Proceedings of the ASME 2019
International Design Engineering Technical Conferences
and Computers and Information in Engineering Conference
IDETC/CIE2019
August 18-21, 2019, Anaheim, CA, USA

IDETC2019-97611

A NEW ROBUSTNESS METRIC FOR ROBUST DESIGN OPTIMIZATION UNDER TIME- AND SPACE-DEPENDENT UNCERTAINTY THROUGH METAMODELING

Xinpeng Wei
Department of Mechanical and Aerospace Engineering
Missouri University of Science & Technology
Rolla, MO, USA

Xiaoping Du[1]
Department of Mechanical and Energy Engineering
Indiana University – Purdue University Indianapolis
Indianapolis, IN, USA

ABSTRACT

Product performance varies with respect to time and space in many engineering applications. This work discusses how to measure and evaluate the robustness of a product or component when its quality characteristics are functions of random variables, random fields, temporal variables, and spatial variables. At first, the existing time-dependent robustness metric is extended to the present time- and space-dependent problem. The robustness metric is derived using the extreme value of the quality characteristics with respect to temporal and spatial variables for the nominal-the-better type quality characteristics. Then a metamodel-based numerical procedure is developed to evaluate the new robustness metric. The procedure employs a Gaussian Process regression method to estimate the expected quality loss that involves the extreme quality characteristics. The expected quality loss is obtained directly during the regression model building process. Three examples are used to demonstrate the robustness analysis method. The proposed method can be used for robustness assessment during robust design optimization under time- and space-dependent uncertainty.

1. INTRODUCTION

Robust design [1] is an optimization design methodology for improving the quality of a product through minimizing the effect of the causes of variation without eliminating the causes [2]. It allows for the use of low grade materials, reduces labor and material cost while improving reliability and reducing operating cost [2].

Robustness analysis, which evaluates and predicts the robustness of a design, is repeated for a number of times during robust design optimization. Many metrics that measure the robustness exist in literature. The most common metric is the Taguchi's quality loss function (QLF) [2]. This metric measures not only the distance between the average quality characteristics (QCs) and their targets, but also the variation in the QCs [3]. There are also other robustness metrics, such as the signal-to-noise ratio [2], the percentile difference [4], and the worst-case QCs [5].

Most of the above robustness metrics are defined for static QCs that do not change over time and space. Some of the metrics could be used for dynamics problems, but they are only applicable for situations where the targets of QCs vary with signals [6, 7], instead of with time. To deal with problems involving time-dependent QCs, Goethals et al. [8] proposed to use the weighted sum of mean values of a QLF at discretized time instances to measure the robustness. The weighted-sum method, however, does not take into consideration of the autocorrelation of the time-dependent QLF, which is modeled as a stochastic process. To overcome this drawback, Du [3] proposed to use the maximal value of the time-dependent QLF to measure the time-dependent robustness.

In addition to the above static and time-dependent problems, more general is the time- and space-dependent (TSD) problem [9]. In many engineering applications, QCs vary with both time and space. There are at least two reasons for the TSD QCs. (1) A QC is a function of TSD variables, such as wind load and road conditions. (2) The QC itself is a function of temporal and spatial variables. A typical example is a wind turbine. Since the wind speed varies with time and location, it is usually modeled as a TSD random field, subjected to which, the QC of the turbine is hence TSD.

There is a research need for measuring robustness for optimization involving TSD problems. The object of this work is to develop a robustness metric for those problems and propose corresponding numerical procedure for evaluating the metric. We use the expectation of the maximum value of a TSD QLF as the robustness metric. The Gaussian process model [10-16] is employed to efficiently obtain the maximal value of the TSD

[1] 799 W. Michigan Street, Indianapolis, IN 46202-5195, USA, Email: duxi@iu.edu

QLF. The contributions of this work are threefold. First, a TSD robustness metric is defined. It can take into consideration of all information of the TSD QLF, including its autocorrelation. Therefore, it is mathematically a rigorous metric for the TSD problems. Second, a Gaussian process based method is developed to effectively compute the TSD metric. Third, two stopping criteria are proposed to effectively train the Gaussian process model, leading to accurate and efficient results.

The paper is organized as follows. Section 2 briefly reviews the time-dependent robustness metric, whose extension to TSD problems is discussed with a new robustness metric in Section 3, followed by a meta-modeling numerical procedure for the new metric in Section 4. Three examples are given in Section 5, and conclusions are provided in Section 6.

2. REVIEW OF STATIC AND TIME-DEPENDENT ROBUSTNESS METRICS

Nominal-the-best, smaller-the-better, and larger-the-better are three types of QCs [3]. In this work, we only focus on the nominal-the-best type. The discussions, however, can be extended to the other two types.

2.1 Static robustness metric

The most common robustness metric is the QLF. Let a QC be defined as

$$\mathrm{Y} = g(\mathbf{X}) \tag{1}$$

where $\mathbf{X} = (X_1, X_2 \dots, X_N)$ are N input random variables whose supports are $\mathbf{\Theta} = (\Theta_1, \Theta_2 \dots, \Theta_N)$. Then the QLF is

$$L = A(Y - m)^2 \tag{2}$$

where m is the target value of Y, and A is a constant determined by a monetary loss. The robustness is measured by the expectation, or the mean E_L of L, which is calculated by

$$E_L = A[(\mu_Y - m)^2 + \sigma_Y^2] \tag{3}$$

where μ_Y and σ_Y are the mean and standard deviation of Y, respectively. The smaller is E_L, the better is the robustness because μ_Y (the average QC) is closer to the target m and σ_Y (variation of the QC) is smaller.

2.2 Time-dependent robustness metric

A time-dependent QC is given by

$$Y = g(\mathbf{X}, t) \tag{4}$$

Note that the input of $g(\cdot)$ may also include random processes, which can be transformed into functions with respect to random variables and t [17]. Thus Eq. (4) does not lose generality. At instant t, the QLF is given as

$$L(t) = A(t)[Y - m(t)]^2 = A(t)[g(\mathbf{X}, t) - m(t)]^2 \tag{5}$$

$L(t)$ can measure only the quality loss at a specific time instant t and is thus called point quality loss function (P-QLF). To measure the quality loss of a product over a time interval $[\underline{t}, \overline{t}]$, Du [3] proposed to use the extreme value or the worst-case value of $L(t)$ over $[\underline{t}, \overline{t}]$. The worst-case quality loss is called interval quality loss function (I-QLF) and is given by

$$L(\underline{t}, \overline{t}) = \max_{t \in [\underline{t}, \overline{t}]} \{A(t)[g(\mathbf{X}, t) - m(t)]^2\} \tag{6}$$

Note that $L(\underline{t}, \overline{t})$ is a random variable while $L(t)$ is a random process. Like static problems, the expectation $E_L(\underline{t}, \overline{t})$ of $L(\underline{t}, \overline{t})$ is also used as the time-dependent robustness metric given by

$$E_L(\underline{t}, \overline{t}) = \mathop{\mathrm{E}}_{\mathbf{\Theta}}[L(\underline{t}, \overline{t})] \tag{7}$$

where $\mathop{\mathrm{E}}_{\mathbf{\Theta}}(\cdot)$ represents expectation over $\mathbf{\Theta}$. Minimizing $E_L(\underline{t}, \overline{t})$ reduces both the deviation of the QC from its target and the variation in the QC over time interval $[\underline{t}, \overline{t}]$.

3. A NEW ROBUSTNESS METRIC FOR TIME- AND SPACE-DEPENDENT QCS

In TSD problems, in addition to random variables and random processes, static random fields and time-dependent random fields are also involved. For convenience, we do not distinguish random processes, static random fields or time-dependent random fields. In this paper, we generally call them random fields. Let $\mathbf{Z} = (\mathrm{S}_1, \mathrm{S}_2, \mathrm{S}_3, t)$ be the vector comprising the three spatial parameters (*x*-, *y*-, and *z*-coordinates) and the time. Note that for problems in one-dimensional and two-dimensional space, $\mathbf{Z} = (\mathrm{S}_1, t)$ and $\mathbf{Z} = (\mathrm{S}_1, \mathrm{S}_2, t)$, respectively. Note that random fields can be transformed into functions with respect to random variables and $\mathbf{Z}$ [17]. With loss of generality, a TSD QC is then given by

$$Y = g(\mathbf{X}, \mathbf{Z}) \tag{8}$$

With the TSD QC, the corresponding QLF is given as

$$L(\mathbf{X}, \mathbf{Z}) = A(\mathbf{Z})[Y - m(\mathbf{Z})]^2 = A(\mathbf{Z})[g(\mathbf{X}, \mathbf{Z}) - m(\mathbf{Z})]^2 \tag{9}$$

$L(\mathbf{X}, \mathbf{Z})$ measures the qualify loss at any specific point $\mathbf{z} \in \Omega$, where Ω is the domain of $\mathbf{Z}$, and we call it the point qualify loss function (P-QLF).

Before defining the TSD robustness metric, we propose some criteria of robustness metrics for the TSD problems, inspired by the criteria of the robustness metrics for time-dependent problems given in [3]. The criteria are as follows:

(a) A metric must represent the maximal quality loss over Ω. The maximal quality loss means that a product is in its worst situation where the product may suffer a significant failure. Therefore, engineers must take the maximal quality loss into consideration at the design stage of the product.

(b) The metric should increase (or at least stay the same) with Ω, given that other conditions stay unchanged. The reason is that when a product involves a larger space and/or is put into service for a longer time interval, the robustness should be worse (or at least the same).

(c) The metric should capture the autocorrelation of the P-QLF $L(\mathbf{X},\mathbf{Z})$ over Ω. Since $L(\mathbf{X},\mathbf{Z})$ is a random field, its autocorrelation is an important property. Two different random fields with the same marginal distribution at any point may have very different performances if they do not share the same autocorrelation.

(d) Minimizing the metric will lead to optimizing the mean QCs and minimizing the variations of the QCs over Ω. This criterion comes from the purpose of the robust optimization [18].

Based on the above criteria, we define the TSD robustness metric $E_L(\Omega)$ as

$$E_L(\Omega) = \mathop{\mathrm{E}}_{\boldsymbol{\Theta}}[L_{\max}(\mathbf{X},\Omega)] \tag{10}$$

where

$$L_{\max}(\mathbf{X},\Omega) = \max_{\mathbf{z}\in\Omega} L(\mathbf{X},\mathbf{z}) \tag{11}$$

is the maximum value of $L(\mathbf{X},\mathbf{Z})$ and is called domain quality loss function (D-QLF). The definition of $L_{\max}(\mathbf{X},\Omega)$ let $E_L(\Omega)$ meet Criterion (a) naturally. Let $\acute{\Omega} \subset \Omega$, then it is obvious that

$$L_{\max}(\mathbf{x},\acute{\Omega}) \le L_{\max}(\mathbf{x},\Omega), \forall \mathbf{x} \in \boldsymbol{\Theta} \tag{12}$$

and hence that $E_L(\acute{\Omega}) \le E_L(\Omega)$. Therefore, $E_L(\Omega)$ meets Criterion (b). Since $L_{\max}(\mathbf{X},\Omega)$ is the maximum value distribution [19, 20] of $L(\mathbf{X},\mathbf{Z})$, the autocorrelation of $L(\mathbf{X},\mathbf{Z})$ is necessary for computing $L_{\max}(\mathbf{X},\Omega)$. Different autocorrelation functions of $L(\mathbf{X},\mathbf{Z})$ will lead to different distributions of $L_{\max}(\mathbf{X},\Omega)$, and hence $E_L(\Omega)$ can capture the autocorrelation of $L(\mathbf{X},\mathbf{Z})$, indicating that $E_L(\Omega)$ meets Criterion (c). Since $L(\mathbf{X},\mathbf{Z})$, and hence $L_{\max}(\mathbf{X},\Omega)$ and $E_L(\Omega)$, are nonnegative, minimizing $E_L(\Omega)$ requires that the QC $g(\mathbf{X},\mathbf{Z})$ gets close to its target $m(\mathbf{Z})$ as much as possible. Therefore, $E_L(\Omega)$ also meets Criterion (d).

4. A META-MODLING APPROACH TO ROBUSTNESS ASSESSMENT

4.1 Overview of the proposed robustness analysis

The main idea of the proposed robustness analysis method is to train a Gaussian process model $\hat{L}(\mathbf{X},\mathbf{Z})$ for $L(\mathbf{X},\mathbf{Z})$. Replacing $L(\mathbf{X},\mathbf{Z})$ in Eq. (11) with $\hat{L}(\mathbf{X},\mathbf{Z})$, we can approximate $L_{\max}(\mathbf{X},\Omega)$ with $\hat{L}_{\max}(\mathbf{X},\Omega)$

$$\hat{L}_{\max}(\mathbf{X},\Omega) = \max_{\mathbf{z}\in\Omega} \hat{L}(\mathbf{X},\mathbf{z}) \tag{13}$$

Then MCS is used to compute $E_L(\Omega)$ through

$$E_L(\Omega) = \frac{1}{n_{\mathrm{MCS}}} \sum_{i=1}^{n_{\mathrm{MCS}}} \hat{L}_{\max}\left(\mathbf{x}^{(i)},\Omega\right) \tag{14}$$

where n_{MCS} is the sample size, and $\mathbf{x}^{(i)}$ represents the i-th sample of $\mathbf{X}$. Since $\hat{L}(\mathbf{X},\mathbf{Z})$ is computationally much cheaper than $L(\mathbf{X},\mathbf{Z})$, the proposed method can significantly improve the efficiency. Generally, the more samples of $L(\mathbf{X},\mathbf{Z})$ are used, the more accurate $\hat{L}(\mathbf{X},\mathbf{Z})$ we obtain. The evaluation of $L(\mathbf{X},\mathbf{Z})$, however, is usually expansive, because in engineering problems, $L(\mathbf{X},\mathbf{Z})$ is usually a black-box function whose evaluation needs the expensive numerical procedures or simulations [21].

To balance the accuracy and the efficiency, we do not require $\hat{L}(\mathbf{X},\mathbf{Z})$ to be accurate globally. Instead, we only need it to be locally accurate so that $\hat{L}(\mathbf{X},\mathbf{Z})$ is accurate only at samples of $\mathbf{X}$ in Eq. (14). To this end, we employ the efficient global optimization (EGO) [12, 22] to adaptively add samples to update $\hat{L}(\mathbf{X},\mathbf{Z})$.

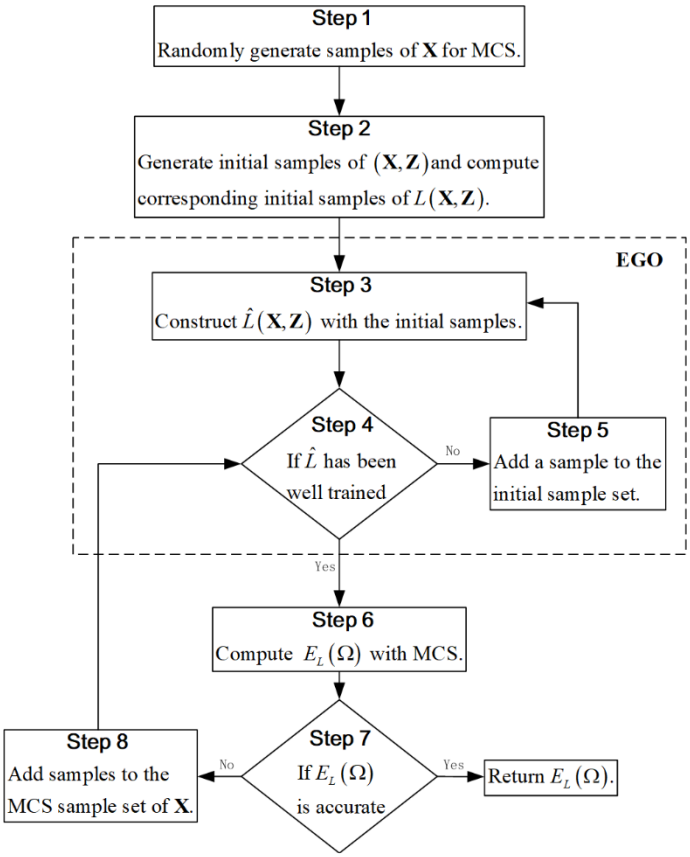

FIGURE 1: SIMPLIFIED FLOWCHART

To have a quick overview to the proposed method, we give a simplified version of the flowchart in Fig. 1. There are in total eight steps in the proposed method. Details of Step 2 will be given in Subsection 4.2. The EGO, which comprises Steps 3-5, will be detailed in Subsection 4.3. We propose two stopping criteria in Step 4 and Step 7. Detailed information is given in

Subsection 4.4. The implementation of the whole algorithm and a detailed version of the flowchart will be given in Subsection 4.5. In Subsection 4.6, we discuss how to deal with a more general problem that involves random fields.

4.2 Initial samples

The principle of generating initial samples for building a Gaussian process model is to spread the initial samples evenly. Commonly used sampling methods include random sampling, Latin hypercube sampling and Hammersley sampling [23]. In this study, we employ the Hammersley sampling method because it performs better in providing uniformity properties over a multidimensional space [24].

Since the dimension of the entire input vector $(\mathbf{X}, \mathbf{Z})$ is $N + N_{\mathrm{Z}}$, where N_{Z} is the dimension of $\mathbf{Z}$, the Hammersley sampling method generates initial samples in a hypercube $[0,1]^{N+N_{\mathrm{Z}}}$. To get initial samples of $\mathbf{X}$, we can simply use the inverse probability method to transform the samples from the hypercube space to $\mathbf{X}$ space. As for the initial samples of $\mathbf{Z}$, we treat $\mathbf{Z}$ as if it was a uniformly distributed random vector and then also use the inverse probability method to transform the samples from the hypercube space to the $\mathbf{Z}$ space.

In this paper, when variables are assembled into a vector, we use a row vector, while samples of the vector are arranged lengthwise to form a matrix. For example, the initial samples $\mathbf{x}^{\mathrm{in}}$ of $\mathbf{X} = (X_1, X_2 \dots, X_N)$ are

$$\mathbf{x}^{\mathrm{in}} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \ddots & x_N^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \ddots & x_N^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(n_{\mathrm{in}})} & x_2^{(n_{\mathrm{in}})} & \ddots & x_N^{(n_{\mathrm{in}})} \end{bmatrix} \tag{15}$$

where n_{in} represents the total number of initial samples. With $\mathbf{x}^{\mathrm{in}}$ and the initial samples $\mathbf{z}^{\mathrm{in}}$ of $\mathbf{Z}$ .We then obtain corresponding initial samples $\mathbf{l}^{\mathrm{in}}$ of $L(\mathbf{X}, \mathbf{Z})$ through Eq. (9).

4.3 Employment of EGO

EGO is based on the Gaussian process model. with $\mathbf{x}^{\mathrm{in}}$, $\mathbf{z}^{\mathrm{in}}$ and $\mathbf{l}^{\mathrm{in}}$ we can build $\hat{L}(\mathbf{X}, \mathbf{Z})$. Because only a limit number of samples, i.e. $(\mathbf{x}^{\mathrm{in}}, \mathbf{z}^{\mathrm{in}}, \mathbf{l}^{\mathrm{in}})$, are used, $\hat{L}(\mathbf{X}, \mathbf{Z})$ has model uncertainty (or epistemic uncertainty), which is measured by $\sigma(\mathbf{X}, \mathbf{Z})$.

Practically, when $\hat{L}(\mathbf{X}, \mathbf{Z})$ is available, we need to discretize Ω to compute the maximal value $\hat{L}_{\max}(\mathbf{X}, \Omega)$ through Eq. (13). If we discretize Z_j, the j-th element of $\mathbf{Z}$, into m_j points, then Ω will be discretized into $n_\Omega = \prod_{i=1}^{N_{\mathrm{Z}}} m_j$ points. For convenience, we denote the domain the n_Ω points of $\mathbf{Z}$ as $\mathbf{z}^\Omega$, whose dimension is $n_\Omega \times N_{\mathrm{Z}}$. Then Eq. (13) is rewritten as

$$\hat{L}_{\max}(\mathbf{X}, \Omega) = \max_{\mathbf{z} \in \mathbf{z}^\Omega} \hat{L}(\mathbf{X}, \mathbf{z}) \tag{16}$$

Since $\hat{L}_{\max}(\mathbf{X}, \Omega)$ may not be the exact global maximum, we need to add samples of $(\mathbf{X}, \mathbf{Z}, L)$ to update $\hat{L}(\mathbf{X}, \mathbf{Z})$ so that the $\hat{L}_{\max}(\mathbf{X}, \Omega)$ will be accurate. To determine how to add the training sample, we use the well-known expected improvement (EI) learning function [22], which is given by

$$\mathrm{EI}(\mathbf{x}, \mathbf{z}) = \left(\hat{L} - \hat{L}_{\max}\right)\Phi\left(\frac{\hat{L} - \hat{L}_{\max}}{\sigma(\mathbf{x},\mathbf{z})}\right) + \sigma(\mathbf{x}, \mathbf{z})\varphi\left(\frac{\hat{L} - \hat{L}_{\max}}{\sigma(\mathbf{x},\mathbf{z})}\right) \tag{17}$$

where $\hat{L} = \hat{L}(\mathbf{x}, \mathbf{z})$ and $\hat{L}_{\max} = \hat{L}_{\max}(\mathbf{x}, \Omega)$; $\Phi(\cdot)$ and $\varphi(\cdot)$ are the cumulative distribution function and probability density function of a standard Gaussian variable, respectively. $\mathrm{EI}(\mathbf{x}, \mathbf{z})$ means that the exact $L_{\max}(\mathbf{x}, \Omega)$ is expected to be $\mathrm{EI}(\mathbf{x}, \mathbf{z})$ larger than the current $\hat{L}_{\max}(\mathbf{x}, \Omega)$. In other words, if we add a sample at $(\mathbf{x}, \mathbf{z})$ to update $\hat{L}(\mathbf{X}, \mathbf{Z})$, we expect to update current $\hat{L}_{\max}(\mathbf{x}, \Omega)$ to $\hat{L}_{\max}(\mathbf{x}, \Omega) + \mathrm{EI}(\mathbf{x}, \mathbf{z})$. In principle, we should update $\hat{L}_{\max}(\mathbf{x}, \Omega)$ by a step size as large as possible so that the algorithm converges quickly. Therefore, we determine the next training point $\left(\mathbf{x}^{(\mathrm{next})}, \mathbf{z}^{(\mathrm{next})}\right)$ through

$$\left(\mathbf{x}^{(\mathrm{next})}, \mathbf{z}^{(\mathrm{next})}\right) = \underset{\mathbf{x} \in \mathbf{x}^{\mathrm{MCS}},\ \mathbf{z} \in \mathbf{z}^\Omega}{\arg\max}\ \mathrm{EI}(\mathbf{x}, \mathbf{z}) \tag{18}$$

where $\mathbf{x}^{\mathrm{MCS}}$ represents the MCS population of $\mathbf{X}$. With Eq. (9), we can obtain the next sample $\mathrm{l}^{(\mathrm{next})}$ of $L(\mathbf{X}, \mathbf{Z})$. Then the initial sample set $(\mathbf{x}^{\mathrm{in}}, \mathbf{z}^{\mathrm{in}}, \mathbf{l}^{\mathrm{in}})$ is updated through

$$\begin{cases} \mathbf{x}^{\mathrm{in}} = \begin{bmatrix} \mathbf{x}^{\mathrm{in}} \\ \mathbf{x}^{(\mathrm{next})} \end{bmatrix} \\ \mathbf{z}^{\mathrm{in}} = \begin{bmatrix} \mathbf{z}^{\mathrm{in}} \\ \mathbf{z}^{(\mathrm{next})} \end{bmatrix} \\ \mathbf{l}^{\mathrm{in}} = \begin{bmatrix} \mathbf{l}^{\mathrm{in}} \\ \mathrm{l}^{(\mathrm{next})} \end{bmatrix} \end{cases} \tag{19}$$

The updated initial sample set $(\mathbf{x}^{\mathrm{in}}, \mathbf{z}^{\mathrm{in}}, \mathbf{l}^{\mathrm{in}})$ is used to update $\hat{L}(\mathbf{X}, \mathbf{Z})$. Then $\hat{L}_{\max}(\mathbf{X}, \Omega)$ in Eq. (16), and hence $E_L(\Omega)$ in Eq. (14), will also be updated. With similar procedures, training samples are iteratively added and $E_L(\Omega)$ is updated iteratively until a stopping criterion is satisfied.

4.4 Stopping criteria

In this subsection, we discuss two stopping criteria in Steps 4 and 7.

The task of the stopping criterion in Step 4 is to judge whether more training samples are necessary to update $\hat{L}(\mathbf{X}, \mathbf{Z})$. A straightforward stopping criterion is

$$\max_{\mathbf{x} \in \mathbf{x}^{\mathrm{MCS}},\ \mathbf{z} \in \mathbf{z}^\Omega} \left|\mathrm{EI}(\mathbf{x}, \mathbf{z}) / \hat{L}_{\max}(\mathbf{x}, \Omega)\right| < c \tag{20}$$

where c is a threshold, which usually takes a small positive number, such as 0.005. This stopping criterion guarantees that for any $\mathbf{x} \in \mathbf{x}^{\mathrm{MCS}}$, the absolute value of the expected improvement rate of $\hat{L}_{\max}(\mathbf{x}, \Omega)$ is small enough. In other words, this stopping criterion guarantees that the n_{MCS} samples

of $\hat{L}_{\max}(\mathbf{X},\Omega)$ are all accurate enough so that $E_L(\Omega)$ is accurate enough. If c is too small, however, it may result in unnecessary iterations and hence an unnecessary computational cost. Besides, the c does not directly measure the accuracy of $E_L(\Omega)$. As a result, it is hard to determine a proper value for c. To resolve this problem, we propose a new stopping criterion

$$W = \left| \left\{ \underset{\mathbf{x}\in\mathbf{x}^{\mathrm{MCS}}}{\mathrm{mean}} \left[\max_{\mathbf{z}\in\mathbf{z}^{\Omega}} \mathrm{EI}(\mathbf{x},\mathbf{z}) \right] \right\} / E_L(\Omega) \right| \le d \tag{21}$$

where d is another threshold, which usually takes a small positive number, such as 0.005. Since $\max_{\mathbf{z}\in\mathbf{z}^{\Omega}} \mathrm{EI}(\mathbf{x},\mathbf{z})$ is the maximal expected improvement of $\hat{L}_{\max}(\mathbf{x},\Omega)$, $\underset{\mathbf{x}\in\mathbf{x}^{\mathrm{MCS}}}{\mathrm{mean}} \left[\max_{\mathbf{z}\in\mathbf{z}^{\Omega}} \mathrm{EI}(\mathbf{x},\mathbf{z}) \right]$ is the expected maximal improvement of $E_L(\Omega)$. Then, W is the absolute value of the expected improvement rate of $E_L(\Omega)$. W directly measures the accuracy of $E_L(\Omega)$, and so we can set the value of d according to specific engineering requirement. For example, if we set $d = 1\%$, the program will stop adding training samples once it realizes that adding more training samples will not update the current $E_L(\Omega)$ by more than 1%.

Step 7 mainly deals with the following question: How many samples of $\hat{L}_{\max}(\mathbf{X},\Omega)$ are enough to obtain accurate $E_L(\Omega)$? Since $L_{\max}(\mathbf{X},\Omega)$ is a random variable, the sample size, which is needed to estimate its mean value $E_L(\Omega)$, is dependent on the standard deviation $\sigma(\Omega)$ of $L_{\max}(\mathbf{X},\Omega)$. If the sample size is n_{MCS}, the deviation coefficient γ of $E_L(\Omega)$ is

$$\gamma = \frac{\sigma(\Omega)}{E_L(\Omega)\sqrt{n_{\mathrm{MCS}}}} \tag{22}$$

where $E_L(\Omega)$ is estimated by Eq. (14), and $\sigma(\Omega)$ is estimated by

$$\sigma(\Omega) = \sqrt{\frac{1}{n_{\mathrm{MCS}}-1}\sum_{i=1}^{n_{\mathrm{MCS}}}\left[\hat{L}_{\max}(\mathbf{x}^{(i)},\Omega) - E_L(\Omega)\right]^2} \tag{23}$$

Eq. (22) shows that the larger is n_{MCS}, the smaller γ will we obtain. A smaller γ means that the estimated $E_L(\Omega)$ is more accurate. Therefore, we use the following stopping criterion in Step 7:

$$\gamma \le k \tag{24}$$

where k is a threshold, which usually takes a small positive number, such as 0.005.

If the stopping criterion in Eq. (24) is not satisfied, how many samples do we need to add to the current sample set $\mathbf{x}^{\mathrm{MCS}}$? Combining Eq. (22) and Eq. (24), we have

$$n_{\mathrm{MCS}} \ge \left[\frac{\sigma(\Omega)}{E_L(\Omega)k}\right]^2 \tag{25}$$

It means that to meet the stopping criterion in Eq. (24), the sample size should be at least $\left[\frac{\sigma(\Omega)}{E_L(\Omega)k}\right]^2$. For convenience, let $n_0 = \mathrm{round}\left\{\left[\frac{\sigma(\Omega)}{E_L(\Omega)k}\right]^2\right\}$ where $\mathrm{round}(\cdot)$ represents the operation to get the nearest integer. Then the number n_{add} of samples we should add to the current sample set $\mathbf{x}^{\mathrm{MCS}}$ is

$$n_{\mathrm{add}} = n_0 - n_{\mathrm{MCS}} \tag{26}$$

However, when $\hat{L}(\mathbf{X},\mathbf{Z})$ is too rough at the first several training iterations, both $E_L(\Omega)$ and $\sigma_{L(\mathbf{X},\Omega)}$ have very poor accuracy. As a result, n_{add} determined by Eq. (26) may be misleading. To resolve this problem, we set a threshold $\tilde{n}_{\mathrm{add}}$ for n_{add}. Then n_{add} is modified to

$$n_{\mathrm{add}} = \begin{cases} \tilde{n}_{\mathrm{add}}, \text{if } n_0 - n_{\mathrm{MCS}} > \tilde{n}_{\mathrm{add}} \\ n_0 - n_{\mathrm{MCS}}, \text{otherwise} \end{cases} \tag{27}$$

4.5 Implementation of the proposed method

In this subsection, we give the detailed procedure of the proposed method. The detailed version of the flowchart is shown in Fig. 2. We have labeled the number of function evaluations in Eq. (9). The total number n_{call} of function evaluations in Eq. (9) is used to measure the main computational cost of the proposed method, because Eq. (9) involves the computation of an expensive black-box function.

4.6 Extension to problems with input random fields

When the TSD QC $g(\cdot)$ involves input random fields, it is straightforward to use the series expansion of the random fields so that the above implementation of the proposed method still holds. For example, a QC is given as

$$Y = g(\mathbf{X},\mathbf{H}(\mathbf{Z}),\mathbf{Z}) \tag{28}$$

where $\mathbf{H}(\mathbf{Z})$ is a vector of random fields. To easily present the idea, we assume there is only one random filed, given by $H(\mathbf{Z})$. The series expansion of a random field $H(\mathbf{Z})$ is denoted as $H(\boldsymbol{\xi},\mathbf{Z})$ where $\boldsymbol{\xi}$ is a vector of random variables. Then Eq. (28) is rewritten as

$$Y = g[\mathbf{X}, H(\boldsymbol{\xi},\mathbf{Z}), \mathbf{Z}] \tag{29}$$

or equivalently as

$$Y = g[(\boldsymbol{\xi},\mathbf{X}), \mathbf{Z}] \tag{30}$$

Treating $(\boldsymbol{\xi},\mathbf{X})$ as the total input random variables, then Eq. (30) shares the same format with Eq. (8), and so the above implementation in Subsection 4.5 also holds.

In this way, the proposed method, however, may suffer from the curse of dimensionality. Since many random variables, i.e. $\boldsymbol{\xi}$, are in the series expansion $H(\boldsymbol{\xi},\mathbf{Z})$, the dimension of $\boldsymbol{\xi}$, and hence of $g[(\boldsymbol{\xi},\mathbf{X}),\mathbf{Z}]$, is high. As a result, the dimension of the surrogate model $\hat{L}[(\boldsymbol{\xi},\mathbf{X}),\mathbf{Z}]$ is high. The high-dimensional surrogate model has as least two drawbacks. First, it is not cheap anymore, losing its expected advantages.

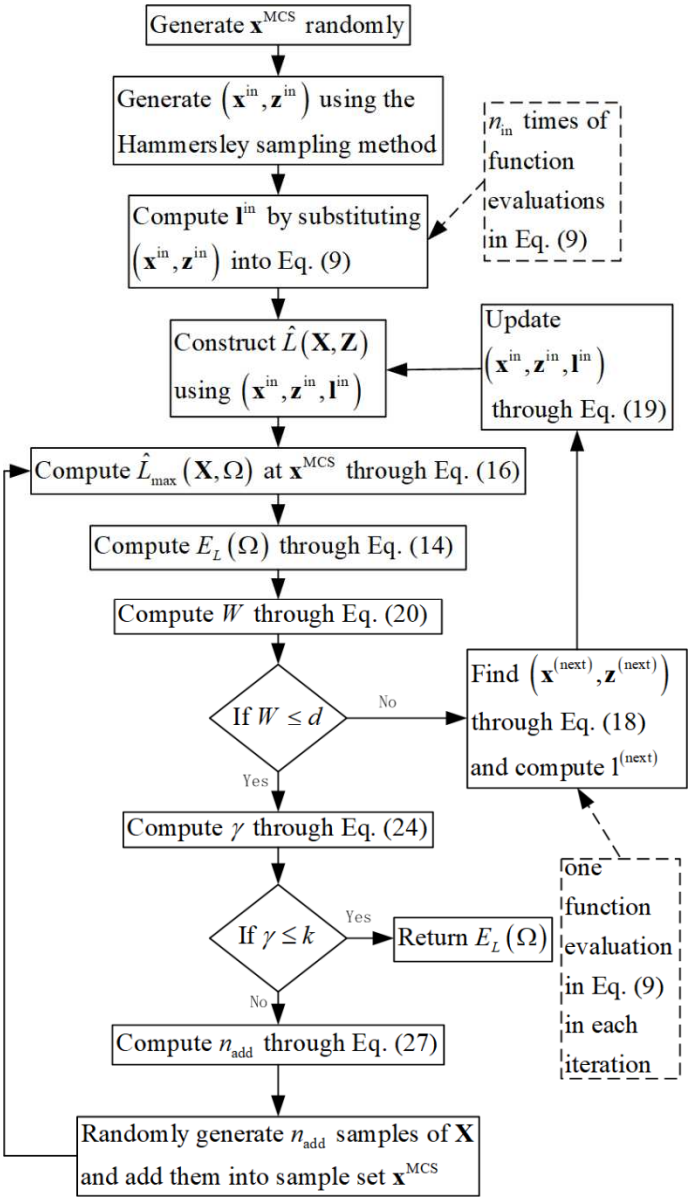

FIGURE 2: DETAILED FLOWCHART

Second, more training points are needed for acceptable accuracy. To overcome this problem, we build a surrogate model $\hat{L}(\mathbf{X}, H, \mathbf{Z})$ with respect to $\mathbf{X}, H$ and $\mathbf{Z}$. Then the surrogate model $\hat{L}[(\boldsymbol{\xi}, \mathbf{X}), \mathbf{Z}]$ with respect to $(\boldsymbol{\xi}, \mathbf{X})$ and $\mathbf{Z}$ is obtained through

$$\hat{L}[(\boldsymbol{\xi}, \mathbf{X}), \mathbf{Z}] = \hat{L}(\mathbf{X}, H, \mathbf{Z}) \circ H(\boldsymbol{\xi}, \mathbf{Z}) = \hat{L}[\mathbf{X}, H(\boldsymbol{\xi}, \mathbf{Z}), \mathbf{Z}] \quad (31)$$

Since the series expansion $H(\boldsymbol{\xi}, \mathbf{Z})$, if truncated, has a simple closed-form expression, if $\hat{L}(\mathbf{X}, H, \mathbf{Z})$ is accurate and efficient, so will be $\hat{L}[(\boldsymbol{\xi}, \mathbf{X}), \mathbf{Z}]$ in Eq. (31). Since the dimension of $\hat{L}(\mathbf{X}, H, \mathbf{Z})$ is $(N_\xi - 1)$, where N_ξ is the dimension of $\boldsymbol{\xi}$, lower than that of $\hat{L}[(\boldsymbol{\xi}, \mathbf{X}), \mathbf{Z}]$, it will be more efficient to train $\hat{L}(\mathbf{X}, H, \mathbf{Z})$ with higher accuracy. When more than one input random fields are involved, the procedures of building the surrogate model $\hat{L}$ are similar.

5. NUMERICAL EXAMPLES

In this section, we use three examples to test the proposed method. The direct MCS is also used to compute the TSD robustness. MCS calls Eq. (11), instead of Eq. (13), to compute the samples of $L_{\max}(\mathbf{X}, \Omega)$. The sample size of MCS is set to 10^5. The results of MCS are treated as the exact ones for the accuracy comparison. The convergence thresholds d, k and n_{in}, are set to 5×10^{-3}, 5×10^{-3}, and 10, respectively. Both methods share the same discretization of Ω.

5.1 An math problem

The QC is given as

$$Y = \sum_{i=1}^{5} X_i^2 + 0.1(Z_1 + Z_2 + 5)^2 \sin(0.1Z_2) \prod_{i=1}^{5} X_i \quad (32)$$

where $(X_1, X_2 \ldots, X_5)$ are five independent identical normal variables with mean and standard deviation being 1 and 0.02, respectively. The domain Ω of $\mathbf{Z} = (Z_1, Z_2)$ is $[0,2] \times [0,5]$. $m(\mathbf{Z})$ is given as

$$m(\mathbf{Z}) = 0.1(Z_1 + Z_2 + 5)^2 \sin(0.1Z_2) \quad (33)$$

and $A(\mathbf{Z}) = \$1000$. Z_1 and Z_2 are discretized into 20 and 50 points, respectively, so $n_\Omega = 10^3$.

The results obtained from the proposed method and MCS are given in Table 1. The robustness computed by the proposed method is $\$2.6592 \times 10^4$, and the robustness by MCS is $\$2.6328 \times 10^4$. The proposed method is very accurate, with a small relative error of 1.0%. Apart from the 10 initial samples, 43 training points are added to update the Gaussian process model, and the proposed method costs with a total of 53 function calls. MCS, however, costs 10^8 function calls, far larger than that of the proposed method. The proposed method adaptively increases the sample size to compute $E_L(\Omega)$, and it can obtain an accurate result with only 463 samples.

TABLE 1: ROBUSTNESS ANALYSIS RESULTS

Methods	Proposed method	MCS
$E_L(\Omega)(\$)$	2.6592×10^4	2.6328×10^4
Relative error (%)	1.0	-
n_{MCS}	463	10^5
n_{call}	53	10^8

5.2 A slider mechanism

Shown in Fig. 3 is a slider mechanism [9]. The location or spatial variables are the offset H and the initial angle θ_0 with the following ranges: $H \in [14.85\text{ m}, 15.15\text{ m}]$ and $\theta_0 \in [-2^\circ, 2^\circ]$. The time span is $t \in [0\text{ s}, 0.1\pi\text{ s}]$. Then the $\mathbf{Z}$ vector is (H, θ_0, t). The random variable vector is $\mathbf{X} = (L_1, L_2)$, which includes two independent random link lengths

$L_1 \sim N(15, 0.015^2)$ m and $L_2 \sim N(35, 0.035^2)$ m. The QC, or the actual position of the slider, is

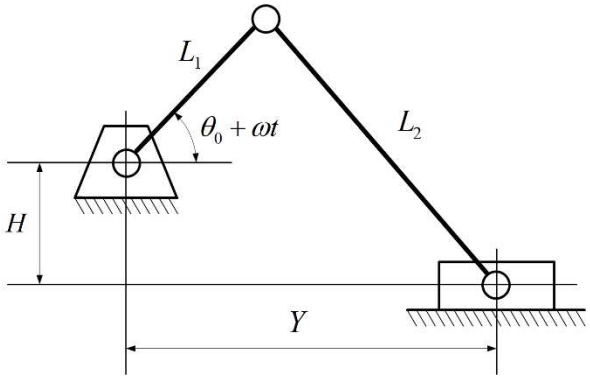

FIGURE 3: A SLIDER MECHANISM

$$Y = L_1 \cos(\theta_0 + \omega t) + \sqrt{L_2^2 - (H + L_1 \sin(\theta_0 + \omega t))^2} \quad (34)$$

where $\omega = 1\ \mathrm{rad/s}$ is the angular velocity. The target QC is

$$m(\mathbf{Z}) = 15 \cos(\omega t) + \sqrt{35^2 - (15 + 15 \sin(\omega t))^2} \quad (35)$$

and $A(\mathbf{Z}) = \$1000/\mathrm{m}^2$. The intervals of h, θ_0 and t are all evenly discretized into 20 points. Accordingly, $\Omega = [14.85\ \mathrm{m}, 15.15\ \mathrm{m}] \times [-2^\circ, 2^\circ] \times [0\mathrm{s}, 0.1\pi\mathrm{s}]$ is discretized into $n_\Omega = 20 \times 20 \times 20 = 8 \times 10^3$ points.

The robustness analysis results are given in Table 2. The proposed method are accurate and efficient with 31 function calls.

TABLE 2: ROBUSTNESS ANALYSIS RESULTS

Methods	Proposed method	MCS
$E_L(\Omega)$ (\$)	88.4325	88.0182
Relative error (%)	0.5	-
n_{MCS}	1492	10^5
n_{call}	31	8×10^8

5.3 A cantilever beam

Shown in Fig. 4 is a cantilever beam. Its span $L = 1$ m. Due to the machining error, the diameter of its cross section is not a constant. Instead, it is modeled as a one-dimensional stationary Gaussian random field $D(x)$. The mean value μ_D and standard deviation σ_D of $D(x)$ are 0.1 m and 0.001 m, respectively. Its autocorrelation coefficient function $\rho_D(x_1, x_2)$ is given as

$$\rho_D(x_1, x_2) = \exp[-(x_1 - x_2)^2] \quad (36)$$

The beam is subjected to a torsion $T(t)$ and a tensile force F at the right endpoint. F is a normal variable with mean μ_F and standard deviation σ_F being 10^3 N and 100 N, respectively. $T(t)$ is a stationary Gaussian process with mean μ_T and standard deviation σ_T being $200\ \mathrm{N \cdot m}$ and $20\ \mathrm{N \cdot m}$, respectively. Its autocorrelation coefficient function $\rho_T(t_1, t_2)$ is given by

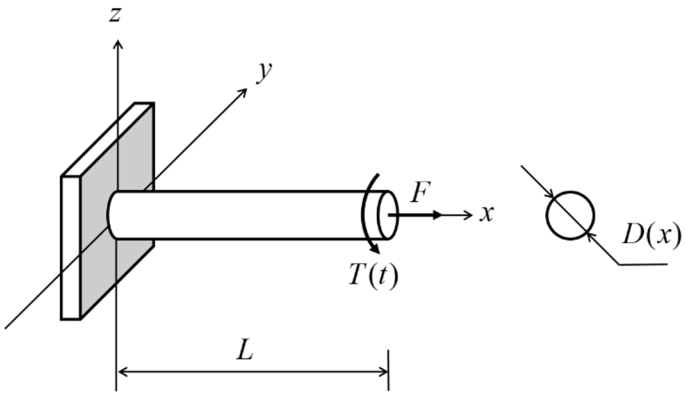

FIGURE 4: A CANTILEVER BEAM

$$\rho_T(t_1, t_2) = \exp\left[-\left(\frac{t_1 - t_2}{2}\right)^2\right] \quad (37)$$

Since there are one stochastic process and one random field, this example involves a more general problem discussed in Sec. 4.6.

The maximum von Misses stress of the beam is the QC and is given by

$$Y = \sqrt{\left(\frac{4F}{\pi D(x)^2}\right)^2 + 3\left(\frac{16T(t)}{\pi D(x)^3}\right)^2} \quad (38)$$

The target $m(\mathbf{Z}) = 0$ and $A(\mathbf{Z}) = \$1000/\mathrm{Mpa}$. The domain Ω of $\mathbf{Z}$ is $[0, 1\ \mathrm{m}] \times [0, 5\ \mathrm{yr}]$ and is evenly discretized into $n_\Omega = 20 \times 50 = 1000$ points.

With $\rho_D(x_1, x_2)$ we can get the autocorrelation coefficient matrix M_D of the one-dimensional random field $D(x)$. Since x is discretized evenly into 20 points in its interval $[0,1\ \mathrm{m}]$, the dimension of M_D is 20×20. The most significant three eigenvalues of M_D are 17.0693, 2.7182 and 0.2026. We use EOLE [17] to generate the series expansion of $D(x)$ and only keep the first three orders. Similarly, we use EOLE to generate the series expansion of $T(t)$ and only keep the first six orders.

The robustness analysis results are given in Table 4. The robustness computed by the proposed method and by MCS are $\$3.8701 \times 10^3$ and 3.8814×10^3, respectively. The relative error of the robustness computed by the proposed method is only -0.3%. The proposed method calls the original quality loss function 11 times, showing its high efficiency.

TABLE 4: ROBUSTNESS ANALYSIS RESULTS

Methods	Proposed method	MCS
$E_L(\Omega)$ (\$)	3.8701×10^3	3.8814×10^3
Relative error (%)	-0.3	-
n_{MCS}	1000	10^5
n_{call}	11	10^8

6. CONCLUSIONS

Existing robustness analysis methods only consider the static or time-dependent problems. More general are time-and space-dependent problems. In this paper, we propose to use the expectation of the maximal value of the quality loss function with respect to time and space to measure the time- and space-dependent robustness. This metric can fully take into consideration of the autocorrelation of the time- and space-dependent quality loss function.

An efficient method based on the Gaussian process model and efficient global optimization is proposed to compute the time- and space-dependent robustness metric. Training points are adaptively added to update the Gaussian process model. A stopping criterion is developed to measure the expected improvement of the current value of the robustness metric. Since a sampling method is used to compute the robustness, uncertainty in the result is inevitable. To control the uncertainty and then get an accurate result, we develop an algorithm to adaptively increase the sample size.

Three examples show that the proposed method can deal with problems with different uncertainty level. For problems with high uncertainty level, the proposed method automatically uses large sample size. The proposed method can obtain accurate robustness very efficiently.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under grant CMMI 1923799 (formerly 1727329).

REFERENCES

[1] Taguchi, G., 1993, Taguchi on Robust Technology Development. Bringing Quality Engineering Upstream, ASME Press, New York.

[2] Phadke, M. S., 1995, Quality engineering using robust design, Prentice Hall PTR.

[3] Du, X., 2012, "Toward Time-Dependent Robustness Metrics," Journal of Mechanical Design, 134(1), p. 011004.

[4] Du, X., Sudjianto, A., and Chen, W., 2004, "An integrated framework for optimization under uncertainty using inverse reliability strategy," Journal of Mechanical Design, 126(4), pp. 562-570.

[5] Gu, X., Renaud, J. E., Batill, S. M., Brach, R. M., and Budhiraja, A. S., 2000, "Worst case propagated uncertainty of multidisciplinary systems in robust design optimization," Structural and Multidisciplinary Optimization, 20(3), pp. 190-213.

[6] Tsui, K.-L., 1999, "Modeling and analysis of dynamic robust design experiments," IIE Transactions, 31(12), pp. 1113-1122.

[7] Wu, F.-C., 2009, "Robust design of nonlinear multiple dynamic quality characteristics," Computers & Industrial Engineering, 56(4), pp. 1328-1332.

[8] Goethals, P. L., and Cho, B. R., 2011, "The development of a robust design methodology for time‐oriented dynamic quality characteristics with a target profile," Quality and Reliability Engineering International, 27(4), pp. 403-414.

[9] Wei, X., and Du, X., 2019, "Uncertainty Analysis for Time- and Space-Dependent Responses With Random Variables," Journal of Mechanical Design, 141(2), p. 021402.

[10] Williams, C. K., and Rasmussen, C. E., 2006, Gaussian processes for machine learning, MIT Press Cambridge, MA.

[11] Lophaven, S. N., Nielsen, H. B., and Søndergaard, J., 2002, DACE: a Matlab kriging toolbox, Citeseer.

[12] Hu, Z., and Du, X., 2015, "Mixed efficient global optimization for time-dependent reliability analysis," Journal of Mechanical Design, 137(5), p. 051401.

[13] Zhu, Z., and Du, X., 2016, "Reliability analysis with Monte Carlo simulation and dependent Kriging predictions," Journal of Mechanical Design, 138(12), p. 121403.

[14] Jian, W., Zhili, S., Qiang, Y., and Rui, L., 2017, "Two accuracy measures of the Kriging model for structural reliability analysis," Reliability Engineering & System Safety, 167, pp. 494-505.

[15] Li, M., Bai, G., and Wang, Z., 2018, Time-variant reliability-based design optimization using sequential kriging modeling.

[16] Hu, Z., and Mahadevan, S., 2016, "A single-loop kriging surrogate modeling for time-dependent reliability analysis," Journal of Mechanical Design, 138(6), p. 061406.

[17] Sudret, B., and Der Kiureghian, A., 2000, Stochastic finite element methods and reliability: a state-of-the-art report, Department of Civil and Environmental Engineering, University of California Berkeley.

[18] Gabrel, V., Murat, C., and Thiele, A., 2014, "Recent advances in robust optimization: An overview," European Journal of Operational Research, 235(3), pp. 471-483.

[19] Bovier, A., 2005, "Extreme values of random processes," Lecture Notes Technische Universität Berlin.

[20] Kotz, S., and Nadarajah, S., 2000, Extreme value distributions: theory and applications, World Scientific, London.

[21] Zienkiewicz, O. C., Taylor, R. L., Nithiarasu, P., and Zhu, J., 1977, The finite element method, McGraw-hill London.

[22] Jones, D. R., Schonlau, M., and Welch, W. J., 1998, "Efficient global optimization of expensive black-box functions," Journal of Global optimization, 13(4), pp. 455-492.

[23] Chen, W., Tsui, K.-L., Allen, J. K., and Mistree, F., 1995, "Integration of the response surface methodology with the compromise decision support problem in developing a general robust design procedure," ASME Design Engineering Technical Conference, pp. 485-492.

[24] Hosder, S., Walters, R., and Balch, M., 2007, "Efficient sampling for non-intrusive polynomial chaos applications with multiple uncertain input variables," Proc. 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, p. 1939.