Structural Engineering and Mechanics, *Vol. 56, No. 4 (2015) 549-567*
DOI: http://dx.doi.org/10.12989/sem.2015.56.4.549

A Gaussian process-based response surface method for structural reliability analysis

Guoshao Su, Jianqing Jiang*, Bo Yu and Yilong Xiao

School of Civil and Architecture Engineering, Key Laboratory of Disaster Prevention and Structural Safety of Ministry of Education, Guangxi Key Laboratory of Disaster Prevention and Engineering Safety, Guangxi University, Nanning 530004, China

(Received May 13, 2014, Revised November 2, 2015, Accepted November 3, 2015)

Abstract. A first-order moment method (FORM) reliability analysis is commonly used for structural stability analysis. It requires the values and partial derivatives of the performance to function with respect to the random variables for the design. These calculations can be cumbersome when the performance functions are implicit. A Gaussian process (GP)-based response surface is adopted in this study to approximate the limit state function. By using a trained GP model, a large number of values and partial derivatives of the performance functions can be obtained for conventional reliability analysis with a FORM, thereby reducing the number of stability analysis calculations. This dynamic renewed knowledge source can provide great assistance in improving the predictive capacity of GP during the iterative process, particularly from the view of machine learning. An iterative algorithm is therefore proposed to improve the precision of GP approximation around the design point by constantly adding new design points to the initial training set. Examples are provided to illustrate the GP-based response surface for both structural and non-structural reliability analyses. The results show that the proposed approach is applicable to structural reliability analyses that involve implicit performance functions and structural response evaluations that entail time-consuming finite element analyses.

Keywords: first-order moment method; structural reliability; response surface method; Gaussian process

1. Introduction

In a structural reliability analysis, the fundamental task is finding a solution to the multi-fold integral representing the probability of failure, denoted as P_f, and defined by

$$P_f = P[g(\boldsymbol{X}) \le 0] = \int_{g(\boldsymbol{X})} f(\boldsymbol{X})\, d\boldsymbol{X} \tag{1}$$

where $f(\boldsymbol{X})$ is the joint probability density function of the vector of basic random variables $\boldsymbol{X}$, which represent uncertain quantities such as material properties, loads, material parameters and geometry. The limit state function $g(\boldsymbol{X})=0$ divides the design space into two regions: a safe region for which $g(\boldsymbol{X})>0$ and a failure region for which $g(\boldsymbol{X})<0$. Commonly used probabilistic analysis

*Corresponding author, Ph.D. Candidate, E-mail: jiangqing880201@163.com

http://www.techno-press.org/?journal=sem&subpage=8 ISSN: 1225-4568 (Print), 1598-6217 (Online)

methods are based on either simulation techniques such as Monte Carlo simulations (MCS) or moment-based methods such as the first-order reliability methods (FORM) or second-order reliability methods (SORM).

An MCS can be used to solve any complex problem of implicit or explicit form for which accurate solutions are either impossible or extremely difficult to obtain through an analytical technique. For example, a limit state involving multiple random variables that is evaluated by a finite element method (FEM) can be easily set up for solution with the Monte Carlo Simulation technique. However, this becomes computationally excessive when the probability of failure Pf is extremely low. An MCS is typically used as a benchmark for verifying the accuracy and comparing the efficiency of other methods that use approximation concepts.

Moment-based methods such as FORM and SORM can drastically reduce computational costs compared to an MCS. Unfortunately, these methods require the evaluation of the derivatives of performance functions with respect to the random variables. However, the performance functions are usually implicit in practical problems, which results in derivatives of the performance functions that are not readily available.

Response surface approximations (RSM) (Bucher and Bourgund 1990, Malur and Bruce 1993, Kim and Na 1997, Guan and Melchers 2001, Zhao *et al*. 2013, Jiang *et al*. 2014) can be used to obtain a closed-form approximation to the limit state function to facilitate a reliability analysis. Response surface approximations typically fit low-order polynomials to the structural response in terms of random variables. The probability of failure can then be inexpensively calculated through a Monte Carlo simulation or FORM and SORM using the fitted polynomials. Therefore, an RSM is a particularly attractive option for computationally expensive problems such as those requiring complex FEM analyses. The most popularly used response surface function is the quadratic polynomial function. However, the quadratic polynomial cannot provide sufficiently accurate approximations of performance functions in cases with high nonlinearity (Christian and Thomas 2008).

Due to recent developments in artificial intelligence research, artificial neural networks (ANNs) and support vector machines (SVMs) are now used to approximate limit state functions. Various studies have shown that ANN-based and SVM-based RSMs are more efficient and accurate than polynomial-based RSMs (Rocco and Moreno 2002, Luc and Dionys 2005, Deng *et al*. 2005, Li and Yue 2006, Hurtado 2007, Cheng and Xia 2008, Zhao 2008). However, ANNs exhibit difficulties in determining appropriate network topologies and sizes, while SVMs cannot avoid the common phenomenon of blindness in the man-made choices in the hyperparameters of kernel function (Vapnik 1998).. This demonstrates a considerable need to develop efficient frameworks for reliability analyses.

A Gaussian process (GP) is a newly developed machine learning technology based on strict theoretical fundamentals and Bayesian theory (MacKay 1998). In recent years, GPs have attracted significant attention in the machine learning community (Williams 1998, Rasmusen and Williams 2006, Su *et al.* 2007, Chen and Martin 2007, Hensman *et al.* 2010). Similar to an ANN and an SVM, a GP can approximate any function. Its major advantage over an ANN is its simplicity: no network size or topology need be selected. In contrast to the weights of an ANN, the hyperparameters of a GP have intrinsic meaning. A further advantage of a GP over an ANN and SVM is that it possesses a theoretical framework for obtaining the optimum hyperparameters in a self-adaptive manner.

This paper proposes a GP-based RSM method of reliability analysis that combines a GP with the frameworks of a FORM and an RSM. Section 2 reviews the FORM method for structural

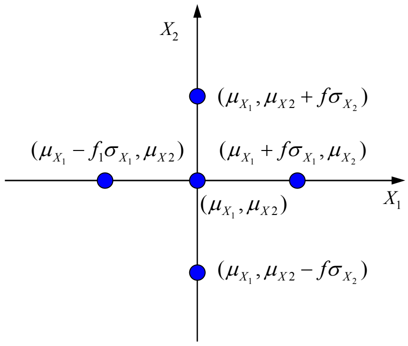

Fig. 1 Location of training datasets for a two-variable problem

reliability analysis and presents the theories and procedure of the proposed method. Section 3 utilizes four numerical examples to illustrate the application and effectiveness of the proposed method. Section 4 discusses several important parameters in the proposed method. Conclusions are drawn in Section 5.

2. The proposed method

2.1 Numerical simulation procedure

This study initially prepares a set of input and output data for establishing the GP model. According to the experimental plan of RSM developed by Bucher and Bourgand (1990), these training datasets are generated along the axis $\boldsymbol{x}$ with coordinates of $x_i = \mu_i \pm f\sigma_i$, where μ_i and σ_i are the mean and standard deviation of the random variable x_i (see Fig. 1). The values of the performance function $y(x_1, x_2, \ldots, x_n)$ and $y(x_1, x_2, \ldots, x_i \pm f\sigma_i, \ldots, x_n)$ are obtained by using a structural analysis code. The initial number of training datasets m is $2n \times s+1$, where f=[1,4] and s is the number of selected f. This paper adopts f=2, s =1, and m=2n+1 as default values. The initial training datasets $\boldsymbol{D}$ of m observations are thus obtained as $\boldsymbol{D}=\{(\boldsymbol{x}_i, \boldsymbol{y}i) \mid i=1,\ldots m\}$, where $\boldsymbol{x}_i$ denotes an input vector and y_i denotes a scalar output or target. The column vector inputs for all m cases are aggregated in the $n \times m$ design matrix $\boldsymbol{X}$, and the targets are collected in the vector $\boldsymbol{y}$.

2.2 Processing of training data

To improve the stability of the GP training process and the degree of generalization accuracy, all of the training datasets must be scaled before presenting them to the model. The following scaling equation is used

$$\boldsymbol{x}^s = \frac{\boldsymbol{x}}{\boldsymbol{\sigma}_X} \tag{2}$$

$$\boldsymbol{y}^s = \boldsymbol{y} - \boldsymbol{\mu}_y \tag{3}$$

where $\boldsymbol{x}^s$ and $\boldsymbol{y}^s$ are the scaled values of the training set, $\boldsymbol{\sigma}_X$ is the standard deviation of the random variables, and $\boldsymbol{\mu}_y$ is the mean of the training dataset outputs.

2.3 Approximation of the limit state function by trained GP

2.3.1 Training a GP

A GP specifies a probabilistic model over a given set of data points and is constructed so that the likelihood of the function value, given the decision variable values, is maximized for all data points. This model can then be extended to predict the mean and standard deviation of the function value at new data points. GPs have a small number of hyperparameters that can be optimized through a maximum likelihood approach. In the following, we present the mail equation for a GP; for more details, refer to the works of Rasmussen and Williams (2006).

A GP is a collection of random variables for which any finite set has a joint Gaussian distribution. A Gaussian Process is completely specified by its mean function $m(\boldsymbol{x})$ and covariance function $k(x, x')$

$$f(\boldsymbol{x}) \sim GP(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')) \tag{4}$$

The goal of Bayesian forecasting is to compute the distribution $p(y_*|\boldsymbol{x}_*, \boldsymbol{D})$ of output y_* given a test input $\boldsymbol{x}_*$ and a training dataset $\boldsymbol{D} = \{(\boldsymbol{x}_i^s, y_i^s) \mid i = 1, \ldots m\}$. The posterior distribution for the Gaussian Process outputs y_* can be obtained by using a Bayesian rule. Through conditioning the observed targets in the training dataset, the predictive distribution is thus Gaussian

$$y_* \left| \boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y} \sim N(\hat{y}(\boldsymbol{x}_*), \hat{\sigma}(\boldsymbol{x}_*)) \right. \tag{5}$$

where the mean and variance are given by

$$\hat{\boldsymbol{y}}(x_*) = k_*^{\boldsymbol{T}} (K + \boldsymbol{\sigma}_n^2 I)^{-1} y \tag{6}$$

$$\hat{\sigma}^2(\boldsymbol{x}_*) = k(\boldsymbol{x}_*, \boldsymbol{x}_*) - \boldsymbol{k}_*^T (\boldsymbol{K} + \sigma_n^2 \boldsymbol{I})^{-1} \boldsymbol{k}_*^T \tag{7}$$

where compact forms of the notation setting for the matrix of the covariance functions are: $\boldsymbol{k}_*=\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{x}_*)$, $\boldsymbol{K}=\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})$; σ_n^2 is the unknown variance of the Gaussian noise.

Assuming that $\boldsymbol{\alpha} = (\boldsymbol{K} + \sigma_n^2 \boldsymbol{I})^{-1} \boldsymbol{y}$, Eq. (6) can be used to display a linear combination of m covariance functions, each one centered on a training point, by writing

$$\hat{y}(\boldsymbol{x}_*) = \sum_{j=1}^{m} \boldsymbol{\alpha k}(\boldsymbol{x}_j, \boldsymbol{x}_*) \tag{8}$$

We initially train the GP model by learning the training datasets $\boldsymbol{D}$. We then make predictions of the performance function on the design point $\boldsymbol{x}^*$ and obtain the predictions $\hat{y}(x^*)$ according Eq. (8). Recall that the training outputs were scaled; we must therefore adjust the predictions of the performance function to approximate the limit state function $g(\boldsymbol{x}^*)$

$$g(\boldsymbol{x}^*) \approx \bar{y}(\boldsymbol{x}^*) = \hat{y}(\boldsymbol{x}^*) + \mu_y = \sum_{j=1}^{m} \boldsymbol{\alpha}_j \boldsymbol{k}(\boldsymbol{x}_j, \boldsymbol{x}^*) + \mu_y \tag{9}$$

There are numerous choices for prior covariance functions. From a modeling point of view, the objective is to specify prior covariances that contain our prior beliefs about the structure of the function that we are modeling. A Gaussian Process procedure can handle varied models by simply using the following general covariance functions:

(1) Squared exponential covariance function

$$k(x_p, x_q) = \sigma_f^2 \exp\left(-\frac{\left| x_p - x_q \right|^2}{2l^2} \right) + \sigma_n^2 \delta_{pq} \tag{10}$$

(2) Linear covariance function

$$k(x_p, x_q) = \frac{x_p x_q}{l^2} + \sigma_n^2 \delta_{pq} \tag{11}$$

(3) Matérn covariance function

$$k(x_p, x_q) = \sigma_f^2 \left(1 + \frac{\sqrt{3}\left| x_p - x_q \right|}{l} \right) \exp\left(-\frac{\sqrt{3}\left| x_p - x_q \right|}{l} \right) + \sigma_n^2 \delta_{pq} \tag{12}$$

where l is the *characteristic length-scale*, σ_f^2 is the *signal variance*, and δ_{pq} is a Kronecker delta. It is noted that the squared exponential covariance function and the linear covariance are both covariance functions within an Automatic Relevance Determination (ARD) (Rasmusen *et al.* 2006) distance measure. The component of l in different dimensions is not same; thus, the hyperparameters of the squared exponential covariance function are $\boldsymbol{\theta}=(l_1, l_2, \ldots, l_n, \sigma_f, \sigma_n)$ and the hyperparameters of the linear covariance function are $\boldsymbol{\theta}=(l_1, l_2, \ldots, l_n, \sigma_n)$. The Matérn covariance function is a covariance function that utilizes an isotropic distance measure; the component of l in different dimensions is the same, and the hyperparameters of the Matérn covariance function are $\boldsymbol{\theta}=(l, \sigma_f, \sigma_n)$.

The hyperparameters $\boldsymbol{\theta}$ can be optimized, based on the following log-likelihood framework

$$L = \log p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\theta}) = -\frac{1}{2} \boldsymbol{y}^T \boldsymbol{C}^{-1} \boldsymbol{y} - \frac{1}{2} \log \det \boldsymbol{C} - \frac{n}{2} \log 2\pi \tag{13}$$

The log-likelihood and its derivative with respect to $\boldsymbol{\theta}$ can be expressed as

$$\frac{\partial L}{\partial \boldsymbol{\theta}} = -\frac{1}{2} tr\left(\boldsymbol{C}^{-1} \frac{\partial \boldsymbol{C}}{\partial \boldsymbol{\theta}} \right) + \frac{1}{2} \boldsymbol{y}^T \boldsymbol{C}^{-1} \frac{\partial \boldsymbol{C}}{\partial \boldsymbol{\theta}} \boldsymbol{C}^{-1} \boldsymbol{y} \tag{14}$$

where $\boldsymbol{C} = \boldsymbol{K} + \sigma_n^2 \boldsymbol{I}$.

The hyperparameters $\boldsymbol{\theta}$ are initialized to random values in a reasonable range and then use an iterative method such as a conjugate gradient to search for the optimal values. We found that this approach can be susceptible to local minima. To overcome this drawback, we randomly selected a number of starting positions within a hyperparameter space.

2.3.2 Evaluation of the GP performance

Once the GP model is trained, the relationship between the limit state function and the various design variables can be readily retrieved by using the model. The next step involves validating and evaluating the trained model, which can be achieved by using common error metrics such as maximum absolute error (MAE) or root-mean-squared error (RMSE) methods. These two error functions can be expressed as

$$\mathrm{MAE} = \max\left| \bar{y}_i - t_i \right| \quad i = 1, \cdots, m \tag{15}$$

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{m}(\bar{y}_i - t_i)^2}{m}} \tag{16}$$

where t is the value of true limit state function.

2.3.3 Explicit formation of reliability index using trained GP

A limit state surface is given as

$$Z = g(\boldsymbol{X}) = 0 \tag{17}$$

where $\boldsymbol{X}$ represents the random variables with $\boldsymbol{\mu_X}$ mean and $\boldsymbol{\sigma_X}$ standard deviations. This assumes there is an initial design point $\boldsymbol{x}^*=(x^*_1, x^*_2,\ldots,x^*_n)^T$ of the limit state surface. While g($\boldsymbol{X}$) is generally a nonlinear function, it can be linearized at $\boldsymbol{x}^*$ by neglecting second order terms

$$Z_Q = g(\boldsymbol{x}^*) + \sum_{i=1}^{n} \frac{\partial g(\boldsymbol{x}^*)}{\partial X_i}\left(X_i - x_i^*\right) \tag{18}$$

We can then compute the mean μ_{X_i} and standard deviation σ_{X_i} at the point $\boldsymbol{x}^*$ of the equivalent normal distribution for those variables that are non-normal by using the Rackwitz and Fiessler method (1976).

$$\sigma_{X_i} = \frac{\varphi\,\{\Phi^{-1}[F_{X_i}(x_i^*)]\}}{f_{X_i}(x_i^*)} \tag{19}$$

$$\mu_{X_i} = x_i^* - \phi^{-1}\left[F_{X_i}(x_i^*)\right]\sigma_{X_i} \tag{20}$$

where Φ^{-1} is the inverse cumulative distribution function (CDF) of the standard normal variable, $F_{X_i}(x_i^*)$ is the CDF of the original non-normal variables, φ and $f_{X_i}(x_i^*)$ are the probability density functions (PDFs) of the equivalent standard normal and the original non-normal random variable, respectively.

Assuming that $\boldsymbol{X}$ is statistically uncorrelated, the reliability index is given as

$$\beta = \frac{\mu_{Z_Q}}{\sigma_{Z_Q}} = \frac{g(\boldsymbol{x}^*) + \sum_{i=1}^{n} \frac{\partial g(\boldsymbol{x}^*)}{\partial X_i}(\mu_{X_i} - x_i^*)}{\sqrt{\sum_{i=1}^{n}\left[\frac{\partial g(\boldsymbol{x}^*)}{\partial X_i}\right]^2 \sigma_{X_i}^2}} \tag{21}$$

According to the FORM method, a tangent hyperplane is fitted to the limit state surface at its "most probable failure point", i.e., the design point. Accordingly, the reliability index β can be defined by the minimum distance from the origin to the design point; the coordinate of design point $\boldsymbol{x}^*$ is denoted as

$$x_i^* = \mu_{x_i} + \beta\sigma_{x_i}\cos\theta_{x_i},\ i = 1,2,\cdots,n \tag{22}$$

where

$$\cos\,\theta_{X_i} = -\frac{\dfrac{\partial g(\boldsymbol{x}^*)}{\partial X_i}\sigma_{X_i}}{\sqrt{\sum_{i=1}^{n}\left[\dfrac{\partial g(\boldsymbol{x}^*)}{\partial X_i}\right]^2 \sigma_{X_i}^2}} \tag{23}$$

In the reliability analysis of complex structures, the limit state function is usually implicit and must be evaluated by a numerical method such as FEM, which is a time-consuming computational process. This creates difficulties during the calculation of the partial derivatives of the performance function in reliability analyses that use FORM. The GP model with the performance functions in explicit form is used in this paper to approximate the original complex and/or implicit limit state function. The partial derivatives of the performance functions can thus be easily obtained.

According to Eq. (9), the first-order partial derivatives of the approximate function can be calculated as follows

$$\frac{\partial g(\boldsymbol{x}^*)}{\partial x_i} \approx \frac{\partial \overline{y}(\boldsymbol{x}^*)}{\partial x_i} = \sum_{j}^{m} \boldsymbol{\alpha}_j \frac{\partial \boldsymbol{k}(\boldsymbol{x}_j, \boldsymbol{x}^*)}{\partial x_i} \tag{24}$$

when a squared exponential covariance function is used, then $\dfrac{\partial \boldsymbol{k}(\boldsymbol{x}_j,\boldsymbol{x}^*)}{\partial x_i} = \dfrac{x^* - x_{ij}}{l_i^2}\sigma_f^2 \exp\left(-\dfrac{1}{2\boldsymbol{l}^2}\left|\boldsymbol{x}_j - \boldsymbol{x}^*\right|^2\right)$; when a linear covariance function is used, then $\dfrac{\partial \boldsymbol{k}(\boldsymbol{x}_j,\boldsymbol{x}^*)}{\partial x_i} = \dfrac{x_i^*}{l_i^2}$; when a Matérn covariance function is used, then $\dfrac{\partial \boldsymbol{k}(\boldsymbol{x}_j,\boldsymbol{x}^*)}{\partial x_i} = \dfrac{3(x^* - x_{ij})}{l_i^2}\sigma_f^2 \exp\left(\dfrac{\sqrt{3}\left|\boldsymbol{x}_j - \boldsymbol{x}^*\right|}{\boldsymbol{l}}\right)$.

When we substitute Eqs. (9) and (24) into Eq. (21), the reliability index β is given as a GP formulation

$$\beta = \frac{\sum_{\boldsymbol{j}=1}^{\boldsymbol{m}} \alpha_j k(x_j, x^*) + \mu_y + \sum_{\boldsymbol{i}=1}^{\boldsymbol{n}}\sum_{\boldsymbol{j}=1}^{\boldsymbol{m}} \alpha_j \dfrac{\partial k(x_j, x^*)}{\partial \boldsymbol{x_i}}(\mu_{\boldsymbol{x}_i} - \boldsymbol{x}_i^*)}{\sqrt{\sum_{\boldsymbol{i}=1}^{\boldsymbol{n}}\sum_{\boldsymbol{j}=1}^{\boldsymbol{m}}\left[\alpha_j \dfrac{\partial k(x_j, x^*)}{\partial \boldsymbol{x_i}}\right]^2 \sigma_{\boldsymbol{x}_i}^2}} \tag{25}$$

When we substitute Eqs. (9) and (24) into Eq. (23), Eq. (23) can then be rewritten as

$$\cos\theta_{x_i} = -\frac{\sum_{j=1}^{m} \boldsymbol{\alpha}_j \dfrac{\partial \boldsymbol{k}(\boldsymbol{x}_j, \boldsymbol{x}^*)}{\partial x_i}\sigma_{x_i}}{\sqrt{\sum_{i=1}^{n}\left[\sum_{j=1}^{m} \boldsymbol{\alpha}_j \dfrac{\partial \boldsymbol{k}(\boldsymbol{x}_j, \boldsymbol{x}^*)}{\partial x_i}\right]^2 \sigma_{x_i}^2}} \tag{26}$$

In addition, if we substitute Eqs. (25) and (26) into Eq. (22), we can obtain the coordinates of design point $\boldsymbol{x}^*$ by using a GP approximation.

2.4 Procedure of the proposed method

The GP-based RSM differentiates itself from the classic RSM by employing the GP to simultaneously approximate the performance function and its first-order partial derivatives.

The procedure of the proposed method is as follows:

Step 1 Assume initial values of the design point $\boldsymbol{x}^*=(x_1,x_2,\ldots,x_n)$. We typically select the mean value of the random variables as the coordinates of design point $\boldsymbol{x}^*$.

Step 2 Build training datasets and scale the datasets.

Step 3 Train the GP model through the training datasets and self-adaptively obtain the optimum hyperparameters. The default of the covariance function in this paper is the squared exponential covariance function.

Step 4 Extract the explicit formulation of the approximate performance function through use of the well-trained GP model.

Step 5 Compute the reliability index $\beta^{(k)}$of the kth iteration step by using Eq. (25).

Step 6 Compute the values of the new design point according to Eq. (22).

Step 7 Check the convergence criterion for $|\beta^{(k)}-\beta^{(k-1)}|\le\varepsilon$($\varepsilon$=0.001 in this paper).

(1) If convergence criteria are not satisfied, then Go to Step 3 and repeat Steps 3-6 until convergence are satisfied. To constantly improve the reconstructing precision at the region that significantly contributes to the failure probability, the new design point and its value of actual performance function, taken as a new training sample, is added into the training datasets.

(2) If convergence criteria are satisfied, then Go to Step 8.

Step 8 Calculate the probability of failure $p_f=\Phi(-\beta)$.

A **MATLAB**-based program was developed to apply the aforementioned methods into a structural reliability analysis.

3. Numerical examples

3.1 Example 1: a hypothetical nonlinear performance function

This example comes from Kim and Na (1997), in which the limit state is defined as

$$g(X)=\exp(0.2x_1+6.2)-\exp(0.47x_2+5.0) \tag{27}$$

where X is assumed to be independent and has a standard normal distribution with a zero mean and a unit standard deviation.

The training set consisting of 5 points was generated according to the method mentioned above, where f=2. The calculated values of the performance function are listed in Table 1. The optimum hyperparameters of the trained GP were $\boldsymbol{\theta}$=(2.4282, 1.4534, 7.0508， -6.8937).

The predicted performance values of the trained GP are also listed in Table 1. The small prediction error of MAE=0.0002 from the GP indicates that the training was successful. If a vector from the basic random variables is input, the GP can generate an adequately accurate value of the corresponding performance function. Therefore, the GP can be employed to represent the true

Table 1 Initial training datasets and the predicted outputs from GP for Example 1

No.	x_1	x_2	y	$\hat{y}$	$\lvert y-\hat{y}\rvert$
1	-2	0	181.8864	181.8863	0.0001
2	0	-2	434.7747	434.7747	0.0000
3	2	0	586.6820	586.6820	0.0000
4	0	2	112.8141	112.8142	0.0001
5	0	0	344.3359	344.3361	0.0002

Table 2 Calculation process of the design point using FORM and proposed method for Example 1

No. of iterations			I	II	III	IV	V	VI	VII
Design point	x^*_1		0.0000	-1.679	-0.887	-0.847	-0.957	-0.900	-0.920
	x^*_2		0.0000	1.392	2.367	2.046	2.146	2.170	2.162
First order derivative	$\frac{\partial g}{\partial x_1^*}$	Analytical	98.550	70.436	82.537	83.201	81.384	82.314	-
		Predictive	121.544	36.851	47.452	82.713	79.503	82.335	-
	$\frac{\partial g}{\partial x_2^*}$	Analytical	-69.754	-134.178	-212.167	-182.429	-191.289	-193.386	-
		Predictive	-100.744	-98.369	-114.665	-185.530	-191.624	-193.412	-
Reliability index β			2.181	2.527	2.214	2.350	2.349	2.349	2.349

performance function.

The GP approximation of the performance function on the points of training set is defined as

$$g(x_i) \approx \sum_{j=1}^{5} \alpha_i k(x_i, x_j) + 332.0987 \tag{28}$$

where the values of $\boldsymbol{\alpha}$ and covariance function $\boldsymbol{K}$ of the trained GP model are as follows:
$\boldsymbol{\alpha}$=[0, -0.0023,0.0117, -0.0103, 0.0004]

$$\boldsymbol{K} = \begin{bmatrix} 6.0348 & 3.7069 & 2.5706 & 3.7069 & 4.8754 \\ & 6.0348 & 3.7069 & 2.0168 & 4.5885 \\ & & 6.0348 & 3.7069 & 4.8754 \\ & \text{sym} & & 6.0348 & 4.5885 \\ & & & & 6.0348 \end{bmatrix} \times 10^4$$

Table 2 shows the values of the design points and their first order derivatives, as well as the reliability indices of all iteration steps using the proposed method. It clearly illustrates that the errors of the predictive values of the first order derivatives of the performance function become ever smaller with the increase of iteration steps.

The failure probabilities and the reliability indices obtained through the various methods are summarized in Table 3. An exact reliability index is obtained by using a direct MCS at 2.9024. It can be seen that the reliability index from RSM (f=2) and the GP-based RSM are very similar to the results from FORM and RSM (Herbert and Armando 2004). The present method, using GP based-RSM, required 7 function evaluations (Fig. 2), while the RSM required 25 function evaluations. Moreover, as seen in Fig. 2, the GP approximation based on the final training datasets

Table 3 Results of Example 1 using different methods

Method	Probability of failure ($\times 10^{-3}$)	Reliability index	Function calls
FORM	9.4036	2.3493	-
RSM (Herbert and Armando 2004)	9.41	2.349	-
RSM (f=2)	9.3980	2.3496	25
GP-based RSM	9.4037	2.3493	10

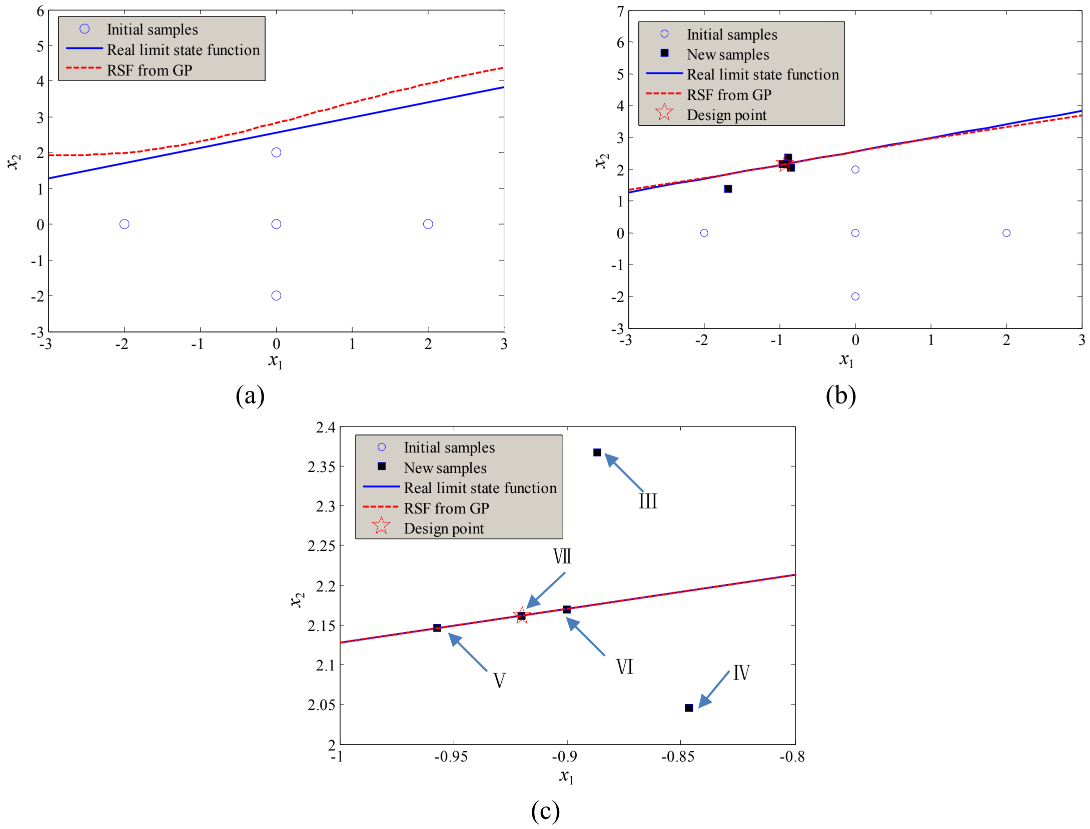

Fig. 2 An approximate limit state function from GP based on (a) initial training datasets and (b) final training datasets; (c) enlarged figure (b) in the range of $-1 \le x_1 \le -0.8$

provided better performance on the approximation of the limit state function than that based on the initial training datasets. This indicates that the dynamic updating scheme of the training datasets played a significant role in improving the accuracy of the GP approximation.

3.2 Example 2: a highly non-linear performance function

This example analyzes the reliability of a cantilever beam with a rectangular cross section, distributed uniform load and linear elastic behavior (Gayton *et al.* 2003). The limit state function is designated as the maximum vertical displacement at the free end of the beam. The displacement must not exceed the serviceability limit of $L/325$, where L is the length of the beam. The length L

Table 4 Statistical data for Example 2

Variables	Mean value	Standard deviation	Unit	Distribution
x_1	1000.0	200.0	MPa	Normal
x_2	250	37.5	mm	Normal

Table 5 Initial training datasets and the predicted outputs from GP for Example 2

No.	x_1	x_2	y	$\hat{y}$	$\|y-\hat{y}\|$
1	1400	250	11.7622	11.7622	0.0000
2	1000	325	16.2835	16.2835	0.0000
3	600	250	15.5904	15.5905	0.0001
4	1000	175	4.5104	4.5102	0.0002
5	1000	250	13.6763	13.6764	0.0001

and the Young's modulus E of the beam are assumed to be deterministic variables with values of L=600 mm and E=2.6×104 MPa. The limit state function is explicitly provided as

$$g(X) = 0.01846154 - 74.76923\frac{x_1}{x_2^3} \tag{29}$$

where x_1 is the load w (MPa) and x_2 is the height h (mm). These two variables have statistical characteristics as shown in Table 4 and are taken as non-correlated variables with a Gaussian probability distribution.

The training samples consisted of 5 points, and their calculated values of the performance function and the predicted values of the trained GP are listed in Table 5. The small prediction error from the GP indicates that the training was successful. The optimal hyperparameters of the trained GP were $\boldsymbol{\theta}$=(3.49, -2.46, 5.43, 4.49, -13.21). The formulation of the GP approximation of the performance function based on the training datasets can be written as

$$g(x_i) \approx \sum_{j=1}^{5} \alpha_i k(x_i, x_j) + 12.3646 \tag{30}$$

where the covariance function $k(x_i, x_j)$ is the column of the matrix $\boldsymbol{K}$; the values of vector $\boldsymbol{\alpha}$ and matrix $\boldsymbol{K}$**,** based on the initial training datasets, are as follows:
$\boldsymbol{\alpha}$=[-0.2533, 0.0969, 1.3305, -0.1941, -1.0284]

$$\boldsymbol{K} = \begin{bmatrix} 40.4610 & 0 & 39.8427 & 0 & 39.8427 \\ & 40.4610 & 0 & 0 & 0 \\ & & 40.4610 & 0 & 39.8427 \\ & \text{sym} & & 40.4610 & 0 \\ & & & & 40.4610 \end{bmatrix}$$

All of the results from the different methods are summarized in Table 6. The probability of failure, the reliability index and number of actual performance function evaluations are compared. These results concur with those provided by RYFES/COMREL (Gayton *et al.* 2003), FORM and

Table 6 Results of Example 2 using different methods

Method	Reliability index β	Probability of failure P_f ($\times 10^{-3}$)	Function calls
RYFES/COMREL (Gayton *et al.* 2003)	2.331	-	-
FORM	2.3309	9.879	9
RSM (f=2)	2.3312	9.870	31
GP-based RSM	2.3309	9.879	17

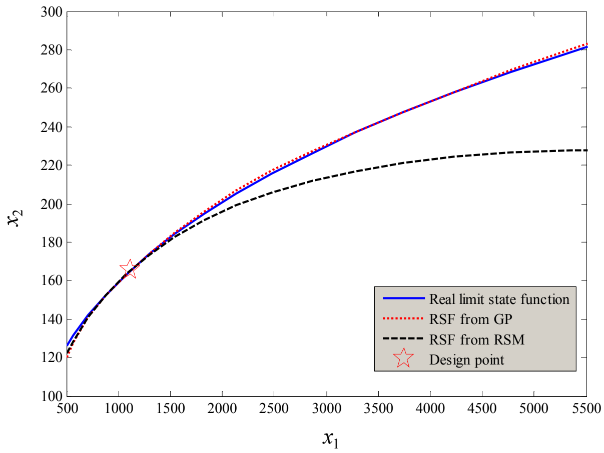

Fig. 3 An approximate limit state function from GP and RSM based on initial training datasets

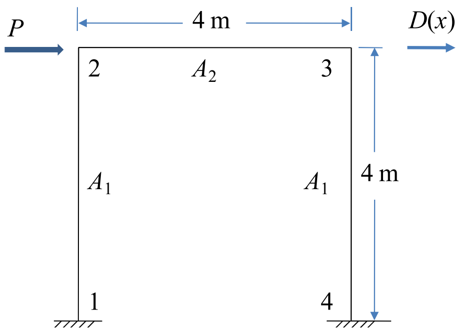

Fig. 4 A portal frame for reliability analysis

RSM (f=2). However, fewer function evaluations are required by the GP-based RSM than the RSM (f=2). Moreover, as shown in Fig. 3, the GP-based RSM can capture the entire shape of the performance function, while the RSM (f=2) can only adequately approximate the performance function around the design points, which indicates that the GP-based RSM is more suitable for highly non-linear problems than the RSM.

3.3 Example 3: a portal frame

The performance function for the portal frame shown in Fig. 4 can be defined as follows

Table 7 Statistical data for Example 3

Variables	Mean value	Unit	Standard deviation	Type of Distribution	Coefficient α_i
A_1	0.36	m^2	0.036	lognormal	0.0833
A_2	0.18	m^2	0.018	lognormal	0.1667
P	20	kN	5.0	Type 1 largest	-

Table 8 Initial training datasets and the predicted outputs from GP for Example 3

No.	A_1	A_2	P	y	$\hat{y}$	$\|y-\hat{y}\|$
1	0.288	0.180	20.000	0.0041	0.0041	0.0000
2	0.360	0.144	20.000	0.0050	0.0050	0.0000
3	0.360	0.180	10.000	0.0078	0.0078	0.0000
4	0.432	0.180	20.000	0.0066	0.0066	0.0000
5	0.360	0.216	20.000	0.0061	0.0062	0.0001
6	0.360	0.180	30.000	0.0035	0.0035	0.0000
7	0.360	0.180	20.000	0.0057	0.0056	0.0001

Table 9 Results of Example 3 using different methods

Method	Reliability index β	Probability of failure P_f ($\times 10^{-3}$)	Number of FEM analysis
ISM (Zhao 1996)	2.8307	2.322	2000
RSM (Zhao 1996)	2.8405	2.250	41
ANN-based RSM (Deng *et al.* 2005)	-	2.3	33
GP-based RSM	2.8317	2.3152	14

$$g(\boldsymbol{x}) = 0.001 - u_3(X) \tag{31}$$

where $u_3(X)$ denotes the horizontal displacement (in meters) at node 3 as a function of basic random variables.

The three basic random variables include the column cross-section area A_1, the beam cross-section area A_2 and the wind load P. The statistical parameters of these basic random variables are listed in Table 7. All of the variables are assumed to be uncorrelated. The Young's modulus of the members is assumed to be deterministic and is equal to 2.0×10^6 kN/ m^2. The moments of inertia for the beam and the columns are related to the cross-section areas as $I_i = \gamma_i A_i^2$ (i=1 or 2), where I_i are the moments of inertia and γ_i are the coefficients for the column (i = 1) and the beam (i = 2).

The performance function of this problem is implicit. The response variable u_3 is dependent on both the random variables and the deterministic variables; it can be evaluated through an FEM. The training datasets are composed of 7 samples, as listed in Table 8. The values of the performance function from the FEM and the GP are also shown in Table 8. It is evident that the GP can be employed to represent the actual performance function through an FEM analysis.

The results shown in Table 9 indicate that a GP-based RSM can achieve better results as compared to an RSM when an exact solution is obtained by using an ISM with 2,000 simulations (Zhao 1996). A significantly lower number of FEM analyses were required from the GP-based

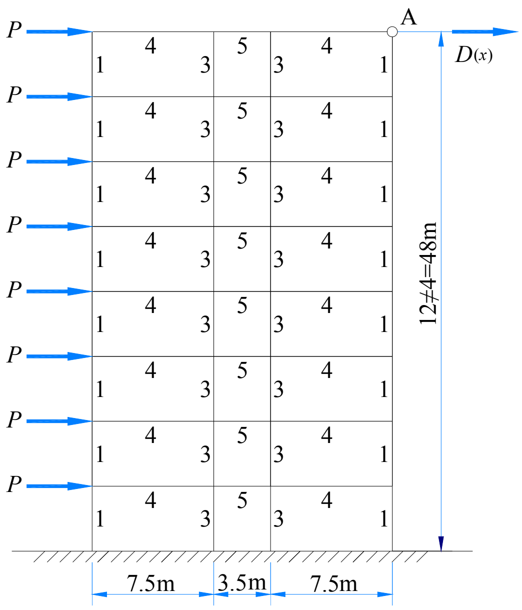

Fig. 5 A three-bay of twelve-story frame

Table 10 Statistical data for Example 4

Variables	Mean value	Standard deviation	Type of Distribution	Coefficient α_i
A_1	0.25 m^2	0.025	lognormal	0.0833
A_2	0.16 m^2	0.016	lognormal	0.0833
A_3	0.36 m^2	0.036	lognormal	0.0833
A_4	0.20 m^2	0.020	lognormal	0.2667
A_5	0.15 m^2	0.015	lognormal	0.2
P	30.0 kN	7.5	Type 1 largest	

ISM when compared with the RSM and ANN-RSM (Deng *et al.* 2005). This indicates that the proposed method is significantly more economical than either an RSM or an ANN-RSM in achieving reasonable accuracy.

3.4 Example 4: a three-bay of twelve-story frame

The performance function for the portal frame structure shown in Fig. 5 may be defined as follows

$$g(X) = 0.096 - u_A(X) \tag{32}$$

where $u_A(X)$ denotes the horizontal displacement (in meters) at node A as the function of basic random variables. In this equation, $g(X)<0$ indicates failure.

The six basic random variables include the column and beam cross-section areas A_1, A_2, A_3, A_4, A_5 and the wind load P. The Young's modulus of all of the members is assumed to be deterministic and is equal to 2.0×10^7 kN/m^2. The statistical parameters of the basic random

Table 11 Initial training datasets and the predicted outputs from GP for Example 4

No.	A_1	A_2	A_3	A_4	A_5	P	y	$\hat{y}$	$\lvert y-\hat{y}\rvert$
1	0.2	0.16	0.36	0.2	0.15	30	0.0215	0.0216	0.0001
2	0.25	0.128	0.36	0.2	0.15	30	0.0267	0.0267	0.0000
3	0.25	0.16	0.288	0.2	0.15	30	0.0240	0.0240	0.0000
4	0.25	0.16	0.36	0.16	0.15	30	0.0162	0.0162	0.0000
5	0.25	0.16	0.36	0.2	0.12	30	0.0253	0.0254	0.0001
6	0.25	0.16	0.36	0.2	0.15	15	0.0626	0.0626	0.0000
7	0.3	0.16	0.36	0.2	0.15	30	0.0340	0.0341	0.0001
8	0.25	0.192	0.36	0.2	0.15	30	0.0309	0.0309	0.0000
9	0.25	0.16	0.432	0.2	0.15	30	0.0324	0.0324	0.0000
10	0.25	0.16	0.36	0.24	0.15	30	0.0373	0.0373	0.0000
11	0.25	0.16	0.36	0.2	0.18	30	0.0320	0.0321	0.0001
12	0.25	0.16	0.36	0.2	0.15	45	-0.0042	-0.0042	0.0000
13	0.25	0.16	0.36	0.2	0.15	30	0.0292	0.0288	0.0004

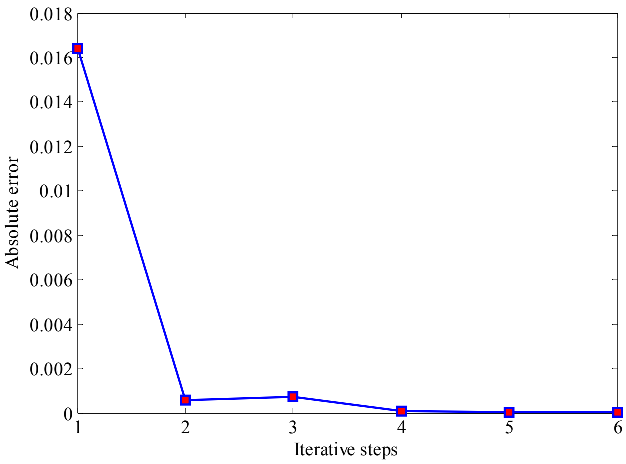

Fig. 6 Absolute errors of prediction of design points vs. iterative step

variables are listed in Table 10. All of the variables are assumed to be uncorrelated. The moments of inertia for the beam and the columns correlate with the cross-section areas as follows: $I_i = \alpha_i A_i^2$, where I_i are the moments of inertia and A_i are the coefficients whose values are listed in Table 10.

The performance function of this problem does not explicitly contain any of the six basic random variables. The response variable u_A is dependent on the random variables and the deterministic variables, which cannot be expressed as a closed-form function—it must be evaluated using an FEM. The performance function is implicit.

The training datasets are composed of 13 samples, which are listed in Table 11. The values of the performance function from the FEM and the GP are also shown in Table 11. We can see that the GP can be employed to represent the actual performance function through the FEM analysis.

As shown in Fig. 6, the gaps between the limit state function values of the tentative design

Table 12 Results of Example 4 using different methods

Method	Probability of failure P_f	Number of FEM analysis
ISM (Zhao 1996)	0.07506	2000
RSM (Zhao 1996)	0.07309	41
RSM (Das and Zheng 2000)	0.05316	26
GP-based RSM	0.07339	19

Table 13 Results obtained by applying number of initial training datasets for Example 1

f	Number of initial training datasets	Reliability index	Probability of failure P_f ($\times 10^{-3}$)
1	5	2.3493	9.4036
1, 2	9	2.3494	9.4026
1, 2, 3	13	2.3493	9.4036
1, 2, 3, 4	17	2.3493	9.4050
1, 2, 3, 4, 5	21	2.3493	9.4036

points from the GP prediction and the FEM quickly become very small with increasing iterative steps. This indicates that the strategy of dynamically updating training datasets in the proposed method is effective at improving the accuracy of the GP model approximation.

The results for the ISM, RSM and GP-based RSM are shown in Table 12. The results indicate that the accuracy of the GP-based RSM is slightly better than that of the classic RSM (Zhao 1996, Das and Zheng 2000) when an exact solution is obtained by using an ISM with 2,000 simulations (Zhao1996), but the number of FEM analysis inquiries of the GP-based RSM is obviously less than that of the classic RSM. A lower number of FEM analysis calls indicates a higher level of efficiency for the proposed method as compared to the classic RSM.

4. Sensitivity study of the proposed method

The proposed GP-based RSM method is primarily based on the following parameters: (1) the number of initial training datasets; (2) the value of the parameter f, which defines the locations of the initial training datasets; and (3) the type of the covariance function. In this section, different values are selected for these parameters in order to determine their effects on the final results. In the interest of simplicity, only Example 1 in the previous section is considered.

4.1 Sensitivity on the number of initial training datasets

Different numbers of initial training datasets are generated by selecting different values of parameter f. In keeping the other parameters unchanged and by only varying the number of training samples for the example, the results are then compared in Table 13. The table shows that changing the number of initial training samples has only a minor effect on the accuracy of estimated values of β. This indicates that the proposed method can perform well even when the number of initial training samples is very small.

Table 14 Results obtained by applying number of initial training datasets for Example 1

f	Reliability index	Probability of failure P_f ($\times 10^{-3}$)
0.01	2.34931	9.40360
1	2.34932	9.40400
2	2.34933	9.40367
3	2.34935	9.40324
4	2.34933	9.40365
5	2.34934	9.40337
10	2.34933	9.40365

Table 15 Results obtained by applying different covariance functions for Example 1

Method	FORM	GP-based RSM		
		Squared Exponential	Matérn (ν=3/2)	Linear
Probability of failure ($P_f\times 10^{-3}$)	9.4036	9.4037	9.4030	3.8392
Reliability index (β)	2.3493	2.3493	2.3494	2.6659

4.2 Sensitivity on the value of parameter f

As discussed in the previous section, the locations of the initial training datasets depend on the value of the mean and standard deviations of random variable $\boldsymbol{X}$ and parameter f. Because parameter f is arbitrarily chosen, the question then arises as to how to set the value of f. In keeping the other parameters unchanged and by only varying the value of f in the example, the results are then listed in Table 14. The table shows that changing the value of initial training samples has only a minor effect on the accuracy of estimating the probability of failure. However, as a rule of thumb, we recommend that f be selected as an integer in the range of 1 to 4.

4.3 Sensitivity on the type of the selected covariance function

Three types of the covariance functions mentioned above are tested in Example 1; the results are shown in Table 15. As seen in the table, the squared exponential and Matérn covariance functions provide better results than the linear covariance function, and the squared exponential and Matérn covariance functions require fewer actual performance function calculations than the linear covariance function. Therefore, the squared exponential and Matérn covariance functions are recommended for solving reliability problems with highly nonlinear limit state functions.

5. Conclusions

A new GP-based RSM was developed for predicting the failure probability of a structure. The method involved the selection of training datasets for establishing a GP model through the design method of a classic RSM method, the approximation of the limit state function and its first-order partial derivative by the trained GP model and an estimation of the failure probability by using FORM. In the proposed method, the use of renewed continuously training datasets may improve the performance of a GP model for approximating the limit state function around the design point. Thus, the GP dramatically reduced the number of required trained datasets and demonstrated the ability to approximate the limit state function and provide an accurate estimation of the probability of failure when connected with the FORM. Four numerical examples involving both structural and non-structural problems illustrated the application and effectiveness of the proposed method. The proposed method is particularly suitable for structural probability problems when a structural response evaluation entails a time-consuming finite element analysis. Compared with a conventional response surface method, the proposed method is significantly more economical in achieving reasonable accuracy for a structural probability analysis. The proposed approach, based on the FORM, is a kind of response surface method. The employed powerful GP response surface can be used to perform a better approximation to the limit state function when comparing with classic quadratic polynomial response surface. Thus, the proposed method is applicable to structural reliability analyses that involve highly non-linear implicit performance function that entail time-consuming finite element analyses. However, it should be noted that the proposed method is not intended as a replacement for the existing classic RSM, but rather as a possible complement for this method.

Acknowledgments

We wish to acknowledge the financial support of the National Natural Science Foundation of China (Grant No. 51369007).

References

Bucher, C.G. and Bourgund, U. (1990), "A fast and efficient response surface approach for structural reliability problems", *Struct. Saf.*, **7**(1), 57-66.

Chen, T., Morris, J. and Martin, E. (2007), "Gaussian process regression for multivariate spectroscopic calibration", *Chemometr. Intell. Labs.*, **87**(1), 59-71.

Cheng, J., Li, Q.S. and Xiao, R.C. (2008), "A new artificial neural network-based response surface method for structural reliability analysis", *Probab. Eng. Mech.*, **23**(1), 51-63.

Christian, B. and Thomas, M. (2008), "A comparison of approximate response functions in structural reliability analysis", *Probab. Eng. Mech.*, **23**(1), 154-163.

Das, P.K. and Zheng, Y. (2000), "Cumulative formation of response surface and its use in reliability analysis", *Probab. Eng. Mech.*, **15**(4), 309-315.

Deng, J., Gu, D.S., Li, X.B. and Yue, Z.Q. (2005), "Structural reliability analysis for implicit performance function using artificial neural network", *Struct. Saf.*, **27**(1), 25-48.

Gayton, N., Bourinet, J.M. and Lemaire, M. (2003), "CQ2RS: a new statistical approach to the response surface method for reliability analysis", *Struct. Saf.*, **25**(1), 99-121.

Guan, X.L. and Melchers, R.E. (2001), "Effect of response surface parameter variation on structural reliability estimates", *Struct. Saf.*, **23**(4), 429-444.

Hensman, J., Mills, R., Pierce, S.G., Worden, K. and Eaton, M. (2010), "Locating acoustic emission sources in complex structures using Gaussian processes", *Mech. Syst. Signal Pr.*, **24**(1), 211-223.

Herbert, M.G. and Armando, M.A. (2004), "Comparison of response surface and neural network with other methods for structural reliability analysis", *Struct. Saf.*, **26**(1), 49-67.

Hurtado, J.E. (2007), "Filtered importance sampling with support vector margin: a powerful method for structural reliability analysis", *Struct. Saf.*, **29**(1), 2-15.

Jiang, S.H., Li, D.Q., Zhou, C.B. and Zhang, L.M. (2014), "Capabilities of stochastic response surface method and response surface method in reliability analysis", *Struct. Eng. Mech*, **49**(1), 111-128.
Kim, S.H. and Na, S.W. (1997), "Response surface method using vector projected sampling points", *Struct. Saf.*, **19**(1), 3-19.
Li, H.S., Lv, Z.Z. and Yue, Z.F. (2006), "Support vector machine for structural reliability analysis", *Appl. Math. Mech.*, **27**(10-11), 35-43.
Luc, S. and Dionys, V.G. (2005), "Benefit of splines and neural networks in simulation based structural reliability analysis", *Struct. Saf.*, **27**(3), 246-261.
MacKay, D.J.C. (1998), *Introduction to Gaussian processes*, Technical report, Cambridge University, London, UK.
Malur, R.R. and Bruce, R.E. (1993), "A new look at the response surface approach for reliability analysis", *Struct. Saf.*, **12**(3), 205-220.
Rackwitz, R. and Fiessler, B. (1976), *Note on discrete safety checking when using non-normal stochastic models for basic variable*, Loads Project Working Session, MIT Press, London.
Rasmusen, C.E. and Williams, C.K.I. (2006), *Gaussian Processes for Machine Learning*, MIT Press, London.
Rocco, C.M. and Moreno, J.A. (2002), "Fast Monte Carlo reliability evaluation using support vector machine", *Reliab. Eng. Syst. Saf.*, **76**(2), 37-43.
Su, G.S., Yan, L.B. and Song, Y.C. (2007), "Gaussian process for non-linear displacement time series prediction of landslide", *J. China Univ. Geosci.*, **18**, 212-219.
Vapnik, V. (1998), *Statistical learning theory*, John Wiley and Sons, New York.
Williams, C.K.I. (1998), *Prediction with Gaussian processes: from the linear regression to linear prediction and beyond*, Technical report, Aston University, Birmingham, UK.
Zhao, G.F. (1996), *Reliability theory and its applications for engineering structures*, Dalian University of Technology Press, Dalian, PRC.
Zhao, H.B. (2008), "Slope reliability analysis using a support vector machine", *Comput. Geotech.*, **35**(4), 59-67.
Zhao, W.T., Qiu Z.P and Yang, Y. (2013), "An efficient response surface method considering the nonlinear trend of the actual limit state", *Struct. Eng. Mech.*, **47**(1), 45-58.

CC