ETH zürich

New insights on reliability analysis for stochastic simulators using surrogate models

A. V. Pires, M. Moustapha, S. Marelli and B. Sudret

Chair of Risk, Safety and Uncertainty Quantification
Stefano-Franscini-Platz 5
CH-8093 Zürich

Risk, Safety & Uncertainty Quantification

Data Sheet

Journal:	Proc. 31st International Conference on Noise and Vibration Engineering (ISMA2024) — 10th International Conference on Uncertainty in Structural Dynamics (USD2024), Leuven (Belgium), September 9 - 11, 2024
Report Ref.:	RSUQ-2024-007
Arxiv Ref.:	-
DOI:	-
Date submitted:	June 28, 2024
Date accepted:	September 11, 2024

New insights on reliability analysis for stochastic simulators using surrogate models

A. V. Pires, M. Moustapha, S. Marelli, B. Sudret
ETH Zurich, Chair of Risk, Safety and Uncertainty Quantification,
Stefano-Franscini-Platz 5, 8093 Zurich, Switzerland

Abstract

Reliability analysis is a field of uncertainty quantification that focuses on calculating the failure probability of a system. In this context, the limit-state function is crucial, as it predicts system failure. Traditionally, this function is assumed to yield the same output when repeatedly evaluated on a given set of input parameters. Recently, non-deterministic limit-state functions have attracted increasing interest in the field. These yield random responses for each set of input parameters. This behavior introduces additional complexity to reliability analysis, as both safe and failed states may be observed for the same input parameters. Our research explores how reliability analysis differs when applied to stochastic simulators. More specifically, we aim to expand the definition of failure probability to cope with these models. In this contribution, we propose simulation methods for dealing with these cases. These, however, lead to a great computational burden. Therefore, we also propose surrogate-based methods to efficiently compute the failure probability.

1 Introduction

Computer simulations are ubiquitous in modern engineering, as they allow virtual replication of real-world systems without costly physical experiments. These virtual models take as input a set of variables that control the characteristics and conditions of the virtual system, ultimately allowing the virtual model to mimic the response of their real-world counterparts.

These virtual representations are essential in the field of reliability analysis. In this context, it is of interest to compute the probability that the uncertainties related to a system will lead to its failure. To represent this uncertainty, we collect the set of input parameters in a random vector $\mathbf{X}$ with joint probability density function (PDF) $f_{\mathbf{X}}$, defined over the domain $\mathcal{D}_{\mathbf{X}} \subset \mathbb{R}^M$.

Additionally, the status of the system, *i.e.*, whether the system fails or not, is implicitly represented through the sign of the limit-state function $g : \mathbf{x} \in \mathcal{D}_{\mathbf{X}} \to \mathbb{R}$. Conventionally, failure is associated to $g(\mathbf{x}) \leqslant 0$. The so-called failure domain can then be defined as $\mathcal{D}_f = \{\mathbf{x} : g(\mathbf{x}) \leqslant 0\}$. Within this mathematical framework, it is possible to define the probability of failure as [1, 2, 3]:

$$P_f = \mathbb{P}(g(\mathbf{X}) \leqslant 0) = \int_{\mathcal{D}_f} f_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x}. \tag{1}$$

Computing the probability of failure directly from Eq. (1) relies on the implicit assumption that the limit-state function exhibits deterministic behavior. In other words, it assumes that a fixed input consistently leads to the same output, which is either failure or success. However, this assumption is not always valid.

In the literature, models that return a random variable as the response to a given set of input parameters are often called non-deterministic. Because every time the model is computed, a different response is observed, both failed and safe outcomes can be observed for the same set of input parameters, making Eq. (1) insufficient.

Non-deterministic behavior can arise for multiple reasons. In some cases, it results from the intrinsic randomness (latent variability) within the model. A classic example of this is a structural model subjected to

earthquake excitations [4]. Such excitations are commonly specified by intensity measures, such as peak ground acceleration or magnitude. However, these measures are only loosely related to the actual earthquake excitations, and infinitely many different earthquakes can share the same intensity measures. Consequently, there are infinitely many different responses for each value of an intensity measure. In these cases, stochasticity is inherent to the model and cannot be disregarded or eliminated. We categorize these models as *stochastic simulators*.

Another common cause for non-deterministic behavior in engineering systems is the presence of noise in the data. In this scenario, an underlying deterministic model exists but is inaccessible because only noisy realizations of its response are available [5, 6, 7, 8, 9]. A well-known example of such behavior is crashworthiness simulations [10]. In this case, the mesh of the finite element model significantly impacts the response, and even minor changes can lead to different outcomes. While these fluctuations are generally small, reliability analysis is very sensitive to the behavior of the limit-state function when it assumes near-zero values. Consequently, these fluctuations can cause a change in the sign of the response of the limit-state function, possibly leading to the misclassification of safe regions of the input domain as failed and vice versa [11, 12, 13]. Due to their different nature from stochastic simulators, we refer to them as *noise-corrupted models*, or briefly *noisy models*.

There are significant differences when performing reliability analysis for stochastic simulators and noisy models. Back to the earthquake example, it is of interest to compute how the stochasticity of the load will affect the probability of observing a failure in a structural frame. On the other hand, in the crashworthiness example, it is of interest to filter out the noise component and to retrieve the underlying deterministic reliability problem. Because these models are different in nature, they require different approaches when performing reliability analysis.

To tackle these issues, this contribution is organized as follows: Section 2 formalizes the two types of non-deterministic models and proposes methods for performing reliability analysis depending on the problem at hand. Section 3 presents two surrogate-based approaches for expediting the reliability estimation in Section 2, due to the expensive nature of simulation-based methods. One of the two surrogate approaches can address stochastic models, whereas the other can filter the noise component out of the problem. Section 4 shows the application of the two proposed approaches to a well-known reliability example cast in two different ways: as a stochastic simulator and as a noisy model. Finally, Section 5 summarizes our findings and provides concluding remarks.

2 Problem statement

2.1 Reliability analysis of stochastic simulators

Stochastic simulators are described in the literature as the following mapping [14]:

$$\begin{aligned} g : \mathcal{D}_{\boldsymbol{X}} \times \Omega &\rightarrow \mathbb{R}, \\ (\boldsymbol{x}, \omega) &\mapsto g(\boldsymbol{x}, \omega), \end{aligned} \tag{2}$$

where $\boldsymbol{x}$ is the input vector that belongs to the input space $\mathcal{D}_{\boldsymbol{X}}$. Ω denotes the sample space of the probability space $\{\Omega, \mathcal{F}, \mathbb{P}\}$ which represents the internal stochasticity of the model. In turn, ω represents a realization of a random event in Ω.

In practice, because computers are deterministic and cannot generate true randomness, the inherent randomness of the systems is represented using the so-called latent variables $\mathbf{Z}$. Then, every evaluation of $g(\boldsymbol{x})$ is associated to a different realization of $\boldsymbol{Z}$. In this sense, calling $g(\boldsymbol{x})$ multiple times yields different real values $g(\boldsymbol{x}, \boldsymbol{z}_1)$, $g(\boldsymbol{x}, \boldsymbol{z}_2)$, which mimics the stochastic behavior of the model.

Reliability analysis for stochastic simulators is not yet a mature field, and it currently lacks a standardized approach in the literature. Recently, [15, 16] have used quantile functions derived from these simulators to estimate reliability. These are deterministic functions obtained from the simulators that depend on a user-defined quantile. The advantage of this approach is that the stochastic model is simplified into a deterministic

one, so that the stochasticity does not need to be taken into account explicitly. Conversely, [17] suggested directly quantifying the stochasticity effects on the problem, even if it is not controllable. This is also the approach we follow, which can be cast as:

$$P_f = \mathbb{P}\left(g\left(\boldsymbol{X}, \boldsymbol{Z}\right) \leqslant 0\right). \tag{3}$$

Similarly to the deterministic case, it is possible to estimate the P_f, defined in Eq. (3), using Monte Carlo simulation (MCS) [1, 18].

MCS provides an unbiased P_f estimate, and its extensively documented properties directly apply to the stochastic problem. Consequently, due to its slow convergence rate, estimating rare events requires, in general, an unfeasibly large amount of calls to the possibly expensive-to-evaluate computational model.

When it comes to deterministic models, many approaches exist to bypass the shortcomings of MCS. In general, they can be split into three categories: (i) approximation methods, where the limit-state function is locally approximated using Taylor expansion (*e.g.*, the first-order reliability method) [19]; (ii) variance-reduction techniques, which are sampling approaches that aim at reducing the number of calls to the limit-state function [20, 21] and (iii) surrogate-based approaches, that use inexpensive mathematical models as proxies of the limit-state functions, providing a more efficient alternative [22].

Since reliability analysis for stochastic simulators is still in its infancy, only a handful of estimation methods are available beyond classical MCS. In fact, to the best of the authors' knowledge, the only exception is stochastic importance sampling developed by [17].

Due to the growing interest in stochastic models within the field of uncertainty quantification over the past few years, stochastic emulators, *i.e.*, surrogate models for stochastic simulators, have been introduced in the literature [23, 24, 25]. These emulators generally focus on predicting the conditional distribution of the model response given a particular input realization $\boldsymbol{x}$. Other approaches rather estimate summarizing statistics such as mean and variance or quantiles. The use of emulators in the context of reliability analysis is yet to be thoroughly explored. In this project, we build on existing stochastic emulators to propose novel approaches for reliability analysis for stochastic simulators. Specifically, we consider here the Generalized Lambda Model (GLaM) first introduced by [14, 25].

2.2 Reliability analysis of noise-corrupted models

The notation introduced in Eq. (2) can be further simplified for noisy models. In this context, we assume that an underlying deterministic limit-state function g exists, but also that its values are not directly accessible. Instead, only the values of the noise-corrupted limit-state function $\tilde{g}$ can be observed, which include the noise component ε, as follows:

$$\tilde{g}(\boldsymbol{x}, \omega) = g(\boldsymbol{x}) + \varepsilon(\boldsymbol{x}, \omega). \tag{4}$$

Since $\varepsilon(\boldsymbol{x}, \omega)$ represents the noise component, it is unbiased by definition. Additionally, although not necessary, it is commonly assumed to be homoskedastic (*i.e.*, it does not depend on $\boldsymbol{x}$) and normally distributed. Formally, it can be simplified as $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$.

In this context, we are interested in the probability of failure of the underlying noise-free limit-state function since the noise is extrinsic to the limit-state function. Therefore, the sought failure probability is the one defined in Eq. (1). We refer to it as the *noise-free* probability of failure. In contrast, the *noisy* probability of failure is defined from the noisy limit-state function $\tilde{g}$ as follows:

$$\tilde{P}_f = \mathbb{P}(\tilde{g}(\mathbf{X}, \omega) \leqslant 0) = \mathbb{P}(g(\mathbf{X}) + \varepsilon(\mathbf{X}, \omega) \leqslant 0). \tag{5}$$

Despite the unbiasedness of the noise, the probability of failure in Eq. (5) is different from the one in Eq. (1), more precisely $\tilde{P}_f \geqslant P_f$.

Obtaining the noise-free probability of failure requires denoising the noise-corrupted limit-state function. Due to the unbiasedness property of the noise, it can be shown that:

$$g(\boldsymbol{x}) = \mathbb{E}_\omega[\tilde{g}(\boldsymbol{x}, \omega)]. \tag{6}$$

The noise-free limit-state response can be estimated via replication, where the limit-state function is evaluated repeatedly for identical input parameters, and the mean of these replicated responses is computed. Then, P_f can be calculated via MCS, utilizing the smoothed estimates from the replications. However, due to the need for many replications over multiple realizations of $\boldsymbol{X}$ to estimate reliability, this approach leads to a prohibitive computational burden.

An affordable solution to this problem is to directly denoise the noisy limit-state function. Here, we propose using regression-based surrogate models, specifically Gaussian process regression (GPR) [26], as a denoising method. The benefit of this approach is twofold: the surrogate denoises the problem, and simultaneously enables performing reliability analysis more efficiently. Using the surrogate model $\hat{g}$, the approximated probability of failure becomes:

$$\hat{P}_f = \int_{\{\boldsymbol{x}:\hat{g}(\boldsymbol{x})\leqslant 0\}} f_{\boldsymbol{X}}(\boldsymbol{x})\, d\boldsymbol{x}. \tag{7}$$

Assuming that $\hat{g}$ is a good approximation of g close to the limit-state surface, $\hat{P}_f$ converges to P_f.

3 Surrogate models

3.1 Generalized Lambda Model

The Generalized Lambda Model (GLaM) is a surrogate model designed for stochastic simulators [14, 25]. It aims to approximate the PDF of the output for a given x_0, denoted as $Y \mid X = x_0$. To achieve this, it assumes that the PDF $f_{Y|\boldsymbol{X}}$ can be well-approximated by a generalized lambda distribution (GLD) [27]. The GLD is a parametric distribution known for its flexibility, capable of producing a wide range of unimodal distribution shapes, such as uniform, Weibull, and Gaussian distributions.

The GLD is defined through its quantile function $Q(u;\lambda)$, where u is in the interval $[0,1]$, and $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ represents the parameters of the distribution:

$$Q(u;\lambda) = \lambda_1 + \frac{1}{\lambda_2}\left(\frac{u^{\lambda_3}-1}{\lambda_3} - \frac{(1-u)^{\lambda_4}-1}{\lambda_4}\right). \tag{8}$$

Here, λ_1 is the location parameter, λ_2 is the scale parameter, and λ_3 and λ_4 are shape parameters. Since obtaining a closed-form solution for the PDF of the GLD is not feasible, the PDF is numerically derived from its quantile function when necessary.

In the generalized lambda model, the $\boldsymbol{\lambda}$ parameters are assumed to be deterministic functions of the input parameters $\boldsymbol{x}$. Thus, the conditional distribution at a given location can be expressed as:

$$Y(\boldsymbol{x}) \sim \mathrm{GLD}\left(\lambda_1(\boldsymbol{x}), \lambda_2(\boldsymbol{x}), \lambda_3(\boldsymbol{x}), \lambda_4(\boldsymbol{x})\right). \tag{9}$$

Given a data set $\left\{\left(\boldsymbol{x}^{(i)}\right), \boldsymbol{\lambda}\left(\boldsymbol{x}^{(i)}\right), i = 1, \ldots, N\right\}$, the lambda functions can be approximated using standard surrogate models. Under the mild assumption that $\lambda_1(\boldsymbol{X}), ..., \lambda_4(\boldsymbol{X})$ have finite variances, they can be efficiently approximated by adaptive sparse polynomial chaos expansions (PCE) [28], as follows:

$$\begin{aligned}
\lambda_l(\boldsymbol{x}) &\approx \lambda_l^{\mathrm{PC}}(\boldsymbol{x};\boldsymbol{c}) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}_l} c_{l,\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}), \quad l = 1, 3, 4, \\
\lambda_2(\boldsymbol{x}) &\approx \lambda_2^{\mathrm{PC}}(\boldsymbol{x};\boldsymbol{c}) = \exp\left(\sum_{\boldsymbol{\alpha}\in\mathcal{A}_2} c_{2,\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})\right),
\end{aligned} \tag{10}$$

where $c_{\boldsymbol{\alpha}}$ are real coefficients, and $\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})$ are multivariate polynomials orthonormal with respect to $f_{\boldsymbol{X}}$. $\mathcal{A}$ corresponds to the truncation set of the $\boldsymbol{\alpha}$ multi-index, identifying the degree of the multivariate polynomial along each input variable. Additionally, since $\lambda_2(\boldsymbol{x})$ must be positive, its associated expansion is constructed in a transformed space to ensure this constraint is satisfied.

There are two approaches to constructing the GLaM emulator. The first relies on replications where each $\boldsymbol{\lambda}\left(\boldsymbol{x}^{(i)}\right)$ is estimated by maximum likelihood considering R replications of the limit-state evaluations at $\boldsymbol{x}^{(i)}$ [14]. This step is then followed by the PCE approximation of the lambda functions. A second generally more efficient approach consists in building a global model for the joint distribution of the inputs and outputs [25]:

$$f_{\boldsymbol{X},Y}(\boldsymbol{x}, y) = f_{Y|\boldsymbol{X}}\left(y|\boldsymbol{x}\right) \cdot f_{\boldsymbol{X}}(\boldsymbol{x}), \tag{11}$$

and approximating the conditional distribution by a generalized lambda distribution:

$$f^{\mathrm{GLD}}_{\boldsymbol{X},Y}(\boldsymbol{x}, y;\, \boldsymbol{c}) = f^{\mathrm{GLD}}_{Y|\boldsymbol{X}}\left(y;\, \boldsymbol{\lambda}^{\mathrm{PC}}(\boldsymbol{x};\, \boldsymbol{c})\right) \cdot f_{\boldsymbol{X}}(\boldsymbol{x}). \tag{12}$$

Given a training set $\mathcal{X} = \left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\right\}$ and $\mathcal{Y} = \left\{y^{(1)}, \ldots, y^{(N)}\right\}$, this approach entails estimating the coefficients using maximum likelihood. The equivalent maximum log-likelihood estimation problem reads:

$$\hat{c} = \arg\max_{\boldsymbol{c}\in\mathcal{C}} \log\left(f^{\mathrm{GLD}}\left(y^{(i)}; \lambda^{\mathrm{PC}}\left(\boldsymbol{x}^{(i)}; \boldsymbol{c}\right)\right)\right). \tag{13}$$

3.2 Gaussian process regression basics

In the case of noise-corrupted limit-states, the experimental design is defined as:

$$\mathcal{E} = \left\{\left(\boldsymbol{x}^{(i)}, \tilde{y}^{(i)}\right) : \tilde{y}^{(i)} = \tilde{g}\left(\boldsymbol{x}^{(i)}, \omega^{(i)}\right) \in \mathbb{R}, \boldsymbol{x}^{(i)} \in D_{\boldsymbol{X}}, i = 1, \ldots, n\right\}. \tag{14}$$

When deploying Gaussian process regression (GPR), we assume that the expensive simulator is a realization of an unknown Gaussian process (GP), generally represented as:

$$\mathcal{M}^{\mathrm{GP}}(x) = \mu(x) + \sigma^2 Z(x;\omega), \tag{15}$$

where $\mu(x)$ is the mean function, σ^2 is the process variance, $Z(x;\omega)$ is a zero-mean, unit-variance stationary process characterized by its auto-correlation $R(x, x';\theta)$ with hyperparameters θ.

Aiming to derive an explicit analytical solution for the GPR predictions, we assume that the data is corrupted with additive homoskedastic Gaussian noise. In this case, the covariance matrix of the noise component can be written as $\Sigma_n^2 = \sigma_n^2 I$, where σ_n^2 is the noise variance and $\boldsymbol{I}$ is the identity matrix of size $N \times N$. The covariance matrix Σ of the GP described in Eq. (15) reads:

$$\Sigma = \sigma^2 \boldsymbol{R} + \sigma_n^2 \boldsymbol{I}. \tag{16}$$

where R is the correlation matrix computed using a given autocorrelation function. Additionally, by introducing the following change of variables $\sigma^2_{\text{total}} = \sigma^2 + \sigma_n^2$ and $\tau = \frac{\sigma_n^2}{\sigma^2_{\text{total}}}$, Σ can be cast as:

$$\Sigma = \sigma^2_{\text{total}}\left(\boldsymbol{R} + \tau \boldsymbol{I}\right). \tag{17}$$

In this work, we consider the Matérn $5/2$ autocorrelation function. The multi-dimensional version is constructed using the ellipsoidal formulation [26]. Additionally, we consider the mean function of the GPR $\mu\left(\boldsymbol{x}\right)$ known and equal to 0, leading to the so-called *simple Kriging* approach. Consequently, the joint Gaussian distribution between the observations and predictions can be cast as:

$$\begin{Bmatrix} \widehat{g}(\boldsymbol{x}) \\ \widetilde{\mathcal{Y}} \end{Bmatrix} \sim \mathcal{N}_{N+1}\left(\boldsymbol{0}, \quad \begin{Bmatrix} \sigma^2 & \boldsymbol{r}^T(\boldsymbol{x}) \\ \boldsymbol{r}(\boldsymbol{x}) & \sigma^2\boldsymbol{R} + \sigma_n^2\boldsymbol{I} \end{Bmatrix}\right), \tag{18}$$

The denoised prediction $\hat{g}(x)$ is obtained by conditioning the GP on the observations, yielding a normal distribution with mean $\mu_{\hat{g}}(x)$ and variance $\sigma^2_{\hat{g}}(x)$:

$$\mu_{\hat{g}}(x) = r(x)^T \tilde{R}^{-1} \tilde{Y}, \tag{19}$$

$$\sigma^2_{\hat{g}}(x) = \sigma^2_{\text{total}} \left(1 - r(x)^T \tilde{R}^{-1} r(x)\right). \tag{20}$$

An analytical estimate of the total variance σ^2_{total}, which is a function of the auto-correlation hyperparameters θ, is derived by maximum likelihood:

$$\sigma^2_{\text{total}} = \sigma^2_{\text{total}}(\theta) = \frac{1}{N} \tilde{y}^T \tilde{R}^{-1} \tilde{y}. \tag{21}$$

The hyperparameters θ and τ are estimated using maximum likelihood estimation (MLE), by solving:

$$\left(\hat{\theta}, \hat{\tau}\right) = \arg\min_{\theta,\tau} \frac{1}{2} \left[\log\|\tilde{R}\| + n \log\left(\sigma^2_{\text{total}}\right) + n\right]. \tag{22}$$

For optimization, we employ the Hybrid Covariance Matrix Adaptation-Evolution Strategy (HCMA-ES) [29], implemented in UQLab [30].

4 Results

4.1 Stochastic simulators

To demonstrate how to perform reliability analysis using stochastic emulators and show their convergence to the associated probability of failure, we convert the well-known $R - S$ reliability analysis problem [1] into a stochastic simulator. This is achieved by introducing latent variables into the limit-state function, defined as:

$$g(\boldsymbol{X}, \boldsymbol{Z}) = Z_1 \cdot R - \frac{S}{Z_2} \tag{23}$$

where $\boldsymbol{X} = \{R, S\}$ represents the input random vector containing two random variables: R, the capacity of the system, and S, the demand on the system. $\boldsymbol{Z} = \{Z_1, Z_2\}$ are the latent variables introduced to emulate the desired stochastic behavior. The moments, distributions, and parameters associated with these variables are shown in Table 1.

Table 1: Moments, distributions, and parameters of the considered variables.

Variable	Distribution	Mean	Std. Deviation	λ	ζ
R	Lognormal	5.0	0.8	1.5968	0.1590
S	Lognormal	2.0	0.6	0.6501	0.2936
Z_1	Lognormal	1.0	0.028	-0.0004	0.0280
Z_2	Lognormal	1.0	0.096	-0.0046	0.0958

By applying a logarithmic transformation to the two components of (23), we can analytically compute the associated probability of failure, which is expressed as:

$$P_f = \Phi\left(\frac{\lambda_{Z_1} + \lambda_R + \lambda_{Z_2} - \lambda_S}{\sqrt{\zeta^2_{Z_1} + \zeta^2_R + \zeta^2_{Z_2} + \zeta^2_S}}\right) = 3.438 \times 10^{-3}. \tag{24}$$

To solve this problem we approximate the stochastic simulator with a generalized lambda model using various experimental designs of increasing sizes: $N = \{100; 500; 1{,}000; 5{,}000; 10{,}000\}$. The probability of failure is estimated using MCS on the resulting emulators with 1×10^6 samples.

Figure 1 shows box plots of the corresponding P_f estimates considering 50 repetitions of the analysis. The reference probability of failure is depicted by the horizontal dashed line.

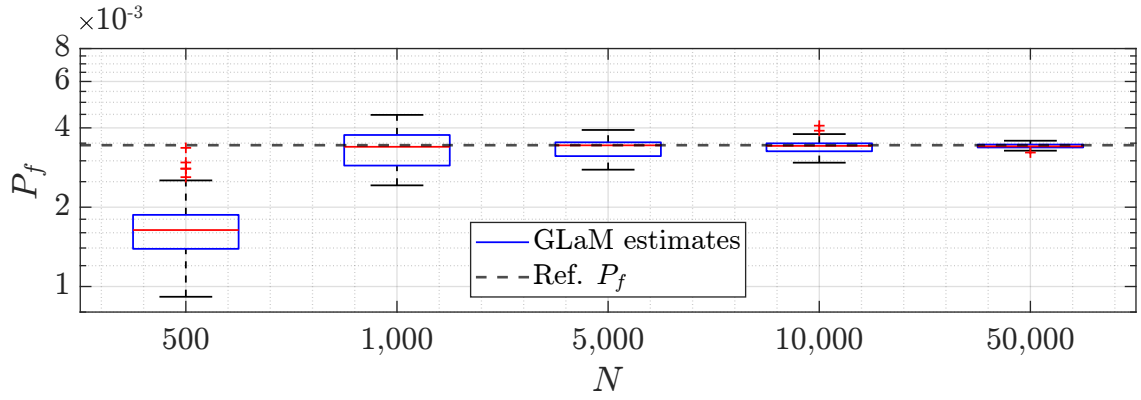

Figure 1: Stochastic $R - S$ problem – Estimation of the P_f of a stochastic model using GLaM as surrogate model. The emulator is trained on N points obtained from Eq. (23).

From Figure 1, the convergence behavior of the emulator is evident: as the size of the experimental design increases, the estimated probability of failure approaches the reference value, and the width of the boxplots decreases. Notably, even with as few as $5{,}000$ points in the experimental design, an unbiased and accurate estimation of the probability of failure can be achieved. In comparison, a direct MCS with $N_{MCS} = 5{,}000$ samples would result in an estimation with a coefficient of variation larger than 25%, well over one order of magnitude larger.

4.2 Noise-corrupted limit-state function

To illustrate the effectiveness of the proposed method, we again consider the $R - S$ problem, but now we cast it as a noisy model. The noise-free and noisy limit-state functions are defined as:

$$g(\boldsymbol{X}) = R - S \quad \text{and} \quad \tilde{g}(\boldsymbol{X}) = R - S + \varepsilon. \tag{25}$$

As in the previous example, the random input vector comprises the two variables: $\boldsymbol{X} = \{R, S\}$, corresponding to the resistance and demand, respectively. To derive a closed-form solution to this problem, we now consider both variables to be normally distributed, *i.e.* $R \sim \mathcal{N}(\mu_R, \sigma_R^2)$, and $S \sim \mathcal{N}(\mu_S, \sigma_S^2)$. The noise term ε is assumed to be independent of R and S, homoskedastic, and Gaussian. The analytical probabilities of failure for these limit-state functions are given by:

$$P_f = \Phi\left(-\frac{\mu_R - \mu_S}{\sqrt{\sigma_R^2 + \sigma_S^2}}\right), \tag{26}$$

and

$$\tilde{P}_f = \Phi\left(-\frac{\mu_R - \mu_S}{\sqrt{\sigma_R^2 + \sigma_S^2 + \sigma_\varepsilon^2}}\right), \tag{27}$$

respectively. These equations show that the variance introduced by the noise term σ_ε^2 increases the probability of failure $\tilde{P}_f$ relative to the noise-free probability P_f.

To denoise the model, we train a Gaussian process regression model as described in Sec. 3.2. Its mean predictor $\mu_{\hat{g}}$ serves as a denoised version of the limit-state function $\tilde{g}$.

For numerical solutions, we assume that $R \sim \mathcal{N}(5, 0.8^2)$ and $S \sim \mathcal{N}(2, 0.6^2)$. The noise term is parametrically defined as $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$, enabling us to test the robustness of the method as the noise level σ_ε^2 increases. In this example, $\hat{P}_f$ is evaluated using MCS.

Figure 2 presents the probability of failure $\hat{P}_f$ for different noise levels $\sigma_\varepsilon^2 \in \{0, 0.25, 0.5, 1\}$ and varying sizes N of the experimental design used to fit the Gaussian process $\hat{g}(r, s)$. The reference value, $P_f = \Phi\left(-\frac{5-2}{\sqrt{0.8^2+0.6^2}}\right) = \Phi(-3) = 1.35 \times 10^{-3}$, is indicated by the horizontal dashed line. All results are depicted as box plots, with the entire procedure replicated 50 times to account for statistical uncertainty.

The noise-free probability of failure is recovered for each noise level when N is sufficiently large. The size N required for accurate P_f estimates increases with σ_ε; 100 points suffice when $\sigma_\varepsilon = 0$ (noise-free limit-state function), while 1,000 to 10,000 points are necessary for higher noise levels.

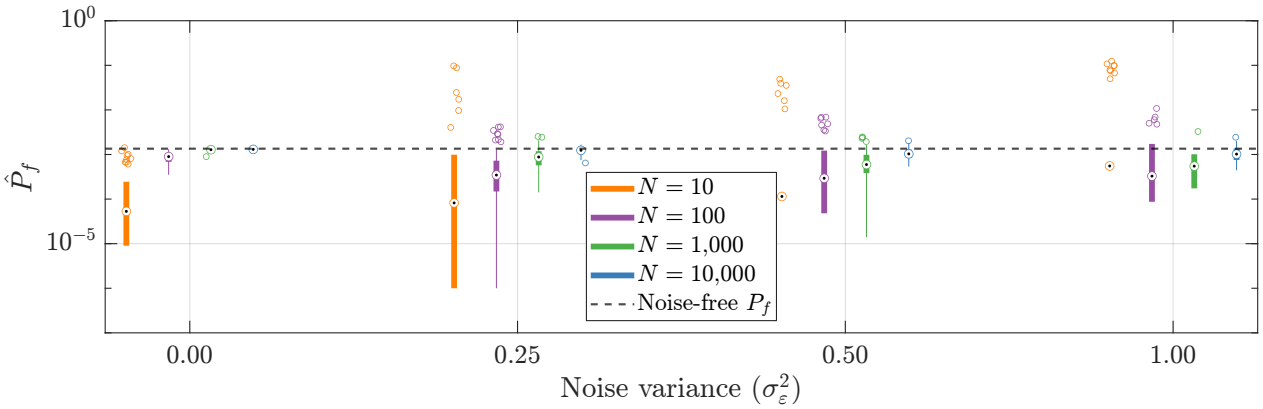

Figure 2: Noisy $R - S$ problem – Estimation of the noise-free P_f using a Gaussian process model based on N points (obtained from the noisy limit-state function $\tilde{g}$) and for different values of the noise variance. Results obtained from [13].

5 Conclusions

In this contribution, we discussed performing reliability analysis on non-deterministic models. Two distinct types of problems arise, depending on the source of the non-deterministic behavior. Therefore, we present two perspectives on the problem. First, when stochasticity is intrinsic to the problem, and second, when it is introduced by an external source of noise. In the first scenario, it is essential to consider how the intrinsic stochasticity affects the problem, as it is a fundamental aspect of the system. On the other hand, when the stochasticity is introduced by an external source, it is not inherent to the problem and should be removed.

We demonstrate that performing reliability analysis using simulation methods is feasible for both perspectives. However, this approach can lead to a high computational burden. To circumvent this issue, we use surrogate models. In the first case, we show that stochastic emulators can accurately mimic the stochasticity of the original model and ultimately converge to the reference probability of failure. In the second case, we show that Gaussian process regression can be used to denoise the problem and estimate the underlying noise-free probability of failure.

Acknowledgements

The project leading to this application has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 955393.

References

[1] R. Melchers and A. Beck, *Structural reliability analysis and prediction*. John Wiley & Sons, 2018.

[2] M. Lemaire, *Structural reliability*. Wiley, 2009.

[3] O. Ditlevsen and H. Madsen, *Structural reliability methods*. J. Wiley and Sons, Chichester, 1996.

[4] X. Zhu, M. Broccardo, and B. Sudret, "Seismic fragility analysis using stochastic polynomial chaos expansions," *Probabilistic Eng. Mech.*, vol. 72, p. 103413, 2023.

[5] A. Forrester, A. Keane, and N. Bressloff, "Design and analysis of "noisy" computer experiments," *AIAA J.*, vol. 44, pp. 2331–2339, 2006.

[6] V. Picheny, T. Wagner, and D. Ginsbourger, "A benchmark of Kriging-based infill criteria for noisy optimization," *Struct. Multidiscipl. Optim.*, vol. 48, no. 3, pp. 607–626, 2013.

[7] C. Chevalier, J. Bect, D. Ginsbourger, E. Vazquez, V. Picheny, and Y. Richet, "Fast parallel Kriging-based stepwise uncertainty reduction with application to the identification of an excursion set," *Technometrics*, vol. 56, no. 4, pp. 455–465, 2014.

[8] H. Jalali, I. V. Nieuwenhuyse, and V. Picheny, "Comparison of Kriging-based algorithms for simulation optimization with heterogeneous noise," *Eur. J. Oper. Res.*, vol. 261, no. 1, pp. 279–301, 2017.

[9] F. Chinesta, E. Cueto, E. Abisset-Chavanne, J. L. Duval, and F. E. Khaldi, "Virtual, digital and hybrid twins: A new paradigm in data-based engineering and engineered data," *Arch. Comput. Methods Eng.*, vol. 27, no. 1, pp. 105–134, 2018.

[10] S. S. Ahmadisoleymani and S. Missoum, "Stochastic crashworthiness optimization accounting for simulation noise," *J. Mech. Des.*, vol. 144, no. 5, pp. 1–15, 2021.

[11] A. P. van den Eijnden, T. Schweckendiek, and M. A. Hicks, "Metamodelling for geotechnical reliability analysis with noisy and incomplete models," *Georisk*, vol. 16, no. 3, pp. 518–535, 2021.

[12] C. Chevalier, "Fast uncertainty reduction strategies relying on gaussian process models," Ph.D. dissertation, University of Bern, 2013.

[13] A. V. Pires, M. Moustapha, S. Marelli, and B. Sudret, "Reliability analysis for data-driven noisy models using active learning," 2024, submitted.

[14] X. Zhu and B. Sudret, "Replication-based emulation of the response distribution of stochastic simulators using generalized lambda distributions," *Int. J. Uncertain. Quantif.*, vol. 10, no. 3, pp. 249–275, 2020.

[15] X. Zheng, J. Zhang, G. Tang, T. Jiang, W. Peng, and W. Yao, "A reliability analysis method based on quantile regression and feedforward neural network," in *12th International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (QR2MSE 2022), Emeishan, China*, 2022, pp. 294–300.

[16] J. Navarro and F. Buono, "Predicting future failure times by using quantile regression," *Metrika*, vol. 86, no. 5, pp. 543–576, 2022.

[17] Y. Choe, E. Byon, and N. Chen, "Importance sampling for reliability evaluation with stochastic simulation models," *Technometrics*, vol. 57, no. 3, pp. 351–361, 2015.

[18] R. Y. Rubinstein and D. P. Kroese, *Simulation and the Monte Carlo method*. John Wiley & Sons, Inc., 2016.

[19] R. Rackwitz and B. Fiessler, "Structural reliability under combined load sequences," *Comput. Struct.*, vol. 9, pp. 489–494, 1978.

[20] R. E. Melchers, "Importance sampling in structural systems," *Struct. Saf.*, vol. 6, pp. 3–10, 1989.

[21] S. K. Au and J. L. Beck, "Estimation of small failure probabilities in high dimensions by subset simulation," *Probabilistic Eng. Mech.*, vol. 16, no. 4, pp. 263–277, 2001.

[22] R. Teixeira, M. Nogal, and A. O'Connor, "Adaptive approaches in metamodel-based reliability analysis: A review," *Struct. Saf.*, vol. 89, p. 102019, 2021.

[23] B. Ankenman, B. Nelson, and J. Staum, "Stochastic Kriging for simulation metamodeling," *Oper. Res.*, vol. 58, no. 2, pp. 371–382, 2010.

[24] L. Torossian, V. Picheny, R. Faivre, and A. Garivier, "A review on quantile regression for stochastic computer experiments," *Reliab. Eng. Syst. Saf.*, vol. 201, p. 106858, 2020.

[25] X. Zhu and B. Sudret, "Emulation of stochastic simulators using generalized lambda models," *SIAM-ASA J. Uncertain.*, vol. 9, no. 4, pp. 1345–1380, 2021.

[26] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*, Internet ed., ser. Adaptive computation and machine learning. Cambridge, Massachusetts: MIT Press, 2006.

[27] M. Freimer, G. Kollia, G. S. Mudholkar, and C. T. Lin, "A study of the generalized Tukey lambda family," *Comm. Stat. Theor. Meth.*, vol. 17, pp. 3547–3567, 1988.

[28] N. Lüthen, S. Marelli, and B. Sudret, "Sparse polynomial chaos expansions: Literature survey and benchmark," *SIAM/ASA J. Uncertain. Quantif.*, vol. 9, no. 2, pp. 593–649, 2021.

[29] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[30] S. Marelli and B. Sudret, "UQLab: A framework for uncertainty quantification in Matlab," in *Vulnerability, Uncertainty, and Risk (Proc. 2nd Int. Conf. on Vulnerability, Risk Analysis and Management (ICVRAM2014), Liverpool, United Kingdom)*, 2014, pp. 2554–2563.