

## Research Article

# Improved Reliability-Based Optimization with Support Vector Machines and Its Application in Aircraft Wing Design

Yu Wang,<sup>1</sup> Xiongqing Yu,<sup>1</sup> and Xiaoping Du<sup>2</sup>

<sup>1</sup>College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

<sup>2</sup>Department of Mechanical and Aerospace Engineering, Missouri University of Science and Technology, Rolla, MO 65409, USA

Correspondence should be addressed to Yu Wang; wangyu@nuaa.edu.cn

Received 25 September 2014; Revised 20 January 2015; Accepted 13 April 2015

Academic Editor: Marc Dahan

Copyright © 2015 Yu Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A new reliability-based design optimization (RBDO) method based on support vector machines (SVM) and the Most Probable Point (MPP) is proposed in this work. SVM is used to create a surrogate model of the limit-state function at the MPP with the gradient information in the reliability analysis. This guarantees that the surrogate model not only passes through the MPP but also is tangent to the limit-state function at the MPP. Then, importance sampling (IS) is used to calculate the probability of failure based on the surrogate model. This treatment significantly improves the accuracy of reliability analysis. For RBDO, the Sequential Optimization and Reliability Assessment (SORA) is employed as well, which decouples deterministic optimization from the reliability analysis. The improved SVM-based reliability analysis is used to amend the error from linear approximation for limit-state function in SORA. A mathematical example and a simplified aircraft wing design demonstrate that the improved SVM-based reliability analysis is more accurate than FORM and needs less training points than the Monte Carlo simulation and that the proposed optimization strategy is efficient.

## 1. Introduction

There are many uncertainties encountered in both of the aircraft manufacturing process and its subsequent flight operation. The physical properties of materials are uncertain. Manufacturing errors produce the aerodynamic shape and structural dimensions different from the original design [1]. Furthermore, the load on the aircraft is not constant during operation. Fuel is consumed continually during cruise; thus, flight parameters keep changing. After an aircraft has been produced, some parameters sometimes need to be adjusted for a new type. If these uncertainties are considered in the conceptual design, the aircraft performance will be more reliable than deterministic design [2], and both of the risk and cost in the design will be reduced.

Reliability analysis is the key part of reliability-based design optimization (RBDO). Reliability is the probability of success. In physics-based reliability, the status of success is specified by limit-state functions, which are derived from

physics principles [3]. Let the limit-state function be  $g(\mathbf{X})$ , where  $\mathbf{X}$  is a vector of random variables with length  $n$ . If the event of success is specified by  $g(\mathbf{X}) \geq 0$ , then the failure event is  $g(\mathbf{X}) < 0$ . Consequently, the probability of failure is computed by

$$p_f = \Pr \{g(\mathbf{X}) < 0\} = \int_{g(\mathbf{X}) < 0} f(\mathbf{x}) d\mathbf{x}, \quad (1)$$

where  $f(\mathbf{x})$  is the joint probability density function (PDF) of  $\mathbf{X}$ .

Since it is difficult to analytically evaluate the probability integral, many approximation methods have been developed. Among them, the First Order Reliability Method (FORM) [4–6] is commonly used. The FORM linearizes the limit-state function  $g(\mathbf{X})$  at the Most Probable Point (MPP), which in the standard normal space has the highest probability of producing the value of limit-state function. Then, (1) becomes a linear combination of normal variables. Since only the first

order derivatives of  $g(\mathbf{X})$  and some basic information are needed, the FORM is efficient. Its accuracy, however, may not be satisfactory for highly nonlinear limit-state functions.

Several methods are available to improve the accuracy of the FORM. One of the methods is the Second Order Reliability Method (SORM) [7]. The SORM provides a second order approximation to  $g(\mathbf{X})$  at the MPP. As a result, it is generally more accurate than the FORM. The MPP-based importance sampling [8, 9] is another alternative method. Random samples are drawn from distributions whose center is shifted to the MPP. A relatively small sample size can then produce a good estimation of the probability of failure. The other strategy is to approximate the safety-failure boundary  $g(\mathbf{X}) = 0$  at the MPP with higher accuracy. In the point-fitting method [10], a piecewise paraboloid surface is built with the fitting points selected from each side of MPP along both forward and backward directions of each random variable. Similarly, response surface modeling has also been used to create a surrogate model for the limit-state function at the MPP [11, 12]. A surrogate model can be created using artificial neural network [13] as well.

Recently, another strategy, the support vector machine (SVM) [14] method, has been introduced in reliability analysis. SVM is a statistical classification method. As indicated in [15], reliability analysis can be viewed as a classification problem where SVM is applicable. Examples of using SVM include the fast Monte Carlo simulation (MCS) [16], the limit-state function identification for discontinuous responses and disjoint failure domains [17], SVM-based MCS [18], and virtual SVM for high-dimensional problems [19]. It is worthwhile to further study SVM for reliability analysis and RBDO.

The conventional approach for solving a reliability-based design optimization problem is to employ a double-loop strategy in which the analysis and the synthesis are nested in such a way that the synthesis loop performs the reliability analysis iteratively for meeting the probabilistic constraints. As the double-loop strategy may be computationally infeasible, various single loop strategies have been studied to improve its efficiency. The method ‘‘approximately equivalent deterministic constraints’’ creates a link between a probabilistic design and a safety-factor based design [20]. The reliability constraints are formulated as deterministic constraints that approximate the condition of the MPP for reliability analysis [21]. A single loop method, Sequential Optimization and Reliability Assessment (SORA), is a very efficient method for RBDO [22]. In this method, optimization and reliability analysis are decoupled from each other; no reliability analysis is required within optimization and the reliability analysis is only conducted after the optimization. Hence, the design is quickly improved from cycle to cycle and the computational efficiency is improved significantly. However, because of FORM employed in SORA based on the limit-state function linearization, its precision may not be high enough for the highly nonlinear problem.

In this work, the accuracy of SVM-based reliability analysis was improved firstly. In addition to the training points around the MPP, the gradient of a limit-state function at the MPP was included in approximating the limit-state

function, to guarantee that the surrogate model not only passes through the MPP but also is tangent to the limit-state function at the MPP. And importance sampling is used to estimate the probability of failure based on the surrogate model. Then, the improved SVM-based reliability analysis was integrated into SORA for wing optimization. Results of the two examples showed that this strategy is more accurate than before with a moderately increased computational cost.

## 2. FORM, IS, and SVM

In this section, the three methods used in this work are reviewed. The methods include the First Order Reliability Method (FORM), importance sampling (IS), and support vector machines (SVM).

*2.1. First Order Reliability Method (FORM).* In this work, we assume all random variables in  $\mathbf{X}$  are independent. FORM involves the following three steps.

(1) *Transformation of Random Variables into Standard Normal Variables.* The original random variables  $\mathbf{X} = (X_1, X_2, \dots, X_n)$  (in the  $X$ -space) are transformed into random variables  $\mathbf{U} = (U_1, U_2, \dots, U_n)$  (in the  $U$ -space) whose elements follow a standard normal distribution. The transformation is given by [23]

$$u_i = \Phi^{-1} [F_i(x_i)], \quad (i = 1, 2, \dots, n), \quad (2)$$

where  $F_i(x_i)$  is the cumulative distribution function (CDF) of  $X_i$  and  $\Phi^{-1}$  is the inverse CDF of a standard normal variable.

(2) *Search the Most Probable Point (MPP).* The MPP is the point at the limit-state  $g(\mathbf{U}) = 0$ , and, at the MPP, the PDF of  $\mathbf{U}$  is at its maximum. Maximizing the joint PDF of  $\mathbf{U}$  and noting that a contour of the PDF of  $\mathbf{U}$  is a concentric hypersphere, we obtain the MPP  $\mathbf{u}^*$  by solving

$$\begin{aligned} \min_{\mathbf{u}} \quad & \|\mathbf{u}\|, \\ \text{subject to} \quad & g(\mathbf{u}) = 0, \end{aligned} \quad (3)$$

where  $\|\cdot\|$  stands for the norm (length) of a vector.

Geometrically, the MPP is the shortest distance point from surface  $g(\mathbf{U}) = 0$  to the origin of the  $U$ -space. The minimum distance  $\beta = \|\mathbf{u}^*\|$  is called the *reliability index*. Then,  $g(\mathbf{U})$  is approximated by the first Taylor expansion series at  $\mathbf{u}^*$  as

$$g(\mathbf{U}) \approx g(\mathbf{u}^*) + \nabla(\mathbf{u}^*)(\mathbf{U} - \mathbf{u}^*) = \nabla(\mathbf{u}^*)(\mathbf{U} - \mathbf{u}^*), \quad (4)$$

where  $\nabla(\mathbf{u}^*)$  is the gradient of  $g$  at the MPP and is given by

$$\nabla g(\mathbf{u}^*) = \left( \frac{\partial g(\mathbf{U})}{\partial U_1}, \frac{\partial g(\mathbf{U})}{\partial U_2}, \dots, \frac{\partial g(\mathbf{U})}{\partial U_n} \right) \Big|_{\mathbf{u}^*}. \quad (5)$$

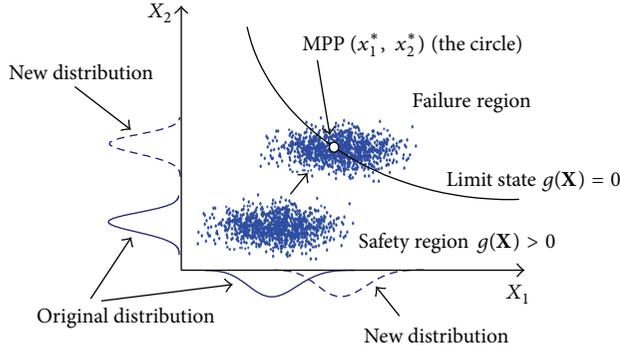


FIGURE 1: Importance sampling.

(c) Calculate the Probability of Failure. As shown in (4),  $g(\mathbf{U})$  is now a linear combination of normal variables. As a result,  $p_f = \Pr\{g < 0\}$  can be easily computed by

$$p_f = P\{g(\mathbf{X}) < 0\} = \Phi(-\beta), \quad (6)$$

where  $\Phi$  is the CDF of a standard normal variable.

2.2. Importance Sampling (IS). The FORM is commonly used due to its good balance between accuracy and efficiency. For highly nonlinear limit-state functions, however, the accuracy may not be good enough. IS can be used to improve accuracy. The probability of failure in (1) can be rewritten as

$$p_f = \int_{-\infty}^{+\infty} I(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}, \quad (7)$$

where  $I(\cdot)$  is an indicator function and is defined as

$$I(\mathbf{x}) = \begin{cases} 1, & \text{if } g(\mathbf{x}) \leq 0, \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

$p_f$  is therefore the expectation of  $I(\mathbf{x})$ ; namely,

$$p_f = E\{I(\mathbf{x})\}, \quad (9)$$

where  $E$  denotes an expectation. With the direct Monte Carlo simulation (MCS),  $p_f$  can be estimated by averaging  $I(\mathbf{x})$ :

$$p_f = \frac{1}{N} \sum_{i=1}^N I(\mathbf{x}_i) = \frac{N_f}{N}, \quad (10)$$

where  $\mathbf{x}_i$  ( $i = 1, 2, \dots, N$ ) are the samples drawn from the joint probability density  $f(\mathbf{x})$  and  $N_f$  is the number of failures.

A large sample size  $N$  is required when  $p_f$  is small because the chance of getting samples in failure region is small. IS draws samples from a new set of distributions such that more samples will be in the failure region. One strategy is to shift the mean values of the random variables to the MPP. As shown in Figure 1, all the samples (the lower cloud) generated from the original distributions of  $X_1$  and  $X_2$  are in the safe region. They do not contribute to the probability

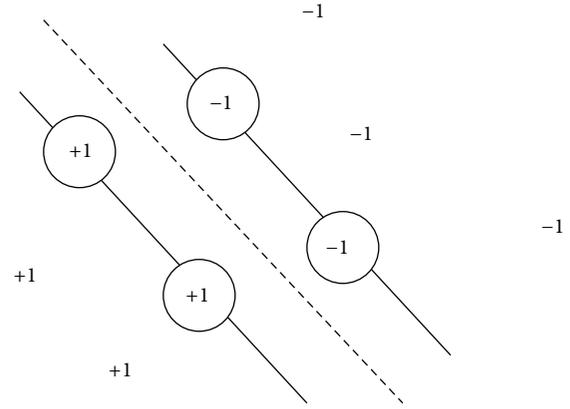


FIGURE 2: Marginal classifiers along with support vectors [27].

estimation. After the mean values are shifted to the MPP, sufficient samples are now in the failure region.

Let the importance sampling density be  $h(\mathbf{x})$ , which is obtained by shifting the means of  $f(\mathbf{x})$  to the MPP. Then,  $p_f$  is estimated by

$$\begin{aligned} p_f &= \int I(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} = \int \left[ I(\mathbf{x}) \frac{f(\mathbf{x})}{h(\mathbf{x})} \right] h(\mathbf{x}) d\mathbf{x} \\ &\approx \frac{1}{N} \sum_{i=1}^N I(\mathbf{x}_i) \frac{f(\mathbf{x}_i)}{h(\mathbf{x}_i)}. \end{aligned} \quad (11)$$

As shown in Figure 1, IS obtains a significant number of samples in the failure region with the same sample size. Because the indicator function  $I(\mathbf{x})$  of a failure point is 1.0, there are a significant number of nonzero terms in the summation in (11). However, it is very difficult for the direct MCS to generate failure samples and the terms in (10) are almost zero. As a result, no estimation of  $p_f$  can be produced. Then, more samples are needed. Therefore, IS more efficient than the direct MCS.

2.3. Support Vector Machine (SVM). The SVM method has been primarily used for solving both linear and nonlinear classification problems in statistics. Because in reliability analysis there are two states, either safety or failure, SVM is also applicable for reliability analysis. Recently, several studies on SVM in reliability analysis have been reported [16–18]. The basics of SVM are briefly reviewed in this subsection.

For a linear limit-state function, SVM classifies training data (samples) into two classes, typically represented by “+1” or “-1,” depending on whether the sample belongs to safety or failure region. Two parallel hyperplanes are obtained by maximizing their distance, as shown in Figure 2. There are no samples between the hyperplanes. They therefore separate the samples into safety and failure groups. The points (the circles in the figure) passed by these hyperplanes are called support vectors (SV). The center hyperplane, which lies in half the distance from both hyperplanes, is actually used as the safety-failure boundary [17, 24].

This center hyperplane is the surrogate model for  $g(\mathbf{X})$  at the limit-state  $g(\mathbf{X}) = 0$  and is represented by

$$y = \mathbf{w} \cdot \mathbf{X} + b, \quad (12)$$

where  $\mathbf{w}$  is a weight vector,  $\mathbf{w} \cdot \mathbf{X}$  is the dot (inner) product of  $\mathbf{w}$  and  $\mathbf{X}$ , and  $b$  is the bias. And one outer hyperplane follows

$$\mathbf{w} \cdot \mathbf{X} + b = +1 \quad (13)$$

and the other hyperplane satisfies

$$\mathbf{w} \cdot \mathbf{X} + b = -1. \quad (14)$$

Then, the distance between the two hyperplanes is  $2/(\mathbf{w} \cdot \mathbf{w})$ . As mentioned previously, the distance must be maximized. Because we can minimize  $(1/2)\mathbf{w} \cdot \mathbf{w}$ . To make sure there are no training points between the two hyperplanes, we need to add the following constraint to the optimization model [24]:

$$y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad (15)$$

where  $\mathbf{x}_i$  ( $i = 1, 2, \dots, m$ ) is the vector of training points and  $m$  is the number of training points.  $y_i$  is +1 if  $\mathbf{x}_i$  is in the safety region and -1 otherwise.

The optimization model is then given by

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2} \mathbf{w} \cdot \mathbf{w}, \\ \text{s.t.} \quad & y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad (i = 1, 2, \dots, m). \end{aligned} \quad (16)$$

The optimization is a quadratic programming problem, and it is usually converted into a Lagrangian dual problem [25, 26]. The dual optimization model is

$$\begin{aligned} \max_{\lambda} \quad & L = \sum_{i=1}^m \lambda_i - \frac{1}{2} \sum_{i,j=1}^m \lambda_i \lambda_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j, \\ \text{s.t.} \quad & \sum_{i=1}^n y_i \lambda_i = 0, \\ & 0 \leq \lambda_i, \quad i = 1, 2, \dots, m, \end{aligned} \quad (17)$$

where  $\lambda$  is the Lagrangian multiplier.

Once  $\lambda$  is obtained, the weight vector in (12) can be computed by

$$\mathbf{w} = \sum_{i=1}^m \lambda_i y_i \mathbf{x}_i. \quad (18)$$

According to Karush-Kuhn-Tucker conditions, only the support vectors have non-zero Lagrangian multipliers [27], that is, only the SV appear in the optimum result [25]. Therefore, the bias  $b$  in (12) can be determined by SV with  $\lambda_j > 0$ :

$$b = y_j - \sum_{i=1}^m \lambda_i y_i \mathbf{x}_i \cdot \mathbf{x}_j. \quad (19)$$

For nonlinear limit-state functions, hypersurfaces must be generated. To do so, we can simply replace the dot product  $\mathbf{x}_i \cdot \mathbf{x}_j$  in (17) with the following kernel function:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\varphi}(\mathbf{x}_i) \cdot \boldsymbol{\varphi}(\mathbf{x}_j), \quad (20)$$

where  $\boldsymbol{\varphi}(\mathbf{x})$  is the feature function of  $\mathbf{x}$ , which is used to map the original variable into a higher-dimensional space called the feature space. In the  $N$ -dimensional feature space,  $\mathbf{x}$  is represented by  $(\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x}), \dots, \varphi_N(\mathbf{x}))$ . It is noted that the kernel function is the dot product of the two feature functions. And the most popular kernel functions are Gaussian kernel and polynomial kernel, which are defined as

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right), \quad (21)$$

$$K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i \cdot \mathbf{x}_j)^d, \quad (22)$$

respectively. Although the feature function  $\boldsymbol{\varphi}(\mathbf{x})$  appears in (20), it will not be used directly.

Then, the simplified (surrogate) model for the limit-state function is

$$y = \mathbf{w} \cdot \boldsymbol{\varphi}(\mathbf{X}) + b = \sum_{i=1}^m \lambda_i y_i K(\mathbf{x}_i, \mathbf{X}) + b. \quad (23)$$

### 3. Improved SVM Method for Reliability Analysis

As discussed above, MPP-based importance sampling (IS) is more accurate than the FORM. But it may still require a significant number of samples for high accuracy. To further improve the accuracy with balanced efficiency, we propose to integrate the FORM, SVM, and IS. To use the full information of the MPP, we also use the gradient of  $g(\mathbf{U})$  at the MPP in the  $U$ -space. With the additional information of the gradient, the SVM surrogate model will be more accurate. We call this method SVM-MPP-G method. Its procedure is as follows:

- (1) FORM: search the MPP  $\mathbf{u}^*$ ;
- (2) IS: generate samples at the MPP;
- (3) SVM: create a surrogate model  $\hat{g}(\mathbf{U})$  for the limit-state function  $g(\mathbf{U}) = 0$  based on the IS samples and the gradient of  $g(\mathbf{U})$  at the MPP in the  $U$ -space;
- (4) IS or MCS: perform IS or MCS around the MPP to evaluate the probability of failure  $p_f$  with  $\hat{g}(\mathbf{U})$ .

The flowchart of the method is shown in Figure 3. The details of the steps are provided below.

**3.1. Step 1: MPP Search.** The MPP is identified with the model in (3). Nonlinear optimization algorithms can be used to solve the model. A specialized search algorithm, however, may be more efficient than optimization. In this work, we use the robust MPP search algorithm, the improved HRLF algorithm, denoted by iHRLF. The algorithm is proposed by Zhang and

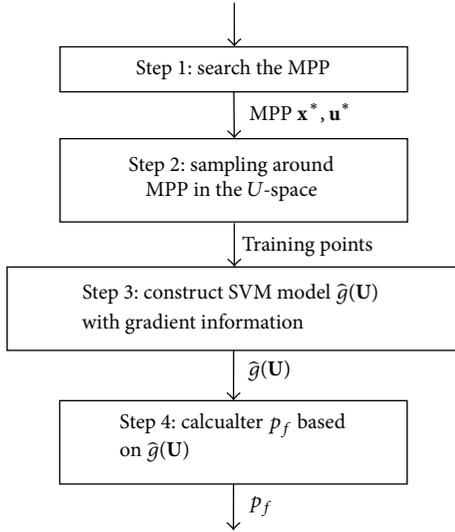


FIGURE 3: Flowchart of the proposed method.

der Kiureghian [28]. iHRLF is efficient; it is also globally convergent because it can converge to a local MPP from any starting point. The algorithm is summarized below.

In iteration  $k + 1$ , the MPP is updated by

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \alpha \mathbf{d}_k, \quad (24)$$

where the search direction  $\mathbf{d}_k$  is given by

$$\mathbf{d}_k = \frac{\nabla g(\mathbf{u}_k) \mathbf{u}_k^T - g(\mathbf{u}_k) \nabla g_y(\mathbf{u}_k)}{\|\nabla g_y(\mathbf{u}_k)\|^2} - \mathbf{u}_k, \quad (25)$$

where  $\nabla g(\mathbf{u}_k) = (\partial g / \partial U_1, \partial g / \partial U_2, \dots, \partial g / \partial U_n)_{\mathbf{u}_k}$ .

The step size  $\alpha$  is determined by minimizing a merit function

$$m(\mathbf{u}) = \frac{1}{2} \|\mathbf{u}\| + c |g(\mathbf{u})| \quad (26)$$

in which the constant  $c$  should satisfy

$$c > \frac{\|\mathbf{u}\|}{\|\nabla g(\mathbf{u})\|}. \quad (27)$$

Practically, searching for the step size  $\alpha$  terminates once the merit function is sufficiently reduced. The following rule is employed to find  $\alpha$ :

$$\alpha = \max_{h \in \mathbb{N}} \{b^h \mid m(\mathbf{u}_k + b^h \mathbf{d}_k) - m(\mathbf{u}_k) < 0\}, \quad b > 0. \quad (28)$$

In this paper,  $b = 0.5$  and  $c = 2\|\mathbf{u}_k\|/\|\nabla g(\mathbf{u}_k)\| + 10$  are used.

Equation (28) indicates that  $\alpha = h$  is the first integer such that the merit function  $m(\mathbf{u}_k + b^h \mathbf{d}_k)$  is less than the previous merit function  $m(\mathbf{u}_k)$ .

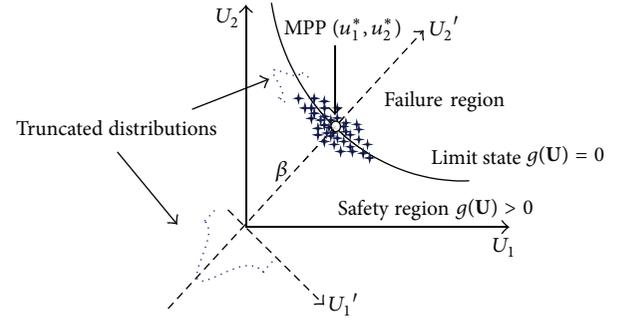


FIGURE 4: Sampling strategy.

If the distance between two consecutive points  $\mathbf{u}_{k+1}$  and  $\mathbf{u}_k$  is less than a small value  $\varepsilon$ ; namely,  $\|\mathbf{u}_{k+1} - \mathbf{u}_k\| \leq \varepsilon$ , the procedure stops. The result is the MPP  $\mathbf{u}^*$  in the  $U$ -space. The MPP  $\mathbf{x}^*$  in the  $X$ -space can be obtained by the inverse transformation based on (2)

$$\mathbf{x}_i^* = F_{x_i}^{-1}[\Phi(u_i^*)], \quad (i = 1, 2, \dots, n). \quad (29)$$

**3.2. Step 2: Sampling around the MPP.** We are interested in creating the limit-state boundary accurately. Since the MPP has the highest probability density at the limit-state, we particularly desire higher accuracy around the MPP. To this end, we draw samples around the MPP. To further improve the sampling efficiency, we also use the gradient information of the limit-state function at the MPP so that the samples will be closer to the limit-state boundary. Since the gradient is a byproduct of the MPP search, no further computations are needed for its use.

To easily draw samples around the MPP, we perform sampling in the  $U$ -space. The sampling distributions are standard normal ones, but their means are shifted to the MPP. If the samples are drawn from the sampling distributions directly, they will form a symmetric hypersphere around the MPP. Some of them may be far away from the limit-state boundary. To make them close to the boundary, we shrink the sampling distributions in the gradient direction. To make the samples closer to the MPP, we also truncate the distributions. As shown in Figure 4, more samples are in the vicinity of the MPP with the truncation and shrinking.

To implement the idea, we first rotate coordinates  $(U_1, U_2, \dots, U_n)$  into a new set of coordinates  $(U'_1, U'_2, \dots, U'_n)$ , so that the last coordinate  $U'_n$  ( $U'_2$  in the figure) is coincident with the vector from the origin to the MPP. In another word,  $U'_n$  is colinear with the gradient of  $g(\mathbf{U})$  at the MPP. In the new linear  $U'$ -space, the coordinates of the MPP become  $[0, \dots, 0, \beta]$ . This treatment makes the shrinking easier.

We actually draw samples in the rotated  $U'$ -space. The distributions are standard normal distributions for  $U'_i$  ( $i = 1, 2, \dots, n-1$ ); namely,  $U'_i \sim N(0, 1)$  ( $i = 1, 2, \dots, n-1$ ). The distribution of  $U'_n$  is also a normal distribution with its mean being  $\beta$ ; namely,  $U'_n \sim N(\beta, 1)$ . To make the samples close to the limit-state boundary, we truncate all the distributions. Suppose the PDF of  $U'_i$  ( $i = 1, 2, \dots, n$ ) is  $\phi_{U'_i}(u)$  and  $U'_i$  lies within the interval  $(\mu_i - \Delta_i, \mu_i + \Delta_i)$ ,  $\Delta_i > 0$ , where  $\Delta_i$  is the

distance between the mean value and one of the truncated tails; the truncated PDF [29] is then given by

$$\phi_{U'_i}^t(u') = \begin{cases} \frac{\phi_{U'_i}(u')}{\Phi_{U'_i}(\mu_i + \Delta_i) - \Phi_{U'_i}(\mu_i - \Delta_i)}, & \text{if } \mu_i - \Delta_i < u' < \mu_i + \Delta_i, \\ 0, & \text{otherwise,} \end{cases} \quad (30)$$

where  $\Phi_{U'_i}(u')$  is the CDF of  $U'_i$ .

For  $U'_i$  ( $i = 1, 2, \dots, n-1$ ), we use the same distance  $\Delta_i$  ( $i = 1, 2, \dots, n-1$ ) to shrink the distribution. In the gradient direction in the  $U'_n$ -axis, we truncate the distribution of  $U'_n$  with a smaller distance  $\Delta_n$ ; namely,  $\Delta_n < \Delta_i$  ( $i = 1, 2, \dots, n-1$ ). Generally, the distances  $\Delta_1 = 1.0$  and  $\Delta_2 = 0.15 \sim 0.25$  could be taken.

Because of its good efficiency, we also use Latin Hypercube Sampling (LHS) [30] instead of the direct Monte Carlo sampling. In general, 10~25 times the number of random variables could be taken as the number of sampling points. After the samples of  $\mathbf{U}'$  are generated, they need to be transformed into the  $U$ -space. If each sample point is represented by a row vector, an orthogonal transformation is implemented by

$$\mathbf{U} = \mathbf{U}'\mathbf{R}, \quad (31)$$

where  $\mathbf{R}$  is the rotation matrix. It could be computed with the gradient information of the limit-state function at the MPP, which has been obtained during the MPP searching. The matrix  $\mathbf{R}$  is derived from another matrix  $\mathbf{R}_0$  with the Gram-Schmidt orthogonalization.  $\mathbf{R}_0$  is given by

$$\mathbf{R}_0 = \begin{bmatrix} \mathbf{I}_{n-1} & 0 \\ \alpha_1 \alpha_2 \cdots \alpha_{n-1} & \alpha_n \end{bmatrix}, \quad (32)$$

where  $\mathbf{I}_{n-1}$  is an  $(n-1) \times (n-1)$  identity matrix, and the last row of  $\mathbf{R}_0$  is a unit vector defined as follows:

$$\boldsymbol{\alpha} = \frac{\mathbf{u}^*}{\|\mathbf{u}^*\|}. \quad (33)$$

Then,  $\mathbf{R}$  can be generated from  $\mathbf{R}_0$  with the following Gram-Schmidt orthogonalization procedure: let the rows of  $\mathbf{R}_0$  and  $\mathbf{R}$  be  $\mathbf{r}_{0i}$  and  $\mathbf{r}_i$  ( $i = 1, 2, \dots, n$ ), respectively. Consider

$$\begin{aligned} \mathbf{r}_n &= \mathbf{r}_{0n}, \\ \mathbf{r}'_k &= \mathbf{r}_{0k} - \sum_{j=k+1}^n \frac{\mathbf{r}_j \mathbf{r}'_{0k}}{\mathbf{r}_j \mathbf{r}'_j} \mathbf{r}_j, \\ \mathbf{r}_k &= \frac{\mathbf{r}'_k}{\|\mathbf{r}'_k\|}, \quad k = n-1, n-2, \dots, 1. \end{aligned} \quad (34)$$

Therefore, we could obtain the training points in the  $U$ -space with (31).

**3.3. Step 3: Surrogate Model Construction.** With the training points obtained above, we now construct the SVM surrogate model  $\hat{g}(\mathbf{U})$  for  $g(\mathbf{U})$ .  $\hat{g}(\mathbf{U})$  is not approximated for actual value of  $g(\mathbf{U})$ . Just the limit-state functions  $\hat{g}(\mathbf{U}) = 0$  is similar to  $g(\mathbf{U}) = 0$  in the vicinity of the MPP. Different from the traditional SVM methods, this method incorporates the gradient information at the MPP in the optimization model. This allows us to use the full information obtained during the MPP search.

For nonlinear limit-state functions, as discussed in Section 2.3, a feature function  $\boldsymbol{\varphi}(\mathbf{x})$  ( $\boldsymbol{\varphi}(\mathbf{U})$  in the following discussions) is used to accommodate the nonlinearity. As shown in (23), the approximated limit-state function is given by

$$y = \hat{g}(\mathbf{U}) = \mathbf{w} \cdot \boldsymbol{\varphi}(\mathbf{U}) + b. \quad (35)$$

The derivative of  $\hat{g}(\mathbf{U})$  with respect to  $U_p$  ( $p = 1, 2, \dots, n$ ) is [31]

$$\left. \frac{\partial \hat{g}}{\partial U_p} \right|_{\mathbf{U}=\mathbf{u}^*} = \mathbf{w} \cdot \boldsymbol{\varphi}'_p(\mathbf{u}^*) = c_p e, \quad (36)$$

where  $\mathbf{c} = \nabla g(\mathbf{u}^*) = (\partial g(\mathbf{U})/\partial U_1, \partial g(\mathbf{U})/\partial U_2, \dots, \partial g(\mathbf{U})/\partial U_n)|_{\mathbf{u}^*}$  is the gradient of limit-state function at the MPP and  $e$  is the proportionality factor.  $e = (\mathbf{w} \cdot \boldsymbol{\varphi}'_h(\mathbf{u}^*))/c_h$ , and  $|c_h| = \max\{|c_1|, |c_2|, \dots, |c_n|\}$ .

We add (36) into the original optimization problem in (16) as additional linear constraints. The new optimization model then becomes

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2} \mathbf{w} \cdot \mathbf{w} \\ \text{s.t.} \quad & y_i [\mathbf{w} \cdot \boldsymbol{\varphi}(\mathbf{u}_i) + b] \geq 1, \quad i = 1, 2, \dots, m, \\ & \left. \frac{\partial \hat{g}}{\partial u_p} \right|_{\mathbf{U}=\mathbf{u}^*} = \mathbf{w} \cdot \boldsymbol{\varphi}'_p(\mathbf{u}^*) = c_p e, \\ & p = 1, 2, \dots, n. \end{aligned} \quad (37)$$

The Lagrangian function of the dual problem changes to

$$\begin{aligned} L = \frac{1}{2} \mathbf{w} \cdot \mathbf{w} & - \sum_{i=1}^m \lambda_i \{y_i [\mathbf{w} \cdot \boldsymbol{\varphi}(\mathbf{u}_i) + b] - 1\} \\ & - \sum_{p=1}^n \gamma_p \left[ \mathbf{w} \cdot \boldsymbol{\varphi}'_p(\mathbf{u}^*) - c_p \frac{\mathbf{w} \cdot \boldsymbol{\varphi}'_h(\mathbf{u}^*)}{c_h} \right] \end{aligned} \quad (38)$$

and  $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \dots, \gamma_n]$ .  $\lambda_i$  ( $i = 1, 2, \dots, 3$ ) should be nonnegative.  $y_i$  is +1 if  $g(\mathbf{u}_i)$  is positive and -1 otherwise. Given the

Karush-Kuhn-Tucker (KKT) complementary conditions [27] for maximizing  $L$ , the following equations hold:

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^m \lambda_i \gamma_i \boldsymbol{\varphi}(\mathbf{u}_i) - \sum_{p=1}^n \gamma_p \left[ \boldsymbol{\varphi}'_p(\mathbf{u}^*) - \frac{c_p \boldsymbol{\varphi}'_h(\mathbf{u}^*)}{c_h} \right] = 0, \quad (39)$$

$$\mathbf{w} = \sum_{i=1}^m \lambda_i \gamma_i \boldsymbol{\varphi}(\mathbf{u}_i) + \sum_{p=1}^n \gamma_p \left[ \boldsymbol{\varphi}'_p(\mathbf{u}^*) - \frac{c_p \boldsymbol{\varphi}'_h(\mathbf{u}^*)}{c_h} \right], \quad (40)$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^m \lambda_i \gamma_i = 0, \quad (41)$$

$$\frac{\partial L}{\partial \gamma_p} = \mathbf{w} \cdot \boldsymbol{\varphi}'_p(\mathbf{u}^*) - c_p \frac{\mathbf{w} \cdot \boldsymbol{\varphi}'_h(\mathbf{u}^*)}{c_h} = 0, \quad (42)$$

$$p = 1, 2, \dots, n; \quad p \neq h.$$

Plugging (40) into (38) yields

$$L = -\frac{1}{2} \sum_{i,j=1}^m \lambda_i \lambda_j \gamma_i \gamma_j \boldsymbol{\varphi}(\mathbf{u}_i) \cdot \boldsymbol{\varphi}(\mathbf{u}_j) + \sum_{i=1}^m \lambda_i \left[ \frac{1}{2} \sum_{p,q=1}^n \gamma_p \gamma_q \left[ \boldsymbol{\varphi}'_q(\mathbf{u}^*) \cdot \boldsymbol{\varphi}'_p(\mathbf{u}^*) - \frac{2c_q}{c_h} \boldsymbol{\varphi}'_h(\mathbf{u}^*) \cdot \boldsymbol{\varphi}'_p(\mathbf{u}^*) + \frac{c_p c_q \boldsymbol{\varphi}'_h(\mathbf{u}^*) \cdot \boldsymbol{\varphi}'_h(\mathbf{u}^*)}{c_h^2} \right] \right]. \quad (43)$$

Because

$$\boldsymbol{\varphi}(\mathbf{u}_i) \cdot \boldsymbol{\varphi}(\mathbf{u}_j) = K(\mathbf{u}_i, \mathbf{u}_j), \quad (44)$$

$$\begin{aligned} \boldsymbol{\varphi}'_q(\mathbf{u}^*) \cdot \boldsymbol{\varphi}'_p(\mathbf{u}^*) &= \left. \frac{\partial^2 K(\mathbf{u}_i, \mathbf{u}_j)}{\partial x_{jp} \partial x_{iq}} \right|_{\mathbf{u}_i=\mathbf{u}_j=\mathbf{u}^*} \\ &= K''_{pq}(\mathbf{u}^*, \mathbf{u}^*), \end{aligned} \quad (45)$$

then

$$L = -\frac{1}{2} \sum_{i,j=1}^m \lambda_i \lambda_j \gamma_i \gamma_j K(\mathbf{u}_i, \mathbf{u}_j) + \sum_{i=1}^m \lambda_i \left[ \frac{1}{2} \sum_{p,q=1}^n \gamma_p \gamma_q \left[ K''_{qp}(\mathbf{u}^*, \mathbf{u}^*) - \frac{2c_q}{c_h} K''_{hp}(\mathbf{u}^*, \mathbf{u}^*) + \frac{c_p c_q K''_{hh}(\mathbf{u}^*, \mathbf{u}^*)}{c_h^2} \right] \right]. \quad (46)$$

And plugging (40) into (42) yields

$$\begin{aligned} \sum_{i=1}^m \lambda_i \gamma_i \left[ \boldsymbol{\varphi}(\mathbf{u}_i) \cdot \boldsymbol{\varphi}'_p(\mathbf{u}^*) - \frac{c_p}{c_h} \boldsymbol{\varphi}(\mathbf{u}_i) \cdot \boldsymbol{\varphi}'_h(\mathbf{u}^*) \right] \\ + \sum_{q=1}^n \gamma_q \left[ \boldsymbol{\varphi}'_q(\mathbf{u}^*) \cdot \boldsymbol{\varphi}'_p(\mathbf{u}^*) - \frac{c_p \boldsymbol{\varphi}'_q(\mathbf{u}^*) \cdot \boldsymbol{\varphi}'_h(\mathbf{u}^*)}{c_h} \right. \\ \left. - \frac{c_q \boldsymbol{\varphi}'_h(\mathbf{u}^*) \cdot \boldsymbol{\varphi}'_p(\mathbf{u}^*)}{c_h} + \frac{c_p c_q \boldsymbol{\varphi}'_h(\mathbf{u}^*) \cdot \boldsymbol{\varphi}'_h(\mathbf{u}^*)}{c_h^2} \right] \\ = 0, \quad p = 1, 2, \dots, n, \quad p \neq h. \end{aligned} \quad (47)$$

With (45) and

$$\boldsymbol{\varphi}(\mathbf{u}_i) \cdot \boldsymbol{\varphi}'_p(\mathbf{u}^*) = \left. \frac{\partial K(\mathbf{u}_i, \mathbf{u}_j)}{\partial x_{jp}} \right|_{\mathbf{u}_j=\mathbf{u}^*} = K'_p(\mathbf{u}_i, \mathbf{u}^*), \quad (48)$$

the additional constraints become

$$\begin{aligned} \sum_{i=1}^m \lambda_i \gamma_i \left[ K'_p(\mathbf{u}_i, \mathbf{u}^*) - \frac{c_p}{c_h} K'_h(\mathbf{u}_i, \mathbf{u}^*) \right] \\ + \sum_{q=1}^n \gamma_q \left[ K''_{qp}(\mathbf{u}^*, \mathbf{u}^*) - \frac{c_p K''_{qh}(\mathbf{u}^*, \mathbf{u}^*)}{c_h} \right. \\ \left. - \frac{c_q K''_{hp}(\mathbf{u}^*, \mathbf{u}^*)}{c_h} + \frac{c_p c_q K''_{hh}(\mathbf{u}^*, \mathbf{u}^*)}{c_h^2} \right] = 0, \end{aligned} \quad (49)$$

$$p = 1, 2, \dots, n, \quad p \neq h.$$

Then, the dual problem is

$$\begin{aligned} \min \quad & \frac{1}{2} \sum_{i,j=1}^m \lambda_i \lambda_j \gamma_i \gamma_j K(\mathbf{u}_i, \mathbf{u}_j) - \sum_{i=1}^m \lambda_i - \frac{1}{2} \sum_{p,q=1}^n \gamma_p \gamma_q \left[ K''_{qp}(\mathbf{u}^*, \mathbf{u}^*) - \frac{2c_q}{c_h} K''_{hp}(\mathbf{u}^*, \mathbf{u}^*) + \frac{c_p c_q K''_{hh}(\mathbf{u}^*, \mathbf{u}^*)}{c_h^2} \right], \\ \text{s.t.} \quad & \sum_{i=1}^m \gamma_i \lambda_i = 0, \\ & \sum_{i=1}^m \lambda_i \gamma_i \left[ K'_p(\mathbf{u}_i, \mathbf{u}^*) - \frac{c_p}{c_h} K'_h(\mathbf{u}_i, \mathbf{u}^*) \right] \end{aligned}$$

$$+ \sum_{q=1}^n \gamma_q \left[ K''_{qp}(\mathbf{u}^*, \mathbf{u}^*) - \frac{c_p K''_{qh}(\mathbf{u}^*, \mathbf{u}^*)}{c_h} - \frac{c_q K''_{hp}(\mathbf{u}^*, \mathbf{u}^*)}{c_h} + \frac{c_p c_q K''_{hh}(\mathbf{u}^*, \mathbf{u}^*)}{c_h^2} \right] = 0, \quad p = 1, 2, \dots, n, \quad p \neq h,$$

$$\lambda_i \geq 0, \quad i = 1, 2, \dots, m. \quad (50)$$

Because  $\hat{g}(\mathbf{U})$  should pass through the MPP  $\mathbf{u}^*$ , we can use  $\hat{g}(\mathbf{u}^*) = 0$  to find the bias  $b$ . Using  $\hat{g}(\mathbf{u}^*) = \mathbf{w} \cdot \boldsymbol{\varphi}(\mathbf{u}^*) + b = 0$  and (40), we obtain

$$b = -\mathbf{w} \cdot \boldsymbol{\varphi}(\mathbf{u}^*)$$

$$= -\sum_{i=1}^m \gamma_i \lambda_i K(\mathbf{u}^*, \mathbf{u}_i)$$

$$- \sum_{p=1}^n \gamma_p \left[ K'_p(\mathbf{u}^*, \mathbf{u}^*) - \frac{c_p K'_h(\mathbf{u}^*, \mathbf{u}^*)}{c_h} \right]. \quad (51)$$

The approximated limit-state function is then given by

$$\hat{g}(\mathbf{U}) = \mathbf{w} \cdot \boldsymbol{\varphi}(\mathbf{U}) + b = \sum_{i=1}^m \lambda_i \gamma_i [K(\mathbf{U}, \mathbf{u}_i)$$

$$- K(\mathbf{u}^*, \mathbf{u}_i)]$$

$$+ \sum_{p=1}^n \gamma_p \left\{ [K'_p(\mathbf{U}, \mathbf{u}^*) - K'_p(\mathbf{u}^*, \mathbf{u}^*)] \right.$$

$$\left. - \frac{c_p [K'_h(\mathbf{U}, \mathbf{u}^*) - K'_h(\mathbf{u}^*, \mathbf{u}^*)]}{c_h} \right\}. \quad (52)$$

**3.4. Step 4: Probability Evaluation.** Once  $\hat{g}(\mathbf{U})$  is available, a reliability analysis method can be used to estimate the probability of failure  $p_f$ . Since  $\hat{g}(\mathbf{U})$  is cheaper to compute, we can afford a large number of function calls to maintain higher accuracy. We can therefore use importance sampling (IS) or the direct Monte Carlo simulation. If IS is selected, the probability of failure is computed with (11).

#### 4. Integration of SORA and SVM Method for Reliability Analysis

A typical model of a probabilistic design is given by

$$\text{Minimize: } f(\mathbf{d}, \mathbf{X}, \mathbf{P}),$$

$$\text{Design Variable: } \mathbf{d}, \boldsymbol{\mu}_X,$$

$$\text{Subject to: } \Pr\{g_t(\mathbf{d}, \mathbf{X}, \mathbf{P}) \geq 0\} \geq R_t,$$

$$t = 1 \sim T, \quad (53)$$

where  $f$  is an objective function,  $\mathbf{d}$  is the vector of deterministic design variables,  $\mathbf{X}$  is the vector of random design variables,  $\mathbf{P}$  is the vector of random design parameters,

$\Pr\{g_t(\mathbf{d}, \mathbf{X}, \mathbf{P}) \geq 0\} \geq R_t$ ,  $t = 1 \sim T$ , are constraint functions,  $R_t$  are the desired probabilities of constraint satisfaction, and  $T$  is the number of constraints. The design variables are  $\mathbf{d}$  and the means ( $\boldsymbol{\mu}_X$ ) of the random design variables  $\mathbf{X}$ .

**4.1. Sequential Optimization and Reliability Assessment (SORA).** The conventional reliability-based optimization employs a double-loop strategy. The outer loop is to solve the optimization problem, and the inner loop is for reliability analysis. The total number of function evaluations will be huge. However, the SORA method decouples the reliability analysis and optimization completely.

In SORA, an equivalent model to the probabilistic constraints in (53) with performance measure approach is given by

$$g_t^R(\mathbf{d}, \mathbf{X}, \mathbf{P}) \geq 0, \quad t = 1 \sim T, \quad (54)$$

where  $g^R$  is the  $R$ -percentile of  $g(\mathbf{d}, \mathbf{X}, \mathbf{P}) \geq 0$ ; namely,

$$\Pr\{g(\mathbf{d}, \mathbf{X}, \mathbf{P}) \geq g^R\} = R. \quad (55)$$

It indicates that the probability of  $g(\mathbf{d}, \mathbf{X}, \mathbf{P})$  greater than or equal to the  $R$ -percentile  $g^R$  is exactly equal to the desired reliability  $R$ . Using the inverse MPP search algorithm, the optimum solution MPP  $\mathbf{u}_{\text{MPP}}$  can be identified and the desired  $R$  percentile is evaluated by

$$g^R = g(\mathbf{u}_{\text{MPP}}) = g(\mathbf{x}_{\text{MPP}}, \mathbf{p}_{\text{MPP}}). \quad (56)$$

Then, the design model (1) is rewritten as

$$\text{Minimize: } f(\mathbf{d}, \mathbf{X}, \mathbf{P}),$$

$$\text{Design Variable: } \mathbf{d}, \boldsymbol{\mu}_X,$$

$$\text{Subject to: } g_t^R(\mathbf{d}, \mathbf{X}_{\text{MPP}t}, \mathbf{P}_{\text{MPP}t}) \geq 0,$$

$$t = 1 \sim T. \quad (57)$$

The concept of SORA method is shown in Figure 5. The optimization in the first cycle is just a deterministic optimization. Then, the reliability analysis is implemented for the deterministic optimum solution to locate the MPP that corresponds to the desired  $R$  level. In Cycle 2, the constraints are modified to shift the MPP onto the deterministic boundary to help insure the feasibility of the probabilistic constraint. The new constraints are

$$g_t(\mathbf{d}, \boldsymbol{\mu}_X - \mathbf{s}_t, \mathbf{p}_{\text{MPP}}) \geq 0, \quad (58)$$

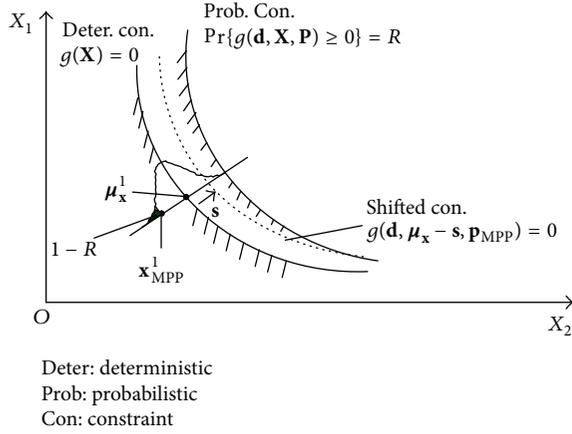


FIGURE 5: Shifting constraint boundary [22].

where  $\mathbf{s}$  is the shifting vector as  $\mathbf{s}_t = \boldsymbol{\mu}_X^1 - \mathbf{x}_{MPP}^1$ . The superscript 1 means the first cycle.

The reliabilities of those violated probabilistic constraints will improve remarkably using this MPP shifting strategy. After the optimization in Cycle 2, the reliability assessment of Cycle 2 is conducted to find the updated MPPs and to check the design feasibility. If some probabilistic constraints are still not satisfied, the procedure is repeated cycle by cycle until the objective converges and the reliability requirement is achieved when all the shifting distances become zero. Since the SORA method requires much less optimization iterations and reliability assessments to converge, the overall efficiency is high.

**4.2. Integration of SORA and Improved Reliability Analysis.** FORM is employed in SORA for reliability analysis. To improve the precision of reliability-based optimization for the highly nonlinear problem, we integrated the improved SVM-based reliability analysis into SORA. The flowchart of this optimization strategy is provided in Figure 6. Firstly, the SORA is operated to find the optimal solution, which satisfies the constraint requirement with the FORM for reliability analysis. Then, the optimal solution is verified by the improved SVM-based reliability analysis. If the approximation for the constraint in SORA could satisfy the accuracy requirement, the procedure stops. Otherwise, the SORA performs again to search a better result in feasible region after reliability index modification, until all the constraints satisfy the reliability requirement and the accuracy is satisfied. The details of the procedure are provided below.

**4.2.1. SORA.** SORA is the first step in the optimization procedure. The optimal solution  $\{\mathbf{d}^{SORA}, \boldsymbol{\mu}_X^{SORA}\}$  and its MPP location  $\mathbf{u}_{MPP}$  are found, which satisfy the requirement of reliability  $R$  with the FORM for reliability analysis.

Performance measure approach is used in SORA, the  $R$  percentile of optimal solution  $g^R$  is greater than or equal 0. If  $g^R$  is equal to 0, it means the constraint is active. If  $g^R$  is greater than 0, it means the constraint is inactive; that is, the design could move towards the constraint more.

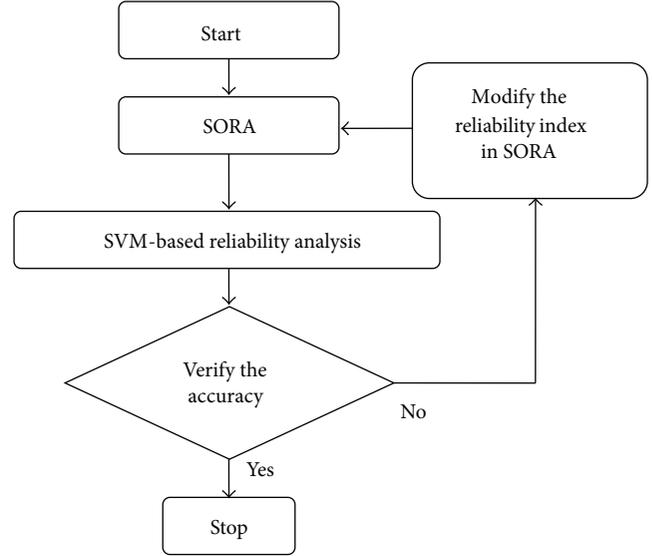


FIGURE 6: Integration of SORA and improved reliability analysis.

**4.2.2. SVM-Based Reliability Analysis and Check Accuracy.** After SORA, the optimal solution  $\{\mathbf{d}^{SORA}, \boldsymbol{\mu}_X^{SORA}\}$  is verified by the improved SVM-based reliability analysis, and its reliability  $R_{SVM}$  is achieved. Then,  $R_{SVM}$  should be compared to the required reliability  $R$  to check the accuracy of linearization approximation in SORA.

If the verification results show that accuracies of reliability analysis satisfy the requirement, namely,  $\|R_{SVM} - R\|/R$  is little enough, the procedure stops. If not, the SORA performs again to search a better result in feasible region after reliability index modification, until all the constraints satisfy the reliability requirement.

**4.2.3. Reliability Index Modification in SORA.** SORA is based on the FORM for reliability analysis.  $\beta_{SORA}$  indicates the reliability index corresponding to the desired reliability  $R$  in SORA, which is used to search the inverse MPP. Therefore, the reliability index is modified in the optimization instead of the reliability  $R$ . The new  $\beta_{SORA}$  in the next iteration is given by

$$\beta_{SORA}^{New} = \beta_{SORA} \cdot \frac{\beta}{\beta_{SVM}}, \quad (59)$$

where  $\beta_{SVM}$  is a generalized reliability index, evaluated with the reliability  $R_{SVM}$  as follows:

$$\beta_{SVM} = \Phi^{-1}(R_{SVM}). \quad (60)$$

Then, the SORA method is operated again with the new  $\beta_{SORA}$  and starts searching at last optimal solution. Since the new optimization is just to search a better design for a higher accuracy of reliability analysis based on the last optimal solution, only several iterations are needed for convergence.

## 5. Examples

**5.1. Example 1-A: Two-Dimensional Function.** In this mathematical example, the optimization problem is defined by

$$\begin{aligned}
& \text{Minimize: } f = -(X_1 + X_2), \\
& \text{Design variables: } X_1, X_2, \\
& \text{Subject to: } \Pr \{g = X_1^2 + X_2^2 - 2X_1X_2 - \sqrt{2}(X_1 + X_2) + 6 < 0\} < 1.0 \times 10^{-3},
\end{aligned} \tag{61}$$

where  $X_1$  and  $X_2$  follow the standard normal distribution. Their design intervals are  $[-4, 4]$ .

According to [32], the number of training points is gradually increased around the MPP used to create the SVM surrogate models with gradient information for this optimal design. Then, a second order polynomial kernel ( $d = 2$ ) and 50 training points could be used for this two-dimensional nonlinear problem. For making the training points closer to the limit-state boundary, these points are generated with truncated normal distributions in rotated  $U'$ -space. A larger truncation for  $U'_2$  than that for  $U'_1$  is used to shrink the samples in the  $U'_2$  direction. The distances are  $\Delta_1 = 1.0$  and  $\Delta_2 = 0.25$ , respectively.

The SORA method is operated to solve this probabilistic optimization followed by the improved SVM-based reliability analysis and modification. Because the proposed SORA + SVM method relies on random samples, the optimization procedure runs 5 times with different random seed numbers to ensure a meaningful comparison.

The results of the first step SORA with  $P_{f,\text{SORA}} < 1.0 \times 10^{-3}$  are shown in Line 2 of Table 1. The optimal solutions are  $(-0.638, -0.638)$  and the objective is  $f = 1.276$ . Take the first SORA + SVM optimization procedure as an example. After verification by the improved SVM-based reliability analysis, the probability of failure of the optimal design in SORA is  $P_{f,\text{SVM}} = 3.48 \times 10^{-4}$ . Not only does it meet the requirements of reliability, but there is a surplus. The design point could be closer to the constraint boundary. It is necessary to modify the reliability index and operate SORA again to search for a better result in the feasible region. After 3 iterations, the final optimal solutions are given in Line 3 of Table 1. It shows that a smaller objective is found in the feasible region by the improved sequential optimization strategy. During the SVM construction in this example, 3 or 4 support vectors are derived.

Monte Carlo simulation is chosen as a benchmark for the comparison of results of these two optimization strategies ((1) only SORA; (2) SORA + SVM). A large sample size of  $10^8$  is used for each of MCS run. As depicted in Table 1, the error of  $P_{f,\text{SORA}}$  is 189.9%. That is because FORM used in SORA just linearizes the limit-state function at the MPP. However, this problem is nonlinear. The error of  $P_{f,\text{SVM}}$  for the same design point analyzed with SVM is just 0.8%. After 3 iterations with modifications of the reliability index, the minimum objective  $f = -0.318$  is achieved. And, in the last iteration, SORA is operated with  $P_{f,\text{SORA}} < 2.78 \times 10^{-3}$  for correcting the error of FORM.

The results of other SORA + SVM optimization procedures with different random seed numbers are listed in Lines 4–7 of Table 1. These optimal solutions are very close to the

first one. The maximum error of  $P_{f,\text{SVM}}$  is 6.38% compared to Monte Carlo simulation, and the minimum is only  $-1.96\%$ .

**5.2. Example 2-A: Light Aircraft Wing.** For an aircraft wing design, the differences between design and real flight are mainly due to uncertainties in the structural geometry, material properties, operation conditions, and so on. Reliability-based design for the wing structure could reduce the risk and cost. In this subsection, the structure of a light aircraft wing is optimized with the proposed reliability-based optimization strategy [33]. The structural design aims to reduce structural weight by selecting the proper size of structural components to meet the requirements on allowable stresses, deformation limitations, and others.

The basic design requirements for the aircraft are that the takeoff weight of the aircraft is around 700 kg, the flight attitude is about 3000 m, the cruise speed is around 200 km/h, and the external shape of the wing is rectangular. The structure model of the wing is shown in Figure 7. The aspect ratio is 8.0; the reference area is  $10 \text{ m}^2$ ; and maximum thickness to chord ratio of the airfoil is 12%. The wing is structurally divided into 7 sections along the wingspan. Each section is a single box beam that consists of several components including the spar caps, the front shear web, the rear shear webs, and the skin. For each section, the bending stress in the spars, the shear stress in the skin, the shear stress in the front and rear webs, and the wing twist deformation are calculated using the beam theory [34].

The wing structure is subject to the aerodynamic loads. Since the light aircraft flies at a subsonic speed and its wing span ratio is relatively large, the lifting line method is used to predict the aerodynamic characteristics, including lift distributions, lift coefficients, and induced drag. The aerodynamic lift distributions are represented by a cubic polynomial function with the coefficients  $[0.52544, -0.33753, -1.04033, -1.30315]$ . The total drag is the sum of induced drag and parasite drag. In this problem, the parasite drag coefficient is assumed to be 0.015.

In this structural optimization problem, the objective is to minimize the structural weight of the aircraft wing  $W$ . The areas of spar caps at each section can be different while the thickness of the skin at each section remains the same. There are 10 design variables  $\mathbf{X} = (X_1, X_2, \dots, X_{10})$ , where  $X_i$  ( $i = 1, 2, \dots, 7$ ) is the area of spar cap in section and  $X_8, X_9, X_{10}$  are the thickness of the skin, the front web, and the rear web, respectively. These design variables and their lower and upper bounds are given in Table 2. Because of the manufacturing error, the design variables are supposed to follow the normal distribution, and the coefficient of variation is 0.05. Moreover, considering the uncertainty of flight condition, two parameters are supposed to be normal

TABLE 1: Results of the mathematical example.

	$X_1$	$X_2$	$f$	$P_{f,MCS}$	$P_{f,SORA}$	$P_{f,SVM}$	Error <sub>SORA</sub>	Error <sub>SVM</sub>
SORA	-0.6380	-0.6380	1.276	$3.45 \times 10^{-4}$	$1.00 \times 10^{-3}$	$3.48 \times 10^{-4}$	189.9%	0.8%
SORA + SVM1	0.1590	0.1590	-0.318	$1.01 \times 10^{-3}$	$2.78 \times 10^{-3}$	$9.74 \times 10^{-4}$	177.2%	3.5%
SORA + SVM2	0.1615	0.1615	-0.3230	$1.02 \times 10^{-3}$	$2.79 \times 10^{-3}$	$1.00 \times 10^{-3}$	173.53%	-1.96%
SORA + SVM3	0.1703	0.1703	-0.3407	$1.05 \times 10^{-3}$	$2.89 \times 10^{-3}$	$1.01 \times 10^{-3}$	175.24%	-3.81%
SORA + SVM4	0.1437	0.1437	-0.2874	$0.94 \times 10^{-3}$	$2.58 \times 10^{-3}$	$1.00 \times 10^{-3}$	174.47%	6.38%
SORA + SVM5	0.1509	0.1509	-0.3017	$0.97 \times 10^{-3}$	$2.66 \times 10^{-3}$	$1.00 \times 10^{-3}$	173.10%	2.67%

TABLE 2: Design variables of wing.

Design variables	$X_1, X_2, \dots, X_7$ (mm <sup>2</sup> )	$X_8$ (mm)	$X_9$ (mm)	$X_{10}$ (mm)	$W$ (kg)	$g^R$
Lower value	50.0	1.0	1.0	1.0	—	—
Upper value	700.0	2.0	1.0	2.0	—	—
Origin design	300.0	1.5	1.5	1.5	88.529	1.125
Optimal design	50, 56.5, 125.6, 221.0, 341.4, 483.8, 642.3	1.0	1.0	1.0	63.975	0.965

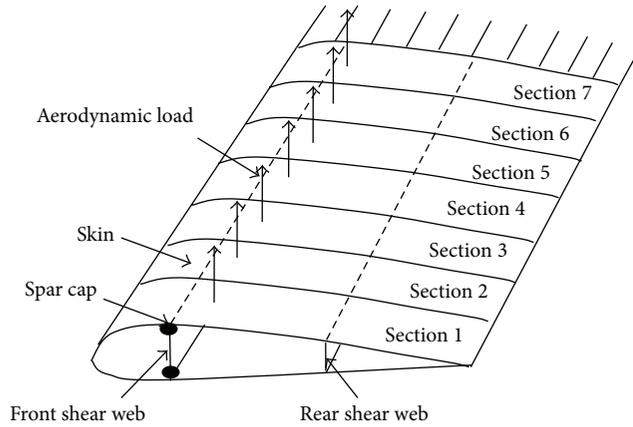


FIGURE 7: The wing structure model [33].

random variables including the flight speed  $V \sim N(200, 20)$  and the flight altitude  $H \sim N(3000, 300)$ .

This optimization problem contains eleven constraints. The bending stresses in the spar cap for each section  $\sigma_i$  ( $i = 1, 2, \dots, 7$ ) should be less than the bending strength of the material  $S_1$ . The maximum shear stress in the skin  $\tau_{skin}$  should be less than the shear strength of the skin  $S_2$ . The shear stress in the front web  $\tau_{fw}$  and the shear stress in the rear web  $\tau_{rw}$  should be less than the shear strength of the spar web  $S_3$ .  $[S_1, S_2, S_3]$  is given by [450, 200, 200] MPa.  $\theta$  is the twist deformation of the wing under the aerodynamic loads, and the allowable twist deformation is a random parameter with  $\theta_0 \sim N(2, 0.2)$ . The required reliability of this probabilistic constraint is 0.9987 corresponding to a reliability index  $\beta = 3$ .

Then, the reliability-based wing optimization problem is modeled as

$$\text{Given: } H \sim N(3000, 300), V \sim N(200, 20), \theta_0 \sim N(2, 0.2),$$

$$\text{Design variables: } \mathbf{X} = (X_1, X_2, \dots, X_{10}), \text{ coefficient of variation is } 0.05,$$

$$\text{Minimize: } W,$$

$$\text{Subject to: } \Pr \{g = \theta_0 - \theta \geq 0\} \geq 0.9987, \quad (62)$$

$$S_1 - \sigma_i \geq 0, \quad (i = 1, 2, \dots, 7),$$

$$S_2 - \tau_{skin} \geq 0,$$

$$S_3 - \tau_{fw} \geq 0, \quad S_3 - \tau_{rw} \geq 0.$$

The structural optimization of wing is solved by the SORA method with  $\beta = 3$  ( $R = 0.9987$ ) at first. The optimal design is listed in Table 2 as well. The weight of wing is reduced. As

shown in the table, the  $R$  percentile of optimal solution  $g^R$  is 0.965, larger than 0. It indicates that the reliability for this constraint is larger than the required one, and the constraint

TABLE 3: Reliability analysis of optimal solution.

	$R_{\text{SORA}}$	$R_{\text{SVM}}$	$P_{f,\text{SORA}}$	$P_{f,\text{SVM}}$	$P_{f,\text{MCS}}$	Error <sub>SORA</sub>	Error <sub>SVM</sub>
SORA + SVM1	0.99870	0.99869	$1.30 \times 10^{-3}$	$1.31 \times 10^{-3}$	$1.36 \times 10^{-3}$	-4.41%	-3.68%
SORA + SVM2	0.99870	0.99866	$1.30 \times 10^{-3}$	$1.34 \times 10^{-3}$	$1.36 \times 10^{-3}$	-4.41%	-1.47%
SORA + SVM3	0.99870	0.99871	$1.30 \times 10^{-3}$	$1.29 \times 10^{-3}$	$1.36 \times 10^{-3}$	-4.41%	-5.15%
SORA + SVM4	0.99870	0.99858	$1.30 \times 10^{-3}$	$1.42 \times 10^{-3}$	$1.36 \times 10^{-3}$	-4.41%	4.41%
SORA + SVM5	0.99870	0.99870	$1.30 \times 10^{-3}$	$1.30 \times 10^{-3}$	$1.36 \times 10^{-3}$	-4.41%	-4.41%

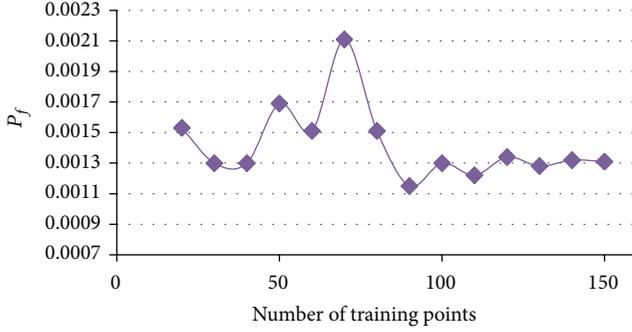


FIGURE 8: Convergence test.

is not active. The improved SVM-based reliability analysis is used to verify the accuracy. Gaussian kernel function ( $\sigma = 10$ ) is chosen in this problem.

Before verification of the optimal design of SORA, we gradually increase the number of training points around the MPP used to create the SVM surrogate models with gradient information for this optimal design. Figure 8 depicts the process of convergence. As shown in the figure, when the number of training points is larger than 100, the fluctuation of curve trends to be gentle. Therefore, 120 training points are used to construct the SVM surrogate models.

The improved SVM-based reliability analysis result is  $R_{\text{SVM}} = 0.99869$ , which is quite close to the required  $R$ . It demonstrates that the FORM in the SORA for reliability analysis satisfies the accuracy requirement. The modification for  $\beta$  and further searching are not needed.

Reliability of the optimal design was analyzed with the improved SVM-based reliability analysis method for 5 times with different random seed numbers to ensure a meaningful comparison. During the SVM construction in this example, 11–17 support vectors are derived. Monte Carlo simulation with  $10^6$  sample points is also used as a benchmark for the comparison of results of these two optimization strategy for this example. The comparison results are given in Table 3. Since the required reliability is very close to 1, the probability of failure  $P_f$  is chosen for comparison in order to reflect the accuracy of the two methods more clearly. As shown in the table, both the errors of SORA (based on FORM) and the improved SVM-based reliability analysis are very small. The error of SORA is only -4.41%. The largest error of the improved SVM-based reliability analysis is -5.15%, and the smallest is only -1.47%. It indicates that the constraint function for the twist deformation of wing is linear in this problem. And the error of the improved SVM-based

reliability analysis is quite small. It declares that the accuracy of this method is high, and it could be applied in both of the linear and nonlinear problems.

In this wing design example, the original function evaluates 438 times in SORA and 558 times in the SORA + SVM. The reliability analysis based on SVM surrogate model runs quite fast. If the Monte Carlo simulation is used to verify the accuracy of SORA,  $10^6$  sample points are needed. Therefore, computational cost of the proposed optimization strategy is moderate.

## 6. Conclusion

In this work, a reliability analysis method based on SVM (support vector machine) and MPP (Most Probable Point) is proposed. SVM is employed to create a surrogate model of the limit-state function at the MPP. The gradient information at the MPP is used to guarantee that the surrogate model passes through the MPP and that the surrogate model is tangent to the limit-state function at the MPP. This can greatly improve the sampling efficiency and improve the accuracy of the surrogate model.

Then, the SORA (Sequential Optimization and Reliability Assessment) and SVM-based reliability analysis are integrated. The SORA decoupled the traditional double-loop reliability optimization into a single loop strategy. SVM-based reliability analysis is used to amend the error from linear approximation for limit-state function in SORA.

The mathematical example demonstrates that SVM-based reliability analysis is more accurate than the FORM (in SORA) and more efficient than the direct Monte Carlo simulation. The proposed reliability-based optimization is applied to a simplified wing structure design under these uncertainties as well. Results show that the weight of the wing is reduced and that all deterministic and probabilistic constraints are satisfied. These examples indicate that accuracy of the proposed SVM-based reliability analysis is high for either linear or nonlinear problem. And the proposed optimization strategy is accurate and its computational cost is moderate.

There are, however, some potential problems associated with the MPP-based SVM method. If multiple MPPs exist, the method could be remedied by incorporating the multiple MPPs in the SVM implementation. Similar to the other SVM methods, it is not straightforward to determine the parameters for a kernel function, for example, the order of a polynomial kernel and the standard deviation of a Gaussian kernel. If the dimension, or the number of random variables, is large, the MPP-based SVM method may not be as good

as the MPP-based importance sampling method because a large number of training points will be required to generate an accurate surrogate model.

## Nomenclature

$b$ :	Bias
$c$ :	Gradient of limit-state function at the MPP
$\mathbf{d}$ :	Vector of deterministic design variables
$e$ :	Proportionality factor
$F$ :	Cumulative distribution function
$f$ :	Joint probability density function; objective function
$g^R$ :	$R$ -percentile of $g(\mathbf{X}) \geq 0$
$g(\mathbf{X})$ :	Limit-state function in $X$ space
$g(\mathbf{U})$ :	Limit-state function in $U$ space
$\tilde{g}(\mathbf{U})$ :	Approximated limit-state function in $U$ space
$H$ :	Flight altitude
$h$ :	Importance sampling density
$I$ :	Indicator function
$K(\mathbf{x}_i, \mathbf{x}_j)$ :	kernel function
$L$ :	Lagrangian function
$m$ :	Number of samples
$n$ :	Length of vector of random variables
$N$ :	Number of samples; standard normal distribution
$\mathbf{p}$ :	Vector of random design parameters
$P_f$ :	Probability of failure
Pdf:	Probability density function
$R$ :	Reliability
$R_i$ :	Desired probability of constraint satisfaction
$\mathbf{R}$ :	Rotation matrix
$\mathbf{s}$ :	Shifting vector
$S_1$ :	Bending strength
$S_2, S_3$ :	Shear strength
$\mathbf{u}^*$ :	MPP in $U$ space
$\mathbf{U}$ :	Vector of random variables in $U$ space
$U_i$ :	$i$ th random variable in $U$ space
$V$ :	Flight speed
$\mathbf{w}$ :	Weight vector
$W$ :	Structural weight of the aircraft wing
$\mathbf{X}$ :	Vector of random variables in $X$ space
$X_i$ :	$i$ th random variable in $X$ space
$y_i$ :	$i$ th output
$\beta$ :	Reliability index
$\mu$ :	Mean value of the random design variable
$\sigma$ :	Parameter in Gaussian kernel, bending stress
$\lambda, \gamma$ :	Lagrangian multiplier
$\Phi$ :	Standard normal distribution
$\varphi(\mathbf{x})$ :	Feature function
$\nabla g$ :	Gradient of $g$
$\Delta$ :	Truncated interval
$\tau$ :	Shear stress
$\theta$ :	Twist deformation of the wing.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

This work is supported by the Intelligent Systems Center at Missouri University of Science and Technology and the University of Missouri Research Board (7116), the National Natural Science Foundation of China (11432007), and the Fundamental Research Funds for the Central Universities (NUAA NS2015098). The supports are gratefully acknowledged.

## References

- [1] T. Zang, M. J. Hemsch, M. W. Hilburger et al., "Needs and opportunities for uncertainty-based multidisciplinary design methods for aerospace vehicles," NASA Report TM-2002-211462, 2002.
- [2] M. Allen and K. Maute, "Reliability-based design optimization of aeroelastic structures," *Structural and Multidisciplinary Optimization*, vol. 27, no. 4, pp. 228–242, 2004.
- [3] S. Mahadevan, "Physics-based reliability models," in *Reliability-Based Mechanical Design*, T. A. Cruse, Ed., pp. 197–232, Dekker, New York, NY, USA, 1997.
- [4] A. M. Hasofer and N. C. Lind, "Exact and invariant second-moment code format," *ASCE Journal of the Engineering Mechanics Division*, vol. 100, no. 1, pp. 111–121, 1974.
- [5] R. Rackwitz and B. Flessler, "Structural reliability under combined random load sequences," *Computers & Structures*, vol. 9, no. 5, pp. 489–494, 1978.
- [6] M. Shinozuka, "Basic analysis of structural safety," *Journal of Structural Engineering*, vol. 109, no. 3, pp. 721–740, 1983.
- [7] L. Tvedt, "Distribution of quadratic forms in normal space—application to structural reliability," *Journal of Engineering Mechanics*, vol. 116, no. 6, pp. 1183–1197, 1990.
- [8] Y. Shin, "Improving probabilistic damage tolerance analysis for inspection optimization: possibilistic-probabilistic approach," in *Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Victoria, Canada, September 2008.
- [9] F. Li and T. Wu, "An importance sampling based approach for reliability analysis," in *Proceedings of the 3rd Annual IEEE Conference on Automation Science and Engineering (CASE '07)*, pp. 956–961, IEEE, Scottsdale, Ariz, USA, September 2007.
- [10] A. Der Kiureghian, H.-Z. Lin, and S.-J. Hwang, "Second-order reliability approximations," *ASCE Journal of Engineering Mechanics*, vol. 113, no. 8, pp. 1208–1225, 1987.
- [11] B. D. Youn and K. K. Choi, "A new response surface methodology for reliability-based design optimization," *Computers and Structures*, vol. 82, no. 2-3, pp. 241–256, 2004.
- [12] P. R. Adduri and R. C. Penmetsa, "Confidence bounds on component reliability in the presence of mixed uncertain variables," *International Journal of Mechanical Sciences*, vol. 50, no. 3, pp. 481–489, 2008.
- [13] J. E. Hurtado and D. A. Alvarez, "Neural-network-based reliability analysis: a comparative study," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 1-2, pp. 113–132, 2001.

- [14] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.
- [15] J. E. Hurtado, "An examination of methods for approximating implicit limit state functions from the viewpoint of statistical learning theory," *Structural Safety*, vol. 26, no. 3, pp. 271–293, 2004.
- [16] C. M. Rocco and J. A. Moreno, "Fast Monte Carlo reliability evaluation using support vector machine," *Reliability Engineering and System Safety*, vol. 76, no. 3, pp. 237–243, 2002.
- [17] A. Basudhar, S. Missoum, and A. H. Sanchez, "Limit state function identification using Support Vector Machines for discontinuous responses and disjoint failure domains," *Probabilistic Engineering Mechanics*, vol. 23, no. 1, pp. 1–11, 2008.
- [18] H.-S. Li, Z.-Z. Lü, and Z.-F. Yue, "Support vector machine for structural reliability analysis," *Applied Mathematics and Mechanics*, vol. 27, no. 10, pp. 1295–1303, 2006.
- [19] H. Song, K. K. Choi, I. Lee, L. Zhao, and D. Lamb, "Adaptive virtual support vector machine for reliability analysis of high-dimensional problems," *Structural and Multidisciplinary Optimization*, vol. 47, no. 4, pp. 479–491, 2013.
- [20] Y. T. Wu, Y. Shin, R. Sues, and M. Cesare, "Safety-factor based approach for probabilistic-based design optimization," in *Proceedings of the 42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference and Exhibit*, Seattle, Wash, USA, 2001.
- [21] X. Chen, T. K. Hasselman, and D. J. Neill, "Reliability based structural design optimization for practical applications," in *Proceedings of the 38th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference and Exhibit and AIAA/ASME/AHS Adaptive Structural Forum*, pp. 2724–2732, Kissimmee, Fla, USA, April 1997.
- [22] X. Du and W. Chen, "Sequential optimization and reliability assessment method for efficient probabilistic design," *Journal of Mechanical Design*, vol. 126, no. 2, pp. 225–233, 2004.
- [23] M. Rosenblatt, "Remarks on a multivariate transformation," *Annals of Mathematical Statistics*, vol. 23, pp. 470–472, 1952.
- [24] B. C. Lovell and C. J. Walder, "Support vector machines for business application," in *Business Applications and Computational Intelligence*, chapter 14, Idea Group, Hershey, Pa, USA, 2006.
- [25] B. Schölkopf, "Statistical learning and Kernel methods," Tech. Rep. MSR-TR-2000-23, Microsoft Research, 2000.
- [26] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*, Cambridge University Press, Cambridge, UK, 2000.
- [27] O. Ivanciuc, "Applications of support vector machines in chemistry," *Reviews in Computational Chemistry*, vol. 23, pp. 291–400, 2007.
- [28] Y. Zhang and A. der Kiureghian, "Two improved algorithms for reliability analysis," in *Proceedings of the 6th IFIP WG7.5 Working Conference on Reliability and Optimization of Structural Systems*, Assisi, Italy, September 1994.
- [29] N. L. Johnson and S. Kotz, *Continuous Univariate Distributions—Volume 1*, John Wiley & Sons, New York, NY, USA, 1970.
- [30] R. L. Iman, J. C. Helton, and J. E. Campbell, "An approach to sensitivity analysis of computer models, Part 1. Introduction, input variable selection and preliminary variable assessment," *Journal of Quality Technology*, vol. 13, no. 3, pp. 174–183, 1981.
- [31] M. Lazaro, I. Santamaria, F. Perez-Cruz, and A. Artes-Rodriguez, "Support vector machine for the simultaneous approximation of a function and its derivative," in *Proceedings of the IEEE 13th Workshop on Neural Networks for Signal Processing (NNSP '03)*, pp. 189–198, Toulouse, France, September 2003.
- [32] Y. Wang, X. Yu, and X. Du, "Reliability analysis with SVM and gradient information at MPP," in *Proceedings of the 7th China-Japan-Korea Joint Symposium on Optimization of Structural and Mechanical Systems*, Huangshan, China, June 2012.
- [33] X. Yu and X. Du, "Reliability-based multidisciplinary optimization for aircraft wing design," *Structure and Infrastructure Engineering: Maintenance, Management, Life-Cycle Design and Performance*, vol. 2, no. 3-4, pp. 277–289, 2006.
- [34] D. H. Allen and W. E. Haisler, *Introduction to Aerospace Structural Analysis*, John Wiley & Sons, New York, NY, USA, 1984.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

