Reliability analysis of arbitrary systems based on active learning and global sensitivity analysis

Maliki Moustapha, Pietro Parisi, Stefano Marelli, Bruno Sudret

ETH zürich

Reliability analysis of arbitrary systems based on active learning and global sensitivity analysis

M. Moustapha, P. Parisi, 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:	Reliability Engineering & System Safety
Report Ref.:	RSUQ-2023-002B
Arxiv Ref.:	https://arxiv.org/abs/2305.19885 [stat.ME] [stat.CO] [stat.AP]
DOI:	10.1016/j.ress.2024.110150
Date submitted:	April 17, 2023
Date accepted:	April 28, 2024

Reliability analysis of arbitrary systems based on active learning and global sensitivity analysis

M. Moustapha[1], P. Parisi[1], S. Marelli[1], and B. Sudret[1]

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

ETH Zurich, Stefano-Franscini-Platz 5, 8093 Zurich, Switzerland

Abstract

System reliability analysis aims at computing the probability of failure of an engineering system given a set of uncertain inputs and limit state functions. Active-learning solution schemes have been shown to be a viable tool but as of yet they are not as efficient as in the context of component reliability analysis. This is due to some peculiarities of system problems, such as the presence of multiple failure modes and their uneven contribution to failure, or the dependence on the system configuration (e.g., series or parallel). In this work, we propose a novel active learning strategy designed for solving general system reliability problems. This algorithm combines subset simulation and Kriging/PC-Kriging, and relies on an enrichment scheme tailored to specifically address the weaknesses of this class of methods. More specifically, it relies on three components: (i) a new learning function that does not require the specification of the system configuration, (ii) a density-based clustering technique that allows one to automatically detect the different failure modes, and (iii) sensitivity analysis to estimate the contribution of each limit state to system failure so as to select only the most relevant ones for enrichment. The proposed method is validated on two analytical examples and compared against results gathered in the literature. Finally, a complex engineering problem related to power transmission is solved, thereby showcasing the efficiency of the proposed method in a real-case scenario.

Keywords: System reliability – Active learning – Surrogate modelling – Global sensitivity analysis – Gaussian process modelling

1 Introduction

Engineering systems generally operate in an uncertain environment. Quantifying the impact of these uncertainties is paramount to designing safe structures. Reliability analysis aims

precisely at predicting the probability of failure of a structure, in which failure is characterized as the violation of some pre-defined limit state function. The latter maps the set of uncertain parameters describing the structure (*e.g.*, material properties) and its environment (*e.g.*, loads) to a scalar value whose sign encodes the state of the system, i.e., whether it is in failure or safety state.

Computing the failure probability is ultimately equivalent to solving a multi-dimensional integration problem, which is generally not tractable analytically. It is instead solved using approximation techniques such as the first- and second-order reliability methods or Monte Carlo simulation (Ditlevsen and Madsen, 1996; Lemaire, 2009; Melchers and Beck, 2018). While approximation techniques are relatively inexpensive, they lack accuracy in cases that involve high-dimensional or non-linear limit states. In contrast, simulation methods can yield accurate results and provide confidence bounds on their estimators, but are computationally intensive, and hence not suitable when the limit state function relies on expensive computational models. In the last decade, they have been complemented by surrogate models, which act as inexpensive approximations of the limit state function. Used in the so-called *active learning* framework, surrogate models have been shown to efficiently solve reliability analysis problems (Teixeira et al., 2021; Moustapha et al., 2022). Examples include active-Kriging Monte Carlo simulation (AK-MCS, Echard et al. (2011)) or efficient global reliability analysis (EGRA, Bichon et al. (2009)), which both use Kriging a.k.a. Gaussian process model (Santner et al., 2003; Rasmussen and Williams, 2006) as a surrogate.

In active learning, a surrogate model is trained using a limited set of evaluations of the original model, known as the *experimental design*. The goal is to ensure its accuracy in the vicinity of the limit state surface (i.e., the boundary between the failure and safety domains) in regions of high probability density. To this end, a *learning function* is used to adaptively and parsimoniously select sets of input parameters that, when added to the experimental design, are most likely to increase the accuracy of the limit state surface approximation and, henceforth, of the estimated probability of failure.

This is an active field of research as shown by the recent profusion of methods in the literature (for a comprehensive review and benchmark analysis, see Teixeira et al. (2021) and Moustapha et al. (2022), respectively). These methods are mainly developed for so-called *component* reliability problems, where failure is described by a single limit state function. Yet in complex engineering systems, multiple limit state functions may be required to properly capture different failure modes. This problem is known as *system* reliability analysis. In contrast to component reliability, there are fewer contributions in the literature, most of which do not address the specific issues related to system reliability, such as the presence of multiple disjoint failure domains or their uneven contribution to system failure.

The most straightforward way to tackle system reliability problems using active learning is to directly approximate the system performance function, which is a combination of the component limit states. By doing so, one solves again a component reliability problem while

reducing the number of models to approximate and evaluate to a single one. However, by construction, the system limit state is a non-smooth function that may exhibit discontinuities or high irregularities, and is thereby difficult to approximate accurately (Tresp, 2000; Meeds and Osindero, 2005; Boroson and Missoum, 2017; Moustapha and Sudret, 2023). This may happen for instance when the system limit state is obtained as the minimum of all the component ones (in case of so-called *parallel systems* limit states).

It is then more convenient and efficient to separately approximate each of the components. This is even more relevant when latter rely on fundamentally different computational models (e.g., in multiphysics analyses). Many approaches proposed in the literature do so by simply adapting component reliability learning functions to solve system problems. For instance, AK-SYS (Fauriat and Gayton, 2014) is an extension of AK-MCS where different ways of computing the learning function (deviation number U) are proposed. Guo et al. (2021) propose an improvement of AK-SYS by considering kernel density based adaptive importance sampling instead of MCS. AK-SYSi (Yun et al., 2019) is a direct improvement of AK-SYS where the issue of inaccurately estimating the modes of the different Kriging models is addressed. More recently, new learning functions have been proposed to address the system limit-state while using separate surrogates at the component level. Examples include the expected system improvement suggested by Yang et al. (2022), a combination of the expected improvement and the U learning function (Wang and Chen, 2023) or the APe (approximation of the probability of error), which is based on the probability of misclassification (Xu et al., 2023). Similarly, Feng et al. (2023) proposed a learning function based on the upper bound of the probability of misclassification, taking into account the correlation between different failure modes.

Arguing that such methods fail to properly discard unimportant limit state functions when their responses differ in magnitude, Yang et al. (2018) propose to define truncated candidate regions (TCR) where unimportant areas are first identified. Only samples not belonging to the TCR are selected as candidate points. The traditional composite criterion of AK-SYS is then used to select the enrichment points and the corresponding limit state function to evaluate. Yang et al. (2020) propose an extension of ALK-TCR considering multi-modal adaptive importance sampling. Zhou et al. (2022) consider ALK-TCR but break down the enrichment scheme in two different strategies according to the system configuration (series or parallel).

From another perspective, Wu et al. (2020) extend the concept of dependent Kriging models where the correlation of the component responses between different candidate points is taken into account (Hu and Mahadevan, 2016) while devising a new learning function for system problems (DKM-SYS). Wang et al. (2021) similarly propose a method relying on the misclassification at the system level. Accounting for the dependencies between different responses at the same location (assuming a single model run returns multiple outputs), Sadoughi et al. (2018) use multivariate Gaussian process models, while Hu and Ludkovski

(2017) propose building composite Kriging models combining both the individual models and others based on an SVD decomposition of the system responses. Other contributions involving multivariate Gaussian processes include Perrin (2016) where nested GP models are considered or Wei and Liu (2018).

All these methods are only applicable to systems that are either parallel or series. In practice, it is always possible to use cut-sets to transform any arbitrary configuration into a series or parallel system. Nevertheless, the reconfigured cut-sets still result in non-smooth, and henceforth difficult to approximate, limit states. More general methods capable of handling complex system configurations have been developed recently. For instance, Yuan et al. (2020) propose an approach where the learning function is built on a system level using system information defined through minimal path sets. The latter is also used by Xiao et al. (2022) to extend DKM-SYS to arbitrary system configurations.

The aforementioned contributions mostly tackle only one or two aspects of the issues raised by system reliability. In this paper, we propose a new general and efficient algorithm for surrogate-based system reliability analysis that address them *all at once*. More precisely, we devise an enrichment scheme that identifies relevant failure modes through subset simulation and density-based clustering. Furthermore, we use Sobol' sensitivity analysis to identify the limit state functions that contribute the most to the system failure, so as to only enrich the corresponding experimental designs. The approach can be used for general systems, *i.e.*, it is not limited to series of parallel ones.

The paper is organized as follows. In Section 2, we state the problem, introduce useful notations and discuss the motivations for this work. In Section 3, we present the proposed method and detail all its components. Finally in Section 4, we validate the proposed method using two benchmark problems and apply it to an engineering problem which consists of a system of transmission towers.

2 Problem statement and motivations

2.1 Problem statement

Let us consider a set of m quantities of interest Y_j describing the performance of a system through a set of *limit state functions* g_j such that

$$Y_j = g_j\left(\boldsymbol{X}_j\right), \quad j = 1, \ldots, m, \tag{1}$$

where $\boldsymbol{X}_j \in \mathcal{D}_{\boldsymbol{X}_j} \subset \mathbb{R}^{M_j}$ is an M_j-dimensional random vector associated to the j-th *component* of the system. These random variables characterize the uncertainties associated to the components of interest, such as the stochastic variability of the loads or manufacturing tolerances. In component reliability analysis, we are interested in finding the probability of

failure of each component, which can be computed as

$$P_{f_j} = \int_{\mathcal{D}_{f_j}} f_{\boldsymbol{X}_j}(\boldsymbol{x}_j)\, \mathrm{d}\boldsymbol{x}_j, \tag{2}$$

where $f_{\boldsymbol{X}_j}$ is the joint density of the random variables gathered in $\boldsymbol{X}_j$ and $\mathcal{D}_{f_j}$ denotes the failure domain of the j-th component. The latter is by convention implicitly defined through the sign of the corresponding limit state function, *i.e.*, $\mathcal{D}_{f_j} = \left\{\boldsymbol{x}_j \in \mathbb{R}^{M_j} : g_j(\boldsymbol{x}_j) \leq 0\right\}$.

In this work, we are primarily interested in *system reliability analysis*, where the performance of the system is jointly described by the component limit states. In this context, the definition of the failure domain is not straightforward and depends on the configuration of the system. Notable configurations include parallel and series systems. A parallel system is one for which system failure occurs only when all components fail, while a series system is one that fails when at least one component fails. Assuming that the system state can be sufficiently described by the knowledge of the state of each component, any arbitrary configuration can be defined as a parallel (resp. series) arrangement of a series (resp. parallel) system using the concepts of minimal path sets (resp. minimal cut sets) (Ross, 2010).

We consider here the most general case when the system configuration is defined through an arbitrary function $h(g_1(\boldsymbol{X}_1), \ldots, g_m(\boldsymbol{X}_m)) = h(\boldsymbol{g}(\boldsymbol{X}))$. We introduce here the M-dimensional random vector describing the entire system $\boldsymbol{X} \in \mathbb{R}^M \sim f_{\boldsymbol{X}}(\boldsymbol{x})$ that contains all components of the vectors $\boldsymbol{X}_j$ without duplication. The support of $\boldsymbol{X}$ is obtained by the union: $\mathcal{D}_{\boldsymbol{X}} = \cup_{j=1}^{m} \mathcal{D}_{\boldsymbol{X}_j}$. This means that the random variables $\boldsymbol{X}_j$ are not necessarily the same, even though there is in general some overlap. While a Boolean composition of min and max operators may typically define the *composition function* h, this work is not restricted to such compositions. We however assume that h is an easy-to-evaluate, analytical function, more precisely that its evaluation cost is comparatively much smaller than that of any of the component limit state functions. This allows us to use sampling-based approaches in the methodology we propose

The system failure probability therefore reads

$$P_f = \mathbb{P}\left[h(\boldsymbol{g}(\boldsymbol{X})) \leq 0\right] = \int_{\mathcal{D}_f} f_{\boldsymbol{X}}(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}, \tag{3}$$

where $f_{\boldsymbol{X}}$ is the joint density of $\boldsymbol{X}$ and the system failure domain $\mathcal{D}_f$ is defined by $\mathcal{D}_f = \left\{\boldsymbol{x} \in \mathbb{R}^M : h(\boldsymbol{g}(\boldsymbol{x})) \leq 0\right\}$.

2.2 Motivation

Active learning reliability methods are arguably the most efficient way to solve the integration problems in Eqs. (2) and (3). Such methods have been thoroughly explored in the context of component reliability, as documented in recent surveys and benchmarks (Teixeira et al., 2021; Moustapha et al., 2022). These methods have also been extended or adapted to system problems. Yet the proposed extensions are generally poor in many aspects: (i) some attempt to approximate $h(\boldsymbol{g}(\boldsymbol{x}))$, which is generally a non-homogeneous or discontinuous

function, (ii) others lack a consistent way of selecting the most appropriate component limit state function to enrich, resulting in enrichment points located in areas with little to no contribution to system failure, and (iii) most methods are specifically tailored to a particular system configuration (usually, series or parallel).

In this paper, we propose a methodology that aims at bypassing all these limitations. More precisely, the proposed method allows solving complex system reliability problems and its implementation does not depend on the system configuration or any knowledge thereof. Furthermore, the methodology allows for a large computational cost saving by enriching only the experimental designs corresponding to relevant limit states, *i.e.*, areas that contribute to component but not to system failure are not enriched. Finally, we aim at solving problems with relatively small failure probabilities.

More precisely, we tackle the three limitations respectively by (i) building a separate surrogate model for each component, (ii) using Sobol' sensitivity analysis on $h(\widehat{\boldsymbol{g}}(\boldsymbol{x}))$ to identify the specific limit state function(s) that need to be updated, hence limiting enrichment to areas effectively contributing to system failure, and (iii) introducing a new system learning function for arbitrary system configurations. Furthermore, to estimate small failure probabilities, we use subset simulation (Au and Beck, 2001), which also provides us with the candidate samples for enrichment (following the recommendation of the comprehensive benchmark by Moustapha et al. (2022)).

3 Proposed method for system reliability

3.1 Flowchart of the proposed method

The proposed method draws upon the traditional active learning reliability framework for component problems, which comprises four ingredients, namely a surrogate model, a reliability estimation algorithm, an enrichment scheme and a stopping criterion (Moustapha et al., 2022). These ingredients are generally assembled in a sequential way in a framework popularized by AK-MCS (Echard et al., 2011).

The same framework is considered here for system reliability, as illustrated in the flowchart in Figure 1. Additional steps are however taken to address the peculiarities of system reliability. They are highlighted in gray in the flowchart.

The algorithm is first summarized in the following steps:

1. **Initialization:** The initial experimental design is built by sampling the input space using a space-filling technique such as Latin hypercube sampling or Sobol' sequences (McKay et al., 1979; Sobol', 1967). To cover as uniformly as possible the input space and reach areas far in the tails of the distribution (which are likely to be in the failure domain), we sample in a hypercube $\prod_{i=1}^{M} \left[x_i^-,\, x_i^+\right]$. There are many ways to define these bounds. For low-dimensional cases, we simply consider $x_i^- = \mu_{x_i} - 5\,\sigma_{X_i}$ and

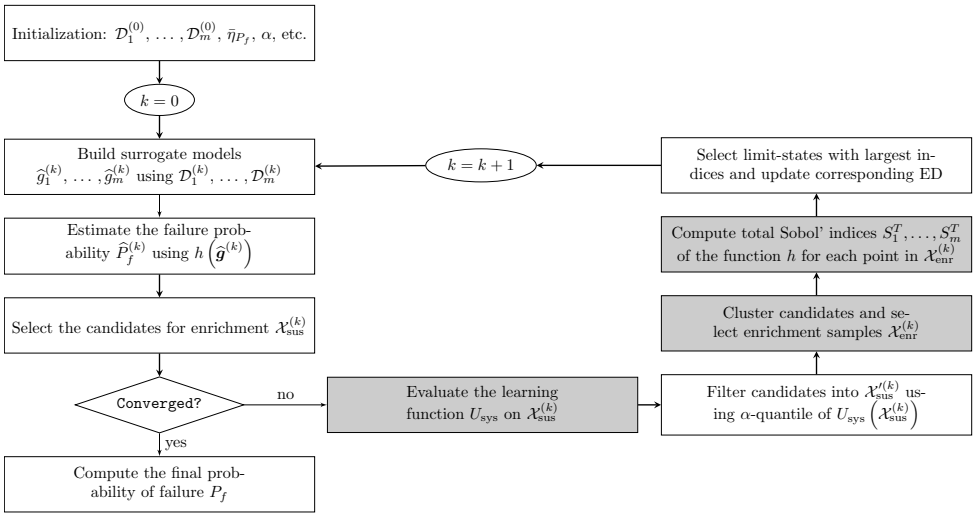


Figure 1: Flowchart of the proposed active learning method for system reliability.

$x_i^+ = \mu_{X_i} + 5\,\sigma_{X_i}$, where μ_{X_i} and σ_{X_i} are respectively the mean and standard deviation of the i-th random variable. For high-dimensional cases, we consider upper and lower quantiles of the distribution in each dimension. Once the inputs are sampled, the experimental design is completed by evaluating the corresponding limit state functions on the generated samples. For each limit state j, the experimental design is denoted by $\mathcal{D}_j = \{\mathcal{X}_j, \mathcal{G}_j\} = \left\{\left(\boldsymbol{x}_j^{(i)}, g_j\left(\boldsymbol{x}_j^{(i)}\right)\right), i = 1, \ldots N_j^{(0)}\right\}$, for $j = 1, \ldots, m$.

2. **Surrogates construction:** m independent surrogate models $\widehat{g}_j$ are then built using the respective experimental designs $\mathcal{D}_j$, $j = \{1, \ldots, m\}$. We consider here surrogates with built-in error measures, namely Kriging and polynomial-chaos Kriging, which are further described in Section 3.2.

3. **System reliability:** Using the approximate system limit state function $h\left(\widehat{g}\left(\boldsymbol{x}\right)\right) = h\left(\widehat{g}_1\left(\boldsymbol{x}_1\right), \ldots, \widehat{g}_m\left(\boldsymbol{x}_m\right)\right)$, the probability of failure of the system is estimated. We consider subset simulation for this task. The samples generated throughout the algorithm define the search space for the enrichment and are denoted by $\mathcal{X}_{\text{sus}}^{(k)}$.

4. **Convergence check:** The convergence of the algorithm is then assessed. In this work, we consider convergence to be achieved when the relative variation of the reliability index within two consecutive iterations,

$$\varepsilon^{(k)} = \frac{\left|\beta^{(k)} - \beta^{(k-1)}\right|}{\beta^{(k-1)}}, \tag{4}$$

is below a given threshold $\bar{\varepsilon}$ *three times in a row*, where $\beta^{(k)} = -\Phi^{-1}\left(P_f^{(k)}\right)$. Here Φ^{-1} is the inverse standard Gaussian cumulative distribution function and ε is a threshold set to 0.005. If this condition is not respected, the algorithm proceeds with the next step, otherwise Step 10 is performed. Let us note here that common random numbers (Spall, 2003) is enabled within different iterations, making the difference within two consecutive iterations mainly due to the actual change in limit-state surface approx-

imation. It is also worth mentioning that this stopping criterion does not guarantee convergence to the underlying true failure probability. However, it has proven to be robust enough for practical purpose and its settings are defined following a large benchmark in Moustapha et al. (2022).

5. **Learning function:** The learning function U_{sys} is then evaluated to find out which points are most likely to improve the accuracy of the system limit state surface, and hence of the failure probability estimate, once added to the experimental design. To this effect, we propose a new system-adapted learning function that is described in Section 3.3.

6. **Candidates filtering:** To select the next samples to evaluate, the search space samples $\mathcal{X}_{\text{sus}}^{(k)}$ are first filtered into a subset of candidates for enrichment denoted by $\mathcal{X}_{\text{sus}}^{\prime(k)}$. This filter simply consists in keeping the most promising samples which correspond to points with the smallest value of U_{sys} as described in Section 3.3. This step allows to avoid possibly costly operations on the whole search space $\mathcal{X}_{\text{sus}}^{(k)}$.

7. **Samples selection:** The filtered candidates are then clustered using the clustering algorithm *DBSCAN* (density-based spatial clustering with applications to noise), which has the advantage of not requiring a pre-assigned number of clusters. The latter is instead directly inferred from the data. In each identified cluster, the sample with the smallest U_{sys} value is selected for enrichment. The set of selected enrichment samples is denoted by $\mathcal{X}_{\text{enr}}^{(k)}$.

8. **Sensitivity analysis:** Sensitivity analysis of the composition function $h\left(z_1, \ldots, z_m\right)$ is carried out to select the limit states $j \in \{1, \ldots, m\}$ to evaluate for each enrichment sample. This crucial step, which allows us to only update the limit states that are the most relevant to system failure, is further detailed in Section 3.3.

9. **Experimental design and surrogate updates:** The experimental designs and surrogates which correspond to the limit states that have been selected in the previous step are eventually updated. The algorithm then returns to Step 2.

10. **Final estimate:** Once the convergence of the algorithm is reached, a final estimate of the probability of failure is performed using another set-up of subset simulation that allows us to increase the precision of the estimate. It is worth stressing that an overkill set-up is also used within the enrichment iterations ($5 \cdot 10^4$ samples per subset level, intermediate probabilities set to $p_0 = 0.25$). Ultimately, we expect the space covered in both the on-line and off-line strategies to be the same.

Each of these steps has been designed so as to enhance the efficiency of the algorithm. We now present in details the most important ingredients.

3.2 Surrogate models

Surrogate models are an important part of the proposed method as they allow us to reduce the computational burden of the algorithm by providing inexpensive proxies of the limit state functions. The latter are approximated separately using independent experimental designs. While the proposed framework does not rely on a specific type, two different surrogate models are considered in this work, namely Kriging and polynomial chaos-Kriging (PC-Kriging). Both are chosen because they feature built-in prediction errors that can be used in the enrichment scheme.

3.2.1 Kriging

Kriging, also known as Gaussian process modeling, is a surrogate type that considers the underlying function to approximate as one realization of a Gaussian process. This is expressed in the form (Santner et al., 2003; Rasmussen and Williams, 2006)

$$g\left(\boldsymbol{x}\right) = \mu\left(\boldsymbol{x}\right) + \sigma^2 Z\left(\boldsymbol{x}, \omega\right), \tag{5}$$

where $\mu\left(\boldsymbol{x}\right)$ is the trend, σ^2 is the Kriging variance and $Z\left(\boldsymbol{x}, \omega\right)$ is a zero-mean unit-variance stationary Gaussian process. The trend can be expressed in many ways but is generally cast as a polynomial of the form $\boldsymbol{\beta}^T \boldsymbol{f}(\boldsymbol{x})$, where $\boldsymbol{f} = \{f_1, \ldots, f_p\}$ is a set of p basis functions and $\boldsymbol{\beta} = \{\beta_1, \ldots, \beta_p\}$ are the corresponding coefficients. The stochastic process $Z\left(\boldsymbol{x}, \omega\right)$, which captures the deviation from the trend, is fully characterized by its auto-correlation function $R\left(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta}\right)$, which is parameterized by $\boldsymbol{\theta}$.

In general, we are interested in making a prediction $\widehat{g}$ of the function g at some test location $\boldsymbol{x}$ using the observations $\mathcal{G}$ in the experimental design. By definition of a Gaussian random field, any subset of variables including the observations and the response at a given test location are jointly Gaussian, *i.e.*

$$\begin{Bmatrix} \widehat{g}(\boldsymbol{x}) \\ \mathcal{G} \end{Bmatrix} \sim \mathcal{N}_{N+1}\left(\begin{Bmatrix} \boldsymbol{f}(\boldsymbol{x})^T \boldsymbol{\beta} \\ \boldsymbol{F}\boldsymbol{\beta} \end{Bmatrix}, \quad \sigma^2 \begin{Bmatrix} 1 & \boldsymbol{r}^T(\boldsymbol{x}) \\ \boldsymbol{r}(\boldsymbol{x}) & \boldsymbol{R} \end{Bmatrix} \right), \tag{6}$$

where $\boldsymbol{F}$ and $\boldsymbol{R}$ are the $N \times N$ observation and auto-correlation matrices with elements respectively defined by $F_{ij} = f_j\left(\boldsymbol{x}^{(i)}\right)$ and $R_{ij} = R\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}; \boldsymbol{\theta}\right)$, $r\left(\boldsymbol{x}\right)$ is the vector of cross-correlations whose elements are $r_i = R\left(\boldsymbol{x}, \boldsymbol{x}^{(i)}; \boldsymbol{\theta}\right)$, $i = 1, \ldots N$ and N is the number of samples in the experimental design.

The prediction is obtained by conditioning the response at the test location with respect to the available observations. This also results in a Gaussian distribution

$$\widehat{g}(\boldsymbol{x}) | \mathcal{G} \sim \mathcal{N}\left(\widehat{\mu}_{\widehat{g}}(\boldsymbol{x}), \widehat{\sigma}^2_{\widehat{g}}(\boldsymbol{x})\right), \tag{7}$$

whose mean and variance are respectively given by

$$\mu_{\widehat{g}}\left(\boldsymbol{x}\right) = \boldsymbol{f}^T\left(\boldsymbol{x}\right)\widehat{\boldsymbol{\beta}} + \boldsymbol{r}^T\left(\boldsymbol{x}\right)\boldsymbol{R}^{-1}\big(\mathcal{G} - \boldsymbol{F}\widehat{\boldsymbol{\beta}}\big), \tag{8}$$

$$\sigma^2_{\widehat{g}}(\boldsymbol{x}) = \widehat{\sigma}^2 \left(1 - \boldsymbol{r}^T(\boldsymbol{x}) \boldsymbol{R}^{-1} \boldsymbol{r}(\boldsymbol{x}) + \boldsymbol{u}^T(\boldsymbol{x}) (\boldsymbol{F}^T \boldsymbol{R}^{-1} \boldsymbol{F})^{-1} \boldsymbol{u}(\boldsymbol{x})\right). \tag{9}$$

In these two equations, $\widehat{\boldsymbol{\beta}} = \left(\boldsymbol{F}^T \boldsymbol{R}^{-1} \boldsymbol{F}\right)^{-1} \boldsymbol{F}^T \boldsymbol{R}^{-1} \mathcal{G}$ is the least-square estimate of the regression coefficients, $\widehat{\sigma}^2 = \frac{1}{N} (\mathcal{G} - \boldsymbol{F}\boldsymbol{\beta})^T \boldsymbol{R}^{-1} (\mathcal{G} - \boldsymbol{F}\boldsymbol{\beta})$ is the maximum likelihood estimate of the process variance and $\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{F}^T \boldsymbol{R}^{-1} \boldsymbol{r}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x})$ has been introduced for convenience.

An important feature of Kriging is that besides prediction, a measure of its own accuracy is also readily provided through the variance in Eq. (9). It generally tends to get large further away from the training points and collapses to zero on them, thus making Kriging an interpolant. This feature is essential in active learning strategies.

3.2.2 Polynomial Chaos-Kriging

Polynomial Chaos-Kriging or PC-Kriging attempts to combine the global behavior of polynomial chaos expansions (PCE, Xiu and Karniadakis (2002)) to the local behavior of Kriging. It is built as a universal Kriging model where the trend is actually a PCE model (Schöbi et al., 2015, 2017):

$$g(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \subset \mathcal{A}} y_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) + \sigma^2 Z(\boldsymbol{x}, \omega), \tag{10}$$

where $\Psi_{\boldsymbol{\alpha}}(\boldsymbol{x})$ are multivariate polynomials orthonormal with respect to the input distribution $f_{\boldsymbol{X}}$, $\boldsymbol{\alpha} \in \mathcal{A} \subset \mathbb{N}^M$ are multi-indices identifying the components of the multivariate components and $y_{\boldsymbol{\alpha}}$ are corresponding regression coefficients.

The multivariate polynomials are assembled using a tensor product of their univariate counterparts, *i.e.*,

$$\Psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \prod_{i=1}^{M} \Psi_{\alpha_i}(x_i). \tag{11}$$

The optimal basis is obtained here using least-angle regression (LAR), which allows one to adaptively select basis functions that lead to a sparse representation (Efron et al., 2004; Blatman and Sudret, 2011). Once the basis is selected, the hyperparameters $\left\{y_{\boldsymbol{\alpha}}, \sigma^2, \boldsymbol{\theta}\right\}$ are estimated using the traditional Kriging calibration method described in the previous section (See Schöbi et al. (2015, 2022) for more details).

3.3 Enrichment scheme

To compute the probability of failure at each iteration of the active learning process, we consider subset simulation (Au and Beck, 2001). Compared to crude Monte Carlo, subset simulation allows us to more thoroughly explore the random variable space and generally provides us with finer reliability estimates at a lower cost.

In this proposed framework, reliability estimation is performed at each iteration using the approximated system limit state $h(\widehat{g}_1(\boldsymbol{x}), \ldots, \widehat{g}_m(\boldsymbol{x}))$. The samples generated by subset simulation, herein denoted by $\mathcal{X}_{\text{sus}}$, are the starting point of the enrichment process as they constitute the candidate pool from which samples to add to the experimental design are

selected. Three components are leveraged to optimize the necessary number of samples to convergence. They are now described in details.

Learning function

The learning function is an important component of active learning as it allows us to select the next point(s) to add to the experimental design. It generally consists of a simple function that is maximized (or minimized) to yield the point that is the most likely to improve the knowledge of the limit state surface, and henceforth the accuracy of the surrogate-based estimate of the failure probability.

There are many such learning functions in the literature and in this work we propose an adaptation of the well-known deviation number (Echard et al., 2011). Considering a Gaussian process approximation of the limit state function $\widehat{g}(\boldsymbol{x}) \sim \mathcal{N}\left(\mu_{\widehat{g}}(\boldsymbol{x}), \sigma^2_{\widehat{g}}(\boldsymbol{x})\right)$, the deviation number simply reads:

$$U(\boldsymbol{x}) = \frac{|\mu_{\widehat{g}}(\boldsymbol{x})|}{\sigma_{\widehat{g}}(\boldsymbol{x})}. \tag{12}$$

The next point to add to the experimental design is then the one that minimizes U. Minimal values of the learning function are obtained when the GP prediction is close to the 0 (points in the vicinity of the limit state surface) or when the prediction variance is large (in other words, points in areas where the prediction sign is most uncertain).

In this work, a *system-adapted deviation number* is considered using the system limit state surrogate given by

$$Z_{\text{sys}}(\boldsymbol{x}) \overset{\text{def}}{=} h(\boldsymbol{Z}) = h\left(\widehat{g}_1(\boldsymbol{x}_1), \ldots, \widehat{g}_m(\boldsymbol{x}_m)\right), \tag{13}$$

where $Z_j \overset{\text{def}}{=} \widehat{g}_j(\boldsymbol{x}_j), j = 1, \ldots, m$ represents the Kriging or PC-Kriging approximation of the j-th limit state function. Note that Z_{sys} is a random variable attached to each point $\boldsymbol{x}$, obtained by inserting m Gaussian variables $Z_1, \ldots, Z_m$ into the composition function h. Note that it is in general *not* Gaussian.

This proposed learning function is then defined by:

$$U_{\text{sys}}(\boldsymbol{x}) = \frac{|\mu_{\text{sys}}(\boldsymbol{x})|}{\sigma_{\text{sys}}(\boldsymbol{x})}, \tag{14}$$

where $\mu_{\text{sys}}(\boldsymbol{x})$ and $\sigma_{\text{sys}}(\boldsymbol{x})$ are respectively the mean and standard deviation of the system response $Z_{\text{sys}}(\boldsymbol{x})$. These two quantities are not readily available as the distribution of Z_{sys} is not generally known for a general system and related composition function h. It is actually not possible to derive them analytically, even when the (Gaussian) distributions of $\{\widehat{g}_j(\boldsymbol{x}_j), j = 1, \ldots, m\}$ are known and the analytical expression of h is given.

We therefore resort to an empirical estimation based on the Kriging/PC-Kriging approximations. In this context, the vector $\boldsymbol{Z}$ in Eq. (13) is an independent multivariate Gaussian since each of its components is obtained from an independent Kriging/PC-Kriging predictor, *i.e.* , $Z_j \sim \mathcal{N}\left(\mu_{\widehat{g}_j}(\boldsymbol{x}_j), \sigma^2_{\widehat{g}_j}(\boldsymbol{x}_j)\right)$ and

$$\boldsymbol{Z}(\boldsymbol{x}) \sim f_{\boldsymbol{Z}} = \prod_{j=1}^{m} \mathcal{N}\left(\mu_{\widehat{g}_j}(\boldsymbol{x}_j), \sigma^2_{\widehat{g}_j}(\boldsymbol{x}_j)\right). \tag{15}$$

The empirical estimate of U_{sys} can therefore be obtained by drawing n samples $\left\{\boldsymbol{z}^{(1)}, \dots \boldsymbol{z}^{(n)}\right\}$ following Eq. (15) and then estimating the mean and variance of the system response as follows

$$\begin{aligned} \widehat{\mu}_{\text{sys}} =& \frac{1}{n} \sum_{i=1}^{n} h\left(\boldsymbol{z}^{(i)}\right), \\ \widehat{\sigma}_{\text{sys}} =& \frac{1}{n-1} \sum_{i=1}^{n} \left(h\left(\boldsymbol{z}^{(i)}\right) - \widehat{\mu}_{\text{sys}}\right)^2, \end{aligned} \tag{16}$$

which are then plugged in Eq. (14) to get

$$\widehat{U}_{\text{sys}}\left(\boldsymbol{x}\right) = \frac{|\widehat{\mu}_{\text{sys}}|}{\widehat{\sigma}_{\text{sys}}}. \tag{17}$$

Similarly to the original U function, samples that minimize U_{sys} are of interest, since they correspond to regions where the system prediction is close to zero and/or where the uncertainty in the prediction is large.

Filtering and clustering

Once the learning function is computed for all the samples in the candidate pool $\mathcal{X}_{\text{sus}}$, the latter is filtered so as to keep only the most relevant samples *i.e.*, the ones such that $\widehat{U}_{\text{sys}}\left(\boldsymbol{x}\right)$ (which is non negative by construction) is close to zero. More specifically, the *reduced enrichment candidate set* is defined by

$$\mathcal{X}'_{\text{sus}} = \left\{\boldsymbol{x} \in \mathcal{X}_{\text{sus}} : U_{\text{sys}}\left(\boldsymbol{x}\right) \leq U_q\right\}, \tag{18}$$

where U_q is the α-th empirical quantile of the set $U_{\text{sys}}\left(\mathcal{X}_{\text{sus}}\right)$. The quantile level α is chosen in the range $[0.01 - 0.05]$ so as to only keep the most important samples.

Following this filtering step, the samples are clustered into C subsets. This allows us to add multiple points per iteration by possibly considering different failure modes. This also contributes to reducing information redundancy, since points that are eventually added are not too close to each other. Moreover, the number of clusters is not predefined but directly inferred from the data. In some cases, each cluster corresponds to a different failure mode of the system. There are many ways to detect such clusters and we consider here the so-called *density-based spatial clustering of applications with noise* a.k.a. DBSCAN, which is a hierarchical density-based clustering method proposed by Ester et al. (1996). The algorithm is described in details in Section A.

Assuming that C clusters $\{\mathcal{X}_c, c = 1, \dots, C\}$ have been identified, one enrichment point is defined for each of them as follows:

$$\widetilde{\boldsymbol{x}}^{(c)} = \arg\min_{\boldsymbol{x} \in \mathcal{X}_c} U_{\text{sys}}\left(\boldsymbol{x}\right), \; c = 1, \dots, C. \tag{19}$$

The resulting set of enrichment points is denoted in the sequel by $\mathcal{X}_{\text{enr}}$. Enriching all the experimental designs with these points is not optimal as each of them is rather connected to a specific failure mode. We therefore consider *sensitivity analysis* of the composition function h to identify for each sample which limit state is the most impacted.

Sensitivity analysis

Sensitivity analysis is used to identify, for each enrichment sample, which limit state contributes the most to system failure. In this work, we consider the popular Sobol' sensitivity indices (Sobol', 1993; Iooss et al., 2022).

Let us consider one sample $\widetilde{\boldsymbol{x}}^{(c)} \in \mathcal{X}_{\text{enr}}$. We can calculate the Sobol' sensitivity indices of the function $h\left(\widehat{g}\left(\widetilde{\boldsymbol{x}}^{(c)}\right)\right) = h\left(\boldsymbol{Z}\left(\widetilde{\boldsymbol{x}}^{(c)}\right)\right)$ with respect to the random variables $\left\{Z_1\left(\widetilde{\boldsymbol{x}}^{(c)}\right), \ldots, Z_m\left(\widetilde{\boldsymbol{x}}^{(c)}\right)\right\}$. These indices can be computed by Monte Carlo simulation since h is an inexpensive function and samples from the Gaussian random vector $\boldsymbol{Z}\left(\widetilde{\boldsymbol{x}}^{(c)}\right)$ can be easily obtained, given that they are independent multivariate Gaussian whose parameters are provided by the Kriging/PC-Kriging models, *i.e.* ,

$$Z_j\left(\widetilde{\boldsymbol{x}}^{(c)}\right) = \widehat{g}_j\left(\widetilde{\boldsymbol{x}}_j^{(c)}\right) \sim \mathcal{N}\left(\mu_{\widehat{g}_j}\left(\widetilde{\boldsymbol{x}}_j^{(c)}\right), \sigma^2_{\widehat{g}_j}\left(\widetilde{\boldsymbol{x}}_j^{(c)}\right)\right), \, j = 1, \ldots, m \tag{20}$$

The total Sobol' indices $\left\{S_1^T\left(\widetilde{\boldsymbol{x}}^{(c)}\right), \ldots, S_m^T\left(\widetilde{\boldsymbol{x}}^{(c)}\right)\right\}$ are chosen to rank the limit state in terms of relevance for each enrichment sample $\widetilde{\boldsymbol{x}}^{(c)}$. The limit state to enrich is therefore selected as the one with the corresponding largest total Sobol' index, *i.e.,*

$$j^*\left(\widetilde{\boldsymbol{x}}^{(c)}\right) = \arg\max_{j=1\ldots m} S_j^T\left(\widetilde{\boldsymbol{x}}^{(c)}\right). \tag{21}$$

In summary, for the c-th enrichment point, the experimental design $\mathcal{D}_{j^*}$ is updated such that $\mathcal{D}_{j^*} \leftarrow \mathcal{D}_{j^*} \cup \left(\widetilde{\boldsymbol{x}}_{j^*}^{(c)}, g\left(\widetilde{\boldsymbol{x}}_{j^*}^{(c)}\right)\right)$, where $\widetilde{\boldsymbol{x}}_{j^*}^{(c)}$ is the relevant subset of $\widetilde{\boldsymbol{x}}^{(c)}$ corresponding to the j^*-th limit state.

Summary of the enrichment scheme

The entire enrichment process at a given iteration is summarized in Algorithm 1. The superscript k, which refers to the iteration number of the active learning process is omitted to simplify the notation.

4 Application examples

We now consider three different problems to illustrate and validate the proposed approach. The first two are based on analytical functions and will be used to benchmark our method against the recent literature. The third example is a finite element model of a system of electrical towers used to showcase the applicability and efficiency of our method on a complex high-dimensional problem.

To assess the accuracy of the proposed method, we consider the following relative error for each example:

$$\varepsilon_\beta = \frac{\left|\beta - \bar{\beta}\right|}{\bar{\beta}}, \tag{22}$$

where $\beta = -\Phi^{-1}\left(P_f\right)$ is the reliability index obtained by a surrogate-assisted approach and $\bar{\beta}$ is the reference solution. The latter is computed using subset simulation tuned such that the resulting coefficient of variation is lower than 0.005.

Algorithm 1 Enrichment process

Initialization:

Get the samples $\mathcal{X}_{\text{sus}}$ generated by subset simulation

Get the current limit state surrogates $\widehat{g}_1, \ldots, \widehat{g}_m$

Get the current experimental designs $\mathcal{D}_1, \ldots, \mathcal{D}_m$

Set the quantile level α for filtering ▷ *e.g.*, $\alpha = 0.01$

Set the maximum number of enrichment points $N_{\max}$ ▷ *e.g.*, $N_{\max} = m$

---

for each $\boldsymbol{x} \in \mathcal{X}_{\text{sus}}$ **do**

  Draw n samples $\left\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(n)}\right\}$ following $f_{\boldsymbol{Z}}$ ▷ $f_{\boldsymbol{Z}} = \prod_{j=1}^{m} \mathcal{N}\left(\mu_{\widehat{g}_j}(\boldsymbol{x}_j), \sigma^2_{\widehat{g}_j}(\boldsymbol{x}_j)\right)$

  Calculate $\widehat{\mu}_{\text{sys}}$ and $\widehat{\sigma}_{\text{sys}}$ following Eq. (16)

  Calculate $\widehat{U}_{\text{sys}}(\boldsymbol{x})$ following Eq. (17)

Calculate U_q, the α-quantile of $\widehat{U}_{\text{sys}}(\mathcal{X}_{\text{sus}})$

Define the reduced candidate set $\mathcal{X}'_{\text{sus}}$ ▷ $\mathcal{X}'_{\text{sus}} = \{\boldsymbol{x} \in \mathcal{X}_{\text{sus}} : U_{\text{sys}}(\boldsymbol{x}) \leq U_q\}$

Partition $\mathcal{X}'_{\text{sus}}$ into C clusters $\{\mathcal{X}_1, \ldots, \mathcal{X}_C\}$ ▷ C is automatically detected by DBSCAN

if $C > N_{\max}$ **then**

  Only keep the $N_{\max}$ samples with smallest U_{sys} values

Get the enrichment samples $\mathcal{X}_{\text{enr}} = \left\{\widetilde{\boldsymbol{x}}^{(1)}, \ldots, \widetilde{\boldsymbol{x}}^{(C)}\right\}$ using Eq. (19)

for each $\widetilde{\boldsymbol{x}}^{(c)} \in \mathcal{X}_{\text{enr}}$ **do**

  Calculate the total Sobol' indices $\left\{S_1^T, \ldots, S_m^T\right\}$ of $h(u_1, \ldots, u_m)$ where $u_j = \boldsymbol{Z}_j\left(\widetilde{\boldsymbol{x}}^{(c)}\right)$

  Find the component j^* with the largest index $S_{j^*}^T$

  Update the corresponding experimental design $\mathcal{D}_{j^*} \leftarrow \mathcal{D}_{j^*} \cup \left(\widetilde{\boldsymbol{x}}_{j^*}^{(c)}, g\left(\widetilde{\boldsymbol{x}}_{j^*}^{(c)}\right)\right)$

  Update the corresponding surrogate model $\widehat{g}_{j^*}$

In active learning, the goal is to find the probability of failure using a limited number of calls to the original model. Therefore, we consider the number of model evaluations N_{eval} as criterion of efficiency. Finally, we assess the robustness of the proposed approach by repeating each analysis 15 times using different initial experimental designs.

Both Kriging and PC-Kriging are considered in the two benchmark examples. For Kriging, we consider a linear trend while for PC-Kriging the maximum polynomial degree of the PCE part is set equal to 3. The convergence threshold in Eq. (4) is set to $\bar{\varepsilon} = 5 \cdot 10^{-3}$. The size of the initial experimental design is defined as $N_j^{(0)} = 2M_j + 1$, where M_j is the effective input dimensionality of the j-th component limit state. Regarding the enrichment scheme, the quantile level for filtering $\mathcal{X}_{\text{sus}}$ into $\mathcal{X}'_{\text{sus}}$ is set equal to $\alpha = 0.01$, while the maximum number of enrichment points allowed per iteration is set to $N_{\max} = M$.

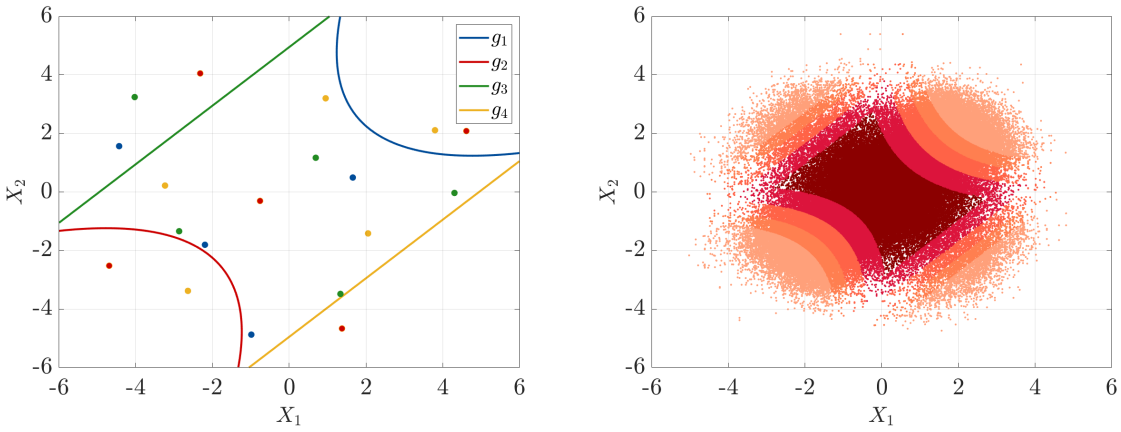
4.1 Four-branch function

This example is popular in the active learning community as it has been used often for benchmarks (Echard et al., 2011). It is a two-dimensional problem, which makes it ideal to illustrate different aspects of the proposed method graphically. We consider a series system which consists of $m = 4$ component limit states defined as (Waarts, 2000):

$$\begin{cases} g_1(\boldsymbol{X}) = 3 + 0.1(X_1 - X_2)^2 - \frac{1}{\sqrt{2}}(X_1 + X_2), \\ g_2(\boldsymbol{X}) = 3 + 0.1(X_1 - X_2)^2 + \frac{1}{\sqrt{2}}(X_1 + X_2), \\ g_3(\boldsymbol{X}) = (X_1 - X_2) + \frac{P}{\sqrt{2}}, \\ g_4(\boldsymbol{X}) = (X_2 - X_1) + \frac{P}{\sqrt{2}}, \end{cases} \tag{23}$$

where $X_1, X_2 \sim \mathcal{N}(0, 1)$ and P is a parameter that has been set to either 6 or 7 in the literature. We consider both values for benchmarking purposes but all illustrations correspond to the case $P = 7$. The reference probability of failure for $P = 7$ (resp. $P = 6$) is $2.24 \cdot 10^{-3}$ (resp. $4.48 \cdot 10^{-3}$). This corresponds to a reliability index of 2.84 (resp. 2.61).

Let us start by illustrating the enrichment process. Figure 2a shows one realization of the initial experimental design together with the limit state surfaces (5 points for each limit state). Different colours correspond to different components. The limit state functions g_1 and g_2 are nonlinear while g_3 and g_4 are linear, and hence easier to approximate. Regarding the initial ED, the use of the hypercube allows us to spread the samples in the input space and reach the failure domain at initialization. This further reduces the chance of premature convergence where one of the failure domains is not discovered.

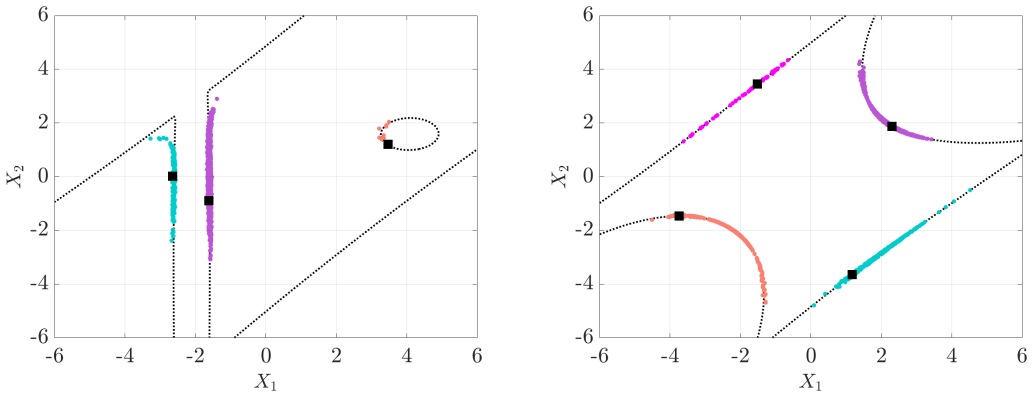


(a) Initial ED and limit state

(b) Candidate pool $\mathcal{X}_{\text{sus}}$

Figure 2: Example 1 - limit state functions, initial experimental designs and candidate pool for enrichment of the four-branch function. In the left panel, the colors of the samples correspond to the limit-state they are associated with. The color shades in the right panel show the different steps of subset simulation.

Figure 2b shows the candidate pool for one enrichment iteration $\mathcal{X}_{\text{sus}}$, which are the samples generated by subset simulation. The different shades of colours represent the different subset iterations. These subsets ensure a higher likelihood of reaching the failure domain compared to crude Monte Carlo simulation. Those samples are then filtered so as to keep a small fraction with the smallest U_{sys}. This is illustrated in Figure 3 for two different iterations when using ordinary Kriging. In these figures, the dotted lines represent the system limit state surface approximation and the coloured samples represent the filtered enrichment samples $\mathcal{X}'_{\text{sus}}$. Each colour corresponds to a different cluster. As it can be seen, the number of identified clusters depends on the current state of the limit state surfaces. At iteration #1 (Figure 3a), three (erroneous) failure modes are identified due to the small ED size that lead to inaccurate surrogate models. Consequently, the enrichment set $\mathcal{X}_{\text{enr}}$ contains three samples which are represented by the black squares. In contrast at iteration #4 (Figure 3b), the surrogate models are more accurate and the four failure modes are properly identified.

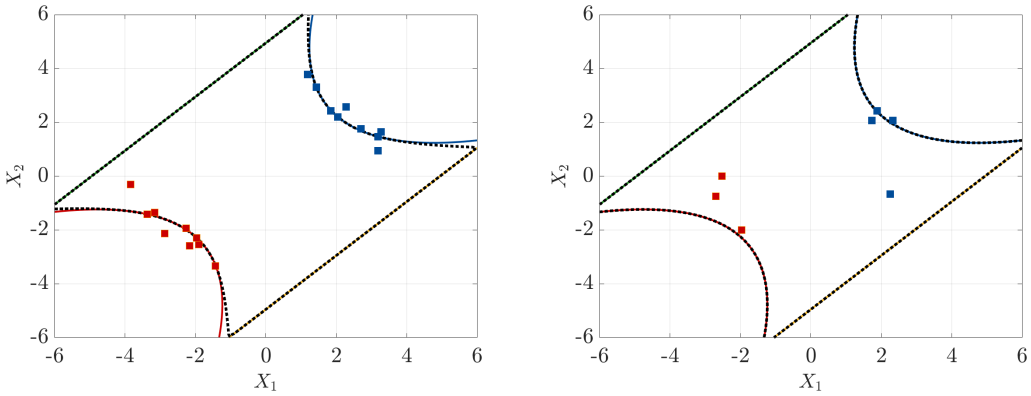


(a) Iteration #1 (b) Iteration #4

Figure 3: Example 1 - limit state functions and enrichment samples (black dots) using ordinary Kriging. The colored dots show the clustered enrichment candidates.

The final set of enrichment samples are illustrated for the median case in Figure 4 for both Kriging and PC-Kriging. The different colours show which limit state was actually evaluated for each sample. It is worth noting here that the linear limit-states g_3 and g_4 are never enriched when using universal Kriging with linear trend or PC-Kriging as they are very well approximated, even when the ED is extremely small.

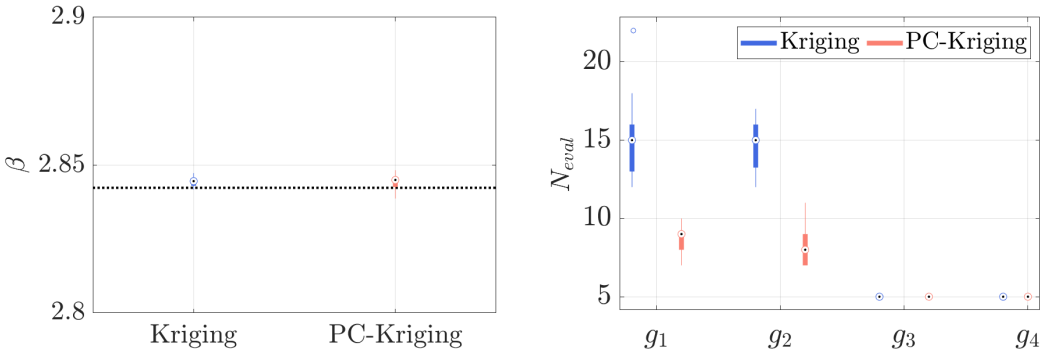
Figure 5 summarizes the results of the analysis considering all 15 repetitions. In details, Figure 5a shows boxplots of the reliability index when using Kriging or PC-Kriging. In the boxplot, the circled dot represents the median solution, the thick line corresponds to the first and third quartiles while the thin whiskers show the range of all solutions, outliers excluded. The horizontal dotted black line corresponds to the reference solution. It can be seen that the proposed framework is extremely accurate in both cases and robust, as very little dispersion can be observed. Figure 5b shows boxplots of the number of models evaluations for each



(a) Kriging (b) PC-Kriging

Figure 4: Example 1 - Enrichment samples for the median case with $P = 7$.

component. As expected limit states g_1 and g_2 which are the more difficult to approximate and which contribute the most to system failure are evaluated more often than g_3 and g_4. As a matter of fact, the latter are very well approximated by both Kriging and PC-Kriging thanks to the polynomial trend and are therefore never enriched in these replications.



(a) Reliability index (b) Number of model evaluations

Figure 5: Example 1 - Boxplots summarizing the results of the 15 repetitions for $P = 7$.

We now compare our results to other methods available in the literature. The case $P = 7$ was mainly used with component reliability where the system limit state $h(\boldsymbol{g}(\boldsymbol{x}))$ was directly approximated. Table 1 shows results gathered by Ling et al. (2019) together with the median results obtained from the repetitions using our proposed method (with Kriging and PC-Kriging). As shown by the error column ε_β, all methods are accurate. However, our method is much more efficient as can be seen from the required number of model evaluations to convergence. The methods from the literature considered all the four limit state function as one, hence the factor 4 in the N_{eval} column. Yet, even without multiplying by four, the methods we propose remain much more efficient. When using Kriging, we converge in 40 model evaluations (median value), which is much smaller than the 61 ($\times 4$) obtained by Ling

et al. (2019). When using PC-Kriging, the convergence is even faster. In total, only 27 samples are required to yield an accurate estimate of the probability of failure.

Table 1: Four-branch function ($P = 7$) performance comparison from Ling et al. (2019).

Method	P_f	β	ε_β	Total N_{eval}
SuS (reference)	$2.239 \cdot 10^{-3}$	2.842	$--$	$\sim 4 \cdot 10^5$
AK-MCS	$2.271 \cdot 10^{-3}$	2.838	$1.524 \cdot 10^{-3}$	102×4
AK-IS	$2.210 \cdot 10^{-3}$	2.847	$1.643 \cdot 10^{-3}$	361.5×4
AK-SS	$2.2480 \cdot 10^{-3}$	2.841	$4.684 \cdot 10^{-4}$	67×4
Ling	$2.2480 \cdot 10^{-3}$	2.841	$4.684 \cdot 10^{-4}$	61.4×4
Proposed (KRG)	$2.224 \cdot 10^{-3}$	2.845	$7.861 \cdot 10^{-4}$	40
Proposed (PCK)	$2.221 \cdot 10^{-3}$	2.845	$9.285 \cdot 10^{-4}$	27

To compare our method with dedicated system reliability methods, we consider the case when $P = 6$. Table 2 shows results gathered by Guo et al. (2021) together with our proposed method. The benchmark methods are this time much more efficient than the ones used above for $P = 7$, since they were precisely developed to solve system problems. They require on average 62 model evaluations. This makes them still largely less efficient than the method we propose, which converges with only 37 and 27 samples for Kriging and PC-Kriging, respectively.

Table 2: Four-branch function ($P = 6$) performance comparison from Guo et al. (2021).

Method	P_f	β	ε_β	Total N_{eval}
SuS (reference)	$4.484 \cdot 10^{-3}$	2.613	$--$	$\sim 3 \cdot 10^5$
AK-SYS	$4.450 \cdot 10^{-3}$	2.616	$1.034 \cdot 10^{-3}$	63.3
ALK-TCR	$4.480 \cdot 10^{-3}$	2.614	$2.692 \cdot 10^{-4}$	61.3
ALK-SIS	$4.450 \cdot 10^{-3}$	2.616	$1.034 \cdot 10^{-4}$	62.4
Proposed (KRG)	$4.500 \cdot 10^{-3}$	2.616	$8.496 \cdot 10^{-4}$	37
Proposed (PCK)	$4.500 \cdot 10^{-3}$	2.616	$8.652 \cdot 10^{-4}$	27

4.2 Roof truss problem

In this example, we consider a roof truss structure which consists of an assembly of beams and bars made of concrete and steel, as illustrated in Figure 6. The structure is subjected to a distributed load on the top beams, which is equivalently modelled as point loads of magnitude $ql/4$ on nodes D and F and $ql/2$ on C, where q is a line load and l is the length of the roof base. It presents three distinct failure modes. The first one is related to the displacement of

node C at the tip of the roof, which is expected to be below a critical threshold set at 3 cm. It is represented by the limit state g_1 in Eq. (24), where A_c and E_c (resp. A_s and E_s) are the cross-sectional areas of the concrete (resp. steel) bars and their constitutive materials Young's moduli. Failure in the second mode (g_2) occurs when the internal force in the bar AD, calculated as $1.185\,ql$, is larger than its ultimate stress $f_c A_c$, where f_c is the compressive strength of the bar AD. Similarly, failure from the third mode (g_3) occurs when the internal force in the bar EC, calculated as $0.75\,ql$, is larger than its ultimate stress $f_s E_s$, where f_s is the tensile strength of the bar EC. System failure is assumed when any of the limit states is violated, leading to a series system $h\left(g_1\left(\boldsymbol{x}\right), g_2\left(\boldsymbol{x}\right), g_3\left(\boldsymbol{x}\right)\right) = \min\left(g_1\left(\boldsymbol{x}\right), g_2\left(\boldsymbol{x}\right), g_3\left(\boldsymbol{x}\right)\right)$ whose components read

$$\begin{cases} g_1\left(\boldsymbol{X}\right) = 0.03 - \frac{ql^2}{2}\left(\frac{3.81}{A_c E_c} + \frac{1.13}{A_s E_s}\right), \\ g_2\left(\boldsymbol{X}\right) = f_c A_c - 1.185\,ql, \\ g_3\left(\boldsymbol{X}\right) = f_s A_s - 0.75\,ql. \end{cases} \tag{24}$$

The reference probability of failure is $3.417 \cdot 10^{-3}$, which corresponds to a reliability index of 2.705 and is obtained from a subset simulation whose setting lead to a coefficient of variation of 0.2%. Note that for this specific example, if we were to assume that this roof is analysed using finite element modelling, a single run would yield all the three limit states. However, for the sake of comparison with other methods, we assume that each limit state is evaluated independently from the others.

The 8 input parameters describing the system are gathered in Table 3 together with their probabilistic distributions. Only a subset of the random variables affects each limit state function, more precisely 6 for g_1 and 4 for g_2 and g_3. The initial EDs are therefore of sizes $N_1^{(0)} = 13$ and $N_2^{(0)} = N_3^{(0)} = 9$.

Table 3: Probabilistic distribution of the random variables associated to the roof truss problem.

Parameter	Distribution	Mean	Coef. of variation
Uniform load q (N/m)	Lognormal	$20,000$	0.07
Length l (m)	Lognormal	12	0.01
Cross-sectional area A_s (m^2)	Lognormal	$9.82 \cdot 10^{-4}$	0.06
Cross-sectional area A_c (m^2)	Lognormal	0.04	0.12
Young's modulus E_s (N/m^2)	Lognormal	$2 \cdot 10^{11}$	0.06
Young's modulus E_c (N/m^2)	Lognormal	$3 \cdot 10^{11}$	0.06
Tensile strength f_s (N/m^2)	Lognormal	$3.35 \cdot 10^{8}$	0.12
Compressive strength f_c (N/m^2)	Lognormal	$1.34 \cdot 10^{7}$	0.18

Figure 7a shows boxplots of the reliability index obtained by repeating the analysis 15 times using Kriging and PC-Kriging. Both cases converge to the reference solution shown

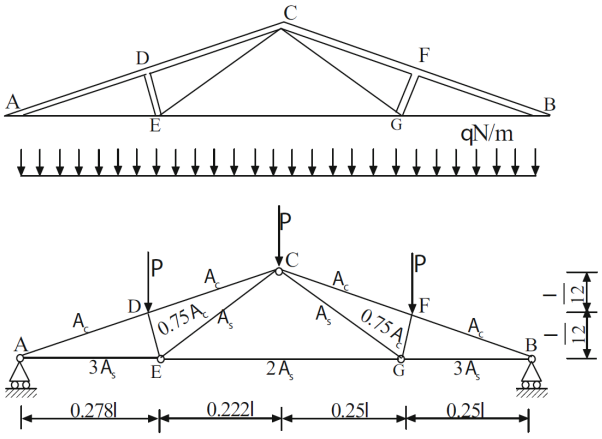
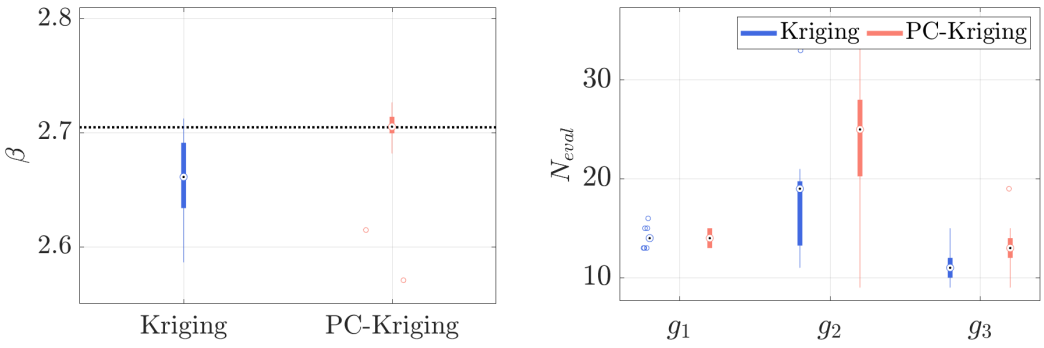


Figure 6: Example 2 - Schematic diagram of the roof structure (from Yun et al. (2019)).

by the black dotted line, but with different degrees of accuracy. While PC-Kriging yields extremely accurate and robust solutions, Kriging is slightly biased. This is due to premature convergence. This can be confirmed by the number of model evaluations shown in Figure 7b, where Kriging requires significantly less model evaluations than PC-Kriging. In general, the limit-state g_2 is the most enriched, showing that it contributes the most to failure, and/or that it is non-linear in nature. Indeed, since it is of comparable complexity as g_3, which is evaluated much less, we may assume that g_2 has a larger contribution to failure. This can actually be confirmed by looking at the component failure probabilities. Using the latest surrogates in one of the repetitions and Monte Carlo simulation, these quantities can be estimated as $P_{f_2} = 3.3 \cdot 10^{-3}$ and $P_{f_3} = 3.5 \cdot 10^{-5}$. P_{f_2} is indeed two order of magnitudes larger than P_{f_3}, which explains why g_2 is more often enriched than g_3 in the active learning scheme.



(a) Reliability index

(b) Number of model evaluations

Figure 7: Example 2 - Boxplots summarizing the results of the 15 repetitions.

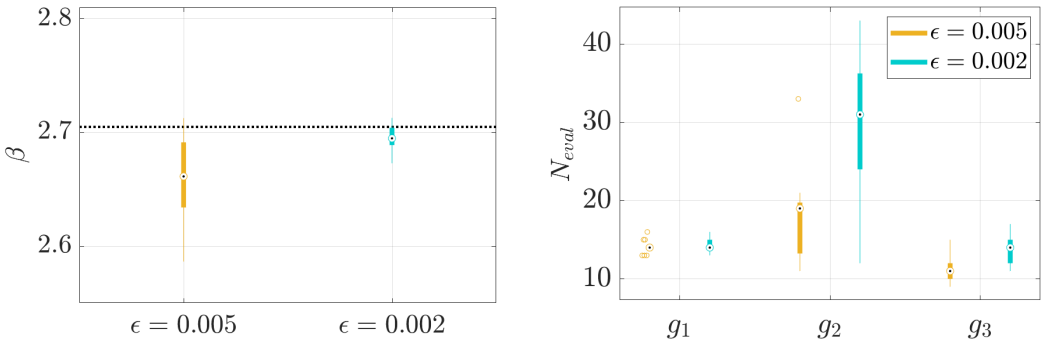
We then compare these results against those gathered by Zhou et al. (2022) in Table 4. All results are accurate, with a relative error on the reliability index of the order of 10^{-4}, except for Kriging for which the median relative error is in the order of 10^{-2}. In terms

of model evaluations, our method is again more twice more efficient than the ones in the literature. PC-Kriging for instance, requires 51 model evaluations in total, while this number grows to 114, 120 and 125 for the methods in the literature.

Table 4: Roof structure results comparison with Zhou et al. (2022).

Method	P_f	β	ε_β	N_1	N_2	N_3	N_{eval}
SuS (reference)	$3.417 \cdot 10^{-3}$	2.705	$--$				$\sim 4 \cdot 10^5$
AK-SYS	$3.337 \cdot 10^{-3}$	2.713	$3.036 \cdot 10^{-3}$	32.1	53.9	34.9	120.4
ALK-TCR	$3.376 \cdot 10^{-3}$	2.709	$1.557 \cdot 10^{-3}$	32	56	37.1	125.1
IK-SRA	$3.350 \cdot 10^{-3}$	2.711	$2.297 \cdot 10^{-3}$	32	48.8	33.2	114.0
Proposed (KRG)	$3.893 \cdot 10^{-3}$	2.661	$1.610 \cdot 10^{-2}$	14	19	11	43
Proposed (PCK)	$3.412 \cdot 10^{-3}$	2.705	$2.062 \cdot 10^{-4}$	14	25	13	51

Kriging requires even less model evaluations, but at the same time is slightly less accurate (although 1.6% error on the reliability index β is more than enough in practice). For a fairer comparison, we run the Kriging-based approach, while setting the convergence criterion to $\bar{\varepsilon} = 0.002$ (previously set at 0.005). Figure 8 shows the resulting reliability index and model evaluations. As expected, they yield more accurate results. The median relative error on β is decreased to 0.37%, while the median number of model evaluations increases to 59. Note that this remains much smaller than the cost of the benchmark methods shown in Table 2.



(a) Reliability index

(b) Number of model evaluations

Figure 8: Example 2 - Results of the 15 repetitions using Kriging with two different stopping criterion thresholds.

4.3 Transmission tower

In this final example, we validate the methodology on a complex engineering system related to power transmission. The system consists of 7 towers, illustrated in Figure 9, and disposed such that electricity is transmitted from Tower 1 to Tower 7. Each tower is subjected to

its own weight and to cable weights. Furthermore, they are in a field with extreme weather conditions and are therefore also subject to wind force.

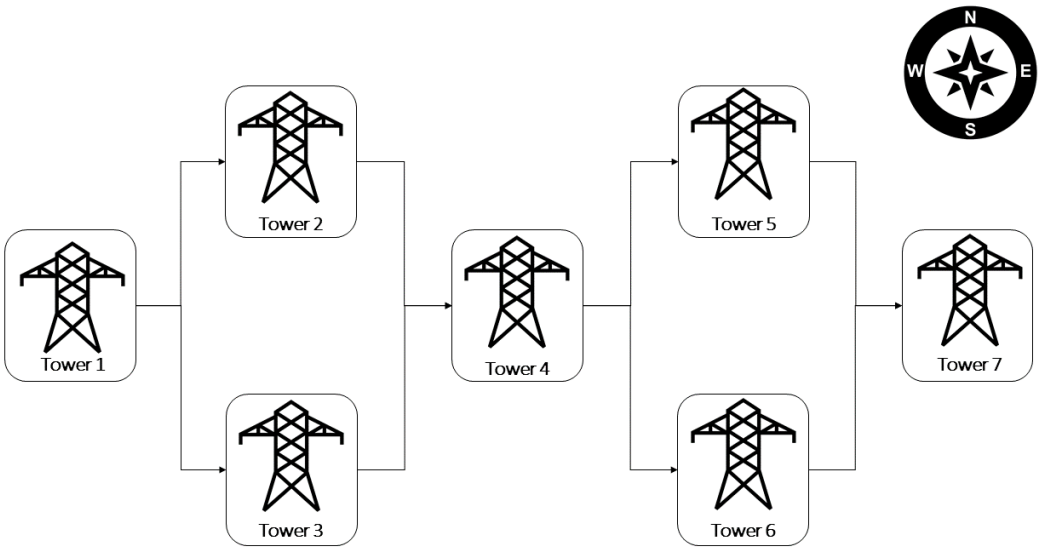


Figure 9: Example 3 - Schematic representation of the power transmission system.

Figure 10 illustrates an archetype of the towers. They consist of 4 groups of bars as shown by the different colours. Each group is characterized by its cross-sectional area and its constitutive material Young's modulus. We further consider that the towers within each row are similar (*front row*: Towers #2 and #5, *middle row*: Towers #1, #4 and #7 and *rear row*: Towers #3 and #6). Their corresponding parameters are denoted by $\{\boldsymbol{A}_f, \boldsymbol{E}_f\}$, $\{\boldsymbol{A}_m, \boldsymbol{E}_m\}$ and $\{\boldsymbol{A}_r, \boldsymbol{E}_r\}$. The cable weights P are applied vertically on the extreme hands of the towers while the wind loads $F_1, \ldots, F_7$ are applied to the towers tips, as illustrated in Figure 10. The direction of the wind for each tower is given by the angles $\alpha_1, \ldots, \alpha_7$.

To model the wind field, we assume that the wind comes from North to South and is therefore stronger on the front row. The dependency on the applied loads which is due to this wind field is described using copula theory (Nelsen, 2006). We consider two different pair copulas (Bedford and Cooke, 2002):

- Wind force copula C_{WF}: The dependency among the wind forces F_1 to F_7 is modelled by a C-Vine copula with a truncation of second order. To account for the strong wind coming from North to South, we use a Gumbel copula in order to model upper tail dependence between towers on the front row (*i.e.* Towers #2 and #5) and the remaining ones. It means that if we observe a strong wind force in the front row, we are likely to also observe strong ones in the remaining towers. We consider however that this correlation is strongly reduced past the first row, therefore we consider the remaining towers independent. In summary, the copula density is written as:

$$c_{\text{WF}}\left(\boldsymbol{u}\right) = \prod_{i=1}^{6} \prod_{j=1}^{7-i} c_{j,j+i|\{1,\ldots,j-1\}}\left(u_{j|\{1,\ldots,j-1\}}, u_{j+i|\{1,\ldots,j-1\}}\right), \tag{25}$$

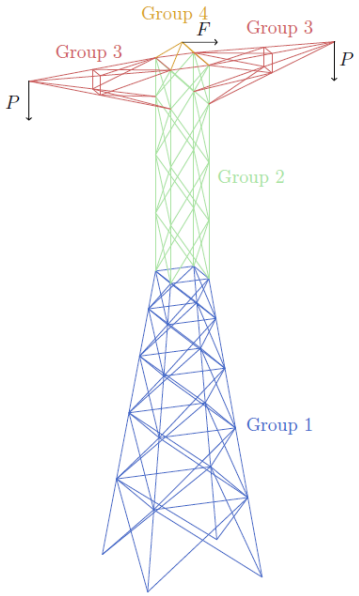


Figure 10: Example 3 - Finite element model of a single transmission tower.

where the pair copulas are given by

$$\begin{cases} c_{2,j} : \text{Gumbel}\,(\theta = 5) & \text{for } j \neq 2 \\ c_{5,j|2} : \text{Gumbel}\,(\theta = 5) & \text{for } j \neq 2, 5 \\ c_{i,j|5,2} : \text{Independent} & \text{for } i \neq j,\ i, j \neq 2, 5. \end{cases} \tag{26}$$

Using this setting the Person's (resp. Kendal's) correlation coefficients (Kendall, 1970) between these seven parameters can be estimated to vary between 0.96 and 0.99 (resp. 0.80 and 0.93).

- Wind angle copula $C_{\mathrm{W}\alpha}$: The dependency between the wind angles α_1 to α_7 is modelled using a truncated C-Vine copula. We consider a t-copula to model both upper and lower tail dependencies. This means that if we observe a strong inclination (either east or west) in the front row, we are likely to observe a strong inclination in the remaining towers. The copula density is written as:

$$c_{\mathrm{W}\alpha}\left(\boldsymbol{u}\right) = \prod_{i=1}^{6}\prod_{j=1}^{7-i} c_{j,j+i|\{1,\ldots,j-1\}}\left(u_{j|\{1,\ldots,j-1\}}, u_{j+i|\{1,\ldots,j-1\}}\right), \tag{27}$$

where the pair copulas are given by

$$\begin{cases} c_{2,j} : t - \text{copula}\,(\theta = 0.4, \nu = 2) & \text{for } j \neq 2 \\ c_{5,j|2} : t - \text{copula}\,(\theta = 0.4, \nu = 2) & \text{for } j \neq 2.5 \\ c_{i,j|5,2} : \text{Independent} & \text{for } i \neq j, ij \neq 2, 5. \end{cases} \tag{28}$$

Using this setting, the Person's (resp. Kendal's) correlation coefficient between these seven parameters can be estimated to vary between 0.23 and 0.48 (resp. 0.17 and 0.36). The reader

is referred to Torre et al. (2019); Lataniotis et al. (2022) for more details about the use of copula for modelling complex engineering systems and their implementation in UQLab.

The system is assumed to satisfy all operational requirements if electricity can be transmitted from Tower 1 to Tower 7. Failure for each tower is assumed when the maximum stress $S_{\max_j}$ in any bar is larger than the yield stress f_y, which leads to the following limit states:

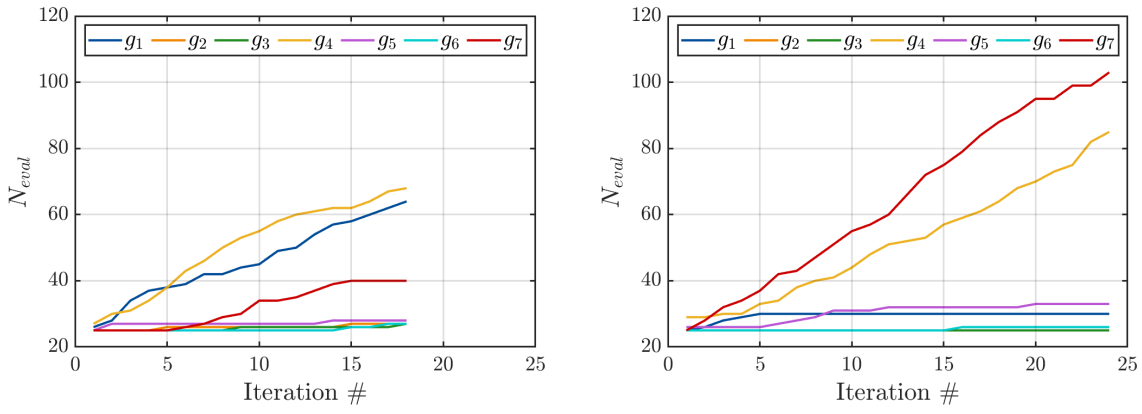
$$g_j\left(\boldsymbol{X_j}\right) = f_y - S_{\max_j}, \qquad j = \{1, \ldots, 7\}\,. \tag{29}$$

The maximum stress is computed using a truss finite element code programmed in Matlab. For each tower, the limit state has an input $\boldsymbol{X}_j$ of dimension 12 even though the entire set inputs $\boldsymbol{X}$ is of dimension 40. Their marginal probabilistic distributions are given in Table 5.

The reference solution, computed using subset simulation, is $P_f = 5.10 \cdot 10^{-4}$ ($\beta = 3.28$) with a coefficient of variation of 0.82%. This is obtained by using a large subset simulation which has required a total of $2.375 \cdot 10^6$ full evaluations of the computational model.

For this example, we run the surrogate-based analysis with a stopping criterion set at $\bar{\varepsilon} = 10^{-3}$. However, we summarize the results for different intermediate thresholds in Table 6. The initial experimental design is of size 25 for each limit state. As with the previous examples, Kriging stops earlier than PC-Kriging but the latter is more accurate. In both cases though, the solutions are relatively accurate with an error in the order of 0.1% to 1% on the reliability index. The number of model evaluations is also remarkably small given the problem dimensionality and complexity. Note that this number is cumulated for all 7 components.

Figure 11 shows the evolution of the number of model evaluations for each component throughout the enrichment iterations. Limit states #1, #4 and #7, which are the most crucial, are the ones that are enriched the most. This is due to the fact that limits states #2 and #5 are in parallel with #3 and #6, hence making them less relevant to system failure.



(a) Kriging

(b) PC-Kriging

Figure 11: Example 3 - Evolution of the number of model evaluations for each component.

Finally, Figure 12 shows the convergence of the reliability index. While limit state g_7 is enriched less than g_1 and g_4 for Kriging, g_1 is hardly enriched for PC-Kriging. These may explain the bias observed in the convergence curves. We can indeed again see that Kriging converges earlier but to a slightly biased solution. PC-Kriging yields a more accurate solution for which the 95% confidence interval is within that of the reference solution. These two cases however, together with the previous example, confirm the need to develop more robust stopping criteria for the proposed methodology.

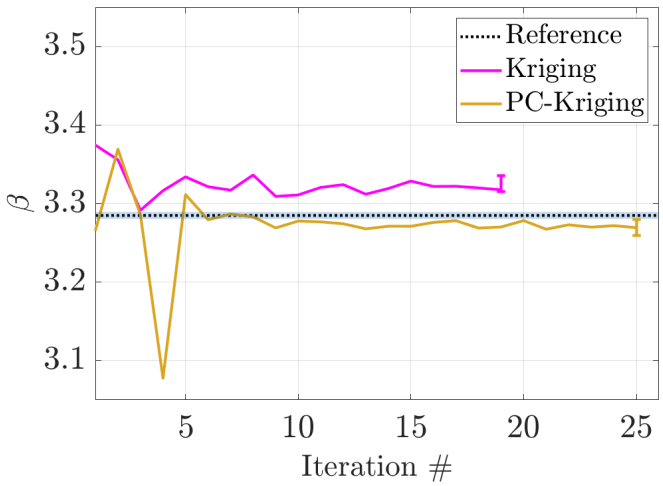


Figure 12: Example 3 - Evolution of the reliability index.

5 Conclusion

The literature on structural reliability analysis generally distinguishes component from system reliability analysis. While the former considers a single limit state function, the latter models more complex and realistic systems using multiple limit states. Component reliability has been largely studied and can be very efficiently solved using active learning approaches. Such approaches are, however, sub-optimal in the context of system reliability because of the presence of multiple failure domains and the uneven contribution of each limit state to system failure.

In this work, we propose an active learning strategy that aims at efficiently solving system reliability problems in any arbitrary configuration. To this end, we consider approximating each limit state function by a separate surrogate model and devise an enrichment scheme that allows updating only the experimental design of the most influential limit states at each iteration. The efficiency of the proposed method stems from (i) a new learning function adapted to systems in arbitrary configuration, which does not assume any typical configuration (series or parallel), (ii) the use of density-based clustering of the enrichment candidates to automatically identify the different failure modes and enrich them separately and (iii) the

use of Sobol' sensitivity analysis to update only the limit states that effectively contribute to system failure.

The algorithm is validated on two analytical examples: a 2-dimensional four-branch function and an 8-dimensional roof structure. Both examples show that the proposed method is efficient and robust. The results are compared against similar methods in the literature and are shown to outperform them. On average the number of calls to the limit state functions is divided by two compared to the most recent state-of-the-art active learning methods. Finally, a complex engineering system comprising 7 towers under extreme weather conditions is analyzed. This case corresponds to a general system (series or parallel systems) with a total of 40 input random variables. In this case study, the developed method is shown to converge to the reference solution using a very limited number of model evaluations (in the range of $250 - 300$), in particular when considering the problem dimensionality and complexity.

The proposed methodology inherits limitations of Kriging/PC-Kriging and DBSCAN in terms of dimensionality. While approximating each limit-state separately helps when only the system is of large dimension (as shown in the transmission tower example), the approach may break if some of the components are high-dimensional. For DBSCAN, we have proposed using dimensionality reduction. This may also be applied for building the surrogate models if the dimension of the component limit-state is too large for traditional Kriging/PC-Kriging.

The methodology we propose focuses on the enrichment scheme. Nonetheless, other aspects may affect its efficiency, especially when dealing with complex failure domain topologies or extremely low probabilities of failure. In such scenarios, the robustness of the methodology can be enhanced by using a larger initial experimental design, or one that is more widely distributed. This approach increases the likelihood of obtaining an initial sample within or in close proximity to the failure domain, which in turn reduces the risk of overlooking a failure domain.

A Density-based clustering (DBSCAN)

DBSCAN is a hierarchical clustering algorithm proposed by Ester et al. (1996). Its main advantages are that it can discover clusters of arbitrary shapes, does not require a pre-defined number of clusters to identify and is efficient on large datasets.

DBSCAN is a density-based method that relies on the premise that a cluster is a subset of data points with high density separated to another cluster by an area of low point density. To formalize this, DBSCAN defines the ε-neighborhood of a point $\boldsymbol{x}$ as the set of points whose distance to $\boldsymbol{x}$ is smaller than ε, *i.e.*,

$$\mathcal{N}_{\varepsilon}\left(\boldsymbol{x}\right) = \left\{\boldsymbol{s} \in \mathcal{X} : d\left(\boldsymbol{x}, \boldsymbol{s}\right) \leq \varepsilon\right\}, \tag{30}$$

where $d\left(\bullet, \bullet\right)$ is a distance measure. The algorithm is independent of the considered distance measure. We consider here the Euclidean distance.

Using the concept of ε-neighborhood, the sample points can then be categorized into:

- *Core points*, which are points that contain a minimum of $N_{\min}$ points in their ε-neighborhood. They constitute the basis for defining a cluster, as each cluster must contain at least one core point.
- *Border points*, which are points that have a core point but fewer than $N_{\min}$ points in their ε-neighborhood.
- *Noise points*, which are neither core nor border points. They do not belong to any cluster.

DBSCAN uses these definitions to recursively partition the data set into clusters of arbitrary shapes. Schematically, the algorithm starts by randomly picking an unlabeled point in the data set. It then gathers points from its ε-neighborhood and if there are at least $N_{\min}$ of them, then they all constitute a new cluster. The neighborhood is then recursively extended for each point in the current neighborhood until no point can be added. The algorithm restarts then with the next unlabeled point, which will then be part of a new cluster.

The algorithm can be summarized as follows:

Algorithm 2 DBSCAN (Ester et al., 1996)

Initialization:

Define ε, the minimal distance to define the close neighborhood of a point

Define $N_{\min}$, the minimal number of points required to consider a region dense enough to be a cluster

Mark all points as unlabelled

Define the unlabelled points as $\mathcal{X}_{\text{unlab}}$

---

for $\boldsymbol{x} \in \mathcal{X}_{\text{unlab}}$ **do**
  if $\boldsymbol{x}$ is unlabelled **then**
    Mark $\boldsymbol{x}$ as labelled
    Find the set of points belonging to $\mathcal{N}_\varepsilon(\boldsymbol{x})$
    if $|\mathcal{N}_\varepsilon(\boldsymbol{x})| < N_{\min}$ **then**
      Label $\boldsymbol{x}$ as noise point
    else
      Label $\boldsymbol{x}$ as a core point
      Start a new cluster and add $\boldsymbol{x}$ to this cluster
      for Each point s in $\mathcal{N}_\varepsilon(\boldsymbol{x})$ **do**
        Assign $\boldsymbol{s}$ to current cluster
        Get the ε-Neighborhood of s: $\mathcal{N}_\varepsilon(\boldsymbol{s})$
        if $|\mathcal{N}_\varepsilon(\boldsymbol{s})| \geq N_{\min}$ **then** $\mathcal{N}_\varepsilon(\boldsymbol{x}) \leftarrow \mathcal{N}_\varepsilon(\boldsymbol{x}) \cup \mathcal{N}_\varepsilon(\boldsymbol{s})$

Table 5: Power transmission tower input marginals properties (θ_1 and θ_2 are the mean and the coefficient of variations for non-uniform variables, and the support bounds for uniform variables.)

Group	Variable	Distribution	θ_1	θ_2
Cross-sectional area - front row	A_{f_1} (m^2)	Gaussian	$1 \cdot 10^{-3}$	0.05
	A_{f_2} (m^2)	Gaussian	$1 \cdot 10^{-3}$	0.05
	A_{f_3} (m^2)	Gaussian	$5 \cdot 10^{-3}$	0.05
	A_{f_4} (m^2)	Gaussian	$5 \cdot 10^{-3}$	0.05
Young's modulus - front row	E_{f_1} (Pa)	Gaussian	$210 \cdot 10^9$	0.05
	E_{f_2} (Pa)	Gaussian	$210 \cdot 10^9$	0.05
	E_{f_3} (Pa)	Gaussian	$210 \cdot 10^9$	0.05
	E_{f_4} (Pa)	Gaussian	$210 \cdot 10^9$	0.05
Cross-sectional area - midlle row	A_{m_1} (m^2)	Gaussian	$1 \cdot 10^{-3}$	0.05
	A_{m_2} (m^2)	Gaussian	$1 \cdot 10^{-3}$	0.05
	A_{m_3} (m^2)	Gaussian	$5 \cdot 10^{-3}$	0.05
	A_{m_4} (m^2)	Gaussian	$5 \cdot 10^{-3}$	0.05
Young's modulus - middle row	E_{m1} (Pa)	Gaussian	$210 \cdot 10^9$	0.05
	E_{m_2} (Pa)	Gaussian	$210 \cdot 10^9$	0.05
	E_{m_3} (Pa)	Gaussian	$210 \cdot 10^9$	0.05
	E_{m_4} (Pa)	Gaussian	$210 \cdot 10^9$	0.05
Cross-sectional area - rear row	A_{r_1} (m^2)	Gaussian	$1 \cdot 10^{-3}$	0.05
	A_{r_2} (m^2)	Gaussian	$1 \cdot 10^{-3}$	0.05
	A_{r_3} (m^2)	Gaussian	$5 \cdot 10^{-3}$	0.05
	A_{r_4} (m^2)	Gaussian	$5 \cdot 10^{-3}$	0.05
Young's modulus - rear row	E_{r_1} (Pa)	Gaussian	$210 \cdot 10^9$	0.05
	E_{r_2} (Pa)	Gaussian	$210 \cdot 10^9$	0.05
	E_{r_3} (Pa)	Gaussian	$210 \cdot 10^9$	0.05
	E_{r_4} (Pa)	Gaussian	$210 \cdot 10^9$	0.05
Wind load - front row	F_2, F_5 (N)	Gumbel	$3.25 \cdot 10^4$	0.30
Wind load - middle row	F_1, F_4, F_7 (N)	Gumbel	$3.25 \cdot 10^4$	0.30
Wind load - rear row	F_3, F_6 (N)	Gumbel	$3.25 \cdot 10^4$	0.30
Wind angle	$\alpha_1, \ldots, \alpha_7$ (deg)	Uniform	-30	30
Cables weight	P (N)	Gumbel	$1 \cdot 10^4$	0.30
Yield stress	f_y (MPa)	Lognormal	355	0.20

Table 6: Example 3 - Results using three different stopping criterion thresholds.

Method	$\bar{\varepsilon}$	$\widehat{P}_f$	$\widehat{\beta}$	ε_β	Total N_{eval}
SuS (reference)	$--$	$5.102 \cdot 10^{-4}$	3.285	$--$	$2.375 \cdot 10^{6}$
Using Kriging	0.005	$4.435 \cdot 10^{-4}$	3.324	$1.195 \cdot 10^{-2}$	245
	0.002	$4.542 \cdot 10^{-4}$	3.317	$9.921 \cdot 10^{-3}$	281
	0.001	$4.542 \cdot 10^{-4}$	3.317	$9.921 \cdot 10^{-3}$	281
Using PC-Kriging	0.005	$5.396. \cdot 10^{-4}$	3.269	$4.827 \cdot 10^{-3}$	221
	0.002	$5.420 \cdot 10^{-4}$	3.268	$5.198 \cdot 10^{-3}$	248
	0.001	$5.390 \cdot 10^{-4}$	3.269	$4.720 \cdot 10^{-3}$	327

References

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.

Bedford, T. and R. M. Cooke (2002). Vines – a new graphical model for dependent random variables. *Annals of Statistics 30*(4), 1031–1068.

Bichon, B. J., S. Mahadevan, and M. S. Eldred (2009). Reliability-based design optimization using efficient global reliability analysis. In *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Volume 47.

Blatman, G. and B. Sudret (2011). Adaptive sparse polynomial chaos expansion based on least angle regression. *Journal of Computational Physics 230*(6), 2345–2367.

Boroson, E. and S. Missoum (2017). Stochastic optimization of nonlinear energy sinks. *Structural and Multidisciplinary Optimization 55*, 633–646.

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

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.

Efron, B., T. Hastie, I. Johnstone, and R. Tibshirani (2004). Least angle regression. *Annals of Statistics 32*, 407–499.

Ester, M., H.-P. Kriegel, J. Sander, and X. Xu (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In E. Simoudis, J. Han, and U. Fayyad (Eds.), *Proc. 2nd international conference on knowledge discovery and data mining, Portland, Oregon, USA, August 2-4, 1996*. AAAI press.

Fauriat, W. and N. Gayton (2014). AK-SYS: An adaptation of the AK-MCS method for system reliability. *Reliability Engineering & System Safety 123*, 137–144.

Feng, K., Z. Lu, Y. Yang, C. Ling, P. He, and Y. Dai (2023). Novel Kriging based learning function for system reliability analysis with correlated failure modes. *Reliability Engineering & System Safety 239*, 109529.

Guo, Q., Y. Liu, B. Chen, and Q. Yao (2021). A variable and mode sensitivity analysis method for structural system using a novel active learning Kriging model. *Reliability Engineering & System Safety 206*, 107285.

Hu, R. and M. Ludkovski (2017). Sequential design for ranking response surfaces. *SIAM/ASA International Journal of Uncertainty Quantification 5*(1), 212–239.

Hu, Z. and S. Mahadevan (2016). Sensitivity analysis-based surrogate modeling of limit states. In *18th AIAA Non-Deterministic Approaches Conference, AIAA SciTech, (AIAA 2016-0427)*.

Iooss, B., B. Sudret, L. Piano, and C. Prieur (2022). Editorial for the special issue on "sensitivity analysis of model outputs" reliability engineering and system safety. *Reliability Engineering & System Safety 223*, 108477.

Kendall, M. G. (1970). *Rank correlation methods*. Griffin.

Lataniotis, C., E. Torre, S. Marelli, and B. Sudret (2022). UQLab user manual – INPUT module. Technical report, Chair of Risk, Safety & Uncertainty Quantification, ETH Zurich. Report # UQLab-V2.0-102.

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

Ling, C., Z. Lu, K. Feng, and X. Zhang (2019). A coupled subset simulation and active learning Kriging reliability analysis method for rare failure events. *Structural and Multidisciplinary Optimization 60*, 2325–2341.

McKay, M. D., R. J. Beckman, and W. J. Conover (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics 2*, 239–245.

Meeds, E. and S. Osindero (2005). An alternative infinite mixture of Gaussian process experts. In B. Weiss, B. Schölkpf, and J. Platt (Eds.), *Advances in Neural Information Processing Systems 18 (NIPS 2005), Vancouver, British Columbia, Canada, December 5th-8th, 2005*.

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

Moustapha, M., S. Marelli, and B. Sudret (2022). Active learning for structural reliability: Survey, general framework and benchmark. *Structural Safety 96*(102174).

Moustapha, M. and B. Sudret (2023). Learning non-stationary and discontinuous functions using clustering, classficiation and Gaussian process modelling. *RComputers & Structures 281*, 107035.

Nelsen, R. B. (2006). *An introduction to copulas* (2nd ed.), Volume 139 of *Lecture Notes in Statistics*. Springer-Verlag, New York.

Perrin, G. (2016). Active learning surrogate models for the conception of systems with multiple failure modes. *Reliability Engineering & System Safety 149*, 130–136.

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

Ross, S. M. (2010). Reliability theory. In S. M. Ross (Ed.), *Introduction to probability models*, Chapter 9. Elsevier Academic Press.

Sadoughi, M., M. Li, and C. Hu (2018). Multivariate system reliability analysis considering highly nonlinear and dependent safety events. *Reliability Engineering & System Safety 180*, 189–200.

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

Schöbi, R., S. Marelli, and B. Sudret (2022). UQLab user manual – PC-Kriging. Technical report, Chair of Risk, Safety & Uncertainty Quantification, ETH Zurich. Report # UQLab-V2.0-109.

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

Schöbi, R., B. Sudret, and J. Wiart (2015). Polynomial-chaos-based Kriging. *International Journal for Uncertainty Quantification 5*(2), 171–193.

Sobol', I. M. (1967). Distribution of points in a cube and approximate evaluation of integrals. *U.S.S.R Computational Mathematics and Mathematical Physics 7*, 86–112.

Sobol', I. M. (1993). Sensitivity estimates for nonlinear mathematical models. *Math. Modeling & Comp. Exp. 1*(4), 407–414.

Spall, J. C. (2003). *Introduction to stochastic search and optimization: Estimation, simulation and control*, Chapter 14: Simulation-based optimization I: regression, common random numbers, and selection methods. John Wiley & Sons.

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

Torre, E., S. Marelli, P. Embrechts, and B. Sudret (2019). A general framework for data-driven uncertainty quantification under complex input dependencies using vine copulas. *Probabilistic Engineering Mechanics 55*, 1–16.

Tresp, V. (2000). Mixture of Gaussian processes. In T. Leen, T. G. Dietterich, and V. Tresp (Eds.), *Proc. Advances in Neural Information Processing Systems 13, Denver, CO, USA, 2000*.

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, The Netherlands.

Wang, J., Z. Sun, and R. Cao (2021). An efficient and robust Kriging-based method for system reliability analysis. *Reliability Engineering & System Safety 216*, 107953.

Wang, L. and Y. Chen (2023). An improved active Kriging method for reliability analysis combining expected improvement and U learning functions. *Proceedings of the Instituation of Mechanical Engineers, Part 0: Journal of Risk and Reliability*.

Wei, P. and C. Liu, F. anf Tang (2018). Reliability and reliability-based importance analysis of structural systems using multiple response Gaussian process model. *Reliability Engineering & System Safety 175*, 183–195.

Wu, H., Z. Zhu, and X. Du (2020). System reliability analysis with autocorrelated Kriging predictions. *Journal of Mechanical Design 142*, 101702–1 – 101702–11.

Xiao, N.-C., K. Yuan, and H. Zhan (2022). System reliability analysis based on dependent Kriging predictions and parallel learning strategy. *Reliability Engineering & System Safety 218*, 108083.

Xiu, D. and G. E. Karniadakis (2002). The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing 24*(2), 619–644.

Xu, C., Y. Yang, H. Wu, and J. Zhou (2023). A new active learning method for system reliability analysis with multiple failure modes. *Reliability Engineering & System Safety 240*, 109614.

Yang, S., H. Jo, L. Lee, and I. Lee (2022). Expected system improvement (ESI): A new learning function for system reliability analysis. *Reliability Engineering & System Safety 222*, 108449.

Yang, X., X. Chen, T. Wang, and C. Mi (2020). System reliability analysis with small failure probability based on active learning Kriging model and multimodal adaptive importance sampling. *Structural and Multidisciplinary Optimization 62*, 581–596.

Yang, X., X. Liu, C. Mi, and C. Tang (2018). System reliability analysis through active learning Kriging model with truncated candidate region. *Reliability Engineering & System Safety 169*, 235–241.

Yuan, K., N.-C. Xiao, Z. Wang, and K. Shang (2020). System reliability analysis by combining structure function and active learning Kriging model. *Reliability Engineering & System Safety 195*, 106734.

Yun, W., Z. Lu, Y. Zhou, and X. Jiang (2019). AK-SYSi: an improved adaptive Kriging model for system reliability analysis with multiple failure modes by a refined U learning function. *Structural and Multidisciplinary Optimization 59*, 263–278.

Zhou, C., N.-C. Xiao, M. J. Zuo, and W. Gao (2022). An improved Kriging-based approach for system reliability analysis with multiple failure modes. *Engineering with Computers 38*, 1813–1833.