A new step towards the integration of probabilistic mu in the aerospace V&V process

Clément Roos, Jean-Marc Biannic, Hélène Evain

A new step towards the integration of probabilistic μ in the aerospace V&V process

Clément Roos[1*], Jean-Marc Biannic[1] and Hélène Evain[2]

[1]Information Processing and Systems Department, ONERA The French Aerospace Lab, 2 avenue Edouard Belin, Toulouse, 31400, France.
[2]Flight Dynamics Subdirectorate, CNES, 18 avenue Edouard Belin, Toulouse, 31400, France.

*Corresponding author. E-mail: clement.roos@onera.fr

Abstract

Probabilistic $\boldsymbol{\mu}$-analysis was introduced 20 years ago as a control system validation means able to quantify the probability of rare and potentially critical events. But for a long time, no practical tool offering both good reliability and reasonable computational time was available, making this technique hardly usable in an industrial context. The STOchastic Worst-case Analysis Toolbox (STOWAT) was introduced a few years ago to bridge this gap between theory and practice. It has been significantly improved since then, thanks to the addition of new features, but above all to increasingly efficient implementations, resulting in a dramatic reduction in CPU time. However, until recently, it could only be applied to small-scale models, with up to 4 or 5 uncertainties. In the perspective of analyzing systems with a larger number of uncertain parameters, a time-consuming and tedious process was carried out. This led to a complete rewrite of the STOWAT, which is now optimized down to the sub-function level, and whose performance is assessed in this paper on a series on benchmarks of increasing complexity with up to about 20 states and 20 uncertainties. This work represents a new step towards the development of a consolidated tool that could reasonably be integrated in the aerospace Verification and Validation process in a near future, finding its place between Monte Carlo simulations – useful for quantifying the probability of sufficiently frequent phenomena – and worst-case $\boldsymbol{\mu}$-analysis – relevant for detecting extremely rare events.

Keywords: uncertain systems, probabilistic $\boldsymbol{\mu}$-analysis, computational tool, aerospace V&V process

1 Introduction

Due to their simplicity, Monte Carlo (MC) simulations [1, 2] have long been the preferred validation means in the aerospace industry. No analytical representation of the system is needed, and the probability that a whole set of requirements are satisfied is easily computed from time-domain simulations, where the uncertain parameters of the system are sampled based on their probability distributions. But such an approach is generally time-consuming, provide only soft bounds [3] and may fail in detecting rare events. In contrast with MC simulations, optimization-based techniques intelligently search the parameter space to find a combination of uncertainties that minimizes or maximises a stability or performance criterion. They are therefore suitable for worst-case analysis problems where the aim is to identify rare but possibly critical events. In particular, stochastic methods such as Genetic Algorithms or Cross Entropy based optimization [4] have been shown to perform well in a wide variety of complex aerospace problems [5–7], although there are no formal convergence proofs. But their main drawback is the lack of probabilistic bounds, such as the Chernoff bound [8], to characterize a confidence level.

On the other hand, less expensive deterministic and simulation-free alternatives exist and have reached a good level of maturity, as is the case for μ-analysis [9–11]. Unlike MC simulations, rare scenarios are no longer missed, but their probability of occurrence is not measured. Moreover, possible correlations between the uncertain parameters are usually ignored, which may lead to consider non-physical representations. Thus, for many problems the worst-case paradigm based on μ can be overly conservative [12] and a control system may be invalidated on the basis of very rare and therefore extremely unlikely events [13–15]. So there is a real need to develop new methods to fill the gap between MC simulations (able to quantify the probability of sufficiently frequent phenomena) and worst-case μ-analysis (relevant to detect extremely rare events). They should be able to detect very rare but nonetheless possible events, and to provide guaranteed bounds on the associated probabilities, which is not possible with Genetic Algorithms or Cross Entropy based optimization. The ambition is to improve the current industrial standard and to fasten the V&V process, which currently accounts for up to 80% of the Attitude and Orbit Control Systems (AOCS) total development time in the space industry.

Researchers started in the 1990s to investigate probabilistic μ-analysis [3, 12, 16], which seeks to combine worst-case bounds determined by μ-analysis with probabilistic information. But although attractive from a theoretical perspective, this approach was far from being applicable at that time, as acknowledged by [12]: *It is still not clear how feasible probabilistic μ is.* The first dedicated software was developed more than twenty years later [13]. It was a major improvement, although still not sufficient to address challenging industrial applications, as highlighted in [14]: *In terms of algorithmic implementation, the situation is similar to when first appeared the preliminary implementations of deterministic μ. That is, in some cases it takes prohibitively*

long and on average takes too long for standard use in the control design cycle – although possibly still acceptable for limited model complexity during analysis in conjunction with Monte Carlo campaigns. The last round began a few years ago, when [17] developed a new Matlab software based on the Skew Mu Analysis Robustness Tools (SMART) Library of the Systems Modeling Analysis and Control (SMAC) Toolbox, that implements state-of-the-art μ-analysis based algorithms to compute robustness margins and performance levels [11]. Shortly afterwards, the first version of the STOchastic Worst-case Analysis Toolbox (STOWAT) was released by [15]. Better integrated with the SMART library, it has been significantly improved since then by [18], thanks to the addition of new features, but above all to increasingly efficient implementations, resulting in a dramatic reduction in CPU time. However, until recently, it could only be reasonably applied to small-scale models, with up to 4 or 5 uncertainties. In the perspective of analyzing systems with a larger number of uncertain parameters, a time-consuming and tedious process was carried out. This led to a complete rewrite of the STOWAT, which is now highly optimized down to the sub-function level, and whose performance is assessed in this paper on a series on benchmarks of increasing complexity. More generally, this work represents a new step towards the development of a consolidated tool that could reasonably be integrated into the aerospace Verification and Validation (V&V) process in a near future. To the best of our knowledge, this is indeed the first time a realistic system with about 20 states and 20 uncertainties – a flexible satellite with sloshing effects in the present case – can be analyzed by a probabilistic μ approach in only a few minutes.

The paper is organized as follows. Section 2 briefly outlines the μ-analysis framework, both from a deterministic and probabilistic point of view. Section 3 presents the latest software tools and shows on a series of benchmarks of increasing complexity how a prototype version limited to academic applications has become in a few years mature enough to integrate an industrial V&V process. These tools are finally applied in Section 4 to the aforementioned satellite benchmark and thoroughly compared to MC simulations.

2 Probabilistic μ-analysis framework

Let us consider the following continuous-time uncertain Linear Time-Invariant (LTI) system:

$$\begin{cases} \dot{x} = A(\delta)x + B(\delta)u \\ y = C(\delta)x + D(\delta)u \end{cases} \tag{1}$$

The **real** uncertain parameters $\delta = (\delta_1, \ldots, \delta_N) \in \mathbb{R}^N$ are **bounded** and belong to the uncertainty domain $\mathcal{D}_\delta = \left[\underline{\delta}_1 \ \overline{\delta}_1\right] \times \cdots \times \left[\underline{\delta}_N \ \overline{\delta}_N\right]$. It is assumed without loss of generality that they are **normalized** (which can always be done using an affine transformation), *i.e.* $\mathcal{D}_\delta = \mathcal{B}_{\boldsymbol{\delta}} = [-1 \ 1]^N$. Moreover, they are **independent random variables**, whose probability density functions are supported on the bounded interval $[-1 \ 1]$. Uniform and truncated normal

distributions are often used in practice, but any other distribution supported on a bounded interval can be used.

Remark 1 Assuming that the uncertainties are bounded and independent is convenient, since the uncertainty domain is simply the N-cube $\mathcal{D}_\delta$. As explained in [15, 18], the proposed V&V tools implement Branch-and-Bound algorithms that partition $\mathcal{D}_\delta$ into smaller and smaller subdomains. If the latter are N-cubes too, the only question is in which directions the division should be performed, which can be solved very quickly by evaluating the μ-sensitivities [19]. Moreover, in many practical applications, these assuptions are reasonable and $\mathcal{D}_\delta$ exactly corresponds to the set of physically meaningful uncertainties. But there also exists cases where the uncertainties are not independent and the corresponding domain has a non-trivial shape. Encompassing this domain in a unique N-cube $\mathcal{D}_\delta$ can therefore lead to consider non-physical models, which might result in the rejection of a satisfactory control architecture. A possible solution is to cover the uncertainty domain with several N-cubes and to apply the proposed V&V tools on each of them. This is equivalent to performing a preliminary partitioning of $\mathcal{D}_\delta$ and setting the probability of the non-physical regions to 0. Another solution is to optimize the shape of the subdomains when applying the B&B algorithm to best fit the exact uncertainty domain, in the spirit of what is done in [20]. However, this optimization process can be quite costly. An intermediate solution consisting in considering simple affine relations between the uncertainties could however allow to adjust the shape of the subdomains using efficient μ-based tools, therefore keeping a reasonable computational time.

It is assumed that $A(\delta)$, $B(\delta)$, $C(\delta)$, $D(\delta)$ are polynomial or rational functions of the δ_i and that system (1) can be transformed into a Linear Fractional Representation (LFR) as in Fig. 1: the uncertainties are separated from the nominal (closed-loop) system $M(s)$ and isolated in a block-diagonal operator $\Delta = \text{diag}(\delta_1 I_{n_1}, \ldots, \delta_N I_{n_N})$, where n_i is the number of ocurrences of δ_i in Δ and I_{n_i} is the $n_i \times n_i$ identity matrix. The question of how this transformation is performed and in which cases exactly it is possible is out of the scope of this paper, but more information can be found in [9]. The set of matrices with the same block-diagonal structure as Δ is denoted $\mathbf{\Delta}$. Let then $k\mathcal{B}_{\mathbf{\Delta}} = \{\Delta \in \mathbf{\Delta} : |\delta_i| \leq k, i \in [1, N]\}$. In the particular case where $k = 1$, $\mathcal{B}_{\mathbf{\Delta}} = \{\Delta \in \mathbf{\Delta} : \delta \in \mathcal{B}_{\boldsymbol{\delta}}\}$ is the subset of $\mathbf{\Delta}$ corresponding to $\mathcal{B}_{\boldsymbol{\delta}}$.

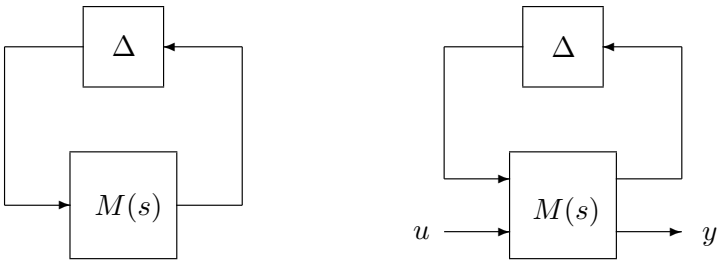

Fig. 1 Standard interconnections for robust stability (left) and performance (right) analysis

With these notations in mind, two main problems can be solved either in a deterministic framework using classical μ-analysis or in a probabilistic framework using probabilistic μ-analysis. Stability is introduced first and the corresponding interconnection is shown in Fig. 1 (left).

Problem 1.1 (Deterministic worst-case stability) *Compute the worst-case stability margin* $k_{wc} = \min\{k \geq 0 : \exists \Delta \in k\mathcal{B}_{\mathbf{\Delta}}, M(s) - \Delta \text{ is unstable}\}$.

Problem 1.2 (Probabilistic robust stability) *Given a desired stability margin* $k \geq k_{wc}$*, compute the probability* $\overline{P}^{k}_{\mathbf{\Delta},f}(M(s))$ *that the interconnection of Fig. 1 (left) is unstable when* $\Delta \in k\mathcal{B}_{\mathbf{\Delta}}$.

Remark 2 The uncertainties δ being normalized, a natural choice is $k = 1$ in Problem 1.2 and the associated probability is simply denoted $\overline{P}_{\mathbf{\Delta},f}(M(s))$.

H_∞ performance is considered next and the corresponding interconnection is shown in Fig. 1 (right). Let $\mathcal{T}_{u\to y}(s,\Delta)$ denote the transfer from u to y. It is assumed that $k_{wc} > 1$, noting that the case where $k_{wc} \leq 1$ can be easily handeled by choosing a subset of $\mathcal{B}_{\mathbf{\Delta}}$ where stability is ensured instead of $\mathcal{B}_{\mathbf{\Delta}}$.

Problem 2.1 (Deterministic worst-case H_∞ performance) *Compute the worst-case performance level* $\gamma_{wc} = \max\limits_{\Delta\in\mathcal{B}_{\mathbf{\Delta}}} \|\mathcal{T}_{u\to y}(s,\Delta)\|_\infty$.

Problem 2.2 (Probabilistic robust H_∞ performance) *Given a desired performance level* $\gamma \in [0\ \gamma_{wc}]$*, compute the probability* $\overline{P}^{\gamma}_{\mathbf{\Delta},f}(M(s))$ *that* $\|\mathcal{T}_{u\to y}(s,\Delta)\|_\infty > \gamma$ *on Fig. 1 (right) when* $\Delta \in \mathcal{B}_{\mathbf{\Delta}}$.

Once computed, the probabilities $\overline{P}^{k}_{\mathbf{\Delta},f}(M(s))$ and $\overline{P}^{\gamma}_{\mathbf{\Delta},f}(M(s))$ can be compared to a given tolerance level ϵ, so as to validate or reject the considered control system, depending on whether they are lower or higher than ϵ.

The theory behind μ-analysis is not presented in this paper due to space limitations, but the interested reader can for example refer to [9, 10, 21] and [13–15, 17, 18] for the classical and the probabilistic versions respectively. Only a few facts are briefly recalled below to facilitate the understanding of Sections 3 and 4.

Classical μ-analysis is based on the computation of the structured singular value $\mu_{\mathbf{\Delta}}$ on the whole frequency range. This computation being NP-hard in general, bounds on k_{wc} and γ_{wc} are usually determined in Problems 1.1 and 2.1 instead of the exact values. Much work has been done in the past decades to reduce the gap between these bounds, and (almost) exact values of k_{wc} and γ_{wc} are now obtained in most cases with a reasonable computational time [22]. The main reason why the gap sometimes remains non-negligible and the computational time significant is the presence of uncertainties repeated many times in Δ, *i.e.* high values of some n_i.

Probabilistic μ-analysis combines the aforementioned μ-based tools with a Branch-and-Bound (B&B) algorithm to explore the whole uncertainty domain $\mathcal{B}_{\boldsymbol{\delta}}$. A simple stability test is first performed at the center of the

domain, *i.e.* for $\Delta = 0$. If the resulting system is stable (resp. unstable), stability (resp. instability) is then investigated on $\mathcal{B}_{\boldsymbol{\delta}}$ using sufficient conditions involving μ upper bound computations. If it cannot be guaranteed on the entire domain, $\mathcal{B}_{\boldsymbol{\delta}}$ is finally partitioned into smaller boxes and this process is repeated until each box has guaranteed stability/instability or is sufficiently small to be neglected (see Algorithm 1 of [15]). This leads to the following partition of the uncertainty domain $\mathcal{B}_{\boldsymbol{\delta}}$:

$$\mathcal{B}_{\boldsymbol{\delta}} = D_s \cup D_{\overline{s}} \cup D_{s_u} \tag{2}$$

where D_s, $D_{\overline{s}}$ and D_{s_u} are three sets of disjoint N-cubes corresponding to the domains where stability is guaranteed, instability is guaranteed and stability is undetermined respectively, with probabilities $p(D_s)$, $p(D_{\overline{s}})$ and $p(D_{s_u})$. The domain D_{s_u} stems from the aforementioned NP-hardness issue, but also from the fact that B&B can only approximate D_s and $D_{\overline{s}}$, and not compute them exactly. The probability $p(D_{s_u})$ can be reduced by increasing the number of iterations of the algorithm, at the price of an increase in the CPU time. Guaranteed lower and upper bounds on the exact probability $\overline{P}_{\boldsymbol{\Delta},f}(M(s))$ of instability are finally obtained as follows, thus solving Problem 1.2:

$$p(D_{\overline{s}}) \leq \overline{P}_{\boldsymbol{\Delta},f}(M(s)) \leq 1 - p(D_s) = p(D_{\overline{s}}) + p(D_{s_u}) \tag{3}$$

Their accuracy depends on the chosen stopping criterion of the B&B algorithm, which allows to handle the trade-off between accuracy and computational time.

Remark 3 When executing the B&B algorithm, the probability distributions of the uncertain parameters δ are only used to decide in which order the boxes are analyzed (usually by decreasing probability). The most relevant way to compare the results obtained with different distributions therefore consists of applying the B&B algorithm only once considering uniform distributions, and then recalculating the bounds of equation (3) for all distributions of interest, which is very fast.

Performance analysis can be done in the same way. On the one hand, a μ-based sufficient condition to guarantee H_∞ performance on a box ($\|\mathcal{T}_{u\to y}(s,\Delta)\|_\infty \leq \gamma$) is easily obtained using the main loop theorem [23]. On the other hand, checking whether non-performance is guaranteed ($\|\mathcal{T}_{u\to y}(s,\Delta)\|_\infty > \gamma$) requires to solve a minimax problem, which cannot be directly reformulated in the μ framework. Nevertheless, a sufficient condition is proposed in [15] (see Proposition 3.1 and Algorithm 2) to overcome this issue. This leads to the following partition of D_s:

$$D_s = D_\gamma \cup D_{\overline{\gamma}} \cup D_{\gamma_u} \tag{4}$$

where D_γ, $D_{\overline{\gamma}}$ and D_{γ_u} correspond to the stability domains where performance is guaranteed, non-performance is guaranteed and performance is undetermined respectively, with probabilities $p(D_\gamma)$, $p(D_{\overline{\gamma}})$ and $p(D_{\gamma_u})$. Guaranteed

bounds on the exact probability $\overline{P}^{\gamma}_{\mathbf{\Delta},f}(M(s))$ of non-performance follow, thus solving Problem 2.2:

$$p(D_{\overline{\gamma}}) \leq \overline{P}^{\gamma}_{\mathbf{\Delta},f}(M(s)) \leq p(D_s) - p(D_{\gamma}) = p(D_{\overline{\gamma}}) + p(D_{\gamma_u}) \tag{5}$$

The only difference with stability is that the investigated domain is limited to the domain of guaranteed stability D_s, since performance analysis only makes sense for stable systems. The following partition of the normalized uncertainty domain $\mathcal{B}_{\boldsymbol{\delta}}$ is finally obtained by combining (2) and (4):

$$\mathcal{B}_{\boldsymbol{\delta}} = D_{\gamma} \cup D_{\overline{\gamma}} \cup D_{\gamma_u} \cup D_{\overline{s}} \cup D_{s_u} \tag{6}$$

In other words, the uncertain system does not meet the performance requirement for a given uncertainty $\Delta \in \mathbf{\Delta}$ if it is unstable or the H_{∞} norm exceeds the desired threshold γ.

3 Computational tools assessment

Problems 1.1 and 2.1 can be solved using the SMART Library of the SMAC Toolbox [11, 21], which was introduced in 2013 and implements state-of-the-art μ-analysis based algorithms to compute robustness margins and performance levels. Based on this library, the STOchastic Worst-case Analysis Toolbox (STOWAT) [15, 18] allows to solve Problems 1.2 and 2.2 by computing *guaranteed* lower and upper bounds on the probabilities $\overline{P}_{\mathbf{\Delta},f}(M(s))$ and $\overline{P}^{\gamma}_{\mathbf{\Delta},f}(M(s))$ with the desired accuracy.

Many improvements have been brought to the STOWAT since the first version presented in [15]. But until recently, it could only be reasonably applied to small-scale models. It generally gave good results in the presence of 4 or 5 uncertainties, but beyond that the computation time increased rapidly if a satisfactory accuracy was required. **In the perspective of analyzing systems with a larger number of uncertain parameters, a time-consuming and tedious process was carried out.** First, a more clever cube splitting technique was implemented in the B&B algorithm. It uses the μ-sensitivities [19], which provide a very efficient way to detect the most critical uncertainties and therefore to decide in which directions to cut the uncertainty domain. Then, the whole Matlab code was investigated to detect and rewrite the most time-consuming parts using differents techniques: vectorization, use of analytical solutions whenever possible (*e.g.* avoid using numerical integration to compute probabilities), store the results of certain calculations to avoid doing them several times, avoid using versatile but usually time-consuming functions, stop the μ computations as soon as accuracy is sufficient to draw a conclusion, use only simple data types such as matrices instead of structure arrays or LTI models... **This led to a complete rewrite of the Matlab code, which is now optimized down to the sub-function level.** This tedious work is the price to pay to be competitive with, or even outperform, simulation-based approaches. The result seems to be well worth the effort, as evidenced

by the following comparison between the successive versions of the tool. Note that the purpose of this section is not to discuss whether probabilistic analysis has added value over worst-case analysis for the considered examples, but to compare the computational efficiency of different versions of the STOWAT for benchmarks of increasing complexity. The added value of probabilistic analysis is shown later in Section 4.3.

Let us first consider the following simple example extracted from [17]:

$$\begin{cases} \dot{x} = \begin{bmatrix} 0 & 1 \\ -a_1(\delta_1) & -a_2(\delta_2) \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \\ y = \begin{bmatrix} 1 & 0 \end{bmatrix} x \end{cases} \tag{7}$$

where $a_1(\delta_1)$, $a_2(\delta_2)$ are two uncertain parameters defined as:

$$\begin{cases} a_1(\delta_1) = 1 + 2\delta_1 \\ a_2(\delta_2) = 0.8 + \delta_2 \end{cases} \tag{8}$$

and δ_1, δ_2 are two normalized real parametric uncertainties with a uniform distribution on $[-1\ 1]$. Four probabilistic μ codes are compared, all implemented with Matlab:

1. the tool described in [17], which is the first to rely on the SMART Library of the SMAC Toolbox,
2. the prototype version (V0) of the STOWAT introduced in 2019 [15], which proposes a totally new implementation as well as additional features such as the determination of $D_{\bar{s}}$ and $D_{\bar{\gamma}}$ in addition to D_s and D_γ,
3. the first consolidated version (V1) of the STOWAT developed under ESA contract RFP/3-16071/19/NL/CRS/hh [18], directly inspired from the prototype one, but further optimized and robustified,
4. the most recent version (V2) of the STOWAT assessed in the present paper, which consists of a brand new code optimized down to the sub-function level as highlighted above.

Codes #1 and #2 have been kindly provided by D. Alazard and S. Thai respectively, while codes #3 and #4 have been implemented by C. Roos and J-M. Biannic. Note that all computational times reported in this paper were obtained using Matlab R2018b runing serially on a single core on a Windows 10 laptop from 2019 with an Intel Core i5-8400H CPU running at 2.5 GHz and 16 GB of RAM.

The four codes are first applied to solve Problem 1.2 (probabilistic robust stability), and results are presented in Fig. 2 and Table 1, where it can be checked that $p(D_s) + p(D_{\bar{s}}) + p(D_{s_u}) = 100\%$ according to equation (2). A drastic reduction of CPU time is observed. The curve in Fig. 2 is indeed almost linear on a logarithmic scale and reveals that the computational effort is divided by about 10 from one version to the next. In particular, the CPU time has been divided by:

- 20 between codes #3 and #4, which shows the relevance of the complete rewrite,
- 1200 since the introduction of the STOWAT in 2019, which makes the most recent version now applicable in an industrial context (see also Table 3 and Section 4).

In the meantime, accuracy has also been improved. Code #1 is not able to quantify $p(D_{\bar{s}})$ and $p(D_{s_u})$. Code #2 can do it, but $p(D_{s_u})$ remains a bit large for such a simple example. Finally, codes #3 to #4 allow to drastically reduce $p(D_{s_u})$ from 0.47% to 0.01%, which means that 99.99% of the uncertainty domain can now be categorized in less than 1s.

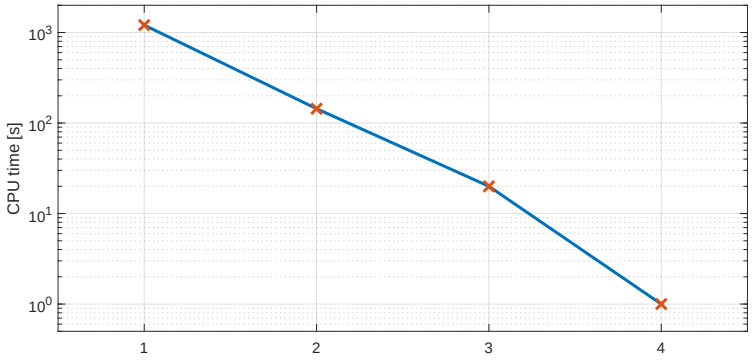

Fig. 2 CPU time for the successive versions of the tool (stability)

Table 1 Numerical results for the successive versions of the tool (stability)

Code #	**1**	**2**	**3**	**4**
CPU time [s]	1208	144	20	<1
$\boldsymbol{p(D_s)}$ **[%]**	92.52	92.59	92.79	92.79
$\boldsymbol{p(D_{\bar{s}})}$ **[%]**	-	6.94	7.20	7.20
$\boldsymbol{p(D_{s_u})}$ **[%]**	-	0.47	0.01	0.01

The four codes are applied next to solve Problem 2.2 (probabilistic robust performance), and results are summarized in Table 2, where it can be checked that $p(D_\gamma) + p(D_{\bar{\gamma}}) + p(D_{\gamma_u}) = p(D_s)$ according to equation (4). The trend is the same as before, and a drastic reduction of CPU time is observed. In the meantime, accuracy is also improved, with $p(D_{\gamma_u})$ decreasing from one version to the next. The decrease in computational time allows to go further in the analysis by choosing a refined stopping condition for the B&B algorithm. This is illustrated in the last column of Table 2, which shows that $p(D_{\gamma_u})$ can be divided by almost 6 with a CPU time of only 18s, and in Fig. 3, where the domains of undetermined stability D_{s_u} or performance D_{γ_u} are hardly visible.

Codes #3 (V1) and #4 (V2) of the STOWAT are now applied to a series of benchmarks of increasing complexity available in the litterature and implemented in the SMAC Toolbox [22]. Only probabilistic stability analysis

Table 2 Numerical results for the successive versions of the tool (performance)

Code #	**1**	**2**	**3**	**4**	
CPU time [s]	3370	132	27	< 2	18
$p(D_\gamma)$ [%]	41.30	41.30	41.33	41.57	42.30
$p(D_{\bar{\gamma}})$ [%]	-	48.68	49.40	49.39	50.18
$p(D_{\gamma_u})$ [%]	-	2.61	2.06	1.82	0.32

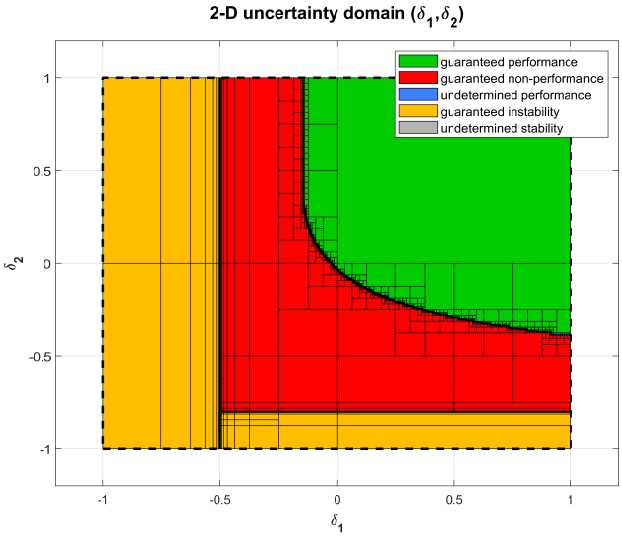

Fig. 3 Partition of the uncertainty domain with a refined stopping criterion

(Problem 1.2) is investigated here, but the same trend can be observed for performance analysis. These test cases are characterized by various fields of application, system dimensions and structures of the uncertainties. Some of them contain poorly damped modes, which usually produce extremely sharp peaks on the μ plot, while others have large state vectors as well as numerous and/or highly repeated uncertainties. All results are gathered in Table 3, whose second column deserves some explanation:

- 3/3 means that Δ is composed of 3 non-repeated uncertain parameters,
- 4/5 means that Δ is composed of 3 non-repeated uncertain parameters and 1 repeated twice,
- the drive-by-wire vehicle has 2 non-repeated uncertain parameters and 7 repeated twice,
- the reentry vehicle has 3 non-repeated uncertain parameters and 2 repeated 4 and 6 times,
- the hard-disk drive has 19 non-repeated uncertain parameters and 4 repeated twice.

The following test protocol is applied to each benchmark. Using the stability margin computed in [22], the bounds on the uncertain parameters are first

modified to get $k_{wc} = 0.95$ in Problem 1. This ensures that the uncertain system is not stable on the whole set $\mathcal{B}_{\mathbf{\Delta}}$, but the probability $\overline{P}_{\mathbf{\Delta},f}(M(s))$ of instability is sufficiently small to justify the use of probabilistic μ. The most recent version of the STOWAT (code #4) is then applied (column 6 of Table 3) with a computational time of 10 s as stopping criterion of the B&B algorithm. And the previous version of the STOWAT (code #3) is finally applied (column 5) with the resulting value of $p(D_{s_u})$ as stopping criterion (column 4) to be able to compare the computational time of both versions for the same accuracy.

The results are consistent with those obtained previously on the academic model (7)-(8). Columns 5 and 6 indeed show that the CPU time is divided by about 10 on average (8.7 to be exact) thanks to the complete rewrite. It remains low even for high-order systems, and a good accuracy – measured here by the value of $p(D_{s_u})$ in column 4 – is obtained in all cases. In view of all these results, it appears that the new version of the STOWAT has reached a good level of maturity, which makes it possible to use it on realistic industrial applications, as shown in Section 4.

Let us now analyze Table 3 a little further. As expected, good results are obtained even for systems with large state vectors. μ-analysis is indeed a frequency-domain approach, and the size of the frequency response of the nominal closed-loop system $M(s)$, *i.e.* the size of the matrix $M(j\omega)$, does not depend on the order of the system but only on the size of the uncertainty matrix Δ. It could however seem more surprising that there is no significant increase in computational time or decrease in accuracy when the number of uncertainties increases. The explanation lies in the fact that at each iteration, the B&B algorithm implemented in the STOWAT divides the uncertainty domain along the direction corresponding to the highest μ-sensitivity [19], *i.e.* to the uncertainty with the greatest influence on stability. With such a strategy, the limiting factor of the proposed approach is not the total number of uncertainties as one might think, but the number of uncertainties that most impact stability or performance. It is therefore preferable to have a system with 20 uncertainties, only 4 of which have a high sensitivity, than a system with 8 uncertainties of equivalent sensitivity. Fortunately, it seems that the number of critical uncertainties usually remains quite reasonable, even when the total number of uncertainties is large, as shown in Table 4. Here, code #4 is applied with a computational time of 100 s as stopping criterion of the B&B algorithm. For each benchmark, all the N-cubes that form the resulting partition (2) of the uncertainty domain are investigated to determine the maximum number of times the range of each uncertainty was cut (column 3). A large number means that the corresponding uncertainty was often the one with the highest μ-sensitivity, and is therefore a measure of its influence on stability. It appears that the number of critical uncertainties for which the number of cuts is greater than or equal to half the largest value (column 4) does not significantly increase with the complexity of the benchmark.

Table 3 Application of the STOWAT to benchmarks of increasing complexity

Benchmark		**Number of uncertainties and size of Δ**	**Number of states**	**Accuracy $p(D_{s_u})$[%]**	**CPU time [s] Code #3**	**CPU time [s] Code #4**
1	Inverted pendulum [24]	3/3	4	0.0195	173.2	10.0
2	Anti-aliasing filter [25]	4/5	2	0.0022	87.0	10.0
3	Bank-to-turn missile [26]	4/4	6	0.0313	93.5	10.0
4	Cassini spacecraft [27]	4/4	17	0.0132	55.3	10.0
5	Drive-by-wire vehicle [28]	9/16	4	0.0244	75.7	10.0
6	Reentry vehicle [29]	5/13	7	0.0054	64.7	10.0
7	Space shuttle [30]	9/9	34	0.0549	80.9	10.0
8	Hard-disk drive [31]	23/27	29	0.0027	64.3	10.0

Table 4 Number of critical uncertainties with a significant influence on stability

Benchmark	Number of uncertainties	Maximum number of cuts	Number of critical uncertainties
1	3	$12-11-8$	3
2	4	$10-9^2-8$	4
3	4	$10-8-7-6$	4
4	4	15^2-4-1	2
5	9	$7-6-4^2-3^4-2$	4
6	5	8^4-7	5
7	9	$11-9-2^2-0^5$	2
8	23	$5-4^3-3^2-2^3-11^2-0^{12}$	6

It must however be kept in mind that an unfavorable configuration will inevitably occur at some point, which will put the analysis tools in trouble due to the exponential-time behavior of the B&B algorithm. To illustrate this behavior, code #4 is run again on all benchmarks without using the μ-sensitivities. This means that all uncertainties are treated as if they had the same influence on stability, so that the uncertainty domain is divided equally in all directions. Two stopping criteria are considered: first a CPU time of 10 s and then the same accuracy $p(D_{s_u})$ as in Table 3. Results are presented in columns 3 and 5 of Table 5 respectively. Columns 2 and 4 correspond to the case where the μ-sensitivities are used. They are similar to column 4 and 6 of Table 3, but they have been included to make the comparison easier. It can be seen that the accuracy and the computational time is roughly the same as before for benchmarks 1-2-6, and sometimes even lower since the calculation time of the μ-sensitivities is saved. For these benchmarks, Table 4 indeed shows that the uncertainties almost all have the same influence on stability. The results are a little less satisfactory for benchmark 3 (highlighted in orange). This is once again consistent with Table 4, where all uncertainties appear to have a non-negligible influence on stability, but some more than others. Finally, the computational effort soars for benchmarks 4-5-7-8, since problems with 4-9-9-23 uncertainties respectively are solved, instead of only 2-4-2-6 when the μ-sensitivities were used.

Table 5 Accuracy and CPU time with/without using the μ-sensitivities

Benchmark	Accuracy $p(D_{s_u})$[%] for a CPU time of 10 s		CPU time [s] for the accuracy of Table 3	
	with	without	with	without
1	0.0195	0.0232	10	18.0
2	0.0022	0.0019	10	6.4
3	0.0313	0.0735	10	138.8
4	0.0132	0.3906	10	> 3600
5	0.0244	13.9648	10	> 3600
6	0.0054	0.0204	10	19.3
7	0.0549	6.2500	10	> 3600
8	0.0027	2.9297	10	> 3600

4 Application to a realistic satellite benchmark

4.1 Open-loop satellite modeling

Without a significant loss of generality, since coupling effects are often negligible, a single-axis satellite model is considered here. Let θ be the attitude angle of the satellite around the considered axis and $J = 1000\,kg.m^2$ its main moment of inertia. The rigid dynamic equation simply reduces to:

$$J\ddot{\theta} = T_W + T_S + T_F + T_D \tag{9}$$

where T_W is the control input torque, T_S is the torque produced by the propellant slosh effects, T_F is induced by the flexible modes of the solar arrays attached to the main body, and T_D captures all remaining disturbance torques (possibly produced by the solar pressure but also by an embedded robotic arm dedicated to on-orbit services). As is usual in the literature, slosh and flexible effects are represented by poorly damped second-order linear models:

$$T_S + T_F = \sum_i \frac{L_i s^2}{s^2 + 2\xi_i \omega_i s + \omega_i^2} \ddot{\theta} \tag{10}$$

where the parameters L_i, ω_i and ξ_i respectively denote the magnitude, the frequency and the damping of each mode. The nominal values are presented in Table 6 and the corresponding open-loop Bode plot of the transfer from T_W to θ is visualized in Fig. 4.

Table 6 Slosh & flexible modes coefficients (nominal values)

Mode #i	1	2	3	4	5
L_i **[kg.m²]**	30	40	50	300	100
ω_i **[rad/s]**	0.1	0.2	0.3	0.6	1
ξ_i	5×10^{-3}	4×10^{-3}	3×10^{-3}	1×10^{-3}	1×10^{-3}

Remark 4 As is further detailed in [32], a more realistic control-oriented representation of the slosh effects is obtained with implicitly time-varying frequency and damping characteristics. The latter generally depend indeed on the angular velocity and acceleration of the satellite. Such variations have been omitted here to generate an uncertain but invariant model as imposed by the μ-analysis framework.

Finally, the control input torque T_W is generated from the commanded torque T_C by a reaction wheel, which can be approximated by a first-order linear model with time constant $\tau = 0.5\,\mathrm{s}$:

$$T_W = \frac{1}{1 + \tau s} T_C \tag{11}$$

Note that this actuator is also rate and magnitude limited. These properties are taken into account in the design phase but not introduced here since the considered maneuvers are assumed to be saturation free.

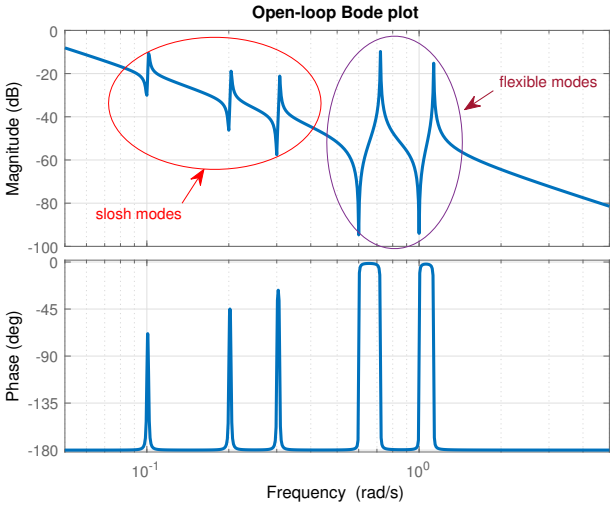

Fig. 4 Bode plot of the nominal open-loop transfer function of the satellite model from the control input torque T_W to the attitude angle θ (with 3 slosh modes and 2 flexible modes)

4.2 Structured H_∞-based robust controller design

A robust attitude tracking controller is designed for the above satellite model. As illustrated in Fig. 5, a second-order reference model $R(s)$ with desired frequency $\omega_r = 0.05\,rad/s$ and damping $\xi_r = 0.75$ is introduced. This model delivers reference signals θ_r and $\dot{\theta}_r$ which are compared to the outputs of the plant. As is clearly visible on the diagram, an extended PID structure is imposed on the controller $K(s)$ which minimizes the H_∞ norm of the weighted transfer from the exogenous inputs $w_1 = \theta_c$ and $w_2 = T_D$ to the exogenous outputs $z_1 = \epsilon = \theta_r - \theta$, its integral z_2 and $z_3 = T_C$. The weighting functions are tuned so that:

- the optimized controller exhibits good robustness properties against actuator delay (delay margin should be above $0.45\,s$),
- the H_∞ norm of the transfer $\mathcal{T}_D(s) = \mathcal{T}_{T_D \to \epsilon}(s)$ from disturbance torque inputs T_D to the attitude error ϵ should ideally not exceed 0.01.

The last objective which ensures good tracking performance despite perturbations is hard to achieve simultaneously with the delay margin constraint. After a reasonably short tuning process, a fifth-order controller $K(s)$ was obtained with the `systune` routine from the Matlab Robust Control Toolbox. This controller exhibits a very good delay margin ($0.45\,s$) and acceptably meets the performance requirement with $\|\mathcal{T}_D(s)\|_\infty = 1.01 \times 10^{-2}$.

Moreover, parametric robustness properties against possible variations of the inertia, the actuator time constant and the 15 parameters of Table 6 are partly ensured via a multi-model design approach. There is however no guarantee that the whole uncertainty domain is cleared. It is indeed numerically

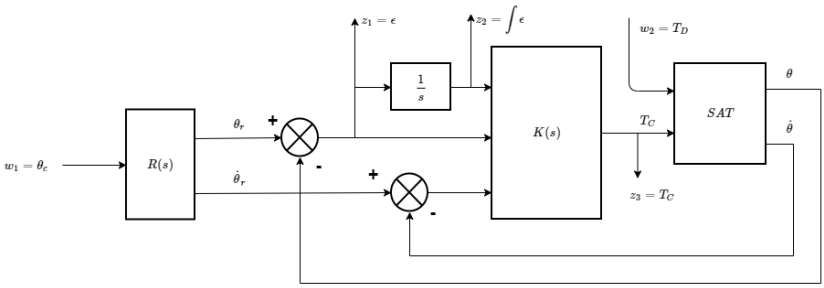

Fig. 5 H_∞ design model

impossible here to include all parametric variations in the design process since there are two many parameters. A validation phase to be performed *a posteriori* is then required.

4.3 Control system validation

Let us now evaluate the pointing performance properties of the closed-loop system by confronting the latest version of the STOWAT (code #4 in Section 3) to a classical Monte Carlo (MC) type approach based on random sampling. Real parametric uncertainties are introduced on the inertia J of the satellite and the time constant τ of the reaction wheel, as well as the magnitudes L_i, damping factors ξ_i and frequencies ω_i of the 5 slosh and flexible modes. An uncertain system with 21 states and 17 uncertainties is therefore obtained. Problem 1.1 is solved using the SMART Library, which shows that stability is guaranteed when all uncertainties are allowed to vary independently of each other by $\pm 11.34\%$ around their nominal values. The main objective is now to determine if the pointing performance degradation with respect to the nominal case is acceptable when the uncertain parameters vary by $\pm 11\%$. For this, the transfer $\mathcal{T}_D(s, \Delta)$ between the external disturbance torque T_D and the attitude error ϵ is considered. As already mentioned in Section 4.2, the nominal H_∞ norm without uncertainties is $\gamma_{nom} = \|\mathcal{T}_D(s, 0)\|_\infty = 1.01 \times 10^{-2}$, which indicates a good rejection of the perturbation in the absence of uncertainties.

Robust performance is first analyzed using MC. $M = 10^5$ samples are generated for each uncertain parameter, following a uniform distribution on an interval of $\pm 11\%$ around their nominal value. The highest H_∞ norm among these 10^5 samples is $\gamma_{mc} = 1.11 \times 10^{-2}$ and is achieved near the frequency $\omega_s = 0.5\, rad/s$. This suggests that performance is only minimally affected by the uncertainties. The Bode plots of the nominal system and the 100 samples with the highest H_∞ norm are shown in red and blue respectively in Fig. 6. There is a fairly large dispersion on the frequency, but much less on the magnitude, which is consistent with the previous results. All computations are performed in $1300\, s$.

The algorithm introduced in [33] and implemented in the SMART Library is now used to address Problem 2.1, and more precisely to compute a lower

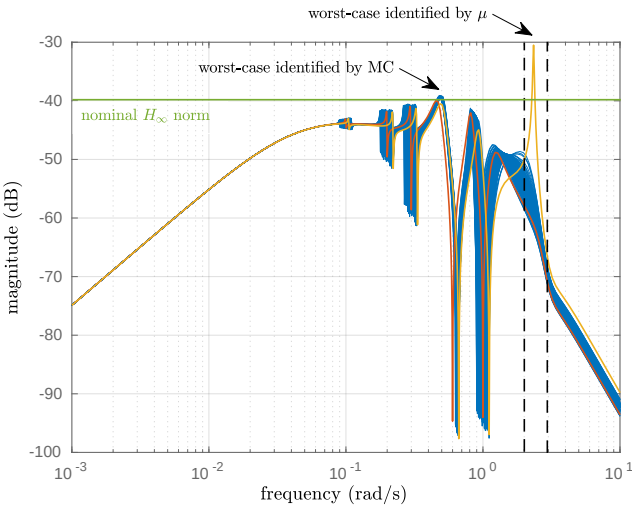

Fig. 6 Bode plot of the transfer $\mathcal{T}_D(s,\Delta)$ between the external disturbance torque and the attitude error (red = nominal, blue = 100 worst samples, yellow = worst-case identified by μ-analysis)

bound on the worst-case H_∞ norm γ_{wc} of the transfer $\mathcal{T}_D(s,\Delta)$. This is a 2-step procedure. Coarse lower bounds are first computed quickly for 100 frequencies covering the entire frequency range. A series of linear programming problems are then solved with respect to both frequency and uncertainties using the previous results as an initialization to get a refined bound, as well as the associated worst-case uncertainty combination. All the calculations take only $9\,s$ and results are partially reproduced in Table 7. It can be seen that in most cases, a lower bound of about 1.1×10^{-2} is obtained close to ω_s, which is in line with the results obtained previously by sampling. But grid point #93 reveals a much higher bound $\gamma_{wc} \geq 2.89\times10^{-2}$ at the frequency $\omega_c = 2.346\,rad/s$, which is missed by the sampling approach despite the fact that 10^5 samples are considered. The Bode plot of this worst-case configuration is plotted in yellow in Fig. 6. It is worth noting that ω_c is quite different from ω_s. Moreover, it does not coincide with any frequency of the slosh and flexible modes (0.1, 0.2, 0.3, 0.6 and $1\,rad/s$). It was therefore not possible before applying μ-analysis to anticipate this phenomenon, which can have a significant impact on the pointing performance. Let us indeed consider a sinusoidal disturbance $T_D(t) = \sin(\omega_c\,t)$. The resulting pointing error is plotted in Fig. 7 with the same color code as above. For the worst-case configuration identified by the SMART Library, it is about 15 times and 30 times larger than for the worst random sample and the nominal configuration respectively.

A critical issue is therefore to compute the probability of observing such a performance degradation, so as to decide whether the proposed control system is reliable enough or not. The latest version of the STOWAT introduced in Section 3 is applied for this purpose. As with the previous sampling approach, all uncertainties follow a uniform distribution. But the frequency range on

Table 7 Worst-case performance analysis results

Grid point	Initial frequency [rad/s]	Final frequency [rad/s]	Lower bound on $\|\mathcal{T}_D(s,\Delta)\|_\infty$
1	0.000	0.475	1.10×10^{-2}
2	0.004	0.483	1.10×10^{-2}
91	1.772	0.489	1.10×10^{-2}
92	2.070	0.484	1.10×10^{-2}
93	2.418	2.346	2.89×10^{-2}
94	2.825	0.476	1.10×10^{-2}
99	6.141	0.491	1.11×10^{-2}
100	7.173	0.499	1.11×10^{-2}

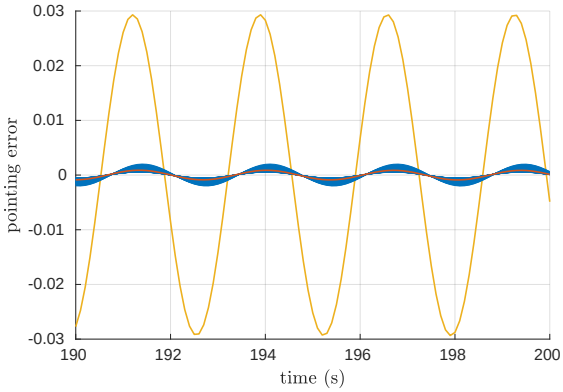

Fig. 7 Pointing error in the presence of a sinusoidal disturbance $T_D(t) = \sin(\omega_c t)$ (red = nominal, blue = 100 worst samples, yellow = worst-case identified by μ-analysis)

which performance is investigated is now limited to the interval [2 3] rad/s (represented by vertical dashed lines in Fig. 6) to focus on the worst-case peak identified above. In this context, let:

$$\overline{P}^{\gamma}_{[2\ 3]} = \mathcal{P}\left(\|\mathcal{T}_D(s,\Delta)\|_{\infty,[2\ 3]} > \gamma\right) \tag{12}$$

be the probability that the frequency-limited H_∞ norm of $\mathcal{T}_D(s,\Delta)$ on [2 3] rad/s is higher than γ. Problem 2.2 is solved with a desired performance level $\gamma = \gamma_{nom} = 1.01 \times 10^{-2}$, which corresponds to the nominal H_∞ norm on the whole frequency range (represented by a horizontal green line in Fig. 6). In other words, the objective is to compute guaranteed bounds on $\overline{P}^{\gamma_{nom}}_{[2\ 3]}$. Two stopping criteria are defined for the B&B algorithm, corresponding to different levels of accuracy: the probability of undetermined performance $p(D_{\gamma_u})$ should be lower than 10^{-3} % (resp. 10^{-4} %). After about 100 s (resp. 1400 s), the probabilities $p(D_\gamma) = 99.999\%$, $p(D_{\overline{\gamma}}) = 0\%$ and $p(D_{\gamma_u}) = 10^{-3}$ % (resp. $p(D_\gamma) = 99.9999\%$, $p(D_{\overline{\gamma}}) = 0\%$ and $p(D_{\gamma_u}) = 10^{-4}$ %) of guaranteed performance, guaranteed non-performance and undetermined performance are obtained. This means according to (5) that $\overline{P}^{\gamma}_{[2\ 3]}$ is upper bounded by 10^{-3} %

(resp. 10^{-4} %). A very large part of the uncertainty domain can therefore be cleared in a very reasonable computational time.

The results obtained by MC are now analyzed from a statistical point of view to allow a comparison with probabilistic μ. The classical Chernoff bound [8, 34] is first applied, noting that all 10^5 samples satisfy $\|\mathcal{T}_D(s,\Delta)\|_{\infty,[2\ 3]} \leq \gamma_{nom}$. For any $\epsilon \in [0\ 1]$ and $\delta \in [0\ 1]$:

$$M \geq \frac{\ln(2/\delta)}{2\epsilon^2} \quad \Rightarrow \quad \mathcal{P}\left(\overline{P}_{[2\ 3]}^{\gamma_{nom}} \leq \epsilon\right) \geq 1-\delta \tag{13}$$

where M denotes the number of samples. Assuming for simplicity that $\delta = \epsilon$, $M = 10^5$ leads to $\epsilon = 5.435 \times 10^{-3}$, *i.e.* 0.5435 %. The bound introduced in [35] is computed next. For any $\epsilon \in [0\ 1]$ and $\delta \in [0\ 1]$:

$$M \geq \frac{\ln(1/\delta)}{\ln(1/(1-\epsilon))} \quad \Rightarrow \quad \mathcal{P}\left(\overline{P}_{[2\ 3]}^{\gamma_{mc,[2\ 3]}} \leq \epsilon\right) \geq 1-\delta \tag{14}$$

where $\gamma_{mc,[2\ 3]}$ is the largest value among the 10^5 samples of the frequency-limited H_∞ norm of $\mathcal{T}_D(s,\Delta)$ on $[2\ 3]$ rad/s, equal to 3.55×10^{-3} $(-49\,dB)$ as can be seen in Fig. 6. Assuming again that $\delta = \epsilon$, $M = 10^5$ leads to $\epsilon = 9.284 \times 10^{-5}$, *i.e.* $9.284 \times 10^{-3} \approx 10^{-2}$ %. Accuracy is therefore much better than with the Chernoff bound. Moreover, as $\gamma_{mc,[2\ 3]} < \gamma_{nom}$, it can be concluded that $\overline{P}_{[2\ 3]}^{\gamma}$ is upper bounded by 10^{-2} % with a probability of $1-\delta \approx 99.99$ %. This result shows that in this example, probabilistic μ is 10 (resp. 100) times more accurate than MC when the computational time is 10 times lower (resp. equivalent). Moreover, the bounds provided by probabilitic μ are guaranteed, whereas those provided by MC come with a confidence level.

Consider now a more realistic case where the uniform distribution is replaced with a normal distribution with zero mean and variance σ^2. All probabilities are quickly recomputed in a few seconds using the STOWAT for several values of σ^2 between 0.1 and 5, without performing the whole analysis again (see Remark 3). In contrast, applying MC requires to generate 10^5 samples and compute the associated H_∞ norm for each considered value of σ^2, thus significantly increasing the computational time. The results are plotted in Fig. 8. They show that $p(D_{\gamma_u})$ increases when σ^2 increases, the case of a uniform distribution being recovered when $\sigma \to \infty$ (the probability density function of the truncated normal distribution becomes constant on the considered support). For example, if $\sigma^2 = 0.2$, the probability of non-performance $\overline{P}_{[2\ 3]}^{\gamma_{nom}}$ is no larger than 1.8×10^{-6} %. And results are even more accurate if $\sigma^2 = 0.1$, since $\overline{P}_{[2\ 3]}^{\gamma_{nom}}$ is bounded by 1.3×10^{-9} %, which is negligible. The performance degradation shown in Fig. 6 and 7 is therefore very unlikely in these cases. So if the system is well enough identified to ensure that σ remains quite small, there is certainly no need to invalidate the control system on the basis of this extremely rare worst-case event. But if σ is larger, performance degradation cannot be ignored, and either a better controller should be designed or the system should

be better identified. In contrast, the accuracy of MC is not influenced by the value of σ in this application. No value of Δ has indeed been found for which $\|\mathcal{T}_D(s,\Delta)\|_{\infty,[2\ 3]} > \gamma_{nom}$. Equations (13) and (14) therefore still hold, as well as the accuracy reported in the previous paragraph, *i.e.* $\overline{P}_{[2\ 3]}^{\gamma_{nom}} \leq 10^{-2}$ with a probability of 99.99%. This realistic example clearly shows the added value of probabilistic μ in detecting rare events and to quantify their probability with precision. The latter is indeed faster and more accurate than MC simulations on this realistic application, but also less conservative than deterministic μ-analysis in the sense that it does not systematically lead to rejecting a controller on the basis of a single (and possible extremely rare) scenario. Moreover, the computational time remains particularly low – a few seconds to build Fig. 8 – considering the complexity of the model, which contains almost 20 uncertainties.

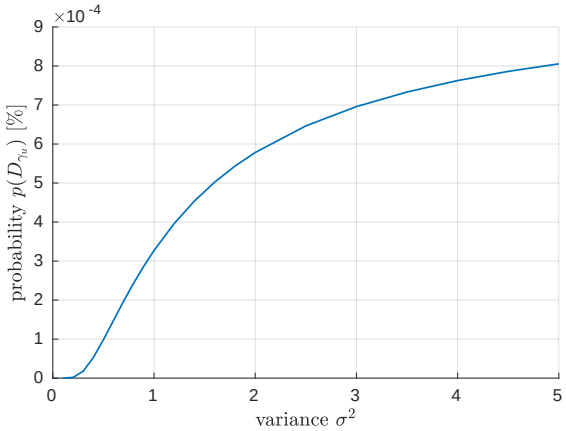

Fig. 8 Probability of undetermined performance as a function of the variance σ^2

It should however be kept in mind that probabilistic μ is dedicated to the analysis of rare events, and it is in this specific case that it outperforms MC. Accuracy and computational efficiency decrease when the probabilities of the considered scenarios increase, and MC usually ends up being more competitive. Moreover, MC analysis does not depend on the number of uncertain parameters and has a wider range of applications including nonlinear and time-varying systems. So the idea is certainly not to totally replace MC with probabilistic μ, but rather to use probabilistic μ during the early stages of control laws validation to speed up the process and reduce the number of MC simulations by better targeting them, as highlighted at the end of [18]. Probabilistic μ could also be incorporated into an iterative control laws design process, where the controller would be designed not only based on the nominal system behavior, but also on critical scenarios with a sufficiently high probability identified by probabilistic μ. Structured H_∞ control [36] would be a good candidate, since existing tools of the Matlab Robust Control Toolbox such as `hinfstruct` and `systune` already implement a multi-model approach.

5 Conclusion

The combination of state-of-the-art algorithms and a highly optimized implementation reveals the interesting potential of probabilistic μ to evaluate the robustness properties of complex systems. The latest version of the Matlab STOchastic Worst-case Analysis Toolbox indeed allows to assess the pointing performance of a flexible satellite subject to slosh effects in the presence of about 20 parametric uncertainties, whereas previously existing tools could hardly handle more than 4 or 5. While it is always possible to further optimize the code, the effort tends to become inversely proportional to the benefit, and it is questionable whether it is worth it. More promising directions to make the analysis faster would be to use parallel computing, and to improve the B&B strategies to explore the uncertainty domain more efficiently. Research is currently underway and suggests that by combining the two, it should be possible to further reduce the computational time by a significant amount, leading to a third generation of the STOWAT capable of handling even more complex systems.

The implementation effort reported in this paper paves the way to the integration of probabilistic μ in the aerospace V&V process in complement to traditional MC-based techniques. Probabilistic μ has indeed the potential to speed up the first validation steps by quickly invalidating a contol system or conversely clearing a large part of the uncertainty domain. The outcomes of this preliminary analysis could then be used to better orientate the MC simulations and/or reduce their number, therefore reducing the total computational time. It should indeed be kept in mind that simulations become unavoidable beyond a certain level of complexity, due to their ability to consider time-varying parameters and non-linearities in addition to uncertainties.

Declarations

Funding. The research leading to these results received funding from the French Centre National d'Etudes Spatiales (CNES) under Grant Agreement R&T CNES R-S20/BS-0005-073.

Conflict of interest. The authors declare that they have no conflict of interests.

References

[1] Helton, J., Johnson, J., Sallaberry, C., Storlie, C.: Survey of sampling-based methods for uncertainty and sensitivity analysis. Reliability Engineering and System Safety **91**, 1175–1209 (2006)

[2] Landau, D., Binder, K.: A Guide to Monte Carlo Simulations in Statistical Physics. Cambridge University Press (2005)

[3] Zhu, X., Huang, Y., Doyle, J.: Soft vs. hard bounds in probabilistic robustness analysis. In: Proceedings of the IEEE Conference on Decision and Control, pp. 3412–3417 (1996)

[4] de Boer, P., Kroese, D., Mannor, S., Rubinstein, R.: A tutorial on the Cross-Entropy method. Annals of Operations Research **134**, 19–67 (2005)

[5] Morio, J., Balesdent, M.: Estimation of Rare Event Probabilities in Complex Aerospace and Other Systems: a Practical Approach. Woodhead Publishing (2015)

[6] Wang, W., Menon, P., Bates, D., Ciabuschi, S., Gomes Paulino, N., Di Sotto, E., Bidaux, A., Kron, A., Salehi, S., Bennani, S.: Verification and validation framework for autonomous rendezvous systems in terminal phase. Journal of Spacecraft and Rockets **52**(2), 625–629 (2015)

[7] Mujumdar, A., Menon, P., Roux, C., Bennani, S.: Cross-entropy based probabilistic analysis of VEGA launcher performance. In: Bordeneuve-Guibé, J., Drouin, A., Roos, C. (eds.) Advances in Aerospace Guidance, Navigation and Control, pp. 719–737. Springer (2015)

[8] Chernoff, H.: A measure of asymptotic efficiency for test of hypothesis based on the sum of observations. Annals of Mathematical Statistics **23**, 493–507 (1952)

[9] Zhou, K., Doyle, J., Glover, K.: Robust and Optimal Control. Prentice-Hall (1996)

[10] Ferreres, G.: A Practical Approach to Robustness Analysis with Aeronautical Applications. Kluwer Academic (1999)

[11] Roos, C.: Systems Modeling, Analysis and Control (SMAC) toolbox: an insight into the robustness analysis library. In: Proceedings of the IEEE International Symposium on Computer-Aided Control System Design, pp. 176–181 (2013). Available with the SMAC Toolbox at w3.onera.fr/smac/smart

[12] Khatri, S., Parrilo, P.: Guaranteed bounds for probabilistic μ. In: Proceedings of the IEEE Conference on Decision and Control, pp. 3349–3354 (1998)

[13] Balas, G., Seiler, P., Packard, A.: Analysis of an UAV flight control system using probabilistic μ. In: Proceedings of the AIAA Guidance, Navigation, and Control Conference (2012)

[14] Marcos, A., Bennani, S., Roux, C.: Stochastic μ-analysis for launcher thrust vector control systems. In: Proceedings of the CEAS EuroGNC

Conference (2015)

[15] Thai, S., Roos, C., Biannic, J.-M.: Probabilistic μ-analysis for stability and H_∞ performance verification. In: Proceedings of the American Control Conference, pp. 3099–3104 (2019)

[16] Zhu, X.: Improved bounds computation for probabilistic μ. In: Proceedings of the American Control Conference, pp. 4336–4340 (2000)

[17] Falcoz, A., Alazard, D., Pittet, C.: Probabilistic μ-analysis for system performances assessment. IFAC-PapersOnLine **50**(1), 399–404 (2017). Proceedings of the 20th IFAC World Congress

[18] Biannic, J.-M., Roos, C., Bennani, S., Boquet, F., Preda, V., Girouart, B.: Advanced probabilistic μ-analysis techniques for AOCS validation. European Journal of Control **62**, 120–129 (2021)

[19] Braatz, R., Morari, M.: μ-sensitivities as an aid for robust identification. In: Proceedings of the American Control Conference, pp. 231–236 (1991)

[20] Seiferth, D., Diepolder, J., Afonso, R., Holzapfel, F.: Probabilistic μ-analysis using mapped uncertainties. In: Proceedings of the American Control Conference, pp. 3670–3676 (2021)

[21] Roos, C., Lescher, F., Biannic, J.-M., Döll, C., Ferreres, G.: A set of μ-analysis based tools to evaluate the robustness properties of high-dimensional uncertain systems. In: Proceedings of the IEEE Multiconference on Systems and Control, pp. 644–649 (2011)

[22] Roos, C., Biannic, J.-M.: A detailed comparative analysis of all practical algorithms to compute lower bounds on the structured singular value. Control Engineering Practice **44**, 219–230 (2015)

[23] Packard, A., Doyle, J.: The complex structured singular value. Automatica **29**(1), 71–109 (1993)

[24] Yazıcı, A., Karamancıoğlu, A., Kasimbeyli, R.: A nonlinear programming technique to compute a tight lower bound for the real structured singular value. Optimization and Engineering **12**(3), 445–458 (2011)

[25] Seiler, P., Packard, A., Balas, G.: A gain-based lower bound algorithm for real and mixed μ problems. Automatica **46**(3), 493–500 (2010)

[26] Wise, K.: Comparison of six robustness tests evaluating missile autopilot robustness to uncertain aerodynamics. AIAA Journal of Guidance, Control and Dynamics **15**(4), 861–870 (1992)

[27] Ly, J., Chiang, R., Goh, K., Safonov, M.: LMI multiplier K_m/μ analysis

of the Cassini spacecraft. International Journal of Robust and Nonlinear Control **8**(2), 155–168 (1998)

[28] Halton, M., Hayes, M., Iordanov, P.: State-space μ analysis for an experimental drive-by-wire vehicle. International Journal of Robust and Nonlinear Control **18**(9), 975–992 (2008)

[29] Biannic, J.-M., Roos, C. Benchmark developed by ONERA (The French Aerospace Lab) in cooperation with industrial partners (2013)

[30] Balas, G., Doyle, J., Glover, K., Packard, A., Smith, R.: Mu-analysis and synthesis toolbox for use with Matlab. User's guide, version 3 (2001)

[31] Gu, D., Petkov, P., Konstantinov, M.: Robust Control Design with Matlab. Advanced Textbooks in Control and Signal Processing. Springer (2005)

[32] Biannic, J.-M., Bourdelle, A., Evain, H., Moreno, S., Burlion, L.: On robust LPV-based observation of fuel slosh dynamics for attitude control design. IFAC-PapersOnLine **52**(28), 170–175 (2019). Proceedings of the 3rd IFAC Workshop on Linear Parameter-Varying Systems

[33] Roos, C.: A practical approach to worst-case H_∞ performance computation. In: Proceedings of the IEEE Multiconference on Systems and Control, Yokohama, Japan, pp. 380–385 (2010)

[34] Wozencraft, J., Jacobs, I.: Principles of Communication Engineering. Wiley (1967)

[35] Tempo, R., Bai, E., Dabbene, F.: Probabilistic robustness analysis: explicit bounds for the minimum number of samples. In: Proceedings of the IEEE Conference on Decision and Control, pp. 3424–3428 (1996)

[36] Apkarian, P., Noll, D.: Nonsmooth H_∞ synthesis. IEEE Transactions on Automatic Control **51**(1), 71–86 (2006)