Randomized Algorithms for Systems and Control: Theory and Applications

Roberto Tempo

IEIIT-CNR

Politecnico di Torino

roberto.tempo@polito.it

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **2008**	2. REPORT TYPE	3. DATES COVERED **00-00-2008 to 00-00-2008**

4. TITLE AND SUBTITLE **Randomized Algorithms for Systems and Control: Theory and Applications**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Politecnico di Torino,Corso Duca degli Abruzzi,24 - 10129 Torino Italy ,**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM002223. Presented at the NATO/RTO Systems Concepts and Integration Panel Lecture Series SCI-195 on Advanced Autonomous Formation Control and Trajectory Management Techniques for Multiple Micro UAV Applications held in Glasgow, United Kingdom on 19-21 May 2008.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**192**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

- Additional documents, papers, etc, please consult

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

- Questions may be sent to

  roberto.tempo@polito.it

IEIIT-CNR

References

- R. Tempo, G. Calafiore and F. Dabbene, “Randomized Algorithms for Analysis and Control of Uncertain Systems,” Springer-Verlag, London, 2005
- R. Tempo and H. Ishii, “Monte Carlo and Las Vegas Randomized Algorithms for Systems and Control: An Introduction,” EJC, Vol. 13, pp. 189-203, 2007
- RACT: Randomized Algorithms Control Toolbox for Matlab *http://ract.sourceforge.net*

Theory and Applications

- Theory of randomized algorithms for control
- UAV applications

Overview

- Preliminaries
- Probabilistic Robustness Analysis and Synthesis
- Sequential Methods for Convex Problems
- Non-Sequential Methods
- A Posteriori Analysis
- RACT
- Systems and Control Applications

Preliminaries

Randomized Algorithms (RAs)

- Randomized algorithms are frequently used in many areas of engineering, computer science, physics, finance, optimization,…but their appearance in systems and control is mostly limited to Monte Carlo simulations…

- Main objective of this NATO LS: Introduction to rigorous study of RAs for uncertain systems and control, with specific UAV applications

IEIIT-CNR

Randomized Algorithms (RAs)

- Computer science (RQS for sorting, data structuring)
- Robotics (motion and path planning problems)
- Mathematics of finance (path integrals)
- Bioinformatics (string matching problems)
- Distributed algorithms (PageRank in Google)
- Computer vision (computational geometry)

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, …

IEIIT-CNR

Limitations of Robust Control - 1

- Researchers realized some drawbacks of robust control
- Consider uncertainty Δ bounded in a set $\mathcal{B}$ of radius ρ. Largest value of ρ such that the system 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

Limitations of Robust Control - 2

- 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
  - static output feedback
  - structured singular value
  - stability of interval matrices

Different 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
- Objective: Breaking the curse of dimensionality[1]

[1] R. Bellman (1957)

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

Probabilistic Robustness Analysis

Uncertain Systems

$M(s)$ System

Δ Uncertainty

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

Worst Case Model

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

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

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

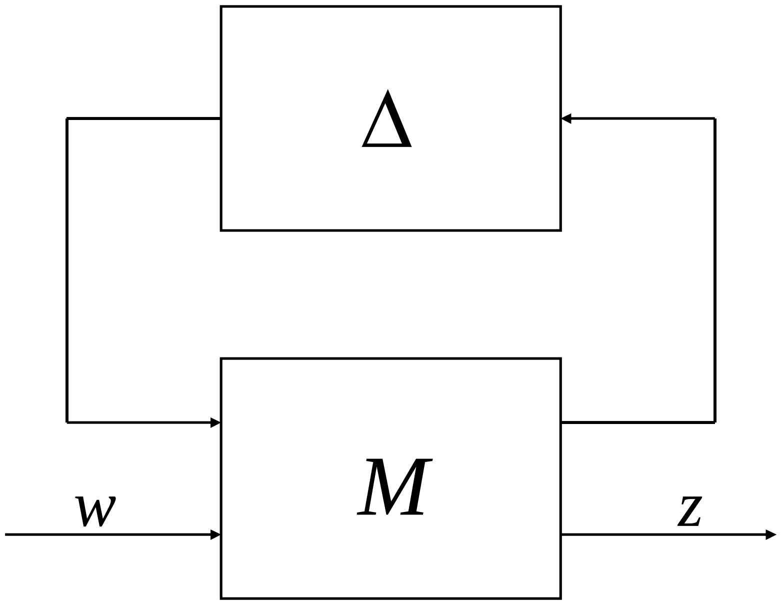
$$q_i \in [q_i^-, q_i^+]$$

- Nonparametric uncertainty

$$\{\Delta_{\mathrm{np}} \in \mathbf{R}^{n,n} : \|\Delta_{\mathrm{np}}\| \leq 1\}$$

IEIIT-CNR

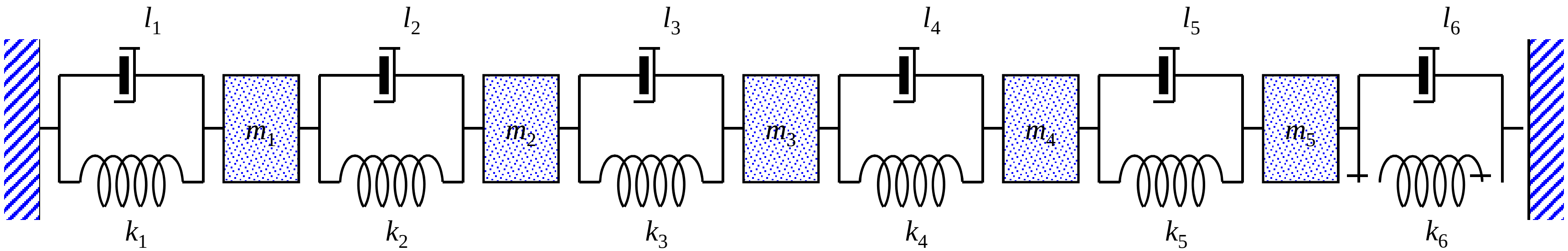
Robustness



- Uncertainty Δ is bounded in a structured set $\mathcal{B}$
- $z = F_u(M,\Delta)\, w$, where $F_u(M,\Delta)$ is the upper LFT

Example: Flexible Structure - 1

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



Flexible Structure - 2

- M-Δ configuration for controlled system and study robustness

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

$$\Delta = \begin{bmatrix} q_1 I_6 & 0 & 0 \\ 0 & q_2 I_6 & 0 \\ 0 & 0 & \Delta_{\mathrm{np}} \end{bmatrix}$$

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

$\Delta_{\mathrm{np}} \in \mathbf{C}^{4,4}$

$\mathcal{B} = \{\Delta : \sigma(\Delta) < 1\}$

Probabilistic Model

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

Performance Function

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

$$J = J(\Delta)$$

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

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

- Compute the $\mathcal{H}_\infty$ norm of the upper LFT $F_u(M,\Delta)$

$$J(\Delta) = \| F_u(M, \Delta)\|_\infty$$

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

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

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

Probability of Performance

- Given a performance level γ, we define the probability of performance

$$\text{Prob}\{J(\Delta) \leq \gamma\}$$

IEIIT-CNR

Measure of Violation and Reliability

- We define the measure of violation

$$V = 1 - \mathrm{Prob}\{J(\Delta) \leq \gamma\} = \mathrm{Prob}\{J(\Delta) > \gamma\}$$

- Probability of performance is also denoted as reliability

$$R = \mathrm{Prob}\{J(\Delta) \leq \gamma\} = 1 - V$$

Probabilistic Estimates

- Computing V and R requires to solve a difficult integration problem
- We use randomized algorithms to determine a probabilistic estimate of V and R

IEIIT-CNR

Randomized Algorithm: Definition

- Randomized Algorithm (RA): An algorithm that makes random choices during its execution to produce a result
- Example of a “random choice” is a coin toss

heads or *tails*

Randomized Algorithm: Definition

- Randomized Algorithm (RA): An algorithm that makes random choices during its execution to produce a result

- For hybrid systems, “random choices” could be switching between different states or logical operations
- For uncertain systems, “random choices” require (vector or matrix) random sample generation

Monte Carlo Randomized Algorithm

- Monte Carlo Randomized Algorithm: A randomized algorithm that may produce incorrect results, but with bounded error probability

IEIIT-CNR

Las Vegas Randomized Algorithm

- Las Vegas Randomized Algorithm: A randomized algorithm that always produces correct results, the only variation from one run to another is the running time

Monte Carlo Experiment

- We draw N i.i.d. random samples of Δ according to the given probability measure

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

- The multisample within $\mathcal{B}$ is

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

- We evaluate

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

IEIIT-CNR

Estimated Probability of Reliability

- We construct the estimated probability of reliability

$$\hat{R}_N = \frac{1}{N}\sum_{i=1}^{N} \mathbf{I}\left(J(\Delta^{(i)})\right)$$

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

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

Sample Complexity

- We need to compute the size of the Monte Carlo experiment (sample complexity)
- This requires to introduce probabilistic accuracy $\varepsilon \in (0,1)$ and confidence $\delta \in (0,1)$
- Given ε, $\delta \in (0,1)$, we want to determine N such that the probability event

$$\left| R - \hat{R}_N \right| \leq \varepsilon$$

holds with probability at least 1- δ

Chernoff Bound[1]

■ Chernoff Bound

Given ε, δ ∈(0,1), if

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

then the probability inequality

$$\left| R - \hat{R}_N \right| \leq \varepsilon$$

holds with probability at least 1- δ

[1] H. Chernoff (1952)

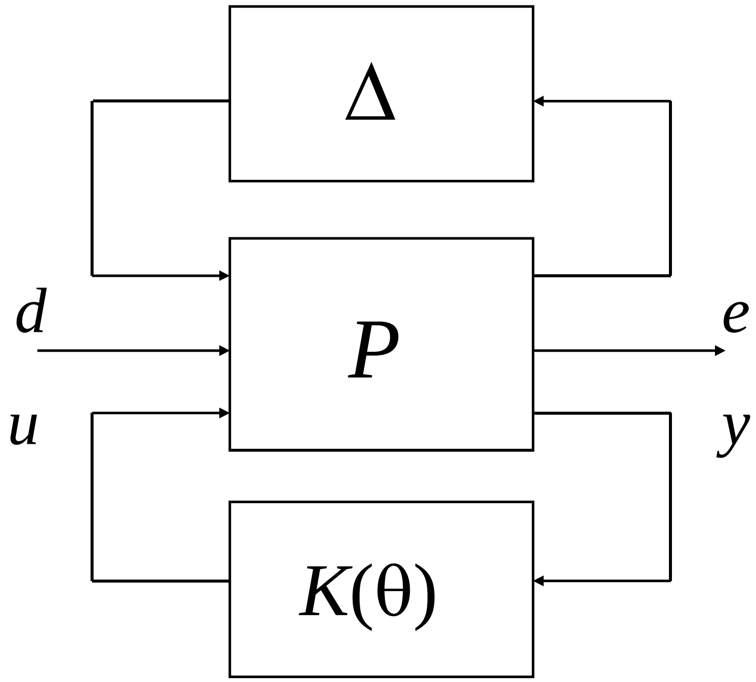
Remarks

- Chernoff bound improves upon other bounds such as the Law of Large Numbers (Bernoulli)
- Dependence is logarithmic on $1/\delta$ and quadratic on $1/\varepsilon$
- Sample size is independent on the number of controller and uncertain parameters

ε	0.1%	0.1%	0.5%	0.5%
$1-\delta$	99.9%	99.5%	99.9%	99.5%
N	$3.9 \cdot 10^6$	$3.0 \cdot 10^6$	$1.6 \cdot 10^6$	$1.2 \cdot 10^5$

Probabilistic Robust Synthesis

Synthesis Paradigm



- Design the parameterized controller $K(\theta)$ to guarantee stability and performance

Synthesis Performance Function

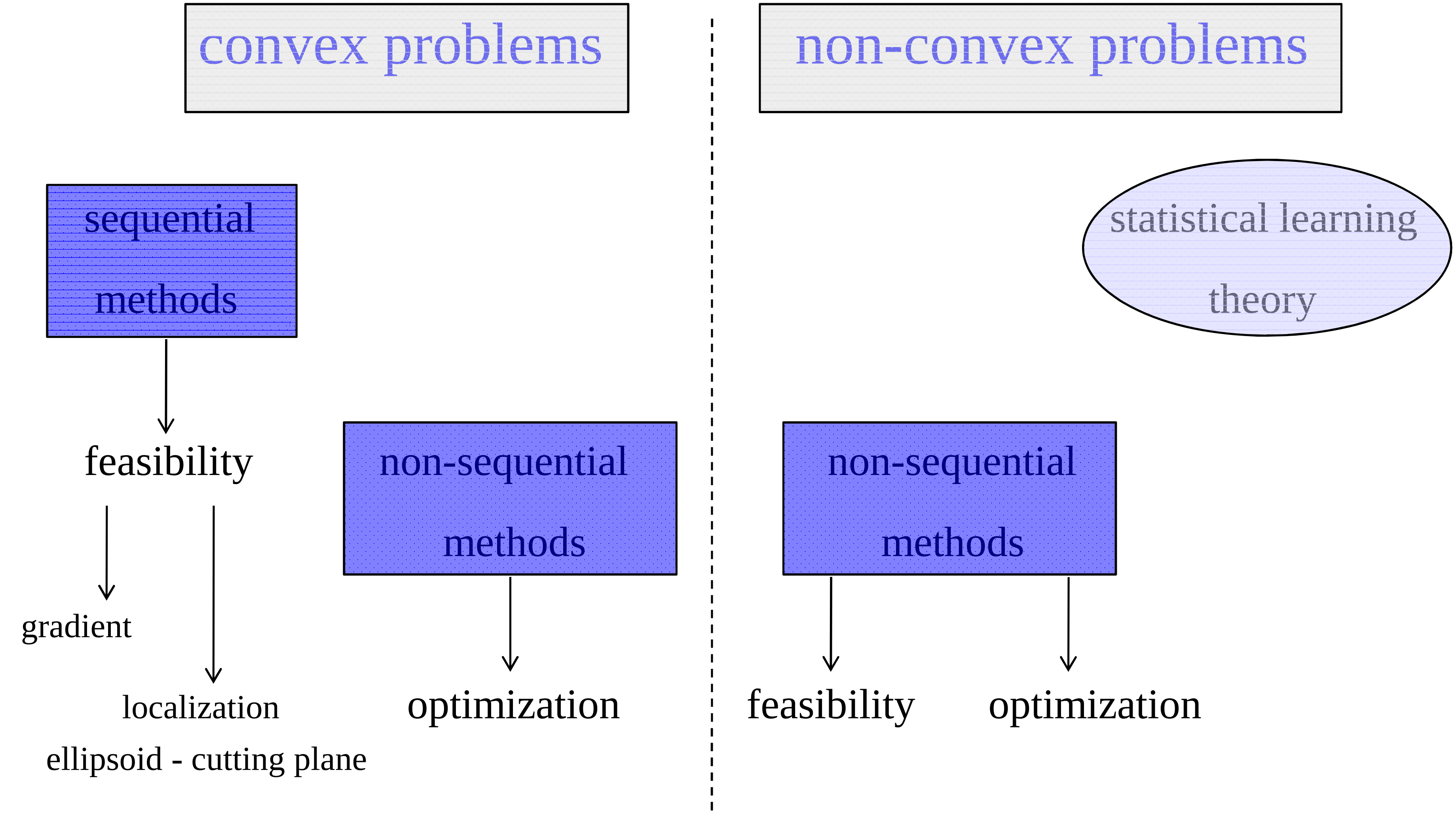
- Parameterized controller $K(\theta)$
- We replace $J(\Delta)$ with a synthesis performance function representing system constraints

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

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

IEIIT-CNR

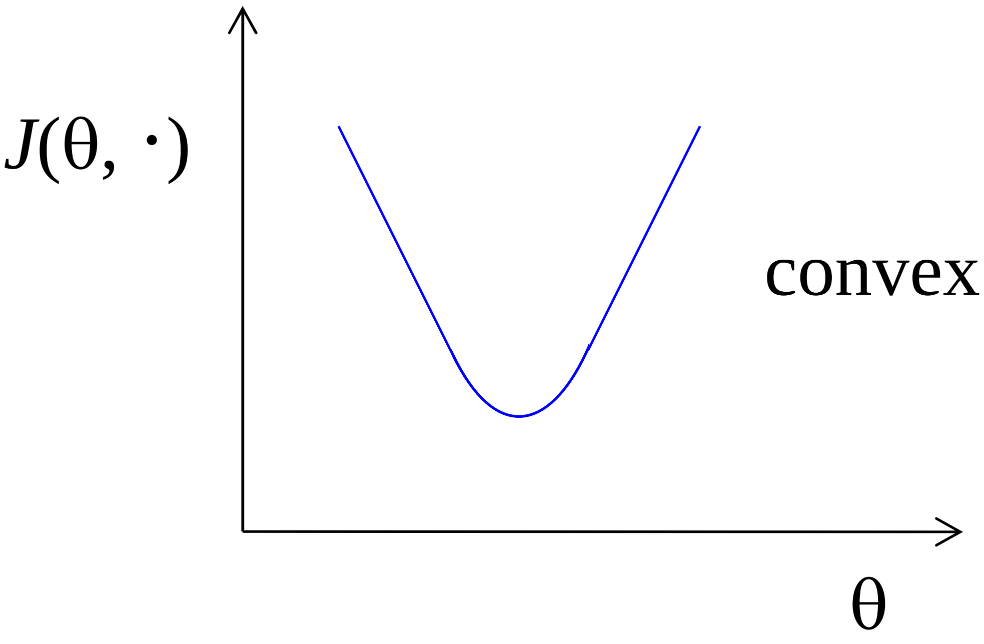
Probabilistic Design Methods: The Big Picture



Quadratic Performance and Convexity

Convexity Assumption

- Convexity Assumption: The function $J(\theta, \Delta)$ is convex in θ for any fixed value of $\Delta \in \mathcal{B}$



IEIIT-CNR

Convex Functions and LQ Regulators

- Examples of convex functions arise when considering various control problems, such as design of LQ regulators
- This is illustrated by means of an application example for control of lateral motion of an aircraft

Example: Control of Lateral Motion of Aircraft[1]

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

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

where A and B are given by

[1] R. Tempo, G. Calafiore and F. Dabbene (2005)

IEIIT-CNR

State Space Matrices

$$A=\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & L_p & L_\beta & L_r \\ g/V & 0 & Y_\beta & -1 \\ N_{\dot\beta}\left(g/V\right) & N_p & N_\beta + N_{\dot\beta} Y_\beta & N_r - N_{\dot\beta} \end{bmatrix}$$

$$B=\begin{bmatrix} 0 & 0 \\ 0 & L_{\delta a} \\ Y_{\delta r} & 0 \\ N_{\delta r} + N_{\dot\beta} Y_{\delta r} & N_{\delta a} \end{bmatrix}$$

IEIIT-CNR

State Variables and Control Inputs

- State variables
  - x_1 bank angle
  - x_2 derivative of bank angle
  - x_3 sideslip angle
  - x_4 jaw rate
- Control inputs
  - u_1 rudder deflection
  - u_2 aileron deflection

Uncertain Parameters

- Each parameter value is perturbed by a relative uncertainty equal to 10% around its nominal value $\bar{\Delta}_i$
- The uncertainty vector (parametric uncertainty)

$$\Delta = [\Delta_1, \Delta_2, \ldots, \Delta_{13}]^T$$

varies in an hyperrectangle centered at the nominal value

$$\mathcal{B} = \{\Delta : \Delta_i \in [0.90\bar{\Delta}_i, 1.10\,\bar{\Delta}_i],\ i=1,\ldots,13\}$$

- We have uncertain matrices $A(\Delta)$ and $B(\Delta)$

IEIIT-CNR

Parameter Nominal Values

L_p=-2.93	L_β=-4.75	L_r=0.78	g/V=0.086	Y_β=-0.11
$N_{\dot\beta}$=0.1	N_p=-0.042	N_β=2.601	N_r=-0.29	$L_{\delta a}$=-3.91
$Y_{\delta r}$=0.035	$N_{\delta r}$=-2.5335	$N_{\delta a}$=0.31		

IEIIT-CNR

Quadratic Performance Function

- We design a state feedback controller $u = Kx$ that robustly stabilizes the system guaranteeing a decay rate $\alpha > 0$
- Define the quadratic performance function

$$\Phi_{QP}(P, W, \Delta) = A(\Delta)P + PA^T(\Delta) + B(\Delta)W^T + WB^T(\Delta) + 2\alpha P$$

where $P = P^T > 0$ and W are matrices of suitable dimensions

Sufficient Condition

- A sufficient condition for the existence of a controller K is to find $P=P^T > 0$ and W such that

$$\Phi_{QP}(P, W, \Delta) \leq 0$$

is satisfied for all $\Delta \in \mathcal{B}$

- Equivalently we find (common) solutions $P=P^T > 0$ and W of the quadratic cost function

$$\Phi_{QP}(P, W, \Delta) \leq 0$$

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

Control Gain

- A control gain which robustly guarantees the decay rate α for all $\Delta \in \mathcal{B}$ is given by

$$K = W^{\mathrm{T}} P^{-1}$$

- This problem can be reformulated in terms of linear matrix inequalities (LMIs)
- The controller is parameterized as $K=K(\theta)$, where

$$\theta = \{P, W\}$$

IEIIT-CNR

Linear Matrix Inequalities (LMIs)

$\mathcal{B}$

- This quadratic constrained problem can be written in the general setting of LMIs
- Find θ such that

$$F(\theta, \Delta) \leq 0$$

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

$$F(\theta, \Delta) = F_0(\Delta) + \theta_1 F_1(\Delta) + \ldots + \theta_n F_n(\Delta)$$

and $F_i(\Delta)$ are real symmetric matrices depending (nonlinearly) on Δ

Performance Function

- To rewrite an LMI in terms of a performance function $J(\theta, \Delta)$ we set

$$J(\theta, \Delta) = \lambda_{\max} F(\theta, \Delta)$$

where $\lambda_{\max}(\cdot)$ is the maximum eigenvalue of $(\cdot)$

Multiobjective Design Problems

- To consider scalar-valued constraints is without loss of generality
- Multiobjective design problems can be easily handled
- Multiple constraints of the form

$$J_1(\theta, \Delta) \leq 0, \ldots, \; J_n(\theta, \Delta) \leq 0$$

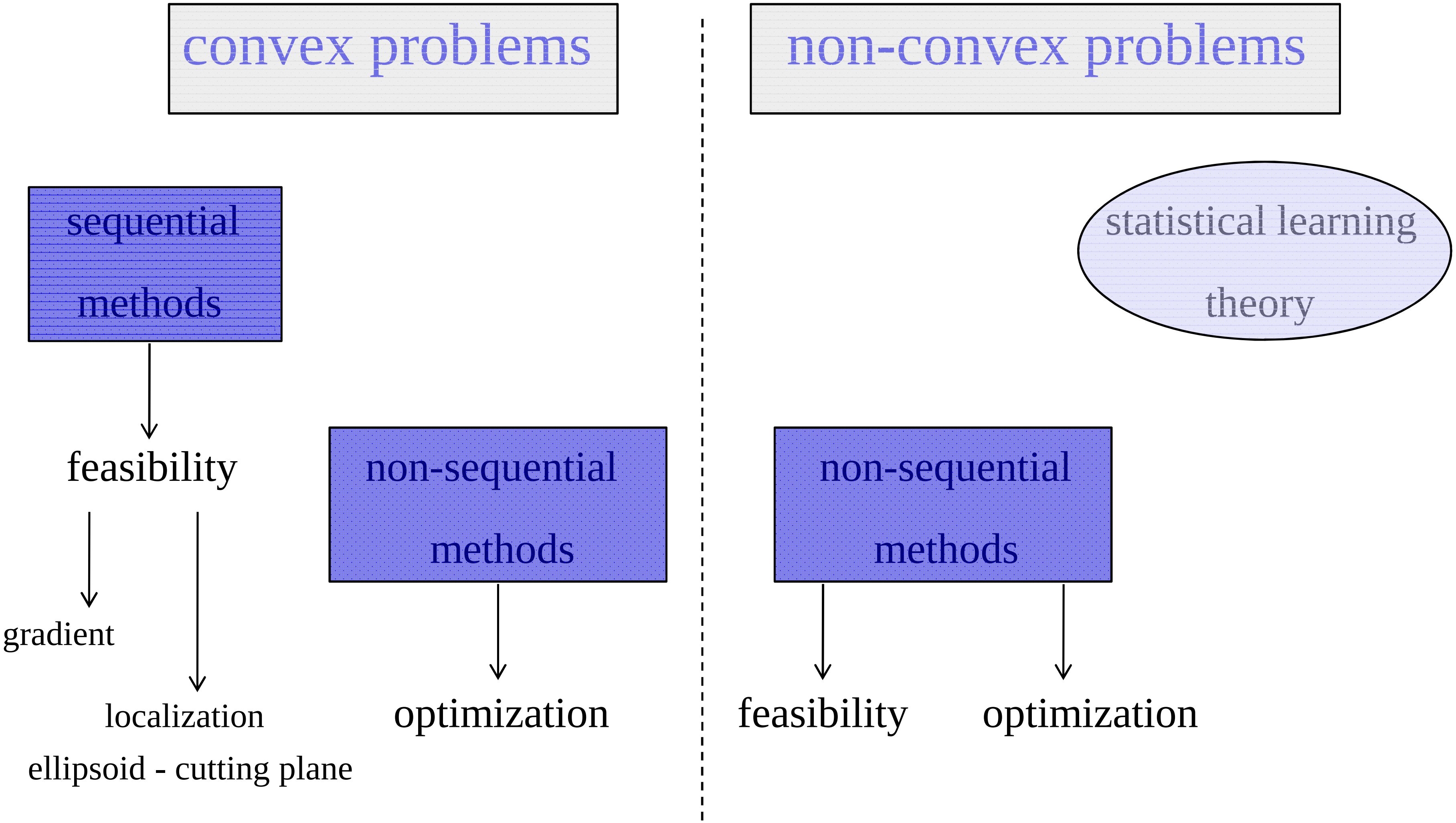
can be reduced to a single scalar-valued constraint setting

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

Sequential Methods for Convex Problems

IEIIT-CNR

Probabilistic Design Methods: The Big Picture



Sequential Methods for Design

- We study randomized sequential methods for finding a probabilistic feasible solution θ
- That is we determine θ satisfying the uncertain inequality

$$J(\theta, \Delta) \leq 0$$

with some probability

Definition of r-feasibility

- r-feasibility: For given $r>0$, we say that $J(\theta, \Delta) \leq 0$ is r-feasible if the solution set

$$\mathcal{S} = \{\theta : J(\theta, \Delta) \leq 0 \text{ for all } \Delta \in \mathcal{B}\}$$

contains a (full-dimensional) ball of radius r

Performance Function

- Let Δ be a random vector distributed according to a probability measure
- Given probabilistic accuracy $\varepsilon \in (0,1)$, we search for $P=P^{\mathrm{T}} > 0$ and W such that

$$\mathrm{Prob}\{\Delta \in \mathcal{B}:\ \Phi_{QP}(P, W, \Delta) \leq 0\} > 1 - \varepsilon$$

- Defining the performance function

$$J(P, W, \Delta) = \lambda_{\max} \Phi_{QP}(P, W, \Delta)$$

the problem is to find $P=P^{\mathrm{T}} > 0$ and W such that

$$\mathrm{Prob}\{\Delta \in \mathcal{B}:\ J(P, W, \Delta) \leq 0\} > 1 - \varepsilon$$

Probability of Violation

- The probability of violation of the controller θ is

$$V(\theta) = \text{Prob}\{\Delta \in \mathcal{B}:\ J(\theta, \Delta) > 0\}$$

- We want to find θ such that the probability of violation is small

$$V(\theta) < \varepsilon$$

- If such θ exists in the feasible set $\mathcal{S}$, then we have a probabilistic feasible solution (probabilistic robust design)

Controller Reliability

- Given accuracy $\varepsilon \in (0,1)$, probabilistic robust design requires finding controller parameters θ such that the controller reliability

$$R(\theta) = 1 - V(\theta)$$

is at least $1 - \varepsilon$

Sequential Methods for Design

- 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

Meta-Algorithm

1. Initialization: set $k = 0$ and choose an initial solution θ_0

2. Oracle: Oracle returns *true* if θ_k is a probabilistic feasible controller and Exit returning $\theta_{seq} = \theta_k$

   Otherwise, the Oracle returns *false* and a violation certificate

3. Update Rule: Construct θ_{k+1} based on θ_k and on Δ_k

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

Probabilistic Oracle

- Oracle is the randomized part of the algorithm and decides probabilistic feasibility of the current solution
- We generate N_k i.i.d. samples of Δ within $\mathcal{B}$ (multisample)

$$\Delta^{(1)}, \ldots , \Delta^{(N_k)} \in \mathcal{B}$$

- The candidate solution θ_k is probabilistic feasible if

$$J(\theta_k, \Delta^{(i)}) \leq 0$$

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

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

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

[1] Y. Oishi (2007)

Algorithm Oracle

- Input: θ_k, N_k
- Output: feasibility (true/false), violation certificate Δ_k

- for i =1, ... , N_k , draw a sample $\Delta^{(i)}$

- Randomized test

- if $J(\theta_k, \Delta^{(i)}) > 0$, set $\Delta_k = \Delta^{(i)}$, feasibility = false
- exit and return Δ_k
- end if
- end for

Update Rule: Gradient Method

- We assume that the subgradient $\partial_k(\theta)$ of $J(\theta,\Delta)$ is computable at Δ_k
- If $J(\theta, \Delta_k)$ is differentiable at θ, then $\partial_k(\theta)$ is the gradient of $J(\theta, \Delta)$

Gradient Step and Stepsize

- 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 $r > 0$, then the stepsize η_k is given by

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

Algorithm Update Rule (Gradient)

IEIIT-CNR

- Input: θ_k, Δ_k
- Output: θ_{k+1}

- compute the subgradient $\partial_k(\theta)$ of $J(\theta, \Delta_k)$
- compute the stepsize $\eta_k = \begin{cases} \dfrac{J(\theta_k, \Delta_k)}{\|\partial_k(\theta_k)\|} + r & \text{if } \partial_k(\theta_k) \neq 0 \\ 0 & \text{otherwise} \end{cases}$
- update

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

Outer Iterations

- Define

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

where R is the distance between the initial solution θ_0 and the center of a ball of radius r contained in the solution set $\mathcal{S}$

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

Algorithm Sequential Design

- Input: $\varepsilon, \delta \in (0,1)$, N_{outer}
- Output: θ_{seq}
- choose θ_0, set $k=0$ and feasibility=false
- Outer iteration
- while feasibility = false and $k < N_{\text{outer}}$
- determine multisample size N_k
- invoke Oracle obtaining feasibility (true/false) and Δ_k
- if feasibility = false then compute θ_{k+1} using Update Rule
- else set $\theta_{\text{seq}} = \theta_k$
- set $k = k + 1$
- end while

Probabilistic Properties of Sequential Design

- Theorem[1]

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

- If Algorithm Sequential Design terminates at some outer iteration $k < N_{outer}$ returning θ_{seq}, then the probability that $V(\theta_{seq}) > \varepsilon$ is at most δ

- If Algorithm Sequential Design reaches the outer iteration N_{outer}, then the problem is not r-feasible

[1] F. Dabbene and R. Tempo (2008)

Remark: Successful/Unsuccessful Exit

- The first situation corresponds to a successful exit: The algorithms returns a probabilistic controller θ_{seq}
- The second situation corresponds to an unsuccessful exit: No solution has been found in N_{outer} iterations
- In this case we have a certificate of violation Δ_k returned by the Oracle showing that the problem is not r-feasible

IEIIT-CNR

Aircraft Example Revisited: Sequential Methods

- Setting α =0.5, we look for a probabilistic solution to the uncertain LMI

$$P=P^{T}>0 \qquad \Phi_{QP}(P, W, \Delta) \leq 0$$

where the quadratic performance function is given by

$$\Phi_{QP}(P, W, \Delta) = A(\Delta)P + PA^{T}(\Delta) + B(\Delta)W^{T} + WB^{T}(\Delta)+2\alpha P$$

- Letting ε =0.01 and $\delta = 10^{-6}$, the sequential algorithm is guaranteed to return (with 99.9999% probability) a solution P, W such that quadratic performance holds with 99% probability

Numerical Results

- Algorithm terminated after k = 28 (outer) iterations
- Quadratic performance was checked by the Oracle for

$$N_k = 2{,}029$$

uncertainty samples

- We obtained …

P_{seq} and W_{seq}

$$P_{seq} = \begin{bmatrix} 0.3075 & -0.3164 & -0.0973 & -0.0188 \\ -0.3164 & 0.5822 & -0.0703 & -0.0993 \\ -0.0973 & -0.0703 & 0.2277 & 0.2661 \\ -0.0188 & -0.0993 & 0.2661 & 0.7100 \end{bmatrix}$$

$$W_{seq} = \begin{bmatrix} -0.0191 & 0.2733 \\ -0.0920 & 0.4325 \\ 0.0803 & -0.3821 \\ 0.4496 & -0.2032 \end{bmatrix}$$

IEIIT-CNR

Probabilistic Controller K_{seq}

- Probabilistic controller $K = W^{T}P^{-1}$ is given by

$$K_{seq} = \begin{bmatrix} -2.9781 & -1.9139 & -3.2831 & 1.5169 \\ 7.3922 & 5.1010 & 4.1401 & -0.9284 \end{bmatrix}$$

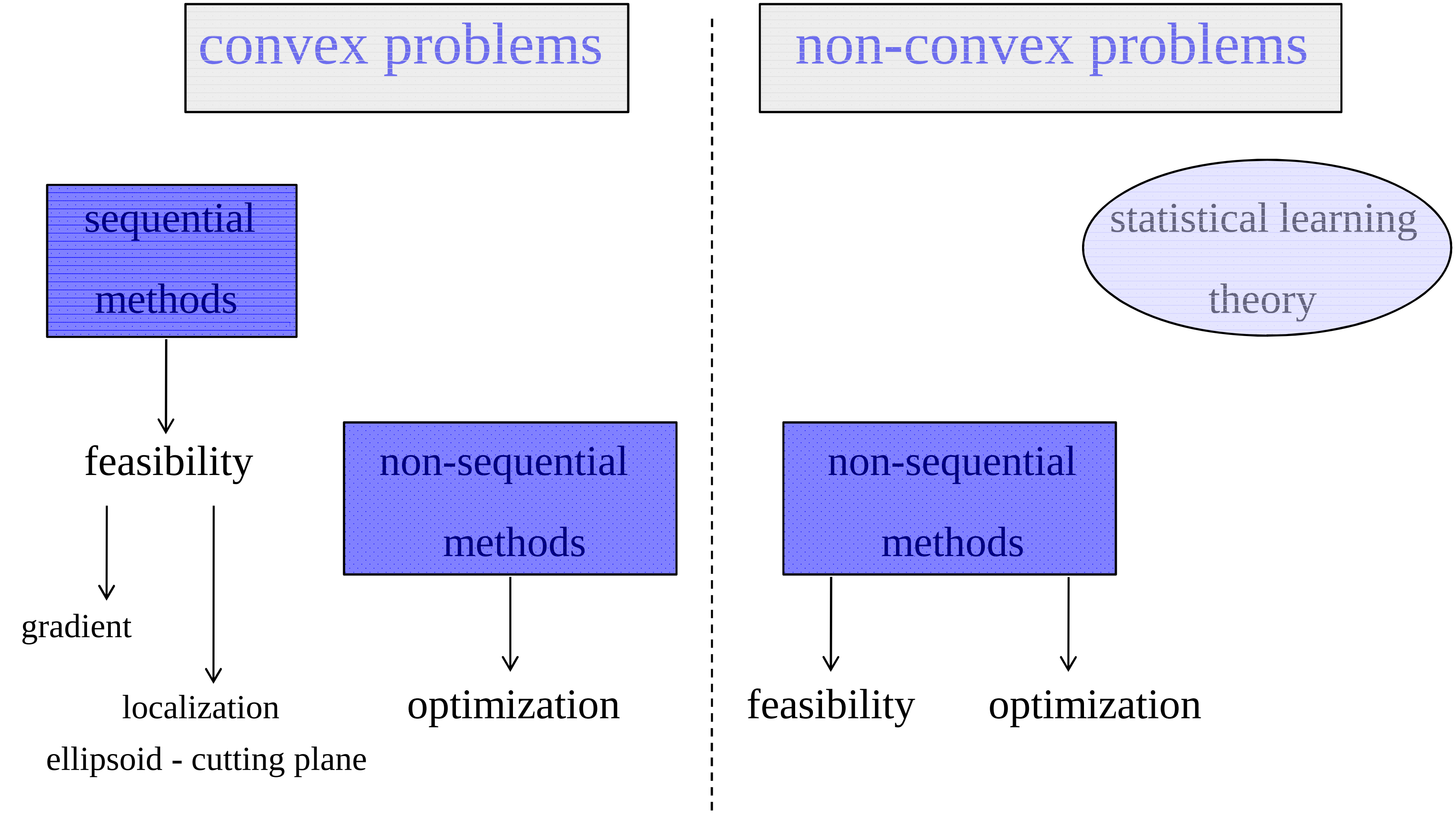
- With an a-posteriori analysis we will check if K_{seq} is a robust controller and its probabilistic properties

IEIIT-CNR

Non-Sequential Methods for Convex Problems

IEIIT-CNR

Probabilistic Design Methods: The Big Picture



Convexity Assumption

- Convexity Assumption: The function $J(\theta, \Delta)$ is convex in θ for any fixed value of $\Delta \in \mathcal{B}$

Scenario Approach

- Non-sequential method which provides a one-shot solution for general uncertain convex problems
- Randomization of $\Delta \in \mathcal{B}$ and solution of a single convex optimization problem
- Derivation of a formula involving sample size, number of controller parameters, probabilistic accuracy and confidence
- Explicit computation of the sample complexity

IEIIT-CNR

Convex Semi-Infinite Optimization

- Semi-infinite optimization problem

$$\min_{\theta \in \mathbf{R}^n} c^T \theta \quad \text{subject to } J(\theta, \Delta) \leq 0 \quad \text{for all } \Delta \in \mathcal{B}$$

where $J(\theta, \Delta) \leq 0$ is convex in θ for all $\Delta \in \mathcal{B}$ and n is the number of design parameters

Scenario Problem

- We construct a scenario problem using randomization
- Taking i.i.d. random samples $\Delta^{(i)}$, $i = 1, \ldots, N$, we construct the sampled constraints

$$J(\theta, \Delta^{(i)}) \leq 0, \quad i = 1, \ldots, N$$

and form the scenario optimization problem (convex problem)

$$\theta_{\text{scen}} = \arg\min_{\theta \in \mathbf{R}^n} c^T \theta \quad \text{subject to} \quad J(\theta, \Delta^{(i)}) \leq 0, \quad i = 1, \ldots, N$$

Convex Scenario Design

- Theorem[1]

Let Convexity Assumption hold. Suppose that $N \geq n$ and $\varepsilon, \delta \in (0,1)$ satisfy the inequality

$$\binom{N}{n}(1-\varepsilon)^{N-n} \leq \delta$$

then, the probability that

$$V(\theta_{scen}) = \text{Prob}\{\Delta \in \mathcal{B}: \ J(\theta_{scen}, \Delta) > 0\} > \varepsilon$$

is at most δ

[1] G. Calafiore and M. Campi (2005)

Remarks

- We have considered the case when the scenario problem admits a feasible solution and this solution is unique
- Clearly, if the scenario problem is unfeasible, then also the original semi-infinite convex problem is unfeasible
- The assumption on uniqueness of the solution can be relaxed in most practical cases

Sample Complexity

- Computing the minimum value of N such that

$$\binom{N}{n}(1-\varepsilon)^{N-n} \leq \delta$$

holds is immediate (given ε, δ and n, is a one-parameter problem)

- A different issue is to derive the sample complexity which is an *explicit* relation of the form

$$N = N(\varepsilon, \delta, n)$$

IEIIT-CNR

Sample Complexity of the Scenario Problem

- Sample complexity can be computed for the scenario problem
- In[1] it has been proven that the relation

$$\binom{N}{n}(1-\varepsilon)^{N-n} \leq \delta$$

holds if

$$N \geq N_{scen}(\varepsilon,\delta,n) = \left\lceil \frac{2}{\varepsilon}\log\left(\frac{1}{2\delta}\right) + 2n + \frac{2}{\varepsilon}\log(4) \right\rceil$$

[1] T. Alamo, R. Tempo and E.F. Camacho (2007)

Algorithm Scenario Design

- Input: ε, δ, n
- Output: θ_{scen}

- compute the sample size $N_{\text{scen}}(\varepsilon, \delta, n)$
- draw $N \geq N_{\text{scen}}(\varepsilon, \delta, n)$ i.i.d. samples $\Delta^{(i)}$
- solve the convex optimization problem

$$\theta_{\text{scen}} = \arg \min_{\theta \in \mathbf{R}^n} c^T \theta \quad \text{subject to} \quad J(\theta, \Delta^{(i)}) \leq 0, \; i = 1, \ldots, N$$

Aircraft Example Revisited: Scenario Design

- The objective is to determine a probabilistic solution to the optimization problem

$$\min_{P,\,W} \quad \mathrm{Tr}\, P \quad \text{subject to} \quad P=P^{\mathrm{T}} > 0,\ \Phi_{\mathrm{QP}}(P, W, \Delta) \leq 0$$

where $\mathrm{Tr}(\cdot)$ denotes the trace of $(\cdot)$

- Setting ε =0.01 and $\delta=10^{-6}$, we compute the sample complexity for n=18 obtaining

$$N_{\mathrm{scen}} = 7{,}652$$

- Hence we need to solve a convex optimization problem with 7,652 constraints and 18 design variables

P_{scen} and W_{scen}

$$P_{scen} = \begin{bmatrix} 0.1445 & -0.0728 & 0.0035 & 0.0085 \\ -0.0728 & 0.2192 & -0.0078 & -0.0174 \\ 0.0035 & -0.0078 & 0.1375 & 0.0604 \\ 0.0085 & -0.0174 & 0.0604 & 0.1975 \end{bmatrix}$$

$$W_{scen} = \begin{bmatrix} 0.0109 & 0.0908 \\ 7.2929 & 3.4846 \\ 0.0439 & -0.0565 \\ 0.6087 & -3.9182 \end{bmatrix}$$

Probabilistic Controller K_{scen}

- Probabilistic controller $K = W^{T} P^{-1}$ is equal to

$$K_{\text{scen}} = \begin{bmatrix} 20.0816 & 40.3852 & -0.4946 & 5.9234 \\ 10.7941 & 18.1058 & 9.8937 & -21.7363 \end{bmatrix}$$

Non-Sequential Methods for Non-Convex Problems

IEIIT-CNR

Probabilistic Design Methods: The Big Picture

convex problems

sequential methods

feasibility

gradient

localization

ellipsoid - cutting plane

non-sequential methods

optimization

non-convex problems

statistical learning theory

non-sequential methods

feasibility

optimization

IEIIT-CNR

Statistical Learning Theory for Control Design of Uncertain Systems

- Statistical learning theory is a branch of the theory of empirical processes
- Significant results have been obtained in various areas, including neural networks, system identification, pattern recognition, …
- We study statistical learning theory for control design of uncertain systems

Statistical Learning Theory

- Main objective is to derive uniform convergence laws (for all controller parameters) and the sample complexity
- This leads to a powerful methodology for control synthesis (feasibility and optimization) which is not based upon a convexity assumption on the controller parameters
- The sample complexity is significantly larger than that derived in the convex case

Controller Reliability

- Recall that the reliability for the controller $K(\theta)$ is

$$R(\theta) = \mathrm{Prob}\{\Delta \in \mathcal{B}:\ J(\theta, \Delta) \leq 0\} = 1 - V(\theta)$$

- Computing $R(\theta)$ requires to solve a difficult integration problem
- For fixed θ we compute a probabilistic estimate of reliability setting a simple Monte Carlo experiment

Monte Carlo Experiment

- We take N i.i.d. random samples of Δ according to the given probability measure

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

- We evaluate

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

IEIIT-CNR

Estimated Probability of Reliability

- Given controller parameters θ, we construct a probabilistic estimated of reliability

$$\hat{R}_N(\theta) = \frac{1}{N}\sum_{\mathrm{i}=1}^{N} \mathbf{I}\left(J(\theta, \Delta^{(i)})\right)$$

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

$$\mathbf{I}\left(\mathrm{J}(\theta, \Delta^{(\mathrm{i})})\right) = \begin{cases} 1 & \text{if } J(\theta, \Delta^{(\mathrm{i})}) \le 0 \\ 0 & \text{otherwise} \end{cases}$$

Law of Large Numbers

- Monte Carlo analysis (Law of Large Numbers) studies the sample complexity such that for *fixed* θ the probability inequality

$$\left| R(\theta) - \hat{R}_N(\theta) \right| \leq \varepsilon$$

holds with probability at least 1- δ

Uniform Convergence Law

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

$$\left| R(\theta) - \hat{R}_N(\theta) \right| \leq \varepsilon$$

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

Optimization of Non-Convex Problems

Constrained Feedback Design with Uncertainty

- The objective is to minimize an objective function $c(\theta)$ subject to the performance constraint

$$J(\theta, \Delta) \leq 0$$

- The problem is formulated in terms of a binary performance function

Binary Performance Function g

- We introduce the performance function g

  $g: \Theta \times \mathcal{B} \rightarrow \{0,1\}$

  which is a binary measurable function defined as

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

Binary Probability of Violation

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

$$V_g(\theta) = \mathrm{Prob}\{\Delta \in \mathcal{B}\colon g(\theta, \Delta) = 1\}$$

Binary Optimization Problem

- Semi-Infinite Optimization Problem: Find the optimal solution of the problem

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

where $c\colon \Theta \rightarrow \mathbf{R}$ is a measurable function

Randomized Non-Convex Optimization Problem

- Generate N i.i.d. samples (multisample) within $\mathcal{B}$

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

according to a given probability measure

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

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

Boolean Binary Function g

- The function g: $\mathbf{R}^n \times \mathcal{B} \rightarrow \{0,1\}$ is (γ, m)-Boolean binary if for fixed Δ can be written as a Boolean expression consisting of m polynomials in the variables θ_i, $i=1,\dots,n$

$$\beta_1(\theta, \Delta), \dots, \beta_m(\theta, \Delta)$$

and the degree with respect to θ_i of all these polynomials is no larger than γ

- Example: For fixed Δ take $m=1$ and

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

Non-Convex Learning Based Design

- Theorem[1]

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

$$N \geq N_{\mathrm{ncon}}(\varepsilon,\delta,n) = \left\lceil \frac{1}{\varepsilon}\left(4.1\log\left(\frac{21.64}{\delta}\right) + 36n\log_2 \max\left\{\frac{2}{\varepsilon}, 4e\gamma m\right\}\right)\right\rceil$$

where e is the Euler number, then the probability that

$$V_g(\theta_{\mathrm{ncon}}) = \mathrm{Prob}\{\Delta \in \mathcal{B}: g(\theta_{\mathrm{ncon}}, \Delta) = 1\} > \varepsilon$$

is at most δ

[1] T. Alamo, R. Tempo and E.F. Camacho (2007)

Comments - 1

- 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)
- The approach consists of uncertainty randomization and deterministic optimization in controller space
- We avoid randomization of controller parameters

Empirical Mean of Violation

- Given N i.i.d. samples within $\mathcal{B}$

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

the empirical mean of violation is equal to

$$\hat{V}_g(\theta) = \frac{1}{N}\sum_{i=1}^{N} g\left(\theta, \Delta^{(i)}\right)$$

- Since g is a binary function

$$\hat{V}_g(\theta) \in [0, 1]$$

Randomized Optimization Problem

- Recall that the randomized optimization problem is given by

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

- This problem is equivalent to

$$\theta_{\text{ncon}} = \arg\min_{\theta \in \mathbf{R}^n} c(\theta) \text{ subject to } \hat{V}_g(\theta) = 0$$

Comments

- Solving the original semi-infinite optimization problem is extremely difficult given the infinite number of constraints
- Using the concept of empirical mean, the optimization problem has only one constraint with a finite sum (for fixed θ)
- Develop a strategy to solve semi-infinite optimization problems such that the empirical mean of violation is zero

Algorithm Non-Convex Design

- Input: ε, δ, n
- Output: θ_{ncon}

- compute the sample size $N_{\text{ncon}}(\varepsilon, \delta, n)$
- draw $N \geq N_{\text{ncon}}(\varepsilon, \delta, n)$ i.i.d. samples $\Delta^{(i)}$
- compute (local) solution of the non-convex problem

$$\theta_{\text{ncon}} = \arg\min_{\theta \in \mathbf{R}^n} c(\theta) \quad \text{subject to} \quad \hat{V}_g(\theta) = 0$$

Aircraft Example Revisited: Learning Design

IEIIT-CNR

- In this example we consider Hurwitz stability instead of quadratic stability (the problem is non-convex)
- The objective is to determine a controller K that computes a probabilistic solution to the optimization problem

$$\min_{\alpha,K}(-\alpha) \text{ subject to } \big(A(\Delta) + B(\Delta)K + \alpha \mathrm{I}\big) \text{ Hurwitz for all } \Delta \in \mathcal{B}$$

$$K_{ij} \in [-\overline{K}_{ij}, \overline{K}_{ij}]$$

Bounds on the Gain Matrix

- The matrix $\overline{K}$ is given by

$$\overline{K} = \begin{bmatrix} 5 & 0.5 & 5 & 5 \\ 5 & 2 & 20 & 1 \end{bmatrix}$$

Sample Complexity

- By means of tedious computations involving reformulation of Hurwitz stability in terms of polynomial Boolean functions we obtain

$$n = 9, \gamma = 10, m = 20$$

- Setting $\varepsilon = 0.01$ and $\delta = 10^{-6}$ the sample complexity can be easily derived

$$N_{\mathrm{ncon}}(\varepsilon, \delta, n) = 366{,}130$$

Probabilistic Controller K_{ncon}

- Probabilistic controller for Hurwitz stability is given by

$$K_{\text{ncon}} = \begin{bmatrix} 0.8622 & 0.2714 & -5.0000 & 2.7269 \\ 5.0000 & 1.4299 & 3.9328 & -1.0000 \end{bmatrix}$$

$\alpha = 3.7285$

- We notice that three gains are saturated, i.e. they are equal to the prespecified bound on the gain matrix

A Posteriori Analysis

A Posteriori Analysis

- When a probabilistic controller K_{prob} has been design with one of the previous methods, we need to verify its performance and address the following questions:

1. Is K_{prob} a robust controller (in the classical sense)?
2. What is the probabilistic performance of K_{prob}?

A Posteriori Deterministic Analysis

Worst-Case Performance

- Deterministic (or worst-case) analysis provides the radius of deterministic performance ρ_{wc}
- The radius ρ_{wc} is the largest value of $\rho > 0$ for which the constraint

$$J(\theta, \Delta) \leq 0$$

is robustly satisfied for all $\Delta \in \mathcal{B}_\rho = \{\Delta \in \rho\mathcal{B}\}$

Aircraft Example Revisited: Worst-Case Analysis

- Consider the previous aircraft example and study the dependence of $A(\Delta)$ and $B(\Delta)$ on uncertain parameters

$$\Delta = [\Delta_1, \Delta_2, \ldots, \Delta_l]^T$$

restricted in the hyperrectangle $\mathcal{B}_\rho$

- We notice that $A(\Delta)$ and $B(\Delta)$ depend multiaffinely on Δ

A function f: $\mathbf{R}^l \rightarrow \mathbf{R}$ is multiaffine if the condition holds: If all components $\Delta_1, \ldots, \Delta_l$ except one are fixed, then f is affine

Multiaffine Dependence

$$A(\Delta) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \Delta_1 & \Delta_2 & \Delta_3 \\ \Delta_4 & 0 & \Delta_5 & -1 \\ \Delta_4\Delta_6 & \Delta_7 & \Delta_8 + \Delta_5\Delta_6 & \Delta_9 - \Delta_6 \end{bmatrix}$$

$$B(\Delta) = \begin{bmatrix} 0 & 0 \\ 0 & \Delta_{10} \\ \Delta_{11} & 0 \\ \Delta_{12} + \Delta_6\Delta_{11} & \Delta_{13} \end{bmatrix}$$

Quadratic Performance and Vertices - 1

- For fixed ρ quadratic performance of state space uncertain systems affected by multiaffine uncertainty is equivalent to quadratic performance of the vertex set $\mathcal{B}_\rho$

Quadratic Performance and Vertices - 2

- Recall that

$$\Phi_{QP}(P, W, \Delta) = A(\Delta)P + PA^T(\Delta) + B(\Delta)W^T + WB^T(\Delta) + 2\alpha P$$

- Then, given P_{seq} and W_{seq}

$$\Phi_{QP}(P_{seq}, W_{seq}, \Delta) \leq 0 \text{ for all } \Delta \in \mathcal{B}_\rho$$

if and only if

$$\Phi_{QP}(P_{seq}, W_{seq}, \Delta_v^i) \leq 0 \text{ for all } i = 1, \ldots, 2^l$$

where Δ_v^i represents the i-th vertex of $\mathcal{B}_\rho$

Line Search for Radius Computation

- Computing the worst-case radius requires to solve a one-dimensional problem in the variable ρ and check if $\Phi_{QP}(P_{seq}, W_{seq}, \Delta_v{}^i) \leq 0$ for all vertices of $\mathcal{B}_\rho$
- This problem can be solved using bisection, but an exponential number of vertices of $\mathcal{B}_\rho$ should be considered (8,192 vertices in this case)

Worst-Case Radius of Performance

- Performing this analysis for P_{seq} and W_{seq} we compute the worst-case radius of performance

$$\rho_{wc} = 0.12$$

- Hence robust quadratic performance is guaranteed for all $\Delta \in \mathcal{B}_\rho$, $\rho = [0,0.12]$

A Posteriori Probabilistic Analysis

Controller Reliability

- Recall that the reliability for the controller $K(\theta)$ is

$$R(\theta) = \mathrm{Prob}\{\Delta \in \mathcal{B}:\ J(\theta, \Delta) \leq 0\}$$

- Take $\theta_{seq} = \{P_{seq}, W_{seq}\}$
- Computing $R(\theta_{seq})$ for fixed θ_{seq} requires to solve a difficult integration problem
- We determine an estimate of this probability setting a simple Monte Carlo experiment

Monte Carlo Experiment

- We take N i.i.d. random samples of Δ according to the given probability measure

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

- We evaluate

$$J(\theta_{\mathrm{seq}}, \Delta^{(1)}), J(\theta_{\mathrm{seq}}, \Delta^{(2)}), \ldots, J(\theta_{\mathrm{seq}}, \Delta^{(N)})$$

IEIIT-CNR

Estimated Probability of Reliability

- Given controller θ_{seq}, we construct the estimated probability of reliability

$$\hat{R}_N(\theta_{\text{seq}}) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{I}\left(J(\theta_{\text{seq}}, \Delta^{(i)})\right)$$

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

$$\mathbf{I}\left(J(\theta_{\text{seq}}, \Delta^{(\mathrm{i})})\right) = \begin{cases} 1 & \text{if } J(\theta_{\text{seq}}, \Delta^{(\mathrm{i})}) \leq 0 \\ 0 & \text{otherwise} \end{cases}$$

IEIIT-CNR

Sample Complexity

- We need to compute the size of the Monte Carlo experiment (sample complexity)
- To this end, given ε, δ ∈(0,1), we need to determine the sample complexity *N* such that the probability event

$$\left| R(\theta_{\mathrm{seq}}) - \hat{R}_N(\theta_{\mathrm{seq}}) \right| \leq \varepsilon$$

  holds with probability at least 1- δ
- Sample complexity is provided by the Chernoff Bound

IEIIT-CNR

Probability Degradation Function

- The next step is to study how the estimated probability $\hat{R}_N(\theta_{\mathrm{seq}})$ degrades as a function of the radius ρ
- This is called the probability degradation function
- We can compare this function with the worst-case radius ρ_{wc} to provide additional information for the control designer

Algorithm Probabilistic Analysis

- Input: ε, δ, θ_{seq}
- Output: $\hat{R}_N(\theta_{\text{seq}})$

- compute the sample size $N_{\text{ch}}(\varepsilon, \delta)$
- draw $N \geq N_{\text{ch}}(\varepsilon, \delta)$ i.i.d. samples $\Delta^{(1)}, \Delta^{(2)}, \ldots, \Delta^{(N)}$
- return

$$\hat{R}_N(\theta_{\text{seq}}) = \frac{1}{N} \sum_{\text{i}=1}^{N} \mathbf{I}\Big(J(\theta_{\text{seq}}, \Delta^{(i)})\Big)$$

Numerical Results - 1

- Taking ε=0.005, $\delta=10^{-6}$, by means of the Chernoff bound we obtain N_{ch} =290,174
- Then, we estimate the probability degradation function for 100 equispaced values of ρ in the range [0.12,0.5]
- For each grid point the estimated probability of reliability (or performance) is computed by means of Algorithm Probabilistic Analysis

Numerical Results - 2

- For each grid point ρ, the inequality

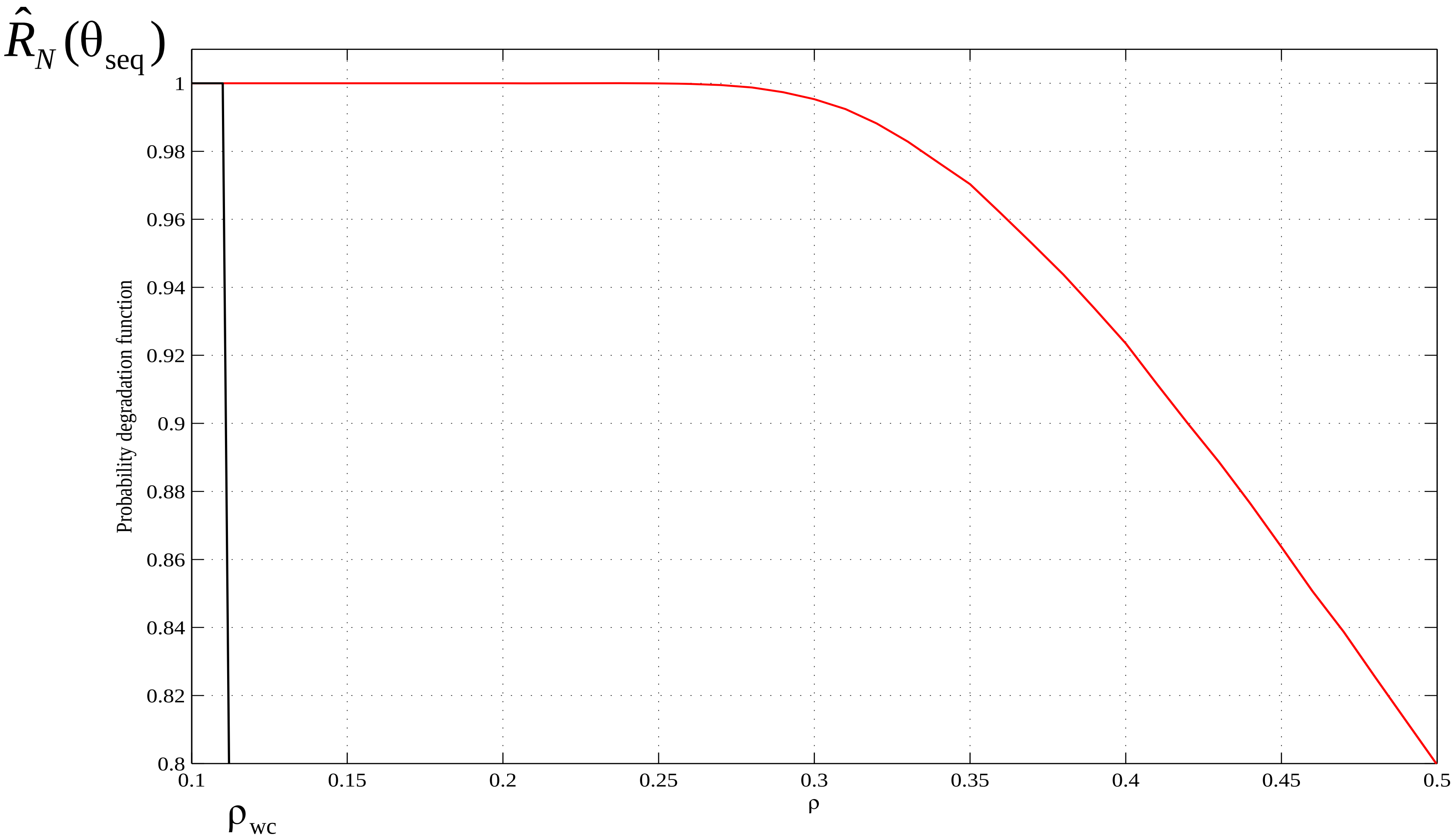
$$\left| R(\theta_{\text{seq}}) - \hat{R}_N(\theta_{\text{seq}}) \right| \leq 0.005$$

holds with probability at least 0.999999

- The probability degradation function is now shown

IEIIT-CNR

Probability Degradation Function

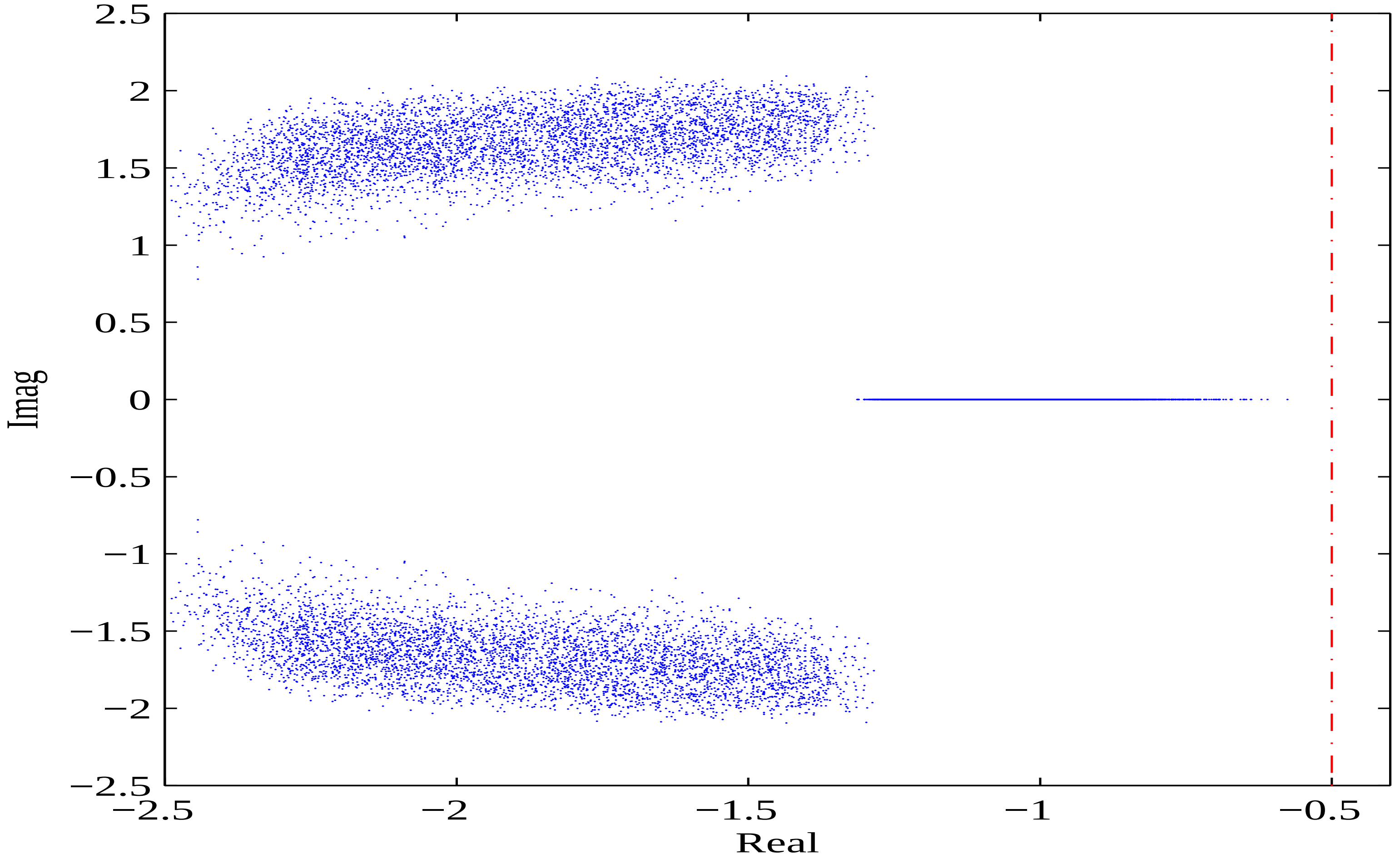


Comments

- We observe that if a 2% loss of probabilistic performance is tolerated, then the performance margin may be increased by 270% with respect to its deterministic counterpart ρ_{wc}
- For ρ=0.34, the estimated probability of performance is 0.98
- Notice that the estimated probability $\hat{R}_N(\theta_{seq})$ is equal to one up to $\rho = 0.26$

IEIIT-CNR

Closed-Loop Eigenvalues for ρ = 0.34



RACT

Randomized Algorithms Control Toolbox

IEIIT-CNR

RACT

- RACT: Randomized Algorithms Control Toolbox for Matlab
- RACT has been developed at IEIIT-CNR and at the Institute for Control Sciences-RAS, based on a bilateral international project
- Members of the project

Andrey Tremba (Main Developer and Maintainer)
Giuseppe Calafiore
Fabrizio Dabbene
Elena Gryazina
Boris Polyak (Co-Principal Investigator)
Pavel Shcherbakov
Roberto Tempo (Co-Principal Investigator)

IEIIT-CNR

RACT

- Main features
- Define a variety of uncertain objects: scalar, vector and matrix uncertainties, with different pdfs
- Easy and fast sampling of uncertain objects of almost any type
- Sequential randomized algorithms for feasibility of uncertain LMIs using stochastic gradient and localization methods (ellipsoid or cutting plane)
- Non-sequential randomized algorithms for optimization of convex problems

RACT

- Under construction
- Non-sequential randomized algorithms for feasibility and optimization of non-convex problems

RACT

- RACT: Randomized Algorithms Control Toolbox for Matlab

http://ract.sourceforge.net

Systems and Control Applications

Systems and Control Applications - 1

- Aerospace control: Applications of randomized strategies for the design of control algorithms for lateral and longitudinal control of aircrafts (e.g. F-16)[1,2]
- Flexible and truss structures: Probabilistic robustness of systems with bounded random uncertainty affecting sensors and actuators[3,4]
- Model (in)validation: Computationally efficient algorithm for robust performance in the presence of structured uncertainty[5]

[1] C.I. Marrison and R.F. Stengel.(1998)
[2] B. Lu and F. Wu (2006)
[3] G. Calafiore, F. Dabbene and R. Tempo (2000)
[4] G.C. Calafiore and F. Dabbene (2008)
[5] M.Sznaier, C.M. Lagoa, and M.C. Mazzaro (2007)

IEIIT-CNR

Systems and Control Applications - 2

- Adaptive control: Methodology for the design of cautious adaptive controllers based on two-step procedure with controller tuning[1]
- Switched systems: Randomized algorithms for synthesis of multimodal systems with state-dependent switching[2]
- Network control: Congestion control of high-speed communication networks using different topologies[3]
- Automotive: Randomization-based approaches for model validation of advanced driver assistance systems[4]

[1] M.C. Campi and M. Prandini (2003)
[2] H. Ishii, T. Basar and R. Tempo (2005)
[3] T. Alpcan, T. Basar and R. Tempo (2005)
[4] O.J. Gietelink, B. De Schutter, and M. Verhaegen (2005)

Systems and Control Applications - 3

- Model predictive control (MPC): Sequential methods (ellipsoid-based) to design robustly stable finite horizon MPC schemes[1]
- Fault detection and isolation: Risk-adjusted randomization approach for robust simultaneous fault detection and isolation of MIMO systems[2]
- Circuits and embedded systems: Performance subject to uncertain components introduced during the manufacturing process[3-4]

[1] S. Kanev and M. Verhaegen (2006)
[2] W. Ma, M.Sznaier and C.M. Lagoa (2007)
[3] C. Lagoa, F. Dabbene and R. Tempo (2008)
[4] C. Alippi (2002)

IEIIT-CNR

Systems and Control Applications - 4

- Unmanned aerial vehicles (UAV): Robust and randomized control design of a mini-UAV[1]

[1] L. Lorefice, B. Pralio and R. Tempo (2007)

IEIIT-CNR

Control Design of a Mini-UAV

IEIIT-CNR

Italian National Project for Fire Prevention

- This activity is supported by the Italian Ministry for Research within the National Project

Study and development of a real-time land control and monitoring system for fire prevention

- Five research groups are involved together with a government agency for fire surveillance and patrol located in Sicily
- The aerial platform is based on the MicroHawk configuration, developed at the Aerospace Engineering Department, Politecnico di Torino, Italy

IEIIT-CNR

MH1000 Platform - 1

- Platform features
  - wingspan 3.28 ft (1 m)
  - total weight 3.3 lb (1.5 kg)

MH1000 Platform - 2

- Main on-board equipment
  - various sensors and two cameras (color and infrared)
- DC motor
- Remote piloting and autonomous flight
- Flight endurance of about 40 min
- Flight envelope
  - min/max velocity: 33 ft/s (10 m/s) – 66 ft/s (17 m/s)
  - average velocity: 43 ft/s (14 m/s)

IEIIT-CNR

Flight Envelope (Limits)

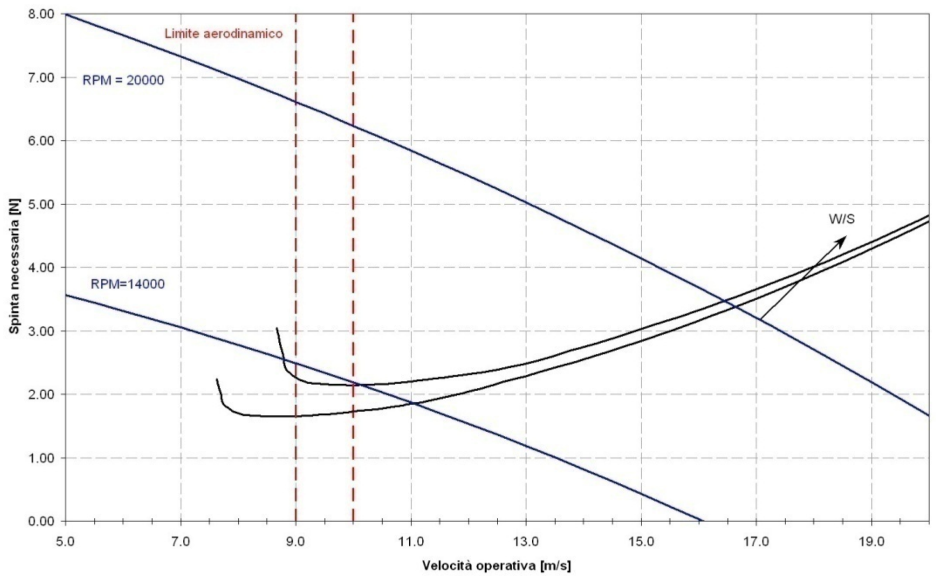
Wing loading effect → total weight

Aerodynamic constraint (red) → minimum flight speed (stall effect)

Propeller sizing effect

Propulsive constraint (blu) → maximum flight speed

velocity: 33 ft/s (10 m/s) – 66 ft/s (17 m/s)



IEIIT-CNR

Basic on-board Systems

DC motor: Hacker B20-15L (4:1)

- weight: 58 g
- dimensions: Ø 20 x 40 mm
- Kv: 3700 rpm/volt

controller: Hacker Master Series 18-B-Flight

- weight: 21 g
- dimensions: 33 X 23 X 7 mm
- current drain: 18 A

battery: Kokam 2000HD (3x)

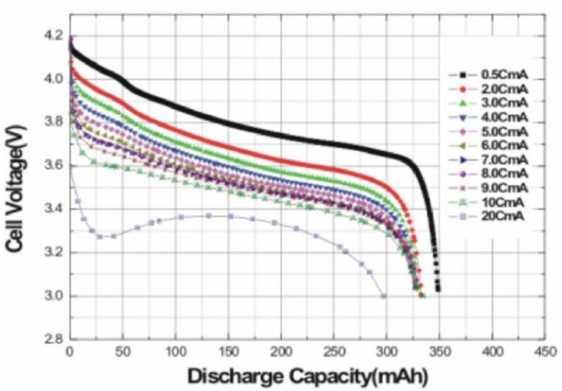
- weight: 160 g
- dimensions: 79 X 42 X 25 mm
- capacity: 2000 mAh

receiver: Schulze Alpha840W

- weight: 13.5 g
- dimensions: 52 X 21 X 13 mm
- 8 channels

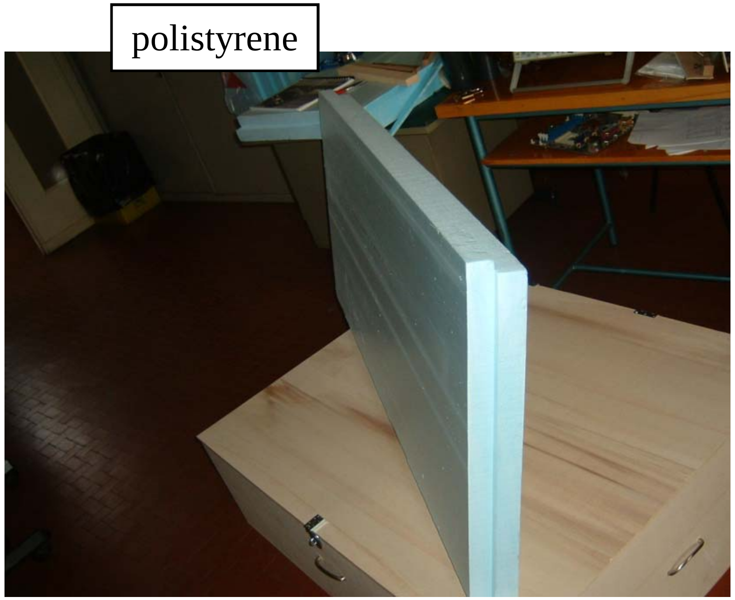
servo: Graupner C1081 (2x)

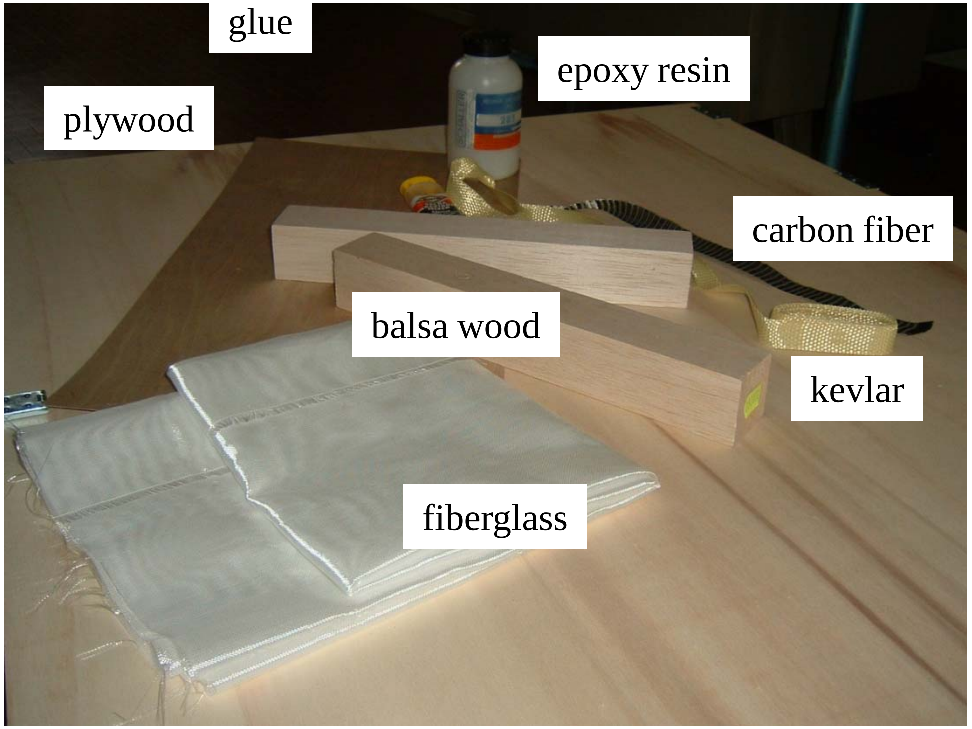
- weight: 13 g
- dimensions: 23 X 9 X 21 mm
- torque: 12 Ncm



Prototype Manufacturing - 1

raw material

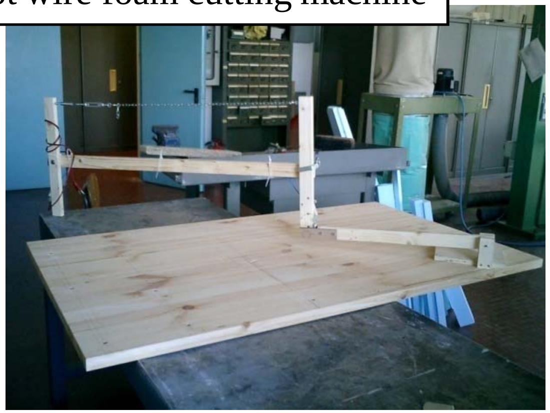




Prototype Manufacturing - 2

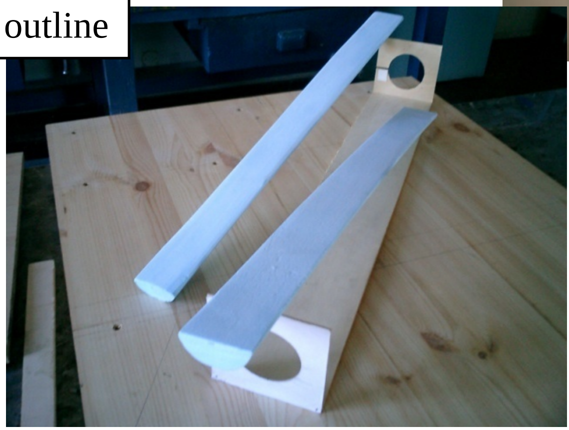
working instruments

hot wire foam cutting machine



lifting surfaces outline

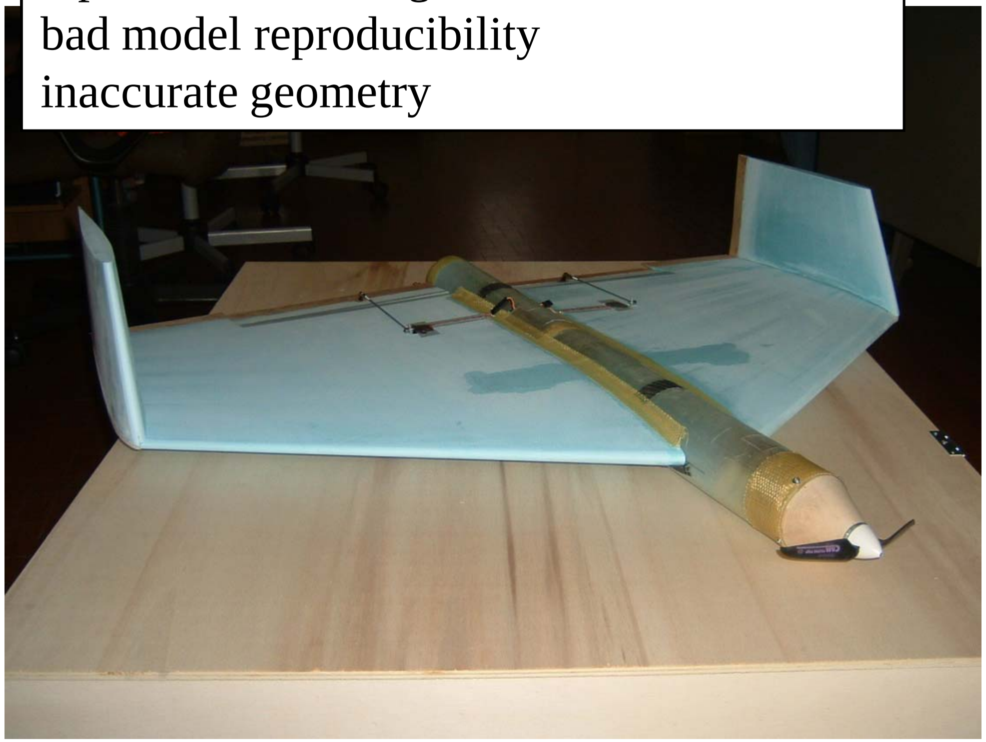
slide outline

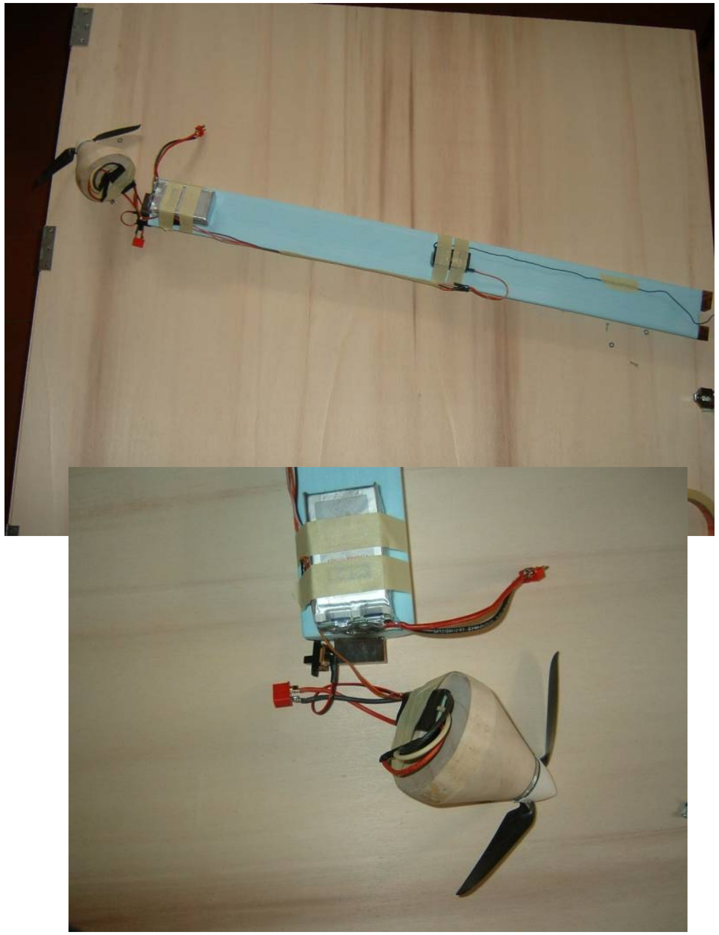


fuselage reference

Prototype Manufacturing - 3

easy construction
rapid manufacturing
bad model reproducibility
inaccurate geometry





State Space Model

- State space formulation obtained by linearization of the full (12 coupled nonlinear ODE) model

$$\dot{x}(t) = A(\Delta)\, x(t) + B(\Delta)\, u(t)$$

$$u(t) = -K\, x(t)$$

where $x = [V, \alpha, q, \theta]^T$ (V flight speed, α angle of attack, q and θ pitch rate and angle), Δ uncertainty

- Consider longitudinal plane dynamics stabilization
- Control u is the symmetrical elevon deflection

Uncertainty Description - 1

- We consider structured parameter uncertainties affecting plant and flight conditions, and aerodynamic database
- Uncertainty vector $\Delta = [\Delta_1,\ldots, \Delta_{16}]$ where $\Delta_i \in [\Delta_i^-, \Delta_i^+]$
- Key point: There is no explicit relation between state space matrices A and B and uncertainty Δ
- This is due to the fact that state space system is obtained through linearization and off-line flight simulator
- The only techniques which could be used in this case are simulation-based which lead to randomized algorithms

IEIIT-CNR

Uncertainty Description - 2

- We consider random uncertainty $\Delta = [\Delta_1,..., \Delta_{16}]^T$
- The pdf is either uniform (for plant and flight conditions) or truncated Gaussian (for aerodynamic database uncertainties)
- Flight conditions uncertainties need to take into account large variations on physical parameters
- Uncertainties for aerodynamic data are related to experimental measurement or round-off errors

IEIIT-CNR

Plant and Flight Condition Uncertainties

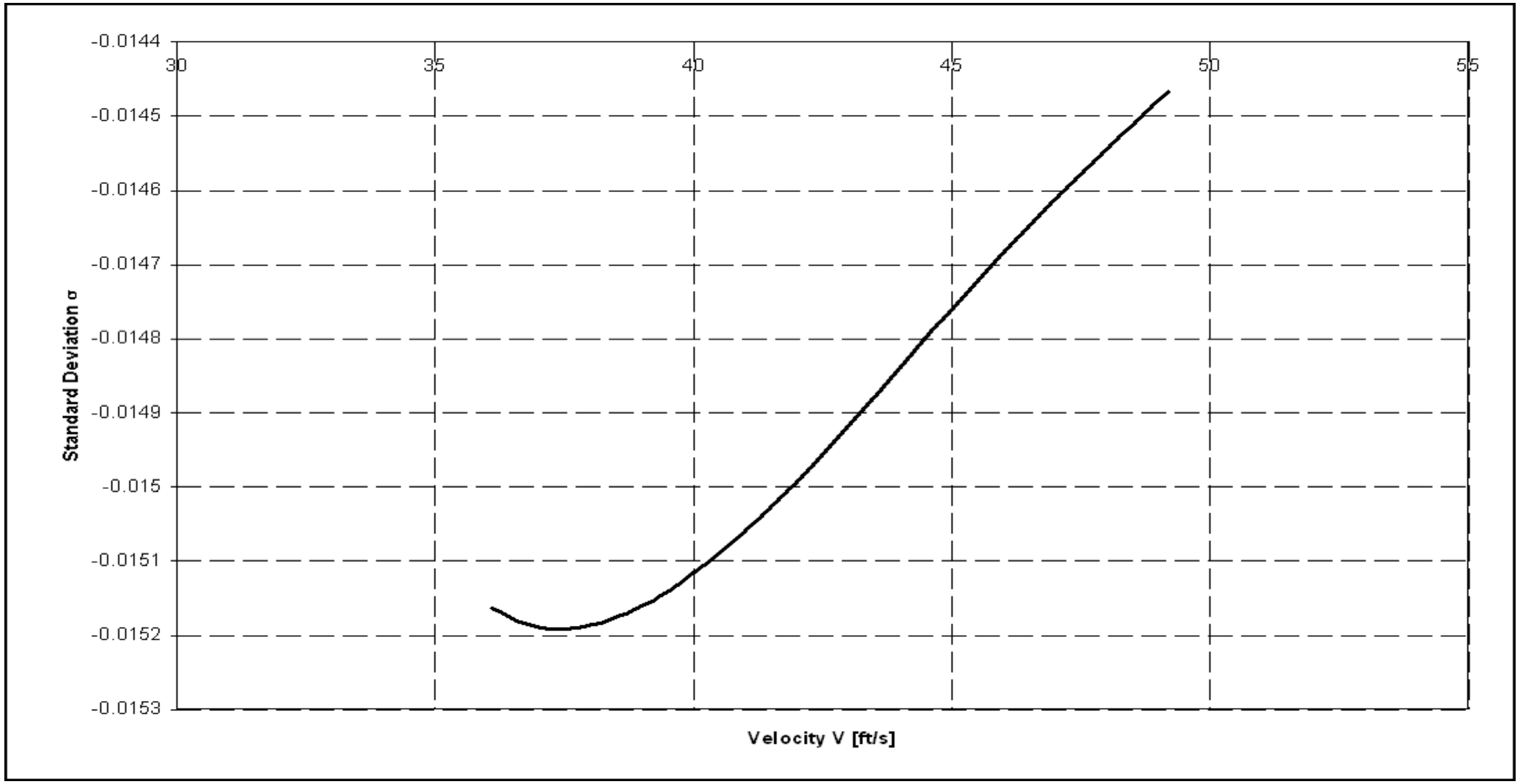
parameter	pdf	$\bar{\Delta}_i$	%	Δ_i^-	Δ_i^+	#
flight speed [ft/s]	$\mathcal{U}$	42.65	± 15	36.25	49.05	1
altitude [ft]	$\mathcal{U}$	164.04	± 100	0	328.08	2
mass [lb]	$\mathcal{U}$	3.31	± 10	2.98	3.64	3
wingspan [ft]	$\mathcal{U}$	3.28	± 5	3.12	3.44	4
mean aero chord [ft]	$\mathcal{U}$	1.75	± 5	1.67	1.85	5
wing surface [ft^2]	$\mathcal{U}$	5.61	± 10	5.06	6.18	6
moment of inertia [lb ft^2]	$\mathcal{U}$	1.34	± 10	1.21	1.48	7

Aerodynamic Database Uncertainties

parameter	pdf	$\bar{\Delta}_i$	σ_i	#
C_X [-]	$\mathcal{G}$	-0.01215	0.00040	8
C_Z [-]	$\mathcal{G}$	-0.30651	0.00500	9
C_m [-]	$\mathcal{G}$	-0.02401	0.00040	10
C_{Xq} [rad^{-1}]	$\mathcal{G}$	-0.20435	0.00650	11
C_{Zq} [rad^{-1}]	$\mathcal{G}$	-1.49462	0.05000	12
C_{mq} [rad^{-1}]	$\mathcal{G}$	-0.76882	0.01000	13
C_X [rad^{-1}]	$\mathcal{G}$	-0.17072	0.00540	14
C_Z [rad^{-1}]	$\mathcal{G}$	-1.41136	0.02200	15
C_m [rad^{-1}]	$\mathcal{G}$	-0.94853	0.01500	16

IEIIT-CNR

Standard Deviation and Velocity



Standard deviation is experimentally computed from the velocity

Critical Parameters and Matrices

- We select flight speed (Δ_1) and take off mass (Δ_3) as critical parameters
- Flight speed is taken as critical parameter to optimize gain scheduling issues
- Take off mass is a key parameter in mission profile definition
- We define critical matrices

$$A_c^1 \quad A_c^2 \quad A_c^3 \quad A_c^4 \qquad B_c^1 \quad B_c^2 \quad B_c^3 \quad B_c^4$$

- They are constructed setting Δ_1, Δ_3 to their extreme values; the remaining Δ_i are set to their nominal values

Phase 1: Random Gain Synthesis (RGS)

- Critical parameters are flight speed and take off mass
- Specification property

$$\mathcal{S}_1 = \{K\text{: } A_c - B_c K \text{ satisfies the specs below}\}$$

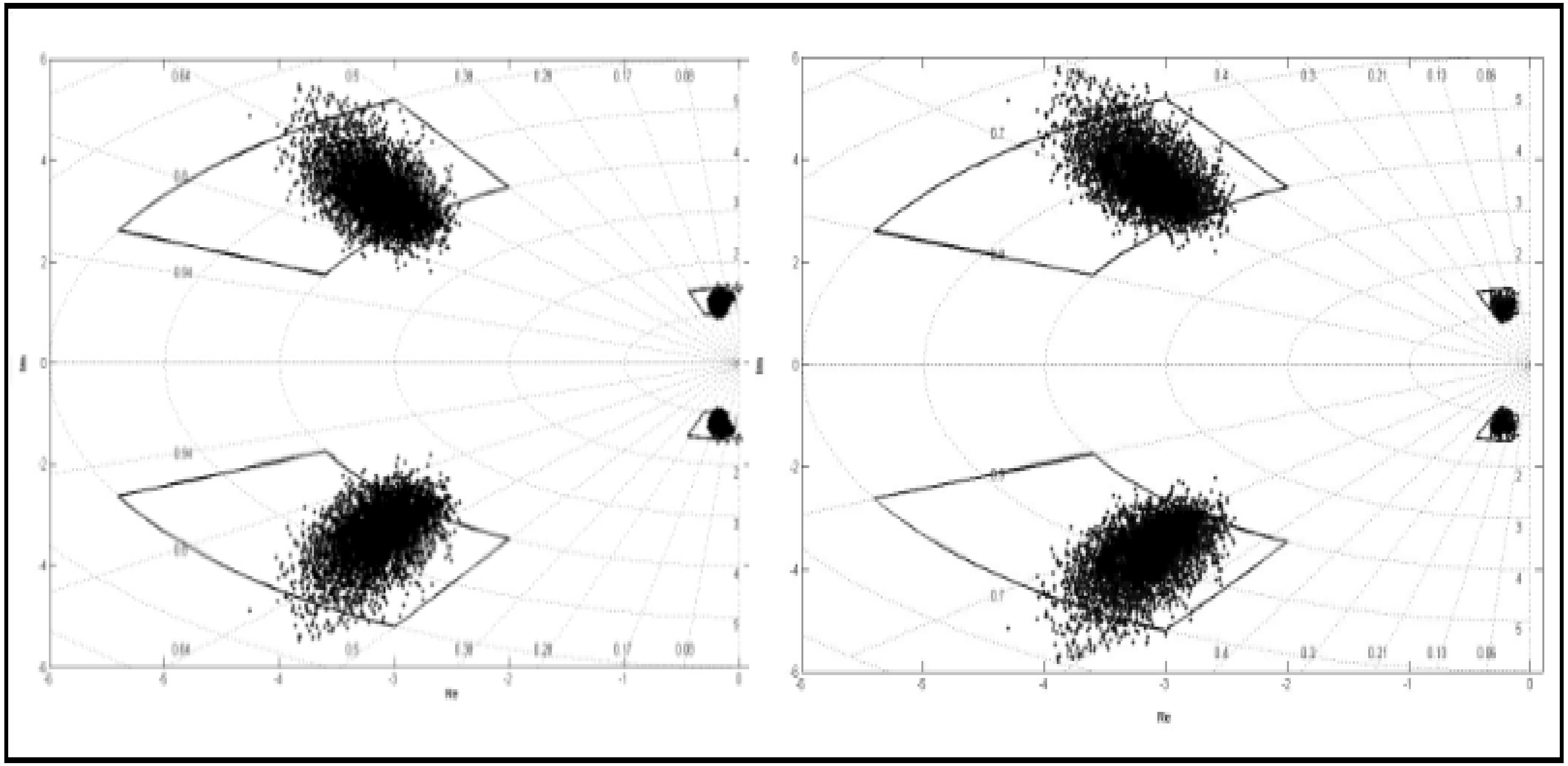
$$\omega_{SP} \in [4.0, 6.0] \text{ rad/s} \qquad \zeta_{SP} \in [0.5, 0.9] \qquad \omega_{PH} \in [1.0, 1.5] \text{ rad/s}$$

$$\zeta_{PH} \in [0.1, 0.3] \qquad \Delta\omega_{SP} < \pm\, 45\% \qquad \Delta\omega_{PH} < \pm\, 20\%$$

where ω and ζ are undamped natural frequency and damping ratio of the characteristic modes; $_{SP}$ and $_{PH}$ denote short period and phugoid mode

IEIIT-CNR

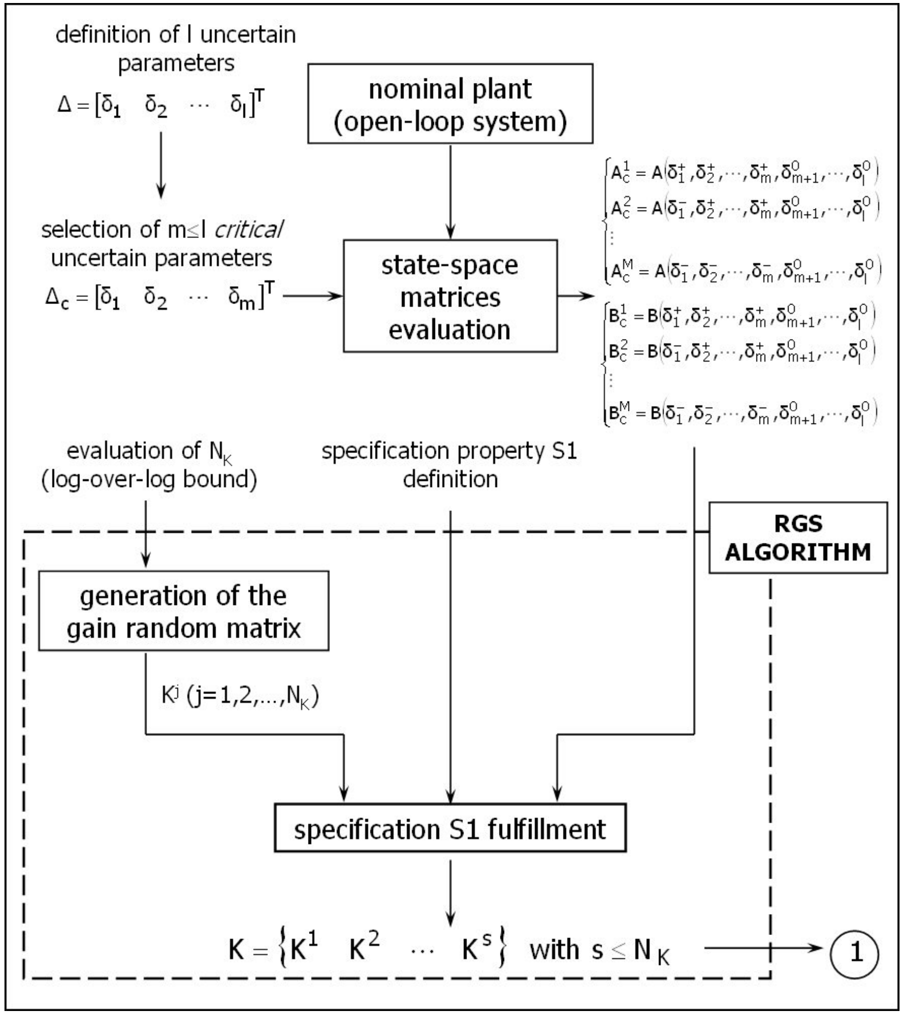
Specs in the Complex Plane



Randomized Algorithm 1 (RGS)

- Uniform pdf for controller gains *K* in given intervals
- Accuracy and confidence

  $\varepsilon = 4 \cdot 10^{-5}$ and $\delta = 3 \cdot 10^{-4}$
- Number of random samples is computed with "Log-over-Log" Bound obtaining $N = 200{,}000$
- We obtained $s = 5$ gains K^i satisfying specification property S_1



Random Gain Set

gain set	K_V	K_α	K_q	K_θ
K^1	0.00044023	0.09465000	0.01577400	-0.00473510
K^2	0.00021450	0.09581200	0.01555500	-0.00323510
K^3	0.00054999	0.09430800	0.01548200	-0.00486340
K^4	0.00010855	0.09183200	0.01530000	-0.00404380
K^5	0.00039238	0.09482700	0.01609300	-0.00417340

IEIIT-CNR

Phase 2: Random Stability Robustness Analysis (RSRA)

- Take $K_{\text{rand}} = K^i$ obtained in Phase 1
- Randomize Δ according to the given pdf and take N random samples Δ^i
- Specification property

$$\mathcal{S}_2 = \{\Delta: A(\Delta) - B(\Delta)\ \ K_{\text{rand}}\ \ \text{satisfies the specs of } \mathcal{S}_1\}$$

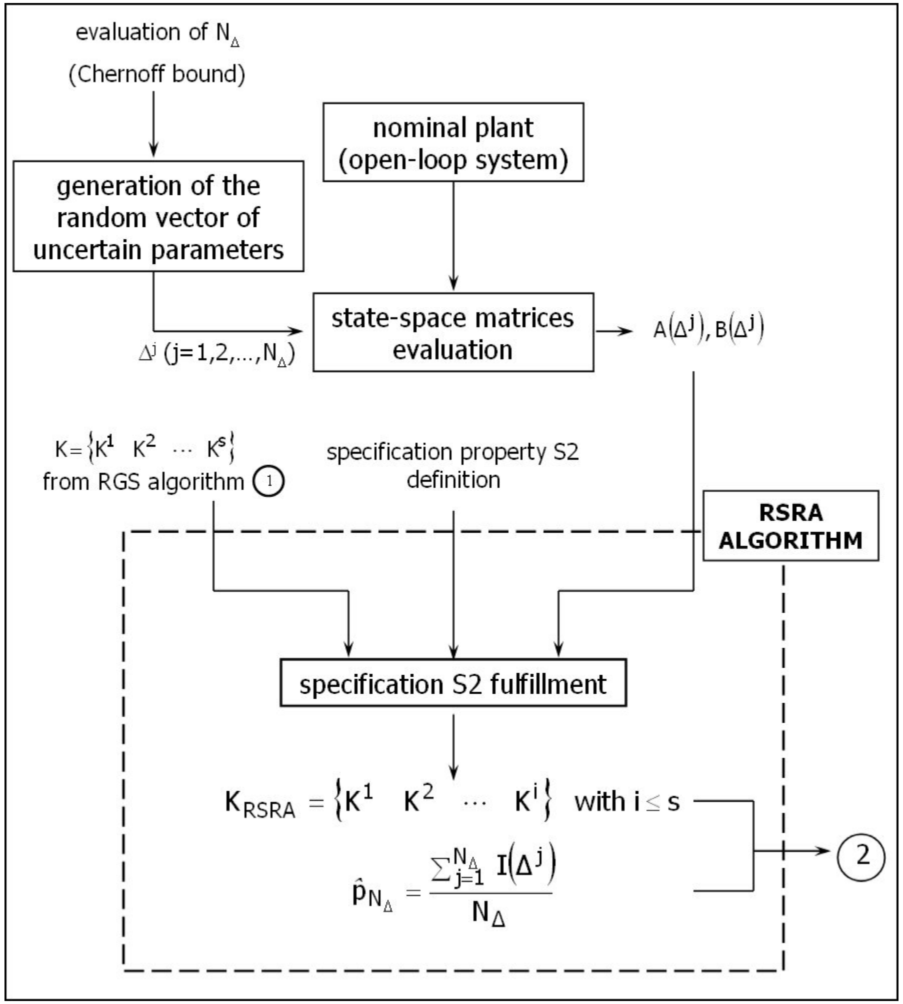
- Computation of the empirical probability of stability

Randomized Algorithm 2 (RSRA)

- Take K_{rand} from Phase 1
- Accuracy and confidence

$$\varepsilon = \delta = 0.0145$$

- Number of random samples is computed with Chernoff Bound obtaining N =5,000
- Empirical probability is computed



Empirical Probability of Stability for Phase 2

gain set	empirical probability
K^1	88.56%
K^2	90.60%
K^3	89.31%
K^4	93.86%
K^5	85.14%

IEIIT-CNR

Probability Degradation Function

- Flight condition uncertainties are multiplied by the radius $\rho > 0$ keeping the nominal value constant

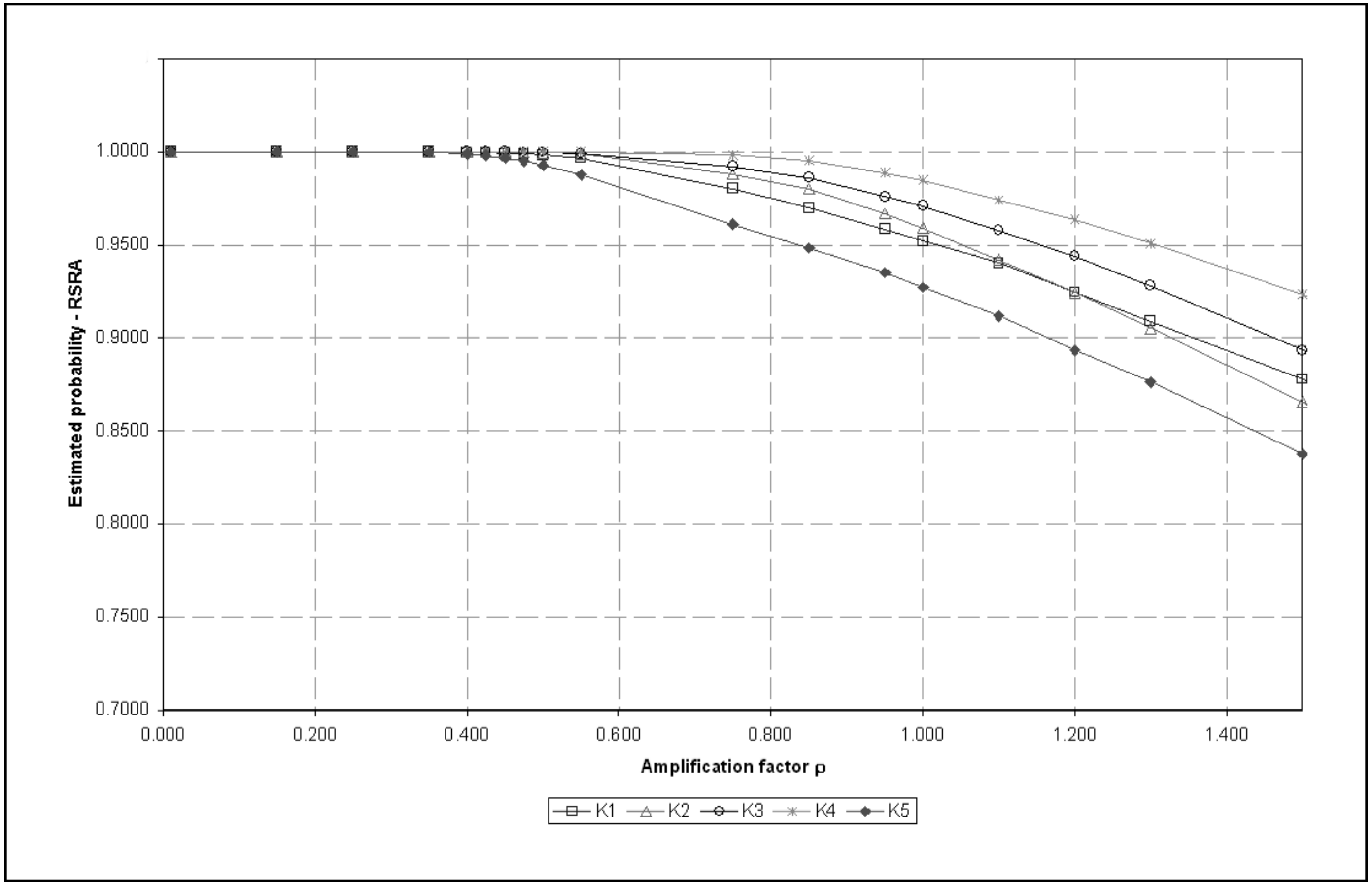
$$\Delta_i \in \rho\,[\Delta_i^-, \Delta_i^+] \qquad \text{for } i = 1, 2, \ldots, 7$$

- No uncertainty affects the aerodynamic database, i.e.

$$\Delta_i = \bar{\Delta}_i \qquad \text{for } i = 8, 9, \ldots, 16$$

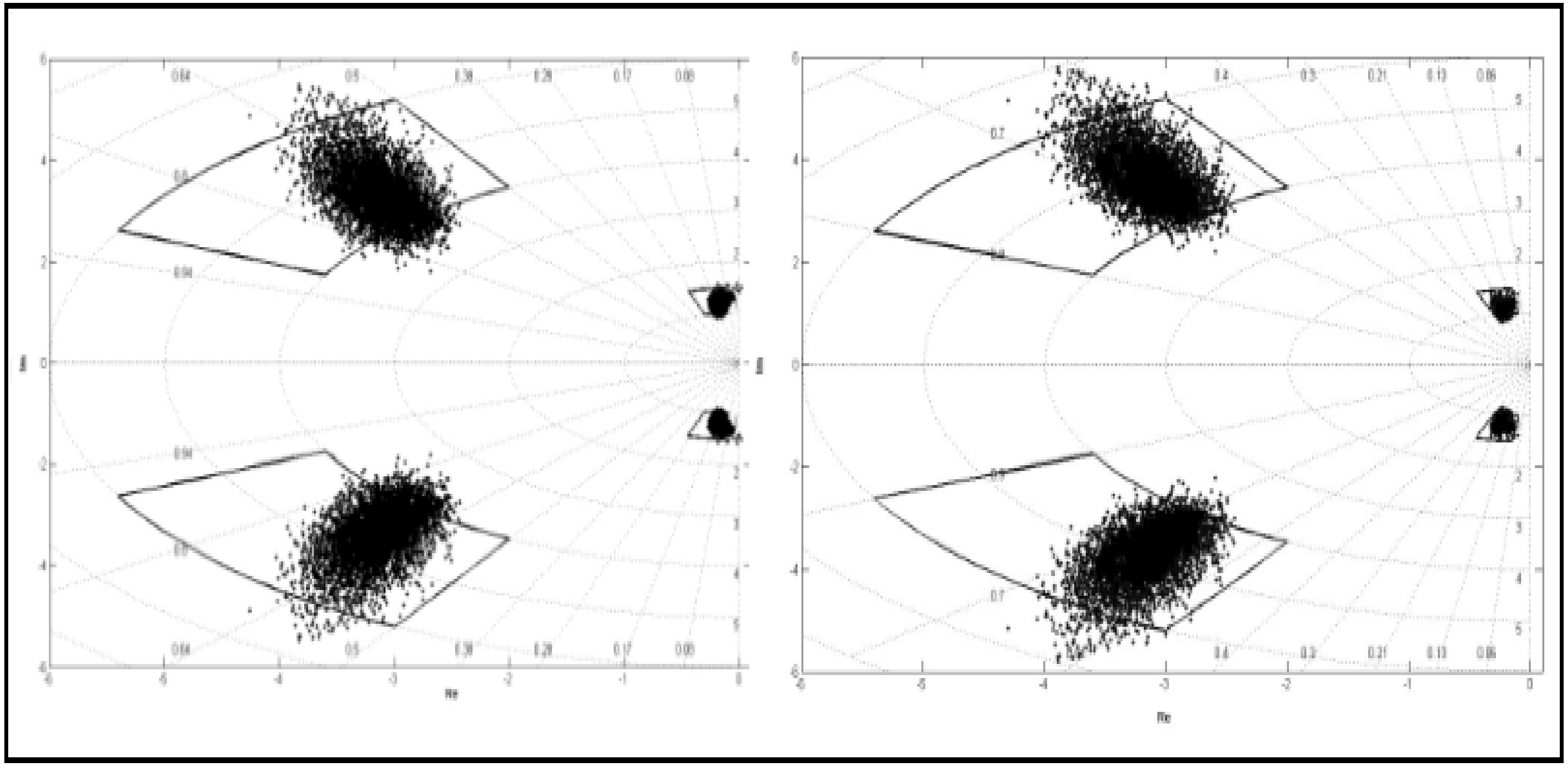
- For fixed $\rho \in [0,1.5]$ we compute the empirical probability for different gain sets K^i
- The plot empirical probability vs ρ is the probability degradation function

Probability Degradation Function for Phase 2



IEIIT-CNR

Root Locus Plot for Phase 2



Root locus for K^2 (left) and K^4 (right)

Phase 3: Random Performance Robustness Analysis (RPRA)

- This phase is similar to Phase 2, but military specs are considered (bandwidth criterion)
- Specification property

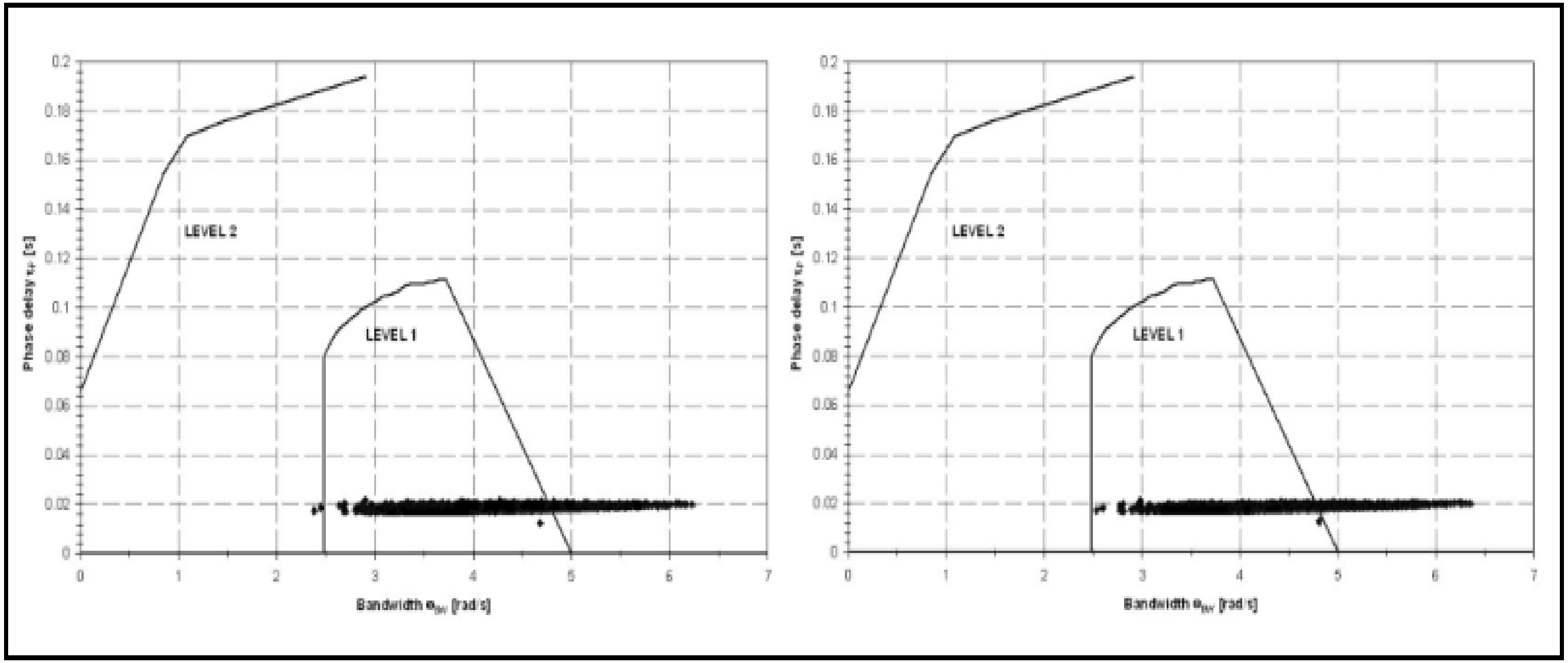
$$S_3 = \{\Delta: A(\Delta) - B(\Delta)\ K_{\text{rand}} \text{ satisfies the specs below}\}$$

$$\omega_{BW} \in [2.5, 5.0] \text{ rad/s} \qquad \tau_P \in [0.0, 0.5] \text{ s}$$

where ω_{BW} and τ_P are bandwidth and phase delay of the frequency response

- Computation of the empirical probability that S_3 is satisfied

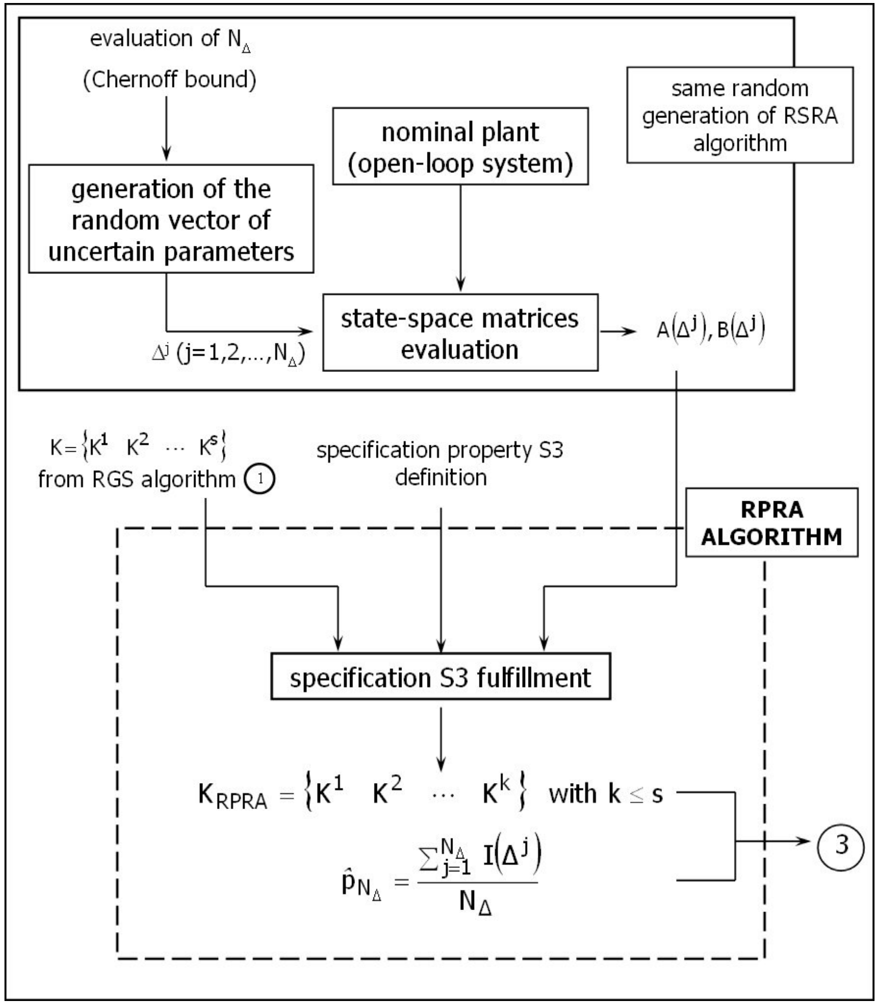
Bandwidth Criterion



IEIIT-CNR

Randomized Algorithm 3 (RPRA)

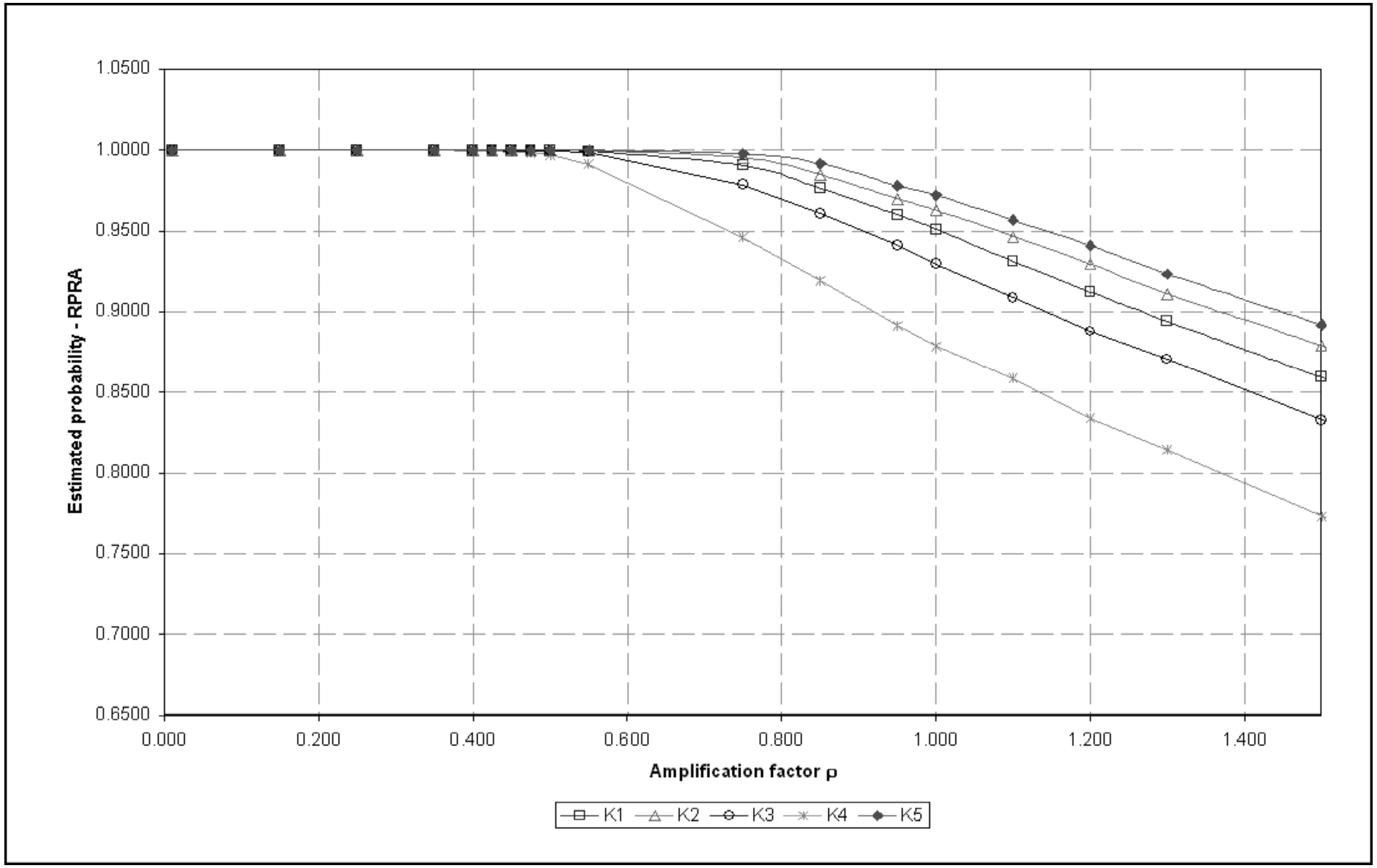
- Take K_{rand} from Phase 1
- Numer of random samples is computed with the Chernoff Bound obtaining N =5,000
- Empirical probability is computed



Empirical Probability of Performance for Phase 3

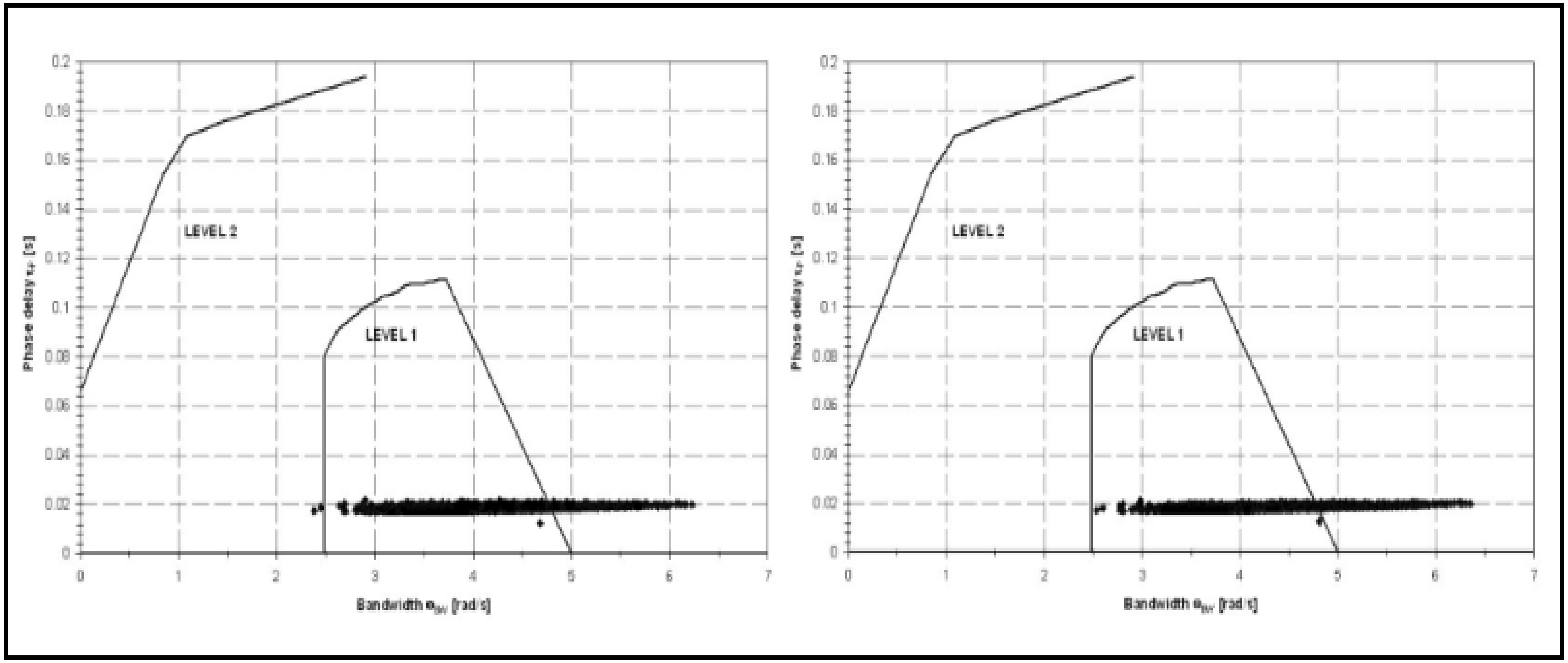
gain set	empirical probability
K^1	93.58%
K^2	95.16%
K^3	90.80%
K^4	84.78%
K^5	96.06%

Probability Degradation Function for Phase 3



IEIIT-CNR

Bandwidth Criterion for Phase 3



Bandwidth criterion for K^1 (left) and K^3 (right)

Gain Selection

- Multi-objective criterion as a compromise between different specifications

Finally we selected gain K^1 as the best compromise between all the specs and criteria

IEIIT-CNR

Conclusions: Flight Tests in Sicily - 1

- Evaluation of the payload carrying capabilities and autonomous flight performance
- Mission test involving altitude, velocity and heading changing was performed in Sicily
- Checking effectiveness of the control laws for longitudinal and lateral-directional dynamics
- Flight control design based on RAs for stabilization and guidance

Conclusions: Flight Tests in Sicily - 2

- Satisfactory response of MH1000
- Possible improvements by iterative design procedure
- Stability of the platform is crucial for the video quality and in the effectiveness of the surveillance and monitoring tasks

IEIIT-CNR

Color Camera: Right Turn

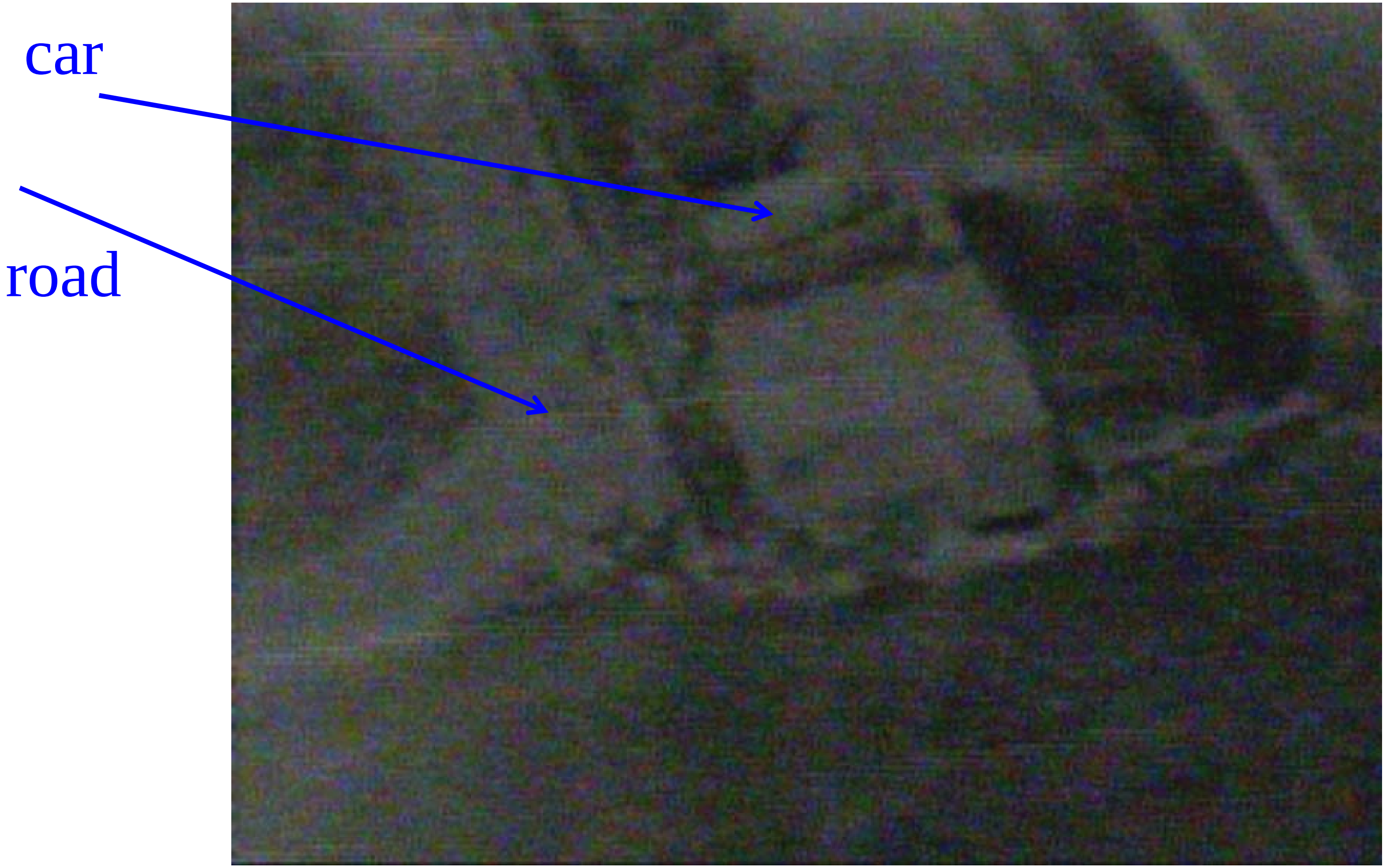
IEIIT-CNR

Color Camera: Landing Phase

Infrared Camera - 1

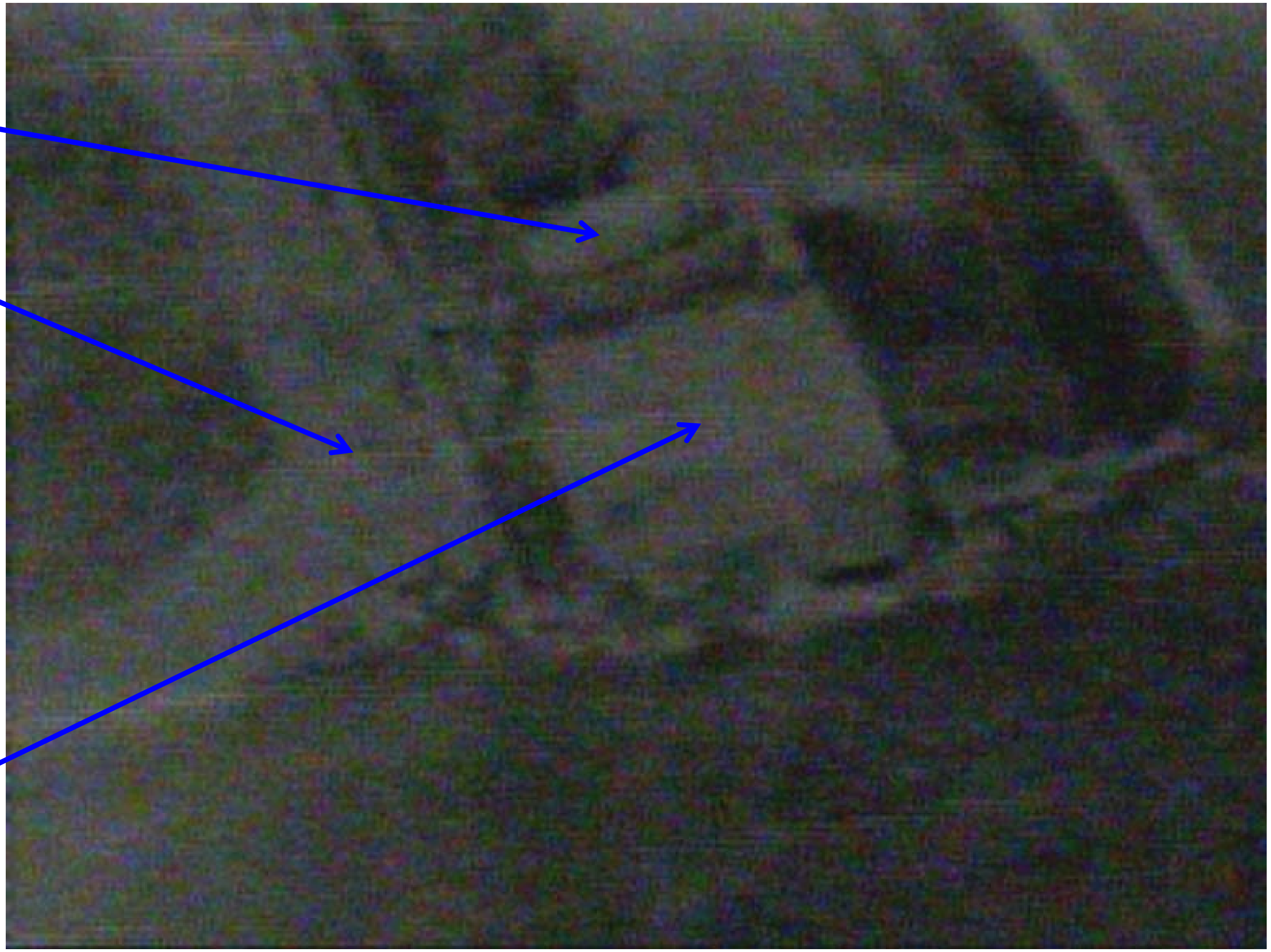
car

Infrared Camera - 1



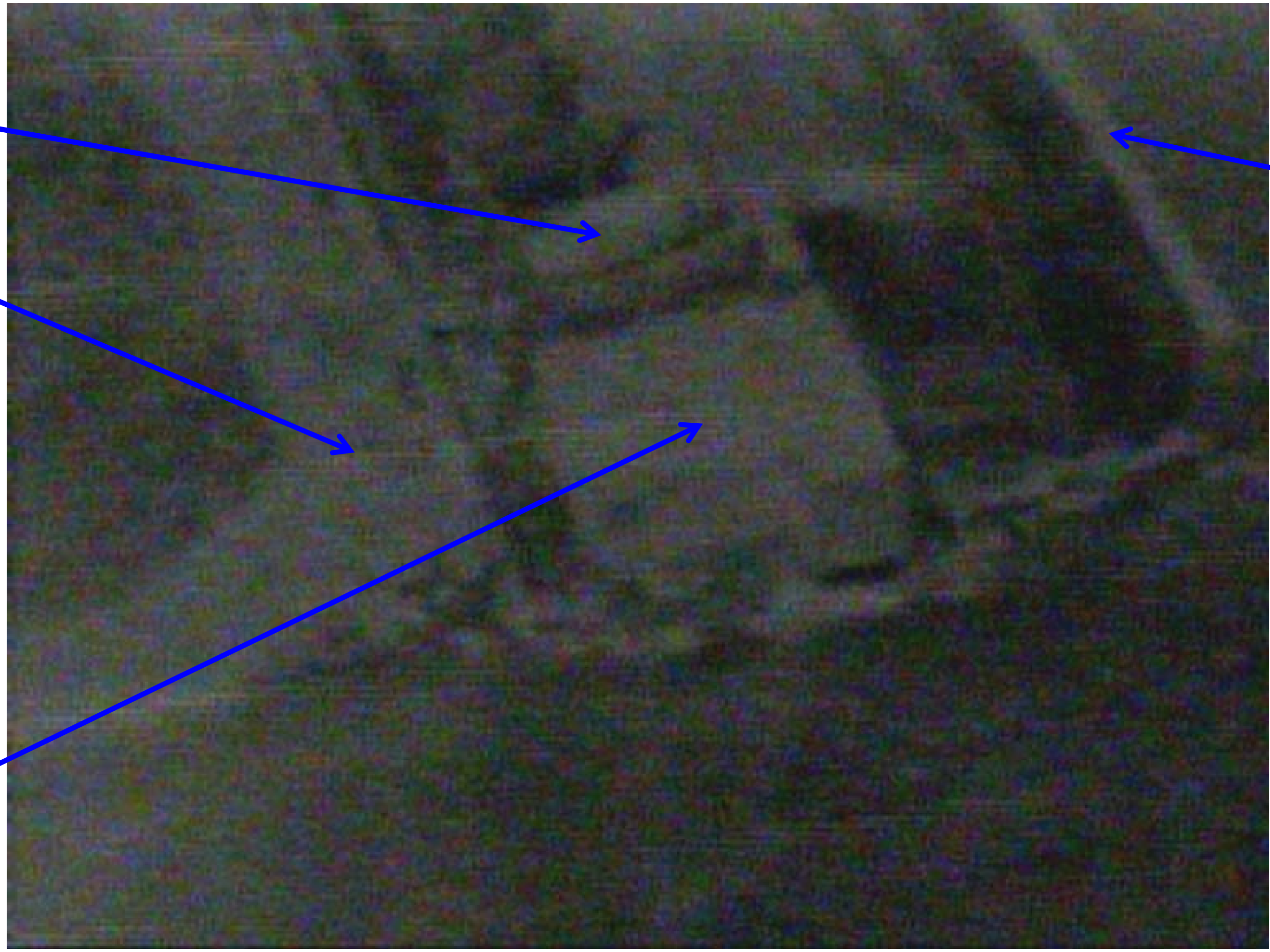
IEIIT-CNR

Infrared Camera - 1



IEIIT-CNR

Infrared Camera - 1



Infrared Camera - 2

Infrared Camera - 3