UNCECOMP 2015
1st ECCOMAS Thematic Conference on
Uncertainty Quantification in Computational Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou (eds.)
Crete Island, Greece, 25–27 May 2015

RELIABILITY ASSESSMENT WITH ADAPTIVE SURROGATES BASED ON SUPPORT VECTOR MACHINE REGRESSION

Jean-Marc Bourinet

IFMA
Institut Pascal, CNRS UMR 6602
Clermont Ferrand, France
e-mail: bourinet@ifma.fr

Keywords: reliability assessment, rare events, adaptive surrogate models, support vector machines, regression

Abstract. *Reliability assessment in the context of rare failure events still suffers from its computational cost despite some available methods widely accepted by researchers and engineers. Monte Carlo simulation methods even in their most efficient version such as subset simulation often require a large number of samples for an acceptable accuracy on the failure probability of interest. For low to moderately high dimensional problems and under the assumption of a rather smooth limit-state function, surrogate models a.k.a. metamodels or response surfaces represent interesting alternative solutions. This paper presents an adaptive method based on SVM surrogates used in regression. The SVM formulation hinges on the ϵ-insensitive loss function and a Gaussian RBF kernel. The key idea of the proposed method is to iteratively construct SVM surrogates which become more accurate as they get closer to the limit-state surface. Subset simulation is applied to the constructed SVM surrogates for assessing probabilities with respect to intermediate threshold values of the LSF and more importantly for generating new training points. The efficiency of the method is tested on several examples featuring various challenges: a parallel system, a highly curved limit-state surface at a single most probable failure point and a smooth high-dimensional limit-state surface with equal curvatures. The paper puts an emphasis on the key role of an optimal selection of SVM surrogate hyperparameters. This is achieved in the present work by minimizing an estimate of the leave-one-out error using the cross-entropy method.*

1 INTRODUCTION

The present paper focuses on the assessment of probabilities of rare events. This problem known as structural reliability in structural engineering has been largely addressed by several researchers for decades. Despite some efficient and widely accepted methods, assessing low to very low failure probabilities is still often too computationally demanding in real applications in which a single call to the physical model may last minutes, hours or even days. We are presently interested in time invariant reliability problems as defined in the structural reliability literature, see *e.g.* Ditlevsen and Madsen [1], also known as static simulation problems or models by some other authors, see *e.g.* Homem-de-Mello and Rubinstein [2] and Cancela et al. [3] and in which time is not an explicit variable. The probability w.r.t. an undesired or unsafe state of the system of interest is expressed in terms of a n-dimensional random vector $\mathbf{X}$ of known continuous joint density function $f_{\mathbf{X}}$. Failure is defined in terms of a so-called limit-state function (LSF) $g : \mathcal{X} \subseteq \mathbb{R}^n \to \mathbb{R}, \mathbf{x} \mapsto g(\mathbf{x})$, where $\mathbf{x}$ represents a realization of the random vector $\mathbf{X}$. We restrict here the analysis to a single function g, but this function may represent a combination of failure modes in more general settings. This LSF divides the space of realizations of the random vector $\mathbf{X}$ in a failure domain $\mathcal{F}_{\mathbf{x}} = \{\mathbf{x} \in \mathcal{X} : g(\mathbf{x}) \leq 0\}$ and a safety domain $\overline{\mathcal{F}_{\mathbf{x}}} = \{\mathbf{x} \in \mathcal{X} : g(\mathbf{x}) > 0\}$. The failure probability p_{f} therefore reads:

$$p_{\mathrm{f}} = \int_{\mathcal{F}_{\mathbf{x}}} f_{\mathbf{X}}(\mathbf{x})\, \mathrm{d}\mathbf{x} = \mathbb{E}_{f_{\mathbf{X}}}\left[\mathbb{1}_{\mathcal{F}_{\mathbf{x}}}(\mathbf{X})\right] \tag{1}$$

where $\mathrm{d}\mathbf{x} = \mathrm{d}x_1 \ldots \mathrm{d}x_n$ and $\mathbb{1}_{\mathcal{D}}$ is the indicator function over $\mathcal{D}$ domain: $\mathbb{1}_{\mathcal{D}}(\mathbf{x}) = 1$ if $\mathbf{x} \in \mathcal{D}$, $\mathbb{1}_{\mathcal{D}}(\mathbf{x}) = 0$ otherwise.

We will assume here that we can reformulate this problem in the so-called standard space $\mathbf{u} \in \mathbb{R}^n$, where $\mathbf{U}$ is a random vector with independent standard normal components and $\varphi_n(\mathbf{u})$ is the n-dimensional standard normal joint probability density function (pdf). This could be achieved *e.g.* by means of the Nataf or Rosenblatt transform, not recalled here for the sake of brevity.

$$p_{\mathrm{f}} = \int_{\mathcal{F}_{\mathbf{u}}} \varphi_n(\mathbf{u})\, \mathrm{d}\mathbf{u} = \mathbb{E}_{\varphi_n}\left[\mathbb{1}_{\mathcal{F}_{\mathbf{u}}}(\mathbf{U})\right] \tag{2}$$

where $\mathcal{F}_{\mathbf{u}} = \{\mathbf{u} \in \mathbb{R}^n : G(\mathbf{u}) = g(\mathbf{x}(\mathbf{u})) \leq 0\}$ and $\mathrm{d}\mathbf{u} = \mathrm{d}u_1 \ldots \mathrm{d}u_n$.

Failure probabilities are usually assessed by means of two main types of methods: sampling methods such as the Monte Carlo method which give an estimate of p_{f} based on samples of the random vector $\mathbf{X}$ or approximation methods which consist in constructing a surrogate of the true limit-state function g (or G in the standard space) based on some a priori selected assumptions. Sampling methods do not make any hypothesis on g but they are on a general basis characterized by their high computational cost which may hamper their use in practice with costly-to-evaluate LSF. Even subset simulation [4] which is one of the most efficient technique still requires thousands of calls to the LSF for estimating p_{f} with an acceptable accuracy. In approximation methods, we formulate some hypothesis about the LSF. For instance, FORM and SORM techniques respectively consider the first- and second-order Taylor polynomial of G at the so-called most probable failure point (MPP) $\mathbf{u}^*$ in the standard space. Several other approximation models have been investigated in the literature for reliability assessment including polynomial response surfaces, artificial neural networks, moving least-squares, Kriging (also known as Gaussian process emulators), support vector machines (SVM), polynomial chaos expansions among others. All these approximations are usually referred to as surrogate models or metamodels. The construction of such surrogate models is most often made adaptively. A

common idea is to start from an initial set of training points which is progressively enriched by sequentially adding new training points. The criteria for selecting new training points are based on the information gained from the currently constructed surrogate model. Such adaptive techniques have been developed *e.g.* with Kriging [5–7]), SVM [8–12] and polynomial chaos expansions [13, 14].

This paper proposes an adaptive technique for assessing low failure probabilities based on SVM surrogates used in regression. The conceptual ideas share some similarities with those of the method developed in [12] in which SVM were used in classification. The SVM model used in the present work is based on the ϵ-insensitive loss function as explored in the context of reliability assessment in a few other works [9, 15–17]. A sequence of adaptive SVR surrogates $\widetilde{G}_s(\mathbf{u})$, $s = 1, \ldots, s_{\max}$, is constructed in the standard space with training points which get closer to the failure domain $\mathcal{F}_\mathbf{u}$ with iteration s. The new training points at each iteration s are generated from the currently constructed SVR surrogate model $\widetilde{G}_s$ by means of Monte Carlo Markov chains (MCMC). A new SVR surrogate is trained at each iterations s. Optimal values for its hyperparameters are accurately determined by minimization of an estimate of the leave-one-out (LOO) error proposed by Chang and Lin [18] using the cross-entropy (CE) method [19]. The approximation of the failure probability p_f is evaluated from the SVR surrogate $\widetilde{G}_{s_{\max}}$ trained at the final iteration.

The paper is organized as follows. SVM regression by means of L1-ϵ-SVR is presented in Section 2. This section also describes the efficient stochastic search technique based on an estimate of the generalization error applied for obtaining accurate SVR surrogates. The adaptive strategy for reliability assessment is described in Section 3. Three challenging application problems are treated in Section 4 in order to demonstrate the efficiency of the proposed strategy. A conclusion is given in Section 5 with some perspectives.

2 SUPPORT VECTOR MACHINE SURROGATES

Support vector machines (SVM) are used as surrogate models of the LSF for reliability assessment in the present work. SVM are parts of statistical learning theory and the reader may refer to [20, 21] for a presentation of their theoretical basis. SVM possess excellent generalization performances and constructing nonlinear surrogate models is convenient thanks to the use of kernels, a property shared with Gaussian process emulators. From a given set of training pairs $(\mathbf{x}_1, y_1) \ldots, (\mathbf{x}_N, y_N)$ (training set), we want to predict a scalar output $y \in \mathcal{Y} = \mathbb{R}$ at any new point $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n$. This univariate regression problem is solved by constructing a model f such that $y = f(\mathbf{x})$. The notations in this section are those commonly used by the SVM community. In the present context, $\mathbf{x}$ stands for a point of the standard space $\mathbf{u}$, y represents the unknown output of the LSF G at a point of interest $\mathbf{u}$, $(\mathbf{x}_i, y_i)$ for $i = 1, \ldots, N$ are the given data of the training set such that $y_i = G(\mathbf{u}_i)$ and the SVR model f represents the surrogate model $\widetilde{G}_s$ which needs to be constructed at current iteration s.

2.1 Regression with SVM

The type of SVM selected in this work is based on the ϵ-insensitive loss function [22, 23]. The corresponding formulation will referred to as L1-ϵ-SVR due to the penalty term of degree one used for slack variables. For constructing the SVM model in regression, we need to solve

the following optimization problem:

$$\min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\xi}^*} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}(\xi_i+\xi_i^*) \text{ s.t. } \begin{cases} y_i - \langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle - b \leq \epsilon + \xi_i \text{ for } i=1,\ldots,N \\ \langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle + b - y_i \leq \epsilon + \xi_i^* \text{ for } i=1,\ldots,N \\ \xi_i, \xi_i^* \geq 0 \text{ for } i=1,\ldots,N \end{cases} \quad (3)$$

where $C \in \mathbb{R}^+$ is a regularization parameter to avoid overfitting, ϵ is a parameter which controls the precision level below which we apply no penalization, ξ_i and ξ_i^* are nonnegative so-called slack variables for $i = 1, \ldots, N$. $\mathbf{w} \in \mathbb{R}^n$ and $b \in \mathbb{R}$ are the usual weight vector and bias term used in SVM formulations.

Φ represents an unkwnown mapping from $\mathcal{X}$ to a high (possibly infinite) dimensional space $\mathcal{H}$ (referred to as feature space) in which the model f becomes linear: $f(\mathbf{x}) = \langle \mathbf{w}, \Phi(\mathbf{x})\rangle + b$. It is important to mention that this mapping does not require to be explicitly defined. Only the kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}, (\mathbf{x}, \mathbf{x}') \mapsto k(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}')\rangle_{\mathcal{H}}$ which computes an inner product in the feature space is required [21]. In other words, $\mathcal{H}$ is a reproducing kernel Hilbert space (RKHS) with reproducing kernel k. In the the present work, the choice is to work with the Gaussian RBF kernel:

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2\right) \quad (4)$$

where $\gamma \in \mathbb{R}^{*+}$ is a parameter that controls the bandwidth of the kernel.

Practically, the optimization problem in Eq. (3) is solved in its dual formulation:

$$\max_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*} -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i - \alpha_i^*)K_{ij}(\alpha_j - \alpha_j^*) + \sum_{i=1}^{N} y_i(\alpha_i - \alpha_i^*) - \epsilon\sum_{i=1}^{N}(\alpha_i + \alpha_i^*)$$
$$\text{s.t. } \begin{cases} \sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0 \\ 0 \leq \alpha_i \leq C \text{ for } i=1,\ldots,N \\ 0 \leq \alpha_i^* \leq C \text{ for } i=1,\ldots,N \end{cases} \quad (5)$$

where α_i, α_i^*, $i = 1, \ldots, N$, are the introduced Lagrange multipliers and $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

The optimization problem in Eq. (5) is a convex and quadratic program, which gives a unique and global minimum when feasible. The constructed SVR model is expressed as follows:

$$f(\mathbf{x}) = \left(\sum_{i \in \mathcal{S} \bigcup \mathcal{S}^*} (\alpha_i - \alpha_i^*)k(\mathbf{x}_i, \mathbf{x})\right) + b \quad (6)$$

where $\mathcal{S} \bigcup \mathcal{S}^*$ is the set of support vectors with $\mathcal{S} = \{i : 0 < \alpha_i < C\}$ and $\mathcal{S}^* = \{i : 0 < \alpha_i^* < C\}$, and where the bias term b may be conveniently obtained as a byproduct of interior point optimization.

2.2 Selection of the SVR surrogate model parameters

The parameters C and ϵ specific to L1-ϵ-SVR and the unique parameter γ of the Gaussian RBF kernel are gathered in a unique vector denoted by $\boldsymbol{\theta}$ known as hyperparameters of the SVR model. For a given training set, we need to find optimal values for these hyperparameters in order to construct the most accurate SVM surrogate model we can. The accuracy is defined here in terms of the generalization ability of the surrogate model, *i.e.* its ability to predict well over unknown points of $\mathcal{X}$.

This generalization ability is inferred in practice by cross-validation (CV) over the training set of known data. The most usual type of CV is the k-fold CV [24]. In the k-fold CV, the training set of data pairs $(\mathbf{x}_1, y_1) \ldots, (\mathbf{x}_N, y_N)$ is randomly split into k mutually exclusive subsets (called folds) of equal (or approximately equal) sizes. For a given subset, a SVR model is trained with the data of the $(k-1)$ remaining subsets and it is tested on the data of the selected subset, considered as validation data. This procedure is repeated for each of the k subsets and the average of the k obtained testing errors gives an estimate of the generalization error. In practice k is often set to 5 or 10.

A limit case is obtained for $k = N$ known as leave-one-out cross validation (LOO-CV). A SVR model $f^{(-i)}$ is trained with $(N-1)$ pairs $(\mathbf{x}_j, y_j)$, $j \in \{1, \ldots, N\} \setminus \{i\}$ and it is tested on the left-out pair $(\mathbf{x}_i, y_i)$. We consider here the following LOO error measure:

$$\mathrm{Err}_{\mathrm{LOO}}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} \left| y_i - f^{(-i)}(\mathbf{x}_i, \boldsymbol{\theta}) \right| \tag{7}$$

where $f^{(-i)}(\mathbf{x}_i, \boldsymbol{\theta})$ represents the prediction at point $\mathbf{x}_i$ of the SVR surrogate model trained on all the data pairs except the i^{th} with given hyperparameters $\boldsymbol{\theta}$.

The cost of training for assessing this error could be high for not so large N. For this reason, bounds or approximations of the LOO error are have derived to avoid true LOO-CV with SVM both in classification (see *e.g.* [25]) and in regression. The present work is based on the span bound approximations derived for L1- and L2-ϵ-SVR by Chang and Lin [18]. For L1-ϵ-SVR, this approximation reads:

$$\widetilde{\mathrm{Err}}_{\mathrm{LOO}}(\boldsymbol{\theta}) = \epsilon + \frac{1}{N} \sum_{i \in \mathcal{S}_{\mathrm{u}}} (\alpha_i + \alpha_i^*) S_i^2 + \frac{1}{N} \sum_{i \in \mathcal{S}_{\mathrm{u}}} (\xi + \xi_i^*) \tag{8}$$

where $\mathcal{S}_{\mathrm{u}} = \{i : 0 < \alpha_i + \alpha_i^* < C\}$ is the set of unbounded support vectors,
and $S_i^2 = \dfrac{1}{\left(\widetilde{\mathbf{K}}_{\mathcal{S}_{\mathrm{u}}}^{-1}\right)_{ii}}$ with $\widetilde{\mathbf{K}}_{\mathcal{S}_{\mathrm{u}}} = \begin{bmatrix} \mathbf{K}_{\mathcal{S}_{\mathrm{u}}} & \mathbf{1}_{|\mathcal{S}_{\mathrm{u}}| \times 1} \\ \mathbf{1}_{|\mathcal{S}_{\mathrm{u}}| \times 1}^{\mathrm{T}} & 0 \end{bmatrix}$ and $\mathbf{K}_{\mathcal{S}_{\mathrm{u}}} = \left(k(\mathbf{x}_i, \mathbf{x}_j)\right)_{i,j \in \mathcal{S}_{\mathrm{u}}}$.

It is important to note that the computation of this approximation of the LOO error only requires an inversion of a matrix of maximal order $(N+1) \times (N+1)$ in addition to the computational cost needed for solving the quadratic program in Eq. (5).

Starting from a given training set $(\mathbf{x}_1, y_1) \ldots, (\mathbf{x}_N, y_N)$, we want to solve the following optimization problem in order to find optimal values $\boldsymbol{\theta}^*$ for the SVR model hyperparameters:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \widetilde{\mathrm{Err}}_{\mathrm{LOO}}(\boldsymbol{\theta}) \tag{9}$$

The first technique commonly used with SVM for solving such a problem is known as grid selection. A given range is discretized for each hyperparameter most often with equally-spaced values in logscale. The combination of hyperparameter values which gives the lowest CV error (or any approximation of the LOO error) is the one which is selected for training the supposedly best SVM model. Gradient descent algorithms may also be applied as explored by Chapelle et al. [25] for SVM classification. It is however worth noting that the optimization problem in Eq. (9) is not straightforward to solve both in classification and regression. This problem is noisy and multimodal, which makes gradient descent algorithms fail. For these reasons, stochastic search algorithms have been preferred by other researchers: simulated annealing in [15, 26], particle swarm optimization method in [27, 28], comprehensive learning particle

swarm optimization combined with a BFGS algorithm in [29]). The present paper is based on the stochastic search algorithm known as the cross-entropy method (CEM) whose efficiency has been observed in several works. The proposed algorithm given in Table 1 is derived from the one proposed for continuous optimization in [19] based on sampling with normal distributions.

// **Initialize**	
$\mathbf{a} = (a_C, a_\epsilon, a_\gamma)^{\mathrm{T}}$, $\mathbf{b} = (b_C, b_\epsilon, b_\gamma)^{\mathrm{T}}$	// lower and upper bounds for $\log_{10} C$, $\log_{10} \epsilon$ and $\log_{10} \gamma$
$\widehat{\boldsymbol{\mu}}_0 = (\mathbf{a} + \mathbf{b})/2$, $\widehat{\boldsymbol{\sigma}}_0 = 100(\mathbf{b} - \mathbf{a})$	// initial mean and standard deviation
$\widetilde{\mathrm{Err}}_{\mathrm{LOO,\,min}} = +\infty$	// minimal value found for LOO error approximation
$N_{\mathrm{el}} = \lfloor \rho K \rceil$	// number of elite samples
$t = 0$	// iteration counter
do	
$t = t + 1$	
Draw $\log_{10} \boldsymbol{\theta}^{(1)}, \ldots, \log_{10} \boldsymbol{\theta}^{(K)}$ from the $\mathcal{N}_{[\mathbf{a},\mathbf{b}]}(\widehat{\boldsymbol{\mu}}_{t-1}, \widehat{\boldsymbol{\sigma}}_{t-1})$ distribution where $\widehat{\boldsymbol{\mu}}_{t-1}$ and $\widehat{\boldsymbol{\sigma}}_{t-1}$ are mean and standard deviation of the untruncated normal distribution	
for $k = 1$ **to** K	
$\left(f^{(k)} \,,\, \widetilde{\mathrm{Err}}_{\mathrm{LOO}}^{(k)} \right) =$ **train** $\left(\left(\mathbf{x}_1, y_1\right), \ldots, \left(\mathbf{x}_N, y_N\right) \,,\, \boldsymbol{\theta}^{(k)} \right)$	
end for	
$\mathcal{I} = \mathrm{sort}\left((\widetilde{\mathrm{Err}}_{\mathrm{LOO}}^{(k)})_{k=1}^{K} \,,\, \text{'ascend'} \right)$	
$\mathcal{I}_{\mathrm{el}} = \mathcal{I}_{1,\ldots,N_{\mathrm{el}}}$	// set of indices of N_{el} elite samples with lowest LOO error
$k_{\mathrm{best}} = \mathcal{I}_1$	// index of sample with lowest LOO error
if $\widetilde{\mathrm{Err}}_{\mathrm{LOO}}^{(k_{\mathrm{best}})} < \widetilde{\mathrm{Err}}_{\mathrm{LOO,\,min}}$ **then**	
$\widetilde{\mathrm{Err}}_{\mathrm{LOO,\,min}} = \widetilde{\mathrm{Err}}_{\mathrm{LOO}}^{(k_{\mathrm{best}})}$, $\boldsymbol{\theta}_{\mathrm{best}} = \boldsymbol{\theta}^{(k_{\mathrm{best}})}$	// store best solution obtained up to now
end if	
$\widetilde{\mu}_{t,j} = \dfrac{1}{N_{\mathrm{el}}} \displaystyle\sum_{k \in \mathcal{I}_{\mathrm{el}}} \theta_j^{(k)}$	// mean update
$\widetilde{\sigma}_{t,j}^2 = \dfrac{1}{N_{\mathrm{el}}} \displaystyle\sum_{k \in \mathcal{I}_{\mathrm{el}}} \left(\theta_j^{(k)} - \widetilde{\mu}_{t,j} \right)^2$	// variance update
$\widehat{\boldsymbol{\mu}}_t = \alpha \widetilde{\boldsymbol{\mu}}_t + (1 - \alpha) \boldsymbol{\theta}^{(k_{\mathrm{best}})}$	// static smoothing of mean
$\widehat{\boldsymbol{\sigma}}_t = \alpha \widetilde{\boldsymbol{\sigma}}_t + (1 - \alpha) \boldsymbol{\sigma}_{t-1}$	// static smoothing of standard deviation
while $(\widehat{\sigma}_{t,\,1} > \sigma_{\mathrm{th},\,C})$ **or** $(\widehat{\sigma}_{t,\,2} > \sigma_{\mathrm{th},\,\epsilon})$ **or** $(\widehat{\sigma}_{t,\,3} > \sigma_{\mathrm{th},\,\gamma})$	
$\boldsymbol{\theta}^* = \boldsymbol{\theta}_{\mathrm{best}}$	// optimal values for hyperparameters

Note: $\lfloor x \rceil$ stands for the nearest integer to x.

Table 1: Pseudo-code of the CE algorithm for optimal hyperparameters selection.

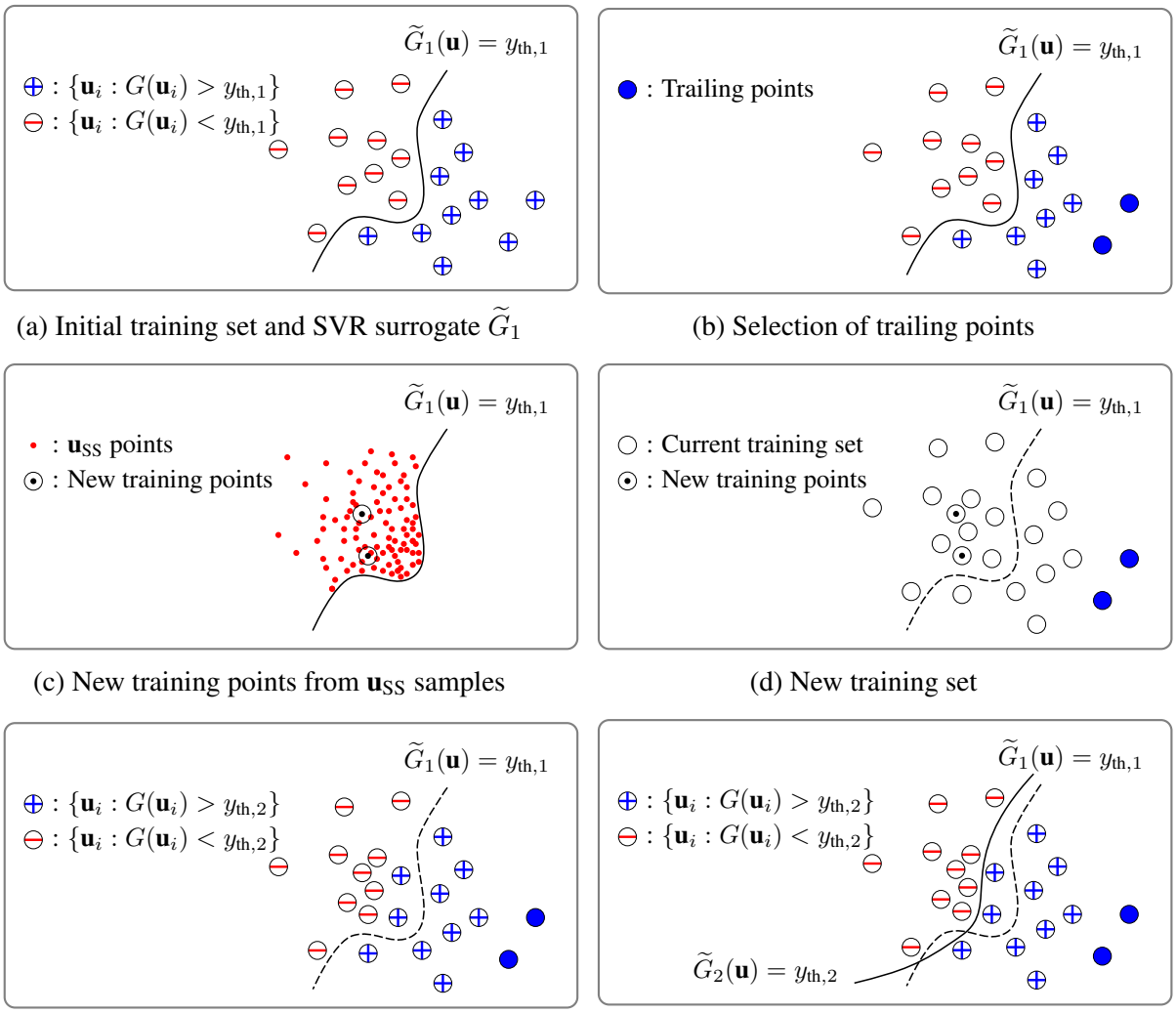
3 RELIABILITY ASSESSMENT WITH ADAPTIVE SVM SURROGATES

The method proposed in the present paper consists in constructing a sequence of SVR surrogate models $\widetilde{G}_s(\mathbf{u})$, $s = 1, \ldots, s_{\max}$, in the standard space. The objective of such an adaptive strategy is that the accuracy of the SVR surrogate close to the limit-state surface (LSS) increases with iteration s. The main idea is to generate new training points which become closer to the failure domain $\mathcal{F}_{\mathbf{u}}$ with iteration s by exploiting the information conveyed by the currently constructed SVR surrogate. It is important to note that generating such points a priori with no information about the location of the failure domain is unfeasible which justify the development of adaptive SVR surrogate models.

The choice made here is to work with SVM surrogate model used in regression and based on the ϵ-insensitive loss function (L1-ϵ-SVR). In reliability assessment, we basically want to know whether points $\mathbf{u}$ of the standard space are in the failure domain $\mathcal{F}_{\mathbf{u}}$ or not. This binary

classification problem have been for this reason explored in several works by means of SVM used in classification, where the two classes represent the safe and failure domains. Knowing the class of a point requires a call to the LSF and it is the author's belief that the value of the LSF brings more information than juts its sign for an accurate approximation of the LSS.

At current iteration s, a SVR surrogate model needs to be trained. The quadratic program in Eq. (5) is solved by means of an interior point method (see *e.g.* [30]). Optimal values for the SVR surrogate hyperparameters are obtained by minimizing the span bound approximation of the LOO error of Chang and Lin [18] with the CE algorithm detailed in Table 1.



(a) Initial training set and SVR surrogate $\widetilde{G}_1$

(b) Selection of trailing points

(c) New training points from $\mathbf{u}_{\text{SS}}$ samples

(d) New training set

(e) New threshold value $y_{\text{th},2}$

(f) Updated SVR surrogate $\widetilde{G}_2$

Figure 1: Main steps of the proposed algorithm

The conceptual ideas of the proposed method are depicted in Figure 1. The algorithm proceeds as follows:

- Starting from an initial training set composed of N points $\mathbf{u}_i$ sampled from the n-dimensional standard normal joint pdf φ_n and their corresponding LSF evaluations $y_i = G(\mathbf{u}_i)$, a first SVR surrogate $\widetilde{G}_1$ is trained. We also define the first intermediate threshold level $y_{\text{th},1}$ as the median of the $(y_i)_{i=1}^N$ sample and we obtain the SVR-based surface corresponding to this level: $\{\mathbf{u} : \widetilde{G}_1(\mathbf{u}) = y_{\text{th},1}\}$. See subplot 1a of Figure 1.

- The $p_{\text{a}}N$ points $\mathbf{u}_i$ of the training set with the largest y_i LSF values are withdrawn from the training set and will therefore not be used for the next SVR training. These points will be part of a set called the trailing set. See subplot 1b of Figure 1.
- Subset simulation is then used with the following intermediate LSF: $\mathbf{u} \mapsto \widetilde{G}_1(\mathbf{u}) - y_{\text{th},1}$. The probability obtained by subset simulation with this intermediate LSF is denoted by p_s. From the samples generated by the modified Metropolis-Hastings during the last subset simulation step for which $\widetilde{G}_1(\mathbf{u}) < y_{\text{th},1}$, we randomly select $p_{\text{a}}N$ points which will become new training points. See subplot 1c of Figure 1. These $p_{\text{a}}N$ points are now added to the training set whose size becomes again N. See subplot 1d of Figure 1.
- A new intermediate threshold level $y_{\text{th},2}$ is defined as the median of the LSF outputs of points of the training set. See subplot 1e of Figure 1.
- A new SVR surrogate $\widetilde{G}_2$ is trained from the newly composed training set, which defines an updated SVR-based surface with respect to the $y_{\text{th},2}$ level: $\{\mathbf{u} : \widetilde{G}_2(\mathbf{u}) = y_{\text{th},2}\}$. See subplot 1f of Figure 1.
- Points of the trailing set are now used for testing. If for any point $\mathbf{u}_i$ of the trailing set we have $\widetilde{G}_2(\mathbf{u}_i) < y_{\text{th},2}$, this point becomes again part of the training set.
- The previous process is repeated until we find $y_{\text{th},s} < 0$. We then impose $y_{\text{th},s} = 0$ for subsequent iterations s.
- New points are added close to the SVR-based surface $\{\mathbf{u} : \widetilde{G}_s(\mathbf{u}) = 0\}$ in order to increase the accuracy of the constructed SVR surrogate with respect to the true LSS.
- The approximation of the failure probability p_{f} is evaluated from the SVR surrogate $\widetilde{G}_{s_{\text{max}}}$ trained at the final iteration: $p_{\text{f}} = p_{s_{\text{max}}} = \mathbb{P}(\widetilde{G}_{s_{\text{max}}}(\mathbf{U}) \leq 0)$.

4 APPLICATION EXAMPLES

The application examples are taken from Bourinet [12] where reference results are also provided. The ratio p_{a} over N of new points to add at each iteration has been set to $1/10$. For the first two examples, we consider both $N = 30$ and $N = 50$ cases. For the high dimensional problem of example 3, we take $N = 50$. The algorithm is run 20 times in order to assess the bias of the estimated failure probability with respect to the reference one and also its coefficients of variation. It is worth mentioning that the failure probability assessed with the proposed algorithm is random due to the randomness introduced in the selection of training points and also the one coming from the use of subset simulation which results in statistical variations of the computed failure probability. The evolution of p_s with the cumulative number of calls to the LSF N_s is plotted for each application example. Blue disks corresponds to the iteration at which $y_{\text{th},s}$ gets negative and red squares to the final iteration.

4.1 Example 1: parallel system

The first example is a 5-random variable parallel system reliability problem studied in [31, 32]. All the five random variables have independent standard normal distributions. The LSF is expressed as follows:

$$G(\mathbf{u}) = \max\{2.677 - u_1 - u_2\ ;\ 2.500 - u_2 - u_3\ ; 2.323 - u_3 - u_4\ ;\ 2.250 - u_4 - u_5\ \} \quad (10)$$

A reference solution is given by $p_{\text{f ref}} = \Phi_4(-\boldsymbol{\beta}, \mathbf{0}, \mathbf{R}) = 2.13 \times 10^{-4}$ where $\boldsymbol{\beta}$ is the vector of HL reliability indices, $\mathbf{R}$ is the correlation matrix between the four LSS and Φ_4 is the 4-dimensional normal cumulative distribution function. The p_s v.s. N_s plots are given in Figure 2. The failure probability estimate averaged over 20 runs of the proposed method is 1.96×10^{-4}

and 1.99×10^{-4}, with a coefficient of variation of 7.6% and 2.8%, respectively for $N = 30$ and $N = 50$. The respective averaged total numbers of calls to the LSF are 306 and 580. The 7-8% bias observed on the failure probabilities come from the selected kernel which appears unable to perfectly fit the geometry of the LSS.

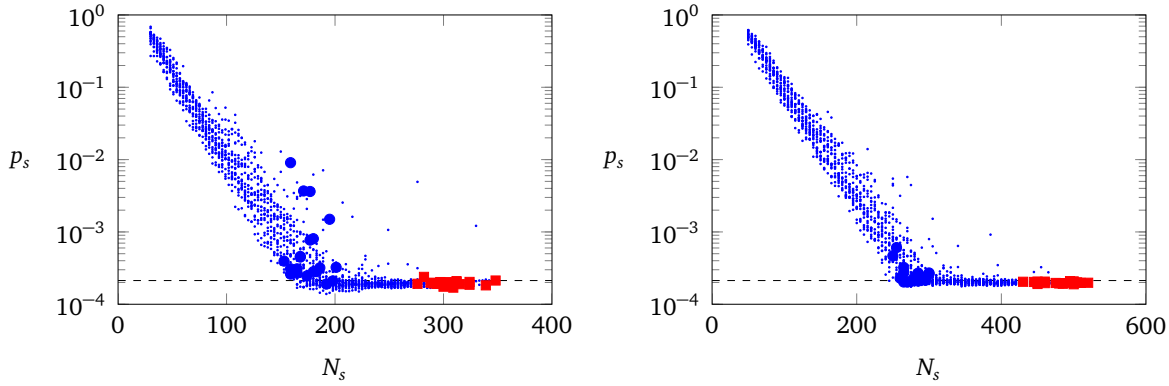


Figure 2: Results of example 1 - $N = 30$ (left), $N = 50$ (right)

4.2 Example 2: two d.o.f. primary/secondary damped oscillator

The second example was initially proposed in the report by De Stefano and Der Kiureghian [33] and used as a benchmark by Bourinet et al. [11]. We are interested in the reliability assessment of a two-degree-of-freedom primary-secondary system under a white noise base acceleration. The uncertainty is modeled by 8 independent random variables. The reliability problem is characterized by a single MPP, with a strongly curved LSS and a very low failure probability. The mean value of the force capacity of the secondary spring F_s is set to 27.5, which results in a reference failure probability $p_{\text{f ref}} = 3.78 \times 10^{-7}$ [11]). The p_s v.s. N_s plots are given in Figure 3. The failure probability estimate averaged over 20 runs of the proposed method is 3.69×10^{-7} and 3.76×10^{-7}, with a coefficient of variation of 6.3% and 1.6%, respectively for $N = 30$ and $N = 50$. The respective averaged total numbers of calls to the LSF are 503 and 606. The bias observed on the failure probability is negligible, which shows that the Gaussian RBF kernel is able to capture quite well the strong curvatures of the LSS at the MPP.

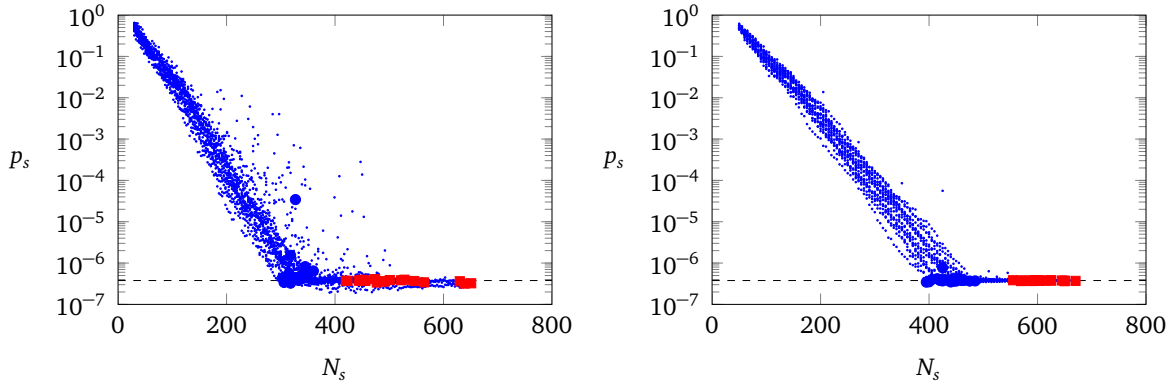


Figure 3: Results of example 2 - $N = 30$ (left), $N = 50$ (right))

4.3 Example 3: smooth equally curved high dimensional example

The last example is a 100-random variable reliability problem with a single MPP and moderate (and equal) curvatures at the MPP [34]. The LSF is given by:

$$g(\mathbf{x}) = \left(n + a\,\sigma\sqrt{n}\right) - \sum_{i=1}^{n} x_i \tag{11}$$

where $X_i,\ i = 1, \ldots, n$, are i.i.d. lognormal random variables, with unit means and the same standard deviation σ equal to 0.2. The reference failure probability is $p_{\text{f ref}} = 1.73 \times 10^{-3}$ with $a = 3$ [11]. The p_s v.s. N_s plot with $N = 50$ is given in Figure 4. The failure probability estimate averaged over 20 runs is 1.69×10^{-3} with a coefficient of variation of 1.1%. The averaged total numbers of calls to the LSF is 559. The proposed method appears quite efficient and accurate on this high-dimensional but smooth LSS.

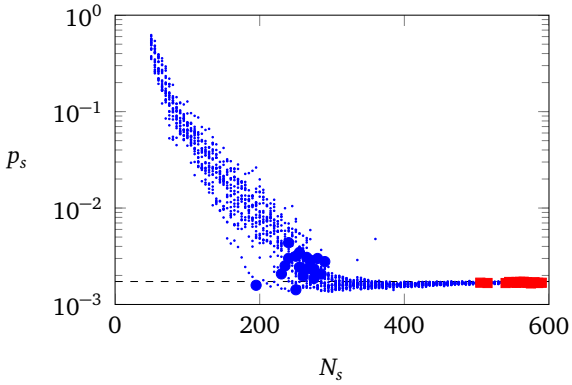


Figure 4: Results of example 3 - $N = 50$

5 CONCLUSION-PERSPECTIVES

The proposed method is able to estimate low failure probabilities quite efficiently (a few hundreds calls to the LSF) with a high precision (coefficient of variation less than 3% for $N = 50$ in the treated examples). The limits are those of the kernel assumption. For the Gaussian RBF kernel, the LSF is required to be smooth enough. This performance has a price to pay: the proposed method requires a substantial effort for training the sequentially constructed SVR surrogate models even though this task could be easily parallelized. One choice made here is to learn a surrogate model over a subset of the already known data, which could be viewed as a local kernel regression triggered towards the failure domain. The main advantages are twofold: 1) the training time is lowered due to the reduced size of the training set and 2) we can use the left out points (trailing set) for testing. Another choice made in the proposed method is to pick up random samples as training points. We could of course expect some gains if we were using some distance-based criteria but this advantage would vanish in high dimensional spaces.

Regarding the perspectives, some ongoing works are undertaken in order to evaluate and improve the robustness of the proposed method on reliability assessment problems characterized by multiple MPP of equal or similar importances. More generally future works will investigate the applicability of the method to high dimensional problems involving random fields or random processes as model inputs within the limit of a very few hundreds of random variables.

REFERENCES

[1] O. Ditlevsen, H. Madsen. *Structural reliability methods*. Internet Edition 2.3.7, 2007.

[2] T. Homem-de-Mello, R.Y. Rubinstein. *Rare event estimation for static models via cross-entropy and importance sampling*. 2002.

[3] H. Cancela, M. El Khadiri, G. Rubino. Rare event analysis by Monte Carlo techniques in static models. In: *Rare Event Simulation using Monte Carlo Methods*. Ed. by G. Rubino, B. Tuffin, 145–170. John Wiley & Sons, 2009.

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

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

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

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

[8] T. Most. An adaptive response surface approach for structural reliability analyses based on support vector machines. Ed. by B.H.V. Topping. Civil-Comp Press. In: *Proc. 11th International Conference on Civil, Structural and Environmental Engineering Computing, St. Julians, Malta, September 18–21, 2007*.

[9] X. Yuan, Z. Lu, Y. Lu. Support vector machine response surface method based on fast Markov chain simulation. IEEE. In: *Proc. 2009 IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS 2009), Shanghai, China, November 20–22, 2009*.

[10] A. Basudhar, S. Missoum. An improved adaptive sampling scheme for the construction of explicit boundaries. *Structural and Multidisciplinary Optimization*, **42**(4), 517–529, 2010.

[11] J.-M. Bourinet, F. Deheeger, M. Lemaire. Assessing small failure probabilities by combined subset simulation and support vector machines. *Structural Safety*, **33**(6), 343–353, 2011.

[12] J.-M. Bourinet. Reliability assessment with sensitivity-based adaptive SVM surrogates. Ed. by G. Deodatis, B.R. Ellingwood, D.M. Frangopol. CRC Press. In: *Proc. 11th International Conference on Structural Safety and Reliability (ICOSSAR 2013), New York, NY, USA, June 16–20, 2013*.

[13] G. Blatman, B. Sudret. An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis. *Probabilistic Engineering Mechanics*, **25**(2), 183–197, 2010.

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

[15] P.-F. Pai, W.-C. Hong. Software reliability forecasting by support vector machines with simulated annealing algorithms. *Journal of Systems and Software*, **79**(6), 747–755, 2006.

[16] K.-Y. Chen. Forecasting systems reliability based on support vector regression with genetic algorithms. *Reliability Engineering & System Safety*, **92**(4), 423–432, 2007.

[17] H. Dai, H. Zhang, W. Wang, G. Xue. Structural reliability assessment by local approximation of limit state functions using adaptive Markov chain simulation and support vec-

tor regression. *Computer-Aided Civil and Infrastructure Engineering*, **27**(9), 676–686, 2012.

[18] M.-W. Chang, C.-J. Lin. Leave-one-out bounds for support vector regression model selection. *Neural Computation*, **17**(5), 1188–1222, 2005.

[19] Z.I. Botev, D.P. Kroese, R.Y. Rubinstein, P. L'Ecuyer. Chapter 3 - The cross-entropy method for optimization. In: *Handbook of Statistics - Machine Learning: Theory and Applications*. Ed. by C.R. Rao, V. Govindaraju. Vol. 31. Handbook of Statistics, 35–59. Elsevier, 2013.

[20] B.E. Boser, I.M. Guyon, V.N. Vapnik. A training algorithm for optimal margin classifiers. ACM Press. In: *Proc. 5th Annual Workshop on Computational Learning Theory (COLT'92), Pittsburgh, PA, USA, July 27–29, 1992*.

[21] V. Vapnik. *The nature of statistical learning theory*. New York, NY, USA: Springer-Verlag, 1995.

[22] V. Vapnik, S.E. Golowich, A.J. Smola. Support vector method for function approximation, regression estimation and signal processing. In: *Advances in Neural Information Processing Systems 9*. Ed. by M.C. Mozer, M.I. Jordan, T. Petsche, 281–287. MIT Press, 1997.

[23] A.J. Smola, B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, **14**(3), 199–222, 2004.

[24] S. Geisser. The predictive sample reuse method with applications. *Journal of the American Statistical Association*, **70**(350), 320–328, 1975.

[25] O. Chapelle, V. Vapnik, O. Bousquet, S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, **46**(1–3), 131–159, 2002.

[26] S.-W. Lin, Z.-J. Lee, S.-C. Chen, T.-Y. Tseng. Parameter determination of support vector machine and feature selection using simulated annealing approach. *Applied Soft Computing*, **8**(4), 1505–1512, 2008.

[27] S.-W. Lin, K.-C. Ying, S.-C. Chen, Z.-J. Lee. Particle swarm optimization for parameter determination and feature selection of support vector machines. *Expert Systems with Applications*, **35**(4), 1817–1824, 2008.

[28] S.-W. Fei, M.-J. Wang, Y.-B. Miao, J. Tu, C.-L. Liu. Particle swarm optimization-based support vector machine for forecasting dissolved gases content in power transformer oil. *Energy Conversion and Management*, **50**(6), 1604–1609, 2009.

[29] S. Li, M. Tan. Tuning SVM parameters by using a hybrid CLPSO-BFGS algorithm. *Neurocomputing*, **73**(10–12). Subspace Learning / Selected papers from the European Symposium on Time Series Prediction, 2089–2096, 2010.

[30] Y. Nesterov, A. Nemirovskii. 3rd ed. Wiley Series in Probability and Statistics. New York, NY, USA: John Wiley & Sons, 2009.

[31] P. Bjerager. Probability integration by directional simulation. *Journal of Engineering Mechanics*, **114**(8), 1285–1302, 1988.

[32] S. Engelund, R. Rackwitz. A benchmark study on importance sampling techniques in structural reliability. *Structural Safety*, **12**(4), 255–276, 1993.

[33] M. De Stefano, A. Der Kiureghian. *An efficient algorithm for second-order reliability analysis*. Report No. UCB/SEMM-90/20. Dept of Civil and Environmental Engineering, University of California, Berkeley, 1990.

[34] R. Rackwitz. Reliability analysis – a review and some perspectives. *Structural Safety*, **23**(4), 365–395, 2001.