Randomization of Uncertain Systems: A New Paradigm for Robust Control

Roberto Tempo

IEIIT-CNR

Politecnico di Torino

roberto.tempo@polito.it

- Additional documents, papers, etc, please consult

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

- Questions after the conference may be sent to

  roberto.tempo@polito.it

- Some MATLAB™ codes are available at

  http://www.polito.it/~dabbene

Overview

- Preliminaries
- Randomized Algorithms for Analysis
- Probabilistic Robust Synthesis
- Randomized Algorithms for Optimal Control (LQR)
- Probabilistic LPV Systems
- Applications

1. High Speed Networks
2. Control of Mini UAV

Preliminaries

IEIIT-CNR

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 tutorial: Introduction to rigorous study of RAs for uncertain systems and control, with specific applications

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)

References

- “Randomized Algorithms” by R. Motwani and P. Raghavan, Cambridge University Press, 1995
- “Randomized Algorithms for Analysis and Control of Uncertain Systems” by R. Tempo, G. Calafiore and F. Dabbene, Springer, 2005

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 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

Limitations of Robust Control - 2

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

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

Conservatism and Complexity Trade-Off

- Uncertain or control design parameters often enter into the system in a nonlinear/nonconvex fashion
- To avoid complexity issues (or just to find a solution of the problem) relaxation techniques such as SOS are used
- Study issues about the accuracy of the approximation introduced and related complexity

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

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 $\mathcal{B}_{\mathbb{D}}$
  - 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 $\mathcal{B}_{\mathbb{D}}$

$$\Delta \in \mathcal{B}_{\mathbb{D}}$$

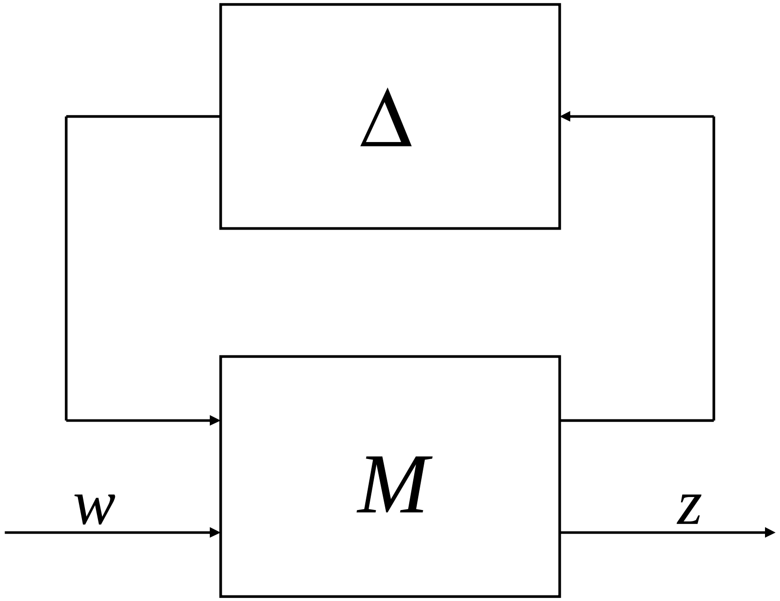
- Real parametric uncertainty $q=[q_1,\ldots, q_\ell] \in \mathbb{R}^\ell$

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

- Nonparametric uncertainty

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

Robustness



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

Objective of Robustness

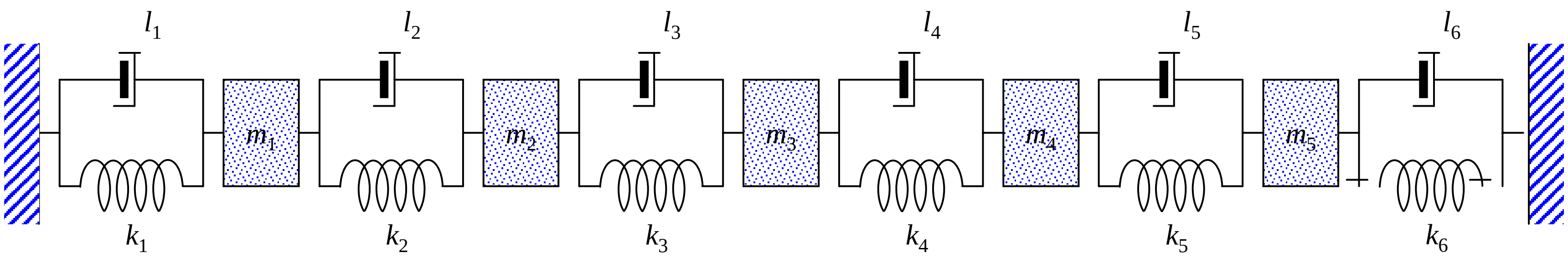
- Objective of robustness: To guarantee stability and performance for all

$$\Delta \in \mathcal{B}_{\mathbb{D}}$$

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

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 systems and study stability of

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

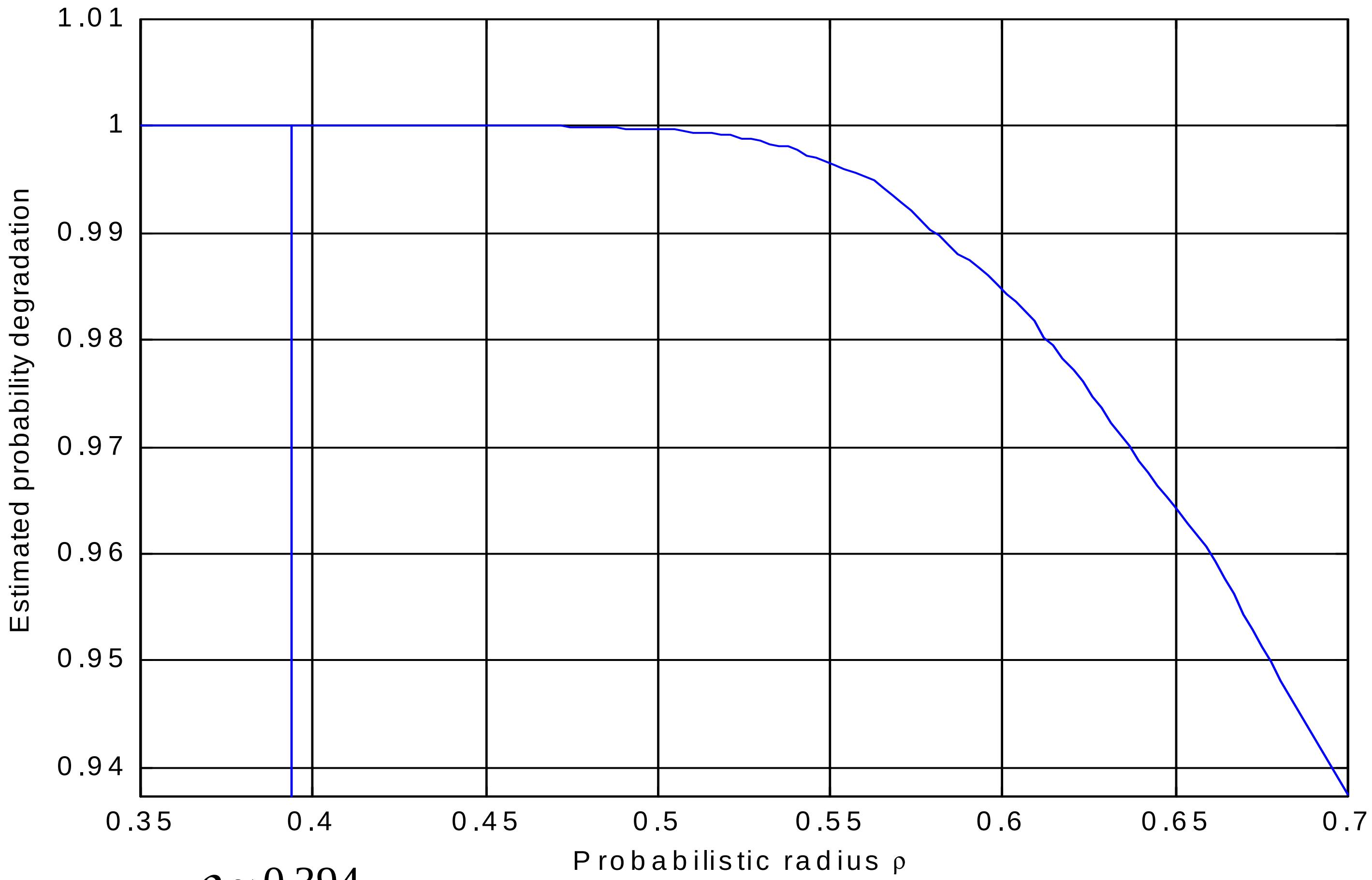
$$\Delta = \begin{bmatrix} q_1 I_5 & 0 & 0 \\ 0 & q_2 I_5 & 0 \\ 0 & 0 & \Delta_1 \end{bmatrix}$$

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

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

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

Probability Degradation Function



$\rho \approx 0.394$

Probabilistic Model

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

IEIIT-CNR

Uniform Density

- Take $f_\Delta(\Delta)=\mathcal{U}[\mathcal{B}_{\mathbb{D}}]$ (uniform density within $\mathcal{B}_{\mathbb{D}}$)

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

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

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

Performance Function

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

$$J = J(\Delta)$$

- Examples: $\mathcal{H}_\infty$ performance and robust stability

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

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

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

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

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

for all Δ in $\mathcal{B}_{\mathbb{D}}$

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

- Continuous time SISO systems with real parametric uncertainty q with upper LFT

$$\mathcal{F}_u(M,\Delta) = \mathcal{F}_u(M,q) =$$

$$\frac{0.5q_1q_2s + 10^{-5}q_1}{(10^{-5} + 0.05q_2)s^2 + (0.00102 + 0.5q_2)s + (2\cdot 10^{-5} + 0.5q_1^2)}$$

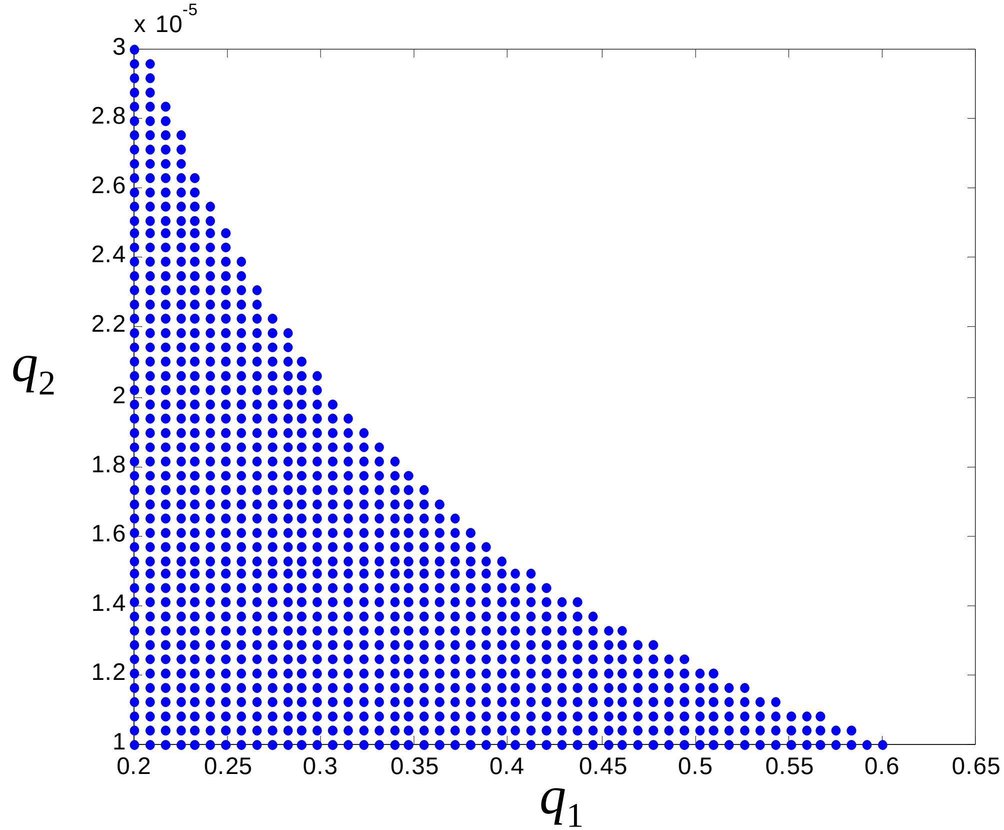
where $q_1 \in [0.2, 0.6]$ and $q_2 \in [10^{-5}, 3\cdot 10^{-5}]$

- Letting $J(q) = \| \mathcal{F}_u(M,q) \|_\infty$, we choose γ=0.003
- Check if $J(q)<\gamma$ for all q in these intervals

IEIIT-CNR

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

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



IEIIT-CNR

Example[1]: Robust Stability - 1

- Consider the closed loop uncertain polynomial

$$p(s,q) =$$

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

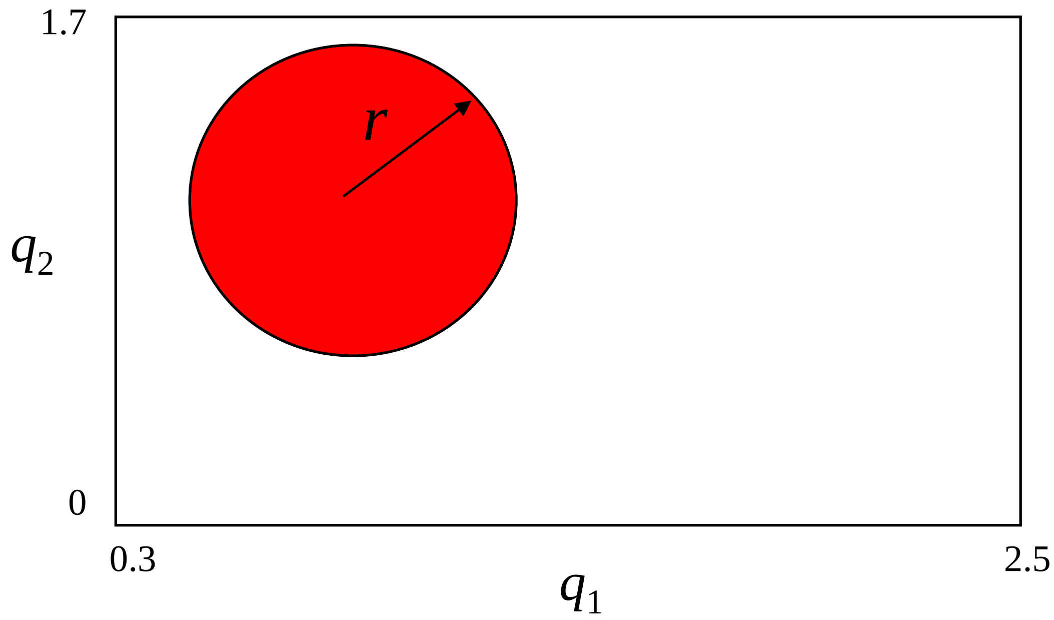
where $q_1 \in [0.3, 2.5]$, $q_2 \in [0,1.7]$ and r=0.5

- Check stability for all q in these intervals

[1] G. Truxal (1961)

Example: Robust Stability - 2

- 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 $\mathcal{B}_{\mathbb{D}}$

P2: Worst-Case Performance

- Find $J_{\max}$ such that

$$J_{\max} = \max_{\Delta \in \mathcal{B}_{\mathbb{D}}} J(\Delta)$$

IEIIT-CNR

Good and Bad Sets

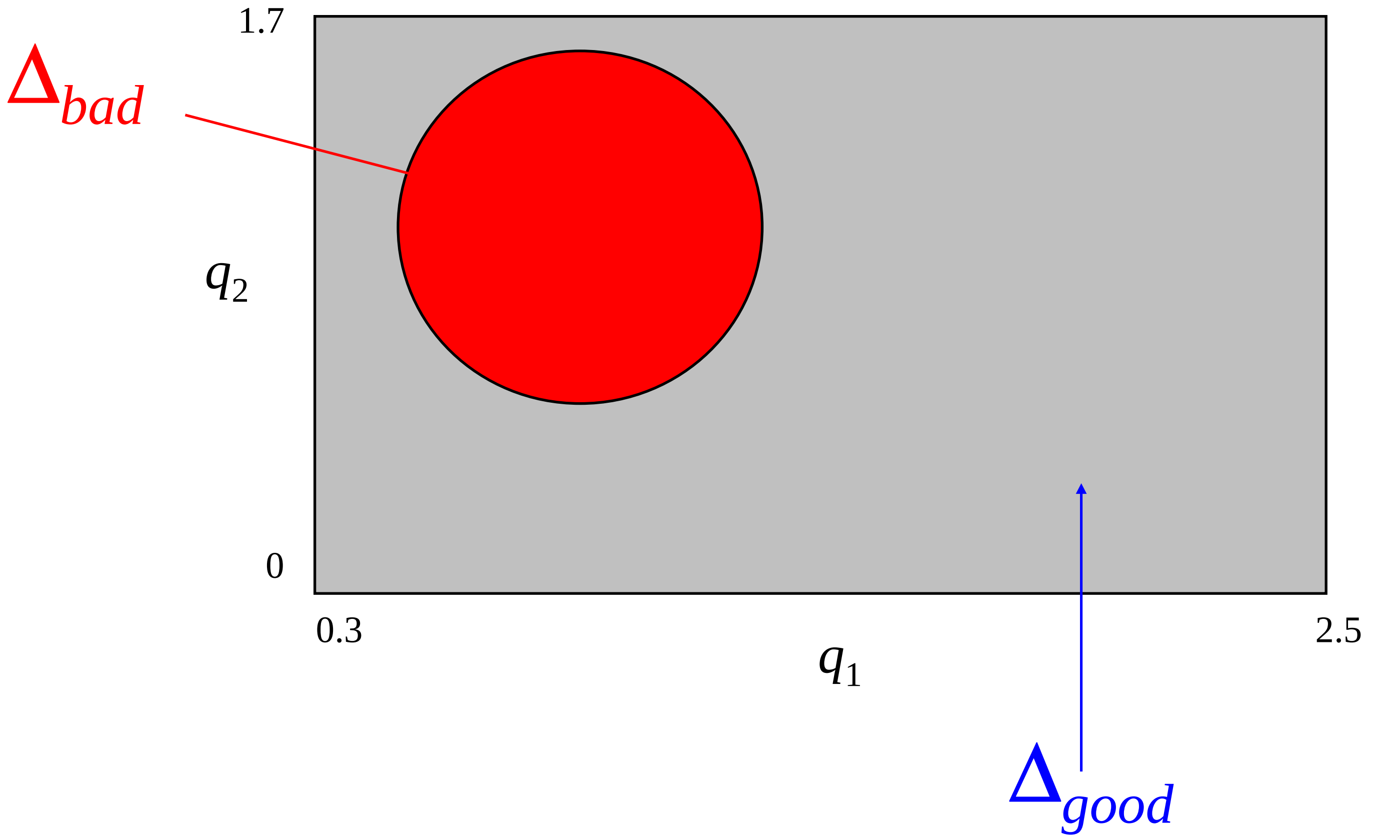
- We define two subsets of $\mathcal{B}_{\mathbb{D}}$

$$\Delta_{good} = \{\Delta : J(\Delta) \leq \gamma\} \subseteq \mathcal{B}_{\mathbb{D}}$$
$$\Delta_{bad} = \{\Delta : J(\Delta) > \gamma\} \subseteq \mathcal{B}_{\mathbb{D}}$$

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

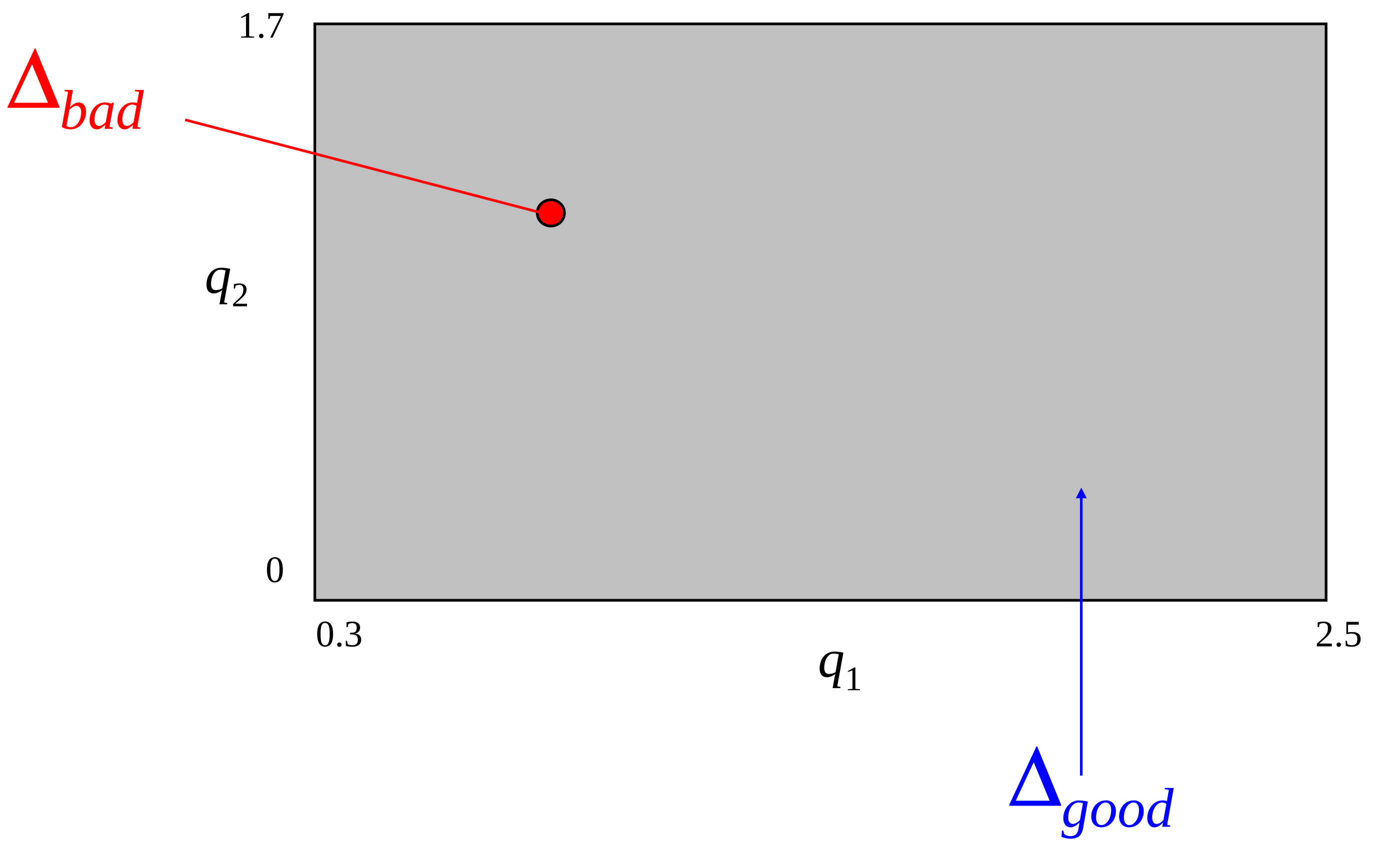
$$vol\left(\Delta_{good}\right) = \int_{\Delta_{good}} d\Delta$$

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

$$\Delta_{good} = \mathcal{B}_{\mathbb{D}}$$

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

$$\frac{vol(\Delta_{good})}{vol(\mathcal{B}_{\mathbb{D}})}$$

is close to one

Probability of Performance[1]

- We define the probability of performance as

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

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

$$p_\gamma = \frac{vol(\Delta_{good})}{vol(\mathcal{B}_{\mathbb{D}})}$$

[1] R.F. Stengel (1980)

Example: Closed-Form Computation

- For Truxal's example, we compute p_γ in closed-form
- For uniform distribution, we have

$$vol(\Delta_{good}) = 3.74 - \pi\, r^2$$

$$vol(\mathcal{B}_{\mathbb{D}}) = 3.74$$

Randomized Algorithms for Analysis

Optimization versus Integration

- Robustness margin computation requires optimization
- Computation of p_γ requires integration
- We are switching from optimization to integration
- Computing multidimensional integrals is a difficult task
- Complexity of multidimensional integration[1]

[1] J.F. Traub, G.W. Wasilkowski and H. Wozniakowski (1988)

Ingredients for RAs

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

$\downarrow$ $\downarrow$

$J = J(\Delta)$ γ

IEIIT-CNR

Randomized Algorithms for Analysis

- Two classes of randomized algorithms for probabilistic robust performance analysis

- P1: Performance verification (compute p_γ)
- P2: Worst-case performance (compute $J_{\max}$)

- Both are based on uncertainty randomization of Δ

- Bounds on the sample size are obtained

IEIIT-CNR

Randomized Algorithms - 2

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

$$\Delta^1, \Delta^2, \ldots, \Delta^N \in \mathcal{B}_{\mathbb{D}}$$

according to the density f_Δ

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

IEIIT-CNR

Empirical Probability

- Construct an indicator function

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

- An estimate of p_γ is the empirical probability

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

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

A Reliable Estimate

- The empirical probability is a reliable estimate if

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

- Find the minimum N such that

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

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

Chernoff Bound[1]

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

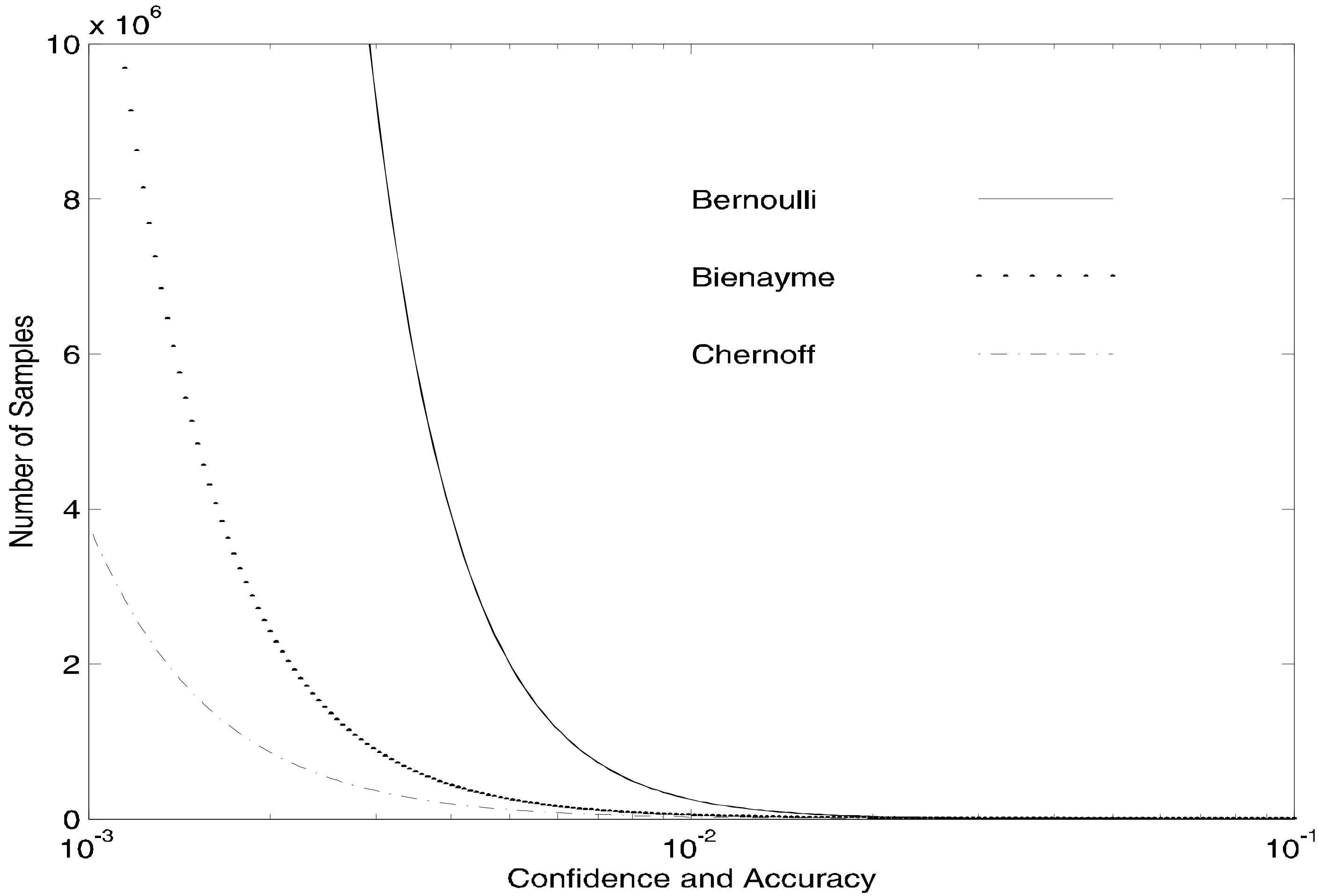
$$N \geq \frac{\log \frac{2}{\delta}}{2\varepsilon^2}$$

then

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

[1] H. Chernoff (1952)

Comparison Between Bounds



IEIIT-CNR

Chernoff Bound

- Remark: Chernoff bound improves upon other bounds such as Bernoulli (Law of Large Numbers)
- Dependence on $1/\delta$ is logarithmic
- Dependence on $1/\varepsilon$ is quadratic

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

IEIIT-CNR

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 associated with the evaluation of $J(\Delta^i)$ for fixed Δ^i is polynomial-time
3. Sample size is polynomial in the problem size and probabilistic levels ε and δ

IEIIT-CNR

1. Random Sample Generation

- Random number generation (RNG): Linear and nonlinear methods for uniform generation in [0,1) such as Fibonacci, feedback shift register, BBS, MT, …
- Non-uniform univariate random variables: Suitable functional transformations (e.g., the inversion method)
- The problem is much harder: Multivariate generation of samples of Δ with pdf $f_\Delta(\Delta)$ and support $\mathcal{B}_{\mathbb{D}}$
- It can be resolved in polynomial-time

2. 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

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

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

3. Bounds on the Sample Size

- Chernoff bound is independent on the size of $\mathcal{B}_{\mathbb{D}}$, on the structure $\mathbb{D}$ on the number of blocks, on the pdf $f_{\Delta}(\Delta)$
- It depends only on δ and ε
- Same comments can be made for other bounds such as Bernoulli

Worst-Case Performance

- Recall that

$$J_{\max} = \max_{\Delta \in \mathcal{B}_{\mathbb{D}}} J(\Delta)$$

- Generate N i.i.d. samples

$$\Delta^1, \Delta^2, \ldots, \Delta^N \in \mathcal{B}_{\mathbb{D}}$$

according to the density f_Δ

- Compute

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

IEIIT-CNR

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 and Comments

- Number of samples is much smaller than Chernoff
- Bound is a specific instance of the fpras (fully polynomial randomized approximated scheme) theory
- 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%
$1-\delta$	99.9%	99.5%	99.9%	99.5%	99.99%	99.999%
N	$6.91 \cdot 10^3$	$5.30 \cdot 10^3$	$1.38 \cdot 10^3$	$1.06 \cdot 10^3$	$9.21 \cdot 10^4$	$1.16 \cdot 10^6$

Volumetric Interpretation

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

$$\Pr\left\{J(\Delta) > \hat{J}_N\right\} = \frac{vol\left(\Delta_{bad}\right)}{vol\left(\mathcal{B}_{\mathbb{D}}\right)}$$

- Therefore

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

is equivalent to

$$\Pr\left\{vol\left(\Delta_{bad}\right) \le \varepsilon\, vol\left(\mathcal{B}_{\mathbb{D}}\right)\right\} \ge 1-\delta$$

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

$$\Pr\left\{ p_L \leq p_\gamma \leq p_U \right\} = 1 - \delta$$

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

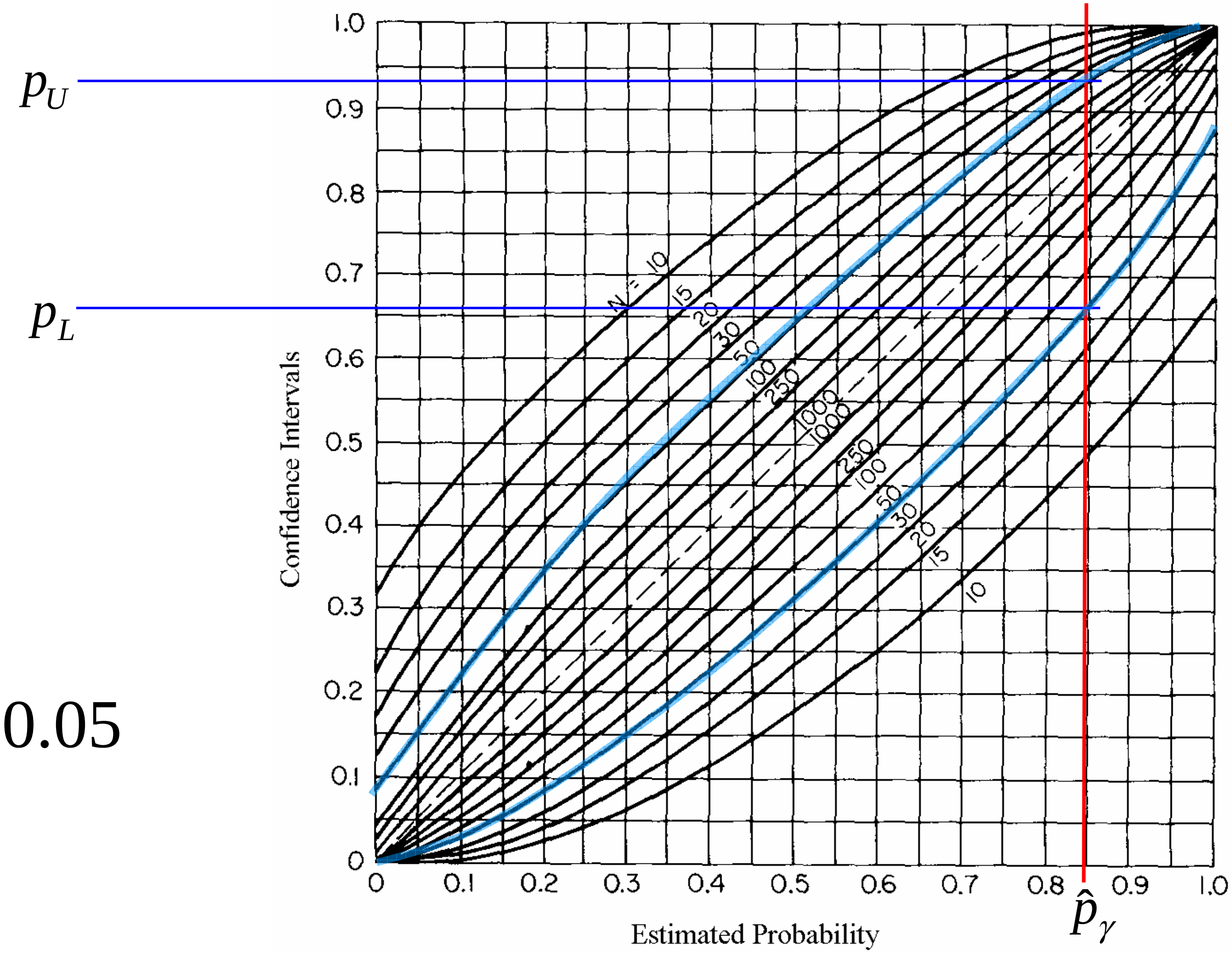
$$\sum_{k=N_{good}}^{N} \binom{N}{k} p_L^k (1-p_L)^{N-k} = \delta_L$$

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

with $\delta_L + \delta_U = \delta$

Confidence Intervals - 3

IEIIT-CNR



$\delta = 0.05$

Computational Learning Theory

- The Chernoff Bound studies the problem

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

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

- Performance function J is fixed
- Computational Learning Theory computes bounds on the sample size for the problem

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

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

VC and P-dimension[1,2]

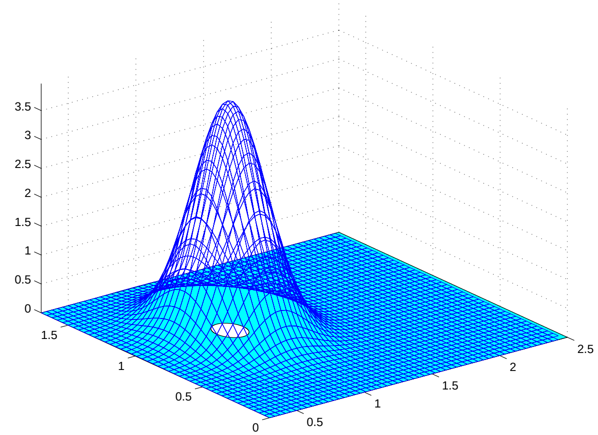
- Computation Learning Theory aims at studying uniform Law of Large Numbers
- The bounds obtained depend on quantities called VC-dimension (if J is a binary valued function), or P-dimension (if J is a continuous valued function)
- VC and P-dimension are measures of the problem complexity
- The bounds obtained are very conservative

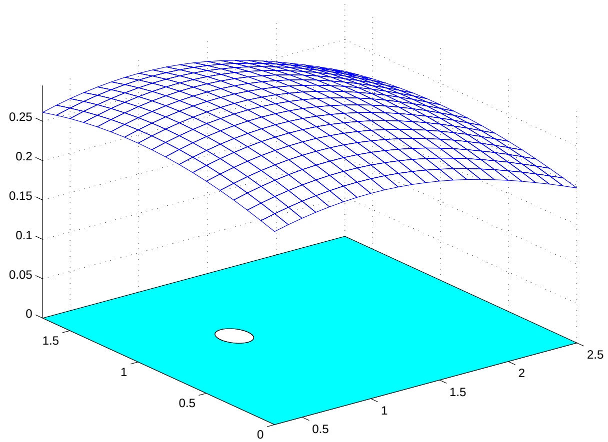
[1] M. Vidyasagar (1997)

[2] E.D. Sontag (1998)

Choice of the Distribution - 1

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





IEIIT-CNR

Choice of the Distribution - 2

- The bounds discussed are independent on the choice of the distribution but for computing $\Pr\{J(\Delta) \leq \gamma\}$ we need to know the distribution $f_\Delta(\Delta)$
- 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 - 3

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

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

Probabilistic Robust Synthesis

IEIIT-CNR

Analysis vs Design with Uncertainty

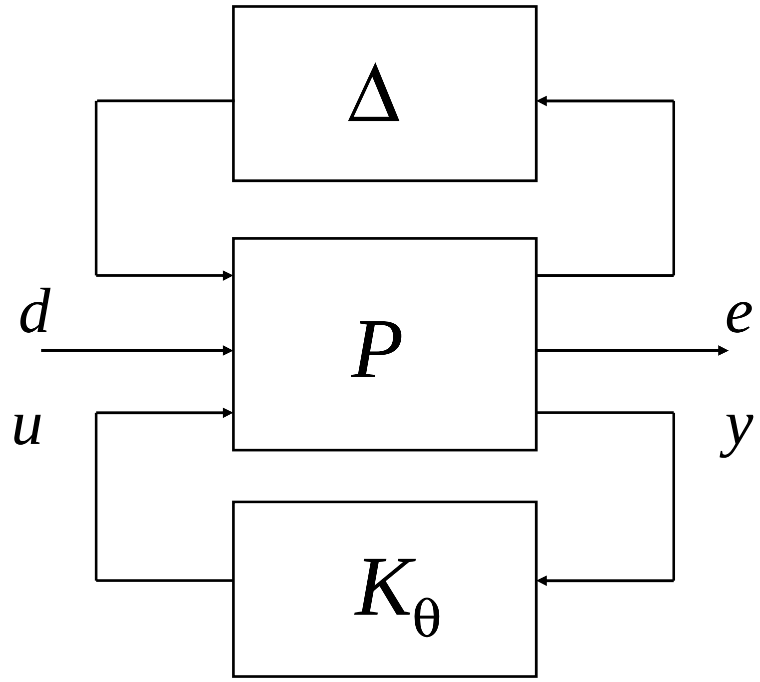
- Starting point: Worst-case analysis versus design
- Consider an interval family $p(s,q)$, $q \in \mathcal{B}_q = \{q \in \mathbb{R}^n, \|q\|_\infty \leq 1\}$
- Analysis problem:
  - Check if $p(s,q)$ is stable for all $q \in \mathcal{B}_q$

    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

Synthesis Paradigm



- Design the parameterized controller K_θ to guarantee stability and performance

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 their bounding set

Randomized Algorithms for Synthesis

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

- Based on expected value minimization
- Utilization of computational learning theory results
- Very general problems can be handled
- Sample complexity bounds are very conservative and controller randomization is required

[1] M. Vidyasagar (1998)

Randomized Algorithms for Synthesis

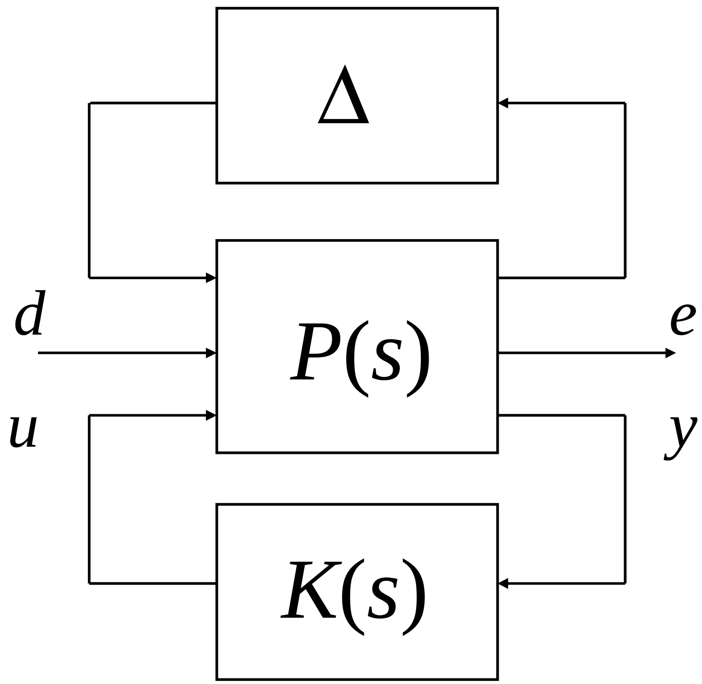
- Robust performance synthesis[1]

- Problem reformulation as robust feasibility
- Only control convex problems can be handled
- Finite-time convergence with probability one is obtained

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

Robust Performance Synthesis

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



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

- 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 $0 \leq \gamma \leq 1$,

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

for all $\Delta \in \mathcal{B}_{\mathbb{D}}$

- Then, the cost

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

is guaranteed for all $\Delta \in \mathcal{B}_{\mathbb{D}}$

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

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

for all vertex matrices A^i

IEIIT-CNR

Probabilistic Quadratic Stabilization

- Critical problem: The number of LMIs is 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 $\Delta^1,\ldots,\Delta^N \in \mathcal{B}_{\mathbb{D}}$. Then, check if the LMI

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

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

Definition

- 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 $\mathcal{C}$ be the cone of positive semi-definite matrices

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

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 $\Delta^k \in \mathcal{B}_{\mathbb{D}}$
- Then, for guaranteed cost we obtain the QMI

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

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 $\mathcal{B}_{\mathbb{D}}$ 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=$B(\Delta)$, $\Delta \in \mathcal{B}_{\mathbb{D}}$

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)

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]

[1] D. Liberzon and R. Tempo (2004)
[2] H. Ishii, T. Basar and R. Tempo (2005)
[3] S. Kanev, B. De Schutter and M. Verhaegen (2002)
[4] Y. Oishi and H. Kimura (2003)

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

$$\min f(x)$$

$$g(x,\Delta) \leq 0 \text{ for all } \Delta \in \mathcal{B}$$

- 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)

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

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

[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

Probabilistic LPV Systems

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)

IEIIT-CNR

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

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

IEIIT-CNR

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 $\varepsilon > 0$

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

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

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

[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

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

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

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

- Here μ^k is a stepsize and

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

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)

IEIIT-CNR

RAs for 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

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

Applications of Randomized Algorithms

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
- Control design of Mini UAV

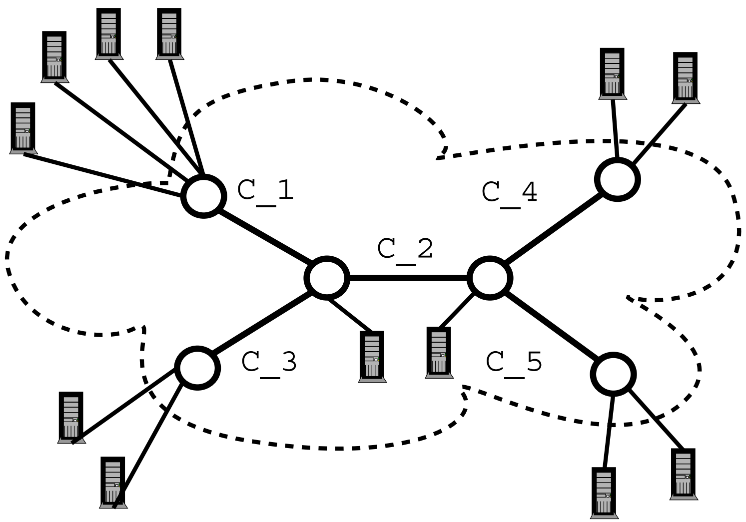
Stability and Performance of High-Speed Networks[1]

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

IEIIT-CNR

Stability and Robustness

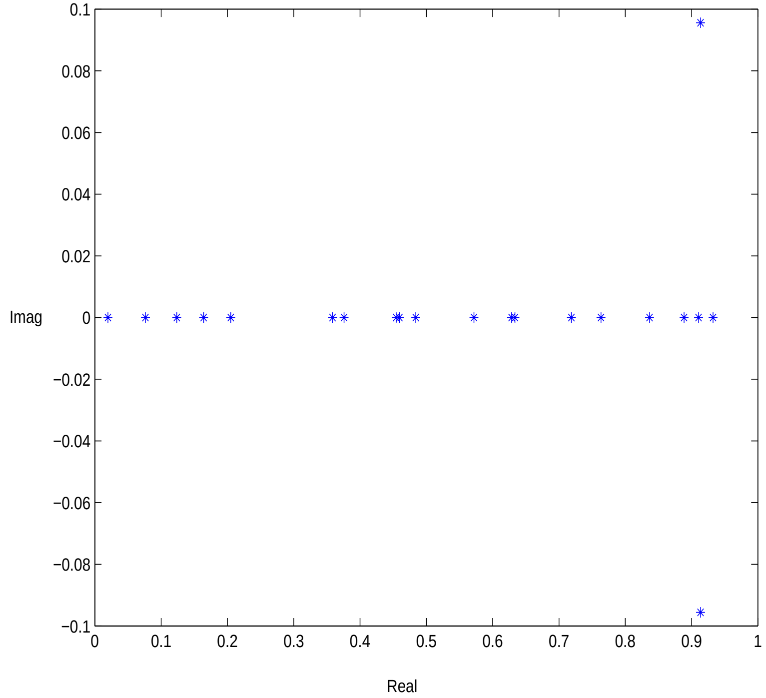
- Network topology
- Source and destination nodes, links (with buffer and capacity)
- Bottleneck link
- Stability and robustness



IEIIT-CNR

Symmetric Single Bottleneck

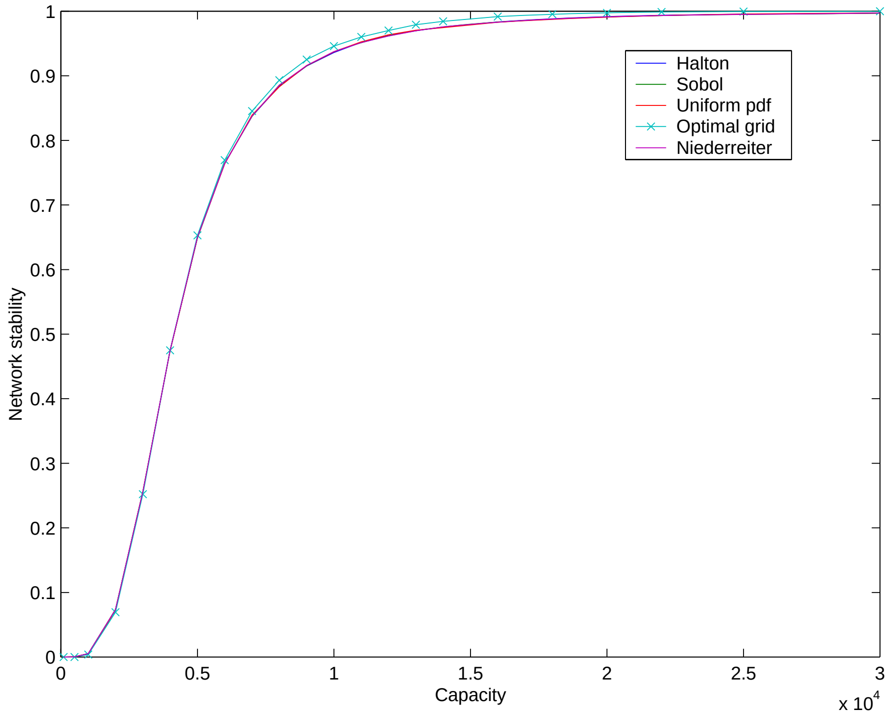
- Parametric stability (discrete time) with real uncertain parameters
- Stability and robustness can be studied in closed form
- Case study with 20 users
- Roots of the closed loop polynomial (discrete-time)



IEIIT-CNR

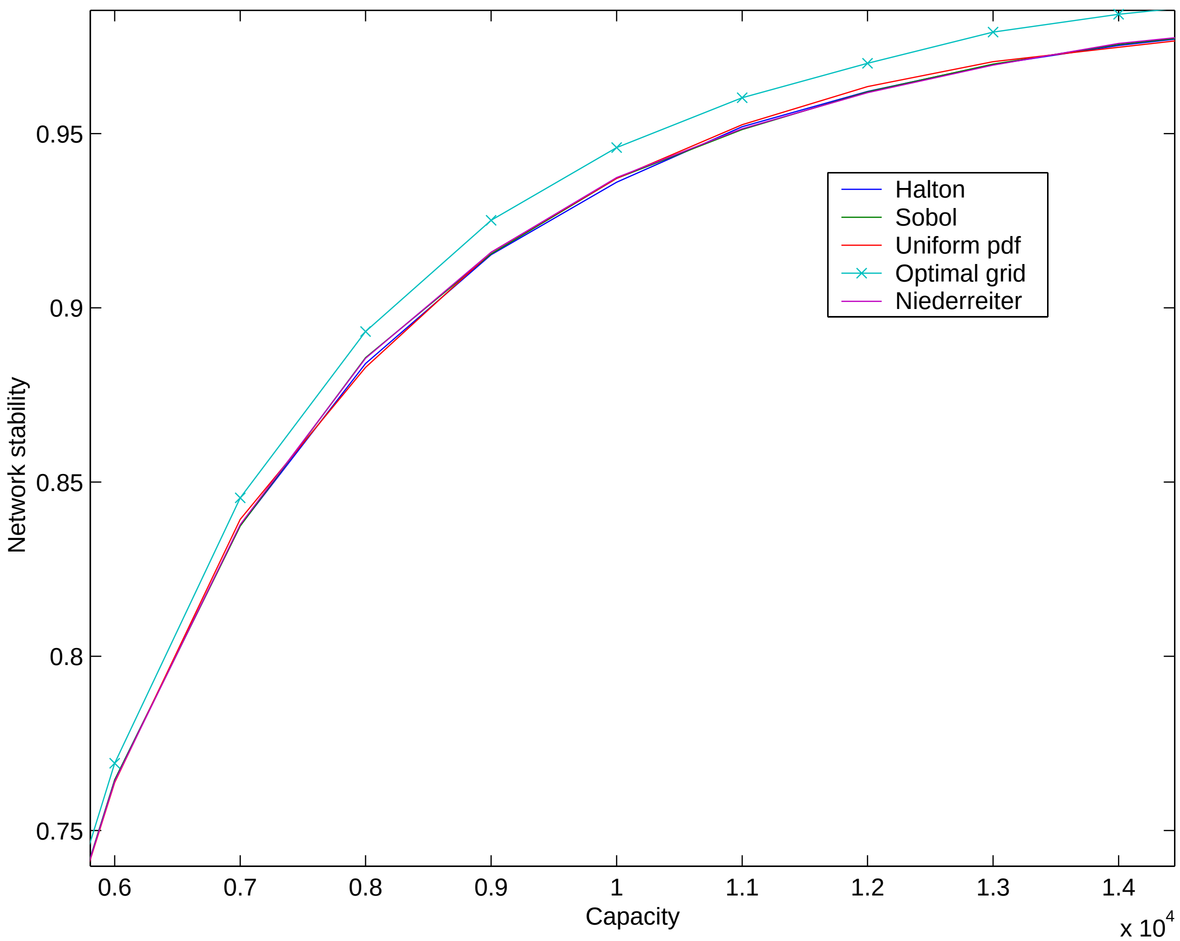
Non-Symmetric Single Bottleneck

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



IEIIT-CNR

Additional Simulations



Mini UAV Control Design

Barbara Pralio and Laura Lorefice

barbara.pralio@polito.it
laura.lorefice@polito.it

Aerospace Engineering Department
Politecnico di Torino

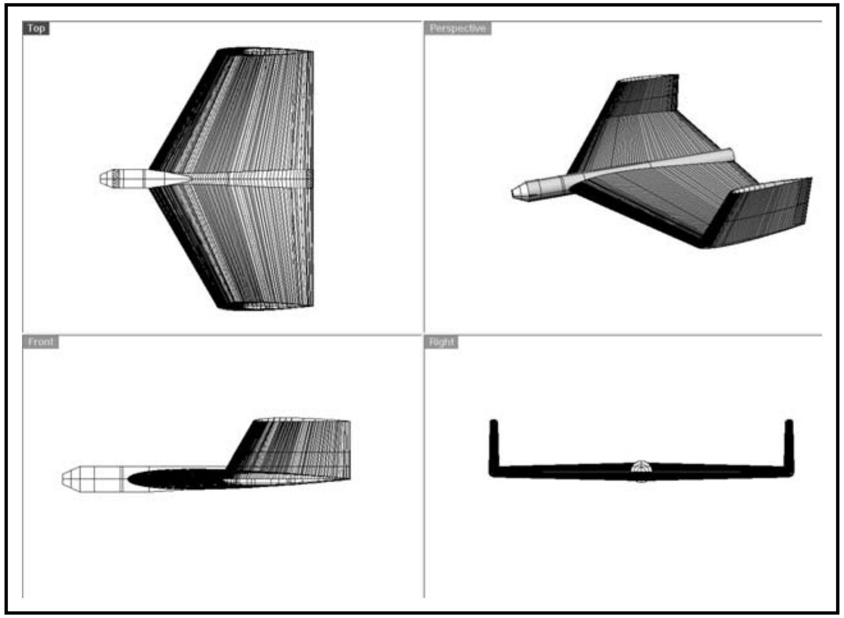
MH1000: Mini UAV Platform

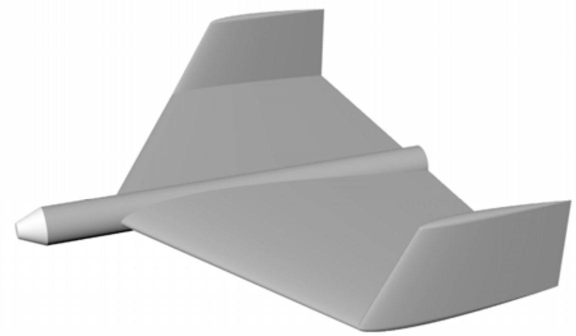
This activity is supported by the Italian Ministry for Research within the National Project:

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

The aerial platform is based on the MicroHawk configuration, developed at the Aerospace Engineering Dept. of Politecnico di Torino (national patent no. TO2003A000702, holder Politecnico di Torino, international request PCT/IB2004/002940)

MH1000: Mini UAV Platform





Design parameter

wind loading W/S = 25 g/dm^2

- wingspan 1000 mm
- maximum take-off weight 1500 g
- DC motor
- remote piloting/autonomous flight
- endurance > 30 min

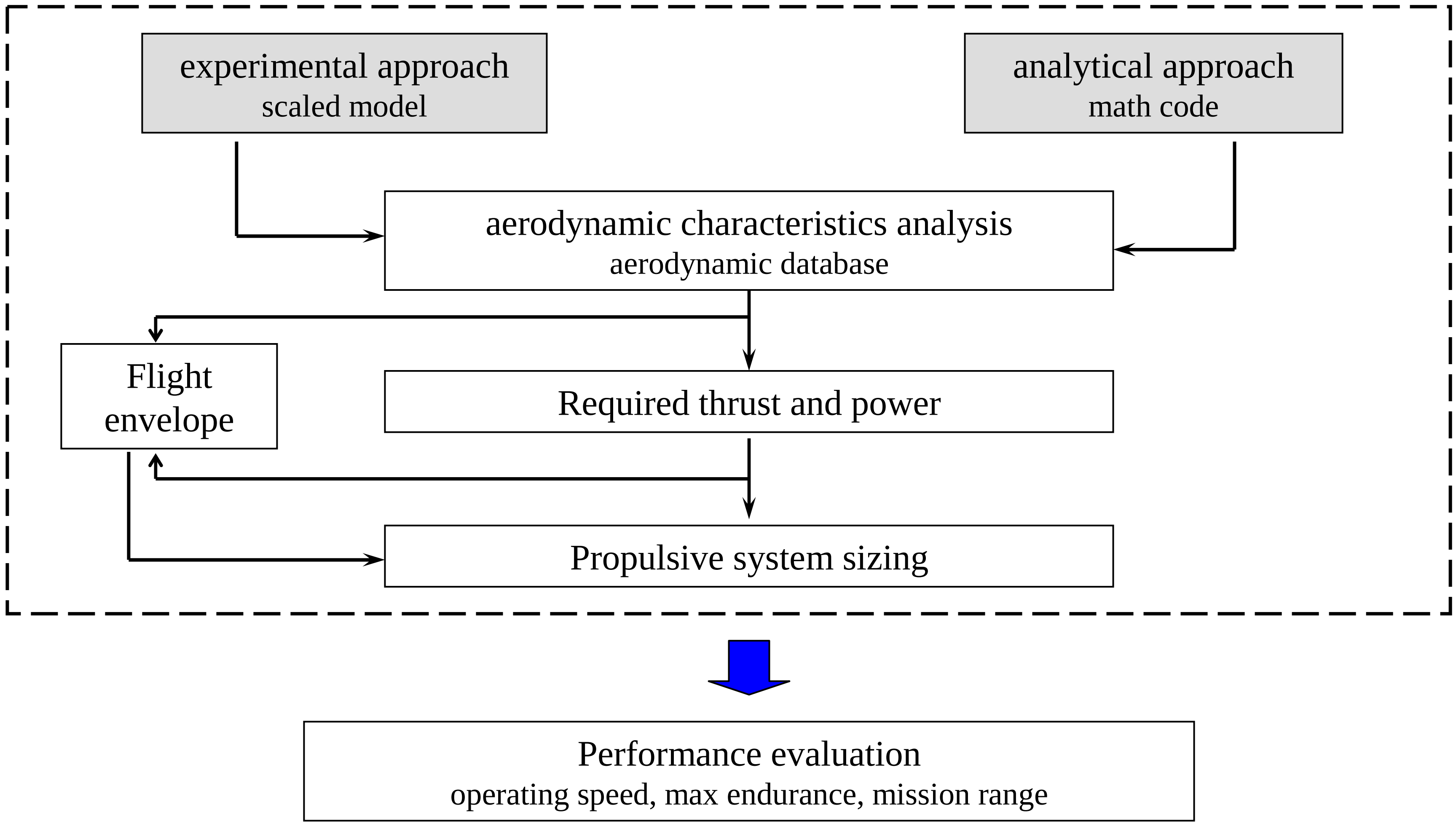
Feasibility Study

- Aerodynamic and performance analysis

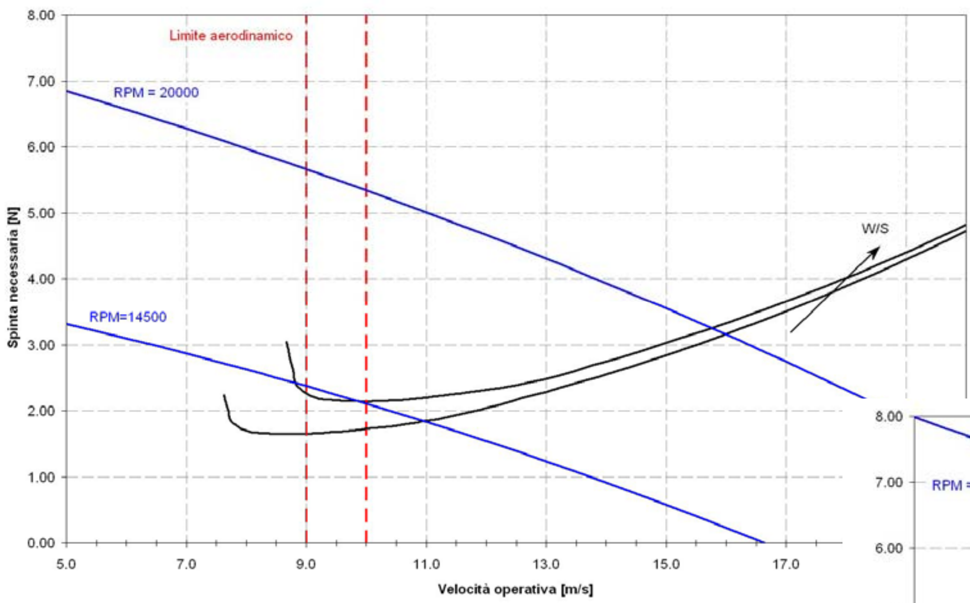
1. Aerodynamic characteristics
2. Flight performance

- Basic on-board system
- Weight distribution

Aerodynamic/mechanical Analysis



Flight Envelope



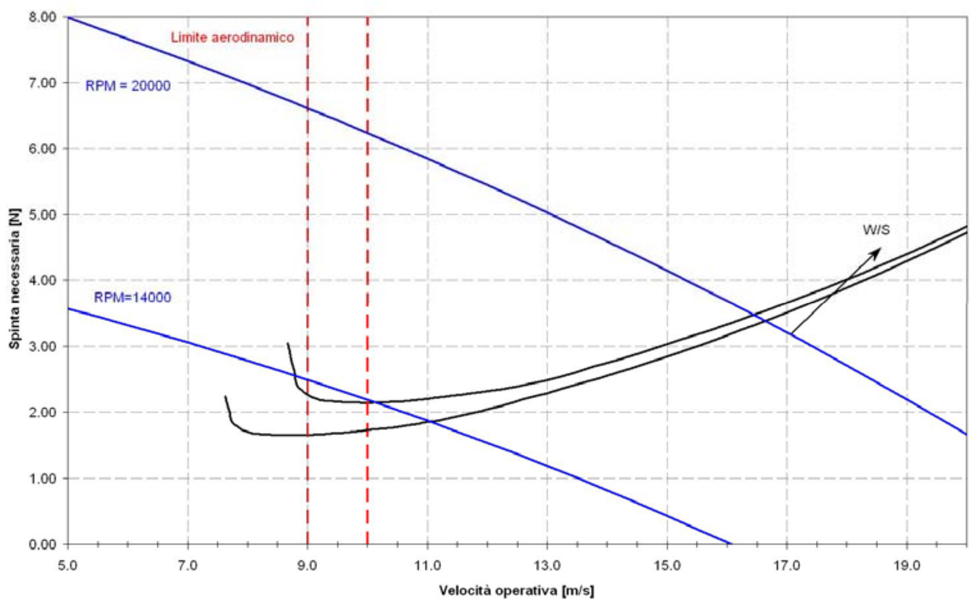
Aerodynamic constraint → minimum flight speed

Propulsive constraint → maximum flight speed

V ≅ 10 ÷ 17 m/s

Wing loading effect → total weight

Propeller sizing effect



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)

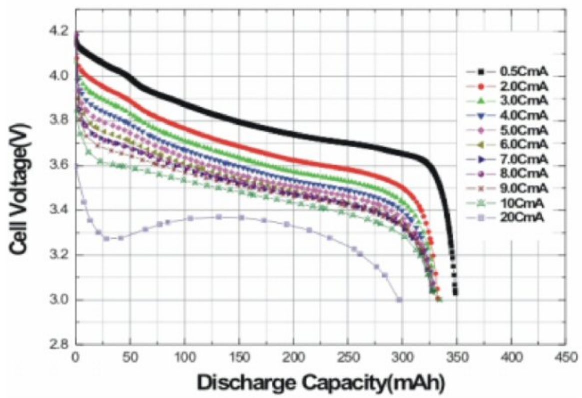
- weight: 160 g
- dimensions: 79 X 42 X 25 mm
- capacity: 2000 mAh

receiver: Schulze Alpha840W

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

servo: Graupner C1081 (2x)

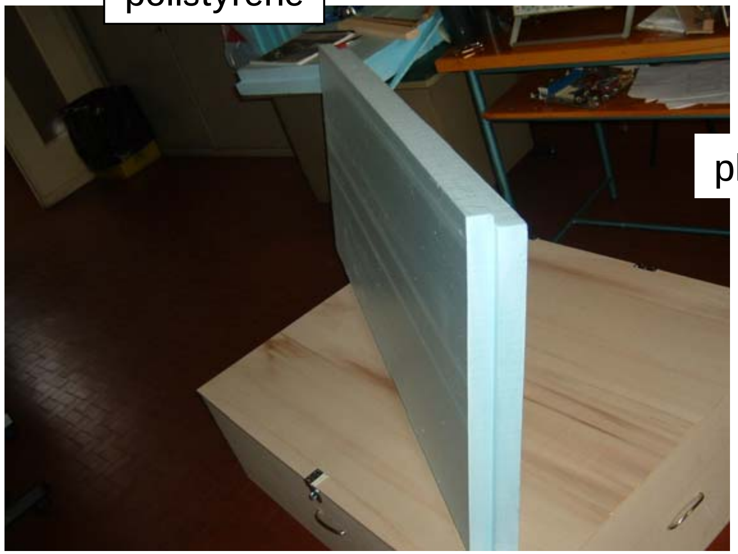
- weight: 13 g
- dimensions: 23 X 9 X 21 mm
- torque: 12 Ncm

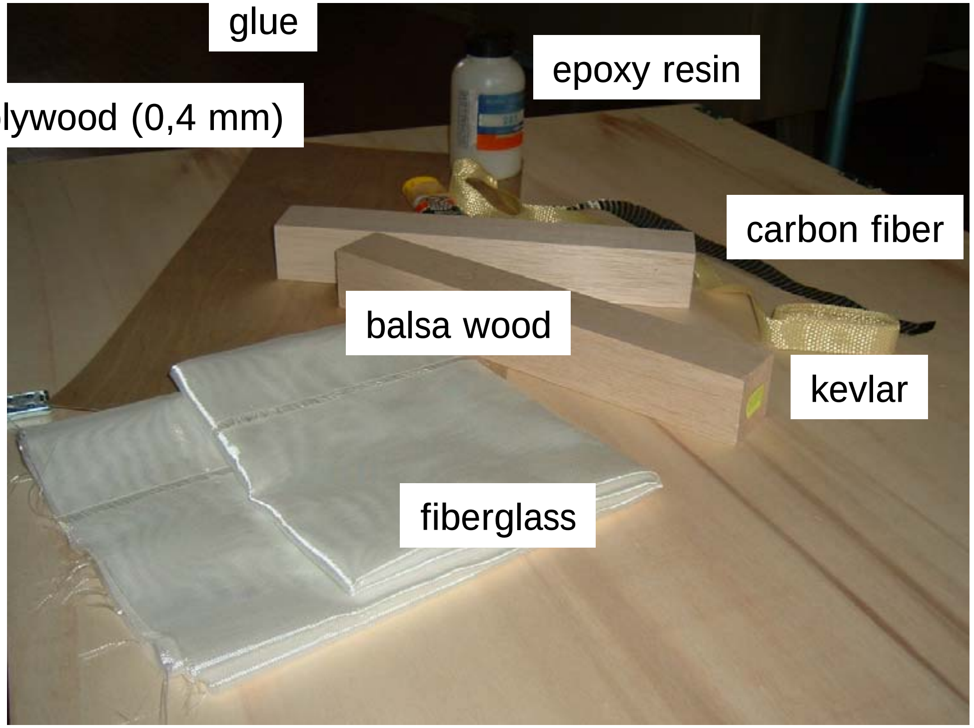


IEIIT-CNR

Prototype Manufacturing - 1

material





IEIIT-CNR

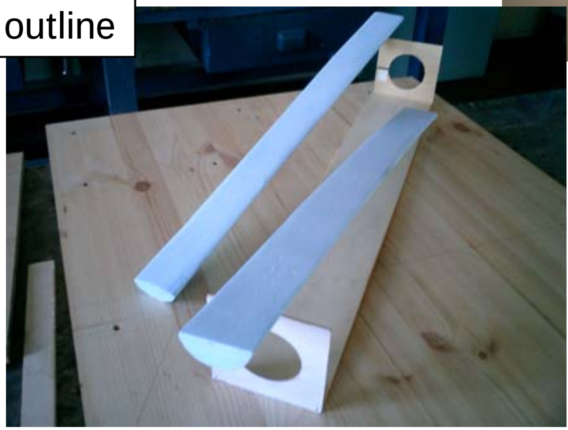
Prototype Manufacturing - 2

hot wire foam cutting machine

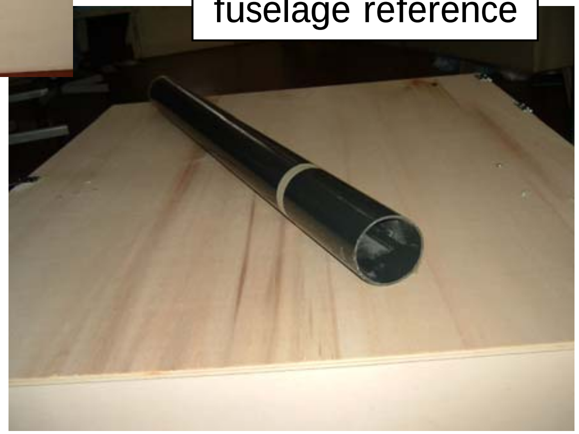
working instruments

lifting surfaces outline

slide outline



fuselage reference

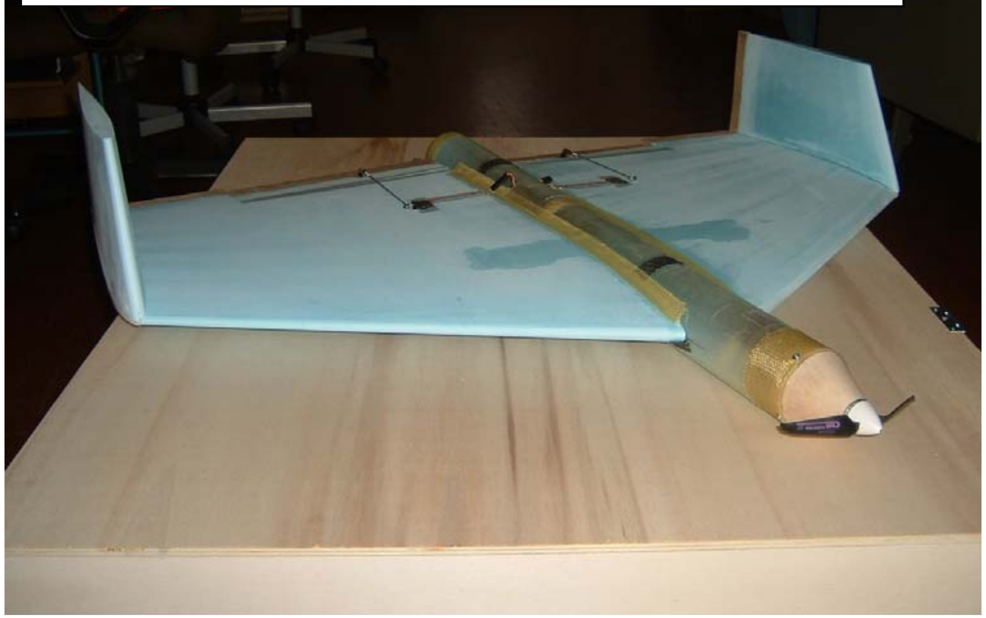


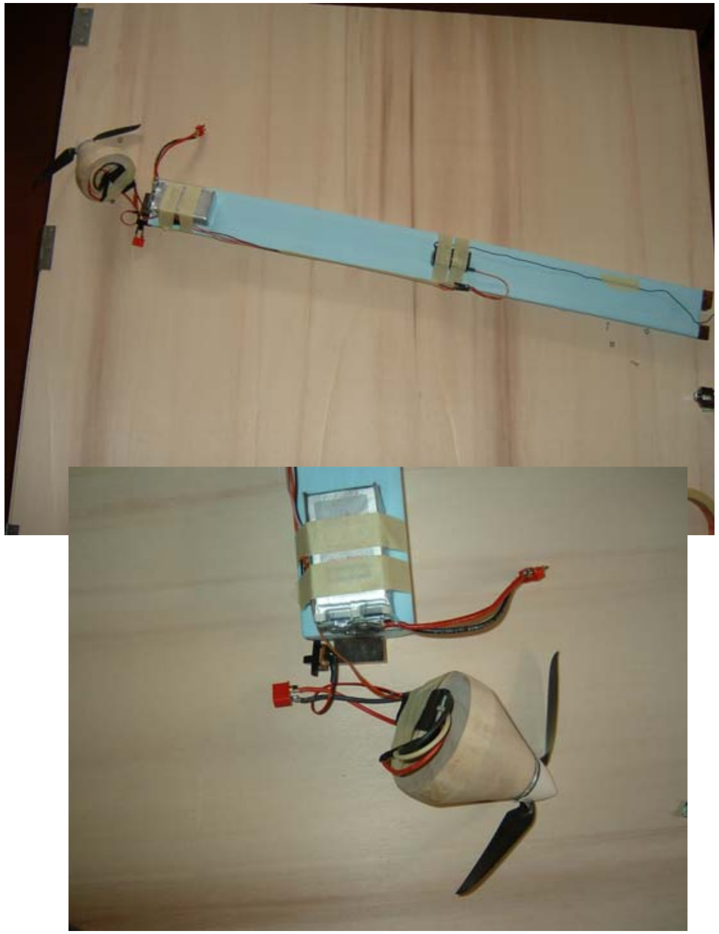
IEIIT-CNR

Prototype Manufacturing - 3

prototype

easy construction
rapid manufacturing
bad model reproducibility
inaccurate geometry

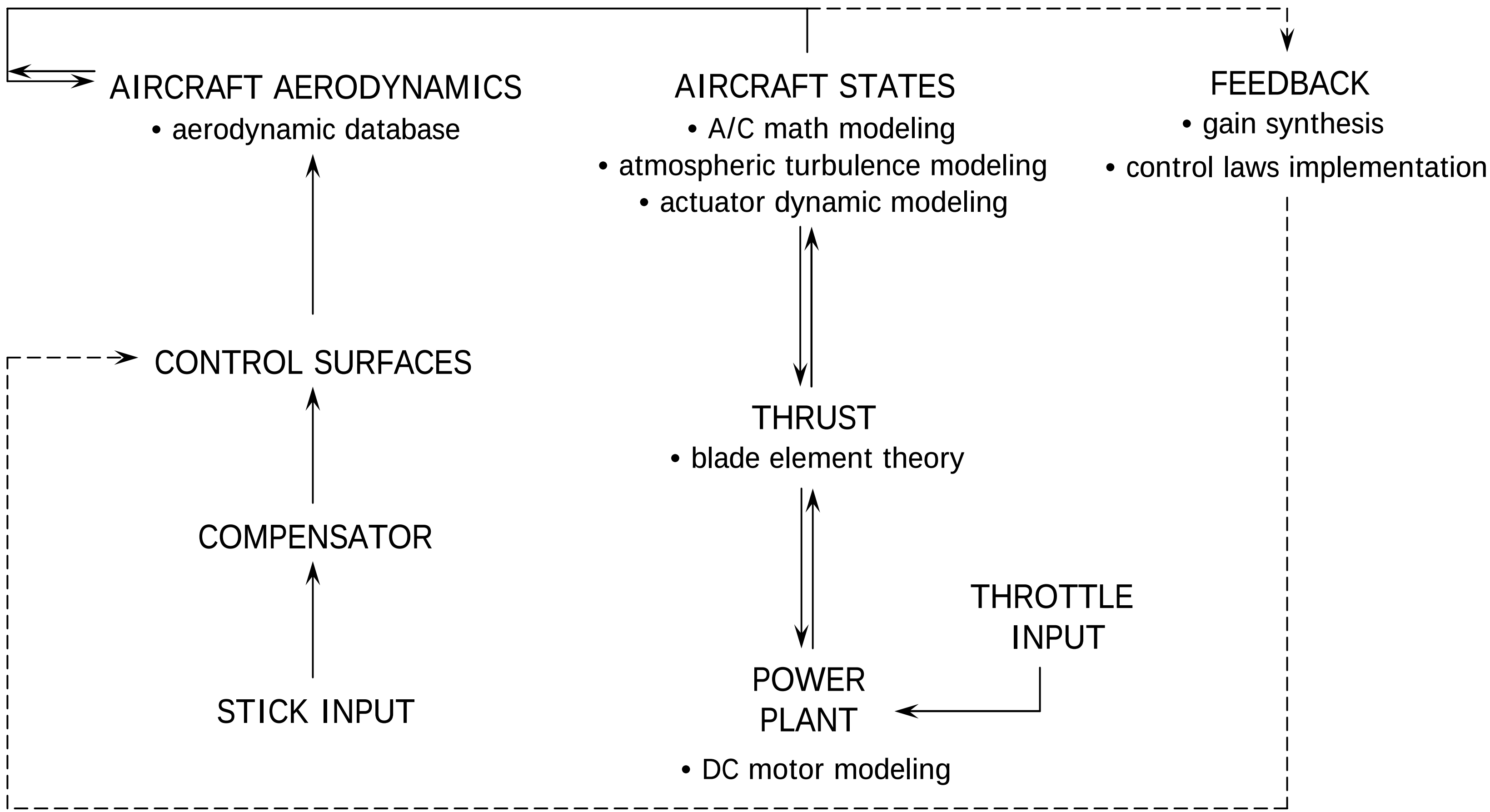




IEIIT-CNR

Aircraft Dynamics - 1

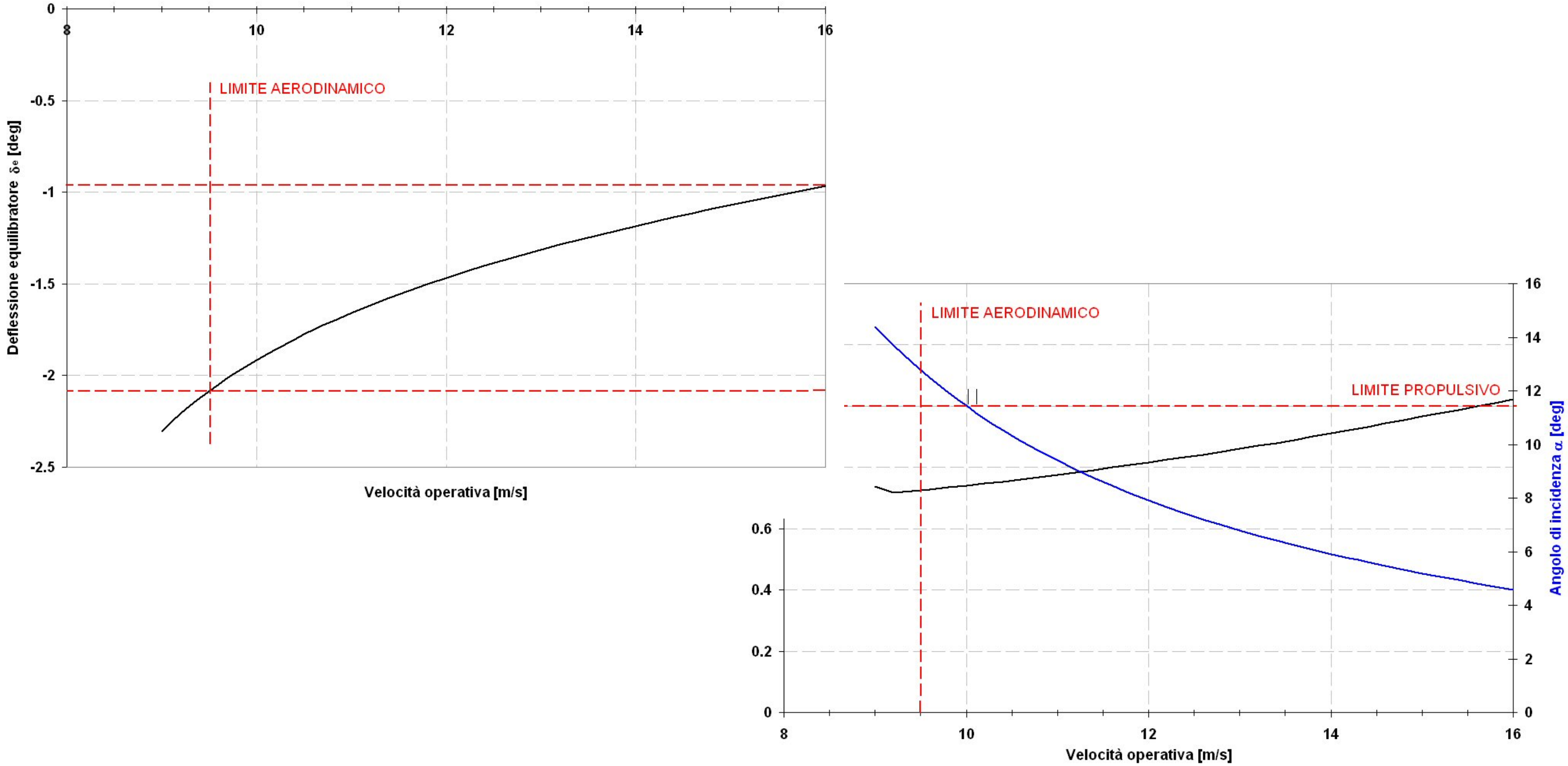
Aircraft math model implementation within an off-line flight simulator



Aircraft Dynamics - 2

Aircraft math model implementation within an off-line flight simulator

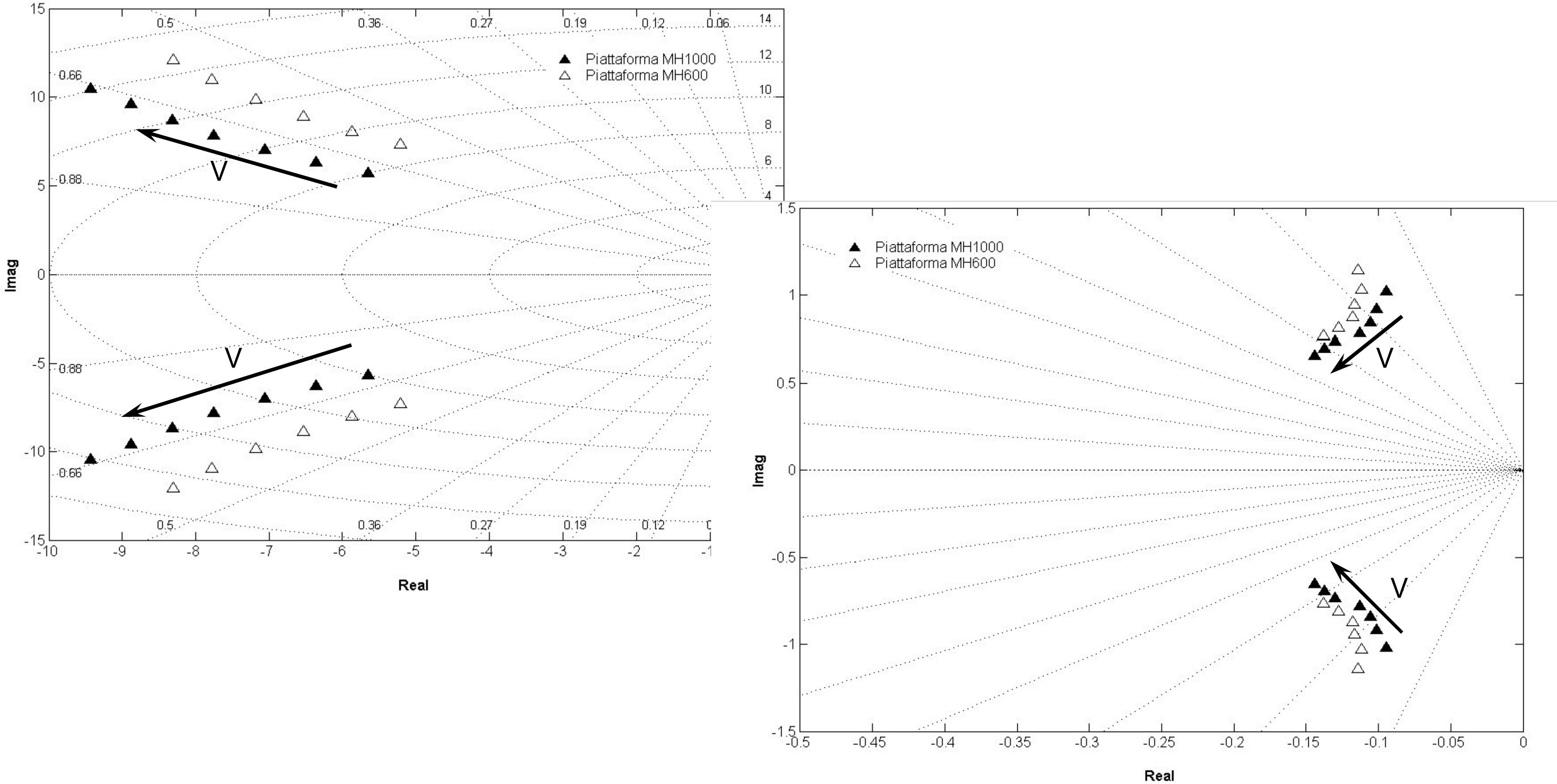
- trimmability analysis



Aircraft Dynamics - 3

Aircraft math model implementation within an off-line flight simulator

- *open loop* dynamics characterization

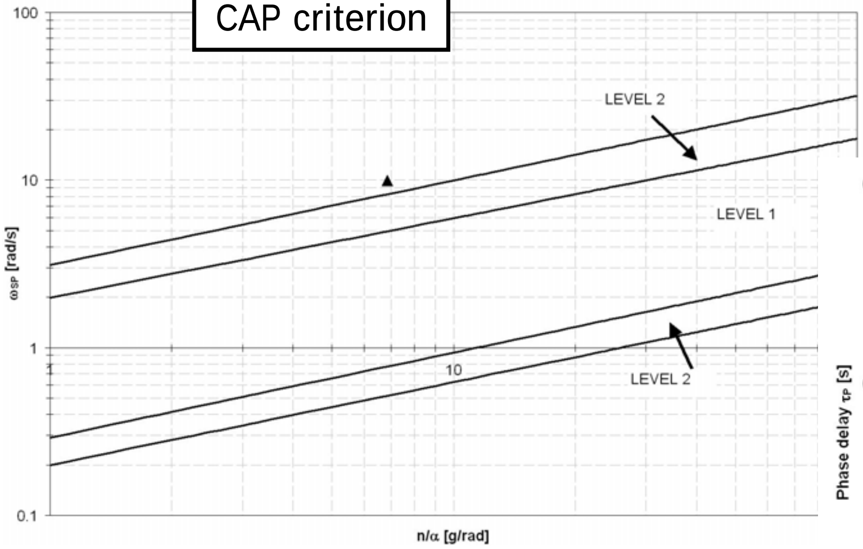


Aircraft Dynamics - 4

Aircraft math model implementation within an off-line flight simulator

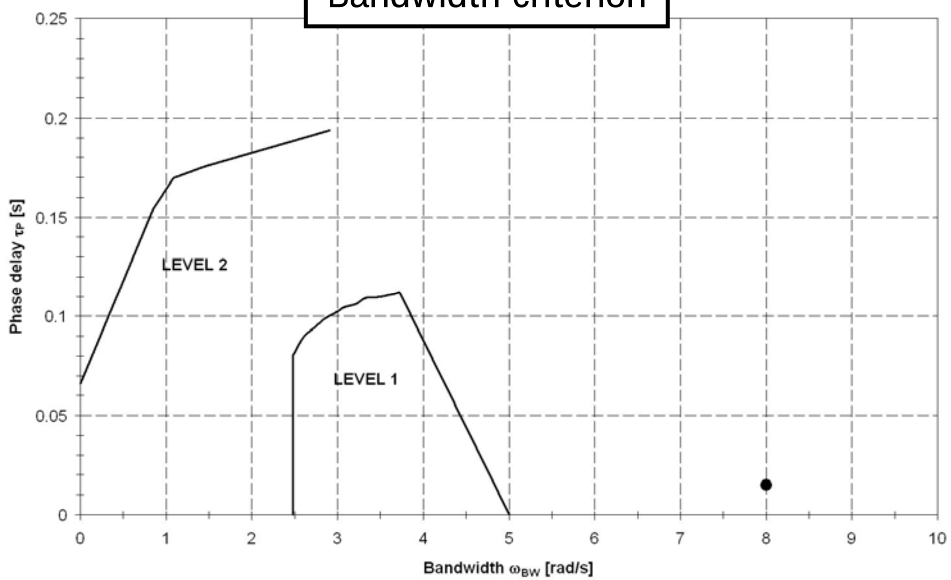
- compliance to standard requirements

CAP criterion



MIL 8785C/MIL 1797A standards

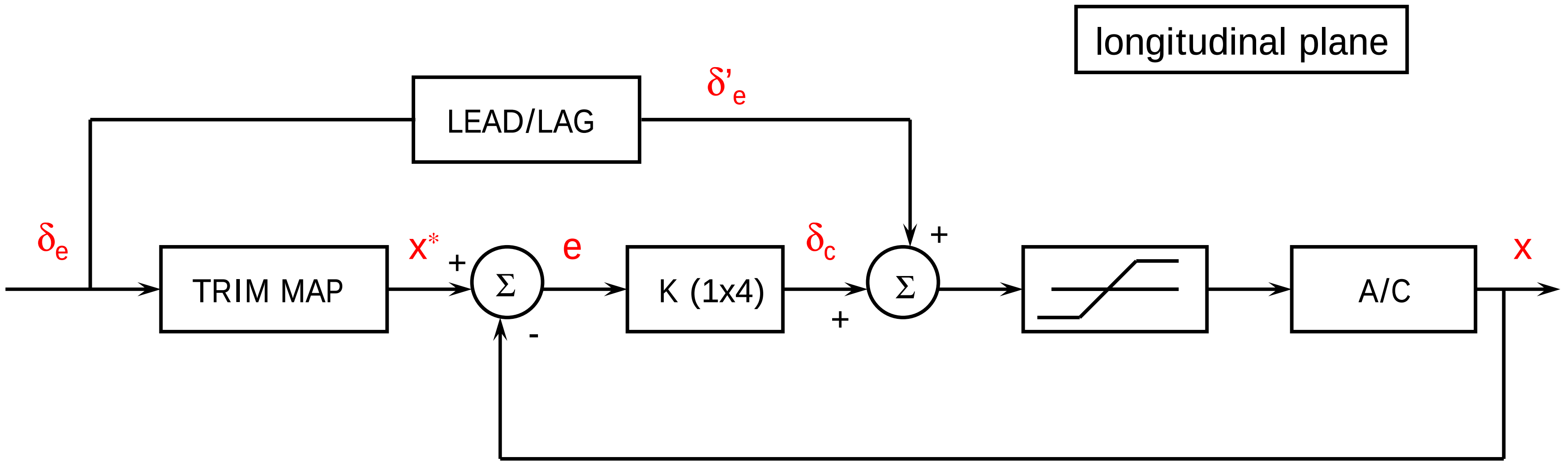
Bandwidth criterion



Controller Design - 1

Flight control system for platform stabilization

➢ definition of controller architecture



X ≡ state vector = {u α q θ}

IEIIT-CNR

Controller Design - 2

Flight control system for platform stabilization

Gain synthesis

- ➢ Randomized algorithms

- ➢ Enlarged stability concept – bounds
  - ✓ stable eigenvalues → $Re(\lambda) < 0$
  - ✓ complex conjugate eigenvalues → $Im(\lambda) \neq 0$
  - ✓ short period undamped frequency → $4 < \omega_{sp} < 6$
  - ✓ short period damping → $0.5 < \zeta_{sp} < 0.9$
  - ✓ phugoid mode undamped frequency → $1 < \omega_{ph} < 1.5$
  - ✓ phugoid mode damping → $0.1 < \zeta_{ph} < 0.15$
  - ✓ $\Delta\omega < \pm 15\%\ \omega_{min}, \omega_{max}$

standard requirements

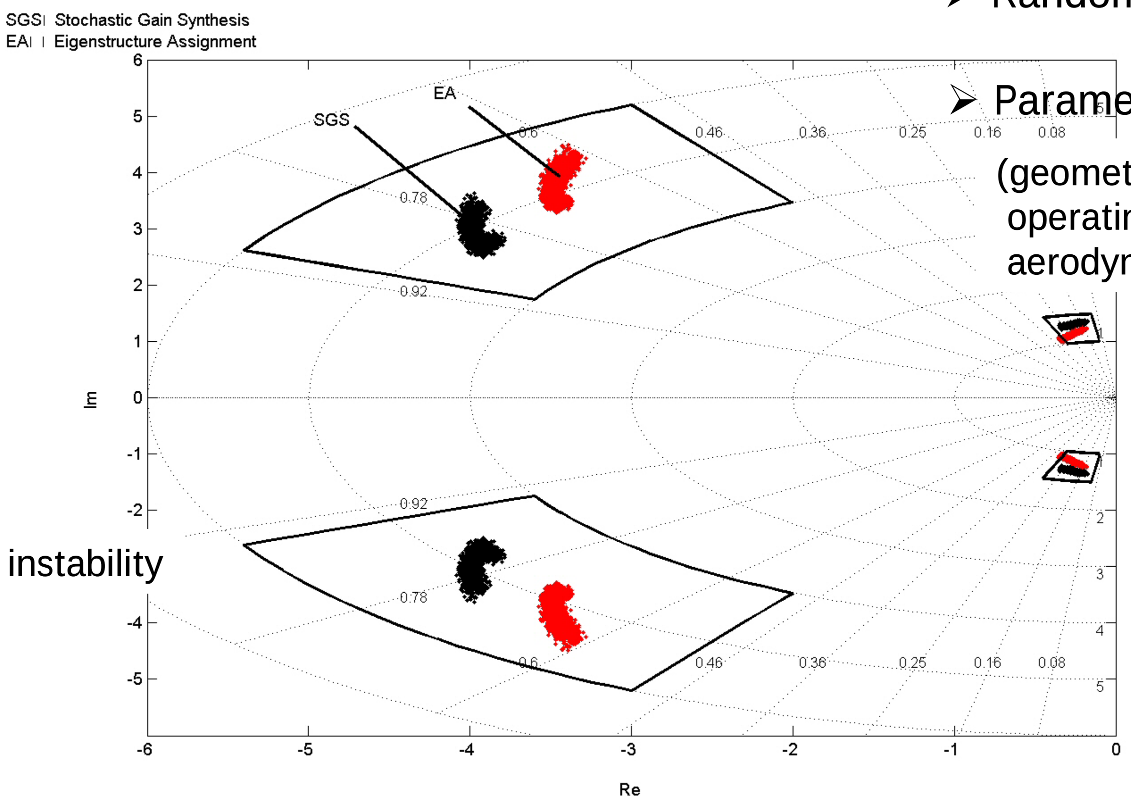
dynamics analysis

IEIIT-CNR

Controller Design - 3

Flight control system for platform stabilization

stability robustness analysis



- Randomized algorithms
- Parameter uncertainties

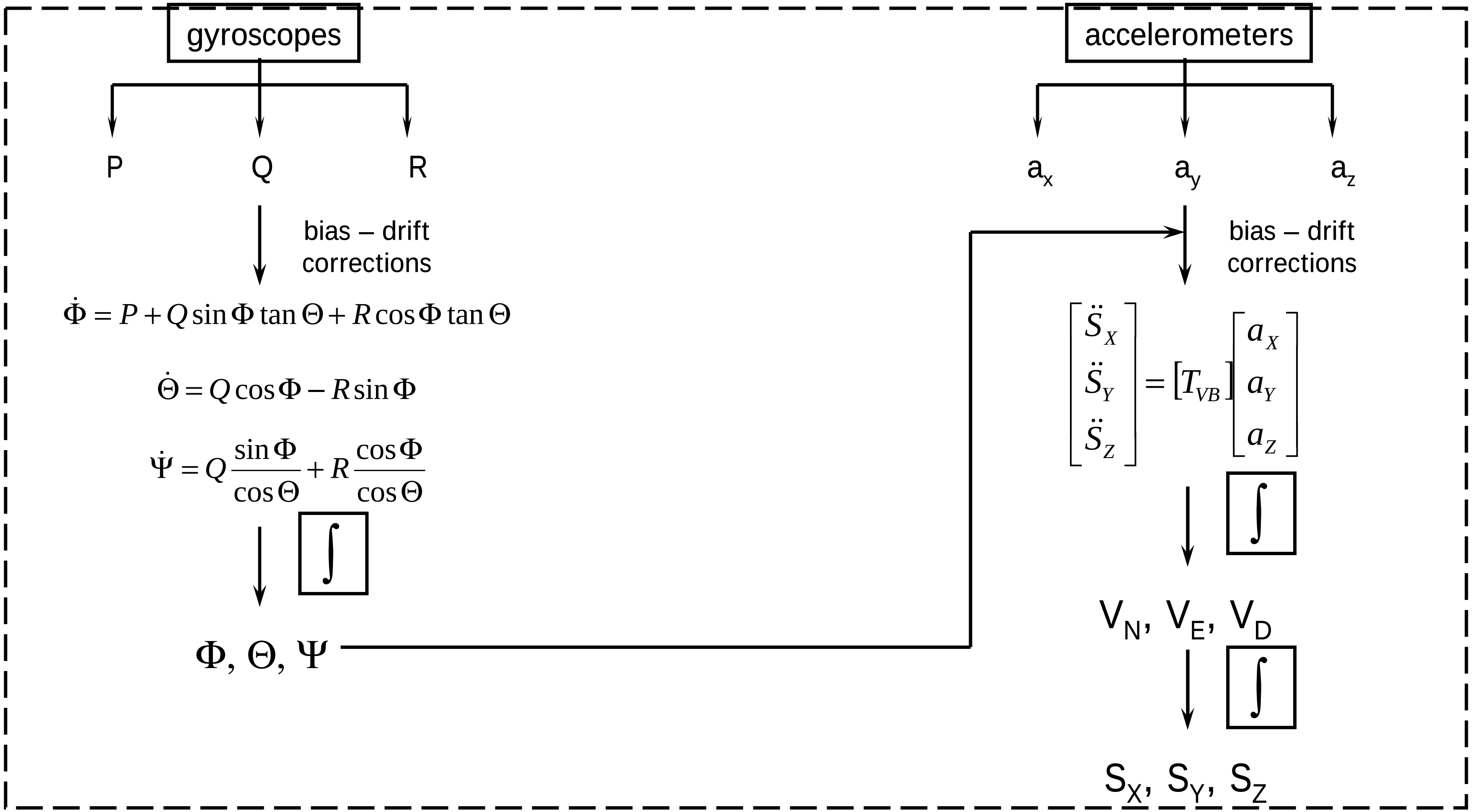
(geometric and inertial data
operating conditions
aerodynamic database)

Probability of instability

IEIIT-CNR

Controller Design - 4

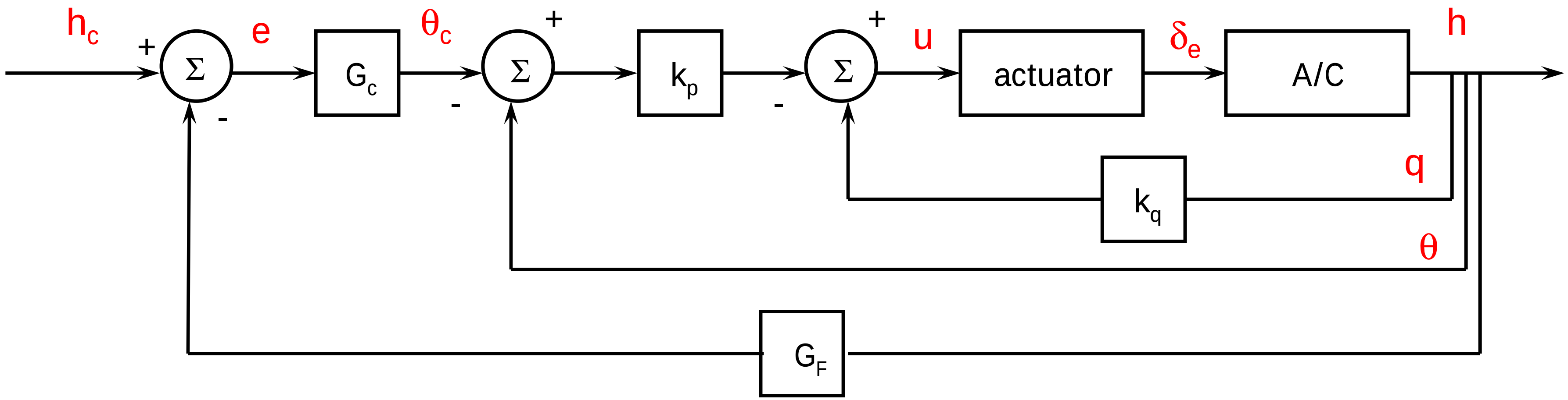
Autopilot



Controller Design - 5

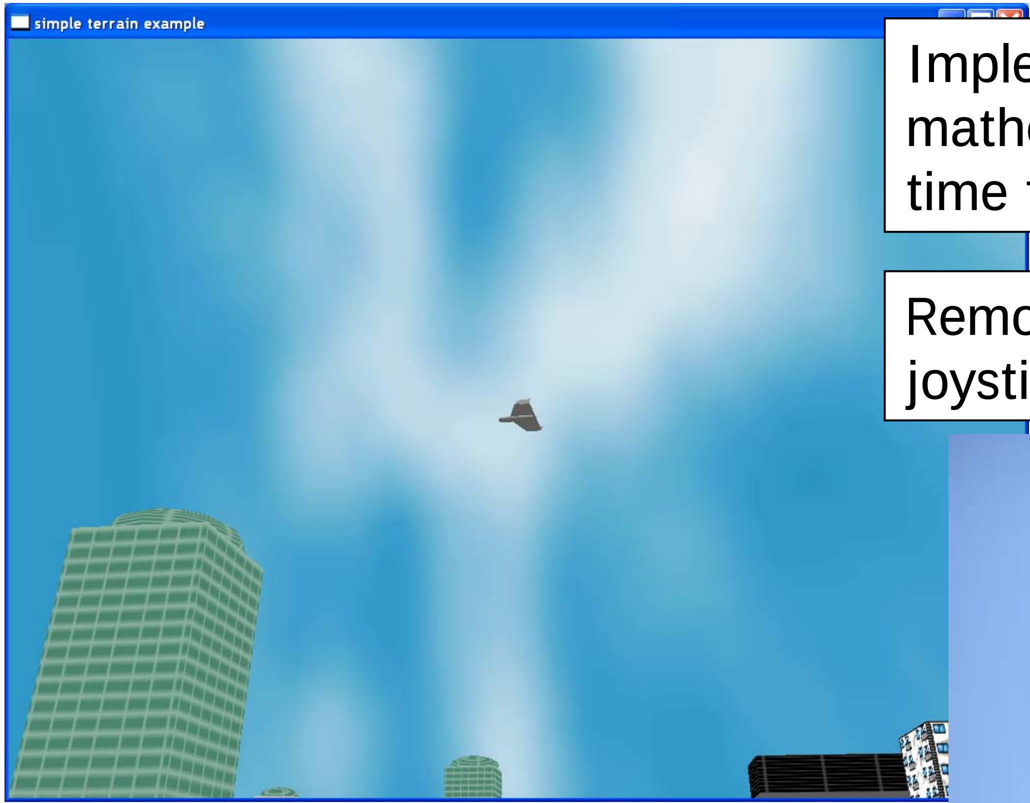
Autopilot

altitude hold



IEIIT-CNR

Flight Simulator



Implementation of the platform mathematical model within a real-time flight simulator

Remote piloting by means of joystick and RC transmitter

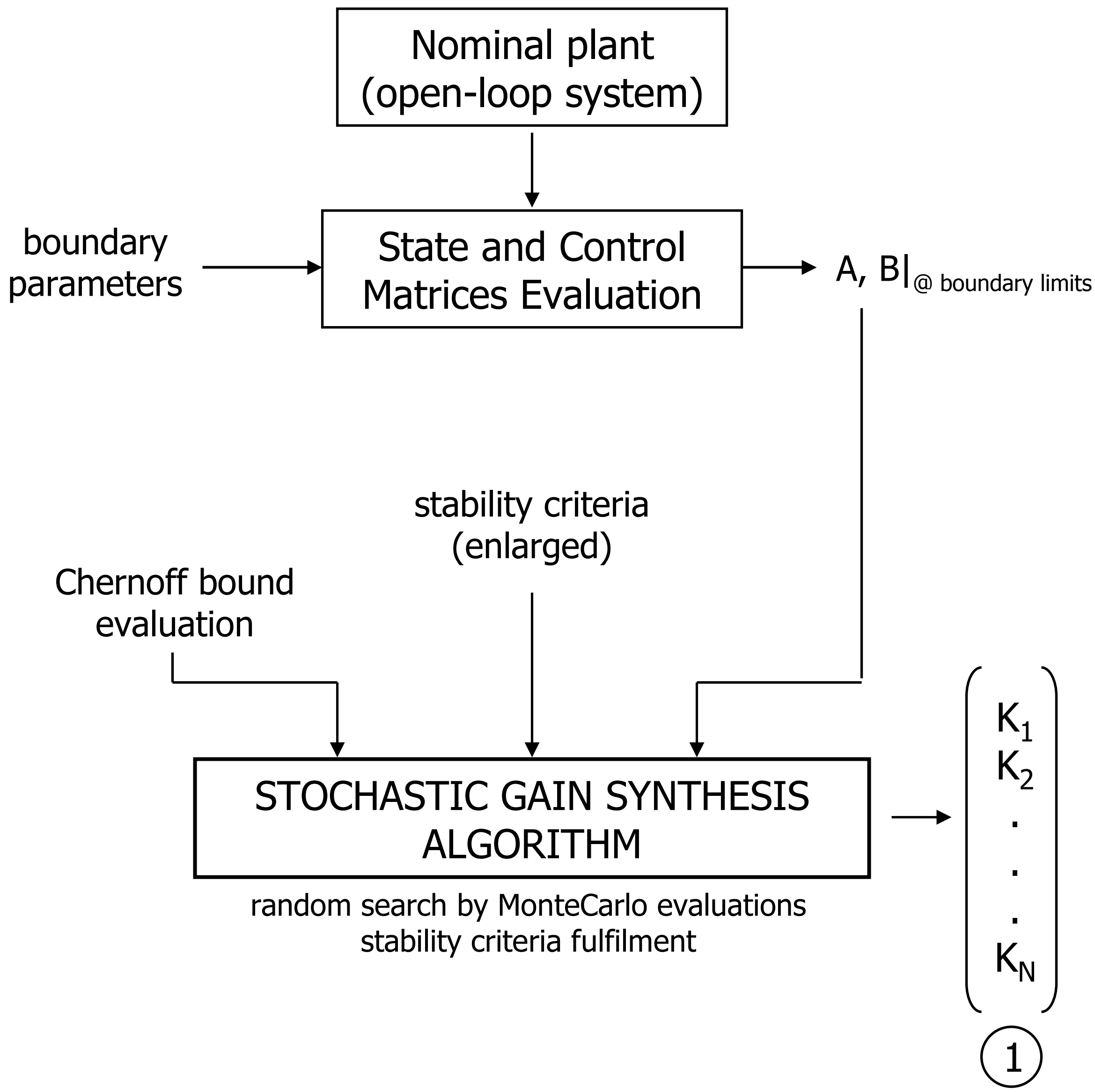
RAs for Mini UAV

- Three main phases

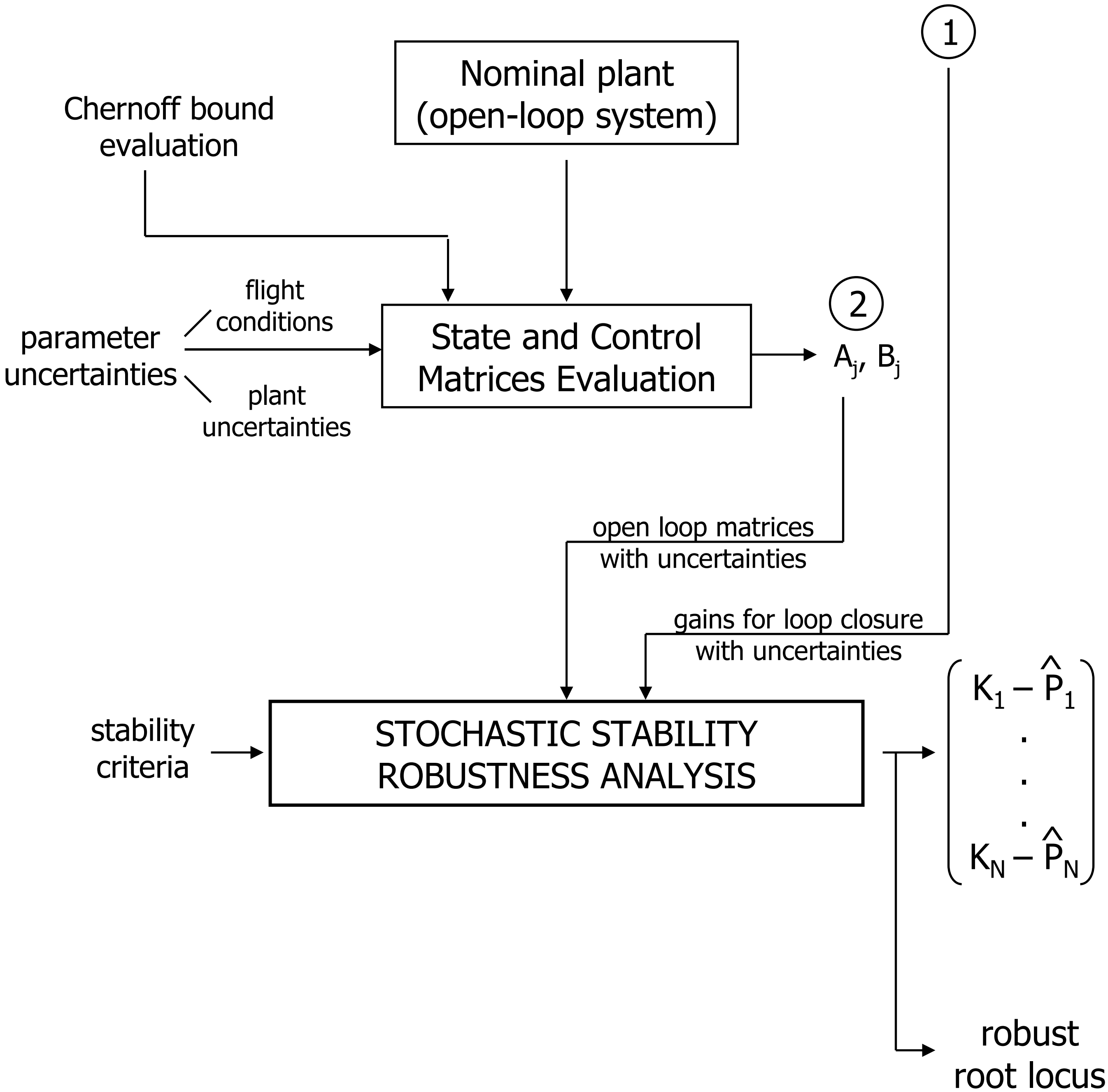
1. Synthesis
2. Stability Analysis
3. Performance Analysis

- Numerical computation of A, B matrices
- User defined specs
- State feedback control
- Randomized algorithms

Phase 1: Synthesis



Phase 2: Stability Analysis



Phase 3: Performance Analysis