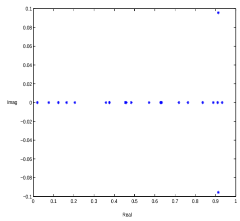
- Network topology
- Source and destination nodes, links (with buffer and capacity)
- Bottleneck link
- Stability and robustness



Symmetric Single Bottleneck

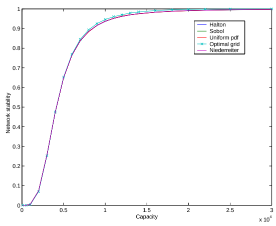
- Parametric stability (discrete time) with real uncertain parameters
- Stability and robustness can be studied in closed form
- Case study with 20 users
- Roots of the closed loop

polynomial (discrete-time)

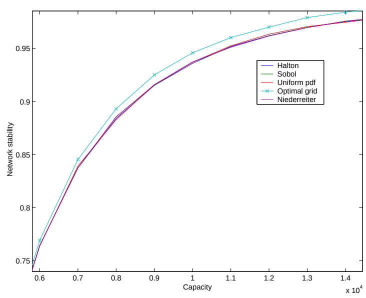


Non-Symmetric Single Bottleneck

- Closed form analysis is not possible
- We use RAs based on MC and QMC



Additional Simulations



Application 3

Probabilistic Structured Real Stability Radius

Course on Randomized Algorithms, Seville © IEIIT – CNR 2005 301

Structured Real Stability Radius

- Let $A \in \mathbb{C}^{n,n}$ be a stable matrix, and consider the perturbed matrix

$$A(\Delta) = A + B\Delta C,\ \Delta \in \mathcal{B}_D$$

with B, C of appropriate dimensions

- The real stability radius is the size of the smallest destabilizing perturbation

Course on Randomized Algorithms, Seville © IEIIT – CNR 2005 302

Probabilistic Stability Radius

- We assume Δ random, and estimate the probability of stability as a function of the uncertainty radius ρ

$$p(\rho) = \Pr\{A(\Delta) \text{ is stable}, \Delta \in \mathcal{B}_D\}$$

- For given p^*, the probabilistic real stability radius is defined as

$$\rho_R(A, p^*) \doteq \sup\{\rho : p(\rho) \geq p^*\}$$

- We estimate the probabilistic stability radius using randomized algorithms

Course on Randomized Algorithms, Seville © IEIIT – CNR 2005 303

Numerical Example

- We studied the example

$$A = \begin{bmatrix} -0.9319 & 0.9633 & 1.1021 & 2.8166 & -1.5852 & -1.3271 \\ -3.5667 & 1.4700 & 2.3962 & 5.2311 & -2.8212 & -4.2641 \\ 1.4202 & -1.1677 & -1.6874 & -3.3362 & 1.0364 & 2.8705 \\ -0.1946 & 0.6813 & 0.0580 & 0.4244 & -0.2107 & -0.6973 \\ 1.2169 & -0.3964 & -0.8681 & -1.9139 & 0.1026 & 0.7190 \\ -2.8445 & 2.0764 & 1.4435 & 4.1812 & -1.8238 & -2.9809 \end{bmatrix}$$

$B = C = I$

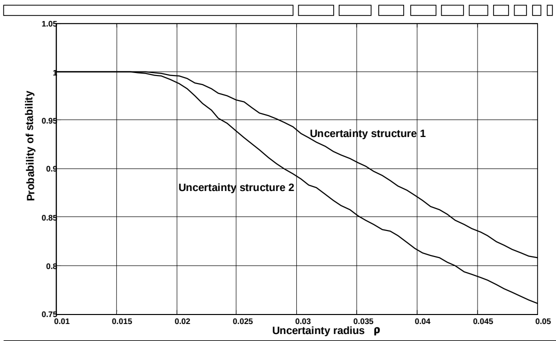
Course on Randomized Algorithms, Seville © IEIIT – CNR 2005 304

Numerical Example - 2

- We computed $p(\rho)$ for $\rho \in [0.01\ 0.05]$ with two different structures:
  - Δ composed by three 2x2 full real blocks
  - Δ composed by a 4x4 and a 2x2 full real blocks

Course on Randomized Algorithms, Seville © IEIIT – CNR 2005 305

Numerical Example - 3



Course on Randomized Algorithms, Seville © IEIIT – CNR 2005 306

Application 4

Probabilistic Robust Least Squares

Uncertain Least Squares

- Consider the problem

$$A(\Delta)x = b(\Delta)$$

with $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $\Delta \in \mathcal{B}_D$

- Robust least squares

$$\hat{x}_{RLS} = \arg\min_x \max_{\Delta \in \mathcal{B}_D} \|A(\Delta)x - b(\Delta)\|^2$$

- Probabilistic least squares

$$\hat{x} = \arg\min_x E\left\{\|A(\Delta)x - b(\Delta)\|^2\right\}$$

Probabilistic Robust Least Squares

- Generate N samples of Δ and compute

$$\hat{x}_N = \arg\min_x \hat{E}_N(x)$$

with

$$\hat{E}_N(x) \doteq \frac{1}{N}\sum_{i=1}^{N} \|A(\Delta^i) - b(\Delta^i)\|^2$$

- We can compute $\hat{x}_N$ recursively

$$\hat{x}_{k+1} = \hat{x}_k + Q_{k+1}^{-1} A^T(\Delta^{k+1})(b(\Delta^{k+1}) - A(\Delta^{k+1})\hat{x}_k),$$
$$Q_{k+1} = Q_k + A^T(\Delta^{k+1})A(\Delta^{k+1})$$

PRLS Example

- We consider an uncertain matrix $A(\Delta)=A+\Delta$, $\|\Delta\|<1$

$$A = \begin{bmatrix} 3 & 1 & 4 \\ 0 & 1 & 1 \\ -2 & 5 & 3 \\ 1 & 4 & 5.2 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 2 \\ 1 \\ 3 \end{bmatrix}$$

PRLS Example - 2

- We compute the estimates obtaining

$$\hat{x}_N = [-0.1768\ 0.1009\ 0.3448]^T;\ N = 10{,}000$$
$$\hat{x}_{LS} = [-10.0000\ -9.7285\ 9.9834]^T$$
$$\hat{x}_{RLS} = [-0.0312\ 0.2073\ 0.2055]^T$$

PRLS Example - 3

- To compare the previous estimates we computed the worst-case and the sample residuals
- Worst-case residuals are defined as

$$r_*^{wc} = \max_\Delta \|A(\Delta)x_* - b(\Delta)\|^2$$

we obtained

$r_N^{wc} = 2.6532$	$r_{LS}^{wc} = 18.9394$	$r_{RLS}^{wc} = 2.5720$

PRLS Example - 4

- Statistics for the sample residuals

$$r_*^i = \left\| A(\Delta^i)x_* - b(\Delta^i) \right\|^2$$

	Average	Peak	Sample Covariance
r_N	2.2650	2.6344	0.0198
r_{LS}	11.8962	18.5164	9.4090
r_{RLS}	2.2848	2.5515	0.0107

Conclusions and Discussion

Conclusions

- A lot of research remains to be done
- Applications: Randomized algorithms and sample generation for specific structures of uncertainty
- Theory: Beautiful and nonstandard math behind randomized algorithms
- Complexity: Studying computational complexity of these algorithms
- Computation: High dimensional problems are ill conditioned
- Other Directions: Randomized algorithms for MPC and FTC

(Controversial) Conclusions

- A number of years ago, closed-form meant writing down a “good looking” equation on a piece of paper
- Notion of closed-form solution has changed substantially
- Various experts (rightfully) convinced us that a new notion of closed-form is to re-write (if possible) a control problem as convex optimization

(Controversial) Conclusions

- Unfortunately, many important control problems are not convex
- In such cases, we can either introduce relaxation or use randomization
- Perhaps the control community should pay more attention to randomization accepting a different (and weaker) notion of problem solution

Discussion

IEIIT-CNR

Randomized Algorithms for Analysis and Control of Uncertain Systems

by RT, GC, FD with a foreword by M. Vidyasagar, Springer-Verlag, London, 2005

http://staff.polito.it/tempo/