Randomized Algorithms for Systems and Control: Theory and Applications

Roberto Tempo

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

Additional documents, papers, etc, please consult
http://staff.polito.it/roberto.tempo/

Questions may be sent to
roberto.tempo@polito.it

References

- RACT: Randomized Algorithms Control Toolbox for Matlab
  http://ract.sourceforge.net

Overview

- Preliminaries
- Randomized Algorithms for Analysis
- Probabilistic Robust Synthesis
- Randomized Algorithms for Optimal Control (LQR)
- Extensions
- Applications: Probabilistic Control of Mini UAVs

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 mini-course: 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 (RQS), …
- Motion and path planning problems
- Mathematics of finance: Computation of path integrals
- Bioinformatics (string matching problems)

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 $H_\infty$ problem by George Zames
- Various “robust” methods to handle uncertainty now exist: Structured singular values, Kharitonov, optimization-based (LMI), $\ell_1$-one optimal control, quantitative feedback theory (QFT)

Limitations of Robust Control - 1

- Researchers realized some drawbacks of robust control
- Consider uncertainty $\Delta$ bounded in a set $\mathcal{B}$ of radius $\rho$
  - Largest value of $\rho$ 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 $NP$-hard (not solvable in polynomial time unless $P=NP$)\cite{1}
- Various robustness problems are $NP$-hard
  - static output feedback
  - structured singular value
  - stability of interval matrices

\cite{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\[1\]


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
- $\Delta$ Uncertainty

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

Worst Case Model

- Worst case model: Set membership uncertainty
- The uncertainty $\Delta$ is bounded in a set $\mathcal{B}_D$
  \[ \Delta \in \mathcal{B}_D \]
- Real parametric uncertainty $q=[q_1, \ldots, q_l] \in \mathbb{R}^l$
  \[ q_i \in [q_i - \epsilon, q_i + \epsilon] \]
- Nonparametric uncertainty $\Delta_i \in \{ \Delta_i \in \mathbb{R}^{n_i}: \| \Delta_i \| \leq 1 \}$
Robustness

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

Objective of Robustness

Objective of robustness: To guarantee stability and performance for all $\Delta \in B_D$
Different probabilistic paradigm based on uncertainty randomization of $\Delta$ within $B_D$

Example: Flexible Structure - 1

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

Flexible Structure - 2

- $M$-$\Delta$ configuration for controlled systems and study stability of
  $$M(s) = C(sI - A)^{-1}B$$
  $$\Delta = \begin{bmatrix} q_1 & 0 & 0 \\ 0 & q_1 & 0 \\ 0 & 0 & \Delta_1 \end{bmatrix}$$
  $q_1, q_2 \in \mathbb{R}$
  $\Delta_1 \in \mathbb{C}^{4,4}$
  $\Delta \in B_D = \{ \Delta \in D : \sigma(\Delta) < \rho \}$

Probability Degradation Function

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

Probabilistic Model
Uniform Density

- Take $f_{\Delta}(\Delta) = U[\mathcal{B}_0]$ (uniform density within $\mathcal{B}_0$)
- $U[\mathcal{B}_0] = \frac{1}{\text{vol}(\mathcal{B}_0)}$ if $\Delta \in \mathcal{B}_0$
- $0$ otherwise
- In this case, for a subset $\mathcal{S} \subseteq \mathcal{B}_0$
- \[\text{Pr}\{\Delta \in \mathcal{S}\} = \frac{\text{vol}(\mathcal{S})}{\text{vol}(\mathcal{B}_0)}\]

Performance Function

- In classical robustness we guarantee that a certain performance requirement is attained for all $\Delta \in \mathcal{B}_0$
- 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 $\mathcal{F}(M, \Delta)$
- $J(\Delta) = \| \mathcal{F}(M, \Delta) \|_\infty$
- For given $\gamma > 0$, check if $J(\Delta) < \gamma$ for all $\Delta$ in $\mathcal{B}_0$

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

- Continuous time SISO systems with real parametric uncertainty $q$ with upper LFT $\mathcal{F}_\Delta(M,q) = \frac{0.5 q_1 q_2 + 10^{-3} q_2}{10^{-5} + 0.05 q_1^2 + (0.00012 + 0.5 q_1) q_2 + (2 \cdot 10^{-3} + 0.5 q_2)}$
- $q_1 \in [0.2, 0.6]$ and $q_2 \in [10^{-5}, 3 \cdot 10^{-5}]$
- Letting $J(q) = \| \mathcal{F}(M,q) \|_\infty$, we choose $\gamma = 0.003$
- Check if $J(q) < \gamma$ for all $q$ in these intervals

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

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

Example[1]: Robust Stability - 1

- Consider the closed loop uncertain polynomial $p(s, q) = (1 + r^2 + 6q_1 + 6q_2 + 2q_1 q_2 + (q_1 + q_2 + 3) s^2 + (q_1 + q_2 + 1)^2 + s^3$
- where $q_1 \in [0.3, 2.5], q_2 \in [0.1, 1.7]$ and $r = 0.5$
- Check stability for all $q$ in these intervals

Example: Robust Stability - 2

- Set of unstable polynomials

Taking \( r = 0 \) the unstable set reduces to a singleton

Good and Bad Sets

- We define two subsets of \( B_D \)
  \[
  \Delta_{\text{good}} = \{ \Delta : J(\Delta) \leq \gamma \} \subseteq B_D \\
  \Delta_{\text{bad}} = \{ \Delta : J(\Delta) > \gamma \} \subseteq B_D
  \]

- \( \Delta_{\text{good}} \) is the set of \( \Delta \)'s satisfying performance
- Measure of robustness is
  \[
  \text{vol}(\Delta_{\text{good}}) = \int_{\Delta_{\text{good}}} d\Delta
  \]

Example of Good and Bad Sets

Taking small \( r \)

Example of Good and Bad Sets - 2

Probabilistic Robustness Measure

- In worst-case analysis we compute \( \gamma \) such that all \( \Delta \) satisfy performance. Equivalently, we evaluate \( \gamma \) such that
  \[
  \Delta_{\text{good}} = B_D
  \]
- In a probabilistic setting, we are satisfied if the ratio
  \[
  \frac{\text{vol}(\Delta_{\text{good}})}{\text{vol}(B_D)}
  \]
  is close to one

Probability of Performance

- We define the probability of performance as
  \[
  p_\gamma = \Pr \{ J(\Delta) \leq \gamma \}
  \]
- Notice that, if \( f_\Delta(\Delta) \) is uniform, then
  \[
  p_\gamma = \frac{\text{vol}(\Delta_{\text{good}})}{\text{vol}(B_D)}
  \]

Example: Closed-Form Computation

- For Truxal’s example, we compute $p_\gamma$ in closed-form
- For uniform distribution, we have
  
  $\text{vol}(\Delta_{\text{good}}) = 3.74 - \pi r^2$
  $\text{vol}(B_\gamma) = 3.74$

P1: Performance Verification

- For given performance level $\gamma$, check whether
  
  $J(\Delta) \leq \gamma$
  for all $\Delta$ in $B_\gamma$
- Compute the probability of performance $p_\gamma$

P2: Worst-Case Performance

- Find $J_{\text{max}}$ such that
  
  $J_{\text{max}} = \max_{\Delta \in B_\gamma} J(\Delta)$
- Compute the worst case performance (or its probabilistic counterpart)

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

  \[ \text{heads or tails} \]
Monte Carlo Randomized Algorithm

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

Las Vegas Randomized Algorithm

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

Randomization of Uncertain Systems

- Consider random uncertainty $\Delta$, associated pdf and bounding set $B$.
- $\Delta$ is a (real or complex) random vector (parametric uncertainty) or matrix (nonparametric uncertainty).
- Consider a performance function $J(\Delta): B \rightarrow \mathbb{R}$ and level $\gamma > 0$.
- Define worst case and average performance
  
  $J_{\text{max}} = \max_{\Delta \in B} J(\Delta)$
  $J_{\text{ave}} = E_{\Delta}(J(\Delta))$

Example: $H_{\infty}$ Performance - 1

- $H_{\infty}$ performance of sensitivity function
  
  $B = \{ \Delta: \Delta = \text{bdag} (\Delta_1, \ldots, \Delta_q) \in F^{n,m}, \sigma_{\text{max}}(\Delta) \leq \rho \}$
  
  $S(s,\Delta) = 1/(1 + P(s,\Delta) C(s))$
  $J(\Delta) = \|S(s,\Delta)\|_\infty$

Example: $H_{\infty}$ Performance - 2

- $H_{\infty}$ performance of sensitivity function
  
  $B = \{ \Delta: \Delta = \text{bdag} (\Delta_1, \ldots, \Delta_q) \in F^{n,m}, \sigma_{\text{max}}(\Delta) \leq \rho \}$
  
  $S(s,\Delta) = 1/(1 + P(s,\Delta) C(s))$
  $J(\Delta) = \|S(s,\Delta)\|_\infty$

- Objective: Check if
  
  $J_{\text{max}} \leq \gamma$ and $J_{\text{ave}} \leq \gamma$

- These are uncertain decision problems

Two Problem Instances

- We have two problem instances for worst case performance
  
  $J_{\text{max}} \leq \gamma$ and $J_{\text{max}} > \gamma$

- and two problem instances for average case performance
  
  $J_{\text{ave}} \leq \gamma$ and $J_{\text{ave}} > \gamma$

- This leads to one-sided and two-sided MCRA.
One-Sided MCRA

- One-sided MCRA: Always provide a correct solution in one of the instances (they may provide a wrong solution in the other instance)
- Consider the empirical maximum

\[ \hat{J}_{\text{max}} = \max_{i=1,...,N} J(\Delta_i) \]

where \( \Delta_i \) are random samples and \( N \) is the sample size
- Check if \( \hat{J}_{\text{max}} \leq \gamma \) or \( \hat{J}_{\text{max}} > \gamma \)

One-Sided MCRA: Case 1

- Algorithm provides a correct solution

Two-Sided MCRA

- Two-sided MCRA: They may provide a wrong solution in both instances
- Consider the empirical average

\[ \hat{J}_{\text{ave}} = \text{ave}_{i=1,...,N} J(\Delta_i) \]

where \( \Delta_i \) are random samples and \( N \) is the sample size
- Check if \( \hat{J}_{\text{ave}} \leq \gamma \) or \( \hat{J}_{\text{ave}} > \gamma \)
Las Vegas Randomized Algorithms

- We also have zero-sided (Las Vegas) randomized algorithms.
- Las Vegas Randomized Algorithm (LVRA): Always give the correct solution.
- The solution obtained with a LVRA is probabilistic, so “always” means with probability one.
- Running time may be different from one run to another.
- We can study the average running time.

Discrete Bounding Set

Consider random uncertainty \( \Delta \), a discrete bounding set \( B \), given pdf and performance function.

The Las Vegas Viewpoint

- Consider discrete random variables.
- The sample space is discrete and \( M^N \) possible choices can be made.
- In the binary case we have \( 2^N \).
- Finding maximum requires ordering the \( 2^N \) choices.
- Las Vegas can be used for ordering real numbers.
- Example: Randomized Quick Sort for sorting real numbers (classical in computer science).

Ingredients for RAs

- Assume that \( \Delta \) is random with pdf \( f_\Delta (\Delta) \) with support \( B_\Delta \).
- Accuracy \( \varepsilon \in (0,1) \) and confidence \( \delta \in (0,1) \) be assigned.
- Performance function for analysis and level

\[
\downarrow \quad \downarrow
\]

\[
J = J(\Delta) \quad \gamma
\]
Randomized Algorithms for Analysis

- Two classes of randomized algorithms for probabilistic robust performance analysis
- P1: Performance verification (compute \( p_\gamma \))
- P2: Worst-case performance (compute \( J_{\text{max}} \))
- Both are based on uncertainty randomization of \( \Delta \)
- Bounds on the sample size are obtained

Randomized Algorithms - 2

- We estimate \( p_\gamma \) by means of a randomized algorithm
- First, we generate \( N \) i.i.d. samples \( \Delta_1, \Delta_2, ..., \Delta_N \in B_D \) according to the density \( f_\Delta \)
- We evaluate \( J(\Delta_1), J(\Delta_2), ..., J(\Delta_N) \)

Empirical Probability

- Construct an indicator function
  \[
  I(\Delta) = \begin{cases} 
  1 & \text{if } J(\Delta) \leq \gamma \\
  0 & \text{otherwise}
  \end{cases}
  \]
- An estimate of \( p_\gamma \) is the empirical probability
  \[
  \hat{p}_\gamma = \frac{1}{N} \sum_{i=1}^{N} I(\Delta_i) = \frac{N_{\text{good}}}{N}
  \]
  where \( N_{\text{good}} \) is the number of samples such that \( J(\Delta) \leq \gamma \)

A Reliable Estimate

- The empirical probability is a reliable estimate if
  \[
  |p_\gamma - \hat{p}_\gamma| = |\Pr(J(\Delta) \leq \gamma) - \hat{p}_\gamma| \leq \epsilon
  \]
- Find the minimum \( N \) such that
  \[
  \Pr(|p_\gamma - \hat{p}_\gamma| \leq \epsilon) \geq 1 - \delta
  \]
  where \( \epsilon \in (0,1) \) and \( \delta \in (0,1) \)

Chernoff Bound\(^1\)

- For any \( \epsilon \in (0,1) \) and \( \delta \in (0,1) \), if
  \[
  N \geq \frac{\log \frac{2}{2\epsilon}}{2\epsilon^2}
  \]
  then
  \[
  \Pr(|p_\gamma - \hat{p}_\gamma| \leq \epsilon) \geq 1 - \delta
  \]

Comparison Between Bounds

---

\(^1\) H. Chernoff (1952)
Chernoff Bound

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

<table>
<thead>
<tr>
<th>$\epsilon$</th>
<th>0.1%</th>
<th>0.1%</th>
<th>0.5%</th>
<th>0.5%</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1-\delta$</td>
<td>99.9%</td>
<td>99.5%</td>
<td>99.9%</td>
<td>99.5%</td>
</tr>
<tr>
<td>$N$</td>
<td>$3.9 \times 10^6$</td>
<td>$3.0 \times 10^6$</td>
<td>$1.6 \times 10^6$</td>
<td>$1.2 \times 10^6$</td>
</tr>
</tbody>
</table>

Computational Complexity of RAs

- RAs are efficient (polynomial-time) because
  1. Random sample generation of $\Delta$ can be performed in polynomial-time
  2. Cost associated with the evaluation of $J(\Delta)$ for fixed $\Delta$ is polynomial-time
  3. Sample size is polynomial in the problem size and probabilistic levels $\epsilon$ and $\delta$

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 $\Delta$ with pdf $f_\Delta(\Delta)$ and support $\mathcal{B}_D$
- 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 $n^2$ for $n$ even, $n^2 - 1$ for $n$ 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 $\mathcal{B}_D$, on the structure $D$ on the number of blocks, on the pdf $f_\Delta(\Delta)$
- It depends only on $\delta$ and $\epsilon$
- Same comments can be made for other bounds such as Bernoulli

Worst-Case Performance

- Recall that $J_{\text{max}} = \max_{\Delta \in \mathcal{B}_D} J(\Delta)$
- Generate $N$ i.i.d. samples $\Delta^1, \Delta^2, \ldots, \Delta^N \in \mathcal{B}_D$ according to the density $f_\Delta$
- Compute the empirical maximum $\hat{J}_{\text{max}} = \max_{i=1,\ldots,N} J(\Delta^i)$
Worst-Case Bound (Log-over-Log)\[^1\]

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

\[
N \geq \frac{\log \frac{1}{\delta}}{\log \frac{1}{\varepsilon}}
\]

then

\[
\Pr \{ \Pr \{ J(\Delta) > \hat{J}_N \} \leq \varepsilon \} \geq 1 - \delta
\]

\[^{[1]}\] 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

<table>
<thead>
<tr>
<th>( \varepsilon )</th>
<th>0.1%</th>
<th>0.1%</th>
<th>0.5%</th>
<th>0.5%</th>
<th>0.01%</th>
<th>0.001%</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 1 - \delta )</td>
<td>99.9%</td>
<td>99.5%</td>
<td>99.9%</td>
<td>99.9%</td>
<td>99.99%</td>
<td>99.999%</td>
</tr>
<tr>
<td>( N )</td>
<td>6.91 ( \times 10^2 )</td>
<td>5.30 ( \times 10^3 )</td>
<td>1.38 ( \times 10^4 )</td>
<td>1.06 ( \times 10^5 )</td>
<td>9.21 ( \times 10^6 )</td>
<td>1.16 ( \times 10^7 )</td>
</tr>
</tbody>
</table>

Volumetric Interpretation

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

\[
\Pr \{ J(\Delta) > \hat{J}_N \} = \frac{\text{vol}(\Delta_{bad})}{\text{vol}(\mathcal{B}_N)}
\]

- Therefore

\[
\Pr \{ \Pr \{ J(\Delta) > \hat{J}_N \} \leq \varepsilon \} \geq 1 - \delta
\]

is equivalent to

\[
\Pr \{ \text{vol}(\Delta_{bad}) \leq \varepsilon \text{vol}(\mathcal{B}_N) \} \geq 1 - \delta
\]

Confidence Intervals

- The Chernoff and worst-case bounds can be computed \textit{a-priori} and provide an explicit functional relation

\[ N = N(\varepsilon, \delta) \]

- The sample size obtained with the confidence intervals is not explicit

- Given \( \delta \in (0, 1) \), upper and lower confidence intervals \( p_L \) and \( p_U \) are such that

\[
\Pr \{ p_L \leq p \leq p_U \} = 1 - \delta
\]

Confidence Intervals - 2

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

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

\[
\sum_{k=0}^{N} \binom{N}{k} p_U^k (1 - p_U)^{N-k} = \delta_U
\]

with \( \delta_L + \delta_U = \delta \)

Confidence Intervals - 3

\( \delta = 0.05 \)
Statistical Learning Theory

- The Chernoff Bound studies the problem
  \[ \Pr\{|p_\gamma - \hat{p}_\gamma| \leq \epsilon\} \geq 1 - \delta \]
  where \( p_\gamma = \Pr\{J(\Delta) \leq \gamma\} \)
- Performance function \( J \) is fixed
- Statistical Learning Theory computes bounds on the sample size for the problem
  \[ \Pr\{\Pr\{J(\Delta) \leq \gamma\} - \hat{p}_\gamma \leq \epsilon, \forall J \in \mathcal{J}\} \geq 1 - \delta \]
  where \( \mathcal{J} \) is a given class of functions

VC and P-dimension\(^{[1,2]}\)

- Statistical 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

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 \( f_\Delta(\Delta) \)

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

Choice of the Distribution - 3

- Minimax properties of the uniform distribution have been studied\(^{[1]}\)

Probabilistic Robust Synthesis

\(^{[1]}\) M. Vidyasagar (1997)
\(^{[2]}\) E.D. Sontag (1998)

\(^{[1]}\) B. R. Barmish and C. M. Lagoa (1997)

\(^{[1]}\) E. W. Bai, R. Tempo and M. Fu (1998)
Analysis vs Design with Uncertainty

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

Design the parameterized controller \( K_\theta \) to guarantee stability and performance

Synthesis Performance Function

- Recall that the parameterized controller is \( K_\theta \)
- We replace \( J(\Delta) \) with a synthesis performance function
  \[
  J(\Delta, \theta) = J(\Delta, \theta)
  \]
  where \( \theta \in \Theta \) represents the controller parameters to be determined and their bounding set

Synthesis Paradigm

Randomized Algorithms for Synthesis

- Two classes of RAs for probabilistic synthesis
- Average performance synthesis\(^1\)
- Based on expected value minimization
- Use of Statistical Learning Theory results
- Very general problems can be handled
- Existing bounds are very conservative and controller randomization is required
- Ongoing research aiming at major reduction of sample size

\(^1\) M. Vidyasagar (1998)

Robust Performance Synthesis

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

\(^1\) B. Polyak and R. Tempo (2001)
We consider a state space description of the uncertain system
\[
\dot{x}(t) = A(x(t)) + Bu(t)
\]
with \(x(0) = x_0, x \in \mathbb{R}^n, u \in \mathbb{R}^m, \Delta \in \mathbb{B}_D\)

For example, \(A(\Delta)\) is an interval matrix with bounded entries
\[
a_{ij}^* \leq a_{ij} \leq a_{ij}^*
\]

We consider interval uncertainty (i.e. when \(\Delta \in \mathbb{B}_D\))

That is, \(a_{ij}\) ranges in the interval for all \(i, k\)

\[
|a_{ij} - a_{ij}^*| \leq w_{ik}
\]

where \(a_{ij}^*\) are nominal values and \(w_{ik}\) are weights

Define the \(N = 2^{2n}\) vertex matrices \(A^1, A^2, \ldots, A^N\)

\[
a_{ik} = a_{ik}^* + w_{ik} \quad \text{or} \quad a_{ik} = a_{ik}^* - w_{ik}
\]

for all \(i, k = 1, 2, \ldots, n\)

Given matrices \(A^*, W\) and feedback \(K\), find a common quadratic Lyapunov function \(Q > 0\) for the system

\[
\dot{x}(t) = (A + BK)x(t) \quad \text{for all} \quad A \in \mathcal{A}
\]

Find \(Q > 0\) such that

\[
L(Q, A) = (A + BK)^T Q + Q (A + BK) < 0 \quad \text{for all} \quad A \in \mathcal{A}
\]

Equivalently, find \(Q > 0\) such that

\[
\lambda_{\text{max}} L(Q, A) < 0 \quad \text{for all} \quad A \in \mathcal{A}
\]

Due to convexity, it suffices to study \(L(Q, A) < 0\) for all vertex matrices\(^1\)

Question: Do we really need to check all the vertex matrices \((N = 2^{2n})?\)

\[^1\] H.P. Horisberger, P.R. Belanger (1976)
Vertex Reduction

- Answer: It suffices to check “only” a subset of $2^{2n}$ vertex matrices.[1]
- This is still exponential (the problem is NP-hard), but it leads to a major computational improvement for medium size problems (e.g. $n = 8$ or 10)
- For example, for $n=8$, $N$ is of the order $10^5$ (instead of $10^{19}$)


Diagonal Matrices and Generalizations

- Transform the original problem from full square matrices $A$ to diagonal matrices $Z \in \mathbb{R}^{2n,2n}$
- It suffices to check the vertices of $Z$
- Extensions for $L_2$-gain minimization and other related LMI problems
- Generalizations for multi-affine interval systems

Las Vegas Randomized Algorithm

- We may perform randomization of the $N = 2^{2n}$ vertices (in the worst case)
- If we select the vertices in random order according to a given pdf, we have a LVRA

Probabilistic Solution

- Randomly generate $A_1, \ldots, A_N$. Then, check if the Lyapunov equation
  \[ A^T Q + Q A \leq 0 \]
  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 $\Delta$
    - 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 $\mathcal{E}_a$ be an Euclidean space
\[ \mathcal{E}_a = \left\{ A = A^T \in \mathbb{R}^n, |A| = \sum_{i=1}^{n} a_i \right\} \]
and $C$ be the cone of positive semi-definite matrices
\[ C = \{ A \in \mathcal{E}_a : A \geq 0 \} \]
Projection on a Cone

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

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

The projection can be computed through the eigenvalue decomposition $A = \Gamma \Lambda \Gamma^T$

Then

$$[A]^+ = \Gamma \Lambda^+ \Gamma^T$$

where $\lambda_{\text{max}} \{\lambda, 0\}$

Phase 1: Uncertainty Randomization

Uncertainty randomization: Generate $\Delta \in B_{Q}$

Then, for guaranteed cost we obtain the Lyapunov equation

$$A(\Delta^T)Q + QA^T(\Delta^T) \leq 0$$

Matrix Valued Function

Define a matrix valued function

$$V(Q, \Delta) = A(\Delta)Q + QA(\Delta)$$

and a scalar function

$$V(Q, \Delta^T) = \left\| V(Q, \Delta) \right\|$$

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

We can also take the maximum eigenvalue of $V(Q, \Delta)$

Phase 2: Gradient Algorithm

We write

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

where $\partial_Q V(Q, \Delta)$ is the subgradient and the stepsize $\mu^k$ is

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

and $r > 0$ is a parameter

Closed-form Gradient Computation

The function $V(Q, \Delta)$ is convex in $Q$ and its subgradient can be easily computed in a closed form

Theorem

Assumption: Every open subset of $B_{Q}$ 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

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

\[
\begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & L_p & L_q & L_t \\
N_p & 0 & Y_p & -1 \\
N_p + N_q & N_p & N_r - N_f & 0
\end{bmatrix}
\begin{bmatrix}
x(t) \\
x(t) \\
x(t) \\
x(t)
\end{bmatrix}
+ 
\begin{bmatrix}
0 \\
0 \\
0.035 \\
0
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2
\end{bmatrix}
\]


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:

- \( L_p = -2.93 \)
- \( L_p = -4.75 \)
- \( L_t = 0.78 \)
- \( g/V = 0.086 \)
- \( Y_p = 0.11 \)
- \( N_p = 0.1 \)
- \( N_r = 0.042 \)
- \( N_p = 2.601 \)
- \( N_r = 0.29 \)

Perturbed matrix \( A(\Delta) \): each parameter can take values in a range of \( \pm 15\% \) of the nominal value

Quadratic stability (\( \gamma = 0 \)): take \( R = I \) and \( S = 0.01I \)

Remark: \( A(\Delta) \) is multiaffine in the uncertain parameters: quadratic stability can be ascertained solving simultaneously \( 2^5 = 512 \) LMIs

Example - 4

Sequential algorithm:

- Initial point \( Q_0 \) randomly selected
- 800 random matrices \( \Delta_k \)
- 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

Extensions
Related Literature and Extensions

- Minimization of a measure of violation for problems that are not strictly feasible\(1\)
- Uncertainty in the control matrix, \(B=B(\Delta), \Delta \in B_\Omega\)

We take the feedback law

\[
\nu = YQ^{-1}x
\]

where \(Y\) and \(Q=Q^T > 0\) are design variables

\[\text{[1]}\] B.R. Barmish and P. Shcherbakov (1999)

---

Optimization Problems\(^{[1]}\)

- Extensions to optimization problems
- Consider convex function \(f(x)\) and function \(g(x,\Delta)\)
  convex in \(x\) for fixed \(\Delta\)
- Semi-infinite (nonlinear) programming problem
  \[
  \min f(x) \\
  g(x,\Delta) \leq 0 \text{ for all } \Delta \in B
  \]
- Reformulation as stochastic optimization
- Drawback: Convergence results are only asymptotic


---

Subsequent Research

- Design of common Lyapunov functions for switched systems\(^{[1]}\)
- From common to piecewise Lyapunov functions\(^{[2]}\)
- Ellipsoidal algorithm instead of gradient algorithm\(^{[3]}\)
- Stopping rule which provides the number of steps\(^{[4]}\)
- Other algorithms have been recently proposed\(^{[5-6]}\)

\[\text{[1]}\] D. Liberzon and R. Tempo (2004)
\[\text{[5]}\] Y. Fujisaki and Y. Oishi (2007)

---

Scenario Approach

- The scenario approach for convex problems\(^{[1]}\)
- Non-sequential method which provides a one-shot solution for general convex problems
- Randomization of \(\Delta \in B\) and solution of a single convex optimization problem
- Derivation of a bound on the sample size\(^{[1]}\)
- A new improved bound based on a pack-based strategy\(^{[2]}\)

\[\text{[2]}\] Y. Alamo, R. Tempo and E.F. Camacho (2007)
Scenario Problem

- Using randomization, we construct a scenario problem
- Taking random samples $\Delta_i, i = 1, 2, \ldots, N$, we construct $f(\theta, \Delta_i) \leq 0$, $i = 1, 2, \ldots, N$

\[ \min c^T \theta \quad \text{subject to} \quad f(\theta, \Delta_i) \leq 0, \ i = 1, 2, \ldots, N \]

Theorem\(^{[1]}\)

- Theorem: For any $\epsilon \in (0,1)$ and $\delta \in (0,1)$, if

\[ N \geq \left\lceil \frac{2/\epsilon \log(1/\delta)}{2n + 2n/\epsilon \log (2/\delta)} \right\rceil \]

then, with probability no smaller than $1 - \delta$:
- either the scenario problem is unfeasible and then also the semi-infinite optimization problem is unfeasible
- or, the scenario problem is feasible, then its optimal solution $\theta_N$ satisfies

\[ \Pr\{ \Delta \in B : f(\theta_N, \Delta) > 0 \} \leq \epsilon \]

\(^{[1]}\) G. Calafiore and M. Campi (2004)

A New Improved Bound\(^{[1]}\)

- A new improved bound (based on a so-called pack-based strategy) has been recently obtained

\[ N \geq \left\lceil \frac{2/\epsilon \log(1/\delta)}{2n + 2n/\epsilon \log 4} \right\rceil \]

- The main difference with the previous bound is that the factor

\[ 2n/\epsilon \log (2/\epsilon) \]

is replaced with

\[ 2n/\epsilon \log 4 \]

\(^{[1]}\) T. Alonso, R. Tempo and E.F. Camacho (2007)

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 Scherbakov
  - Roberto Tempo (Co-Principal Investigator)

- 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
  - Randomized algorithms for probabilistic performance verification and probabilistic worst-case performance
  - Randomized algorithms for feasibility of uncertain LMIs using stochastic gradient, ellipsoid or cutting plane methods (YALMIP needed)

\(^{[1]}\) Randomized Algorithms for Systems and Control: Theory and Applications
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

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

Probabilistic Control of Mini-UAVs[1]

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)

Flight Envelope (Limits)

- Wing loading effect: total weight
- Propeller sizing effect: maximum flight speed

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

Basic on-board Systems

- DC motor: Hacker B20-15L (4:1)
- Controller: Hacker Master Series 18-B-Flight
- Battery: Kokam 2000HD (3x)
- Receiver: Schulze Alpha840W
- Servo: Graupner C1081 (2x)

Prototype Manufacturing - 1

- Raw material: polystyrene, kevlar, fiberglass, carbon fiber, epoxy resin, plywood, balsa wood, glue
- Working instruments: hot wire foam cutting machine, slide, fuselage reference

Prototype Manufacturing - 2

- Working instruments: working surface, slide, fuselage reference

Prototype Manufacturing - 3

- Prototype: 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, \( \alpha \) angle of attack, \( q \) and \( \theta \) pitch rate and angle), \( \Delta \) uncertainty
- Consider longitudinal plane dynamics stabilization
- Control \( u \) is the symmetrical elevon deflection
We consider structured parameter uncertainties affecting plant and flight conditions, and aerodynamic database uncertainties. The uncertainty vector $\mathbf{\Delta} = [\mathbf{\delta}_1, \ldots, \mathbf{\delta}_n]$ where $\mathbf{\delta}_i \in [\mathbf{\delta}_i^-, \mathbf{\delta}_i^+]$.

Key point: There is no explicit relation between state space matrices $\mathbf{A}$ and $\mathbf{B}$ and uncertainty $\mathbf{\Delta}$. 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.

We consider random uncertainty $\mathbf{\Delta} = [\mathbf{\delta}_1, \ldots, \mathbf{\delta}_n]^T$. The pdf is either uniform (for plant and flight conditions) or 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.

### Plant and Flight Condition Uncertainties

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Pdf</th>
<th>$\delta_1$</th>
<th>$\delta_2$</th>
<th>$\delta_3$</th>
<th>$\delta_4$</th>
<th>$\delta_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flight speed [ft/s]</td>
<td>$\mathcal{U}$</td>
<td>42.65</td>
<td>15</td>
<td>36.25</td>
<td>49.05</td>
<td>1</td>
</tr>
<tr>
<td>Altitude [ft]</td>
<td>$\mathcal{U}$</td>
<td>164.04</td>
<td>100</td>
<td>0</td>
<td>328.08</td>
<td>2</td>
</tr>
<tr>
<td>Mass [lb]</td>
<td>$\mathcal{U}$</td>
<td>3.3</td>
<td>10</td>
<td>2.98</td>
<td>3.64</td>
<td>3</td>
</tr>
<tr>
<td>Wing span [ft]</td>
<td>$\mathcal{U}$</td>
<td>3.28</td>
<td>5</td>
<td>5.12</td>
<td>5.44</td>
<td>4</td>
</tr>
<tr>
<td>Mean aero chord [ft]</td>
<td>$\mathcal{U}$</td>
<td>1.75</td>
<td>5</td>
<td>1.67</td>
<td>1.85</td>
<td>5</td>
</tr>
<tr>
<td>Wing surface [ft$^2$]</td>
<td>$\mathcal{U}$</td>
<td>5.61</td>
<td>10</td>
<td>5.06</td>
<td>6.18</td>
<td>6</td>
</tr>
<tr>
<td>Moment of inertia [lb ft$^2$]</td>
<td>$\mathcal{U}$</td>
<td>1.34</td>
<td>10</td>
<td>1.21</td>
<td>1.48</td>
<td>7</td>
</tr>
</tbody>
</table>

### Aerodynamic Database Uncertainties

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Pdf</th>
<th>$\mathcal{U}$</th>
<th>$\mathcal{U}$</th>
<th>$\mathcal{U}$</th>
<th>$\mathcal{U}$</th>
<th>$\mathcal{U}$</th>
<th>$\mathcal{U}$</th>
<th>$\mathcal{U}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_L$ [-]</td>
<td>$\mathcal{U}$</td>
<td>-0.0115</td>
<td>-0.0040</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C_L$ [-]</td>
<td>$\mathcal{U}$</td>
<td>-0.02651</td>
<td>-0.0050</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C_L$ [-]</td>
<td>$\mathcal{U}$</td>
<td>-0.02401</td>
<td>0.0040</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C_D$ [rad$^{-1}$]</td>
<td>$\mathcal{U}$</td>
<td>-0.2435</td>
<td>0.0650</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C_D$ [rad$^{-1}$]</td>
<td>$\mathcal{U}$</td>
<td>-1.4946</td>
<td>0.0500</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C_{D_1}$ [rad$^{-1}$]</td>
<td>$\mathcal{U}$</td>
<td>-0.7688</td>
<td>0.0100</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C_{D_2}$ [rad$^{-1}$]</td>
<td>$\mathcal{U}$</td>
<td>-0.1707</td>
<td>0.0540</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C_J$ [rad$^{-1}$]</td>
<td>$\mathcal{U}$</td>
<td>-1.4113</td>
<td>0.0220</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C_J$ [rad$^{-1}$]</td>
<td>$\mathcal{U}$</td>
<td>-0.94853</td>
<td>0.0150</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Standard Deviation and Velocity

Standard deviation is experimentally computed from the velocity.

### Critical Parameters and Matrices

- We select flight speed ($\mathbf{\delta}_1$) and take off mass ($\mathbf{\delta}_3$) 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 $A_1, A_2, A_3, A_4, B_1, B_2, B_3, B_4$.
- They are constructed setting $\mathbf{\delta}_1, \mathbf{\delta}_3$ to the extreme values $\mathbf{\delta}_1^-, \mathbf{\delta}_1^+, \mathbf{\delta}_3^-, \mathbf{\delta}_3^+$ and all the remaining $\mathbf{\delta}_i$ are equal to $\bar{\mathbf{\delta}}$. 
Phase 1: Random Gain Synthesis (RGS)

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

\[ S_1 = \{ K \mid A_\theta - B_\theta K \text{ satisfies the specs below} \} \]

\[ \omega_{SP} \in [4.0, 6.0] \text{ rad/s} \]
\[ \zeta_{SP} \in [0.5, 0.9] \]
\[ \omega_{PH} \in [1.0, 1.5] \text{ rad/s} \]
\[ \zeta_{PH} \in [0.1, 0.3] \]
\[ \Delta \omega_{SP} < \pm 45\% \]
\[ \Delta \omega_{PH} < \pm 20\% \]

where \( \omega \) and \( \zeta \) are undamped natural frequency and damping ratio of the characteristic modes; \( SP \) and \( PH \) denote short period and phugoid mode.

Volume of the Good Set

- Define a bounding set \( B \) of gains \( K \)

\[ B = \left \{ K \mid k_i \in [k_{i-}, k_i^+], i = 1, \ldots, 4 \right \} \]

- Define the volume of the good set

\[ \text{Vol}_{good} = \int_B dK \]

where \( A = \{ K \in B \cap S_1 \} \)

- \( \text{Vol}_A \) is simply the volume of the hyperrectangle \( B \)

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 5 gains \( K \) satisfying specification property \( S_1 \)

Random Gain Set

<table>
<thead>
<tr>
<th>( K_1 )</th>
<th>( K_2 )</th>
<th>( K_3 )</th>
<th>( K_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00044023</td>
<td>0.09465000</td>
<td>0.01577400</td>
<td>-0.00473510</td>
</tr>
<tr>
<td>0.00021450</td>
<td>0.09581200</td>
<td>0.01555000</td>
<td>-0.00323510</td>
</tr>
<tr>
<td>0.00054999</td>
<td>0.09430800</td>
<td>0.01548200</td>
<td>-0.00486340</td>
</tr>
<tr>
<td>0.00010855</td>
<td>0.09183200</td>
<td>0.01530000</td>
<td>-0.00404380</td>
</tr>
<tr>
<td>0.00059238</td>
<td>0.09482700</td>
<td>0.01609300</td>
<td>-0.00417340</td>
</tr>
</tbody>
</table>
Phase 2: Random Stability Robustness Analysis (RSRA)

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

\[ S_2 = \{ \Delta : A(\Delta) - B(\Delta) \cdot K_{rand} \text{ satisfies the specs of } S_1 \} \]

- Computation of the empirical probability of stability $\hat{P}_N$

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 defined using an indicator function

Randomized Algorithm 2 (RSRA)

- Consider fixed gain $K_{rand}$
- Define the probability

\[ P_{true} = \int_C p(\Delta) \, d\Delta \]

where $C = \{ \Delta \in B' \cap S_2 \}$ and $p(\Delta)$ is the given pdf

- Then, we introduce a "success" indicator function $\delta(\Delta) = 1$ if $\Delta \in S_2$
  or $\delta(\Delta) = 0$ otherwise

- The empirical probability for $S_2$ is given by

\[ \hat{P}_N = \frac{N_{good}}{N} \]

where $N_{good}$ is equal to the number of successes

Empirical Probability

Given $\varepsilon, \delta \in (0,1)$, RSRA returns the empirical probability $\hat{P}_N$ that $S_2$ is satisfied for a gain $K_{rand}$ provided by Algorithm 1

1. Compute $N$ using the Chernoff Bound;
2. Generate $N$ random vectors $\Delta \in B'$ according to the given pdf;
3. For fixed $j=1,2,\ldots,N$, compute the closed-loop matrix

\[ A_{cl}(\Delta) = A(\Delta) - B(\Delta) \cdot K_{rand} \]

- if $A_{cl}(\Delta) \in S_2$, set $\delta(\Delta) = 1$;
- otherwise, set $\delta(\Delta) = 0$;
4. End;
5. Return the empirical probability $\hat{P}_N$

Empirical Probability of Stability for Phase 2

<table>
<thead>
<tr>
<th>gain set</th>
<th>empirical probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K'$</td>
<td>88.56%</td>
</tr>
<tr>
<td>$K'$</td>
<td>90.60%</td>
</tr>
<tr>
<td>$K'$</td>
<td>89.31%</td>
</tr>
<tr>
<td>$K'$</td>
<td>93.86%</td>
</tr>
<tr>
<td>$K'$</td>
<td>85.14%</td>
</tr>
</tbody>
</table>

Probability Degradation Function

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

\[ \delta_i = \rho [\delta_i, \delta_i] \quad \text{for } i = 1, 2, \ldots, 7 \]

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

\[ \delta_i = \bar{\delta} \quad \text{for } i = 8, 9, \ldots, 16 \]

- For fixed $\rho \in [0,1.5]$ we compute the empirical probability for different gain sets $K'$

- The plot empirical probability vs $\rho$ is the probability degradation function
Probability Degradation Function for Phase 2

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
  \[ S_3 = \{ \Delta: A(\Delta) - B(\Delta) \text{ } K_{\text{rand}} \text{ satisfies the specs below} \} \]
  \[ \omega_{BW} \in [2.5,5.0] \text{ rad/s} \quad \tau_P \in [0.0,0.5] \text{ s} \]
  where $\omega_{BW}$ and $\tau_P$ are bandwidth and phase delay of the frequency response
- Computation of the empirical probability that $S_3$ is satisfied

Bandwidth Criterion

Randomized Algorithm 3 (RPRA)

- Take $K_{\text{rand}}$ from Phase 1
- Number of random samples is computed with the Chernoff Bound obtaining $N=5,000$
- Empirical probability is defined using an indicator function

Randomized Algorithm 3 (RPRA)

Given $N$ and $A_j(\Delta)$, $j=1,2,\ldots,N$, provided by Algorithm 2, RPRA returns the empirical probability $\hat{p}_N$ that $S_3$ is satisfied for a gain $K_{\text{rand}}$ provided by Algorithm 1

1. For fixed $j=1,2,\ldots,N$
   - if $A_j(\Delta) \in S_3$, set $\hat{p}_N = 1$;
   - otherwise, set $\hat{p}_N = 0$;
2. End;
3. Return the empirical probability $\hat{p}_N$
Empirical Probability of Performance for Phase 3

<table>
<thead>
<tr>
<th>Gain Set</th>
<th>Empirical Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K^1$</td>
<td>93.58%</td>
</tr>
<tr>
<td>$K^2$</td>
<td>95.16%</td>
</tr>
<tr>
<td>$K^3$</td>
<td>90.80%</td>
</tr>
<tr>
<td>$K^4$</td>
<td>84.78%</td>
</tr>
<tr>
<td>$K^5$</td>
<td>96.06%</td>
</tr>
</tbody>
</table>

Probability Degradation Function for Phase 3

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!

Bandwidth Criterion for Phase 3

Bandwidth criterion for $K^1$ (left) and $K^2$ (right)

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
Acknowledgment

Thanks to Andrea Sanna and his students for the flight test video and the computer graphics animation.

Conclusions

PAC Algorithms

- Randomized algorithms are Probably Approximately Correct (PAC)
- We give up a guaranteed deterministic solution
- This implies accepting a “small” risk of giving a wrong solution
- The risk can be made arbitrarily small (but not zero) taking suitable values of so-called confidence and accuracy
PAC Algorithms

- Two open problems
- Optimization with sequential methods
- Derive “reasonable” bounds for the statistical learning theory approach