RELIABILITY ANALYSIS FOR DATA-DRIVEN NOISY MODELS USING ACTIVE LEARNING

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

CHAIR OF RISK, SAFETY AND UNCERTAINTY QUANTIFICATION
STEFANO-FRANSCINI-PLATZ 5
CH-8093 ZÜRICH

Data Sheet

Reliability analysis for data-driven noisy models using active learning

Anderson V. Pires [*1], M. Moustapha [†1], Stefano Marelli[‡1], and Bruno Sudret[§1]

[1]*Chair of Risk, Safety and Uncertainty Quantification, ETH Zürich, Switzerland*

January 19, 2024

Abstract

Reliability analysis aims at estimating the failure probability of an engineering system. It often requires multiple runs of a limit-state function, which usually relies on computationally intensive simulations. Traditionally, these simulations have been considered deterministic, *i.e.,* running them multiple times for a given set of input parameters always produces the same output. However, this assumption does not always hold, as many studies in the literature report non-deterministic computational simulations (also known as noisy models). In such cases, running the simulations multiple times with the same input will result in different outputs. Similarly, data-driven models that rely on real-world data may also be affected by noise. This characteristic poses a challenge when performing reliability analysis, as many classical methods, such as FORM and SORM, are tailored to deterministic models. To bridge this gap, this paper provides a novel methodology to perform reliability analysis on models contaminated by noise. In such cases, noise introduces latent uncertainty into the reliability estimator, leading to an incorrect estimation of the real underlying reliability index, even when using Monte Carlo simulation. To overcome this challenge, we propose the use of denoising regression-based surrogate models within an active learning reliability analysis framework. Specifically, we combine Gaussian process regression with a noise-aware learning function to efficiently estimate the probability of failure of the underlying noise-free model. We showcase the effectiveness of this methodology on standard benchmark functions and a finite element model of a realistic structural frame.

Keywords: Structural reliability – Noisy limit-state functions – Surrogate models – Gaussian process regression – Active learning

[*]apires@ethz.ch

[†]moustapha@ibk.baug.ethz.ch

[‡]marelli@ibk.baug.ethz.ch

[§]sudret@ethz.ch

1 Introduction

Computer simulations are virtual representations of systems and are ubiquitous in current engineering practice. These virtual prototypes give insights into the behavior of the system under different scenarios, reducing the need for costly experimental setups. Such scenarios are controlled by the so-called *input parameters*, a set of variables defined by the user which describe the characteristics and physical properties of the simulated system and its operating conditions.

Because these virtual models enable the simulation of the system under different conditions, they play a crucial role in reliability analysis (Melchers and Beck, 2018). The latter aims at quantifying the probability that the uncertainties in the input parameters lead to a performance failure. To compute such a probability of failure P_f, we consider a probabilistic framework where the uncertain input parameters are represented by a random vector denoted by $\boldsymbol{X}$. Their associated uncertainty can be entirely characterized by the joint probability density function (PDF) $f_{\boldsymbol{X}}$, defined on the domain $\mathcal{D}_{\boldsymbol{X}} \subset \mathbb{R}^M$.

The state of the system, *i.e.*, failure or safe operation, is defined through the so-called *limit-state function* $g : \boldsymbol{x} \in \mathcal{D}_{\boldsymbol{X}} \to \mathbb{R}$, which maps the input domain $\mathcal{D}_{\boldsymbol{X}}$ to the real line. This function is constructed by comparing the response of the computational model to one or more limit-states of the system under consideration. Conventionally, failure occurs when $g(\boldsymbol{x}) \leqslant 0$. The *failure domain* $\mathcal{D}_f$ can be defined as the subset of the input domain on which the limit-state function returns a non-positive value, *i.e.*, $\mathcal{D}_f = \{\boldsymbol{x} : g(\boldsymbol{x}) \leqslant 0\}$. Additionally, the particular set of points $\boldsymbol{x} \in \mathcal{D}_{\boldsymbol{X}}$ such that $g(\boldsymbol{x}) = 0$ is commonly referred to as the *limit-state surface*. Given this framework, it is possible to define the probability of failure as follows (Ditlevsen and Madsen, 1996; Lemaire, 2009; Melchers and Beck, 2018):

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

The integration domain depends on the response of the limit-state function g and implicitly assumes that only a single scalar value $y = g(\boldsymbol{x}_0)$ can be computed for any given vector $\boldsymbol{x}_0$, *i.e.*, this definition relies on deterministic limit-state functions. Nevertheless, not all computational models are deterministic. The so-called *non-deterministic models* provide stochastic responses for a given set of inputs, a behavior that, in practice, is seen as noisy.

The inherent variability observed in measurements acquired during physical experiments is a classic example of noise. However, this is not the only case where noise can be found. Giunta et al. (1994); Narducci et al. (1995); Papila and Haftka (2000); Forrester et al. (2006) report the existence of numerical noise in computational fluid dynamics (CFD) simulations, whereas Forrester et al. (2006); Duddeck (2007); Zhu et al. (2013); Paz et al. (2020); Ahmadisoleymani and Missoum (2021) report a similar behavior in crashworthiness simulations. Biermann et al. (2008); Abbiati et al. (2022) comment on noise in models combining physical experiments with simulations, known as *hybrid simulators* in civil engineering. Similar issues also arise for the class of so-called grey-box models (Chinesta et al., 2018), which combine data-based models

with computational simulations. Noise can also arise from intrinsic properties of the model, for instance in wind turbine simulations. In those models, wind loads are represented by three-dimensional random fields that depend on macroscopic parameters, such as the mean wind speed at a reference altitude, the turbulence intensity, the wind shear exponent, the air density and the blade pitch angle (Dimitrov et al., 2014; Abdallah et al., 2019). Despite fixing the value of their input parameters, infinitely many wind fields can be generated, leading to infinitely many wind turbine responses, characterizing the model response as noisy. In fact, the attention towards this class of models has grown in the past few years, with an increasing body of literature on so-called *stochastic simulators* and *emulators* (Ankenman et al., 2010; Moutoussamy et al., 2015; Azzi et al., 2019; Torossian et al., 2020; Zhu and Sudret, 2020, 2021a,b, 2023; Zhu et al., 2023; Lüthen et al., 2023).

This paper aims to address reliability analysis for limit-state functions corrupted by noise. Although only recently considered in this field, noise has been previously accounted for explicitly in Bayesian optimization. In general, the methods proposed in the latter capitalize on *surrogate models*. These are inexpensive-to-evaluate mathematical approximations trained on a limited set of full-scale simulations called *experimental design* (ED) and can be split into two broad classes: *interpolation-* and *regression-* based methods. Interpolation methods are generally used when assuming that the training points are not affected by random noise. Consequently, they precisely interpolate through the points of the ED. In contrast, regression methods aim at minimizing a global loss function, usually the mean squared error between the model predictions and the values collected in the experimental design. Surrogate models are usually employed to replace expensive simulations, and in the reliability analysis field, interpolation-based ones have often been the preferred choice. For noise-corrupted limit-state functions, however, regression-based surrogate models bring another advantage, as they can denoise the computational models (Torre et al., 2019).

Many of the methods proposed in the Bayesian optimization field employ Gaussian process regression (GPR) as a surrogate (Rasmussen and Williams, 2006), with an emphasis on its adaptive version (Jones et al., 1998; Shahriari et al., 2016), which involves sequentially enriching the experimental design in view of improving the current optimum. As most methods rely on GPR, the main difference between them is the so-called *infill criteria*, a function that defines where the experimental design is enriched at each iteration. Picheny et al. (2013); Zhan and Xing (2020) review the available methods and infill criteria for the optimization of noisy models, with a focus on homoskedastic noise, *i.e.*, when the noise variance is identical for all outputs. Jalali et al. (2017) focus their review on heteroskedastic noise. More recently, techniques employing multi-fidelity models have been proposed by Ficini et al. (2021) and Pellegrini et al. (2022).

In the context of reliability analysis, Chevalier (2013) addresses a noisy reliability problem related to nuclear safety. In their case, one of the parameters is computed through Monte Carlo simulation (MCS), which yields a small simulation noise due to the finite sample size used in practice. Similarly to the optimization literature, they capitalize on Gaussian process regression

to denoise the problem and sequentially enrich the experimental design using the *Stepwise Uncertainty reduction* (SUR) method (Vazquez and Bect, 2009; Bect et al., 2012; Chevalier et al., 2014). More recently, van den Eijnden et al. (2021) presented a reliability problem with a high target reliability index, the estimation of which depends on a noise-corrupted limit-state function. Similarly, they employ GPR and extend the classical U learning function (Echard et al., 2011) to cope with noisy data and sequentially enrich the experimental design.

Although a small number of works addressing the reliability analysis of mildly noisy problems can be found in the literature, a formalization of this problem, a detailed study of its effects, and a formal and robust methodology for coping with the challenges it poses are still lacking. In this paper, we aim to fill this gap by showing the particularities of performing reliability analysis on noisy models. Specifically, we show that traditional simulation-based methods for performing reliability analysis tend to converge to a wrong probability of failure, herein referred to as noisy probability of failure $\tilde{P}_f$. To circumvent this issue and converge to the actual noise-free P_f, denoising the problem is needed. To do that, we capitalize on regression-based surrogate models.

The paper is organized as follows: In Section 2, the problem of reliability analysis for noisy limit-state functions is first formulated. Section 3 demonstrates how the use of regression-based denoising leads to the estimation of the correct probability of failure. Section 4 outlines an efficient adaptive approach aimed at enhancing the computational efficiency of the algorithm. Section 5 showcases the results of the proposed methodology on two common academic examples and a complex, realistic structural frame problem. Finally, Section 6 presents our final thoughts and conclusions.

2 Problem statement

In the context of reliability analysis for noisy models, we consider an underlying noise-free limit-state function $g : \boldsymbol{x} \in \mathcal{D}_{\boldsymbol{X}} \subset \mathbb{R}^M$, whose values are, however, not directly accessible. Instead, we can measure or compute the value $\tilde{g}(\boldsymbol{x}, \omega)$, which is corrupted by some zero-mean noise $\varepsilon(\boldsymbol{x}, \omega)$. We pose:

$$\tilde{g}(\boldsymbol{x}, \omega) = g(\boldsymbol{x}) + \varepsilon(\boldsymbol{x}, \omega), \tag{2}$$

where we assume $\mathbb{E}_{\omega}[\varepsilon(\boldsymbol{x}, \omega)] = 0$ for any $\boldsymbol{x} \in \mathcal{D}_{\boldsymbol{X}}$. Note that ε may or may not depend on $\boldsymbol{x}$ in practical applications.

The notation $\tilde{g}(\boldsymbol{x}, \omega)$ explicitly introduces the random outcome ω of a sample space Ω that represents the variability attributed to the noise. In other words, when querying $g(\boldsymbol{x})$ several times, we get *different* real values $\tilde{g}(\boldsymbol{x}, \omega_1)$, $\tilde{g}(\boldsymbol{x}, \omega_2)$ and so forth.

We are interested in the probability of failure of the underlying noise-free limit-state function (Eq. (1)):

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

4

which we further refer to as the *noise-free* probability of failure. In contrast, the so-called *noisy* probability of failure is defined from the noisy limit-state function $\tilde{g}$ as follows

$$\tilde{P}_f = \mathbb{P}\left(\tilde{g}\left(\mathbf{X}, \omega\right) \leqslant 0\right) = \mathbb{P}\left(g\left(\mathbf{X}\right) + \varepsilon\left(\mathbf{X}, \omega\right) \leqslant 0\right). \tag{4}$$

Despite the noise unbiasedness, a different probability of failure is computed when blindly considering the noisy model and, in general, $\tilde{P}_f \geqslant P_f$. This occurs because the noise term introduces more uncertainty to the problem. To better illustrate this point, let us consider a simple linear limit-state function, such as the well-known $R - S$ case. For this problem, the random input vector contains two Gaussian variables $\boldsymbol{X} = \{R, S\}$, where $R \sim \mathcal{N}\left(\mu_R, \sigma_R^2\right)$ corresponds to the resistance of the system, and $S \sim \mathcal{N}\left(\mu_S, \sigma_S^2\right)$ corresponds to the demand. Assuming that the noise term is independent on R and S and $\varepsilon \sim \mathcal{N}\left(0, \sigma_\varepsilon^2\right)$, the noise-free and noise-corrupted limit-state functions read:

$$g\left(\boldsymbol{X}\right) = R - S \text{ and } \tilde{g}\left(\boldsymbol{X}, \omega\right) = R - S + \varepsilon\left(\omega\right). \tag{5}$$

Then, the analytical probabilities of failure are equal to

$$P_f = \Phi\left(-\frac{\mu_R - \mu_S}{\sqrt{\sigma_R^2 + \sigma_S^2}}\right), \tag{6}$$

and

$$\tilde{P}_f = \Phi\left(-\frac{\mu_R - \mu_S}{\sqrt{\sigma_R^2 + \sigma_S^2 + \sigma_\varepsilon^2}}\right), \tag{7}$$

respectively. From these two equations, it is clear that the only difference is the variance introduced by the noise term and that the noisy probability of failure $\tilde{P}_f$ is greater than the searched one P_f.

A crucial consequence of this observation is that simulation methods such as Monte Carlo simulation (Rubinstein and Kroese, 2016), subset simulation (Au and Beck, 2001) or importance sampling (Melchers, 1989) will converge to the noisy probability of failure (Eq. (4)). Similarly, approximation methods, such as the first and second-order reliability methods (Hasofer and Lind, 1974; Rackwitz and Fiessler, 1978), will also fail in estimating the noise-free probability of failure, as these methods depend on the gradient of the limit-state function, which would be stochastic and lead to unstable results. In both cases, obtaining the noise-free probability of failure of the system (Eq. (3)) is not possible.

3 Retrieving the noise-free probability of failure

Estimating the noise-free probability of failure while only having access to a noisy limit-state function is a matter of denoising. Following the approach from the works on optimization outlined earlier (Picheny et al., 2013; Zhan and Xing, 2020), we propose the use of regression-based

surrogate models to simultaneously denoise and cheaply estimate the probability of failure, as follows:

$$\hat{P}_f = \int_{\{\boldsymbol{x}:\hat{g}(\boldsymbol{x})\leqslant 0\}} f_{\boldsymbol{X}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}, \tag{8}$$

where $\hat{P}_f$ and $\hat{g}$ respectively represent the estimated probability of failure and the surrogate of the limit-state function.

The regression component of the algorithm reconstructs a denoised version of the limit-state function, enabling the computation of the noise-free probability of failure. More precisely, because regression-based surrogate models are typically unbiased, they converge to the expected value of the noisy limit-state function (see, *e.g.* Torre et al. (2019); van den Eijnden et al. (2021)).

Examples of regression-based surrogate models include Gaussian process regression (Rasmussen and Williams, 2006), sparse polynomial chaos expansions (PCE, Blatman and Sudret, 2011; Lüthen et al., 2022), and support vector regression (SVR, Smola and Schölkopf, 2004). Due to its versatility and widespread adoption in the context of reliability analysis, in the following we focus on Gaussian process regression.

In principle, it is also possible to denoise the limit-state function by computing $\mathbb{E}_{\omega}\left[\tilde{g}\left(\boldsymbol{X},\omega\right)\right]$ via direct Monte Carlo simulation, by simply drawing multiple realizations of the noisy limit-state function for each parameter set $\boldsymbol{x}$. However, this would require many replications of the Monte Carlo samples, which would in turn lead to an intractable computational costs, especially for small failure probabilities.

3.1 Gaussian process regression basics

Surrogate models are a widely-used approach to reduce the need for costly model evaluations. In this approach, the expensive model is considered as a *black-box*, *i.e.*, only its inputs and outputs are known. The aim is to learn the input-output mapping, using a limited-size experimental design $\mathcal{E}$:

$$\mathcal{E} = \left\{\left(\boldsymbol{x}^{(i)}, y^{(i)}\right) : y^{(i)} = \tilde{g}\left(\boldsymbol{x}^{(i)}, \omega^{(i)}\right) \in \mathbb{R}, \, \boldsymbol{x}^{(i)} \in \mathcal{D}_{\boldsymbol{X}} \subset \mathbb{R}^M, i = 1, \dots, N\right\}, \tag{9}$$

where $\boldsymbol{x}^{(i)}$ represents an M-dimensional vector of input parameters, $y^{(i)}$ is the corresponding output from the noisy limit-state function and N is the total number of available observations.

Gaussian process modeling (Rasmussen and Williams, 2006) is a statistical technique frequently utilized in surrogate modeling. It assumes that the underlying function to approximate is one realization of an unknown Gaussian process. This model is generally expressed as:

$$\mathcal{M}^{\mathrm{GP}}\left(\boldsymbol{x}\right) = \mu\left(\boldsymbol{x}\right) + \sigma^2 Z(\boldsymbol{x};\omega), \tag{10}$$

where $\mu\left(\boldsymbol{x}\right)$ is a trend, which is assumed constant and equal to zero herein (simple Kriging), σ^2 is the process variance and $Z(\boldsymbol{x};\omega)$ is a zero-mean, unit-variance stationary process fully characterized by its auto-correlation $R\left(\boldsymbol{x}, \boldsymbol{x}';\boldsymbol{\theta}\right)$ with hyperparameters $\boldsymbol{\theta}$.

While this model is aimed at approximating the underlying limit-state function $g(\boldsymbol{x})$, the observations in the experimental design are noisy. Gaussian process regression assumes an additive noise ε which follows a zero-mean Gaussian distribution. We further assume that the noise is homoskedastic, *i.e.*, it is the same for all observations. In this case, the covariance of the noise term is $\boldsymbol{\Sigma}_n^2 = \sigma_n^2\,\mathbf{I}$, where σ_n^2 is the noise variance and $\mathbf{I}$ is the identity matrix of size $N \times N$.

By definition, the joint distribution between the response at a new location and the noisy observations $\mathcal{Y} = \left\{y^{(i)}, i = 1, \ldots, N\right\}$ is Gaussian and reads:

$$\left\{\begin{matrix} \widehat{g}(\boldsymbol{x}) \\ \mathcal{Y} \end{matrix}\right\} \sim \mathcal{N}_{N+1}\left(\mathbf{0}, \quad \left\{\begin{matrix} \sigma^2 & \mathbf{r}^T(\boldsymbol{x}) \\ \mathbf{r}(\boldsymbol{x}) & \sigma^2\mathbf{R} + \sigma_n^2\mathbf{I} \end{matrix}\right\}\right), \tag{11}$$

where $\mathbf{R}$ is an $N \times N$ auto-correlation matrix whose elements are defined by $R_{ij} = R\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}; \boldsymbol{\theta}\right)$ and $r(\boldsymbol{x})$ is the vector of cross-correlations whose elements are $\left\{r_i = R\left(\boldsymbol{x}, \boldsymbol{x}^{(i)}; \boldsymbol{\theta}\right), i = 1, \ldots, N\right\}$.

By deriving the conditional distribution $\widehat{g}(\boldsymbol{x})|\mathcal{Y} \sim \mathcal{N}\left(\mu_{\widehat{g}}(\boldsymbol{x}), \sigma_{\widehat{g}}^2(\boldsymbol{x})\right)$, we get the predictive equations for Gaussian process regression. Upon introducing the total process variance

$$\sigma_{\text{total}}^2 = \sigma^2 + \sigma_n^2, \tag{12}$$

and the ratio

$$\tau = \frac{\sigma_n^2}{\sigma_{\text{total}}^2}, \tag{13}$$

the mean and variance of $\widehat{g}(\boldsymbol{x})$ can be written as

$$\mu_{\hat{g}}(\boldsymbol{x}) = \tilde{\mathbf{r}}(\boldsymbol{x})^T\tilde{\mathbf{R}}^{-1}\mathcal{Y}, \tag{14}$$

$$\sigma_{\hat{g}}^2(\boldsymbol{x}) = \sigma_{\text{total}}^2\left(1 - \tilde{\mathbf{r}}^T(\boldsymbol{x})\tilde{\mathbf{R}}^{-1}\tilde{\mathbf{r}}(\boldsymbol{x})\right). \tag{15}$$

where $\tilde{\mathbf{r}} = (1 - \tau)\,\mathbf{r}$ and $\tilde{\mathbf{R}} = (1 - \tau)\,\mathbf{R} + \tau\,\mathbf{I}$ have been introduced for simplicity.

Eq. (14) provides the Gaussian process prediction for a new location $\boldsymbol{x}$, while Eq. (15) is the associated built-in error estimate. To fully define both equations, an auto-correlation function needs to be selected and its parameters, together with the process and noise variances, estimated. We consider here the Matérn 5/2 auto-correlation function, which reads:

$$R\left(x, x'; \theta, \nu = 5/2\right)\left(1 + \sqrt{5}\frac{|x - x'|}{\theta} + \frac{5}{3}\left(\frac{|x - x'|}{\theta}\right)^2\right)\exp\left[-\sqrt{5}\frac{|x - x'|}{\theta}\right], \tag{16}$$

The multi-dimensional case is derived using the following ellipsoidal formulation (Rasmussen and Williams, 2006):

$$R\left(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta}\right) = R(h), \qquad h = \left[\sum_{i=1}^{M}\left(\frac{x_i - x_i'}{\theta_i}\right)^2\right]^{\frac{1}{2}}. \tag{17}$$

An analytical estimate of the total variance σ_{total}^2, which is a function of the auto-correlation hyperparameters $\boldsymbol{\theta}$, is derived by maximum likelihood (Dubourg, 2011; Santner et al., 2003):

$$\hat{\sigma}_{\text{total}}^2 = \sigma_{\text{total}}^2(\boldsymbol{\theta}) = \frac{1}{N}\mathcal{Y}^T\tilde{\boldsymbol{R}}^{-1}\mathcal{Y}. \tag{18}$$

7

The hyperparameters $\boldsymbol{\theta}$ and the noise variance ratio τ are, in turn, jointly obtained by solving the following optimization problem:

$$\hat{\boldsymbol{\theta}}, \hat{\tau} = \underset{\boldsymbol{\theta} \in \mathcal{D}_{\boldsymbol{\theta}}, \tau \in (0,1)}{\arg\min} \frac{1}{2} \left[\log(\det \tilde{\boldsymbol{R}}) + N \log\left(2\pi\hat{\sigma}_{\text{total}}^2\right) + N \right]. \tag{19}$$

In this paper, we use the implementation of the Gaussian process module of the UQLAB software (Marelli and Sudret, 2014; Lataniotis et al., 2022).

3.2 Estimation of P_f via regression-based surrogate models

Once the surrogate model is trained, its mean predictor $\mu_{\hat{g}}$ can be used as a denoised version of the limit-state function $\tilde{g}$. Because $\mu_{\hat{g}}$ is a deterministic function, any reliability method can be used for computing $\hat{P}_f$. To demonstrate the feasibility of the proposed strategy, we denoise the $R - S$ function described in the previous section. To numerically solve the problem, we consider that $R \sim \mathcal{N}\left(5, 0.8^2\right)$, and $S \sim \mathcal{N}\left(2, 0.6^2\right)$. The noise term is parametrically defined as $\varepsilon \sim \mathcal{N}\left(0, \sigma_\varepsilon^2\right)$, which allows testing the robustness of the method as the noise level σ_ε^2 increases. For this particular example, we evaluate $\hat{P}_f$ using Monte Carlo simulation (Rubinstein and Kroese, 2016).

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

For each level of noise, the noise-free probability of failure is recovered when N is sufficiently large. The size N required to get fairly accurate results increases with σ_ε, however; 100 points are sufficient when $\sigma_\varepsilon = 0$ (noise-free limit-state function), whereas at least 1,000 to 10,000 are necessary for larger noise levels.
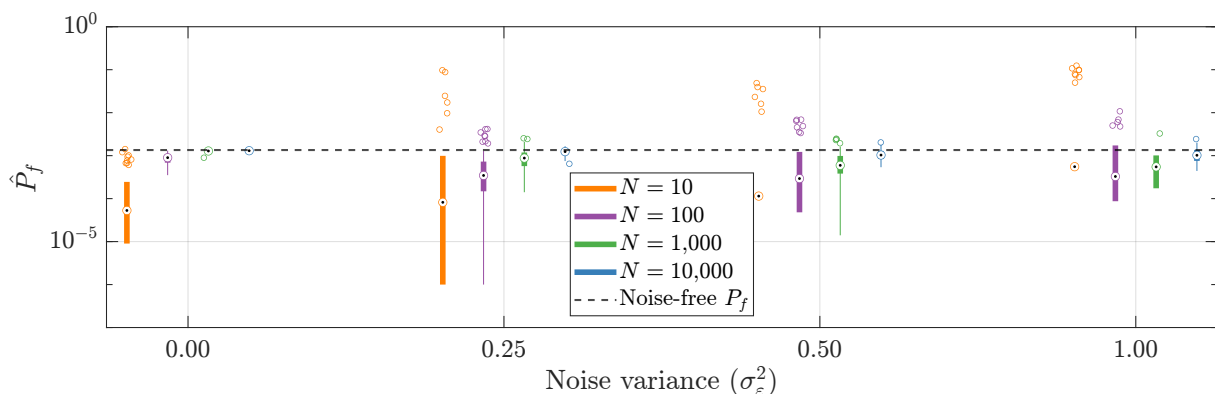

Figure 1: Noisy $R - S$ problem – Estimation of the noise-free P_f using a Gaussian process model based on N points (obtained from the noisy limit-state function $\tilde{g}$ (Eq. (2))) and for different values of the noise variance.

Nevertheless, relying on large EDs can be unfeasible in practice. First, obtaining such a large ED can be costly in the presence of computationally expensive limit-state functions. Second, as the size of the ED increases, the training and prediction steps can become expensive (Chevalier et al., 2014). To overcome these challenges and improve the general efficiency of the method, we employ an adaptive approach based on state-of-the-art active learning-based reliability analysis methods (Teixeira et al., 2021; Moustapha et al., 2022) developed for deterministic limit-state functions.

4 Active learning methods for reliability analysis

Active learning methods are iterative, surrogate-based approaches to efficiently solve complex reliability analysis problems (Teixeira et al., 2021). A Gaussian process is initially trained using an experimental design of small size, which is iteratively expanded. Essentially, the algorithm identifies at each iteration the most informative data points to add to the current ED, with the goal of maximizing the surrogate accuracy. Thanks to this rational selection of training points, this method significantly reduces the computational resources required for estimating the probability of failure.

Moustapha et al. (2022) show that the vast majority of active learning methods feature four main components: a surrogate model, a reliability estimation method, a learning function and a stopping criterion. The surrogate model is the most crucial component of the method as its accuracy determines that of the estimated probability of failure. The reliability estimation method dictates how the probability of failure is calculated at each iteration. The learning function is responsible for selecting the point or set of points to be added to the ED at each iteration. The stopping criterion governs the termination of the algorithm. A step-by-step breakdown of the method and an illustration of how these components work together is shown in Algorithm 1.

Algorithm 1 Pseudo-code for active learning reliability methods
---
Require: $\mathcal{E}^{(0)}$ (Initial ED)

    $i \leftarrow 0$

    repeat

        Build *surrogate model* $\hat{g}^{(i)}\left(\boldsymbol{x}; \mathcal{E}^{(i)}\right)$

        Compute $P_f^{(i)}$ using $\hat{g}^{(i)}\left(\boldsymbol{x}; \mathcal{E}^{(i)}\right)$ and a *reliability estimation method*

        Select $\boldsymbol{x}^{next}$ using the *learning function*

        Enrich the ED: $\mathcal{E}^{(i+1)} \leftarrow \mathcal{E}^{(i)} \cup \left\{\boldsymbol{x}^{next}, \tilde{g}\left(\boldsymbol{x}^{next}\right)\right\}$

        $i \leftarrow i + 1$

    until *stopping criterion* is fulfilled

We structure our active learning algorithm as follows. First, we select Gaussian process regression as surrogate model. In principle, any regression-based surrogate model that enables

active learning, such as GPR, polynomial chaos expansions or support vector machines, can be used. For reliability estimation, we employ subset simulation, configured according to the *overkill* setup introduced in Moustapha et al. (2022). Given that the enrichment step significantly impacts the efficiency and must take the noise into account in our case, a more comprehensive discussion is provided in the following section. Lastly, we establish a maximum budget of evaluations of the (noisy) limit-state function as the stopping criterion.

4.1 Enrichment step

During the enrichment step, we determine the next set of training points to include in the experimental design. This process is guided by a *learning function* $\mathcal{L}$, which is a utility function that assigns numerical scores to any point $\boldsymbol{x}$ within the domain. This score indirectly quantifies the information gained by introducing this point $\boldsymbol{x}$ as an additional training point. Notably, in the context of reliability analysis, these learning functions focus on minimizing the chance that the surrogate model makes a mistake in the sign of its prediction, leading to the misclassification of points on the safe domain as belonging to the failure domain, and vice-versa.

In theory, the scores provided by the learning function serve as the basis for solving an optimization problem that ultimately selects the points for inclusion in the ED. However, because computing the learning function is usually inexpensive, it is common practice to simplify this optimization problem into a discrete one, avoiding the need for complex optimization algorithms. In such cases, a *candidate set* $\mathcal{X}_c$ comprises a pre-defined collection of points that are potential candidates for inclusion in the ED. The learning function then computes the score of each point of $\mathcal{X}_c$, which enables selecting the best enrichment point:

$$\boldsymbol{x}^{\text{next}} = \underset{\boldsymbol{x} \in \mathcal{X}_c}{\arg\max} \, \mathcal{L}(\boldsymbol{x}). \tag{20}$$

Note that a batch of points can be selected at each step instead of a single one, using for instance clustering methods (Dubourg et al., 2013; Schöbi et al., 2017).

Because Gaussian process models are the most commonly used surrogate models in reliability analysis, many learning functions capitalising on their properties are available in the literature, *e.g.* Bichon et al. (2008); Echard et al. (2011); Lv et al. (2015); Zhang et al. (2019); Yi et al. (2020); Shi et al. (2020). They are usually based on both the GP mean and variance. The critical distinction among these functions lies in how they define information. For instance, the *U* learning function by Echard et al. (2011) identifies the point whose sign is most likely to be mistakenly predicted by the GP model, whereas the EFF (Bichon et al., 2008) aims at estimating the potential improvement over the current prediction. In contrast, the *H* learning function proposed by Lv et al. (2015) relies on information entropy.

These learning functions generally aim to balance two competitive behaviors: *exploration* and *exploitation*. The former refers to the ability of the learning function to identify enrichment points

in regions of high uncertainty and possibly discover multiple disjoint failure regions throughout the domain. The latter is related to the ability to efficiently refine the surrogate model in the vicinity of the current limit-state surface approximation. There are however two caveats to their use. On the one hand, learning functions with a highly explorative behavior may favor points in regions with insignificant probability mass and may lead to failure in accurately describing the limit-state surface. On the other hand, learning functions with highly exploitative behavior tend to select points close to one another, accurately describing a particular region of the limit-state surface but failing to find all failure regions.

All the learning functions mentioned above pose a challenge when dealing with noisy problems. They rely on the fact that the variance of the Gaussian process is zero at all points of the experimental design and tends to zero for points in their neighborhood. This is *not* the case when considering Gaussian process regression for noisy limit-state functions. Indeed, the minimum GPR variance we can expect is the learned noise level σ_n^2. For this reason, these learning functions will show a strong tendency to repeatedly identify enrichment points within a confined region at the expense of other potential failure regions, a behavior reported by Wackers et al. (2020); Ficini et al. (2021). To address this issue, noise-aware learning functions are needed. In the following section, we present a noise-aware extension of the well-known U learning function from Echard et al. (2011).

4.1.1 Learning function U_N

The U_N learning function, proposed by van den Eijnden et al. (2021), aims at identifying the point that will reduce the probability of misclassification P_m the most. The latter was originally introduced by Echard et al. (2011) and corresponds to the probability that the surrogate makes a mistake in the sign of a prediction, *i.e.*, classifies as safe a sample in the failure domain or vice-versa. It can be computed as follows:

$$P_m\left(\boldsymbol{x}\right) = \Phi\left(-\frac{|\mu_{\hat{g}}(\boldsymbol{x})|}{\sigma_{\hat{g}}(\boldsymbol{x})}\right), \tag{21}$$

where $\mu_{\hat{g}}(\boldsymbol{x})$ is the surrogate prediction, $\sigma_{\hat{g}}(\boldsymbol{x})$ is the associated standard deviation of the GP, and Φ is the standard normal cumulative distribution function (CDF) (Bect et al., 2012).

The U_N learning function at a given point $\boldsymbol{x}$ is defined as the difference between the current probability of misclassification and the one computed after the point is added to the ED, that is:

$$U_N\left(\boldsymbol{x}\right) = P_m\left(\boldsymbol{x}\right) - P_{m+1}\left(\boldsymbol{x}\right), \tag{22}$$

which is equivalent to,

$$U_N\left(\boldsymbol{x}\right) = \Phi\left(-\frac{|\mu_{\hat{g}}(\boldsymbol{x})|}{\sigma_{\hat{g}}(\boldsymbol{x})}\right) - \Phi\left(-\frac{|\mu_{\hat{g}+1}(\boldsymbol{x})|}{\sigma_{\hat{g}+1}(\boldsymbol{x})}\right), \tag{23}$$

where $P_{m+1}(\boldsymbol{x})$ is the residual probability of misclassification after $\boldsymbol{x}$ is added to the ED, while $\mu_{\hat{g}+1}(\boldsymbol{x})$ and $\sigma_{\hat{g}+1}(\boldsymbol{x})$ are the corresponding next step Gaussian process prediction and standard deviation.

van den Eijnden et al. (2021) suggest estimating $P_{m+1}(\boldsymbol{x})$ through a one-step look-ahead approach. The major issue with this approach is that it requires computing the Gaussian process prediction after fitting the model with the additional training point for all candidate points. Because this would require running the expensive procedure for all $\boldsymbol{x} \in \mathcal{X}_{\mathrm{c}}$, the authors propose approximating $\mu_{\hat{g}+1}(\boldsymbol{x})$ by $\mu_{\hat{g}}(\boldsymbol{x})$. In this case, leveraging on the Gaussian properties of the model, it is possible to express the one-step look-ahead variance $\sigma_{\hat{g}+1}^2(\boldsymbol{x})$ and misclassification probability $P_{m+1}(\boldsymbol{x})$ with the following closed-form solutions:

$$\sigma_{\hat{g}+1}^2(\boldsymbol{x}) = \sigma_{\hat{g}}^2(\boldsymbol{x}) \frac{\sigma_n^2}{\sigma_{\hat{g}}^2(\boldsymbol{x}) + \sigma_n^2} \tag{24}$$

and

$$P_{m+1}(\boldsymbol{x}) = \Phi\left(-\frac{|\mu_{\hat{g}}(\boldsymbol{x})|}{\sigma_{\hat{g}+1}(\boldsymbol{x})}\right). \tag{25}$$

As conclusion, the U_{N} learning function boils down to:

$$U_{\mathrm{N}}(\boldsymbol{x}) = \Phi\left(-\frac{|\mu_{\hat{g}}(\boldsymbol{x})|}{\sigma_{\hat{g}}(\boldsymbol{x})}\right) - \Phi\left(-\frac{|\mu_{\hat{g}}(\boldsymbol{x})|\sqrt{\sigma_{\hat{g}}^2(\boldsymbol{x}) + \sigma_n^2}}{\sigma_{\hat{g}}(\boldsymbol{x})\,\sigma_n}\right). \tag{26}$$

Finally, the next sample is chosen as follows:

$$\boldsymbol{x}^{\mathrm{next}} = \arg\max_{\boldsymbol{x} \in \mathcal{X}_{\mathrm{c}}} U_N(\boldsymbol{x}). \tag{27}$$

4.1.2 Multi-point enrichment

When multiple computing cores are available, enriching the surrogate models with batches of K points at each iteration can significantly reduce the computation time of the reliability analysis (Schöbi et al., 2017). However, as discussed in Chevalier et al. (2014), this approach may result in a suboptimal experimental design compared to one of the same size enriched with one point at a time.

Many criteria for selecting multiple points at a time exist. Sacks et al. (1989) is one of the first works to address this issue and proposes a sequential look-ahead approach for selecting the multiple points. In this case, the learning function is run K times for each step of the enrichment process, selecting a single point at each interaction. More recently, single-stage methods have been preferred. In this case, the K points are found in the same loop. To achieve this, Schöbi et al. (2017) suggests employing partitioning through weighted $K-$means, with the probability of misclassification serving as the weight. Lelièvre et al. (2018) propose a similar approach but employ a different weighting method.

To alleviate the computational burden of clustering, we reduce the size of $\mathcal{X}_c$ according to Schöbi et al. (2017). Then, we perform K-means clustering (Zaki and Meira, 2014) on the reduced set of candidate points and select the optimal point within each given cluster. To perform the partition, the user must define the number K of clusters in advance. We consider 1, 3 and 10 partitions in the sequel.

5   Results on benchmark applications

5.1   Artificially corrupting noise-free models

In this section, we show the results of our denoising methodology for two widely used benchmarks, the hat function and the four-branch series system, and for a finite element model of a realistic structural frame. Because these are noise-free models, we corrupt them with noise before testing the proposed methodology. In real applications with noisy models, this step would not be necessary.

All tested models are corrupted with additive noise, following Eq. (2). Moreover, Picheny et al. (2013) suggest the noise to be normally distributed as this is widely accepted due to its simplicity. Consistent with this approach and considering homoskedastic unbiased noise, each realization of the noise component is sampled from independent and identically distributed (i.i.d) normal random variables $\mathcal{N}(0, \sigma_\varepsilon)$. Consequently, determining the noise level σ_ε suffices for corrupting the noise-free model.

Specifying a suitable value for σ_ε can be challenging, especially when comparing different benchmark functions, where no physical meaning can be linked to the noise level. The reason behind this is that the gradient of the limit-state function significantly affects how the failure probability responds to the noise variance. In general, functions with steeper gradients close to the limit state surface are more robust to noise, whereas functions with flatter regions are more sensitive to it. As an illustration, we introduce a constant $\gamma > 0$ for controlling the norm of the gradient in the previously mentioned $R - S$ example, as follows:

$$g\left(\boldsymbol{X}\right) = \gamma\left(R - S\right) \text{ and } \tilde{g}\left(\boldsymbol{X}, \omega\right) = \gamma\left(R - S\right) + \varepsilon\left(\omega\right). \tag{28}$$

The norm of the gradient of the limit-state function is constant and corresponds to $\|\nabla \tilde{g}\| = \gamma \sqrt{2}$. The analytical noise-free and noisy probabilities of failure respectively read:

$$P_f = \Phi\left(-\frac{\mu_R - \mu_S}{\sqrt{\sigma_R^2 + \sigma_S^2}}\right) \text{ and } \tilde{P}_f = \Phi\left(-\frac{\gamma\left(\mu_R - \mu_S\right)}{\sqrt{\gamma^2\left(\sigma_R^2 + \sigma_S^2\right) + \sigma_\varepsilon^2}}\right). \tag{29}$$

Eq. (29) shows that γ does not affect P_f, but has an influence on its noisy counterpart. Moreover, to understand the relation between the gradient and the noise component, an asymptotic study

can be carried out. In this case, a steep gradient corresponds to $\gamma \to \infty$, whereas a flat gradient is given by $\gamma \to 0$. Taking the limit of $\tilde{P}_f$ for these cases leads to:

$$\lim_{\gamma \to \infty} \tilde{P}_f = P_f \text{ and } \lim_{\gamma \to 0} \tilde{P}_f = 0.5. \tag{30}$$

Eq. (30) shows that noise does not significantly impact limit-state surfaces with steep gradients, and the estimated $\tilde{P}_f$ converges to the noise-free one. On the other hand, it reveals that noise severely affects a rather flat limit-state surface, ultimately leading to a constant 50% probability of failure, which corresponds to the probability of failure obtained for the noise term alone.

To take into account the gradient around the limit-state surface when corrupting the models, we propose defining σ_ε as a function of the variance around its vicinity. Consequently, the obtained noise level can be made comparable across different benchmark limit-state functions. To this aim, we first introduce a *region of interest* (ROI) by defining margins of the limit-state function using quantiles:

$$ROI(\alpha; \mathcal{Y}) = \{y \in \mathcal{Y} : |y| < q_\alpha\}, \tag{31}$$

where q_α is defined by:

$$q_\alpha = \inf\{q \in \mathbb{R} : \mathbb{P}\left(|g(\mathbf{X})| < q\right) > \alpha\}, \tag{32}$$

and $\mathcal{Y}$ is defined by:

$$\mathcal{Y} = \left\{g(\boldsymbol{x}^{(1)}), \dots, g(\boldsymbol{x}^{(n)})\right\}, \tag{33}$$

where $\left\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(n)}\right\}$ corresponds to samples drawn according to $f_{\mathbf{X}}$.

Finally, our proxy to the level of introduced noise is defined as the variance of the samples inside the ROI, as follows:

$$\sigma_\varepsilon^2(\alpha) \stackrel{\text{def}}{=} \text{Var}\left[g(\boldsymbol{X}) \mid g(\boldsymbol{X}) \in ROI\right]. \tag{34}$$

With this approach, the noise level is a function of $\alpha \in (0, 1)$. The closer α is to 0, the milder the noise is, independently of the limit-state function. As this method requires simulating many samples, it can only be used for inexpensive benchmark functions. In actual applications, however, a physical meaning should be attributed to the noise level whenever possible, typically related to the expected measurement or numerical noise levels. Again, the procedure described in this section is only used to build coherent noisy benchmark functions and facilitate cross-comparison.

5.2 Four-branch function

The four-branch function, initially proposed by Waarts (2000), has been commonly used to benchmark reliability estimation algorithms. The corresponding reliability analysis problem has a total of four distinct failure regions. Converging to the correct probability of failure is,

therefore, non-trivial, and methods such as FORM/SORM and importance sampling tend to significantly underestimate the probability of failure. The noise-free limit-state function reads:

$$g(\boldsymbol{x}) = \min \left\{ \begin{array}{c} 3 + 0.1\,(x_1 + x_2)^2 - \frac{x_1 + x_2}{\sqrt{2}} \\ 3 + 0.1\,(x_1 + x_2)^2 + \frac{x_1 + x_2}{\sqrt{2}} \\ (x_1 - x_2) + \frac{6}{\sqrt{2}} \\ (x_2 - x_1) + \frac{6}{\sqrt{2}} \end{array} \right\}. \tag{35}$$

This problem comprises two random input variables, modeled as independent standard normal distributions, *i.e.*, $X_1, X_2 \sim \mathcal{N}(0, 1)$. The reference noise-free probability of failure is computed via Monte Carlo simulation using 10^7 samples and is equal to $4.51 \cdot 10^{-3}$. The associated coefficient of variation is smaller than 1%.

We have chosen this model to test our denoising methodology, because learning functions generally tend to get stuck in exploitative loops for this problem, *i.e.*, they fail to discover the four branches. This toy example is corrupted according to the methodology described in Section 5.1. Due to noise corruption, depicting the limit-state surface is impossible. Instead, Figure 2a depicts a heatmap of the probability of misclassification caused solely by the noise component, defined as:

$$\tilde{P}_m(\boldsymbol{x}) = \Phi\left(-\frac{|g(\boldsymbol{x})|}{\sigma_\varepsilon}\right). \tag{36}$$

Figure 2a was obtained for a model corrupted with $\alpha = 0.05$.

The Gaussian process model is trained with an initial ED containing $N_{\mathrm{ini}} = 10$ points, obtained via space-filling Latin hypercube sampling (LHS, McKay et al., 1979). Enrichment is performed with the U_N learning function, and three enrichment strategies are tested by adding 1, 3 or 10 points per iteration. Finally, the algorithm is terminated after 600 points are added. Each setup of the problem is repeated 50 times with different initial experimental designs.

Figure 3 depicts the results of these experiments for two different levels of noise: $\alpha = 0.01$ and 0.05. Figure 3a shows that the learning function can converge to the noise-free P_f in all tested scenarios. Moreover, no significant difference in performance is observed for multi-point enrichment. Figure 3b depicts the boxplots showing the convergence behavior of the experiments. Additionally, it is possible to observe that the model converges to P_f with around 200 points, suggesting that a proper stopping criterion might reduce the required computational effort. Finally, Figure 2b depicts the denoised limit-state surface for the experiment that leads to the median $\hat{P}_f$ for $\alpha = 0.05$. The observed alignment of the points around the limit-state surface can be attributed to a limit behavior of the U_N learning function, a feature explored in greater detail in Appendix A.
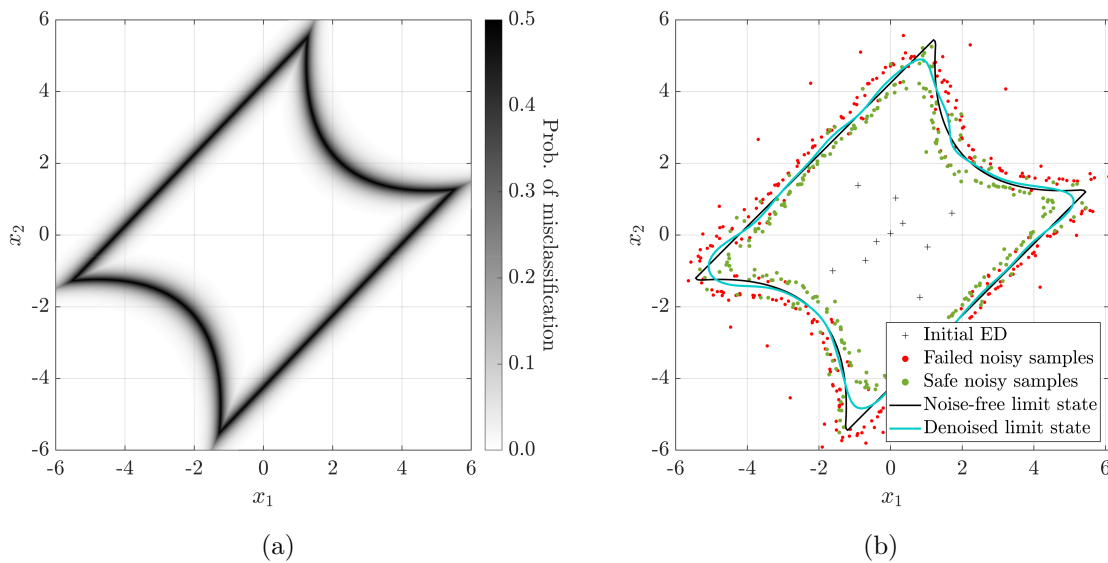
Figure 2: Four-branch function – (a): heatmap of the probability of misclassification as described in Eq. (36) for a noise level of $\alpha = 0.05$. (b): noise-free limit-state surface (black line), final limit-state surface estimated by the GPR (cyan line), initial ED (black cross). The green (resp. red) dots are the safe (resp. failed) samples selected by the U_N learning function for a noise level of $\alpha = 0.05$.
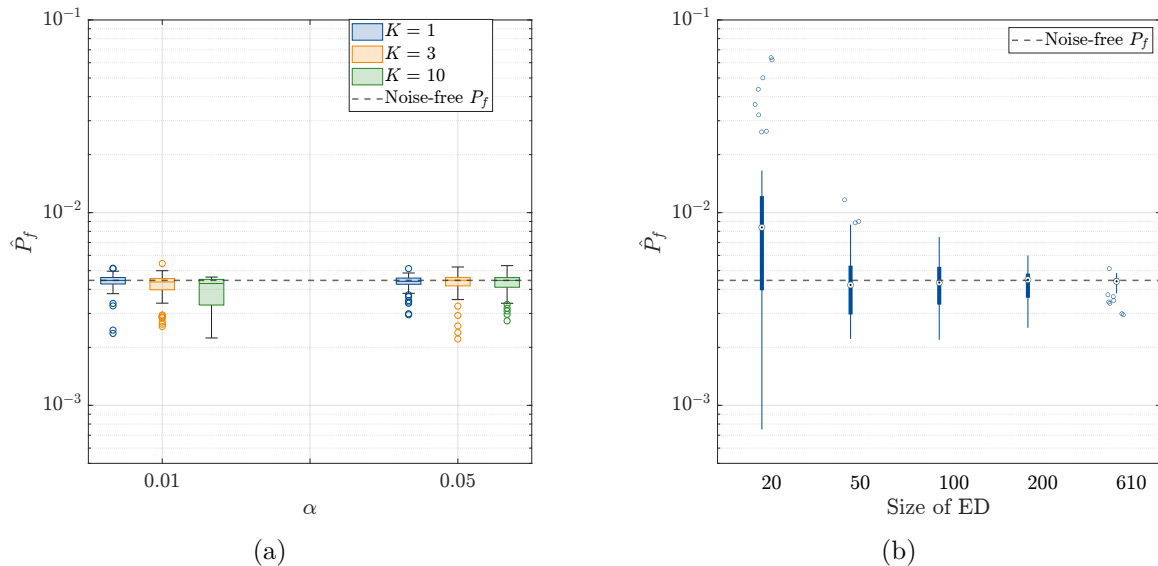

Figure 3: Four-branch function – (a): final probability of failure estimated using GPR with learning function U_N. (b): boxplots depicting the convergence behavior for $\alpha = 0.05$ and $K = 1$.

5.3 Hat function

The Hat function is another frequently used benchmark in reliability analysis. It is a polynomial function that reads:

$$g(\boldsymbol{x}) = 12 - (x_1 - x_2)^2 - 8(x_1 + x_2 - 4)^3. \tag{37}$$

The random input vector comprises two i.i.d normal random variables $X_1, X_2 \sim \mathcal{N}(0.25, 1)$. The reference $P_f = 9.76 \cdot 10^{-4}$ was computed via Monte Carlo simulation using $2 \cdot 10^7$ samples, and its associated coefficient of variation is smaller than 1%.

Similarly to the previous example, the limit-state function is corrupted according to the methodology described in Section 5.1. In this example, the limit-state surface lies in a flat region (small gradient), which hinders the denoising process, as discussed above. A graphical illustration of this issue is given in Figure 4a, which depicts the heatmap of the probability of misclassification for $\alpha = 0.05$. A considerable area around the limit-state surface presents a significant probability of misclassification, since even a small amount of noise can swap the sign of $g(\boldsymbol{x})$.

Similarly to the previous example, 50 experiments are carried out, and the size of the initial ED for each replication corresponds to $N_{\text{ini}} = 10$ points obtained via LHS. Moreover, three enrichment strategies are tested, adding 1, 3 and 10 points at a time. The algorithm is terminated after 600 points are added to the experimental design.

Figure 5a depicts the boxplots of the computed probability of failure for two levels of noise. All setups converge accurately to the noise-free probability of failure. Figure 5b showcases the convergence of the estimator $\hat{P}_f$ with respect to the size of the experimental design. It reveals that the algorithm converged to the noise-free probability of failure already at around 200 samples. Finally, Figure 4b depicts the denoised limit-state surface and final ED of the experiment, which yielded the median $\hat{P}_f$ for $\alpha = 0.05$.
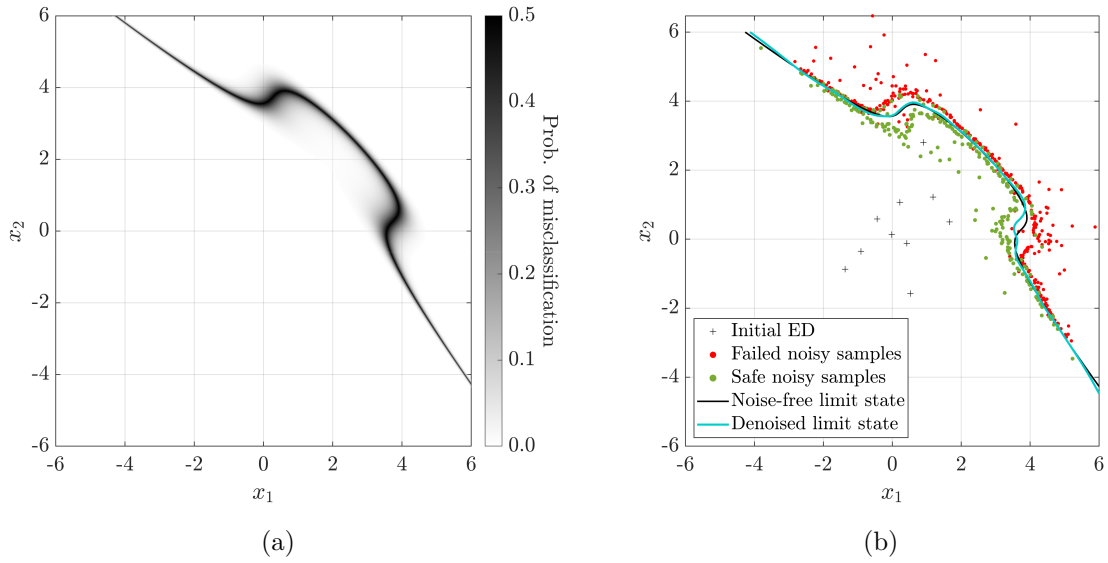
Figure 4: Hat function – (a): heatmap of the probability of misclassification as described in Eq. (36) for a noise level of $\alpha = 0.05$. (b): noise-free limit-state surface (black line), final limit-state surface estimated by the GPR (cyan line), initial ED (black cross). The green (resp. red) dots are the safe (resp. failed) samples selected by the learning function.
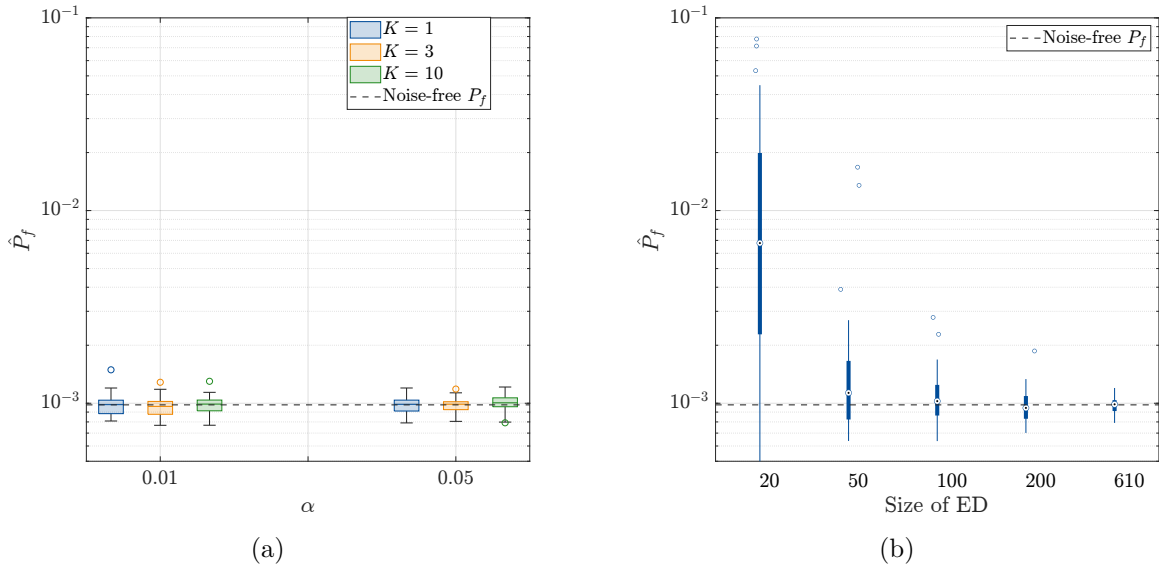

Figure 5: Hat function – (a): final probability of failure estimated using GPR with learning function U_{N}. (b): boxplots depicting the convergence behavior for $\alpha = 0.05$ and $K = 1$.

5.4   Structural frame

Real-world applications are generally high-dimensional and more complex than the previous benchmark examples. To showcase the performance of the algorithm for such cases, we apply the methodology to the finite element model of a structural frame depicted in Figure 6. This

problem was first introduced in the reliability analysis literature by Liu and Der Kiureghian (1991) and has often been used in the context of surrogate modeling and reliability analysis (Blatman and Sudret, 2010; Marelli and Sudret, 2018; Moustapha et al., 2022). Of interest is the probability that the horizontal displacement u of the right top corner of the building exceeds a critical threshold u_{max}. The associated limit-state function reads:

$$g\left(\boldsymbol{x}\right) = u_{max} - u\left(\boldsymbol{x}\right). \tag{38}$$

We consider a maximum allowed displacement $u_{max} = 50$ mm. The top-floor displacement of the building is computed using a finite element model, where geometrical and physical non-linearities are not considered. Additionally, the model takes 21 random variables as input. P_1, P_2, P_3 are the horizontal loads (see Figure 6). The geometric and material properties, such as Young's moduli, moments of inertia, and cross-sections, are provided respectively $E_{4,5}$, $I_{6,...,13}$ and $A_{14,...,21}$, and their marginal distributions are given in Table 1. Table 2 displays the structural properties of each structural element depicted in Figure 6. The dependence between the random input variables is modelled using a Gaussian copula, and its correlation matrix $\mathbf{R}$ is defined as follows:

- The correlation between the cross-sectional area of a given element and its inertia moment is set equal to $R_{A_i,I_i} = 0.95$;

- The correlation between properties of different elements is set equal to $R_{A_i,I_j} = R_{I_i,I_j} = R_{A_i,A_j} = 0.13$;

- The correlation between Young's moduli is set equal to $R_{A_i,A_j} = 0.90$;

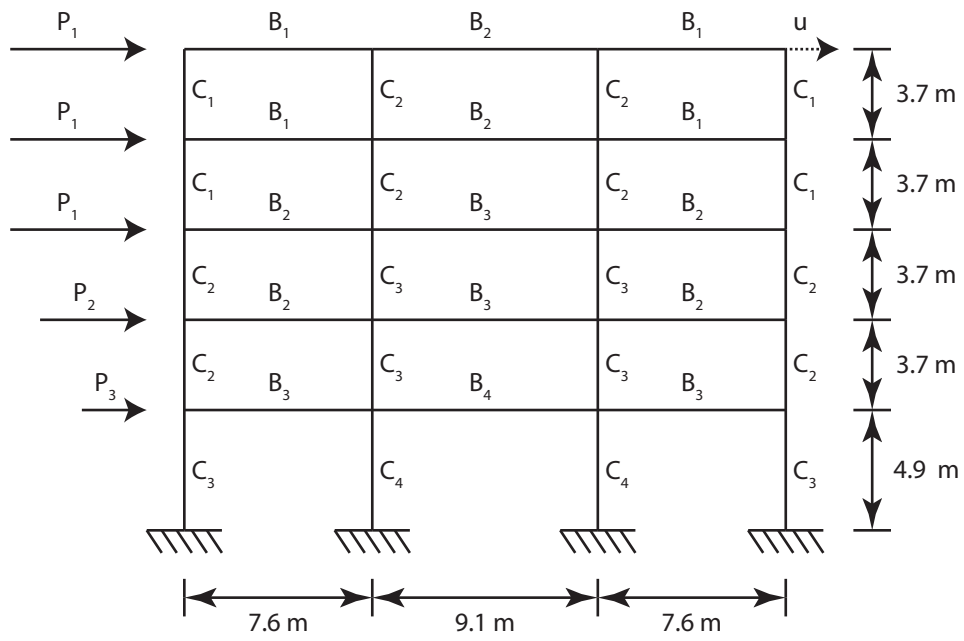- The remaining correlations coefficients of $\mathbf{R}$ are set equal to 0.

Figure 6: Sketch of a 3-span, 5-story structural frame.

Variable	Distribution	Mean	Standard Deviation
P_1 (kN)	Lognormal	133.454	40.04
P_2 (kN)	Lognormal	88.97	35.59
P_3 (kN)	Lognormal	71.175	28.47
E_4 (kN/m^2)	Truncated Gaussian	2.1738×10^7	3.8304×10^6
E_5 (kN/m^2)	Truncated Gaussian	2.3796×10^7	3.8304×10^6
I_6 (m^4)	Truncated Gaussian	8.1344×10^{-3}	1.0834×10^{-3}
I_7 (m^4)	Truncated Gaussian	1.1509×10^{-2}	1.2980×10^{-3}
I_8 (m^4)	Truncated Gaussian	2.1375×10^{-2}	2.5961×10^{-3}
I_9 (m^4)	Truncated Gaussian	2.5961×10^{-2}	3.0288×10^{-3}
I_{10} (m^4)	Truncated Gaussian	1.0812×10^{-2}	2.5961×10^{-3}
I_{11} (m^4)	Truncated Gaussian	1.4105×10^{-2}	3.4615×10^{-3}
I_{12} (m^4)	Truncated Gaussian	2.3279×10^{-2}	5.6249×10^{-3}
I_{13} (m^4)	Truncated Gaussian	2.5961×10^{-2}	6.4902×10^{-3}
A_{14} (m^2)	Truncated Gaussian	3.1256×10^{-1}	5.5815×10^{-2}
A_{15} (m^2)	Truncated Gaussian	3.7210×10^{-1}	7.4420×10^{-2}
A_{16} (m^2)	Truncated Gaussian	5.0606×10^{-1}	9.3025×10^{-2}
A_{17} (m^2)	Truncated Gaussian	5.5815×10^{-1}	1.1163×10^{-1}
A_{18} (m^2)	Truncated Gaussian	2.5302×10^{-1}	9.3025×10^{-2}
A_{19} (m^2)	Truncated Gaussian	2.9117×10^{-1}	1.0232×10^{-1}
A_{20} (m^2)	Truncated Gaussian	3.7303×10^{-1}	1.2093×10^{-1}
A_{21} (m^2)	Truncated Gaussian	4.1860×10^{-1}	1.9537×10^{-1}

Table 1: Marginal distributions associated with the probabilistic input of the structural frame shown in Figure 6. All Gaussian are truncated in the domain $[0, +\infty]$. The quoted moments refer to the full, untruncated Gaussian distributions.

Element	Young's modulus	Moment of inertia	Cross-sectional area
B_1	E_4	I_{10}	A_{18}
B_2	E_4	I_{11}	A_{19}
B_3	E_4	I_{12}	A_{20}
B_4	E_4	I_{13}	A_{21}
C_1	E_5	I_6	A_{14}
C_2	E_5	I_7	A_{15}
C_3	E_5	I_8	A_{16}
C_4	E_5	I_9	A_{17}

Table 2: Material and geometrical properties of the structural elements depicted in Figure 6.

The associated noise-free probability of failure was computed via Monte Carlo simulation using $2 \cdot 10^6$ samples and is equal to $5.07 \cdot 10^{-3}$. Its associated coefficient of variation is smaller than 1%.

Similar to the previous examples, the model is corrupted by i.i.d Gaussian random variables $\mathcal{N}(0, \sigma_\varepsilon)$. In this case, however, it is possible to attribute a physical meaning to σ_ε by considering a hypothetical scenario where measuring the top-floor displacement is possible. Under these conditions, the noise level can be associated with the accuracy of the measuring device. The accuracy varies according to the device and the techniques on which they rely. In the case of top-floor displacement, the measurement is usually done by GPS or vision-based techniques. Regarding GPS techniques, Nickitopoulou et al. (2006); Lee and Shinozuka (2006) report up to ± 10 mm of accuracy, and some more recent papers (Sofi et al., 2022) report a precision of a few millimeters. Regarding vision-based techniques, Guo et al. (2020) report accuracies spanning from a few millimeters to a few centimeters, depending on the equipment and the distance of the measured object. Aiming to account for the significant variability in accuracy between different methods, we consider a noise level σ_ε of 10 mm, which results in a challenging denoising problem, especially when compared to the critical threshold of 50 mm.

For this example, the initial ED comprises $N_{\text{ini}} = 150$ points sampled using LHS. This larger computational budget aims to address both the high dimensionality of the model and the presence of significant noise. The surrogate model is iteratively enriched based on the U_N learning function, and similarly to the previous experiments, different enrichment strategies are tested, *i.e.*, introducing 1, 3, and 10 points at a time. The maximum budget allowed for the enrichment step for this example is 1,000 samples. Once again, 50 analyses are carried out.

Figure 7a displays the performance of the algorithm. The noisy and noise-free probabilities of failure are computed via Monte Carlo simulation and are $1.70 \cdot 10^{-2}$ and $5.07 \cdot 10^{-3}$, respectively. In this case, the noisy probability of failure is more than three times larger than its noise-free counterpart, causing a change in the order of magnitude of the probability of failure. Our algorithm converges to a median estimate of $\hat{P}_f = 7.26 \cdot 10^{-3}$ for $K = 1$. Unlike the previous examples, these results converge to a higher probability of failure compared to the noise-free reference. Nevertheless, we consider these results satisfactory, given the challenging nature of this example, which is characterised by both high dimensionality and high noise level.

Figure 7a depicts the results obtained by the different enrichment strategies, showing that introducing multiple points simultaneously leads to more effective denoising of the limit-state function. These results could suggest that the U_N learning function favors exploitative behavior for $K = 1$. Indeed, despite extensive testing, the exact reason for such results could not be determined. Nonetheless, given that overly exploitative behavior is not desired in learning functions, alternative ones are currently being investigated. For instance, SUR (Bect et al., 2012) has been shown to be promising but it is computationally more expensive. Finally, Figure 7b shows the convergence curve for the replication that led to the estimation of the median denoised probability of failure. It is possible to notice that the presence of noise intensifies the already
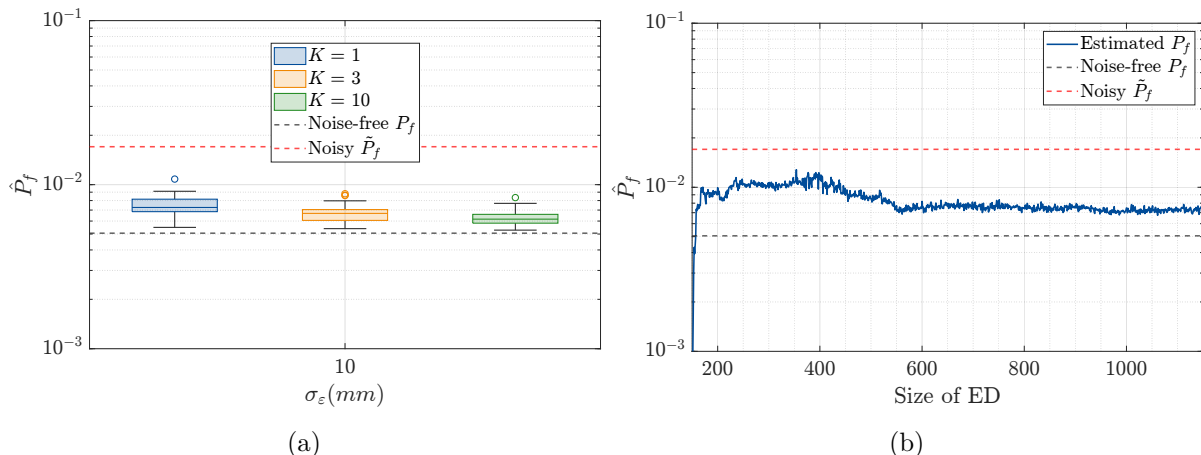
22

noisy behavior of the convergence curves.

(a)

(b)

Figure 7: Structural frame – (a): final probability of failure estimated using GPR on the limit-state function depicted in Eq. (38), corrupted with a noise level of 10 mm. (b): convergence curve of the experiment that resulted in the median $\hat{P}_f$ when enriched with one point at a time. The reference noise-free P_f and the noisy $\tilde{P}_f$ are shown by the black and red dashed lines, respectively.

6 Conclusions

In this paper, we formalize a new category of problems in reliability analysis, characterized by a limit-state function corrupted by noise. Our analysis reveals that the latter causes simulation-based reliability methods to estimate an incorrect, larger probability of failure. In this scenario, the main objective is to estimate the probability of failure associated with the noise-free limit-state function, *i.e.*, the noise-free probability of failure.

We show that computing the noise-free probability of failure can be achieved through denoising. For this purpose, we propose the use of regression-based surrogate models, and focus particularly on Gaussian process regression. We demonstrate the feasibility of the proposed approach with the $R - S$ problem by showing that one can successfully recover the noise-free probability of failure using GPR when the experimental design is sufficiently large. However, as a large experimental design can be computationally demanding, we introduce an active learning approach to minimize the computational cost associated with the problem. Within this framework, we adopt the U_N learning function, which is noise-aware.

Furthermore, we discuss the significant impact of the gradient around the limit-state surface in the denoising process. We also propose a novel methodology to create coherent noisy problems from noise-free benchmarks. Our results, obtained from two benchmark functions and a realistic structural frame, confirm the efficiency of our active learning approach in estimating the noise-free probability of failure in complex scenarios such as series systems, flat-gradient limit-state surfaces, and high-dimensional models. The final example demonstrates the feasibility of reliability analysis

23

on experimental systems where the limit-state is *measured* by an appropriate device instead of *computed* by a simulator. Moreover, we show that the U_N learning function not only allows the method to effectively converge to the noise-free probability of failure but also maintains efficiency in a multi-point enrichment strategy. Notably, in one of the case studies, the multi-point enrichment improves the estimation of the probability of failure, suggesting that the U_N function might be prone to exploitative loops. To overcome this issue, we are currently exploring alternative noise-aware learning functions and enrichment strategies.

While our study focuses on homoskedastic Gaussian noise, the unique requirement for our method is unbiased noise. However, the effectiveness of Gaussian process regression in handling heteroskedastic non-Gaussian noise remains to be studied. Finally, developing a suitable stopping criterion for our approach, crucial for optimizing its efficiency and applicability, is currently under investigation.

A   Limit behavior of the U_N learning function

In this section, we show that, if the predicted Gaussian process variance $\sigma_{\hat{g}}^2(\boldsymbol{x})$ converges to the estimated noise level σ_n^2, the optimization problem described in Eq. (27) becomes non-informative. With this aim, we first compute the one-step look-ahead variance $\sigma_{\hat{g}+1}^2(\boldsymbol{x})$ when $\sigma_{\hat{g}}^2(\boldsymbol{x}) \to \sigma_n^2$, as follows:

$$\lim_{\sigma_{\hat{g}}^2(\boldsymbol{x}) \to \sigma_n^2} \sigma_{\hat{g}+1}^2(\boldsymbol{x}) = \lim_{\sigma_{\hat{g}}^2(\boldsymbol{x}) \to \sigma_n^2} \sigma_{\hat{g}}^2(\boldsymbol{x}) \frac{\sigma_n^2}{\sigma_{\hat{g}}^2(\boldsymbol{x}) + \sigma_n^2} = \frac{\sigma_n^2}{2}. \tag{39}$$

Inserting Eq. (39) into Eq. (23) and considering that $\mu_{\hat{g}+1}(\boldsymbol{x}) = \mu_{\hat{g}}(\boldsymbol{x})$, we obtain:

$$U_N(\boldsymbol{x}) = \Phi\left(-\frac{|\mu_{\hat{g}}(\boldsymbol{x})|}{\sigma_n}\right) - \Phi\left(-\frac{\sqrt{2}\,|\mu_{\hat{g}}(\boldsymbol{x})|}{\sigma_n}\right). \tag{40}$$

Then, we show that Eq. (40) has an analytical optimum. For the sake of simplicity, we introduce the following change of variables:

$$t = \frac{|\mu_{\hat{g}}(\boldsymbol{x})|}{\sigma_n}. \tag{41}$$

The argument $t \geqslant 0$ that maximizes $U_N(t)$ may be obtained by setting its derivative to zero and finding the critical points. The derivative of $U_N(t)$ with respect to t reads:

$$\frac{\mathrm{d}\,U_N(t)}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\Phi(-t) - \Phi\left(-\sqrt{2}t\right)\right). \tag{42}$$

Noting that $\Phi(t) = \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-x^2}{2}\right) \mathrm{d}x$ and by differentiating under the integral sign, the derivative of $U_N(t)$ vanishes for t such that:

$$\exp\left(-\frac{t^2}{2}\right) + \left(-\sqrt{2}\right) \exp\left(-t^2\right) = 0 \tag{43}$$

Solving Eq. (43) yields the positive solution $t = \sqrt{\ln 2}$, which means that the U_{N} learning function reaches its global maxima when:

$$|\mu_{\hat{g}}(\boldsymbol{x})| = \sigma_n \cdot \sqrt{\ln 2}. \tag{44}$$

From this result, we can conclude that if $\sigma_{\hat{g}}^2(\boldsymbol{x}) \to \sigma_n^2$, the learning function will select the points of the candidate set $\mathcal{X}_{\mathrm{c}}$ that best satisfy the condition shown in Eq. (44). Such points are however not close to the limit-state surface when σ_n is large.

Acknowledgments

References

Abbiati, G., S. Marelli, C. Ligeikis, R. Christenson, and B. Stojadinović (2022). Training of a classifier for structural component failure based on hybrid simulation and Kriging. *Journal of Engineering Mechanics 148*(1).

Abdallah, I., C. Lataniotis, and B. Sudret (2019). Parametric hierarchical Kriging for multi-fidelity aero-servo-elastic simulators — application to extreme loads on wind turbines. *Probabilistic Engineering Mechanics 55*, 67–77.

Ahmadisoleymani, S. S. and S. Missoum (2021). Stochastic crashworthiness optimization accounting for simulation noise. *Journal of Mechanical Design 144*(5), 1–15.

Ankenman, B., B. Nelson, and J. Staum (2010). Stochastic Kriging for simulation metamodeling. *Operations Research 58*(2), 371–382.

Au, S. K. and J. L. Beck (2001). Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Engineering Mechanics 16*(4), 263–277.

Azzi, S., Y. Huang, B. Sudret, and J. Wiart (2019). Surrogate modeling of stochastic functions – application to computational electromagnetic dosimetry. *International Journal for Uncertainty Quantification 9*(4), 351–363.

Bect, J., D. Ginsbourger, L. Li, V. Picheny, and E. Vazquez (2012). Sequential design of computer experiments for the estimation of a probability of failure. *Statistics and Computing 22*(3), 773–793.

Bichon, B. J., M. S. Eldred, L. Swiler, S. Mahadevan, and J. McFarland (2008). Efficient global reliability analysis for nonlinear implicit performance functions. *AIAA Journal 46*(10), 2459–2468.

Biermann, D., K. Weinert, and T. Wagner (2008). Model-based optimization revisited: Towards real-world processes. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, June 1-6, 2008, Hong Kong, China.

Blatman, G. and B. Sudret (2010). An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis. *Probabilistic Engineering Mechanics 25*, 183–197.

Blatman, G. and B. Sudret (2011). Adaptive sparse polynomial chaos expansion based on Least Angle Regression. *Journal of Computational Physics 230*, 2345–2367.

Chevalier, C. (2013). *Fast uncertainty reduction strategies relying on Gaussian process models*. Ph. D. thesis, University of Bern.

Chevalier, C., J. Bect, D. Ginsbourger, E. Vazquez, V. Picheny, and Y. Richet (2014). Fast parallel Kriging-based stepwise uncertainty reduction with application to the identification of an excursion set. *Technometrics 56*(4), 455–465.

Chinesta, F., E. Cueto, E. Abisset-Chavanne, J. L. Duval, and F. E. Khaldi (2018). Virtual, digital and hybrid twins: A new paradigm in data-based engineering and engineered data. *Archives of Computational Methods in Engineering 27*(1), 105–134.

Dimitrov, N., A. Natarajan, and M. Kelly (2014). Model of wind shear conditional on turbulence and its impact on wind turbine loads. *Wind Energy 18*(11), 1917–1931.

Ditlevsen, O. and H. Madsen (1996). *Structural reliability methods.* J. Wiley and Sons, Chichester.

Dubourg, V. (2011). *Adaptive surrogate models for reliability analysis and reliability-based design optimization.* Ph. D. thesis, Blaise Pascal University.

Dubourg, V., B. Sudret, and F. Deheeger (2013). Metamodel-based importance sampling for structural reliability analysis. *Probabilistic Engineering Mechanics 33*, 47–57.

Duddeck, F. (2007). Multidisciplinary optimization of car bodies. *Structural and Multidisciplinary Optimization 35*(4), 375–389.

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.

Ficini, S., U. Iemma, R. Pellegrini, A. Serani, and M. Diez (2021). Assessing the performance of an adaptive multi-fidelity Gaussian process with noisy training data: A statistical analysis. In *AIAA AVIATION 2021 FORUM*, August 2-6, 2021, virtual event.

Forrester, A., A. Keane, and N. Bressloff (2006). Design and analysis of "noisy" computer experiments. *American Institute of Aeronautics and Astronautics 44*, 2331–2339.

Giunta, A. A., J. M. Dudley, R. Narducci, B. Grossman, R. T. Haftka, W. H. Mason, and L. T. Watson (1994). Noisy aerodynamic response and smooth approximations in HSCT design. In *5th Symposium on Multidisciplinary Analysis and Optimization*, September, 07-09, Panama City Beach, USA.

Guo, J., Y. Xiang, K. Fujita, and I. Takewaki (2020). Vision-based building seismic displacement measurement by stratification of projective rectification using lines. *Sensors 20*(20), 5775.

Hasofer, A.-M. and N.-C. Lind (1974). Exact and invariant second moment code format. *Journal of Engineering Mechanics (ASCE) 100*(1), 111–121.

Jalali, H., I. V. Nieuwenhuyse, and V. Picheny (2017). Comparison of Kriging-based algorithms for simulation optimization with heterogeneous noise. *European Journal of Operational Research 261*(1), 279–301.

Jones, D. R., M. Schonlau, and W. J. Welch (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization 13*(4), 455–492.

Lataniotis, C., S. Marelli, and B. Sudret (2022). UQLab user manual – Kriging. Technical report, Chair of Risk, Safety & Uncertainty Quantification, ETH Zurich. Report # UQLab-V2.0-105.

Lee, J. J. and M. Shinozuka (2006). A vision-based system for remote sensing of bridge

displacement. *NDT & E International 39*(5), 425–431.

Lelièvre, N., P. Beaurepaire, C. Mattrand, and N. Gayton (2018). AK-MCSi: A Kriging-based method to deal with small failure probabilities and time-consuming models. *Structural Safety 73*, 1–11.

Lemaire, M. (2009). *Structural reliability.* Wiley.

Liu, P.-L. and A. Der Kiureghian (1991). Optimization algorithms for structural reliability. *Structural Safety 9*, 161–177.

Lüthen, N., S. Marelli, and B. Sudret (2022). Automatic selection of basis-adaptive sparse polynomial chaos expansions for engineering applications. *International Journal for Uncertainty Quantification 12*(3), 49–74.

Lüthen, N., S. Marelli, and B. Sudret (2023). A spectral surrogate model for stochastic simulators computed from trajectory samples. *Computer Methods in Applied Mechanics and Engineering 406*(115875), 1–29.

Lv, Z., Z. Lu, and P. Wang (2015). A new learning function for Kriging and its applications to solve reliability problems in engineering. *Computers & Mathematics with Applications 70*(5), 1182–1197.

Marelli, S. and B. Sudret (2014). 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)*, pp. 2554–2563.

Marelli, S. and B. Sudret (2018). An active-learning algorithm that combines sparse polynomial chaos expansions and bootstrap for structural reliability analysis. *Structural Safety 75*, 67–74.

McKay, M. D., R. J. Beckman, and W. J. Conover (1979, May). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics 21*(2), 239.

Melchers, R. and A. Beck (2018). *Structural reliability analysis and prediction.* John Wiley & Sons.

Melchers, R. E. (1989). Importance sampling in structural systems. *Structural Safety 6*, 3–10.

Moustapha, M., S. Marelli, and B. Sudret (2022). Active learning for structural reliability: Survey, general framework and benchmark. *Structural Safety 96*, 102714.

Moutoussamy, V., S. Nanty, and B. Pauwels (2015). Emulators for stochastic simulation codes. *ESAIM: Proceedings and Surveys 48*, 116–155.

Narducci, R., B. Grossman, M. Valorani, A. Dadone, and R. Haftka (1995). Optimization methods for non-smooth or noisy objective functions in fluid design problems. In *12th Computational Fluid Dynamics Conference*, June 19-22, 1995, San Diego, USA. American Institute of Aeronautics and Astronautics.

Nickitopoulou, A., K. Protopsalti, and S. Stiros (2006). Monitoring dynamic and quasi-static deformations of large flexible engineering structures with GPS: Accuracy, limitations and promises. *Engineering Structures 28*(10), 1471–1482.

Papila, M. and R. T. Haftka (2000). Response surface approximations: Noise, error repair, and modeling errors. *American Institute of Aeronautics and Astronautics 38*(12), 2336–2343.

Paz, J., J. Díaz, and L. Romera (2020). Analytical and numerical crashworthiness uncertainty quantification of metallic thin-walled energy absorbers. *Thin-Walled Structures 157*, 107022.

Pellegrini, R., J. Wackers, R. Broglia, A. Serani, M. Visonneau, and M. Diez (2022). A multi-fidelity active learning method for global design optimization problems with noisy evaluations. *Engineering with Computers*, 1–24.

Picheny, V., T. Wagner, and D. Ginsbourger (2013). A benchmark of Kriging-based infill criteria for noisy optimization. *Structural and Multidisciplinary Optimization 48*(3), 607–626.

Rackwitz, R. and B. Fiessler (1978). Structural reliability under combined load sequences. *Computers & Structures 9*, 489–494.

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

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

Sacks, J., W. J. Welch, T. J. Mitchell, and H. P. Wynn (1989). Design and analysis of computer experiments. *Statistical Science 4*, 409–435.

Santner, T. J., B. J. Williams, and W. I. Notz (2003). *The Design and Analysis of Computer Experiments.* Springer, New York.

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).

Shahriari, B., K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas (2016). Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE 104*(1), 148–175.

Shi, Y., Z. Lu, R. He, Y. Zhou, and S. Chen (2020). A novel learning function based on Kriging for reliability analysis. *Reliability Engineering & System Safety 198*, 106857.

Smola, A. J. and B. Schölkopf (2004). A tutorial on support vector regression. *Statistics and Computing 14*, 199–222.

Sofi, A., J. J. Regita, B. Rane, and H. H. Lau (2022). Structural health monitoring using wireless smart sensor network – an overview. *Mechanical Systems and Signal Processing 163*, 108113.

Teixeira, R., M. Nogal, and A. O'Connor (2021). Adaptive approaches in metamodel-based reliability analysis: A review. *Structural Safety 89*, 102019.

Torossian, L., V. Picheny, R. Faivre, and A. Garivier (2020). A review on quantile regression for stochastic computer experiments. *Reliability Engineering & System Safety 201*, 106858.

Torre, E., S. Marelli, P. Embrechts, and B. Sudret (2019). Data-driven polynomial chaos expansion for machine learning regression. *Journal of Computational Physics 388*, 601–623.

van den Eijnden, A. P., T. Schweckendiek, and M. A. Hicks (2021). Metamodelling for geotechnical reliability analysis with noisy and incomplete models. *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards 16*(3), 518–535.

Vazquez, E. and J. Bect (2009). A sequential Bayesian algorithm to estimate a probability of failure. In *15th IFAC Symposium on System Identification*, July 6-8, 2009, Saint-Malo, France.

Waarts, P.-H. (2000). *Structural reliability using finite element methods: an appraisal of DARS: Directional Adaptive Response Surface Sampling.* Ph. D. thesis, Technical University of Delft, Delft, The Netherlands.

Wackers, J., M. Visonneau, S. Ficini, R. Pellegrini, A. Serani, and M. Diez (2020). Adaptive N-fidelity metamodels for noisy CFD data. In *AIAA AVIATION 2020 FORUM*, June 15-19, 2020, virtual event.

Yi, J., Q. Zhou, Y. Cheng, and J. Liu (2020). Efficient adaptive Kriging-based reliability analysis combining new learning function and error-based stopping criterion. *Structural and Multidisciplinary Optimization 62*(5), 2517–2536.

Zaki, M. J. and W. J. Meira (2014). *Data Mining and Analysis: fundamental Concepts and Algorithms.* Cambridge University Press.

Zhan, D. and H. Xing (2020). Expected improvement for expensive optimization: a review. *Journal of Global Optimization 78*(3), 507–544.

Zhang, X., L. Wang, and J. D. Sørensen (2019). REIF: A novel active-learning function toward adaptive Kriging surrogate models for structural reliability analysis. *Reliability Engineering & System Safety 185*, 440–454.

Zhu, P., W. Pan, W. Chen, and F. Viana (2013). Lightweight design of vehicle parameters under crashworthiness using conservative surrogates. *Computers in Industry 64*(3), 280–289.

Zhu, X., M. Broccardo, and B. Sudret (2023). Seismic fragility analysis using stochastic polynomial chaos expansions. *Probabilistic Engineering Mechanics 72*, 103413.

Zhu, X. and B. Sudret (2020). Replication-based emulation of the response distribution of stochastic simulators using generalized lambda distributions. *International Journal for Uncertainty Quantification 10*(3), 249–275.

Zhu, X. and B. Sudret (2021a). Emulation of stochastic simulators using generalized lambda models. *SIAM/ASA Journal on Uncertainty Quantification 9*(4), 1345–1380.

Zhu, X. and B. Sudret (2021b). Global sensitivity analysis for stochastic simulators based on

generalized lambda surrogate models. *Reliability Engineering & System Safety 214*, 107815.

Zhu, X. and B. Sudret (2023). Stochastic polynomial chaos expansions to emulate stochastic simulators. *International Journal for Uncertainty Quantification 13*(2), 31–52.