Performance Evaluation: A Randomization Viewpoint

Roberto Tempo

IEIIT-CNR
Politecnico di Torino
tempo@polito.it

IEIIT-CNR

Performance Evaluation in Traditional Communities

- Methodologies and tools for performance evaluation in traditional communities including
  - control
  - signal processing
  - optimization
  - communication networks
  - computer science

Outline of this Lecture

- Performance evaluation with uncertainty
- Deterministic (worst case) and probabilistic approaches
- Monte Carlo simulations
- From Monte Carlo to randomized algorithms
- Sequential methods for design
- Statistical learning techniques

Performance Evaluation

Performance Evaluation

- Consider a stable transfer function $G(s)$

$$z = G(s)\, w$$

where w and z are disturbances and errors

Objective: Check if the peak of the Bode plot (magnitude) is smaller than a given performance level $\gamma > 0$

IEIIT-CNR

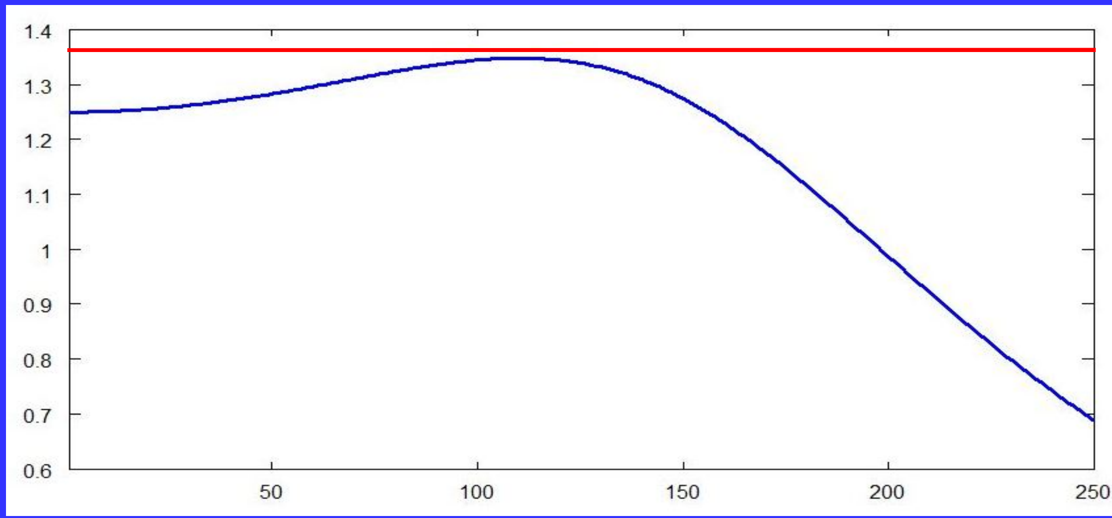
Performance Evaluation

- Consider a stable transfer function $G(s)$

$$z = G(s)\, w$$

where w and z are disturbances and errors

Bode plot (magnitude)



IEIIT-CNR

Example: Performance Evaluation

- Consider the linear system

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w \qquad z = \begin{bmatrix} 1 & 0 \end{bmatrix} x$$

with (nominal) parameters

$$a_0 = 1 \qquad a_1 = 0.8$$

- The transfer function is given by

$$G(s) = \frac{1}{s^2 + s + 0.8}$$

$\mathcal{H}_\infty$ Norm

- Compute the peak of the modulus of the frequency response of $G(s)$

$$z = G(s)\, w$$

- We are dealing with SISO systems
- If the system is stable, this peak is given by the $\mathcal{H}_\infty$ norm of the transfer function

$$\|G(s)\|_\infty = \sup_\omega |G(j\omega)|$$

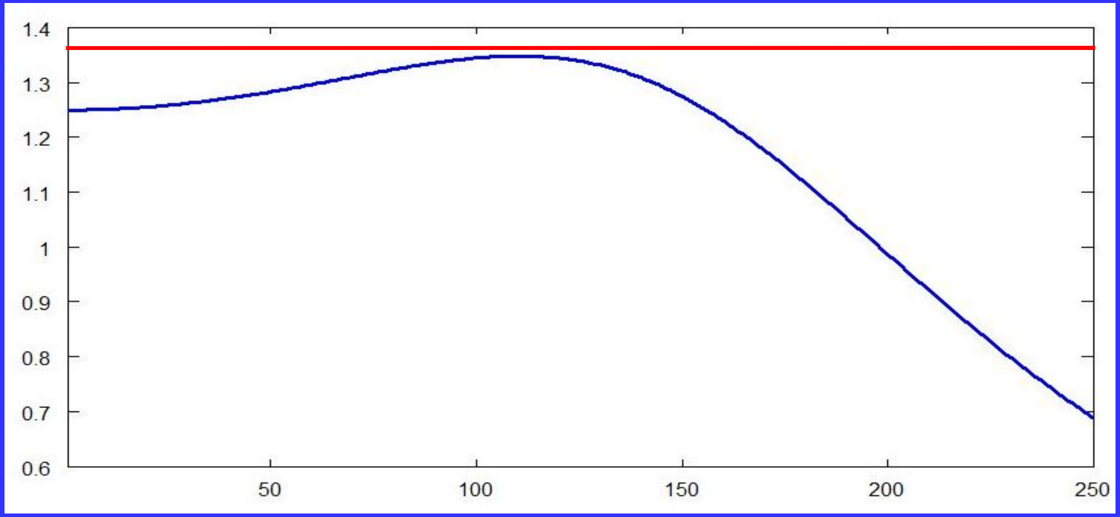
Example: $\mathcal{H}_\infty$ Norm

- $\mathcal{H}_\infty$ performance

$$\|G(s)\|_\infty = \sup_\omega |G(j\omega)| \leq \gamma$$

- Performance is satisfied for $\gamma = 1.35$

Bode plot (magnitude)



Performance Evaluation with Uncertainty

Performance Evaluation with Uncertainty

- Consider an uncertain stable transfer function $G(s,q)$

$$z = G(s,q)\,w$$

where w and z are disturbances and errors and q represents *uncertainty* bounded in a set Q of radius $\rho > 0$

Example[1]: Performance Evaluation with Uncertainty

IEIIT-CNR

- Consider the uncertain linear system

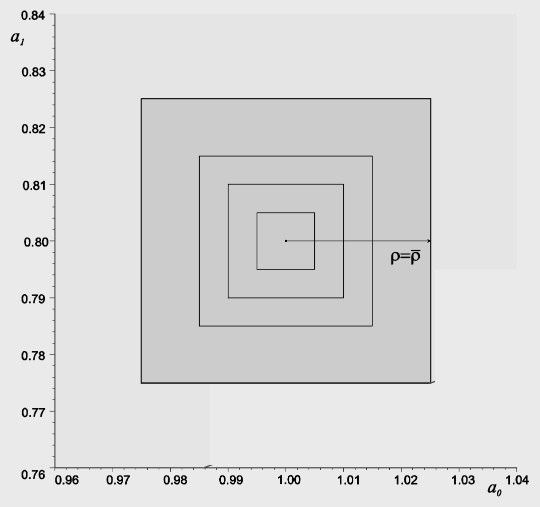
$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w \qquad z = \begin{bmatrix} 1 & 0 \end{bmatrix} x$$

with parameters

$$a_0 = 1 + q_0 \qquad a_1 = 0.8 + q_1$$

and bounding set

$$Q = \{q = [q_0\ q_1]^T : \|q\|_\infty \leq \rho\}$$



[1] G. Calafiore, F. Dabbene, R. Tempo (2010)

IEIIT-CNR

Interval Transfer Function

- Take ρ = 0.025, the interval transfer function is given by

$$G(s,q)=\frac{1}{s^2+(1+q_0)s+(0.8+q_1)}$$

where $q_0 \in [-0.025,0.025]$, $q_1 \in [-0.025, 0.025]$

$$G_1(s)=\frac{1}{s^2+(1+0.025)s+(0.8+0.025)} \qquad G_3(s)=\frac{1}{s^2+(1+0.025)s+(0.8-0.025)}$$

$$G_2(s)=\frac{1}{s^2+(1-0.025)s+(0.8-0.025)} \qquad G_4(s)=\frac{1}{s^2+(1-0.025)s+(0.8+0.025)}$$

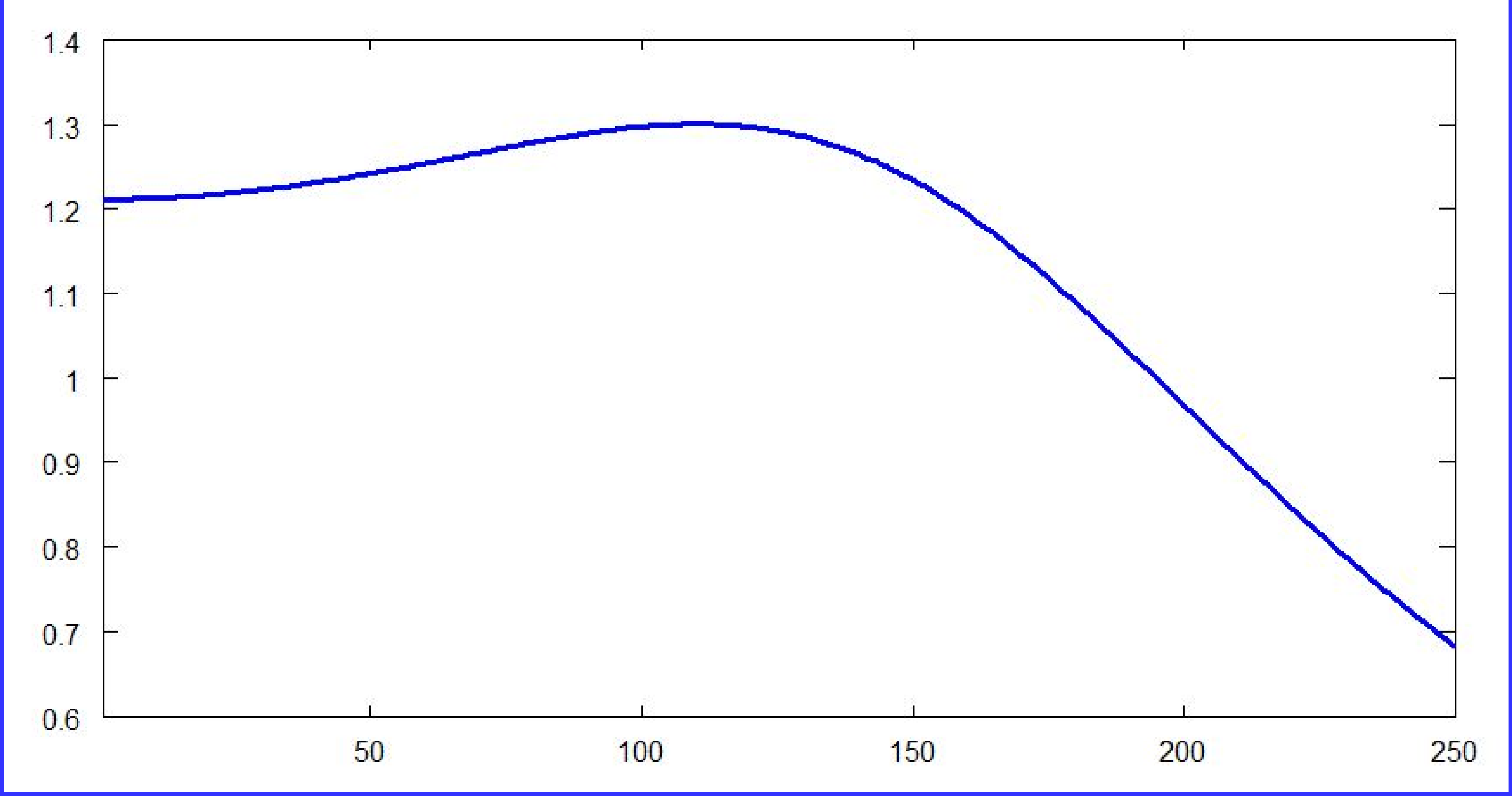
The Bode Envelope

- The *Bode envelope* of (general) interval transfer functions can be easily constructed by means of robust control theory
- Construct some fixed transfer functions using a subset of the vertices of Q
- For each frequency compute the maximum/ minimum values using the Bode plots of these vertices
- Worst case $\mathcal{H}_\infty$ norm can be easily computed[1]

[1] C. V. Hollot, R. Tempo (1991)

Example: Bode Plot of $G_1(s)$

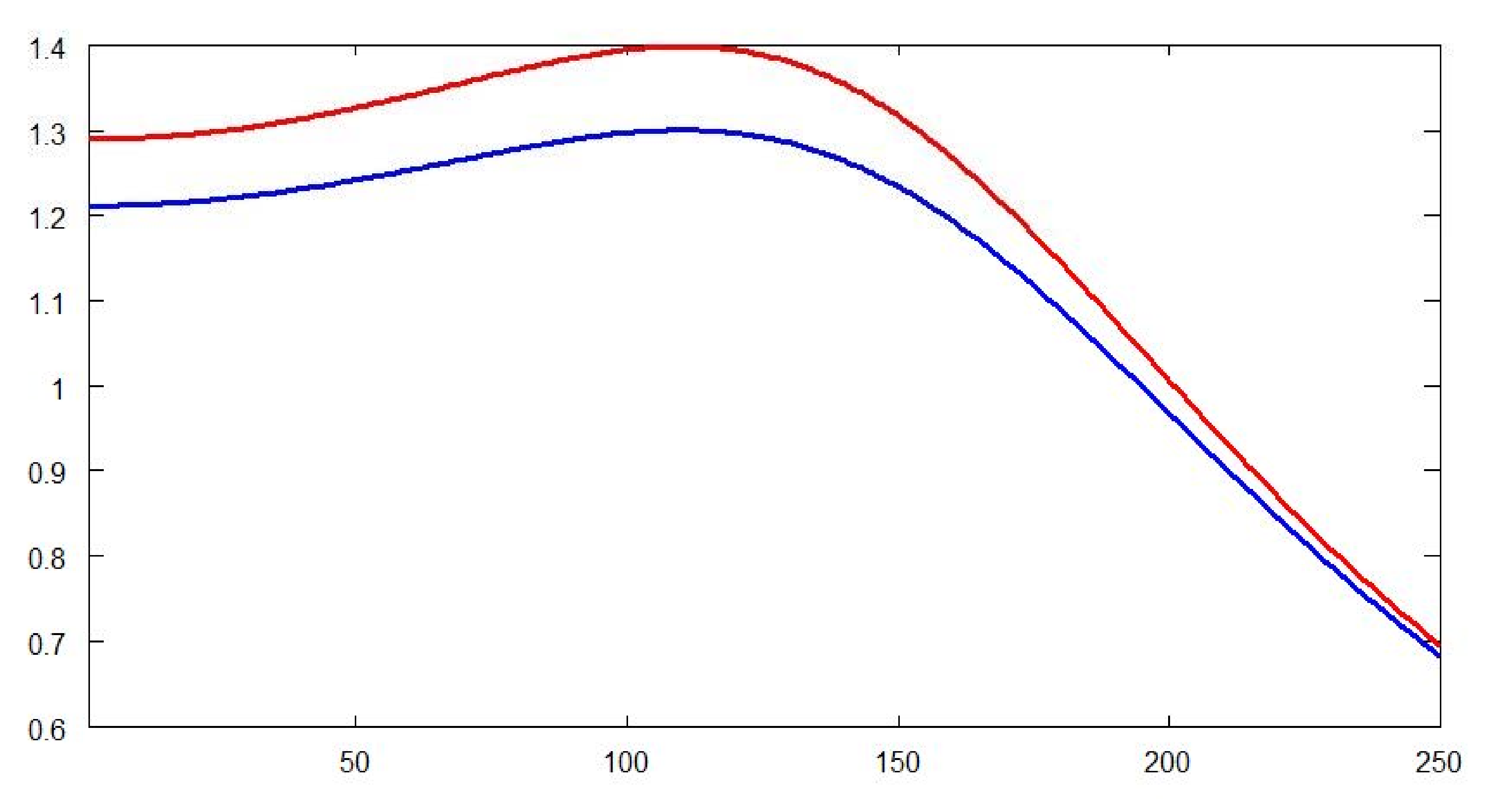
- Bode plot of the transfer function $G_1(s)$



Example: Bode Plot of $G_2(s)$

IEIIT-CNR

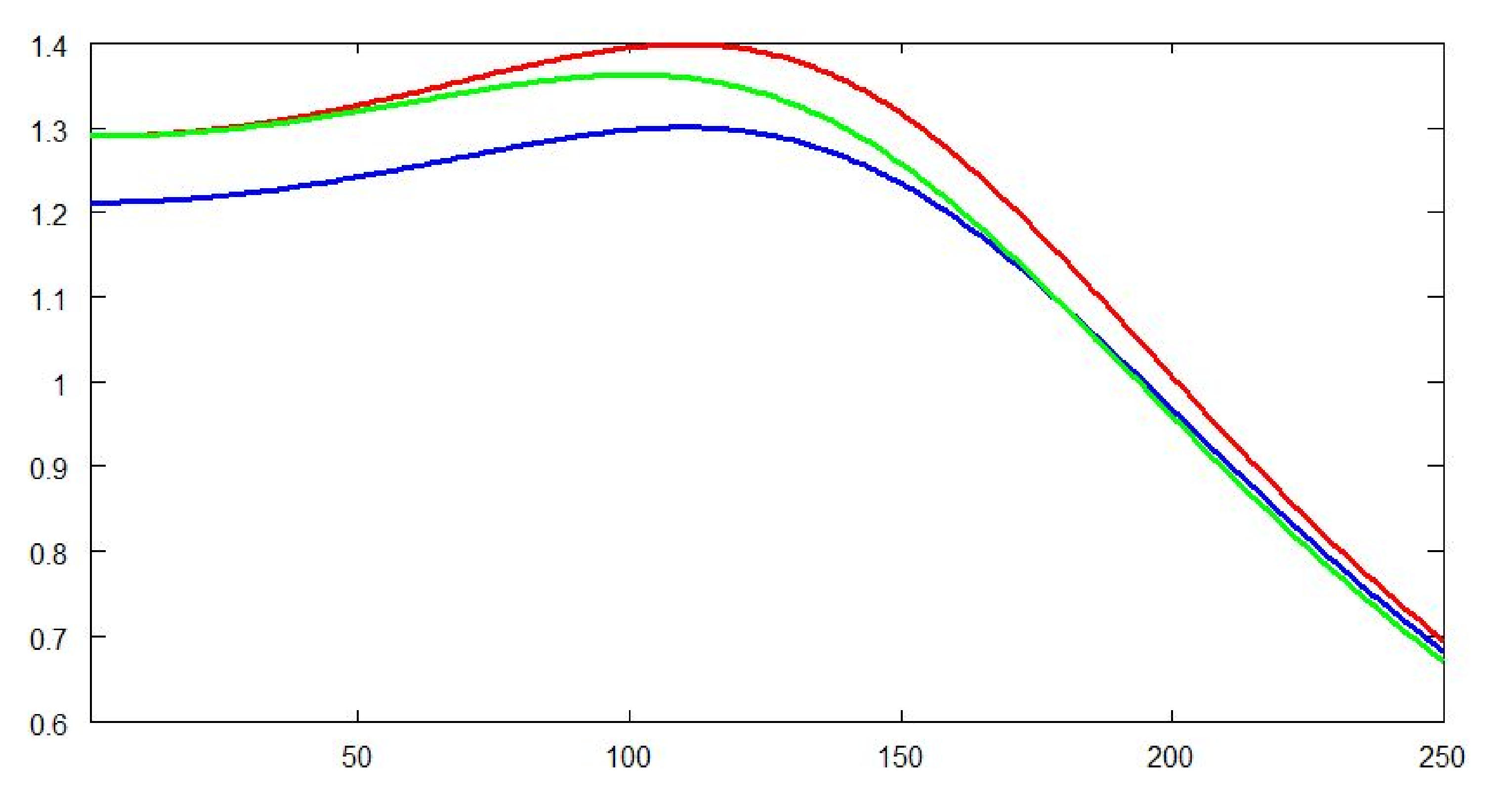
- Bode plot of the transfer function $G_2(s)$



Example: Bode Plot of $G_3(s)$

IEIIT-CNR

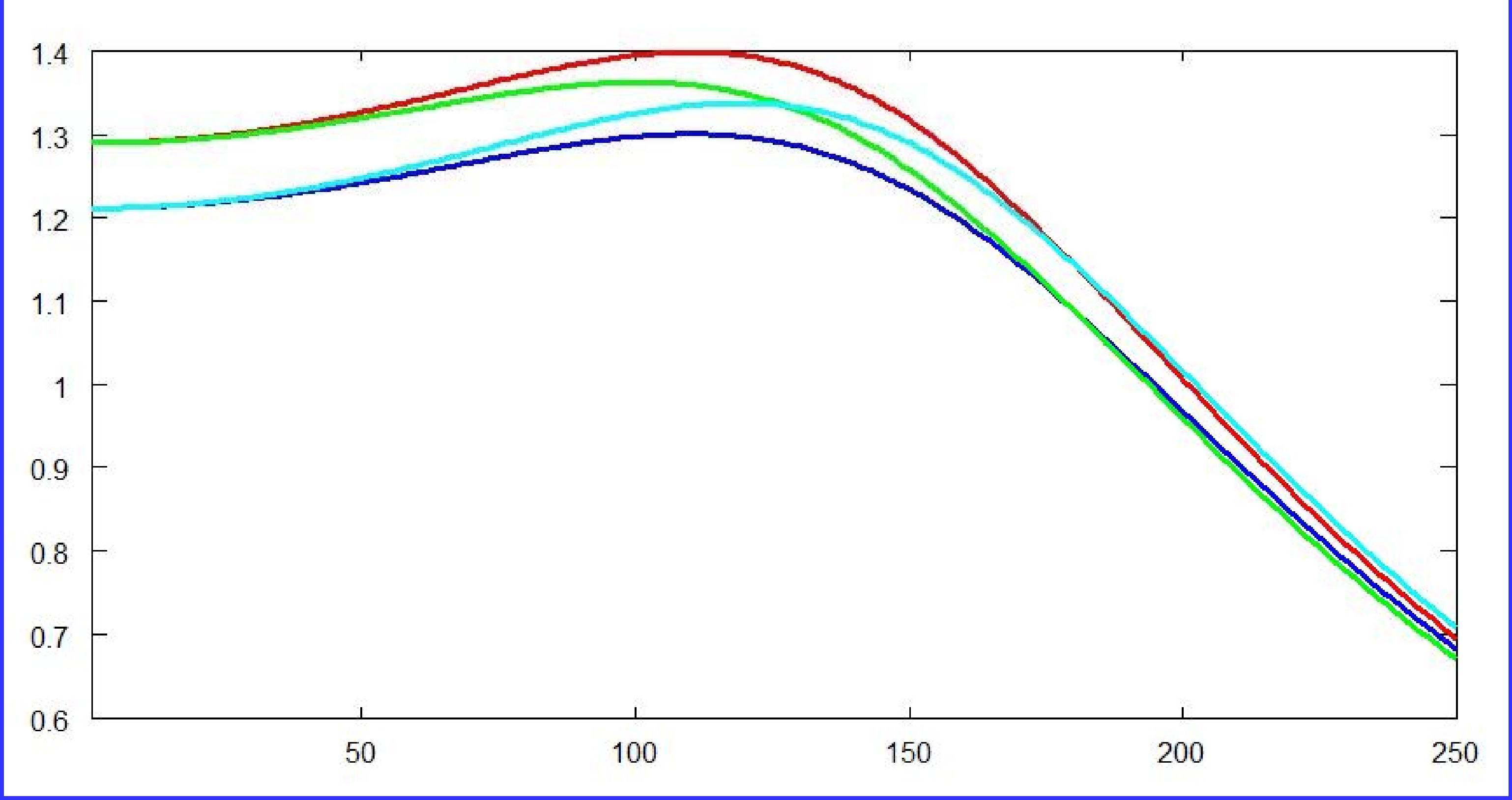
- Bode plot of the transfer function $G_3(s)$



IEIIT-CNR

Example: Bode Plot of $G_4(s)$

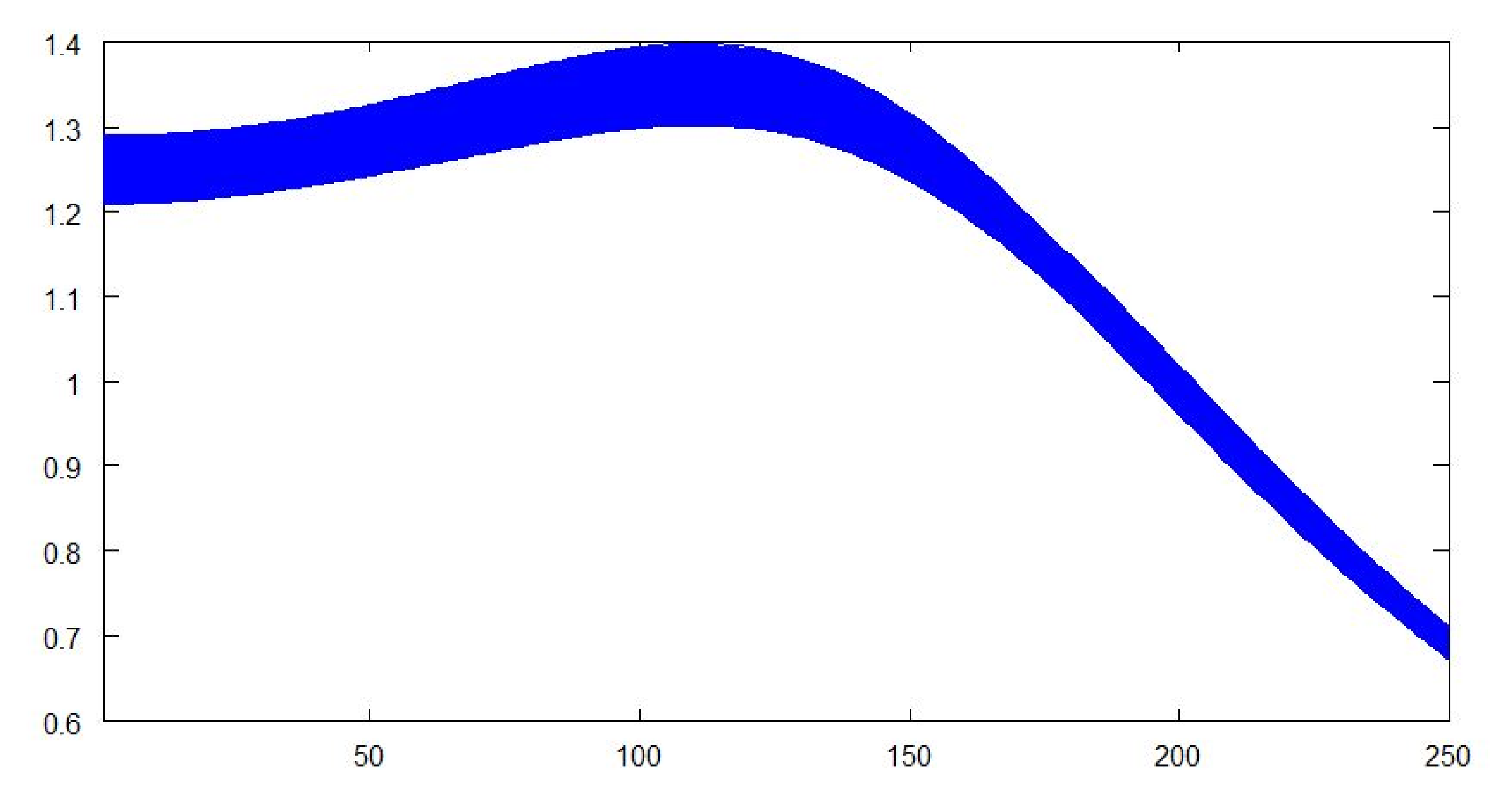
- Bode plot of the transfer function $G_4(s)$



IEIIT-CNR

Example: Bode Envelope of $G(s,q)$

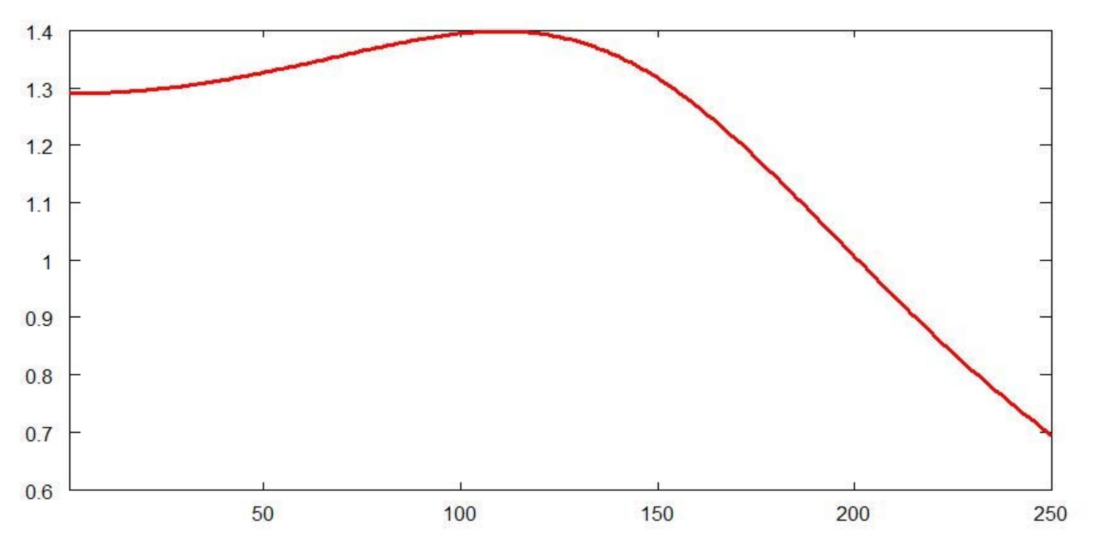
- Bode envelope of the interval transfer function



Example: Worst Case $\mathcal{H}_\infty$ Norm of $G(s,q)$

IEIIT-CNR

- Construction of the Bode envelope of the interval transfer function $G(s,q)$
- Worst case $\mathcal{H}_\infty$ norm is equal to $\gamma=\sqrt{2}$
- This value is achieved by $G_2(s)$



IEIIT-CNR

Example: Radius of Uncertainty

- Given performance level $\gamma=\sqrt{2}$ the objective is to compute the maximal radius $\bar{\rho}$ of Q such that

$$G(s,q) \text{ is stable and } \|G(s,q)\|_\infty \leq \gamma$$

for all $q \in Q$

- $G(s,q)$ is stable and $\|G(s,q)\|_\infty \leq \gamma$ if and only if

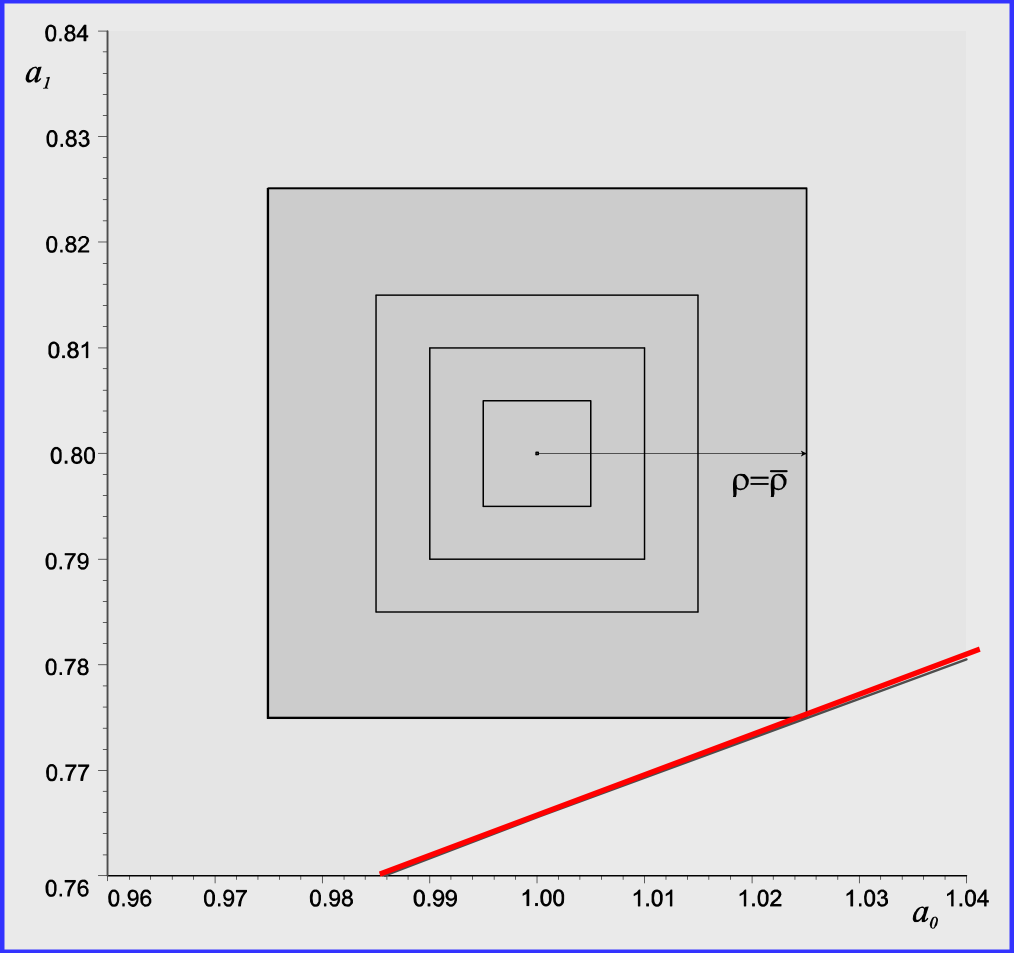
$$\rho < 0.8 \quad \text{and} \quad \frac{(0.8-\rho)^2}{2-\sqrt{2}} > 1+\rho$$

IEIIT-CNR

Example: Radius of Uncertainty

Largest radius of Q such that performance is satisfied is $\bar{\rho} = 0.025$

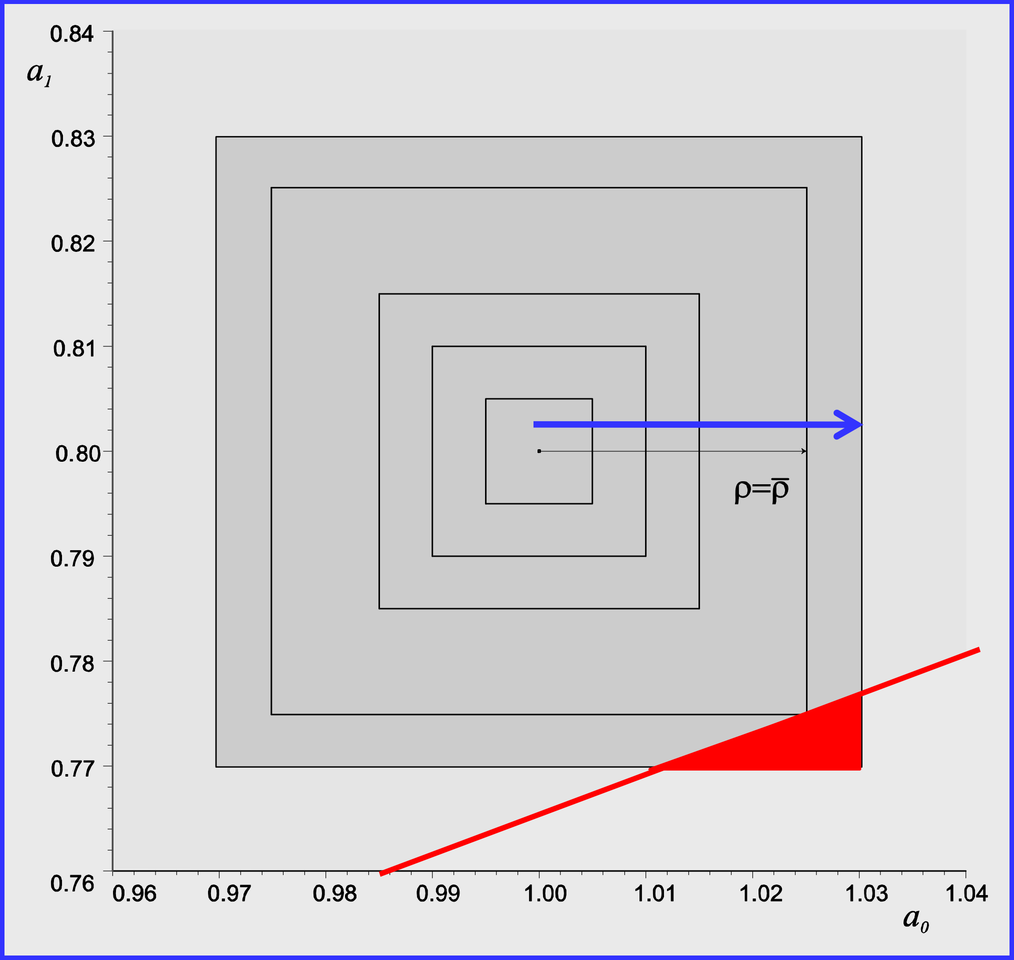
Conclusion: Stability and performance are satisfied for all $q \in Q$ with radius $\bar{\rho} = 0.025$



Example: Performance Violation

Increase the radius ρ

Observation: If we allow a small *violation* of performance we may increase the radius ρ significantly



Probabilistic Methods

IEIIT-CNR

Performance Violation

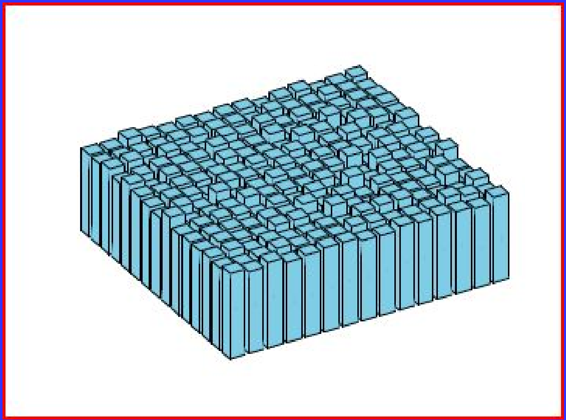
- **Question:** How can we quantify a violation of performance?
- We introduce a probabilistic model of uncertainty

IEIIT-CNR

Probabilistic Model of Uncertainty

- Probability density function associated to Q
- Assume that q is a random vector or matrix with given density function and support Q
- **Example:** Uniform density

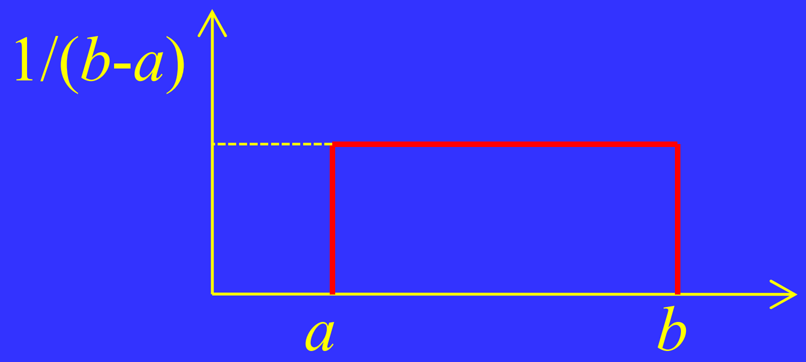
  $\mathcal{U}[Q]$ within Q



IEIIT-CNR

Uniform Density $\mathcal{U}[Q]$

- Univariate uniform density



- Multivariate uniform density

$$\mathcal{U}[Q] = \begin{cases} \dfrac{1}{vol(Q)} & \text{if } q \in Q \\ 0 & \text{otherwise} \end{cases}$$

Probability of Performance

- Define a performance function

$$J(q): Q \rightarrow \mathbf{R}$$

- Given level γ, the probability of performance is

$$\mathbf{P}_J = \mathrm{Prob}\{q \in Q: J(q) \leq \gamma\}$$

- **Example:** If $G(s,q)$ is stable and $J(q) = \|G(s,q)\|_\infty$

$$\mathbf{P}_J = \mathrm{Prob}\{q \in Q: \|G(s,q)\|_\infty \leq \gamma\}$$

Measure of Performance Violation

IEIIT-CNR

- **Objective:** Achieve probabilistic performance

$$\mathbf{P}_J = \mathrm{Prob}\{q \in Q: J(q) \leq \gamma\} \geq 1 - \varepsilon$$

where $\varepsilon \in (0,1)$ is a probabilistic parameter called *accuracy*

Computation of Probability of Performance

- Computing

$$\mathbf{P}_J = \text{Prob}\{q \in Q\colon J(q) \leq \gamma\}$$

requires to solve a difficult integration problem

- Taking uniform density $\mathcal{U}[Q]$

$$\text{Prob}\left\{q \in Q : J(q) \leq \gamma\right\} = \frac{\int_{J(q)\leq\gamma} \mathrm{d}\, q}{vol(Q)}$$

- In some special cases we can easily compute this probability

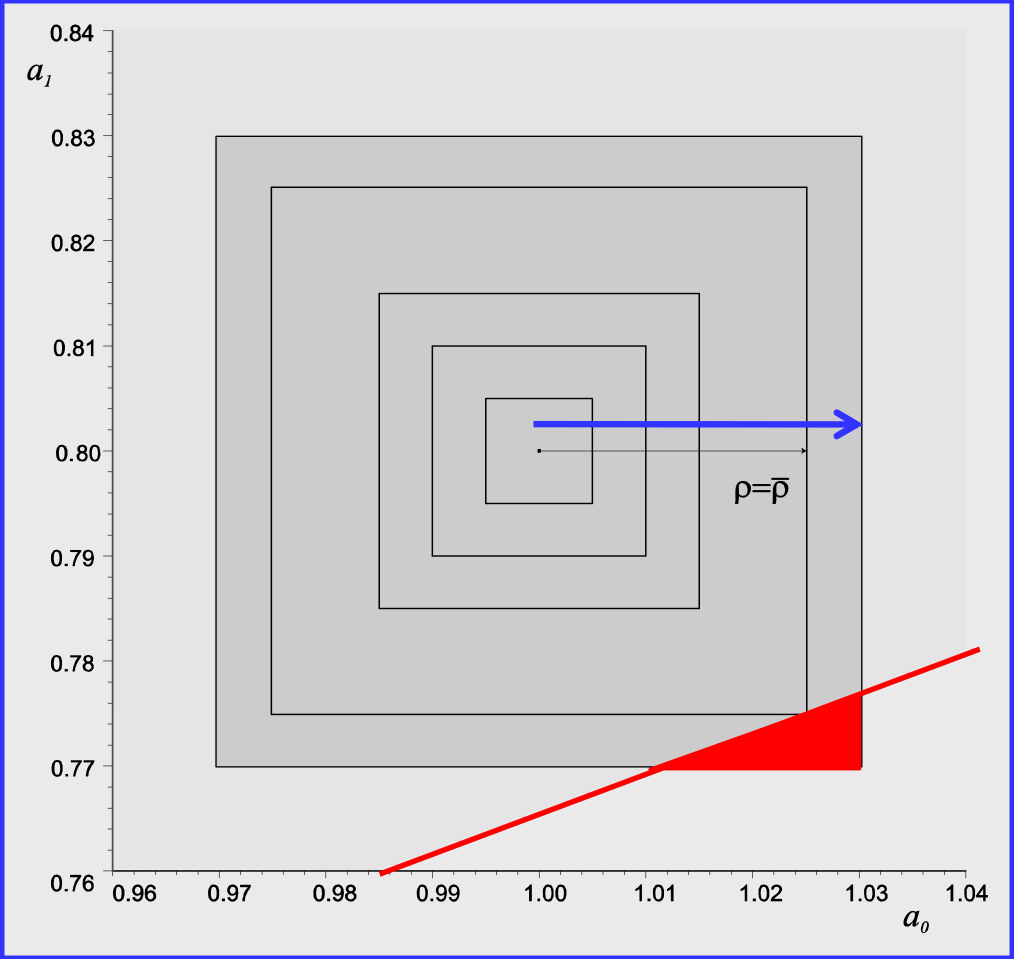
Example: Performance Violation

IEIIT-CNR

Take uniform pdf in Q

Allowing 5% violation we increase ρ of 54% obtaining 0.038 (instead of 0.025)

For several values of ρ we compute $\mathbf{P}_J(\rho)$

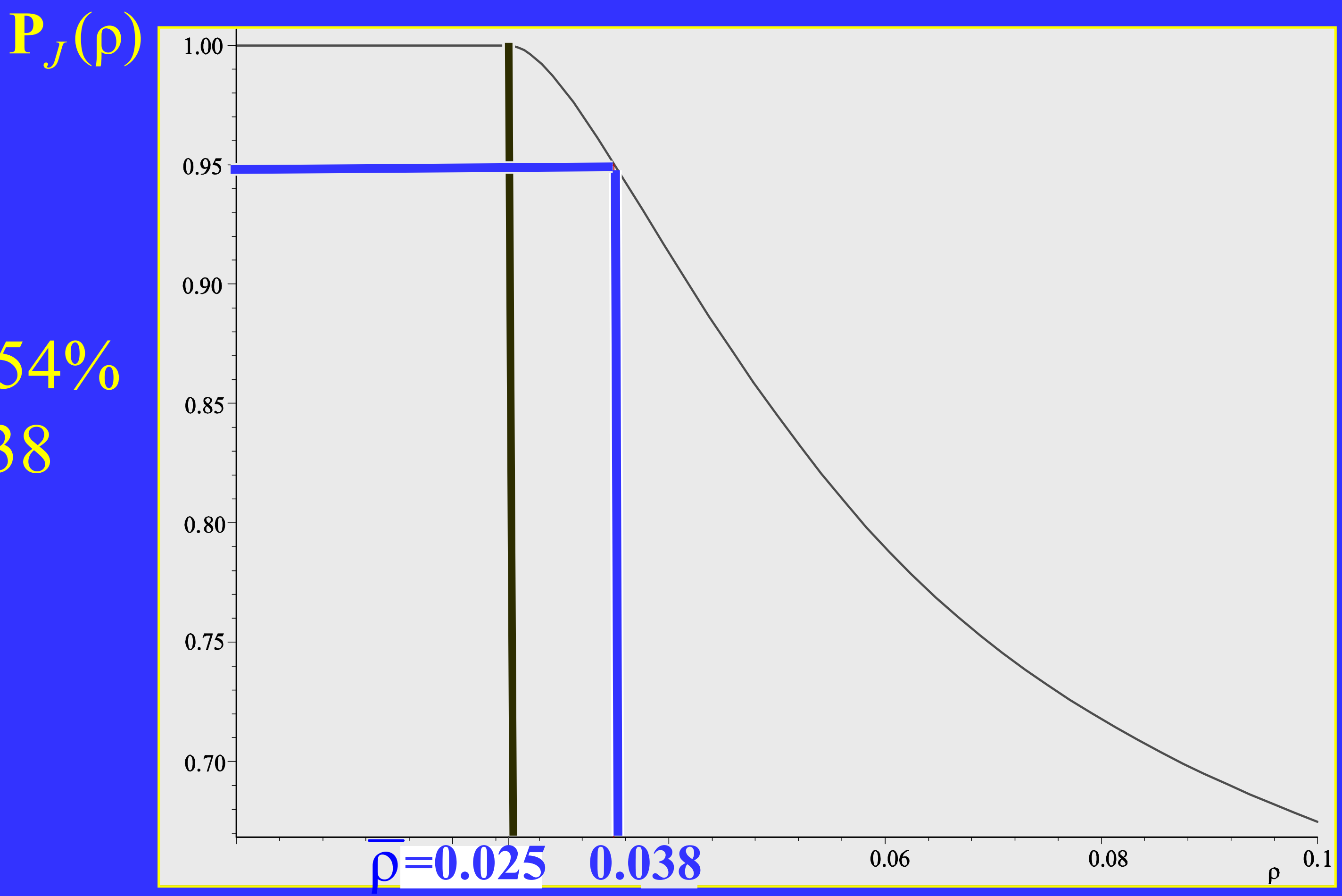


IEIIT-CNR

Degradation of Performance

If a 5% violation is allowed we increase ρ of 54% obtaining 0.038

Radius 0.038 compared to $\overline{\rho} = 0.025$



Advanced Simulation Tools

Monte Carlo Simulations

- Computation of probability of performance requires Monte Carlo simulations
- Take N i.i.d. random samples of q according to the given probability measure

$$q^{(1)}, q^{(2)}, \ldots, q^{(N)} \in Q$$

- This is called multisample

$$\mathrm{q}^{1,\ldots,N} = \{q^{(1)}, q^{(2)}, \ldots, q^{(N)}\}$$

- N is the sample complexity

IEIIT-CNR

Empirical Probability of Performance

- Evaluate

$$J(q^{(1)}),\, J(q^{(2)}),\, \ldots,\, J(q^{(N)})$$

- Construct the empirical probability of performance

$$\hat{\mathbf{P}}_J^N = \frac{1}{N} \sum_{i=1}^{N} \mathbf{I}\left(J(q^{(i)}) \right)$$

where $\mathbf{I}\,(\cdot)$ denotes the indicator function

$$\mathbf{I}(J(q^{(i)})) = \begin{cases} 1 & \text{if } J(q^{(i)}) \leq \gamma \\ 0 & \text{otherwise} \end{cases}$$

IEIIT-CNR

Law of Large Numbers

- Consider a probabilistic parameter $\delta \in (0,1)$ called confidence
- Monte Carlo analysis (Law of Large Numbers) studies the sample complexity such that the probability inequality

$$\left| \mathbf{P}_J - \hat{\mathbf{P}}_J^N \right| \leq \varepsilon$$

holds with probability at least 1- δ

IEIIT-CNR

(Additive) Chernoff Bound[1]

- Given ε, δ ∈(0,1), if

$$N \geq N_{\text{ch}} = \left\lceil \frac{\log \frac{2}{\delta}}{2\varepsilon^2} \right\rceil$$

then the probability inequality

$$\left| \mathbf{P}_J - \hat{\mathbf{P}}_J^N \right| \leq \varepsilon$$

holds with probability at least 1- δ

[1] H. Chernoff (1952)

IEIIT-CNR

Large Deviation Inequality

- Given $\varepsilon \in (0,1)$, the Hoeffding inequality states that

$$\mathrm{Prob}\{\mathrm{q}^{1,\ldots,N} : \left|\mathbf{P}_J - \hat{\mathbf{P}}_J^N\right| \geq \varepsilon\} \leq 2\mathrm{e}^{-2N\varepsilon^2}$$

  where e denotes the Euler number
- Hoeffding inequality provides a bound on the tail distribution
- Taking N sufficiently large, we can make the probability of error arbitrarily small
- Theory of rare events

IEIIT-CNR

Hoeffding Inequality and Chernoff Bound

- Consider the Hoeffding inequality

$$\mathrm{Prob}\{\mathrm{q}^{1,\ldots,N} : \left|\mathbf{P}_J - \hat{\mathbf{P}}_J^N\right| \geq \varepsilon\} \leq 2\mathrm{e}^{-2N\varepsilon^2}$$

- To guarantee confidence $\delta \in (0,1)$, we take N samples such that $2\mathrm{e}^{-2N\varepsilon^2} \leq \delta$ holds
- We obtain the (additive) Chernoff bound

$$N \geq N_{\mathrm{ch}} = \left\lceil \frac{\log \frac{2}{\delta}}{2\varepsilon^2} \right\rceil$$

IEIIT-CNR

Sample Complexity

- Chernoff bound provides a fundamental *explicit* relation (sample complexity) $N_{ch} = N_{ch}(\varepsilon, \delta)$ showing that $1/\varepsilon$ enters quadratically and $1/\delta$ logarithmically
- Sample complexity is independent on the problem dimensionality
- Confidence δ is "cheap" because of logarithm
- Accuracy ε is computationally more expensive
- Can we improve upon quadratic dependence?

(Multiplicative) Chernoff Bound

- (Multiplicative) Chernoff Bound

  Fox fixed $\beta=\beta(\hat{\mathbf{P}}_J^N)$ and for given ε, δ $\in(0,1)$, if

$$N \geq N_{\mathrm{mu}} = \left\lceil \frac{2\log\frac{1}{\delta}}{\varepsilon(1-\beta)^2} \right\rceil$$

  then the probability inequality

$$\left|\mathbf{P}_J - \hat{\mathbf{P}}_J^N\right| \leq \varepsilon$$

  holds with probability at least 1- δ

IEIIT-CNR

A Priori and A Posteriori Analysis

- Multiplicative Chernoff Bound has sample complexity $N_{mu} = N_{mu}(1/\varepsilon, \delta, \beta)$ but requires the parameter $\beta=\beta(\hat{\mathbf{P}}_J^N)$ which depends on the empirical probability (a posteriori analysis)
- Additive Chernoff Bound may be used for a priori analysis and has sample complexity $N_{ch} = N_{ch}(1/\varepsilon^2, \delta)$

IEIIT-CNR

Parallel and Distributed Simulations

- Samples $q^{(1)}, q^{(2)}, \ldots, q^{(N)}$ are i.i.d.
- Contrary to Markov Chain Monte Carlo (MCMC) or sequential Monte Carlo, this approach leads to parallel and distributed simulations
- Sample generation requires tools from important sampling techniques
- Connections with the theory of random matrices[1]

[1] G. Calafiore, F. Dabbene, R. Tempo (2000)

IEIIT-CNR

Finite Families of Performance Functions

- Chernoff bounds and the Hoeffding inequality hold only for *fixed* performance function J
- Some results are available for a *finite* number of performance functions

Design Paradigm: Randomized Algorithms

Performance Function for Design

- Consider design parameters θ to be determined
- Study a *design* performance function

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

representing system constraints

- Replace

$$J(q) \leftrightarrow J(\theta, q)$$

$$\mathbf{P}_J \leftrightarrow \mathbf{P}_J(\theta)$$

Design Objective

- Recall that the objective for analysis is to satisfy the probabilistic constraint

$$\mathbf{P}_J = \mathrm{Prob}\{q \in Q \colon J(q) \leq \gamma\} \geq 1 - \varepsilon$$

- For design, the objective is to find θ such that the probability inequality

$$\mathbf{P}_J(\theta) = \mathrm{Prob}\{q \in Q \colon J(\theta, q) \leq \gamma\} \geq 1 - \varepsilon$$

  is satisfied

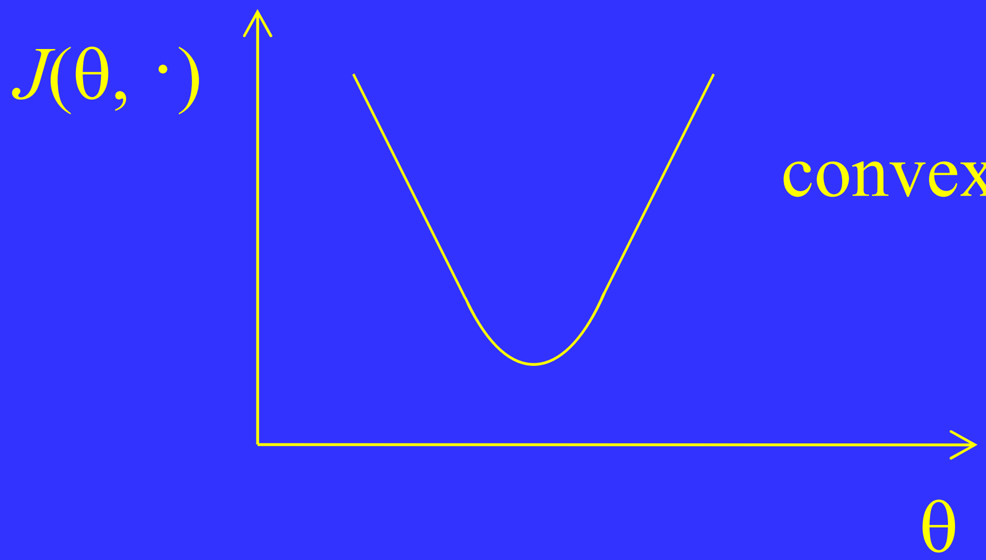
- We study randomized sequential algorithms for finding a *probabilistic feasible* solution θ

Randomized Sequential Algorithms

Convexity Assumption for Design Parameters

IEIIT-CNR

- **Convexity:** The function $J(\theta, q)$ is convex in θ for any fixed value of $q \in Q$



- The function $J(\theta, q)$ is measurable in q for any fixed value of θ

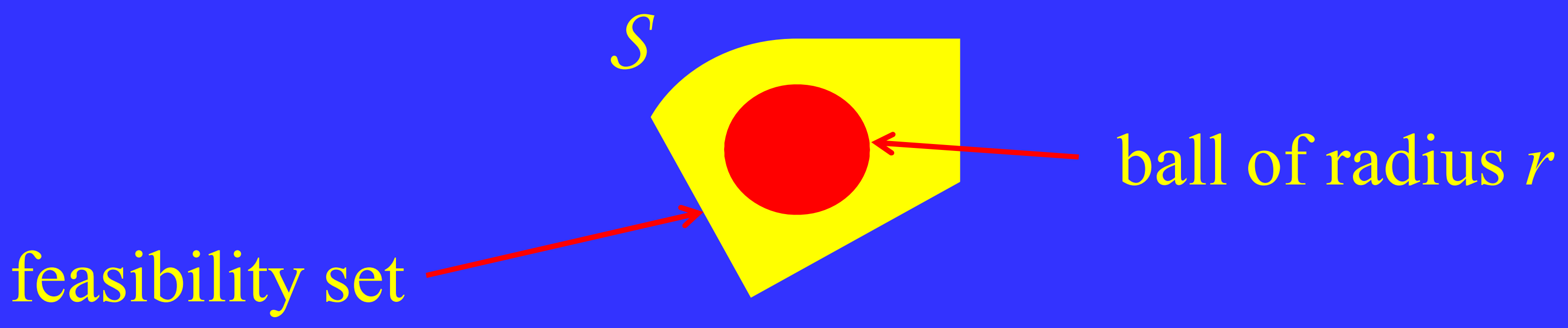
IEIIT-CNR

Definition of r-feasibility

- ***r*-feasibility**: For given $r > 0$, we say that $J(\theta, q) \le \gamma$ is r-feasible if the set

$$S = \{\theta : J(\theta, q) \le \gamma \text{ for all } q \in Q\}$$

contains a (full-dimensional) ball of radius r



IEIIT-CNR

Uncertainty Radius and Feasibility

- Consider the case when

$$Q = \{q : \|q\| \leq \rho\}$$

- If ρ is large the problem is "too hard" and no feasible solution θ exists
- Consider radius ρ as a parameter and perform analysis to check its largest value such that a solution exists
- Interplay between uncertainty radius ρ and feasibility radius r
- Problems which are unfeasible may be handled with the idea of discarded constraints

IEIIT-CNR

Sequential Methods for Design[1]

- Randomized sequential algorithms for finding a probabilistic feasible solution θ are based on two fundamental ingredients

  i) Oracle checking probabilistic feasibility of a candidate solution

  ii) Update rule exploiting convexity to construct a new candidate solution based on the oracle outcome

[1] G. Calafiore, F. Dabbene, R. Tempo (2010)

IEIIT-CNR

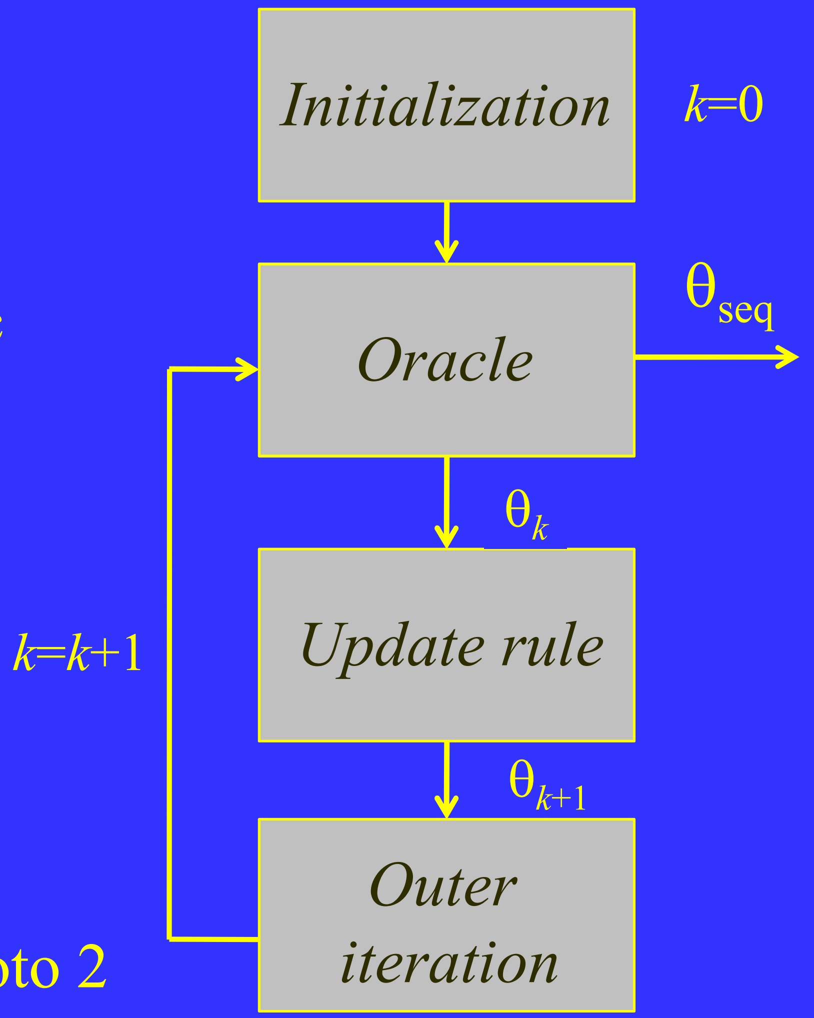
Meta-Algorithm

1. Initialization: set $k = 0$; choose θ_0

2. Oracle:
   - return *true* if θ_k is probabilistic feasible; Exit return $\theta_{seq} = \theta_k$
   - otherwise, return *false* and violation certificate

3. Update rule: Construct θ_{k+1} based on θ_k and on q_k

4. Outer iteration: Set $k=k+1$ and Goto 2



Probabilistic Oracle - 1

- Oracle is the randomized part of the algorithm and decides probabilistic feasibility of the current solution
- At step *k*, need to check if the *candidate solution* θ_k satisfies

$$\mathbf{P}_J(\theta_k) = \mathrm{Prob}\{q \in Q : J(\theta_k, q) \leq \gamma\} \geq 1 - \varepsilon$$

- To this end we perform a Monte Carlo simulation

IEIIT-CNR

Probabilistic Oracle - 2

- Generate N_k i.i.d. samples of q within Q (multisample)

$$q^{(1)}, \dots , q^{(N_k)} \in Q$$

- The candidate solution θ_k is *probabilistic feasible* if

$$J(\theta_k, q^{(i)}) \leq \gamma$$

for all $i = 1, \dots , N_k$

- Otherwise if $J(\theta_k, q^{(i)}) > 0$ we set $q_k = q^{(i)}$

IEIIT-CNR

Oracle (Inner) Iterations

- Consider the multisample size[1]

$$N_k \geq N_{\text{oracle}} = \left\lceil \frac{\log \frac{\pi^2 (k+1)^2}{6\delta}}{\log \frac{1}{1-\varepsilon}} \right\rceil$$

where ε , $\delta \in (0,1)$ are accuracy and confidence

- N_k is the number of Oracle (inner) iterations

- Slightly better bound has been obtained using the Riemann function

[1] Y. Oishi (2007)

IEIIT-CNR

Update Rule: Gradient Method

- Update rule is a classical gradient step

$$\theta_{k+1} = \begin{cases} \theta_k - \eta_k \dfrac{\partial_k(\theta_k)}{\left\|\partial_k(\theta_k)\right\|} & \text{if } \partial_k(\theta_k) \neq 0 \\ \theta_k & \text{otherwise} \end{cases}$$

- Let $\alpha > 0$, then the stepsize η_k is given by

$$\eta_k = \begin{cases} \dfrac{J(\theta_k, q_k)}{\left\|\partial_k(\theta_k)\right\|} + \alpha & \text{if } \partial_k(\theta_k) \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

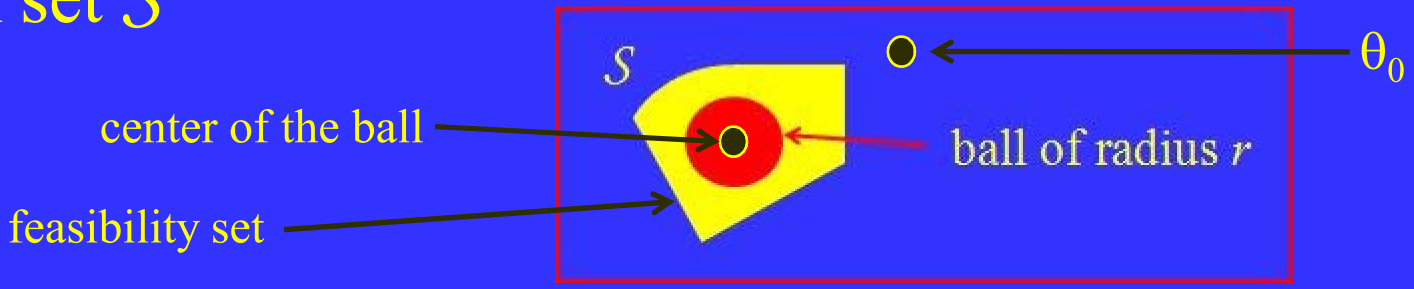
IEIIT-CNR

Outer Iterations

- Define

$$N_{\text{outer}} = \left\lceil \frac{D^2}{r^2} \right\rceil$$

where D is the distance between the initial solution θ_0 and the center of a ball of radius r contained in the solution set S



- r is imposed by the desired radius of feasibility
- If D is unknown, then we replace it with an upper bound which can be easily estimated

Successful/Unsuccessful Exit

- *Successful* exit: The algorithms returns a probabilistic controller θ_{seq}
- Establish probabilistic properties of θ_{seq}
- *Unsuccessful* exit: No solution has been found in N_{outer} iterations
- We have a *certificate of violation* q_k returned by the Oracle showing that the problem is not r-feasible

Probabilistic Properties of the Algorithm

❖ **Theorem[1]**

Let Convexity Assumption hold and let $\varepsilon, \delta \in (0,1)$

- The probability that the algorithm terminates at some outer iteration $k < N_{outer}$ returning θ_{seq} having large violation

$$\mathrm{Prob}\{q \in Q\colon J(\theta_{seq}, q) > \gamma\} > \varepsilon$$

is less than δ

- If the algorithm reaches the outer iteration N_{outer} then the problem is not r-feasible

[1] F. Dabbene , R. Tempo (2010)

Remarks

- Several key differences with stochastic approximation algorithms
- Explicit use of convexity
- Closed-form computation of the subgradient in many cases (e.g. LQ regulators, LMIs)
- Emphasis on finite termination criterion
- Unfeasible problems: Discard a few violated constraints (outliers) to make it feasible[1,2]

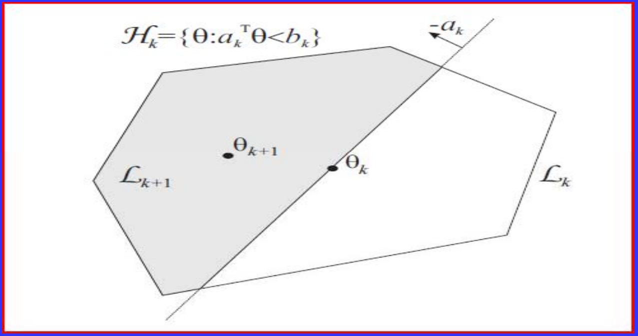
[1] E. W. Bai, H. Cho, R. Tempo, Y. Ye (2002), M. C. Campi, G. Calafiore and S. Garatti (2009)

IEIIT-CNR

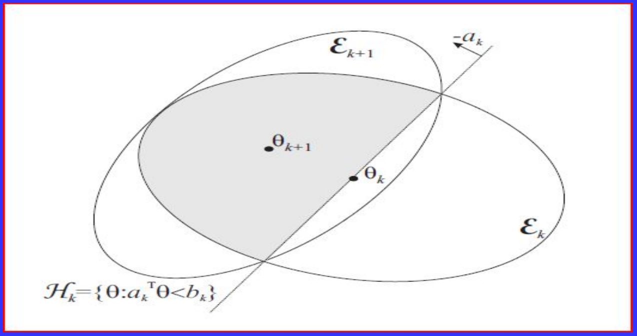
Advanced Techniques for Update Rule

- More advanced techniques falling in the class of localization methods can be used instead of gradient update

- In probabilistic *cutting plane* methods localization set is a polytope; update rule computes the analytic center



- In probabilistic *ellipsoid algorithm* localization set is an ellipsoid; update rule computes the center of the ellipsoid



Statistical Learning Theory

Statistical Learning Theory

- Statistical learning theory[1] is a branch of the theory of empirical processes
- Significant results and applications have been obtained in various areas, including neural networks, system identification, pattern recognition, control
- Main objective is to derive uniform convergence laws and the sample complexity
- No convexity assumption is required

[1] V. N. Vapnik, A. Ya. Chervonenkis (1981)

Uniform Convergence Laws

- Statistical learning theory studies the sample complexity such that the probability inequality

$$\left| \mathbf{P}_J(\theta) - \hat{\mathbf{P}}_J^N(\theta) \right| \leq \varepsilon$$

holds *uniformly* for all θ with probability at least 1- δ

- Recall that Monte Carlo simulation deals with *fixed* parameter θ or with *finite families* of parameters

Constrained Feedback Design with Uncertainty

- Consider design parameters $\theta \in \mathbf{R}^n$
- **Objective:** Minimize an objective function $c(\theta)$ subject to the performance constraints

$$J(\theta, q) \le \gamma$$

  for all $q \in Q$
- The problem is reformulated by means of a binary performance function

IEIIT-CNR

Binary Performance Function g

- Consider the (measurable) binary performance function

$$g\colon \mathbf{R}^n \times Q \to \{0,1\}$$

defined as

$$g(\theta, q) = \begin{cases} 0 & \text{if } J(\theta, q) \le \gamma \\ 1 & \text{otherwise} \end{cases}$$

Binary Probability of Violation

- Given $\theta \in \mathbf{R}^n$, the binary probability of violation for the function $g(\theta, q)$ is defined as

$$V_g(\theta) = \mathrm{Prob}\{q \in Q : g(\theta,q) = 1\}$$

Semi-Infinite Optimization Problem

- Find the optimal/suboptimal solution of the problem

$$\min_{\theta \in \mathbf{R}^n} \quad c(\theta) \quad \text{subject to } g(\theta, q) = 0 \text{ for all } q \in Q$$

where c: $\mathbf{R}^n \to \mathbf{R}$ is a measurable function

IEIIT-CNR

Randomized Non-Convex Optimization Problem

- Generate N i.i.d. samples (multisample) within Q

$$\mathrm{q}^{1,\ldots,N} = \{q^{(1)}, \ldots , q^{(N)}\}$$

according to a given probability measure

- Compute a (local) solution of the non-convex randomized optimization problem

$$\theta_{\mathrm{ncon}} = \arg \min_{\theta \in \mathbf{R}^n} c(\theta) \ \text{ subject to } \ g(\theta, q^{(i)}) = 0, \ \ i = 1, \ldots, N$$

IEIIT-CNR

Boolean Binary Function g

- The function g: $\mathbf{R}^n \times Q \to \{0,1\}$ is (α, m)-Boolean binary if for fixed q it can be written as a Boolean expression involving m polynomials

$$\beta_1(\theta, q), \ldots, \beta_m(\theta, q)$$

in the variables θ_i, $i=1,\ldots,n$ and the degree with respect to θ_i of all these polynomials is no larger than α

- **Example:** For fixed q take $m=1$ and

$$g = \beta_1(\theta) = 3 + 2\,\theta_1^2 - 5\,\theta_2^4\,\theta_3 + \ldots + 4\,\theta_1^2\,\theta_2\,\theta_4^7 \qquad \alpha = 7$$

IEIIT-CNR

Non-Convex Randomized Design

- **Theorem**[1]

  Let $g(\theta, q)$ be (α, m)-Boolean. Given $\varepsilon \in (0,0.14)$ and $\delta \in (0,1)$, if

$$N \geq N_{\mathrm{ncon}}(\varepsilon, \delta, n) = \left\lceil \frac{4.1}{\varepsilon}\left(\log\left(\frac{21.64}{\delta}\right) + 4.39 n \log_2\left(\frac{8e\alpha m}{\varepsilon}\right)\right)\right\rceil$$

where e is the Euler number, then the probability that

$$\mathbf{V}_g(\theta_{\mathrm{ncon}}) = \mathrm{Prob}\{q \in Q: g(\theta_{\mathrm{ncon}}, q) = 1\} > \varepsilon$$

is at most δ

[1] T. Alamo, R. Tempo, E. F. Camacho (2009)

IEIIT-CNR

Comments

- The function g is a Boolean expression consisting of polynomials; constraints and objective function are non-convex
- Sample complexity result holds for any suboptimal (local) solution
- We can use linearization algorithms to obtain a local solution (no need to compute a global solution)

Main References

- R. Tempo, G. Calafiore and F. Dabbene, "Randomized Algorithms for Analysis and Control of Uncertain Systems," *Springer-Verlag*, London, 2005 (second edition in preparation)
- F. Dabbene and R. Tempo, "Probabilistic and Randomized Tools for Control Design," The Control Handbook (W. S. Levine Ed.), *Taylor & Francis*, 2010
- G. Calafiore, F. Dabbene and R. Tempo "Research on Probabilistic Design Methods," Automatica, 2011 (to appear, preliminary version in my website)

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

RACT
Randomized Algorithms Control Toolbox

http://ract.sourceforge.net

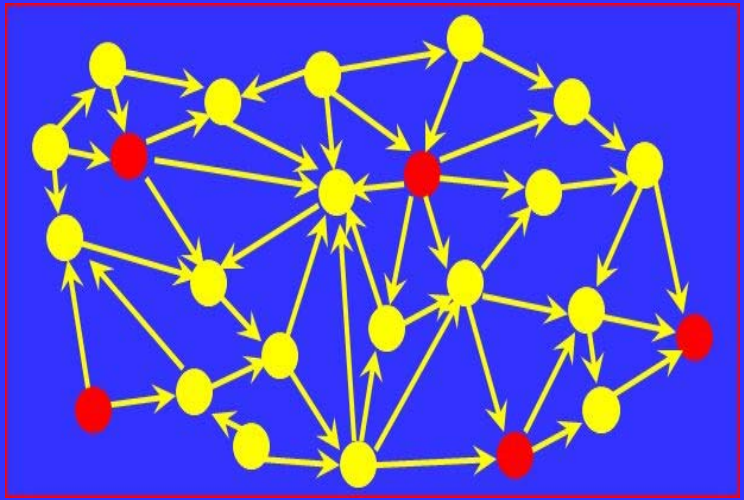
Conclusions

IEIIT-CNR

Randomized Algorithms for Systems and Control Applications

- *Aerospace control and unmanned aerial vehicles* (UAVs)[1]

- *PageRank Computation in Google, consensus and aggregation*[2]



[1] L. Lorefice, B. Pralio, R. Tempo (2009)
[2] H. Ishii, R. Tempo (2010)