Proceedings of the 35th European Safety and Reliability & the 33rd Society for Risk Analysis Europe Conference
Edited by Eirik Bjorheim Abrahamsen, Terje Aven, Frederic Bouder, Roger Flage, Marja Ylönen
 Published by Research Publishing, Singapore.
doi: 10.3850/978-981-94-3281-3_ESREL-SRA-E2025-P7226-cd

Adaptive Gaussian process-based strategies for solving the NASA-DNV UQ challenge 2025

Julien Demange-Chryst, Nathalie Bartoli, Sylvain Dubreuil, Jérôme Morio
ONERA/DTIS, Université de Toulouse, Toulouse, F-31055, France. E-mail: firstname.lastname@onera.fr
Fédération ENAC ISAE-SUPAERO ONERA, Université de Toulouse, 31000, Toulouse, France.

Mathieu Balesdent, Loïc Brevault
ONERA/DTIS, Université Paris Saclay, Palaiseau, F-91123, France. E-mail: firstname.lastname@onera.fr

This paper describes a dedicated approach to solve the 2025 NASA-DNV UQ challenge problem using adaptive Gaussian process strategies. The uncertainty model is determined through a calibration problem using an optimization approach to identify the aleatory variable joint distribution and the epistemic variable uncertainties. The estimation of the prediction interval for the model output components consists of a quantile estimation problem based on an adaptive Gaussian process strategy. Eventually, the design optimization problems are solved using Bayesian optimization controlling the noise level involved in the estimation of the objective and constraint functions.

Keywords: Uncertainty quantification, optimization, Gaussian process, active learning.

1. Introduction

This paper presents different methodologies to solve the two problems of the 2025 NASA-DNV UQ challenge. These methods mostly rely on adaptive Gaussian process strategies. In a first section, the computational model is briefly described. Then, in Section 2.1, the uncertainty model is determined based on a calibration problem using an optimization strategy to determine both the joint distribution of the aleatory variables and the best estimate of the epistemic variables. Then, in Section 2.2, an adaptive Gaussian process strategy is used to determine the bounds of the prediction intervals for each component of the simulation model outputs. Eventually, in Section 3, the design optimization problems are solved relying on Bayesian optimization while controlling the noise level involved in the estimation of the objective and constraint functions.

The computational model is a black-box function $f(\cdot)$ (Figure 1) that takes as inputs a vector $\mathbf{X}$ and a seed number s, and retrieves as output a multivariate time series $\mathbf{Y}$. The input vector $\mathbf{X}$ is composed of three vectors $\mathbf{X} = (\mathbf{X}_a, \mathbf{X}_e, \mathbf{X}_c)$ where:

- $\mathbf{X}_a \in [0,1]^{n_a}$ is the vector of aleatory uncertainties with $n_a = 2$,
- $\mathbf{X}_e \in [0,1]^{n_e}$ is the vector of epistemic uncertainties with $n_e = 3$,
- $\mathbf{X}_c \in [0,1]^{n_c}$ is the vector of control variables ($n_c = 3$) that can be chosen by the analyst.

The output of the computational model is a multivariate time-series of dimension 6 defined as $\mathbf{Y}(\mathbf{X}_a, \mathbf{X}_e, \mathbf{X}_c, \omega, t) = [Y_1(\mathbf{X}_a, \mathbf{X}_e, \mathbf{X}_c, \omega, t), \ldots, Y_6(\mathbf{X}_a, \mathbf{X}_e, \mathbf{X}_c, \omega, t)]$. The time-series is discretized in 60 nodes (assuming $t \in [1, 60]$ without loss of generality). Furthermore, the simulation model is supposed to be stochastic. The stochasticity of this simulator is emulated through an aleatory variable ω that belongs to a probability space $(\Omega, \mathcal{F}, P_\omega)$. It is assumed that an unknown aleatory integer seed s of a random generator governs the stochastic nature of the model. Eventually, the output of the system can be written as $\mathbf{Y}(\mathbf{X}_a, \mathbf{X}_e, \mathbf{X}_c, s, t) \in \mathbb{R}^{6\times 60}$.

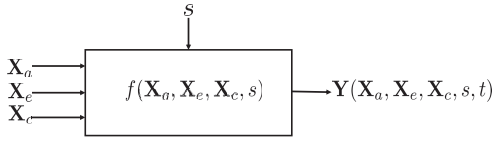

Fig. 1. Computational model.

2. Problem 1: uncertainty quantification

2.1. *Determination of the uncertainty model*

In this section, a calibration method is proposed to identify a probabilistic model of the input aleatory and epistemic uncertainties $\mathbf{X}_a$ and $\mathbf{X}_e$ considering a set of observations provided by NASA and DNV. The goal is to find the joint distribution of $\mathbf{X}_a$, named f_a and the value of epistemic uncertainty vector $\mathbf{X}_e^*$ that best fits the given observations.

2.1.1. *Data analysis and overall methodology*

A preliminary data analysis has been carried out to analyze the influence of all the inputs, *i.e.*, $\mathbf{X}_a$, $\mathbf{X}_e$, $\mathbf{X}_c$ and s on the different time-series output components. For that purpose, assuming an independent uniform distribution in $[0, 1]$ for all the uncertain continuous input variables and a uniform discrete distribution for s, a large sample has been generated and propagated through the local model. The resulting time-series output components have been analyzed using sensitivity analysis. This analysis allowed to point out that the seed has the greatest impact on the output trajectories, and plays a major role in shaping the output time-series (relative evolution with respect to time). The other variables ($\mathbf{X}_a$, $\mathbf{X}_e$ and $\mathbf{X}_c$) mainly act on the magnitude of the time-series but not on its relative evolution. It has also been observed that a large number of the output time-series realizations are "saturated" for the first three output components *i.e*, they are constant to a maximum or a minimum value for all the time nodes.

In the literature, for inverse problem (also called calibration problem) it is possible to distinguish two main approaches Lee et al. (2019): calibration using optimization techniques and Bayesian techniques. As the seed (representing the stochasticity of the simulator) has been identified as the main contributor on the evolution of the output components, Bayesian inference Marin (2007) seems very difficult to apply. In this work, a dedicated methodology based on an optimization technique has been set up to accurately estimate f_a (joint distribution of $\mathbf{X}_a$) and $\mathbf{X}_e^*$ depending on the observed data. The method is composed of four steps that are described in detail in the following:

(1) determination of informative values of $\mathbf{X}_c$,
(2) identification of the seed values corresponding to the available observations,
(3) identification of the values of $\mathbf{X}_a$ and $\mathbf{X}_e$ for each observation,
(4) inference of the uncertainty model for the identified values of epistemic and aleatory uncertainties.

2.1.2. *Determination of informative* $\mathbf{X_c}$

The goal of this step is to find values of $\mathbf{X}_c$ that are informative in order to infer f_a and $\mathbf{X}_e^*$ based on available observations. An initial distribution is considered for all the input uncertain variables under the form of independent uniform distributions. A large stratified Monte Carlo sample is generated according to the initial distribution in which a first Design of Experiments (DoE) of $\mathbf{X}_c$ has been sampled and for each realization of $\mathbf{X}_c$, a sample is generated for $\mathbf{X}_a$, $\mathbf{X}_e$ and s. Then, this sample is propagated through the local model. The best $\mathbf{X}_c$ (named $\mathbf{X}_c^*$) is selected as the one minimizing the number of saturated output stochastic process trajectories and maximizing the diversity of trajectories (maximizing the standard deviation σ of the trajectories) expressed as follows :

$$\mathbf{X}_c^* = \underset{\mathbf{X}_c}{\text{argmax}}\ \mathbb{E}_t\left[\sigma_{\mathbf{X}_a,\mathbf{X}_e,s}\left[\mathbf{Y}(\mathbf{X}_a,\mathbf{X}_e,\mathbf{X}_c,s,t)\right]\right]$$

This process allows to select $\mathbf{X}_c^* = [0.773, 0.256, 0.820]^T$. The obtained trajectories with the local model for the second component of $\mathbf{Y}$ are illustrated in Figure 2.

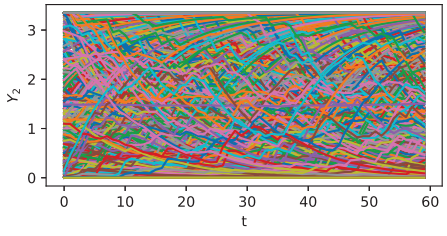

Fig. 2. Local model realizations for the second component of the output for $\mathbf{X}_c^* = [0.773, 0.256, 0.820]^T$.

2.1.3. *Identification of the seed values*

For the chosen $\mathbf{X}_c^*$, 100 observations of the output (called in the following exact observations) have been provided by NASA and DNV. Then, a method to identify the corresponding seed value $s_{i\in[1,\ldots,100]}$ for each of the $i = 1, \ldots, 100$ observations has been set up. This consists in finding the seed value s_i that matches to the temporal dynamics of the exact observations. For one observation, this temporal dynamics may be synthesized through the time positions of the local minima and maxima of the output. To identify the seed values, a complete enumeration of one million of seed values from 0 to 10^6 has been generated, with a fixed value for $\mathbf{X}_a$, $\mathbf{X}_e$ and $\mathbf{X}_c^*$. The local model has been evaluated to determine the corresponding outputs. Then, for each exact observation, a seed value s_i has been determined by finding in the 10^6 local model output realizations, the one that perfectly matches the time position of the local extrema (minima and maxima). From the DoE of seeds, only one perfect correspondence per exact observation has been identified s_i^*, confirming that the seed is the main contributor to the

temporal dynamics of the outputs. For this identification, the fourth component output of the exact observations has been selected because it does not present any saturation. Example of identification is provided at the top of Figure 3.

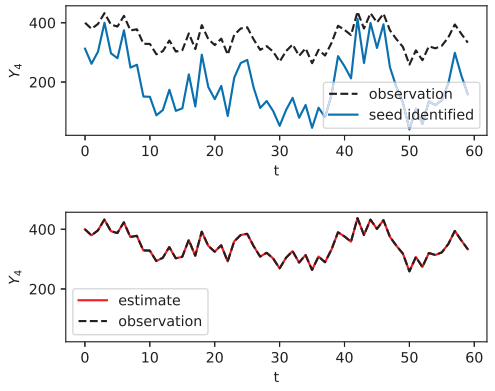

Fig. 3. Exact observation of the output fourth component (black dotted line), local model output corresponding to an identified seed (blue) and estimate output by optimization of $\mathbf{X_a}$ and $\mathbf{X_e}$ for the identified seed (red).

2.1.4. *Determination of $\mathbf{X}_a$ and $\mathbf{X}_e^*$*

To identify the values of $\mathbf{X}_a$ and $\mathbf{X}_e^*$ for each of the exact observations and identified seeds $s^*_{i\in[1,\dots,100]}$, a weighted least-squares optimization has been carried out using Covariance Matrix Adaptation - Evolution Strategies algorithm Hansen et al. (2003). Weights have been introduced to scale the different components of the observations. This calibration through an optimization technique allows to identify one value for $\mathbf{X}_e^*$ and a set of values of $\mathbf{X}_a$ (one per each exact observation). This process enables to find values of $\mathbf{X}_a$ and for $\mathbf{X}_e^*$ in order to perfectly match the exact observations, as illustrated in Figure 4. Finally, all the optimizations converged to the same value of $\mathbf{X}_e = [0.333, 0.597, 0.375]^T$ defining $\mathbf{X}_e^*$. The estimated output corresponding to the observation illustrated in Section 2.1.3 is provided at the bottom of Figure 3.

2.1.5. *Inference of f_a, the joint distribution of $\mathbf{X_a}$*

The calibration process through optimization described in Section 2.1.4 has been replicated for 4 different values of $\mathbf{X}_c$ obtained with optimized Latin Hypercube Sampling, to get sufficient data to estimate the joint distribution of $\mathbf{X}_a$. The estimated 400 samples of $\mathbf{X}_a$ are illustrated in Figure 5. Then, margins and copula of the joint distribution of $\mathbf{X}_a$ have been estimated using OpenTURNS library Baudin et al. (2015). For the margins, among the 30 tested parametric distributions, the Weibull minimum extreme value distribution has been selected as fitting the best to the data, according to Bayesian Informative Criterion Schwarz (1978). The parameters of the identified margins for X_{a1} and X_{a2} are given in Table 1. Alternatively,

Table 1. Values of Weibull distribution hyperparameters for X_{a1} and X_{a2}.

Parameters	Values	95% confidence interval
β_1	0.302	[0.261 ,0.335]
α_1	2.478	[2.057, 2.956]
γ_1	0.002	[0.000, 0.034]
β_2	0.197	[0.178, 0.213]
α_2	1.706	[1.511, 1.885]
γ_2	0.005	[0.000, 0.0165]

non parametric distributions have been tested, providing the same level of confidence. Concerning the copula, the independent copula has reached the best score in Bayesian Informative Criterion, so the margins have been considered as independent. The joint distribution of f_a has then been validated using quantile-quantile plot as illustrated in Figure 6 for the first margin of f_a. The knowledge on f_a could be consolidated by the acquisition of additional exact observations for other $\mathbf{X}_c$ values.

The proposed approach based on the identification on the seed value associated to each trajectory might not be easily generalized to problems that do not have this particular behavior. As mentioned earlier, a more generalizable method based on Bayesian inference could be developed with a particular focus on the stochastic nature of the model.

2.2. *Prediction intervals of output components*

In this section, a method to estimate the bounds of the prediction interval for each of the 6 components of the model output is proposed. For a given confidence level $\alpha \in]0, 1[$, the bounds

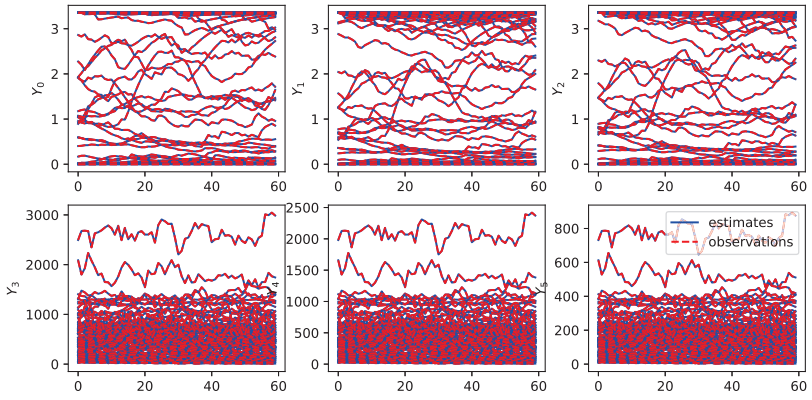

Fig. 4. Estimations of trajectories for $\mathbf{X}_c^*$ by a calibration approach (blue lines) and the corresponding exact observations (red dotted lines) provided by the online exact model. The calibration approach allows to perfectly match to the exact observations.

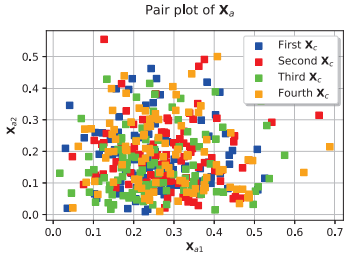

Fig. 5. 400 identified samples of $\mathbf{X}_a$ for the 4 different values of $\mathbf{X}_c$.

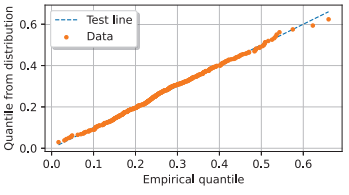

Fig. 6. Quantile-quantile plot for the first margin of f_a.

are defined by:

$$\bar{y}(\mathbf{X}_c, \alpha) = \inf\{u \,|p_u(\mathbf{X}_e, \mathbf{X}_c, u) \leq 1-\alpha\}$$
$$\underline{y}(\mathbf{X}_c, \alpha) = \sup\{l \,|p_l(\mathbf{X}_e, \mathbf{X}_c, l) \leq 1-\alpha\},$$

with $p_u(\mathbf{X}_e, \mathbf{X}_c, u) = \mathbb{P}(Y_M > u)$ and $p_l(\mathbf{X}_e, \mathbf{X}_c, l) = \mathbb{P}(Y_m < l)$, and where $Y_M := \max_{0\leq t\leq 1} y(\mathbf{X}_a, \mathbf{X}_e, \mathbf{X}_c, s, t)$ and $Y_m := \min_{0\leq t\leq 1} y(\mathbf{X}_a, \mathbf{X}_e, \mathbf{X}_c, s, t)$ are introduced to lighten the notations. In other words, for all $t \in [0,1]$, $y(\mathbf{X}_a, \mathbf{X}_e, \mathbf{X}_c, s, t)$ belongs to the interval $[\underline{y}(\mathbf{X}_c, \alpha), \bar{y}(\mathbf{X}_c, \alpha)]$ with probability $2\alpha - 1$.

For the following developments, the epistemic vector $\mathbf{X}_e$ is fixed at $\mathbf{X}_e^*$ as explained and justified in Section 2.1.4. Moreover, the baseline design is set at $\mathbf{X}_c = [0.533, 0.666, 0.5]^T$.

2.2.1. *Quantile estimation problem*

For the fixed value $\mathbf{X}_e^*$, the estimation of the bounds $\bar{y}(\mathbf{X}_c, \alpha)$ and $\underline{y}(\mathbf{X}_c, \alpha)$ is equivalent to a quantile estimation problem. Indeed, for all $u \in \mathbb{R}$, it is possible to rewrite $p_u(u)$ as:

$$p_u(u) = \mathbb{P}(Y_M > u) = 1 - F_{Y_M}(u),$$

where F_{Y_M} stands for the CDF of Y_M. Then:

$$\begin{aligned}\{u \mid p_u(u) \leq 1-\alpha\} &= \{u \mid 1 - F_{Y_M}(u) \leq 1-\alpha\} \\ &= \{u \mid F_{Y_M}(u) \geq \alpha\} \\ &= F_{Y_M}^{-1}([\alpha, 1]).\end{aligned}$$

Thus, the upper bound $\bar{y}(\mathbf{X}_c, \alpha) = \inf F_{Y_M}^{-1}([\alpha, 1])$ is exactly the quantile of order α of the random variable Y_M.

Similarly, $p_l(l)$ can be written for all $l \in \mathbb{R}$ as:

$$p_l(l) = \mathbb{P}(Y_m < l) = F_{Y_m}(l) - \mathbb{P}(Y_m = l),$$

where F_{Y_m} stands for the CDF of Y_m. Then:

$$\{l \mid p_l(l) \leq 1 - \alpha\} = \{l \mid F_{Y_m}(l) - \mathbb{P}(Y_m = l) \leq 1 - \alpha\}.$$

It can be shown that the lower bound $\underline{y}(\mathbf{X}_c, \alpha)$, which is defined as the supremum of this set, exactly coincides with the quantile of order $1 - \alpha$ of the random variable Y_m, unless if F_{Y_m} is constant exactly at level $1 - \alpha$. A visual inspection of the empirical CDF associated to each trajectory provided in Figure 7 strongly suggests that F_{Y_m} is strictly increasing at level $1 - \alpha$. In that case, $\underline{y}(\mathbf{X}_c, \alpha)$ is exactly the quantile of order $1 - \alpha$ of the random variable Y_m. In summary, estimating

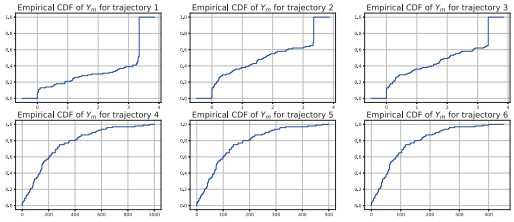

Fig. 7. Empirical CDF of the random variable Y_m associated to each output component. Each of them has been computed with the available sample from the provided exact dataset.

the bounds of the prediction interval consists in estimating the quantiles of order α of Y_M and of order $1 - \alpha$ of Y_m.

2.2.2. *AK-MCS algorithm*

Since evaluating the numerical model is quite computationally intensive, the quantiles of interest are estimated by combining both Monte Carlo simulation and Kriging metamodeling Williams and Rasmussen (2006). Indeed, the proposed procedure is an adaptation to the case of quantile estimation of the AK-MCS algorithm Echard et al. (2011).

The AK-MCS algorithm consists in iteratively enrich a conditioned Gaussian process until reaching a stopping criterion. The latter quantifies the quality of the approximation of the metamodel near the limit state, such that it can discriminate with high accuracy failed samples from safe ones. The main difference with the initial AK-MCS algorithm designed for rare event estimation is that the limit state parameter is the quantile itself Schöbi et al. (2017), which is re-estimated after each iteration.

For the estimation of the upper bound $\bar{y}(\mathbf{X}_c, \alpha)$, the Gaussian process built during the AK-MCS algorithm is an approximation of the function: $\mathbf{X}_a \in [0,1]^2 \mapsto Y_M(\mathbf{X}_a)$, where $Y_M(\mathbf{X}_a)$ has been averaged over the random seed for all $\mathbf{X}_a \in [0,1]^2$, *i.e.*:

$$Y_M(\mathbf{X}_a) = \frac{1}{n_{\text{rep}}} \sum_{i=1}^{n_{\text{rep}}} Y_M\left(\mathbf{X}_a, s^{(i)}\right)$$

and with $\left(s^{(i)}\right)_{i \in [\![1, n_{\text{rep}}]\!]}$ an i.i.d. n_{rep}-sample of random seeds. In other words, a double loop over first the random seed and second $\mathbf{X}_a$ instead of a single loop over the pair $(\mathbf{X}_a, s)$ is performed in order to estimate the probabilities of exceedance $p_u(u)$. This choice has been made because it provides more stable and consistent numerical results. Furthermore, the estimation error of $Y_M(\mathbf{X}_a)$ caused by the Monte Carlo approximation is ignored. Thus, the function to approximate is considered as deterministic. The same procedure is performed with Y_m for the estimation of the lower bound $\underline{y}(\mathbf{X}_c, \alpha)$.

2.2.3. *Numerical results*

For the following experiments, the OpenTURNS library Baudin et al. (2015) is used to draw the initial DoE and the Monte Carlo sample, and the SMT toolbox Saves et al. (2024) is used to build the heteroscedastic Gaussian processes.

First, the estimation of the bounds of the prediction intervals for output components 4, 5 and 6 has been done with the AK-MCS algorithm. To do so, an initial DoE of size $N_{\text{DoE}} = 10$ is generated by Latin Hypercube Sampling (LHS) to build an initial Gaussian process. Then, the latter is iteratively enriched until the minimum of the U-function, defined in Echard et al. (2011), evaluated on the Monte Carlo sample exceeds a given threshold ($U \geq 2.0$). The estimated bounds for $\alpha = 0.95$ and $\alpha = 0.999$ are given in Table 2. Second, a more direct analysis can be done for output components 1, 2 and 3. Indeed, as shown in Figure 7, these components have a high saturation rate at 3.35 (more than 30%). Thus, a straightforward estimator of the upper bound $\bar{y}(\mathbf{X}_c, \alpha)$ for output components 1, 2 and 3 is 3.35 for both $\alpha = 0.95$ and $\alpha = 0.999$. Moreover, the minimum of more than 10% of the trajectories for output components 2 and 3 is equal to 0. So a straightforward estimator of the lower bound $\underline{y}(\mathbf{X}_c, \alpha)$ for output components 2 and 3 is 0. for both $\alpha = 0.95$ and $\alpha = 0.999$. At last, the available dataset from the real system is not informative enough for estimating $\underline{y}(\mathbf{X}_c, 0.95)$ for output trajectory 1 because the first non-zero value is very

Table 2. Bounds of the prediction intervals for output components 4, 5 and 6.

α	Output component	$\underline{y}(\mathbf{X}_c, \alpha)$	$\bar{y}(\mathbf{X}_c, \alpha)$
0.95	4	14.43	981.30
	5	7.11	482.27
	6	6.01	407.70
0.999	4	3.93×10^{-3}	1824.76
	5	1.99×10^{-3}	896.86
	6	1.65×10^{-3}	758.16

close to quantile level $1 - 0.95$. So the AK-MCS has been used to estimate this bound. In summary, the estimated bounds for both $\alpha = 0.95$ and $\alpha = 0.999$ for output components 1, 2 and 3 are given in Table 3.

Table 3. Bounds of the prediction intervals for output components 1, 2 and 3.

α	Output component	$\underline{y}(\mathbf{X}_c, \alpha)$	$\bar{y}(\mathbf{X}_c, \alpha)$
0.95	1	7.63×10^{-3}	3.35
	2	0.	3.35
	3	0.	3.35
0.999	1	0.	3.35
	2	0.	3.35
	3	0.	3.35

3. Problem 2: Design optimization

Different optimization problems have to be solved in order to identify the optimal control parameter $\mathbf{X}_c$ with respect to different objective functions and constraints. In these three problems, objective and constraint functions are either expected values or probability that can only be estimated. In the following these estimations are performed by sampling approaches (Monte Carlo, AK-MCS, *etc*.) leading to noisy evaluation of the objective function and constraints. This estimation noise is an epistemic uncertainty that can be reduced by adding more samples to the estimator *i.e.*, increasing the dimension of the Monte Carlo sample. In order to handle these optimization problems under epistemic uncertainty, the strategy developed in Dubreuil et al. (2020) is retained. The first optimization problem can be written as:

$$\mathbf{X}_c^* = \underset{\mathbf{X}_c}{\text{argmin}} \; -J(\mathbf{X}_e, \mathbf{X}_c)$$

where the objective function is given by,

$$J(\mathbf{X}_e, \mathbf{X}_c) = \int_0^1 \sum_{i \in I_1} \mathbb{E}[y_i(\mathbf{X}_a, \mathbf{X}_e, \mathbf{X}_c, s, t)]\mathrm{d}t,$$

where $\mathbb{E}[\cdot]$ is the expected operator with respect to the random vector $\mathbf{X}_a$, the seed s and I_1 the set of index of the output components 1, 2 and 3. In the following, the problem is simplified by assuming a fixed value for $\mathbf{X}_e$ (as specified in Section 2.1.4), therefore, the dependence w.r.t. $\mathbf{X}_e$ is omitted. A prerequisite for solving the optimization problems is the ability to estimate the value of $J(\cdot)$ for a given $\mathbf{X}_c \in [0,1]^3$. To do so, the objective function $J(\cdot)$ is rewritten as:

$$J(\mathbf{X}_c) = \mathbb{E}\left[\int_0^1 y_{I_1}(\mathbf{X}_a, \mathbf{X}_c, s, t)\,\mathrm{d}t\right],$$

with $y_{I_1} = y_1 + y_2 + y_3$. This alternative writing allows to use Monte Carlo integration to estimate $J(\mathbf{X}_c)$ as:

$$\widehat{J}^{(n)}(\mathbf{X}_c) = \frac{1}{n}\sum_{i=1}^{n} \int_0^1 y_{I_1}\left(\mathbf{X}_a^{(i)}, \mathbf{X}_c, s^{(i)}, t\right)\mathrm{d}t,$$

where $\left(\mathbf{X}_a^{(i)}\right)_{i \in [\![1,n]\!]} \sim f_a$ is an i.i.d. n-sample and where $\left(s^{(i)}\right)_{i \in [\![1,n]\!]}$ is an i.i.d. n-sample of random seeds. Furthermore, the integral of y_{I_1} with respect to $t \in [0,1]$ is estimated by the trapeze method using the discrete evaluations $(y_{I_1}^{(i)})_{i \in [\![1,m]\!]}$ (where $m = 60$ is the number of time steps in the numerical simulation). At last, the central limit theorem provides the asymptotic behavior of the estimator:

$$\sqrt{n}\left(\widehat{J}^{(n)}(\mathbf{X}_c) - \mu_J(\mathbf{X}_c)\right) \xrightarrow[n \to +\infty]{d} \mathcal{N}(0, \sigma_J(\mathbf{X}_c)),$$

where:

$$\mu_J(\mathbf{X}_c) = J(\mathbf{X}_c)$$
$$\sigma_J(\mathbf{X}_c) = \sqrt{\mathbb{V}\left(\int_0^1 y_{I_1}(\mathbf{X}_a, \mathbf{X}_c, s, t)\,\mathrm{d}t\right)},$$

and $\mathbb{V}[\cdot]$ the variance operator. Therefore, $\widehat{J}^{(n)}(\mathbf{X}_c)$ can be considered as a Gaussian random variable of mean $\mu_J(\mathbf{X}_c)$ and standard deviation $\sigma_{\widehat{J}^{(n)}}(\mathbf{X}_c) = \sigma_J(\mathbf{X}_c)/\sqrt{n}$ for n large enough. The optimization problem is thus:

$$\mathbf{X}_c^* = \underset{\mathbf{X}_c}{\text{argmin}} \; -\mu_J(\mathbf{X}_c). \tag{1}$$

Obtaining an accurate estimation $\mu_J(\mathbf{X}_c)$ (*i.e.*, with a low $\sigma_{\widehat{J}^{(n)}}(\mathbf{X}_c)$) can be quite computationally intensive as it may require a large sample size n. The ability of the optimization algorithm to handle epistemic noise and its capacity to focus the computational budget on promising area of the design

space leads us to choose Bayesian optimization Williams and Rasmussen (2006) and more precisely the algorithm developed in Dubreuil et al. (2020) designed to iteratively decreased the epistemic noise $\sigma_{\widehat{J}^{(n)}}(\mathbf{X}_c)$ only in promising $\mathbf{X}_c$ points *i.e.* the MC sample size n is increased adaptively if needed (see the following description). The main steps of the proposed algorithm are described in the following:

(1) Build a DoE by using LHS of size N, for each point $\mathbf{X}_{ci} \in [0,1]^3$ associate $(\mu_J(\mathbf{X}_{ci}), \sigma_{\widehat{J}^{(n)}}(\mathbf{X}_{ci}))\quad \forall i = 1,\dots,N$ and compute the coefficient of variation $cv(\widehat{J}^{(n)}(\mathbf{X}_{ci})) = \sigma_{\widehat{J}^{(n)}}(\mathbf{X}_{ci})/\mu_J(\mathbf{X}_{ci})$.

(2) Build a heteroscedastic Gaussian Process regression model (GP) Williams and Rasmussen (2006) from which one can obtain a predicted mean value $\mu_{\text{GP}}(\mathbf{X}_c)$ and a predicted variance $\sigma^2_{\text{GP}}(\mathbf{X}_c), \forall\, \mathbf{X}_c \in [0,1]^3$.

(3) For each DoE point $\mathbf{X}_{ci}$, compute the probability, $P_{\min}(\mathbf{X}_{ci})$, that the point $\mathbf{X}_{ci}$ solves the optimization problem given by Eq. 1. These probabilities are computed by Monte Carlo sampling and the numerical cost associated is negligible as it only involves the resolution of a discrete optimization problem over the DoE.

(4) Identify within the DoE the interesting points, denoted by the set $(\mathbf{X}_c)^{\text{DoE}}_{\min}$, with high probability $P_{\min}$ (defined in practice by $P_{\min}\left[(\mathbf{X}_c)^{\text{DoE}}_{\min}\right] \geq 1/N$), and with high accuracy estimate defined by a threshold value ε_{cv} of the coefficient of variation $cv(\widehat{J}^{(n)}((\mathbf{X}_c)^{\text{DoE}}_{\min}))$. At this step the local numerical model is called until $cv(\widehat{J}^{(n)}((\mathbf{X}_c)^{\text{DoE}}_{\min})) \leq \epsilon_{cv}$. Usually this enrichment leads to the identification of a unique point in the set $(\mathbf{X}_c)^{\text{DoE}}_{\min}$ *i.e.*, only one point of the DoE satisfies both $P_{\min}[\mathbf{X}_c] \geq 1/N$ and $cv(\widehat{J}^{(n)}(\mathbf{X}_c)) \leq \epsilon_{cv}$). If it is not the case, either the problem has equal local minima or the threshold value ϵ_{cv} is too large to identify a unique global minimum. In the following a unique value of $(\mathbf{X}_c)^{\text{DoE}}_{\min}$ is needed to define the infill criterion, in practice the one with the highest $P_{\min}\left[(\mathbf{X}_c)^{\text{DoE}}_{\min}\right]$ is retained.

(5) Maximize the infill criterion (for instance the Expected Improvement Williams and Rasmussen (2006)) computed analytically from the GP surrogate (step 2) and $(\mathbf{X}_c)^{\text{DoE}}_{\min}$ (step 4) to identify a potential candidate: $\mathbf{X}^*_c$

$$\text{EI}(\mathbf{X}) = (\mu_{\text{GP}}(\mathbf{X}) - (\mathbf{X}_c)^{\text{DoE}}_{\min})\Phi(Z) + \sigma_{\text{GP}}(\mathbf{X})\phi(Z)$$

where $Z = \frac{\mu_{\text{GP}}(\mathbf{X}) - (\mathbf{X}_c)^{\text{DoE}}_{\min}}{\sigma_{\text{GP}}(\mathbf{X})}$, $\Phi(\cdot)$ and $\phi(\cdot)$ are respectively the cumulative and probability density functions of $\mathcal{N}(0,1)$.

(6) Compute $J(\mathbf{X}^*_c)$ and update the DoE (return to step 2).

(7) Stop the algorithm after a fixed number of iterations.

For the numerical experiments, the SMT toolbox Saves et al. (2024) is used to build the heteroscedastic Gaussian processes.

3.1. *Optimization problem 1 for performance-based design*

The performance-based design $\mathbf{X}^*_c$ is solution of

$$\mathbf{X}^*_c = \underset{\mathbf{X}_c}{\text{argmin}}\ -J(\mathbf{X}_{e\text{fixed}}, \mathbf{X}_c). \tag{2}$$

Beginning with five initial DoE consisting of 10 points each, five runs are performed with 30 iterations and $\varepsilon_{cv} = 0.3\%$. The convergence plots are given in the Figure 8 and the identified optimal points are presented in the Table 4. For each of these five points, $cv(\widehat{J}^{(n)}(\mathbf{X}^*_{\mathbf{c}})) = 0.24\%$, indicating that it is challenging to determine the optimal value of $J(\cdot)$.

Table 4. Optimal values for $\mathbf{X}^*_c$ and $J(\mathbf{X}_{e\text{fixed}}, \mathbf{X}^*_c)$.

X_{c1}	X_{c2}	X_{c3}	$J_{\min}$
0.162	0.414	0.558	-7.215
0.139	0.369	0.657	-7.235
0.684	0.847	0.792	-7.211
0.710	0.337	0.649	-7.262
0.137	0.818	0.641	-7.247

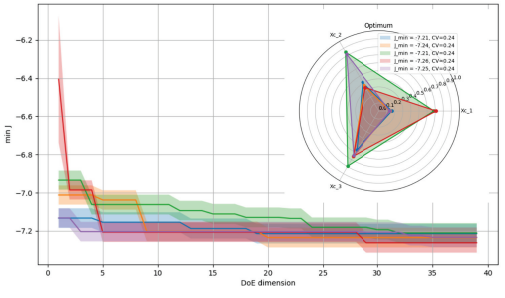

Fig. 8. Convergence plots of five runs for problem 2.1, featuring a 99% confidence interval of for the J value, alongside the associated $\mathbf{X}^*_c$ values on the radar plot.

3.2. *Optimization problem 2 for reliability-based design.*

The reliability-based design $\mathbf{X}_c^*$ is given by

$$\mathbf{X}_c^* = \underset{\mathbf{X}_c}{\text{argmin}}\ \text{pof}_{\text{sys}}(\mathbf{X}_{e\,\text{fixed}}, \mathbf{X}_c)$$
$$\text{with} \quad \text{pof}_{\text{sys}}(\mathbf{X}_e, \mathbf{X}_c) = \mathbb{P}[\min_{i \in I_2} g_i(\mathbf{X}_a, \mathbf{X}_e, \mathbf{X}_c, s)],$$

where I_2 is the set of index of the output components 4, 5 and 6.

The same Bayesian optimization strategy is used to solve this problem. Figure 9 presents the results obtained by two different runs (different initial DoE). Both converge towards a probability of failure of order 10^{-7} for the same $\mathbf{X}_c^* \approx [0.37, 0.60, 1.0]$.

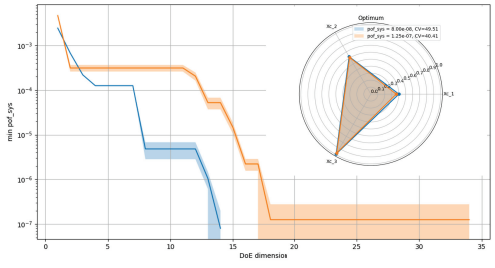

Fig. 9. Results on problem 2.2. Convergence plot for the failure probability pof_{sys} value, alongside the associated $\mathbf{X}_c^*$ value on the radar plot.

4. Conclusions and perspectives

This paper is an attempt to answer to the NASA-DNV UQ challenge 2025 relying on adaptive Gaussian process-based strategies. Given some exact model observations, a dedicated calibration technique based on an optimization approach has been implemented to identify the aleatory variable joint distribution and the epistemic variable uncertainties while managing the stochastic nature of the exact model (through the seed). Then, adaptive Gaussian process-based methods have been derived to identify prediction interval for the model output components via AK-MCS and to solve design optimization under uncertainty through Bayesian optimization involving stochastic simulation model. The proposed approaches could be enriched first by taking into account the Monte Carlo estimation error of $Y_M(\mathbf{X}_a)$ and $Y_m(\mathbf{X}_a)$ while building the Gaussian process for the estimation of the bounds of the predictions intervals, and second by considering a set for the epistemic variables introducing an additional layer of analyses. At last, it could be valuable to develop a more generic approach for identifying uncertainty models, particularly for problems where the stochastic nature of the simulation has a significant impact.

Acknowledgement

The work of the authors has been funded by ONERA - The French Aerospace Lab in the context of the HORUS project.

References

Baudin, M., A. Dutfoy, B. Iooss, and A.-L. Popelin (2015). Open turns: An industrial software for uncertainty quantification in simulation. *arXiv preprint arXiv:1501.05242*.

Dubreuil, S., N. Bartoli, C. Gogu, and T. Lefebvre (2020). Towards an efficient global multidisciplinary design optimization algorithm. *Structural and Multidisciplinary Optimization 62*(4), 1739–1765.

Echard, B., N. Gayton, and M. Lemaire (2011). AK-MCS: an active learning reliability method combining Kriging and Monte Carlo simulation. *Structural safety 33*(2), 145–154.

Hansen, N., S. D. Müller, and P. Koumoutsakos (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary computation 11*(1), 1–18.

Lee, G., W. Kim, H. Oh, B. D. Youn, and N. H. Kim (2019). Review of statistical model calibration and validation—from the perspective of uncertainty structures. *Structural and Multidisciplinary Optimization 60*, 1619–1644.

Marin, J. (2007). Bayesian core: a practical approach to computational bayesian statistics.

Saves, P., R. Lafage, N. Bartoli, Y. Diouane, J. Bussemaker, T. Lefebvre, J. T. Hwang, J. Morlier, and J. R. R. A. Martins (2024). SMT 2.0: A surrogate modeling toolbox with a focus on hierarchical and mixed variables Gaussian processes. *Advances in Engineering Sofware 188*, 103571.

Schöbi, R., B. Sudret, and S. Marelli (2017). Rare event estimation using polynomial-chaos kriging. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering 3*(2), D4016002.

Schwarz, G. (1978). Estimating the dimension of a model. *The annals of statistics*, 461–464.

Williams, C. K. and C. E. Rasmussen (2006). *Gaussian processes for machine learning*, Volume 2. MIT press Cambridge, MA.