Rapid #: -22236109

CROSS REF ID: **354610**

LENDER: **EUX (University of Edinburgh) :: Main Library**

BORROWER: **EYR (Oakland University) :: Kresge Library**

TYPE:	Book Chapter
BOOK TITLE:	ASME 2007 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference
USER BOOK TITLE:	ASME 2007 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference
CHAPTER TITLE:	A Monte Carlo Reliability Assessment for Multiple Failure Region Problems Using Approximate Metamodels
BOOK AUTHOR:	Kuczera, R.C.,
EDITION:	
VOLUME:	
PUBLISHER:	
YEAR:	2007
PAGES:	1135-
ISBN:	9780791848074
LCCN:	
OCLC #:	

Processed by RapidX: 3/15/2024 3:30:53 AM

Proceedings of the ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2007
September 4-7, 2007, Las Vegas, Nevada, USA

DETC2007-34957

A MONTE CARLO RELIABILITY ASSESSMENT FOR MULTIPLE FAILURE REGION PROBLEMS USING APPROXIMATE METAMODELS

Ramon C. Kuczera
GKN Driveline North America
3300 University Drive, Auburn Hills, MI 48348

Zissimos P. Mourelatos
Mechanical Engineering Department
Oakland University, Rochester, MI 48309
Phone: 248-370-2686
Fax: 248-370-4416
Email: mourelat@oakland.edu

Michael Latcha
Mechanical Engineering Department
Oakland University, Rochester, MI 48309

ABSTRACT

An efficient Monte Carlo reliability assessment methodology is presented for engineering systems with multiple failure regions and potentially multiple most probable points. The method can handle implicit, nonlinear limit-state functions, with correlated or non-correlated random variables, which can be described by any probabilistic distribution. It uses a combination of approximate or "accurate-on-demand," global and local metamodels which serve as indicators to determine the failure and safe regions. Samples close to limit states define transition regions between safe and failure domains. A clustering technique identifies all transition regions which can be in general disjoint, and local metamodels of the actual limit states are generated for each transition region. A Monte Carlo simulation calculates the probability of failure using the global and local metamodels. A robust maximin "space-filling" sampling technique is used to construct the metamodels. Also, a principal component analysis addresses the problem dimensionality making therefore, the proposed method attractive for problems with a large number of random variables. Two numerical examples highlight the accuracy and efficiency of the method.

INTRODUCTION

The design of any engineering system requires the assurance of its reliability and quality. Variations and uncertainties in the inputs and properties of an engineering system cause variations in its performance. Uncertainties in the system characteristics prevent such assurances from being given with absolute certainty. One common approach is to quantify the reliability of performance, or risk of failure, in probabilistic terms.

For large-scale systems, the reliability prediction is usually based on efficient computational methods. Both analytical and simulation-based methods are available. The analytical methods are based on the most probable point (MPP) concept. They include the well known first-order reliability method (FORM) which has been widely used [1, 2], second-order reliability methods (SORM) [3-5], and multi-point approximation methods [6, 7]. Among the simulation-based methods, the Monte Carlo (MC) method is very simple and accurate. However, its computational cost is prohibitively high. For this reason, more efficient simulation-based techniques have been proposed [8-11]. Among them, the adaptive importance sampling (AIS) techniques have been popular [8, 9]. Also, multi-modal importance sampling methods have been proposed where the sampling density emphasizes all important sample points in the failure domain, each in proportion to the true probability density [10, 11]. A combination of analytical and simulation-based methods has also been used [12]. The analytical methods are generally simple and efficient, but for complex problems, their accuracy cannot be guaranteed. In

simulation-based methods, the accuracy can be controlled but the efficiency is generally not satisfactory.

Various math-based methods can be used to estimate the probability of failure using defined limit state functions, which separate the failure and the safe regions. In the probabilistic analysis of complex engineering systems, the limit state functions are in general, implicit and nonlinear. Furthermore, each function evaluation is usually computationally expensive. For these reasons, it is important to choose a method that minimizes the number of function evaluations in estimating the probability of failure without sacrificing accuracy.

In this paper, an accurate and efficient computational method is presented for reliability assessment of engineering systems. It provides a substantial improvement over our previous work in [13] because it can address problems with multiple MPPs with disjoint failure domains. It can also be used for system reliability assessment of problems with multiple limit states. The method can easily handle implicit, nonlinear limit-state functions, with correlated or non-correlated random variables, which are described by any probabilistic distribution. It is based on "accurate-on-demand" global and local metamodels (response surfaces) of the limit states which serve as indicators to determine the "failure" and "safe" regions. High accuracy of these metamodels is not needed away from the limit states because they are simply used as "failure" indicators. However, improved accuracy is necessary close to the limit states where sample points have a "low" absolute value. In this work, points with "low" limit-state absolute value, define a transition region between the safe and failure regions. All samples in the transition region are identified, and more accurate *local* metamodels of each actual limit state are generated. The local metamodels are used to determine if the transition region points are in the safe or failure domains. Finally, a Monte Carlo simulation (MCS) calculates the probability of failure.

If a limit state has multiple failure regions which may be even disjoint, or if multiple limit states exist due to a system reliability assessment, the transition region points may be grouped in clusters. A clustering technique is therefore, used to identify these groupings so that we can build local metamodels for each group. This is essential for handling multiple MPP problems which often exist in a variety of engineering applications such as vibratory problems, for example. Recently, some work has been reported for multiple MPP problems [14-16].

Although different metamodeling techniques can be successfully used in this work, we have chosen the Cross-Validated Moving Least Squares (CVMLS) [17] method. To construct a metamodel, a so-called "space-filling" sampling algorithm is needed. However, the commonly used algorithms such as Latin Hypercube (LH) and Optimal Symmetric LH (OSLH) sampling [18] among others, place most of the samples in the interior of the domain. To avoid this problem, we use a maximin "space-filling" sampling technique. We also use a Principal Component Analysis (PCA) [19] in order to address the problem dimensionality when generating the local metamodels. This makes the proposed method attractive for problems with a large number of random variables. Two numerical examples highlight the accuracy and efficiency of the proposed method.

The proposed method is the first step towards developing a gradient-free, simulation-based (not analytical) reliability-based design optimization (RBDO) algorithm which can handle "noisy" limit state problems with multiple failure regions.

DESCRIPTION OF PROPOSED METHOD

It has been mentioned that in simulation-based reliability methods, we need the sign of the limit state function value, and not its actual function value. This is achieved in this paper by constructing approximate metamodels which can efficiently identify the safe (high value) and failure (low value) regions. A similar idea has been reported in [14] using an indicator response surface and in [15] using discriminative sampling.

The entire sampling region is divided into a safe region, a failure region and a transition region between the safe and failure regions. The safe and failure regions are easily identified using an approximate global metamodel $\tilde{g}_G(\mathbf{x})$ which is also used to identify the transition region. Details are provided in section 2.4.

The samples in the transition region are evaluated using a more accurate metamodel. For that, a local metamodel $\tilde{g}_L(\mathbf{x})$ is constructed. Good accuracy is ensured by 1) sampling in a lower dimensional space, obtained using PCA, 2) by a series of refinements using a "space-filling" algorithm which places samples very close to the limit state within the transition region and 3) by constructing a metamodel of a small, local region. The transition region includes all samples **x**, such that $\tilde{g}_G(\mathbf{x}) \in (h_l, h_u)$ where h_u and h_l are the upper and lower bound values respectively, estimated using the global metamodel.

An efficient "space-filling" sampling is used based on a maximin distance algorithm, which efficiently selects a few samples with "space-filling" properties, from a previously selected large MC set of samples. Details are presented in section 2.1.

The probability of failure p_f is calculated as,

$$p_f = \frac{n_f}{N} = \frac{1}{N}\sum_{j=1}^{N} I(\mathbf{x}_j) \qquad (1)$$

where $\mathbf{x}_j, j = 1,...,N$ is a MC sample, n_f is the number of failures out of N samples, and $I(\mathbf{x})$ is an indicator function. If **x** is in the safe region ($\tilde{g}_G(\mathbf{x}) > h_u$) or in the transition region ($\tilde{g}_L(\mathbf{x}) \in (h_l, h_u)$ and $\tilde{g}_L(\mathbf{x}) > 0$), $I(\mathbf{x})$=0. Also, if **x** is in the failure region ($\tilde{g}_G(\mathbf{x}) < h_l$) or in the transition region ($\tilde{g}_L(\mathbf{x}) \in (h_l, h_u)$ and $\tilde{g}_L(\mathbf{x}) < 0$), $I(\mathbf{x})$=1.

The efficiency in constructing a local metamodel of the transition region, is greatly enhanced by a dimension reduction technique using PCA. The samples in the transition region

exhibit geometric correlation because they are placed close to the limit state. This correlation is exploited using PCA. Assuming that each sample is *k*-dimensional, *q* principal directions, where $q<k$, are calculated which account for most of the variation in the location of all samples. Subsequently, samples are mostly spread only along the *q* principal directions instead of the original *k* dimensions. This addresses the so-called "curse of dimensionality," improving therefore, the efficiency of the local metamodel construction. Details are provided in sections 2.3 and 2.4.

A MAXIMIN SAMPLING METHOD

An efficient and accurate construction of a metamodel requires a so-called "space-filling" sampling algorithm. A variety of such algorithms have been reported in the literature including Latin Hypercube (LH) sampling and its variations such as randomized orthogonal arrays, symmetric LH sampling, optimal LH sampling and Optimal Symmetric LH (OSLH) sampling. OSLH has been extensively used [18] due its desirable uniform projection properties. However, OSLH has the tendency of placing most of the samples in the interior of the domain. This is especially true for moderate to high-dimensional problems with a relatively small number of available samples. In such cases, the metamodel is forced to extrapolate in order to predict the response of a boundary point. Such a prediction is usually inaccurate. In this paper, we avoid this problem by using a sampling algorithm which places samples both on the boundary and the interior of the domain. Although this requires a larger number of samples, it is necessary from the accuracy point of view.

We use an easy to implement maximin method in which the addition of new samples preserves by default, the "space-filling" properties. This is a very important property which allows us to construct "converged" metamodels with a small number of samples. Convergence is practically achieved if the reconstructed metamodel after a few more samples are added, gives similar predictions with the previous version. Similar predictions are judged based only on the sign, and not the estimated actual value, of the limit state.

Consider a design $D_n = \{\mathbf{x}_1, \ \mathbf{x}_2, \ \cdots \ \mathbf{x}_n\}$ which is composed of a collection of *k*-dimensional samples $\mathbf{x}_i \in \Re^k, i = 1,...,n$. A few judiciously selected samples from D_n define the maximin distance design *D* where, $D \subseteq D_n$. Let $d(\mathbf{u},\mathbf{v})$ be the Eucledian distance between samples **u** and **v**, where $\mathbf{u},\mathbf{v} \in D$. Design *D* simply maximizes the minimum inter-site distance $\min_{\mathbf{u},\mathbf{v}\in D} d(\mathbf{u},\mathbf{v})$, i.e.

$$\max_{D} \min_{\mathbf{u},\mathbf{v}\in D} d(\mathbf{u},\mathbf{v}). \qquad (2)$$

More information on maximin and minimax distance criteria is provided in [20, 21]. Ref. [22] discusses their use in metamodel construction.

In this paper, the maximin sampling technique of Eq. (2) is implemented in a straightforward way. Assume that a design D_n of *n*, *k*-dimensional samples is available, consisting of *n* MC samples. Each of the *k* components of the *n* MC samples is normalized between zero and one in order to account for potentially different units. If *x* denotes the k^{th} component, its normalized value is $x^* = (x - x^L)/(x^H - x^L)$, where x^L and x^L are the low and high values respectively, among all *n* samples.

Point $\mathbf{x}_1^* = \{\bar{x}_1^* \ \cdots \ \bar{x}_i^* \ \cdots \ \bar{x}_k^*\}$, where $\bar{x}_i^*, i = 1,...,k$ is the mean x_i^* coordinate of all *n* normalized points, is first selected as a "seed" point. Then the distance of all *n* normalized points from $\mathbf{x}_1^*$ is calculated and the point $\mathbf{x}_2^*$ with the largest distance is selected. Subsequently, the distances of the remaining (*n*-2) samples from both $\mathbf{x}_1^*$ and $\mathbf{x}_2^*$ are calculated and the point $\mathbf{x}_3^*$ with the maximum minimum distance is selected. The process is repeated *m* times in order to create the maximin distance design *D* with *m* samples. It should be noted that the method first places samples on the boundary of the domain. As more samples are created, they are placed in the interior of the domain providing therefore, a "space-filling" design. A very useful property of the method is that the addition of new samples preserves the "space-filling" properties of the new design. This is essential in order to keep the number of function evaluations low.

DATA CLUSTERING

Data clustering is a technique that is used to determine similarities in data and to segregate the data into groups based on those similarities. In this paper, the data is sorted using a Euclidian distance measure. Points that are in the same "neighborhood" are grouped together. For problems with multiple disjoint failure regions, clustering collects points into groups allowing us therefore, to create separate local metamodels for each failure region.

Given a set of N points to be clustered, an N by N distance matrix is formed with elements $d(\underline{x}^i, \underline{x}^j) = \sqrt{(\underline{x}^i - \underline{x}^j)^2}$ representing the Euclidian distance between points x^i and x^j. Subsequently, a hierarchical cluster tree is created using the following steps [23].

1. Assign each point to a cluster, so that N clusters are formed with one point each. The distance between the clusters is therefore, the same with the distance between the points they contain.

2. Find the closest pair of clusters and merge them into a single cluster, so that we have one cluster less.

3. Compute the distances between the new cluster and each of the old clusters.

4. Repeat steps 2 and 3 until all points are potentially clustered into a single cluster of size N. There is obviously no point in having all N points grouped in a single cluster. However, after we have a complete hierarchical tree, k clusters can be formed by simply cutting the *k* - 1 longest links.

Step 3 can be performed in different ways, which is what distinguishes single-linkage from complete-linkage and

average-linkage clustering. In single-linkage clustering, the distance between two clusters is equal to the shortest distance from any point of one cluster to any point of the other cluster. In this work, the single-linkage clustering is used. The described hierarchical clustering is also called agglomerative because it merges clusters iteratively.

PRINCIPAL COMPONENT ANALYSIS

Principal Component Analysis (PCA) [19] is a well known statistical method for finding trends in data. It is a very popular technique for dimensionality reduction and is discussed at length in most texts on multivariate analysis. It has been used in many application areas including data compression, image analysis, visualization, pattern recognition, regression and time series prediction. Here, PCA is used to perform a sort of dimension reduction by sampling only in identified important directions, and perturbing the sampled points in the unimportant directions. This reduces the number of samples because the points do not need to evenly span each dimension.

The PCA provides a linear projection which maximizes the variance in the projected space identifying therefore, the directions along which the data possesses the largest variation. For a set of *N* available *k*-dimensional samples $\mathbf{x}_n, n=1,...,N$, the *q* principal axes $\mathbf{w}_j, j=1,...,q$ form a set of orthonormal axes onto which the retained variance under projection is maximum. It can be shown that the vectors $\mathbf{w}_j$ are defined by the *q* dominant eigenvectors corresponding to the largest eigenvalues λ_j, of the sample covariance matrix $[S]=\sum_n (\mathbf{x}_n-\bar{x})(\mathbf{x}_n-\bar{x})^T/N$, where $\bar{x}$ is the data sample mean. They therefore, satisfy the relation $[S]\mathbf{w}_j=\lambda_j\mathbf{w}_j$. The *q* principal components of the observed vector $\mathbf{x}_n$ are given by the vector

$$\mathbf{y}_n=[W]^T(\mathbf{x}_n-\bar{x}), \qquad (3)$$

where $[W]=\lfloor \mathbf{w}_1,\ \mathbf{w}_2,\ \cdots,\ \mathbf{w}_q \rfloor$. The variables y_j are uncorrelated such that the covariance matrix $\sum_n \mathbf{y}_n\bar{\mathbf{y}}_n^T/N$ is diagonal with elements λ_j. The value of $\lambda_j/\sum_k \lambda_j$ gives the proportion of variation explained by the j^{th} principal component. Geometrically, the principal components are the axes of a new coordinate system obtained by rotating the axes of the original system. The new axes represent the directions of maximum variability.

PCA is often used to reduce the dimension of a data set, replacing a large number of correlated variables with a smaller number of orthogonal variables which still contain most of the information in the original data set. In this work, $\mathbf{x}_n, n=1,...,N$ includes all *N* samples of the transition region.

ALGORITHM OF THE PROPOSED METHOD

The beginning of section 2 provided an overview of the proposed method. This section provides all algorithmic details. The construction of an approximate global metamodel is described first (steps 1 through 7). Subsequently, a clustering technique is used which identifies potentially disjoint failure domains. The construction of local metamodels, including a series of refinements, is explained in steps 9 through 13. Finally, the probability of failure is calculated in step 14.

1. Generate *N* MC samples according to the statistical distribution of all random variables. Eq. (4) [24] estimates *N* by keeping the error bound less than $\varepsilon\%$ with confidence level ρ. In this paper, we use $\varepsilon=10$ and $\rho=0.05$.

$$\varepsilon\% = 100\Phi^{-1}\left(1-\frac{\rho}{2}\right)\sqrt{\frac{(1-P_f^T)}{NP_f^T}} \qquad (4)$$

2. Using the maximin method of section 2.1, *m* samples are selected from the available *N* MC samples of step 1. Although *m* is kept small, it is at least equal to *n*+1 where *n* is the number of variables.
3. Calculate the limit state value for all *m* points of step 2 and construct an approximate global metamodel $\tilde{g}_G(\mathbf{x})$. A different global metamodel is constructed for each limit state. The choice of the metamodeling technique may slightly affect the efficiency of the proposed method. It is not however, vital to the success of the method. Note that $\tilde{g}_G(\mathbf{x})$ covers only the "cloud" area of the MC samples from step 1.
4. Use $\tilde{g}_G(\mathbf{x})$ to predict the value of each limit state for the *N* MC samples.
5. Identify the upper and lower limits h_u and h_l respectively, of the transition region. The *N* values of step 4 are sorted in decreasing order. If $\max\tilde{g}_G$ and $\min\tilde{g}_G$ represent the maximum and minimum values, h_u and h_l are equal to $h_u=\eta(\max\tilde{g}_G)$ and $h_l=\eta(\min\tilde{g}_G)$, where η is a predetermined small percentage, so that the number of points in the transition region is less than ten percent of *N*. By limiting the number of transition region points to less than ten percent of *N*, we can create a sufficiently small local metamodel with good accuracy, while keeping the number of required function evaluations low. The transition region includes all MC samples **x** so that $h_l\le\tilde{g}_G(\mathbf{x})\le h_u$. Their number is denoted by *NT*. It should be emphasized that *NT* includes the transition points from all limit states.
6. Define the safe region and the failure region. The former includes all MC samples **x** so that $\tilde{g}_G(\mathbf{x})>h_u$ and the latter includes all MC samples with $\tilde{g}_G(\mathbf{x})<h_l$. The number of points in the failure region is n_G. For

the points in the safe and failure regions, the indicator function is assigned a value of 0 and 1, respectively (see Eq. 1).

7. Add *p more* points to the *m* points of step 2, and go to step 3. Repeat steps 3 to 6 until convergence is achieved. We have convergence, if the number of failure region points from step 6, does not change more than ten percent between two successive iterations. This "crude" convergence provides a compromise between accuracy and efficiency for the approximate global metamodels. It also defines the transition zone with enough accuracy for the next steps. Using Eq. (4) and the estimated probability of failure, we can determine if enough number of samples have been generated. If not, additional MC samples are generated.

At this point, we assume that the samples in the identified safe and failure regions have indeed a positive and negative value respectively. However due to the potentially low accuracy of the approximate global metamodels from step 7, we are not certain about the positive or negative sign of the transition points. For this reason, local metamodels of the transition region points are developed which are expected to be more accurate because of their smaller domains. A clustering technique determines the number of local metamodels.

If a limit state has multiple failure regions or if multiple limit states exist due to a system reliability assessment, the *NT* transition points from all "converged" global metamodels of step 7, may form clusters. The clustering technique of section 2.2 is then used in step 8 to identify potential clusters.

8. Using all points in the transition region, clustering groups them in *k* clusters. Because we do not know the number of existing clusters a priori, we start with a relatively large number of clusters. Depending on how many points are grouped in each of the *k* clusters, we easily determine the number of existing clusters k_c where $k_c < k$. Subsequently, clustering is repeated with only k_c clusters.

Even the most accurate metamodel can not accurately predict the sign of points very close to the limit state. To alleviate this problem, we refine each local metamodel by adding samples close to the limit state. For a balance between accuracy and efficiency (computational effort) of the local metamodels, considering the "curse of dimensionality," we also use a PCA-based, dimension reduction measure according to section 2.3.

The *NT* points in the transition region (see step 5) are expected to be along the limit state(s), exhibiting therefore, a geometric correlation due to their proximity to a limit state. PCA exploits this correlation. It should be noted that the less nonlinear the limit state is around the most probable point (MPP), the stronger the correlation is. The PCA will define a new, low-dimensional coordinate system for the transition region points using the dominant principal coordinates (see section 2.3). The dimension reduction will allow us to build the local metamodels with fewer "space-filling" samples. Steps 9 through 13 describe how a "converged" local metamodel is obtained for the k^{th} cluster.

9. Perform a PCA using the *NT(k)* transition region samples of the k^{th} cluster. All original *NT(k)* points are then projected onto the dominant directions and their coordinates are normalized between zero and one.

The geometric variation of points along the neglected, insignificant principal coordinates is very low. However, it has a strong influence on the probability of failure. For example, two points which are geometrically very close but in the opposite sides of a limit state, will have a positive and negative value, respectively. To account for this essential effect, the samples of step 9 are randomly perturbed along the insignificant principal coordinates.

It has been mentioned in section 2.1, that the maximin algorithm chooses among a large number of *n* samples which are normalized between zero and one. The projected *NT(k)* points onto the dominant principal coordinates are normalized between zero and one and subsequently, perturbed along the insignificant principal coordinates between zero and 0.1. When sampling is therefore performed, the samples will "space-fill" the significant coordinate space but will also vary geometrically along the insignificant coordinates.

Fig. 1 shows an example of a two-dimensional space where the horizontal and vertical directions are significant and insignificant, respectively. The hash marks indicate transition region points which are projected onto the horizontal direction and subsequently, perturbed along the vertical direction. The maximin algorithm was used to select 30 points, indicated by the diamonds, which "space-fill" the entire domain. Because the distance between points is uniform, they are mostly spread along the significant direction, providing however, some variation along the insignificant direction.

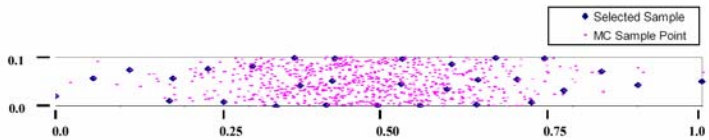

Figure 1. Hypothetical example with one significant and one insignificant principal direction

10. Perturb the projected *NT(k)* transition points from step 9, along each insignificant direction and normalize their coordinates between 0 and 0.1. A uniformly distributed random perturbation is used. Specifically for each point, the coordinates in the dominant directions are left unchanged and the coordinates in the insignificant directions are discarded and replaced with a uniformly distributed random number between 0.0 and 0.1. By decreasing the magnitude in the insignificant directions, the maximin approach to select points in the next step will mainly fill the space in the dominant directions.

11. A "converged" local metamodel of the *NT(k)* transition points is built for each limit state using the iterative process of steps 3 through 7. Points are successively selected from the *perturbed NT(k)* points of step 10, using the maximin approach. At each iteration, the current version of the local metamodel is used to evaluate all *NT(k)* points after they are transformed back to the original coordinates, and identify the number *mm(k)* of the failed points; i.e points **x** with $\tilde{g}_L(\mathbf{x}) < 0$. Convergence is achieved if *mm(k)* does not change more than 5% from the previous iteration. A "converged" local metamodel is achieved with $n_L(k)$ points.
12. The converged local metamodel from step 11 is used to identify $NT_L(k)$ points **x** from the group of the *NT(k)* transition points, so that $\alpha(\max \tilde{g}_L) \le \tilde{g}_L(\mathbf{x}) \le \alpha(\min \tilde{g}_L)$, a=0.1. These points are located very close to the limit state having therefore, a small absolute value.
13. At this point, a series of refinements for each local metamodel is performed. The maximin approach is used to identify *m* out of the $NT_L(k)$ points of step 12 which are added to the existing $n_L(k)$ points of step 11, and the local metamodel is updated. Again, all *NT(k)* points are evaluated using the updated local metamodels and a new number *mm(k)* of failed points is identified. The process is repeated by adding *m* more points and updating the local metamodels, until the number *mm(k)* is converged within a relative error of 1% from the previous iteration.
14. Finally the probability of failure is calculated as

$$p_f = \frac{\sum_{j=1}^{N} I(\mathbf{x}_j)}{N} = \frac{n_G + \sum_k mm(k) + n_o}{N}, \qquad (5)$$

by dividing the overall number of failures by the number *N* of the original MC samples. In Eq. (5), n_o represents failures among potential transition points which did not belong to any of the identified main clusters. Actual function evaluations are used for these points.

NUMERICAL EXAMPLES

In this section, the accuracy and efficiency of the proposed method are demonstrated with two examples; a vibration absorber and a two-bar bracket.

Example 1: A Vibration Absorber

A tuned damper system is shown in Figure E1. The system consists of an original system and a vibration absorber. For simplicity, the original system is assumed to have a single degree of freedom and is subject to a harmonic excitation $F(t) = \cos(\omega \cdot t)$. The absorber is attached to the original system in order to reduce its vibration amplitude.

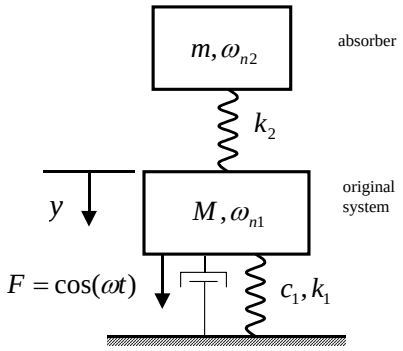

Figure 2. Tuned vibration absorber

The amplitude of the original system is a function of four parameters. In this example, we normalized it by the amplitude of its static response as follows

$$y = \frac{\left|1 - (\frac{1}{\beta_2})^2\right|}{\sqrt{\left[1 - R(\frac{1}{\beta_1})^2 - (\frac{1}{\beta_1})^2 - (\frac{1}{\beta_2})^2 + \frac{1}{\beta_1^2 \beta_2^2}\right]^2 + 4\varsigma^2\left[(\frac{1}{\beta_1}) - \frac{1}{\beta_1 \beta_2^2}\right]^2}}.$$

In Eq. (6), R is the mass ratio of the absorber to the original system, ς is the damping ratio of the original system, and β_1 and β_2 are the ratios of the natural frequency of the original system and vibration absorber with respect to the excitation frequency, respectively. It is assumed that the absorber does not provide additional damping to the overall system (see Fig. 2). For illustration purposes, R and ς are treated as deterministic variables with values R=0.01 and ς=0.01 respectively. Only β_1 and β_2 are random variables. They are both assumed normally distributed with mean 1.0 and standard deviation 0.025.

The objective of the absorber is to reduce the risk of the normalized amplitude $y(\beta_1, \beta_2)$ being larger than 28, taking into account the uncertainties in the parameters. Therefore, the limit state equation is $g(\beta_1, \beta_2) = 28 - y(\beta_1, \beta_2)$. Fig. 3 shows $y(\beta_1, \beta_2)$, indicating the existence of two disjoint failure domains.

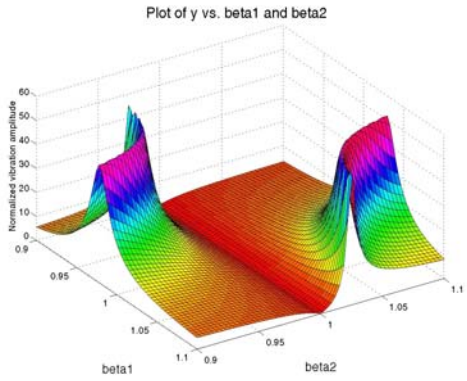

Figure 3. Plot of the normalized amplitude y vs. β_1 and β_2

One hundred thousand MC samples (N=100,000) are generated using the $\beta_1 \sim N(1,0.025)$ and $\beta_2 \sim N(1,0.025)$ distributions (step 1 of section 2.4). If we do not have an estimate of the probability of failure, we use a relatively "high" number of samples. In this problem we initially generated 100,000 samples. They are indicated by small dots in Fig. 4.

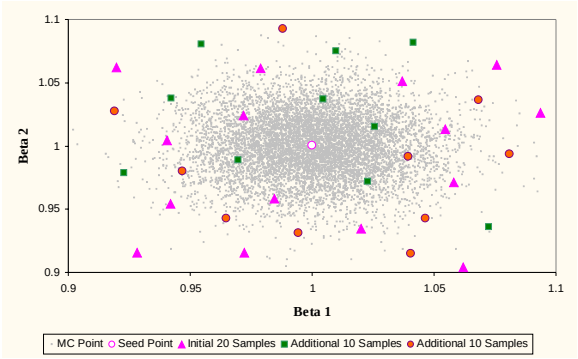

Figure 4. Samples for vibration absorber example

A global metamodel is first generated using the iterative process of steps 2 through 7. CVMLS metamodels are created progressively with m=5 (step 2) until convergence is achieved. The maximin method of section 2.1 selects the points from the 100,000 MC samples. The selected points for m=20, 30, and 40 are indicated in Fig. 4 with triangles, squares and filled circles, respectively. As points are added, we increasingly "space fill" the design space. The intent is to use a minimum number of points and thus, function evaluations without substantial loss of accuracy. Global metamodels are generated and subsequently, used to predict the limit state value for all 100,000 MC samples (step 4). The safe, failure and transition regions are identified according to steps 5 and 6. Table 1 summarizes the number of points in each region for different number of selected points m.

Table 1. Global metamodel convergence for vibration absorber

Number of Samples (m)	Number of Safe Points	Number of Failed Points (N_G)	Number of Transition Points
20	99,987	0	13
25	99,941	0	59
30	99,451	17	532
35	99,667	15	318
40	97,408	514	2,078
45	97,416	600	1,984
50	97,611	563	1,826

According to step 7, convergence of the global metamodel is achieved with m=50. The 563 failure points are within 10% of the 600 failure points of the previous iteration (m=45). Therefore, n_G is equal to 563 (see step 6) and NT is equal to 1826 (see step 5). Both these values are listed in the last row of Table 1. For the definition of the transition region, we have used $\eta = 0.20$ in this example (see step 5 of algorithm).

At this point, an *approximate* global metamodel has been obtained with "accuracy-on-demand." Although it can not accurately predict all successes and failures among the 100,000 MC samples, it is accurate enough to ensure that all safe 97,611 samples of Table 1 are indeed safe and the 563 samples are indeed failures. It simply does not have the necessary accuracy to determine if the NT=1826 transition samples are in the safe or failure domains. Also we do not know yet, if we have multiple disjoint failure regions and therefore, multiple transition regions. Now we can determine if an adequate number of samples were used in estimating the probability of failure. Based on the estimated probability of failure of $P_f = 563/100{,}000 = 0.00563$, the required number of samples according to Eq. (4), is less than 70,000. Because we initially used 100,000 samples, no additional MC points are required.

After a converged global metamodel is obtained and used to segregate the N=100,000 samples into safe, unsafe and transition points, a cluster analysis is performed using the NT = 1826 transition points, according to step 8 of the algorithm. It is important to also confirm that we have not selected an excessive number of points in the transition zone. According to step 5, the number of transition points NT must be less than 10% of N. In our case, NT = 1826 is only 1.8% of N = 100,000.

We initially use a large number of clusters in order to make sure we identify all existing clusters. In this example, we used 10 clusters. Fig. 5 shows the number of points in each cluster.

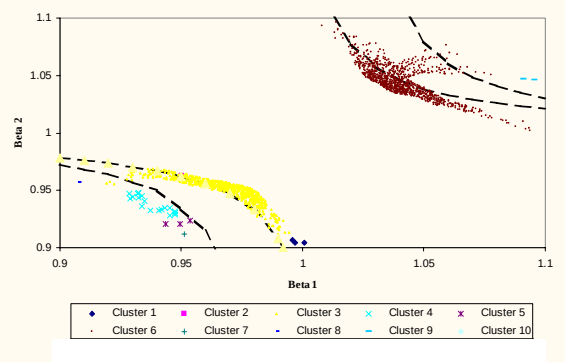

Figure 5. Clusters for vibration absorber

It is clear that there are only two clusters containing the majority of the transition points. This is an indication of the existence of two disjoint failure regions. Fig. 6 shows the points in the ten clusters indicating the existence of only two dominant clusters.

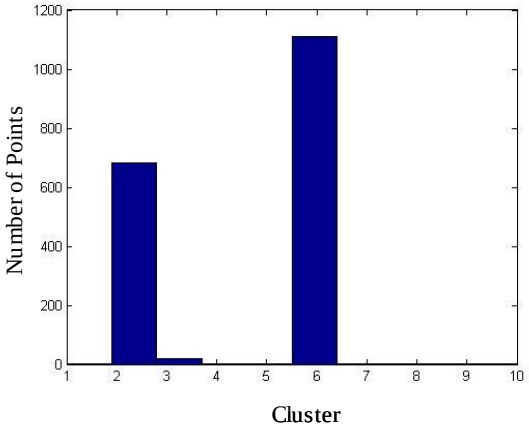

Figure 6. samples in each cluster for vibration absorber

Based on this observation, clustering is repeated using only two clusters. Out of the 1826 transition points, the first cluster includes 710 points and the second cluster includes the remaining 1116 points. A local metamodel will be created for each of the two clusters in order to identify the "correct" number of failures for each failure region. The local metamodels will be more accurate than the original approximate global metamodel.

Based on step 9, a PCA is performed using the 710 transition points of the first cluster in an effort to reduce the problem dimensionality. Fig. 7 shows the two principal coordinates and the transition points.

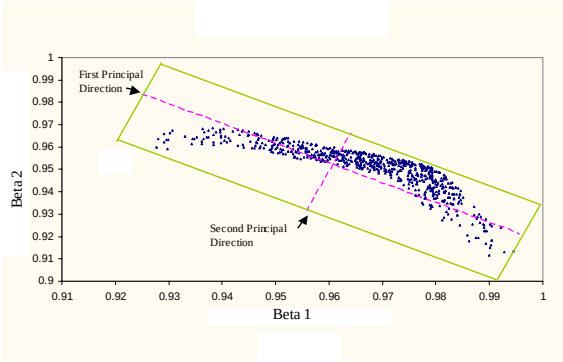

Figure 7. Transition region points and principal coordinates for vibration absorber

The minimum and maximum value of the 710 transition points along the two principal coordinates, have been also identified. They define the indicated rectangle. The two eigenvectors of the covariance matrix define the two principal coordinates. The corresponding eigenvalues are equal to 2.87×10^{-4} and 4.95×10^{-5}, respectively. The first eigenvalue is an order of magnitude larger than the second, indicating that there is sizeable scatter of the 710 points along the first principal coordinate (see Fig. 7). In this example, we will neglect the second coordinate.

Based on steps 9 and 10, the 710 transition points are projected onto the first principal coordinate and then perturbed randomly along the second. Fig. 8 shows the perturbed points (small dots). They are now more uniformly placed within the dashed rectangle. It also shows the original transition points (triangles) before they are perturbed along the second coordinate.

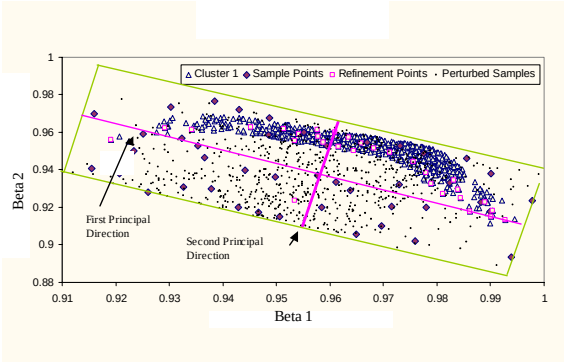

Figure 8. Perturbed transition region points for vibration absorber

A local metamodel is subsequently, built according to step 11. The maximin algorithm is used to successively select m=5 points (starting from 20 points) from the perturbed group of Fig. 6, and a local metamodel is built using the CVMLS algorithm. A converged local metamodel is achieved with n_L=45 points using the 5% stopping criterion of step 11. The selected 45 points are shown in Fig. 8 with diamonds. Using the converged local metamodel, a group of NT_L=286 points

with a small absolute value (close to limit state) has been identified (step 12). Convergence details of the local metamodel are shown in Table 2.

Table 2. Local metamodel convergence for vibration absorber (1st cluster)

Number of Samples (*m*)	Number of Safe Points	Number of Failed Points (*mm*)	Number of Points in Cluster
20	575	135	710
25	627	83	710
30	551	159	710
35	572	138	710
40	579	131	710
45	581	129	710

The converged local metamodel from step 11 is used to evaluate all NT=710 points in the first cluster and identify the failure points. Subsequently according to step 13, a few refinements of the local metamodel are obtained by adding m=5 points from the NT_L group. Each refined local metamodel is then used to evaluate all 710 points and identify the number of failures. The process is repeated until convergence is achieved. The results are summarized in Table 3. At convergence, m=30 and the number of failures is mm=147. There is a relative error of less than 1% from the 148 failures of the previous iteration (m=25).

Table 3. Refinement of local metamodel for vibration absorber (1st cluster)

Number of Samples (*m*)	Number of Safe Points	Number of Failed Points (*mm*)	Number of Points in Cluster
5	579	131	710
10	561	149	710
15	564	146	710
20	558	152	710
25	561	148	710
30	563	147	710

A local metamodel is also built for the second cluster and refined for improved accuracy. The results are shown in Tables 4 and 5.

Table 4. Local metamodel convergence for vibration absorber (2nd cluster)

Number of Samples (*m*)	Number of Safe Points	Number of Failed Points (*mm*)	Number of Points in Cluster
20	1,016	100	1,116
25	1,030	86	1,116
30	886	230	1,116
35	850	266	1,116
40	843	273	1,116

Table 5. Refinement of local metamodel for vibration absorber (2nd cluster)

Number of Samples (*m*)	Number of Safe Points	Number of Failed Points (*mm*)	Number of Points in Cluster
5	858	258	1,116
10	865	251	1,116
15	872	244	1,116
20	841	275	1,116
25	849	267	1,116
30	851	265	1,116

The probability of failure, according to step 14 is $p_f = \frac{n_G + \sum_{k=1}^{2} mm(k)}{N} = \frac{563+147+265}{100000} = 0.00975$ where n_G = 563 is the number of failures from the global metamodel, $mm(1)$=147 and $mm(2)$=265 is the number of failures from the local metamodels of the two clusters. Note that n_o in Eq. (5), is equal to zero in this example. The number of function evaluations is 50 (global metamodel) +45 (local metamodel for 1st cluster) + 70 (local metamodel for 2nd cluster) = 165. For comparison purposes, a traditional Monte Carlo simulation was also conducted using the same N=100,000 samples resulting in a probability of failure of $p_f = 0.00992$.

Example 2: A Two-Bar Bracket

The two-bar bracket problem (Fig. 9) of this section is adapted from [25]. A variation of the same problem has been also considered in a reliability-based design study in [26]. The objective is to support a force W without structural failure. The force is applied at an angle θ. The height and the base width of the bracket are h and s respectively. The two bars have a circular cross-section with outer and inner diameters of d_o and d_i, respectively.

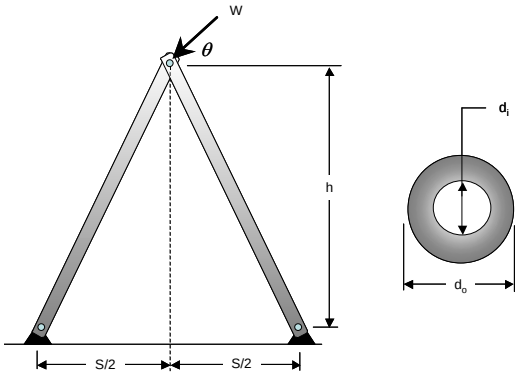

Figure 9. Two-bar bracket

Under the load *W*, bar one (left side) is always in compression. However, bar 2 (right side) can be in either tension or compression depending on the angle θ. Failure is considered if the tensile or compressive stress in either bar is greater than the material yield strength. Therefore, we have three failure criteria (limit state functions) as follows [43]

$$G_1 = S_1 - 2W\frac{\sqrt{h^2+(0.5s)^2}}{\pi\left(d_{o1}^2 - d_{i1}^2\right)}\left(\frac{\sin\theta}{h} + 2\frac{\cos\theta}{s}\right) \le 0 \qquad (7)$$

$$G_2 = S_2 - 2W\frac{\sqrt{h^2+(0.5s)^2}}{\pi\left(d_{o2}^2 - d_{i2}^2\right)}\left(\frac{\sin\theta}{h} - 2\frac{\cos\theta}{s}\right) \le 0 \qquad (8)$$

$$G_3 = S_2 + 2W\frac{\sqrt{h^2+(0.5s)^2}}{\pi\left(d_{o2}^2 - d_{i2}^2\right)}\left(\frac{\sin\theta}{h} - 2\frac{\cos\theta}{s}\right) \le 0 \qquad (9)$$

where S_1 and S_2 are the material yield strength for bar 1 and bar 2, respectively. The force *W* is assumed deterministic with *W*=31 kN. The remaining nine parameters are assumed normally distributed with $d_{oi} \sim N(30,1)$, $d_{ii} \sim N(20,1)$ for *i*=1,2, $h \sim N(100,5)$, $s \sim N(100,5)$, $\theta \sim N(59,1)$, $S_1 \sim N(200,25)$, $S_2 \sim N(100,25)$ where d_o, d_i, *h* and *s* are in mm, θ is in degrees and S_1, S_2 are in GPa. Note that we have assumed that bar 1 and bar 2 are made of different materials with different yield strength. This example represents a series system reliability assessment because there are multiple failure criteria. The system probability of failure is defined as

$$P_f = P\left(G_1 \le 0 \ \cup \ G_2 \le 0 \ \cup \ G_3 \le 0\right). \qquad (10)$$

The first step in estimating the probability of failure is to generate a large number of Monte Carlo samples. In this example, *N* = 200,000 samples are initially generated according to the statistical distributions of the nine random parameters.

Using the maximin sampling method, 30 points are selected from *N* and an approximate global metamodel is created for each limit state. The metamodels are then used to predict the limit state values for all 200,000 MC samples (step 4). It was found that 199,497 points are safe, 39 points are unsafe and 464 points are in the transition zone. The safe, failure and transition regions are identified according to steps 5 and 6. As in the previous example, the global metamodels are generated using the iterative process of steps 2 through 7. CVMLS metamodels are progressively created starting with *m*=30 (step 2) and adding 10 points until convergence is achieved. The maximin method selects the points from the 200,000 MC samples. Table 6 summarizes the number of points in each region for different number of selected points. According to Eq. (10), a point is in the failure or safe regions if $\left(G_1 \le 0 \ \cup \ G_2 \le 0 \ \cup \ G_3 \le 0\right)$ or $\left(G_1 \le 0 \ \cap \ G_2 \le 0 \ \cap \ G_3 \le 0\right)$, respectively. Also, a point is in the transition region if it belongs to the transition region of any of the three limit states.

Table 6. Convergence of global metamodels for two-bar bracket

Number of Samples (*m*)	Number of Safe Points	Number of Failed Points (N_G)	Number of Transition Points
30	199,497	36	464
40	199,557	33	410
50	199,564	39	397
60	199,548	42	410

As shown in the last row of Table 6, the global metamodels have converged with *m*=60. The 42 failure points with *m*=60 are within 10% of the 39 failure points of the previous iteration (*m*=50). Based on the converged global metamodels, we have 199,548 safe points, n_G =42 failure points and *NT*=410 points in the transition region. For the definition of the transition region, we have used $\eta = 0.10$ (see step 5 of algorithm).

At this point, an initial estimate of the probability of failure is obtained as P_f = 42/200,000 = 0.00021. Using this estimate in Eq. (4), a total of 1.4 million samples are needed. Therefore, 1.2 million additional samples are generated and evaluated using the global metamodels. The total number of safe and failed points is equal to 1,396,758, and n_G = 334 respectively, and the number of points in the transition region is *NT* = 2,908.

Using the *NT* = 2,908 points, the clustering method of step 8 identified two clusters of 1,351 and 1,505 points, respectively. Fig. 10 shows the number of points in each cluster. A maximum of five clusters was requested. In addition to the two clusters, there are 52 points which do not belong to either cluster. These points will be evaluated using the actual limit states.

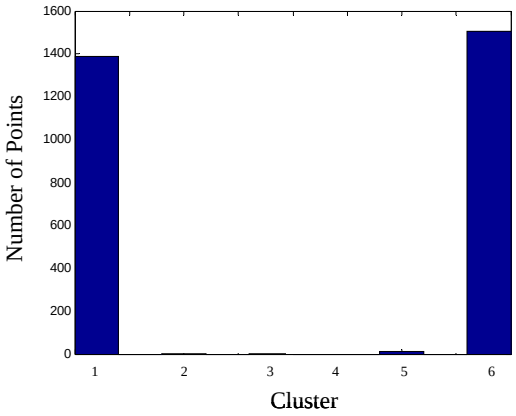

Figure 10. Clusters for two-bar bracket

A principal component analysis (PCA) is performed using the 1,351 points in the first cluster to determine if a dimension reduction can be achieved. The eigenvalues of the data covariance matrix (see section 2.3) are $\lambda = [0.076 \quad 0.930 \quad 0.957 \quad 1.040 \quad 1.911 \quad 11.51 \quad 26.5 \quad 47.2 \quad 62.6]$. Based on their relative magnitude, the first five coordinates are considered insignificant. Therefore, the 1,351 transition points are projected onto the four principal coordinates and then perturbed randomly along the five insignificant coordinates. Subsequently, the maximin sampling method selects points from the perturbed group of 1,351 points in order to build local metamodels (see steps 11 through 13).

Table 7 shows the convergence history of the local metamodels. Convergence has been achieved with m=50 samples indicating the existence of 9 failures out of the 1,351 points. The local metamodels are then used to identify NT_L =237 points (see step 12) which are close to the limit states and a few refinements are performed using maximin to select additional samples from the NT_L group. Table 8 shows the convergence of the local metamodels after the refinements with mm=14 failure points.

Table 7. Convergence of local metamodels for two-bar bracket (1st cluster)

Number of Samples (*m*)	Number of Safe Points	Number of Failed Points (*mm*)	Number of Points in Cluster
20	1,333	18	1,351
30	1,337	14	1,351
40	1,342	9	1,351
50	1,342	9	1,351

Table 8. Refinement of local metamodels for two-bar bracket (1st cluster)

Number of Samples (*m*)	Number of Safe Points	Number of Failed Points (*mm*)	Number of Points in Cluster
10	1,337	14	1,351
20	1,337	14	1,351

The procedure to build local metamodels is repeated for the transition points in the second cluster. In this case, the eigenvalues from PCA are $\lambda = [0.086 \quad 0.822 \quad 0.950 \quad 1.00 \quad 1.06 \quad 23.4 \quad 24.6 \quad 397 \quad 528]$ indicating again five insignificant coordinates. Tables 9 and 10 show the convergence history of the local metamodels, before and after refinement.

Table 9. Convergence of local metamodels for two-bar bracket (2nd cluster)

Number of Samples (*m*)	Number of Safe Points	Number of Failed Points (*mm*)	Number of Points in Cluster
20	1,503	2	1,505
30	1,408	97	1,505
40	1,434	71	1,505
50	1,440	65	1,505
60	1,442	63	1,505

Table 10. Refinement of local metamodels for two-bar bracket (2nd cluster)

Number of Samples (*m*)	Number of Safe Points	Number of Failed Points (*mm*)	Number of Points in Cluster
10	1,445	60	1,505
20	1,443	62	1,505
30	1,443	62	1,505

Finally, the probability of failure is $P_f = \frac{n_G + \sum_{k=1}^{2} mm(k) + n_o}{N} = \frac{334+14+62+3}{1{,}400{,}000} = \frac{413}{1{,}400{,}000} 0.000295$, where n_o=3 failures identified among the 52 transition points which did not belong to any of the two clusters. For comparison purposes, a traditional MC analysis using the same 1.4 million samples resulting in 398 failures and a probability of failure $P_f = \frac{398}{1{,}400{,}000} 0.00028428$. The proposed method predicted the probability of failure with a $\frac{0.000295 - 0.000284}{0.000284} \times 100 = 3.87\%$ error using only 272 function evaluations (60 for global metamodel, 50+20 for local

metamodel of first cluster, 60+30 for local metamodel of second cluster, 52 for points not belonging to clusters).

SUMMARY AND CONCLUSIONS

A Monte Carlo reliability assessment methodology has been presented for systems with multiple, disjoint failure regions and multiple most probable points. The method has all the advantages of Monte Carlo simulation but it is considerably more efficient. It uses a combination of approximate or "accurate-on-demand," global and local metamodels which serve as indicators to determine the failure and safe regions. A maximin "space-filling" sampling technique is used to construct the metamodels. A principal component analysis addresses the problem dimensionality making therefore, the method attractive for problems with a large number of random variables. A vibration absorber example with disjoint failure regions and a two-bar bracket example have been used to demonstrate that the proposed method has similar accuracy with the traditional MC simulation but it is substantially more efficient. In future research, the proposed method will be used in a gradient-free, simulation-based (not analytical) reliability-based design optimization algorithm which can handle "noisy" limit state problems with multiple failure regions.

REFERENCES

1. Hasofer, A. M. and Lind, N. C., "An Exact and Invariant First Order Reliability Format," *ASCE Journal of Engineering Mechanics*, 100, 111-121, 1974.

2. Rackwitz, R. and Fiessler, B., "Structural Reliability under Combined Random Load Sequences," *Computer and Structures*, 9(5), 484-494, 1978.

3. Breitung, K., "Asymptotic Approximations for Multinormal Integrals," *ASCE Journal of Engineering Mechanics*, 110, 357-366, 1984.

4. Tvedt, L., "Distribution of Quadratic Forms in Normal Space-Application to Structural Reliability," *ASCE Journal of Engineering Mechanics*, 116(6), 1183-1197, 1990.

5. Zhao, Y. and Ono, T., "A General Procedure for First/Second-Order Reliability Method (FORM/SORM)," *Structural Safety*, 21(2), 95-112, 1999.

6. Der Kiureghian, A. and T. Dakessian, "Multiple Design Points in First and Second-Order Reliability," *Structural Safety*, 20(1), 37-49, 1998.

7. Mahadevan, S. and P. Shi, "Multiple Linearization Method for Nonlinear Reliability Analysis," *ASCE Journal of Engineering Mechanics*, 127(11), 1165-1173, 2001.

8. Bucher, C. G., "Adaptive Sampling – An Iterative Fast Mote Carlo Procedure," *Structural Safety*, 5, 119-126, 1988.

9. Melchers, R. E., "Improved Importance Sampling Methods for Structural System Reliability Calculation," *Proceedings of International Conference on Structural Safety and Reliability (ICOSSAR)*, 1185-1192, 1989.

10. Karamchandani, A., Bjerager, P. and Cornell, C. A., "Adaptive Importance Sampling," *5th International Conference on Structural Safety and Reliability*, (Eds. Ang, A. H.-S., Shinozuka, M. and Schuëller, G. I.), ASCE, 855-862, New York, 1989.

11. Mahadevan, S. and Raghothamachar, P., "Adaptive Simulation for System Reliability Analysis of Large Structures," *Computers and Structures*, 77, 725-734, 2000.

12. Zou, T., Mahadevan, S., Mourelatos, Z. P. and Meernik, P. R., "Reliability Analysis of Automotive Body-Door Subsystem," *Reliability Engineering and System Safety*, 78(3), 315-324, 2002.

13. Mourelatos, Z. P., Kuczera, R. and Latcha, M., "An Efficient Monte Carlo Reliability Analysis using Global and Local Metamodels," *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA-2006-7087, Portsmouth, VA, September 6-8, 2006.

14. Zou, T., Mourelatos, Z. P., Mahadevan, S. and Tu, J., "An Indicator Response Surface-Based Monte Carlo Method for Efficient Component and System Reliability Analysis," *Proceedings of ASME Design Engineering Technical Conferences*, Paper# DETC2003/ DAC-48708, 2003.

15. Shan, S. and Wang, G. G., "Failure Surface Frontier for Reliability Assessment on Expensive Performance Function," *ASME Journal of Mechanical Design*, 128, 1227-1235, 2006.

16. Pitchumani, R. and Mawardi, A., "SAMS: Stochastic Analysis with Minimal Sampling – A fast Algorithm for Analysis and Design under Uncertainty," *ASME Journal of Mechanical Design*, 127, 558-571, 2005.

17. Tu, J. and Jones, D. R., "Variable Screening in Metamodel Design by Cross-Validated Moving Least Squares Method," *Proceedings of 44th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, AIAA-2003-1669, Norfolk, VA, April 7-10, 2003.

18. Ye, K. Q., Li, W. and A. Sudjianto, "Algorithmic Construction of Optimal Symmetric Latin Hypercube Designs," *Journal of Statistical Planning and Inference*, 90, 145-159, 2000.

19. Jolliffe, I. T., *Principal Component Analysis*, Springer-Verlag, New York, 1986.

20. Johnson, M. E., Moore, L. M. and Ylvisaker, D., "Minimax and Maximin Distance Design," *Journal of Statistical Planning and Inference*, 26, 131-148, 1990.

21. Currin, C., Mitchell, T., Morris, M. and Ylvisaker, D., "Bayesian Prediction of Deterministic Functions with Applications to the Design and Analysis of Computer Experiments," *Journal of Amer. Statist. Assoc.*, 86, 953-963, 1991.

22. Fang, K-T., Li, R. and Sudjianto, A., *Design and Modeling for Computer Experiments*, Chapman & Hall/CRC, Taylor & Francis Group, 2006.

23. Johnson, S. C., "Hierarchical Clustering Schemes," *Psychometrika*, 2, 241-254, 1967.

24. Haldar, A. and Mahadevan, S., *Probability, Reliability and Statistical Methods in Engineering Design*, John Wiley & Sons, Inc., 2000.

25. Arora, J. S., *Introduction to Optimum Design, 2nd Edition*, Elsevier Academic Press, 2004.

26. Du X., "Reliability-Based Design using Saddlepoint Approximation," *Proceedings of ASME Design Engineering Technical Conferences*, Paper# DETC2006-99077, 2006.