EFFICIENT UNCERTAINTY QUANTIFICATION WITH HIGH DIMENSIONALITY

by

Jianhua Yin

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

School of Mechanical Engineering

West Lafayette, Indiana

May 2022

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Xiaoping Du, Co-Chair

Department of Mechanical and Energy Engineering

Dr. Jitesh H. Panchal, Co-Chair

School of Mechanical Engineering

Dr. Jing Zhang

Department of Mechanical and Energy Engineering

Dr. Honglang Wang

Department of Mathematical Sciences

Approved by:

Dr. Nicole L. Key

Dedicated to my parents for their yearly unconditional love and support, and to my beloved fiancée, Bingrong Xu, for her support and encouragement.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor, Dr. Xiaoping Du, for his continuous encouragement, insightful guidance, and immense help and support during my graduate study at Purdue University. He patiently guides me through every stage of my research with invaluable suggestions and inspiration. His constant encouragement and trust endow me with tremendous confidence in exploring new research areas and challenging myself to do better work. His rigorous and diligence attitude to work will continuously inspire me in my future career. It is a great honor and privilege to work with him.

I also would like to extend my gratitude to my co-advisor, Dr. Jitesh H. Panchal, my dissertation committee members, Dr. Jing Zhang, Dr. Honglang Wang. Without their insightful comments and guidance, this dissertation would not possibly be finished.

In addition, I would like to thank Dr. Lingxi Li for the excellent collaborations and valuable suggestions on the transportation project. The kind instruction and help from Mr. Golub Michael Ira in Mechanical Engineering Laboratory are highly appreciated.

Moreover, I would like to thank my labmates and friends, Mr. Hao Wu, Ms. Huiru Li, Dr. Li Wan, Mr. Yikang Yu, Mr. Yuheng Wang, Dr. Jian Zhang, Mr. Dan Shen, Mr. Taotao Jing, Dr. Yang Yang, Dr. Gang Liu, Dr. Zhuolong Zhou, Mr. Junyi Yang, Mr. Zhen Zhang, and many other friends, for their kind help and valuable discussion. The discussion with them greatly broadens my eyes and enriches my knowledge of different disciplines and research areas. The time spent together would be an unforgettable memory. I would like to express my deepest appreciation and love to my fiancée, Bingrong Xu, for her constant accompanies, encouragement, and understanding, to my parents for their unconditional love and support throughout my life.

The financial support from the National Science Foundation under Grand No. 1923799, the Institute of Integrative at Indiana University Purdue University Indianapolis (IUPUI), and IUPUI are gratefully acknowledged.

PREFACE

This is an article-based dissertation consisting of two published journal papers and one manuscript submitted to a journal for publication.

Pages 19–45 were initially published in the proceedings of *ASME 2021 International Design and Engineering Technical Conferences* & *Computers and Information in Engineering Conference*, and then were selected to publish in *ASME Journal of Mechanical Design*.

Pages 46–81 have been published in *Structural Safety*. Minor changes are applied to address the comments from committee members.

Pages 82–109 are preparing to submit to a journal for publication, whose preliminary work has been published in the proceedings of *AIAA SciTech 2022 Forum*.

All papers have been prepared in the style utilized by Purdue University.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Uncertainty exists everywhere in scientific and engineering applications. To avoid potential risk, it is critical to understand the impact of uncertainty on a system by performing uncertainty quantification (UQ) and reliability analysis (RA). However, the computational cost may be unaffordable using current UQ methods with high-dimensional input. Moreover, current UQ methods are not applicable when numerical data and image data coexist.

To decrease the computational cost to an affordable level and enable UQ with special high dimensional data (e.g. image), this dissertation develops three UQ methodologies with high dimensionality of input space. The first two methods focus on high-dimensional numerical input. The core strategy of Methodology 1 is fixing the unimportant variables at their first step most probable point (MPP) so that the dimensionality is reduced. An accurate RA method is used in the reduced space. The final reliability is obtained by accounting for the contributions of important and unimportant variables. Methodology 2 addresses the issue that the dimensionality cannot be reduced when most of the variables are important or when variables equally contribute to the system. Methodology 2 develops an efficient surrogate modeling method for high dimensional UQ using Generalized Sliced Inverse Regression (GSIR), Gaussian Process (GP)-based active learning, and importance sampling. A cost-efficient GP model is built in the latent space after dimension reduction by GSIR. And the failure boundary is identified through active learning that adds optimal training points iteratively. In Methodology 3, a Convolutional Neural Networks (CNN) based surrogate model (CNN-GP) is constructed for dealing with mixed numerical and image data. The numerical data are first converted into images and the converted images are then merged with existing image data. The merged images are fed to CNN for training. Then, we use the latent variables of the CNN model to integrate CNN with GP to quantify the model error using epistemic uncertainty. Both epistemic uncertainty and aleatory uncertainty are considered in uncertainty propagation.

The simulation results indicate that the first two methodologies can not only improve the efficiency but also maintain adequate accuracy for the problems with high-dimensional numerical input. GSIR with active learning can handle the situations that the dimensionality

cannot be reduced when most of the variables are important or the importance of variables are close. The two methodologies can be combined as a two-stage dimension reduction for high-dimensional numerical input. The third method, CNN-GP, is capable of dealing with special high-dimensional input, mixed numerical and image data, with the satisfying regression accuracy and providing an estimate of the model error. Uncertainty propagation considering both epistemic uncertainty and aleatory uncertainty provides better accuracy. The proposed methods could be potentially applied to engineering design and decision making.

1. INTRODUCTION

1.1 Background

Uncertainty [1] is the opposite of certainty, a state of limited knowledge or lack of information to understand the exact state of physical systems or the future outcome which is called quantify of interest (QoI). Unfortunately, uncertainty or natural variation is inevitable in scientific and engineering applications, such as manufacturing error, heterogeneous material properties, and varied system load. Those uncertain parameters are the input of the uncertainty quantification (UQ) task. Given the distributions of the input, UQ is to obtain the probability density function (PDF) or cumulative density function (CDF) of QoI so the effect of uncertainty on QoI could be understood.

Reliability [2] is the probability that a component or system fulfills its intended function without failure given the existence of uncertainty. Reliability analysis (RA) is a special case of UQ and also is one of the most applications of UQ, which usually focuses on extreme events or tail events with low probability. By identifying the failure boundary between the safe and failure region of model response and integrating the join PDF of input variables in the safe region, reliability is obtained.

UQ and RA are important for managing and avoiding the potential risks at the design stage of products, such as aircraft [3]–[5], automobiles [6]–[10], and many infrastructures [11]–[14]. Without knowing the effect of uncertainty on the target product, catastrophic consequences could be introduced.

To investigate the impact of uncertainty, computer simulations are increasingly used for UQ and RA in facing the prevalent existence of uncertainty in the physical world. It repeats the simulation with considerable times. The model can be physical models which represent the sophisticated physical principles solved by numerical methods or can be data-based models. However, those models usually are either computationally demanding or too expensive in collecting enough data for sufficient accuracy.

Current UQ methods can be categorized into three types. The first is called sampling methods or simulation methods, such as Monte Carlo Simulation (MCS) [15], [16], Importance Sampling (IS) [17], [18], and Subset Simulation (SS) [19], [20]. MCS is the most accurate method with sufficient samples, which induces high computational costs. IS and SS focus on the extreme events or part of the probability space. Their computational cost is lower than MCS, but a relatively large number of samples are essential to obtain accurate results. Also, IS and SS may not be accurate in the whole probability space. The computational cost of those methods usually is unaffordable for engineering design when the computation model is black-box simulations, although those methods are insensitive to the dimensionality of the input space.

The second type is the most probable point (MPP) based approximation methods [21]–[24]. Typical approximation methods include the first-order reliability method (FORM) [22], the second-order reliability method (SORM) [22], and the first-order/second-order saddle-point point approximation (SPA) methods [25]–[27]. These methods simplify the original computational model by Taylor expansion at the MPP. The first order and second order represent the Taylor expansion order. In general, the second-order methods are more accurate than the first-order methods with the increasing nonlinearity of the given computation model, but the efficiency is worse than the latter. The computational cost of those methods is directly related to the dimensionality of the input space. The complicated engineering systems nowadays bring challenges due to the high dimensionality and nonlinearity.

The third type is called meta-modeling or surrogate-modeling methods [28]–[31]. Those methods create computationally economical regression models using response surface method (RSM) [29], classical machine learning (ML) methods [30], [32], [33], or deep learning (DL) [34]–[36]. The commonly used classical ML methods are Gaussian Process (GP) modeling (or Kriging) [30], [37], [38], Support Vector Machine (SVM) [32], [39], [40], and shallow Artificial Neural Networks (ANN) [33], [41]. Many meta-modeling methods have been developed in the past years and have achieved considerable success. However, severe over-fitting or under-fitting exist in those methods in high-dimensional probability space without sufficient samples. Generating more samples means increasing cost.

In addition, the current meta-modeling methods for UQ target numerical input but are not able to accommodate image data or mixed data. In the age of data exploration, we have more complicated data forms with high dimensionality, such as images and videos. For example, doctors use engineering tools, Computational Fluid Dynamics (CFD), to help with the severity assessment of stenosis [42]–[44]. The spatial domain of CFD simulation is extracted from the CT image of patients. If a regression model that is capable of handling image data with limited CFD simulations can be obtained, the time-consuming CFD simulation can be replaced by the cheap regression model.

In summary, to improve the efficiency of UQ in high-dimensional probability space, integrating dimension reduction techniques to current UQ methods is critical. Also, novel regression methods that are capable of handling special high-dimensional data (e.g., image data) are needed for UQ. This dissertation attempts to solve the above problems. The proposed methods can potentially reduce the design cost in engineering design and lower the cost to an affordable level. Other areas, such as engineering management, reliability engineering, and statistics, could also benefit from the proposed methods in this dissertation.

1.2 Research Objective

The objective of this dissertation is to mitigate the curse of dimensionality of current UQ and improve the efficiency of UQ for high-dimensional problems. Three research tasks labeled as RT1, RT2, and RT3 are carried out to achieve the objective. RT1 and RT2 can be considered as a two-stage dimension reduction to achieve efficient reliability analysis. RT3 is to develop a methodology that can deal with special high-dimensional input, the mixed image and numerical data. In this dissertation, we assume that the numerical inputs are independent. When the inputs are dependent, we can transform them into independent variables [45].

The first stage dimension reduction (RT1) focuses on the problems that their high-dimensional input variables can be divided into two groups, important and unimportant variables according to their influence on the response [46], [47]. Traditional methods fix the unimportant variables at their means and overlook their coupled influence in reliability analysis. However, their influence may not be insignificant so that the error is introduced. The proposed method counts the influence of both important and unimportant variables. The unimportant variables are fixed at their one-step MPP or their percentiles so that the dimensionality is reduced. After that, reliability analysis is performed in a low-dimensional space with high efficiency. The final reliability is obtained by counting the influence of both the important and unimportant variables.

The second stage dimension reduction (RT2) is to address the situation that the dimensionality cannot be reduced after the first stage dimension reduction. A nonlinear dimension reduction technique from sufficient dimension reduction (SDR) theory [48], [49], called generalized sliced inverse regression (GSIR) [50], is used for dimension reduction. Theoretically, SDR can preserve the distribution of the model response after dimension reduction. Combining with GP and active learning, the computational cost of reliability analysis is significantly decreased.

RT3 aims to develop an accurate regression model that can not only deal with special high-dimensional input, mixed data (image and numerical data), but also predict model uncertainty along with the prediction. The proposed regression model is then used for UQ. The strategy is first converting numerical data into image data and then merging them with existing images. The merged images are fed to Convolutional Neural Network (CNN) [51]–[53] for model training. Then, the latent variables are retrieved by treating CNN as a supervised dimension reduction process. A GP model is constructed with respect to the latent variables and model response. The ultimate model named CNN-GP is used for prediction and UQ.

This work could fill up the gap in dimension scalability with a set of enabling tools that can be used practically in large-scale design. The designers could perform what-if analysis and optimization to come up with cost-efficient, reliable designs.

1.3 Dissertation Organization

To address the aforementioned three search tasks, three dimension reduction aided UQ methodologies are proposed resulting in three research papers which are provided in Chapter 2, Chapter 3, and Chapter 4. As shown in Figure 1.1, the three research tasks serve for the same objective, UQ with high dimensionality, forming of the main body of the dissertation.

Chapter 2 (RT1) focuses on the high-dimensional numerical input that can be divided into important and unimportant variables according to their influence on the system. Chapter 3 (RT2) proposes another high dimensional UQ method to address the issue that RT1 may not

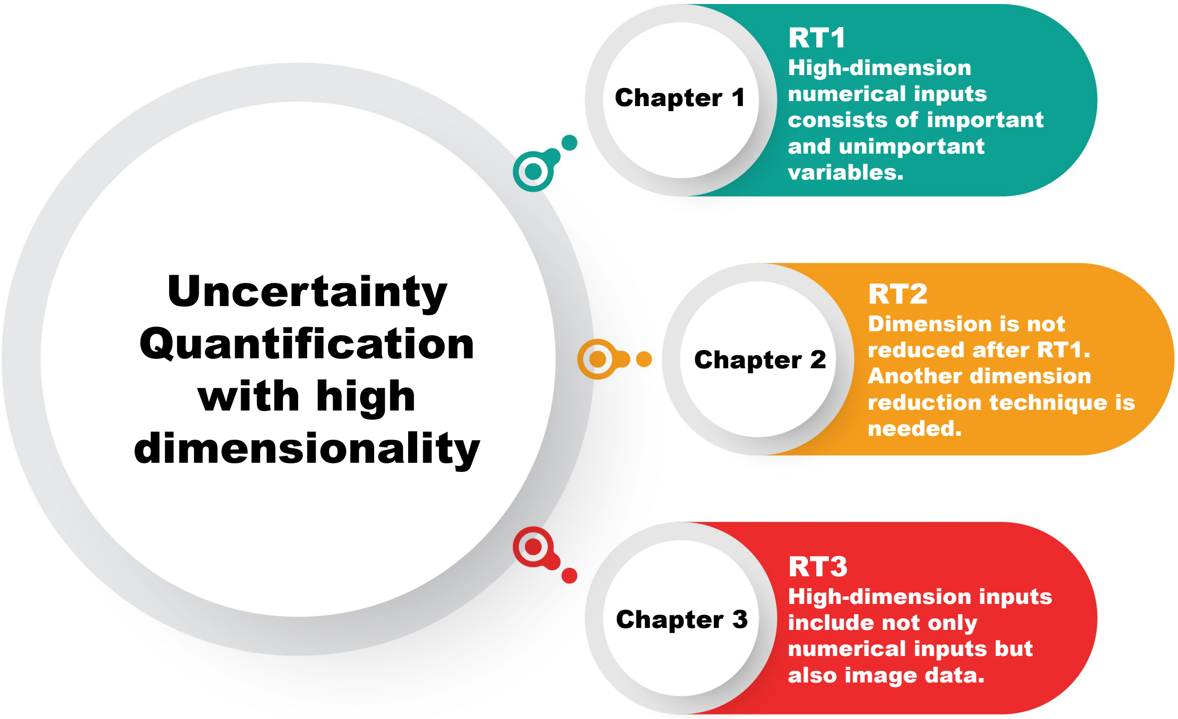

Uncertainty Quantification with high dimensionality
Chapter 1
RT1
High-dimension numerical inputs consists of important and unimportant variables.
Chapter 2
RT2
Dimension is not reduced after RT1. Another dimension reduction technique is needed.
Chapter 3
RT3
High-dimension inputs include not only numerical inputs but also image data.

Figure 1.1. Structure of the dissertation.

be applicable: most of the input variables are important and the dimensionality cannot be reduced by RT1. In Chapter 4 (RT3), the high-dimensional space is extended where image data and numerical data coexist. An UQ method that can accommodate both image and numerical data is proposed. Conclusions are provided in Chapter 5.

2. HIGH-DIMENSIONAL RELIABILITY METHOD ACCOUNTING FOR IMPORTANT AND UNIMPORTANT INPUT VARIABLES

Jianhua Yin[1,2], Xiaoping Du[1]

[1]Department of Mechanical and Energy Engineering, Indiana University - Purdue University Indianapolis, 723 W. Michigan Street, Indianapolis, IN 46202-5195

[2]School of Mechanical Engineering, Purdue University, West Lafayette, IN, 47907, United States

Published in *ASME. J. Mech. Des.*, doi: https://doi.org/10.1115/1.4051982

Author Contributions

The authors confirm contribution to the paper as follows: Jianhua Yin and Xiaoping Du designed the study and contributed to the writing of the manuscript. Jianhua Yin developed detailed methodology, testing problems, and the code.

2.1 Background

In engineering design, physics-based reliability is commonly used to predict the probability of failure using physical models derived from physical principles. Such a model is called a limit-state function and is given by

$$Y = g(\mathbf{X}) \tag{2.1}$$

where $\mathbf{X}$ is a vector to represent input random variables, and Y is a response that indicates the occurrence of a failure.

Physics-based reliability methods can be divided into three categories: numerical methods [21]–[23], [54], [55], surrogate methods [28]–[31], [56], [57], and simulation methods [58]–

[61]. Typically, numerical methods simplify the limit-state function using the first or second order Taylor expansion. The reliability is approximated by the simplified function. The surrogate methods construct an easy-access model utilizing sensitivity analysis, Design of Experiments (DOE), active learning methods, etc., and the reliability obtained by calling the surrogate model instead of the original limit-state function. However, both numerical and surrogate methods suffer from the curse of dimensionality that makes reliability analysis computationally expensive for high-dimensional problems. Because reliability prediction repeatedly calls limit-state functions which are typically complex, resource-intensive numerical models. The number of function calls grows drastically as the increase of dimensionality of the input variables. Although the efficiency of simulation methods, such as Monte Carlo Simulation (MCS) [62] and Importance Sampling (IS) method [63], is not affected by the dimensionality, they are still computationally expensive when the reliability is high and may not be practically used in engineering design.

High-dimensional reliability analysis is encountered in many engineering and science fields [64]–[69]. Current high-dimensional reliability analysis methods are roughly classified into three types. The first type [70]–[73] uses high-dimensional model representation (HDMR) to decompose a high dimensional limit-state function $g(\mathbf{X})$ into the sum of several lower-dimensional functions. The moments (means, variance, etc.) of the response can be approximated by several low dimensional numerical integrations. However, the accuracy of the reliability obtained by HDMR may not be accurate enough if the interaction terms are dominant. The low dimensional functions are usually approximated by Taylor expansion, which also could introduce errors. Although the accuracy of the reliability assessment can be improved by increasing the approximation order, the number of function evaluations may increase drastically. Several recent studies [74]–[76] combine adaptive metamodeling approaches (Polynomial Chaos Expansion, Kriging) and statistical model selection methods. Their goal is to find the optimal integration points or training points for metamodeling. The balance between the prediction accuracy and efficiency is still a challenge.

The second type of method [35], [77]–[79] combines dimension reduction with surrogate modeling and machine learning. Three steps are usually involved. Step 1 is the dimension reduction performed by the sliced inverse regression (SIR) [48], [79], or other methods [70],

[78] at specific training points , usually generated through DOE [80]. Important input variables are identified. In Step 2, a surrogate model is constructed with respect to important input variables in the reduced dimensional space. Many regression and machine learning methods could be used for this purpose, including Polynomial Chaos Expansion (PCE) [81], Gaussian Process Regression (GPR) [82], Support Vector Machines (SVM) [83], and Neural Networks (NNs) [35]. Step 3 is the surrogate model validation. After the accuracy of the surrogate model is validated by MCS, it is used to estimate the reliability. Sufficient training points are needed to ensure good accuracy of the surrogate model. The number of training points, thereby the number of function calls, increases greatly with the increase of dimensionality of input variables.

The third most commonly used method is principal component analysis (PCA) [84], [85]. PCA reduces the dimension of the input variables by making use of the correlations between the input variables. Therefore, PCA works well for the elements of input variables that are strongly correlated. When the input variables are independent or only weakly correlated, PCA may not work well for dimension reduction. Besides, PCA does not use the information of the response Y, and it is, therefore, an unsupervised dimension reduction technique. Although dimension reduction is optimal in the given data space, it may be suboptimal for the entire regression space.

Overall, despite the progress, numerous challenges remain in the path toward routinely accommodating high dimensional problems in reliability analysis. In most of the successful applications, only dozens of random input variables can be practically handled except the special cases involving functional data [77], [81]. However, the dimension in input variables could easily add up to hundreds or thousands in system design. For example, the aircraft wing optimization design [86] involves structural mechanics and aerodynamics. The numbers of design variables, random variables, and constraints could be in hundreds or thousands. Moreover, when the reliability requirement is high, accurately predicting the reliability is extremely computational demanding.

In real engineering applications, not all the elements of $\mathbf{X}$ contribute significantly to the response Y. The majority elements of $\mathbf{X}$ may have insignificant effects that are therefore unimportant variables. Their total effect, however, may not be negligible because the unimportant variables may count for most of $\mathbf{X}$. Traditional dimension reduction methods usually neglect the contribution of the unimportant variables because they are fixed at their means, which can lead to a error.

In this study, we account for the total effect of unimportant variables by fixing them at their percentiles so that the dimension is reduced but the influence of unimportant variables is not neglected. The proposed method does not require random sampling for dimension reduction; instead, it bases on a numerical method, specifically the First Order Reliability Method (FORM). After dimension reduction, any reliability method with higher accuracy can be used to predict the reliability since the computational effort will be reduced significantly in the reduced space. Then the predicted reliability is integrated with the contribution of the unimportant variables to produce the final reliability prediction.

The remainder of this paper is organized as follows. Section 2.2 reviews the methodologies that this study uses. Section 2.3 discusses the details of the proposed method, followed by three examples in Section 2.4. The conclusions are provided in Section 2.5.

2.2 Review

In this section, we briefly review the basic knowledge that is related to the proposed method, including FORM, the Second Order Reliability Method (SORM), and the Second Order Saddlepoint Approximation (SOSPA). The rules of symbols in this paper are: 1) a capitalized letter in bold denotes a vector of random variables (e.g. $\mathbf{X}$ or $\mathbf{U}$), 2) a italicized lower-case letter in bold denotes a vector of deterministic variables (e.g. $\boldsymbol{x}$ or $\boldsymbol{u}$), 3) an italicized capital letter denotes a random variable (e.g. X or U), and 4) an italicized lowercase letter of denotes a deterministic variable (e.g. x or u).

2.2.1 FORM and SORM

The reliability is defined by the following probability:

$$R = \Pr\{g(\mathbf{X}) \geq 0\} \tag{2.2}$$

The probability of failure p_f is then given by

$$p_f = 1 - R = \Pr\{g(\mathbf{X}) < 0\} = \int_{g(\mathbf{X})<0} f_X(\boldsymbol{x}) < 0 d\boldsymbol{x} \tag{2.3}$$

where $f_X(\boldsymbol{x})$ is the joint probability density function (PDF) of $\mathbf{X}$. The limit-state function $g(\mathbf{X})$ is usually a nonlinear function. In this study, we assume all the elements in $\mathbf{X}$ are independent. Directly integrating the PDF in the failure region ($g(\mathbf{X}) < 0$) is often impractical and computationally expensive. It is the reason that many approximation methods have been developed, including FORM [21] and SORM [22], where three steps are involved.

1) Transform $\mathbf{X}$ to be the standard normal variables $\mathbf{U}$ by

$$F_{X_{\mathrm{i}}}\left(X_{\mathrm{i}}\right) = \Phi\left(U_{\mathrm{i}}\right) \tag{2.4}$$

where $F_{X_{\mathrm{i}}}(\cdot)$ and $\Phi(\cdot)$ represent the cumulative density function (CDF) of X_{i} and U_{i}, respectively. Denote the transformation by $\mathbf{X} = T(\mathbf{U})$, and Eq. (2.3) is rewritten as

$$\Pr\{g(\mathbf{X}) < 0\} = \int_{g(T(\mathbf{U}))<0} f_{\mathbf{U}}(T(\boldsymbol{u})) < 0 d\boldsymbol{u} \tag{2.5}$$

where $f_{\mathbf{U}}(\cdot)$ is the joint PDF of $\mathbf{U}$.

2) Find the most probable point (MPP) which is a point with the highest PDF on the surface of $g(\mathbf{U}) = 0$. Geometrically, MPP has the shortest distance from the surface to the origin in U-space, and then MPP ($\mathbf{u}^*$) is found by

$$\begin{aligned} \min_{\mathbf{u}} \quad & \beta = \|\mathbf{u}\| \\ \text{s.t.} \quad & g(\mathbf{U}) = 0 \end{aligned} \tag{2.6}$$

where $\|\cdot\|$ stands for the length of a vector. $\beta = \|\mathbf{u}^*\|$ is the reliability index because it is related to the probability of failure as will be shown in Eq. (2.9).

3) Approximate the limit-state function linearly (FORM) or quadratically (SORM) at $\mathbf{u}^*$. The use of $\mathbf{u}^*$ can minimize the error of the approximation. The two approximations are given by

$$g(\mathbf{U}) \approx g\left(\mathbf{u}^*\right) + \nabla g\left(\mathbf{u}^*\right)^{\mathrm{T}}\left(\mathbf{U}-\mathbf{u}^*\right) \tag{2.7}$$

$$g(\mathbf{U}) \approx g\left(\mathbf{u}^*\right) + \nabla g\left(\mathbf{u}^*\right)^{\mathrm{T}}\left(\mathbf{U}-\mathbf{u}^*\right) + \frac{1}{2}\left(\mathbf{U}-\mathbf{u}^*\right)^{\mathrm{T}} H\left(\mathbf{u}^*\right)\left(\mathbf{U}-\mathbf{u}^*\right) \tag{2.8}$$

where $\nabla g\left(\mathbf{u}^*\right)$ and $H\left(\mathbf{u}^*\right)$ are the gradient and the Hessian matrix of $g(T(\mathbf{U}))$ with respect to $\mathbf{u}^*$, respectively.

After the three steps, the probability of failure calculated by FORM is given by

$$p_f = \Phi(-\beta) \tag{2.9}$$

As mentioned previously, β is called the reliability index. When FORM is used, β also is the magnitude of the MPP as indicated in Eq. (2.6). Therefore, we call β from FORM the FORM-reliability index throughout the paper. The solution from SORM is more accurate in general and is obtained by multiplying Eq. (2.9) with a correction term [22].

2.2.2 SOSPA

SOSPA [26] is a second-order approximation method based on SORM and saddlepoint approximation (SPA) [25], [87]. SOSPA uses the cumulant generating function (CGF) $K_Y(t)$, which can be derived analytically from the approximated response in Eq. (2.8). Once $K_Y(t)$ is available, the saddlepoint t_s is obtained by solving

$$K_Y'(t) = 0 \tag{2.10}$$

where $K_Y'(t)$ is the first order derivative of the CGF. Then, p_f is computed by [88]

$$p_f = \Phi(\omega) + \phi(\omega)\left(\frac{1}{\omega} - \frac{1}{v}\right) \tag{2.11}$$

where $\phi(\cdot)$ represents the PDF of the standard normal distribution.

$$\omega = \operatorname{sgn}(t_s)\left\{2\left[-K_Y(t_s)\right]\right\}^{\frac{1}{2}} \tag{2.12}$$

$$v = t_s\left[K_Y''(t_s)\right]^{\frac{1}{2}} \tag{2.13}$$

where sgn$(\cdot)$ is the signum function, which equals to 1, -1, or 0 when t_s is positive, negative or zero; $K_Y''(t_s)$ is the second order derivative of the CGF with respect to t.

2.3 Methodology

The distinctive strategy of the proposed method is to use an accurate reliability method in the reduced space and accounts for the contributions of both important and unimportant input variables to the reliability.

2.3.1 Overview

The purpose of dimension reduction is to identify important and unimportant variables in $\mathbf{X}$. We will use FORM to perform dimension reduction since the MPP from FORM can directly measure the importance of input variables for two reasons. First, the reliability is determined by the FORM-reliability index or the magnitude of the MPP since $\beta = \|\mathbf{u}^*\| = \sqrt{\sum_{\mathrm{i}=1}^{n}\left(u_{\mathrm{i}}^*\right)^2}$; second, the components of the MPP $\mathbf{u}^* = (u_{\mathrm{i}}^*)_{\mathrm{i}=1,n}$ determine the importance of the elements of $\mathbf{X}$ or their contributions to the reliability. As shown in Figure 2.1, a farther distance from the mean (or median) means a larger value of the MPP component and therefore a higher contribution. Hence, we can use the MPP components to identify both important and unimportant input variables. Since the MPP components of the unimportant input variables do not change significantly during the MPP search, we propose to use the MPP obtained from the first iteration of the MPP search. This can greatly reduce the computational effort.

Once the MPP is obtained from the first iteration, important and unimportant input variables are identified by their MPP components. Then, the subsequent analysis will be

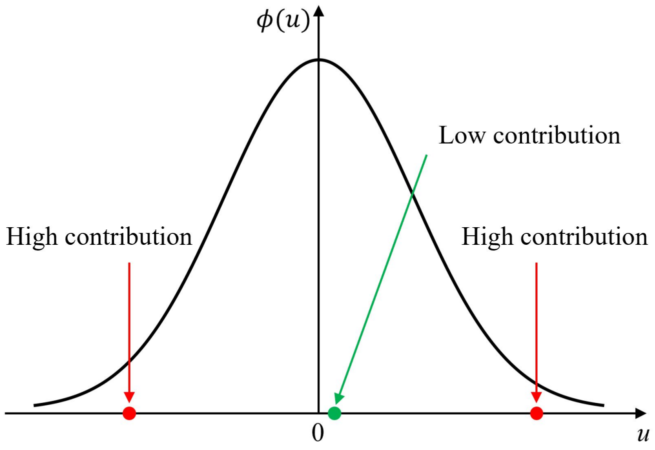


Figure 2.1. Percentile of a random variable.

conducted with only important variables. A reliability method with higher accuracy can be used with the unimportant input variables fixed at their MPP components. Using a high accurate reliability method is affordable because the number of function calls can be reduced in the reduced space. Then the final reliability is obtained by integrating the reliability obtained in the reduced space and the FORM-reliability index of unimportant input variables.

The proposed method involves three steps: 1) dimension reduction, 2) reliability analysis in the reduced space, and 3) reliability analysis in the original space.

2.3.2 Dimension Reduction

The purpose of the first step is to identify important and unimportant input variables. This step involves the first iteration of the MPP search that starts from the origin of the U-space. By Setting the initial point at the origin $\mathbf{u}_0 = (0, 0, \ldots, 0)^{\mathrm{T}}$, we obtain the gradient $\nabla g\left(\mathbf{u}_0\right)$ and approximate the limit-state function by

$$g(\mathbf{U}) \approx g\left(\mathbf{u}_0\right) + \nabla g\left(\mathbf{u}_0\right)^{\mathrm{T}} \mathbf{U} \tag{2.14}$$

The unit vector α of $\nabla g(\mathbf{U})$ at $\mathbf{u}_0$ is given by

$$\alpha = \frac{\nabla g(\mathbf{u}_0)}{\|\nabla g(\mathbf{u}_0)\|} \tag{2.15}$$

Then the FORM-reliability index of one-step MPP is obtained by

$$\beta_1 = \beta_0 + \frac{g(\mathbf{u}_0)}{\|\nabla g(\mathbf{u}_0)\|} = \frac{g(\mathbf{u}_0)}{\|\nabla g(\mathbf{u}_0)\|} \tag{2.16}$$

Using the fact that the MPP vector is in the opposite direction of the gradient [89], we have the first iteration of the MPP $\mathbf{u}_1$.

$$\mathbf{u}_1 = -\beta_1 \boldsymbol{\alpha} = -\frac{g(\mathbf{u}_0)\nabla g(\mathbf{u}_0)}{\|\nabla g(\mathbf{u}_0)\|^2} \tag{2.17}$$

And it can be proved that $\beta_1 = \|\mathbf{u}_1\|$ holds for Eqs. (2.16) and (2.17).

We now discuss how to distinguish important input variables from unimportant ones by using the first-iteration MPP. The probability of failure is approximated by $p_f = \Phi(-\beta_1) = \Phi\left(-\sqrt{u_{11}^2 + u_{12}^2 + \ldots u_{1n}^2}\right)$, where $u_{1\mathrm{i}}$ is the i-th component of $\mathbf{u}_1$. The magnitudes of the components of $\mathbf{u}_1$ therefore indicate their importance to the probability of failure. More specifically, we examine the sensitivity of p_f with respect to the components of $\mathbf{u}_1$. The sensitivity is defined by

$$s_\mathrm{i} = \frac{\partial p_f}{\partial u_\mathrm{i}} = -\varphi(-\beta_1)\frac{u_{1\mathrm{i}}}{\beta_1} \tag{2.18}$$

Since $\varphi(-\beta_1)$ is a constant in Eq. (2.18), $\frac{u_{1\mathrm{i}}}{\beta_1}$ indicates the relative importance of each component. We can therefore use the following indicator to identify unimportant input random variables,

$$c_\mathrm{i} = \frac{|u_{1\mathrm{i}}|}{\beta_1} \tag{2.19}$$

If c_i is less than a threshold c_{thres}, X_i is considered unimportant. The higher is the threshold, the more input random variables will be classified as unimportant ones, and the higher dimensions will be reduced. Using different thresholds, a user can know how many important variables will be included for the subsequent accurate reliability analysis. The user will then be able to determine an appropriate threshold given his or her computational budget. Based on our experience with from the test problems, we recommend that the user could start from $c_{thres} = 3\%$ or 5% when searching for a suitable threshold.

We group the important variables into a vector $\bar{\mathbf{U}}$ and group the unimportant variables into a vector $\underline{\mathbf{U}}$ with the dimensions of $\bar{n}$ and $\underline{n}$, respectively. Then the input variables are partitioned into two parts.

$$\mathbf{U} = (\bar{\mathbf{U}}; \underline{\mathbf{U}}) \tag{2.20}$$

Accordingly, the first-iteration MPP is also partitioned into two parts.

$$\mathbf{u}_1 = (\bar{\mathbf{u}}_1; \underline{\mathbf{u}}_1) \tag{2.21}$$

where $\bar{\mathbf{u}}_1$ and $\underline{\mathbf{u}}_1$ are the important and unimportant elements of $\mathbf{u}_1$, respectively. Therefore, we have

$$\beta_1 = \|\mathbf{u}_1\| = \|\bar{\mathbf{u}}_1; \underline{\mathbf{u}}_1\| = \sqrt{\|\bar{\mathbf{u}}_1\|^2 + \|\underline{\mathbf{u}}_1\|^2} \tag{2.22}$$

We let $\bar{\beta}_1$ and $\underline{\beta}_1$ to be the FORM-reliability index of the important and unimportant portion of $\mathbf{u}_1$, respectively, which are denoted by

$$\bar{\beta}_1 = \|\bar{\mathbf{u}}_1\| \tag{2.23}$$

$$\underline{\beta}_1 = \|\underline{\mathbf{u}}_1\| \tag{2.24}$$

The overall FORM-reliability index is obtained by

$$\beta_1 = \sqrt{\bar{\beta}_1^2 + \underline{\beta}_1^2} \tag{2.25}$$

The final MPP elements of the unimportant variables will be different from $\underline{\mathbf{u}}_1$, but the difference will be insignificant because the contributions of the unimportant variables are relatively small. For this reason, we fix the unimportant variables $\underline{\mathbf{U}}$ at $\underline{\mathbf{u}}_1$, but we will still consider their contributions indicated by their FORM-reliability index $\underline{\beta}_1$ in the final stage of the reliability analysis. Then the limit-state function becomes a function of $\bar{\mathbf{U}}$ with reduced dimension. The new function is given by

$$Y = G(\bar{\mathbf{U}}) = g\left(\bar{\mathbf{U}}; \underline{\mathbf{u}}_1\right) \tag{2.26}$$

For brevity, we denote the limit-state function as $G(\bar{\mathbf{U}})$.

2.3.3 Reliability Analysis in the Reduced Space

We next perform reliability analysis in the reduced dimensional space ($\bar{\mathbf{U}}$). Once the dimension is reduced, the reliability can be solved either by numerical methods (FORM, SORM, SOSPA, etc.) or surrogate methods (Kriging, PCE, Machine Learning, etc.).

In this study, we use SOSPA for demonstration. SOSPA is a second order numerical method and is used to obtain the probability of failure of $G(\bar{\mathbf{U}})$. The first step of SOSPA is to find the MPP of $G(\bar{\mathbf{U}})$ which is $\bar{\mathbf{U}}_G^*$ by Eq. (2.6). The magnitude of $\bar{\mathbf{U}}_G^*$ or the FORM-reliability index is

$$\bar{\beta}_G = \|\bar{\mathbf{u}}_{\mathrm{G}}^*\| \tag{2.27}$$

Once $\bar{\mathbf{U}}_G^*$ is available, we approximate $G(\bar{\mathbf{U}})$ at $\bar{\mathbf{U}}_G^*$ by the second order Taylor expansion using Eq. (2.8) and have

$$G(\bar{\mathbf{U}}) \approx G\left(\bar{\mathbf{u}}_{\mathrm{G}}^*\right) + \nabla G\left(\bar{\mathbf{u}}_{\mathrm{G}}^*\right)^{\mathrm{T}}\left(\bar{\mathbf{U}} - \bar{\mathbf{u}}_{\mathrm{G}}^*\right) + \frac{1}{2}\left(\bar{\mathbf{U}} - \bar{\mathbf{u}}_{\mathrm{G}}^*\right)^{\mathrm{T}} H\left(\bar{\mathbf{u}}_{\mathrm{G}}^*\right)\left(\bar{\mathbf{U}} - \bar{\mathbf{u}}_{\mathrm{G}}^*\right) \tag{2.28}$$

Then the CGF $K_G(t)$ of $G(\bar{\mathbf{U}})$ is derived analytically by Eq. (2.28). The detail derivations can be found in [26]. The Saddlepoint t_S is obtained by solving $K_G'(t) = 0$. The probability of failure of $G(\bar{\mathbf{U}})$ is calculated by Eq. (2.11), whose solution is denoted by $\bar{p}_f$. The reliability index from SOSPA then is given by

$$\bar{\beta}_{G,SPA} = \left|\Phi^{-1}\left(\bar{p}_f\right)\right| \tag{2.29}$$

If all the derivatives are evaluated by the finite difference method, the number of function evaluations with respect to the dimension of $\bar{\mathbf{U}}$ is $k(\bar{n}+1)+\frac{1}{2}\bar{n}(\bar{n}+1)$, where k is the number of iterations of the MPP search.

2.3.4 Final Reliability Analysis

The final step is to integrate the reliability results from Steps 1 and 2 so that the contributions of both important and unimportant variables are accommodated. Next, we derive the equation for the integration. We first look at the case where we do not do any dimension reduction. Let the MPP obtained without any dimension reduction be $\mathbf{u}^*$, it is partitioned into

$$\mathbf{u}^* = (\bar{\mathbf{u}}^*; \underline{\mathbf{u}}^*) \tag{2.30}$$

where $\bar{\mathbf{u}}^*$ and $\underline{\mathbf{u}}^*$ are the important and unimportant elements of the MPP $\mathbf{u}^*$. According to Eqs. (2.23), (2.24), and (2.25), we have $\bar{\beta} = \|\bar{\mathbf{u}}^*\|$, $\underline{\beta} = \|\underline{\mathbf{u}}^*\|$, and therefore

$$\beta = \sqrt{\|\bar{\mathbf{u}}^*\|^2 + \|\underline{\mathbf{u}}^*\|^2} = \sqrt{\bar{\beta}^2 + \underline{\beta}^2} \tag{2.31}$$

We now look at the case with dimension reduction. As discussed in Step 1, we assume the MPP of unimportant variables to be the MPP from the first iteration, namely, $\underline{\mathbf{u}}^* = \underline{\mathbf{u}}_1$. Then we have

$$\underline{\beta} \approx \|\underline{\mathbf{u}}_1\| \tag{2.32}$$

In Step 2, we also perform the MPP search in the reduced space with unimportant variables fixed at $\underline{\mathbf{u}}_1$. This produces the MPP $\bar{\mathbf{u}}^*$ and FORM-reliability index $\bar{\beta}_G = \|\underline{\mathbf{u}}_G^*\|$. Next, we prove that $\bar{\mathbf{u}}_G^* = \bar{\mathbf{u}}^*$, and therefore $\bar{\beta} = \bar{\beta}_G$. Then we can use Eq. (2.31) to integrate the results in Steps 1 and 2.

Because, in the original space, $\mathbf{u}^*$ is found at the limit state $g(T(\mathbf{U})) = 0$, which means

$$g\left(T\left(\mathbf{u}^*\right)\right) = g\left(T\left(\bar{\mathbf{u}}^*; \underline{\mathbf{u}}^*\right)\right) = 0 \tag{2.33}$$

In the reduced space, for the same reason we have

$$G\left(\bar{\mathbf{u}}_{\mathrm{G}}^{*}\right)=0 \tag{2.34}$$

Assume that the MPPs of $g\left(T\left(\bar{\mathbf{U}};\underline{\mathbf{U}}\right)\right)$ and $G(\bar{\mathbf{U}})$ are unique, in other words, $\mathbf{u}^{*}=(\bar{\mathbf{u}}^{*};\underline{\mathbf{u}}^{*})$ and $\mathbf{u}_{G}^{*}$ are unique.

By substituting the MPP $\mathbf{u}^{*}$ into Eqs. (2.15) and (2.17), we have

$$\mathbf{u}^{*}=-\beta\boldsymbol{\alpha}=-\beta\frac{\left(\frac{\partial g(T(\mathbf{u}^{*}))}{\partial u_{\mathrm{i}}^{*}}\right)_{1,2,\ldots,n}}{\left\|\nabla g\left(T\left(\mathbf{u}^{*}\right)\right)\right\|} \tag{2.35}$$

Therefore, the important elements of the MPP can be expressed as

$$\bar{\mathbf{u}}^{*}=-\beta\bar{\boldsymbol{\alpha}}=-\beta\frac{\left(\frac{\partial g}{\partial\bar{U}_{\mathrm{i}}}\right)_{1,\bar{n}}\Big|_{\mathbf{u}^{*}}}{\left\|\nabla g\left(T\left(\mathbf{u}^{*}\right)\right)\right\|}=-\,\beta'\left(\frac{\partial g}{\partial\bar{U}_{\mathrm{i}}}\right)_{1,\bar{n}}\Bigg|_{\mathbf{u}^{*}} \tag{2.36}$$

$$\beta'=\frac{\beta}{\left\|\nabla g\left(T\left(\mathbf{u}^{*}\right)\right)\right\|} \tag{2.37}$$

Now we relate $\left(\frac{\partial g}{\partial\bar{U}_{\mathrm{i}}}\right)_{1,\bar{n}}\Big|_{\mathbf{u}^{*}}$ with the reduced space.

$$\left(\frac{\partial g}{\partial\bar{U}_{\mathrm{i}}}\right)_{1,\bar{n}}\Bigg|_{\mathbf{u}^{*}}=\left(\frac{\partial g\left(T\left(\bar{\mathbf{U}};\underline{\mathbf{u}}^{*}\right)\right)}{\partial\bar{U}_{\mathrm{i}}}\right)_{1,\bar{n}}\Bigg|_{\bar{\mathbf{u}}^{*}}=\left(\frac{\partial G}{\partial\bar{U}_{\mathrm{i}}}\right)_{1,\bar{n}}\Bigg|_{\bar{\mathbf{u}}^{*}}=\nabla G\left(\bar{\mathbf{u}}^{*}\right) \tag{2.38}$$

where $\nabla G\left(\bar{\mathbf{u}}^{*}\right)$ is the gradient of $G(\bar{\mathbf{U}})$ at $\bar{\mathbf{u}}^{*}$.

Then $\bar{\mathbf{u}}^{*}$ is rewritten as

$$\bar{\mathbf{u}}^{*}=-\beta'\nabla G\left(\bar{\mathbf{u}}^{*}\right) \tag{2.39}$$

which indicates that $\bar{\mathbf{u}}^{*}$ is perpendicular to $G(\bar{\mathbf{U}})=0$. Since $g(\mathbf{u}^{*})=g(\bar{\mathbf{u}}^{*};\underline{\mathbf{u}}^{*})$, we have $G(\bar{\mathbf{u}}^{*})=0$, which means that $\bar{\mathbf{u}}^{*}$ is on the surface of $G(\bar{\mathbf{U}})=0$ and is in the opposite direction of the gradient $\nabla G\left(\bar{\mathbf{u}}^{*}\right)$. Therefore, $\bar{\mathbf{u}}^{*}$ is the shortest distance point from the original to the limit state surface $G(\bar{\mathbf{u}}^*) = 0$ in the space of $\bar{\mathbf{U}}$ and is the MPP of $G(\bar{\mathbf{U}})$, namely

$$\bar{\mathbf{u}}^* = \bar{\mathbf{u}}_{\mathrm{G}}^* \tag{2.40}$$

Since $\bar{\beta} = \|\bar{\mathbf{u}}^*\|$ and $\bar{\beta}_G = \|\bar{\mathbf{u}}_G^*\|$, we have

$$\bar{\beta}_G = \bar{\beta} \tag{2.41}$$

Then Eq. (2.31) can be rewritten as

$$\beta = \sqrt{\bar{\beta}_G^2 + \underline{\beta}^2} \tag{2.42}$$

Because $\underline{\mathbf{u}}_1 \le c$, $\underline{\beta} = \|\underline{\mathbf{u}}_1\|$ is far less than $\bar{\beta}_G$, namely, $\underline{\beta} \ll \bar{\beta}_G$, which means that $\bar{\beta}_G$ dominates the accuracy of β. We now replace the FORM-reliability index $\bar{\beta}_G$ with the more accurate reliability index $\bar{\beta}_{G,SPA}$ in Eq. (2.29), and then we obtain the final reliability index

$$\beta_{\text{overall}} = \sqrt{\bar{\beta}_{G,SPA}^2 + \underline{\beta}^2} \tag{2.43}$$

Then the final probability of failure is obtained by

$$p_{f,\text{ overall}} = \Phi\left(-\beta_{\text{overall}}\right) \tag{2.44}$$

2.3.5 Numerical Procedure

The numerical procedure of the proposed high dimensional reliability analysis method is summarized below.

1) Dimension reduction: Perform one-iteration FORM to obtain one-step MPP $\mathbf{u}_1$; identify the important and unimportant random variables by $u_{1\mathrm{i}} \le c$ and partition input variables the corresponding MPP as $\mathbf{U} = (\bar{\mathbf{U}}; \underline{\mathbf{U}})$ and $\mathbf{u}_1 = (\bar{\mathbf{u}}_1; \underline{\mathbf{u}}_1)$; then calculate FORM-reliability index $\underline{\beta} = \|\underline{\mathbf{u}}_1\|$; by fixing the unimportant variables $\underline{\mathbf{U}}$ at $\underline{\mathbf{u}}_1$, a new limit-state function $G(\bar{\mathbf{U}}) = g\left(\bar{\mathbf{U}}; \underline{\mathbf{u}}_1\right)$ is obtained with reduced dimension.

2) Reliability analysis in $\underline{\mathbf{U}}$ space: Use an accurate reliability method such as SOSPA to find the probability of failure $\bar{p}_f$ based on $G(\bar{\mathbf{U}})$ and calculate the corresponding reliability index, which is $\bar{\beta}_{G,SPA}$ if SOSPA is used.

3) Final reliability analysis: Calculate the final reliability index by $\beta_{\text{overall}} = \sqrt{\bar{\beta}_{G,SPA}^2 + \underline{\beta}^2}$ and the final probability of failure by $p_{f,\text{ overall}} = \Phi\left(-\beta_{\text{overall}}\right)$.

2.4 Examples

In this section, we use three examples to demonstrate the proposed method. Example 1 is a mathematical problem with all the input variables normally distributed. It is presented step by step to show all the details of the proposed method so that an interested reader can easily repeat the process and reproduce the result. Example 2 involves a cantilever beam with over 200 random variables, some of which follow non-normal distributions. Example 3 shows a truss system with 52 bars and 110 random variables, some of which follow extreme value distributions, and the limit-state function is a black-box function. For all the examples, we use the same threshold value $c_{thres} = 3\%$ to divide the input variables into important and unimportant variables.

For comparison, we use MCS, FORM, SOSPA, HDMR-SOSPA (specifically univariate dimension reduction), and DR-SOSPA for all examples. MCS, FORM, and SOSPA are performed without dimension reduction. For HDMR-SOSPA, we first decompose the original limit-state function into n univariate functions and then create surrogate models for all univariate functions with three and five points; after this the reliability is calculated by SOSPA based on the surrogate models. The two HDMR methods denoted by HDMR-3-SOSPA and HDMR-5-SOSPA. DR-SOSPA is the proposed method that employs SOSPA in the reduced dimensional space and accounts for the effects of eliminated variables. To evaluate the advantage of accounting for the effects of eliminated variables, we also compare DR-SOSPA with the method that employs SOSPA in the reduced dimensional space, but the eliminated variables are fixed at their means. We denoted the latter method DR-SOSPA-M.

The result of MCS is served as a reference for accuracy comparison, and the relative error of a non-MCS method with respect to MCS is defined by

$$\varepsilon = \left| \frac{p_f - p_{f,\mathrm{MCS}}}{p_{f,\mathrm{MCS}}} \right| \times 100\% \tag{2.45}$$

where p_f and $p_{f,MCS}$ are the probabilities of failure obtained by non-MCS and MCS, respectively. The number of function calls (FC) and the coefficient of efficiency (CoE) are used to measure the efficiency. The latter is defined by

$$\mathrm{CoE} = \frac{\text{The number of function calls}}{\text{The dimension of original limit state function}} \tag{2.46}$$

2.4.1 A Mathematical Problem

The mathematical problem is a parabolic function given by

$$g(\mathbf{U}) = 20 - 3\sum_{\mathrm{i}=1}^{5} U_{\mathrm{i}}\left(1 + 0.1U_{\mathrm{i}}\right) - \sum_{\mathrm{i}=6}^{100} k_{\mathrm{i}} U_{\mathrm{i}} \tag{2.47}$$

where $U_{\mathrm{i}}, \mathrm{i} = 1, 2, \ldots, 100$ are all independent standard normal random variables, namely $U_{\mathrm{i}} \sim N\left(0, 1^2\right)$, k_{i} is the coefficient of a linear term, $k_{\mathrm{i}} = 0.08$ for $\mathrm{i} = 6, 7, \ldots, 100$.

Following the procedure in Section 2.3.5, we first perform one-iteration FORM to obtain the one-iteration MPP $\mathbf{u}_1$. By setting the threshold $c_{thres} = 3\%$ and using $\frac{|u_{1\mathrm{i}}|}{\beta} > C_{thres}$ to identify important variables, we find that five variables are important that are $\bar{\mathbf{U}} = \left(U_1, U_2, U_3, U_4, U_5\right)^{\mathrm{T}}$. The unimportant variables are $\underline{\mathbf{U}} = \left(U_6, U_7, \ldots, U_{100}\right)^{\mathrm{T}}$. Then $\mathbf{u}_1$ is partitioned into $(\bar{\mathbf{u}}_1; \underline{\mathbf{u}}_1)$, accordingly. The reliability index of unimportant variables is given by $\underline{\beta} = \|\underline{\mathbf{u}}_1\| = 0.3419$. It represents the contribution of the unimportant variables to the reliability. Then, we fix $\underline{\mathbf{U}}$ at $\underline{\mathbf{u}}_1$ and have

$$g(\mathbf{U}) \approx G(\bar{\mathbf{U}}) = 20 - 3\sum_{\mathrm{i}=1}^{5} U_{\mathrm{i}}\left(1 + 0.1U_{\mathrm{i}}\right) - \sum_{\mathrm{j}=6}^{100} k_{\mathrm{j}} \underline{u}_{1\mathrm{j}} \tag{2.48}$$

Thus, the dimension is reduced to 5 from 100.

Next, we conduct reliability analysis in the $\bar{\mathbf{U}}$ space. We first perform the MPP search for $G(\bar{\mathbf{U}})$, which results in the MPP $\bar{\mathbf{u}}_G^* = (1.1770, 1.1770, 1.1770, 1.1770, 1.1770)^{\mathrm{T}}$. We then calculate the Hessian matrix of $G(\bar{\mathbf{U}})$ at $\bar{\mathbf{u}}_G^*$. Using SOSPA, we have the probability of failure that is $\bar{p}_f = 6.7352 \times 10^{-3}$. Then the reliability index of the important variables is obtained that is $\bar{\beta}_{G,SPA} = 2.4711$. The total reliability index, which accommodates both important and unimportant variables, is calculated by $\beta_{\text{overall}} = \sqrt{\bar{\beta}_{G,SPA}^2 + \underline{\beta}^2} = 2.4946$. The final probability of failure is given by $p_{f,\text{ overall}} = \Phi\left(-\beta_{\text{overall}}\right) = 6.3044 \times 10^{-3}$. The results of all the methods are summarized in Table 2.1.

Table 2.1. Results of different methods for Example 1

Methods	p_f	Error (%)	FC	CoE
MCS	6.3416×10^{-3}	-	1e^7	1e^5
FORM	3.9966×10^{-3}	36.98	404	4.04
SOSPA	6.3515×10^{-3}	0.16	5,555	55.55
DR-SOSPA-M	6.1501×10^{-3}	3.02	146	1.46
HDMR-3-SOSPA	1.792×10^{-3}	71.7	201	2.01
HDMR-5-SOSPA	1	-	401	4.01
DR-SOSPA	6.3044×10^{-3}	0.59	146	1.46

As shown in Table 2.1, SOSPA, DR-SOSPA, and DR-SOSPA-M accurately predict the probability of failure. Compared with the results of SOSPA with 5,555 function calls and an error of 0.16%, the proposed method needs 146 function calls and CoE = 1.46, only increasing the error to 0.59%. Although DR-SOSPA-M maintains the same efficiency as the proposed method, the accuracy of DR-SOSPA-M is worse than DR-SOSPA because it ignores the joint influence of the unimportant variables. FORM does not produce an accurate result. The two HDMR methods cannot produce accurate results for this example either. To find the cause of inaccuracy, we perform MCS directly using the surrogate models from HDMR instead of SOSPA and obtain almost the same results as those of HDMR-SOSPA. This indicates that the surrogate models from HDMR are not accurate. The Hessian matrixes of the surrogate models are significantly different from those of the original limit-state function.

2.4.2 A Cantilever Beam

A cantilever is shown in Figure 2.2. It is subjected to 106 random forces on the top surface, in which six of them ($F_{\mathrm{i}}, \mathrm{i} = 1, 2, \ldots, 6$) are lognormally distributed and the rest ($F_{\mathrm{i}}, \mathrm{i} = 7, 8, \ldots, 106$) follow normal distributions. The locations of the forces are random variables that are normally distributed, which are denoted by $l_{F_{\mathrm{i}}}, \mathrm{i} = 1, 2, \ldots, 106$. The width w, height h, and yield strength S_y are normally distributed. All the random variables are independent. The distributions are shown in Table 2.2.

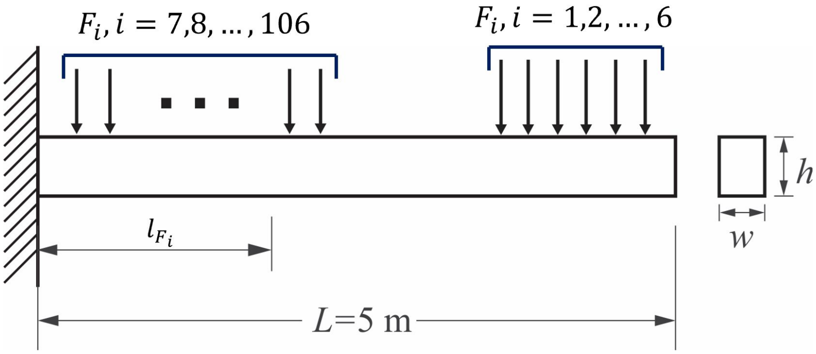


Figure 2.2. A cantilever beam

Table 2.2. Distributions of random variables in Example 2

Random variables	Distribution	Mean	Standard deviation
S_y (MPa)	Normal	720	60
w (m)	Normal	0.2	0.001
h (m)	Normal	0.4	0.001
$F_{\mathrm{i}}, \mathrm{i} = 1, 2, \ldots, 6$ (kN)	Lognormal	$30 + 5\mathrm{i}$	$2.4 + 0.4\mathrm{i}$
$l_{F_{\mathrm{i}}}, \mathrm{i} = 1, 2, \ldots, 6$ (m)	Normal	$4.3 + 0.1\mathrm{i}$	0.01
$F_{\mathrm{i}}, \mathrm{i} = 7, 8, \ldots, 106$ (kN)	Normal	10	1
$l_{F_{\mathrm{i}}}, \mathrm{i} = 7, 8, \ldots, 106$ (m)	Normal	$0.02\mathrm{i}$	0.01

The serviceability state depends on the stress in the beam. The maximal stress should not exceed the yield strength, and then the limit-state function is given by

$$g(\mathbf{X}) = S_y - \frac{6 \sum_{\mathrm{i}=1}^{106} F_{\mathrm{i}} l_{F_{\mathrm{i}}}}{wh^2} \tag{2.49}$$

We first perform the one-iteration FORM to obtain the first-step MPP $\mathbf{u}_1$. Using $c_{thres} = 3\%$, we obtain nine important variables $\bar{\mathbf{U}} = (S_y, w, h, F_1, F_2, \ldots, F_6)^{\mathrm{T}}$ and the reliability index of unimportant variables $\underline{\beta} = 0.1666$. By performing reliability analysis in $\bar{\mathbf{U}}$ space using SOSPA, we have $\bar{p}_f = 1.9481 \times 10^{-6}$ and the corresponding reliability index is $\bar{\beta}_{G,SPA} = 4.6168$. The total reliability index, which accommodates both important and unimportant variables, is calculated by $\beta_{\text{overall}} = \sqrt{\bar{\beta}^2_{G,SPA} + \underline{\beta}^2} = 4.6199$. The probability of failure for the original limit state function is given by $p_{f,\text{ overall}} = \Phi\left(-\beta_{\text{overall}}\right) = 1.9201 \times 10^{-6}$. The results are summarized in Table 2.3.

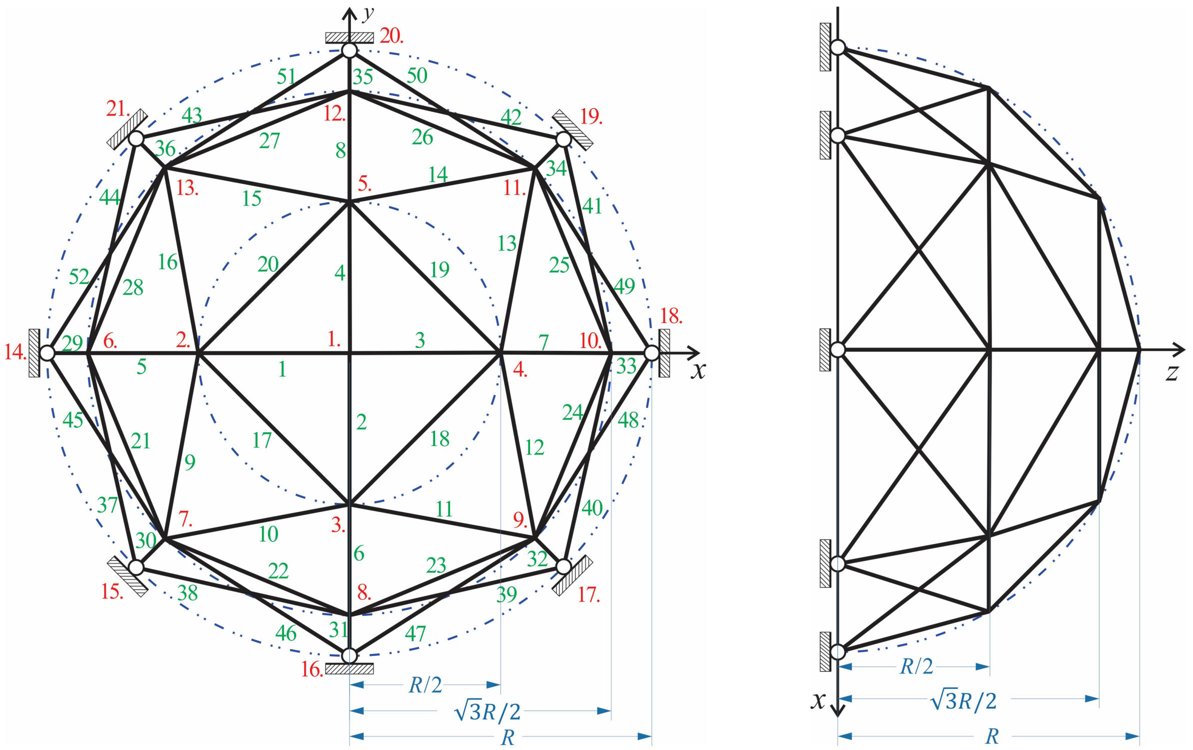
Table 2.3. Results of different methods for Example 2

Methods	p_f	Error (%)	FC	CoE
MCS	1.9106×10^{-6}	-	1.6×10^9	7.4×10^6
FORM	1.7964×10^{-6}	6.0	648	3.0
SOSPA	1.9200×10^{-6}	0.5	24,084	112.0
DR-SOSPA-M	1.8926×10^{-6}	1.0	301	1.4
HDMR-3-SOSPA	1.8158×10^{-6}	5.0	431	2.0
HDMR-5-SOSPA	3.4526×10^{-6}	80.7	861	4.0
DR-SOSPA	1.9201×10^{-6}	0.5	301	1.4

As the results indicate, FORM is the least accurate although it is efficient. SOSPA has an error of 0.5%, but its efficiency is the worst with 24,084 function calls and CoE=112. DR-SOSPA outperforms other methods with the same accuracy (0.5%) as SOSPA and the highest efficiency (FC=301 and CoE=1.4).

2.4.3 A Truss System

This example is modified from [90]. The dome truss system consists of 52 bars with 21 nodes, as shown in Figure 2.3. The truss structure is similar to the roof of a stadium. To distinguish the difference between nodes and bars, the numbers with a dot mean nodes and the numbers without dot denote bars. All the nodes lie on the imaginary hemisphere with a radius of 240 in. The young's moduli and the cross-sectional areas of bars follow normal distributions. The structure is subjected to six random forces at nodes $1-13$, where F_1 is applied to node 1, F_2 is applied to nodes 2 and 4, F_3 is applied to nodes 3 and 5, F_4 is applied to nodes 6 and 10, F_5 is applied to nodes 8 and 12, and F_6 is applied to nodes 7, 9, 11, and 13. The directions of all the forces point to the center of the imaginary hemisphere. All the random variables are independent and their distributions are shown in Table 2.4.



(a) Top view

(b) Side view

Figure 2.3. A 52-bars truss system

Table 2.4. Distributions of random variables in Example 3

Random variables	Distribution	Mean	Standard deviation
$E_{\rm i}, {\rm i} = 1 \sim 50$ (ksi)	Normal	2.5×10^4	1000
$A_{\rm i}, {\rm i} = 1 \sim 8$, and $29 \sim 36$ (in^2)	Normal	2	0.001
$A_{\rm i}, {\rm i} = 9 \sim 16$ (in^2)	Normal	1.2	0.0006
$A_{\rm i}, {\rm i} = 17 \sim 28$, and $37 \sim 52$ (in^2)	Normal	0.6	0.0003
F_1 (kip)	Normal	45	3.6
F_2 (kip)	Extreme	40	6.0
F_3 (kip)	Extreme	35	5.25
F_4 (kip)	Normal	30	4.5
F_5 (kip)	Normal	25	3.75
F_6 (kip)	Normal	20	3

The limit-state function is given in Eq. (2.50) and is solved by the finite element method (FEM).

$$Y = \delta_0 - g(\mathbf{E}; \mathbf{A}; \mathbf{F}) \tag{2.50}$$

where, δ_0 the threshold displacement of node 1. A failure occurs when the displacement of node 1 exceeds $\delta_0 = 0.7$ in. $\mathbf{E} = [E_1, E_2, \ldots, E_{52}]^{\mathrm{T}}$ and $\mathbf{A} = [A_1, A_2, \ldots, A_{52}]^{\mathrm{T}}$ are vectors of the young's moduli and cross-sectional areas, respectively. $\mathbf{F} = [F_1, F_2, \ldots, F_6]^{\mathrm{T}}$ is the vector of the loads.

Following the procedure in Section 2.3.5, we obtain the one-iteration MPP. Nine variables are identified as important variables by setting $c_{thres} = 3\%$, which are $[F_1, \ldots, F_5, E_1, \ldots, E_4]^{\mathrm{T}}$. Then, the probability of failure is obtained by integrating the influence of important and unimportant variables. The results are summarized in Table 2.5. FORM produces a large error. SOSPA produces the most accurate result, but its efficiency is poor as it needs 6,660 function calls with CoE=60.54. The error of DR-SOSPA is 2.29%, which is smaller than the error of DR-SOSPA-M and is larger than SOSPA, and its computational burden is relieved significantly with only 206 function calls and CoE=1.87. The proposed method DR-SOSPA is better than HDMR-SOSPA both in accuracy and efficiency.

Table 2.5. Results of different methods for Example 3

Methods	p_f	Error (%)	FC	CoE
MCS	5.10×10^{-3}	-	10^7	9.09×10^4
FORM	5.7678×10^{-3}	13.09	444	4.03
SOSPA	5.0481×10^{-3}	1.02	6,660	60.54
DR-SOSPA-M	4.8532×10^{-3}	4.84	179	1.63
HDMR-3-SOSPA	4.3053×10^{-3}	15.6	221	2.01
HDMR-5-SOSPA	4.6776×10^{-3}	8.3	441	4.01
DR-SOSPA	4.9833×10^{-3}	2.29	206	1.87

We also modify this example to examine a case with a large probability of failure by reducing the threshold value δ_0 in Eq. (2.50) to 0.5 in. The threshold is still 3% and nine variables are important. The results show that the proposed method is effective for a large probability of failure problems as well.

Table 2.6. Results of large probability of failure for Example 3

Methods	p_f	Error (%)	FC	CoE
MCS	0.2781	-	10^5	909
FORM	0.2978	7.10	333	3.03
SOSPA	0.2763	0.65	6,549	59.54
DR-SOSPA-M	0.2756	0.90	196	1.78
HDMR-3-SOSPA	0.2669	4.02	221	2.01
HDMR-5-SOSPA	0.4730	70.1	441	4.01
DR-SOSPA	0.2758	0.84	196	1.78

The main computer code of the truss example can be found in Supplementary Material A. Interested readers can test the proposed method or other methods based on the code using the truss example.

2.5 Summary

The proposed method partitions the input random variables into two parts, important and unimportant variables, which is achieved by using the information from the first iteration of FORM. With the unimportant random variables fixed at their percentile values obtained from one-iteration FORM, the dimension is reduced to the dimension of important input random variables. Then the probability of failure is found by an accurate reliability method in the reduced space. The final probability of failure is obtained by integrating the probability of failure in the reduced space and the contributions of unimportant variables. Hence, the dimension is reduced, and the contributions of all input variables are also accommodated, resulting in high accuracy and efficiency of high-dimensional reliability analysis.

The proposed method works better if fewer important input variables are important. It cannot effectively reduce the dimension, however, when all input variables are important. If dimension is not reduced, the proposed dimension reduction strategy will not affect the performance of the method used in the second step (the high accurate reliability method in the reduced space in Section 2.3.5. In this case, one may use other dimension reduction methods that can reduce the dimension of the linear combinations of the original input variables. Another limitation is that the proposed method may not be accurate for highly nonlinear problems since the one-iteration MPP may not be accurate to identify the real importance of random variables. More iterations of the MPP search may be helpful in finding the real importance of the variables, but the efficiency will deteriorate.

Our future work will improve the proposed method when most of the input variables are important. We will also study the possibility of applying the proposed method to reliability-based design optimization.

3. ACTIVE LEARNING WITH GENERALIZED SLICED INVERSE REGRESSION FOR HIGH-DIMENSIONAL RELIABILITY ANALYSIS

Jianhua Yin[1,2], Xiaoping Du[1]

[1]Department of Mechanical and Energy Engineering, Indiana University - Purdue University Indianapolis, 723 W. Michigan Street, Indianapolis, IN 46202-5195

[2]School of Mechanical Engineering, Purdue University, West Lafayette, IN, 47907, United States

Published in *Structural Safety*, doi: https://doi.org/10.1016/j.strusafe.2021.102151

Author Contributions

The authors confirm contribution to the paper as follows: Jianhua Yin and Xiaoping Du designed the study and contributed to the writing of the manuscript. Jianhua Yin developed detailed methodology, testing problems, and the code.

3.1 Background

Reliability is measured by the probability that a system performs its intended function without failure. Reliability analysis is a core task in engineering design, where the probability of failure is predicted for a given design. If the probability of failure exceeds the design requirement, the design is updated, and the reliability analysis is performed again. This process repeats until the reliability target is achieved. The probability of failure can be predicted by physical models derived from physical principles or data-driven models. It is given by

$$p_f = \Pr\{Y = g(\mathbf{X}) < 0\} \tag{3.1}$$

where $\mathbf{X} = (X_1, X_2, \ldots, X_p)^{\mathrm{T}}$ is a vector of input random variables, $g(\mathbf{X})$ is a performance function that could be a physical model derived from physical principles or a regression model based on data, and Y is a response that indicates the state of the product. Conventionally, when $Y < 0$, a failure occurs. In this study, we assume the input random variables $\mathbf{X}$ are independent. If they are not independent, they could be transformed into independent ones [45].

There are three types of reliability analysis methods: 1) approximation methods [21], [22], [91], 2) meta-modeling methods [28]–[30], [56], [92]–[95], and 3) sampling methods [60], [63], [96]–[98]. Commonly used approximation methods include the first order reliability method (FORM) [21] and the second order reliability method (SORM) [22]. They approximate the performance function by making use of Taylor expansion. Meta-modeling methods construct a surrogate model to replace the performance function using regression or interpolation methods. Design of Experiments (DoE) [80] is a commonly used tool to generate optimal training points to build the surrogate model. The efficiency of mate-modeling based reliability analysis methods can be improved by active learning [99]. Sampling methods, such as Monte Carlo Simulation (MCS) [100], importance sampling (IS) [101], and subset simulation (SS) [19], are not affected by the dimensionality. However, their computational effort is still very high regardless of the dimension, especially when the probability of failure is low. Although meta-modeling approaches may be more efficient, a dimension reduction is still needed to handle high-dimension problems.

A commonly used dimension reduction approach is the principal component analysis (PCA) [84], [85], [102]. PCA reduces the dimension of random variables by exploiting their correlation structure. If the random variables are strongly correlated, PCA can effectively reduce the dimension by linear combinations of the random variables, resulting in the so-called principal components. It does not work well for independent random variables. PCA is an unsupervised method that does not use the information of the response Y. High-dimensional model representation (HDMR) [70]–[72] is another high-dimensional reliability method, which decomposes $g(\mathbf{X})$ into the sum of several low- dimensional functions. However, when the interaction terms dominate the performance function, the accuracy is poor.

Machine learning and regression methods have recently been used in high dimensional reliability analysis. Several studies [35], [77]–[79], [81], [103] combine meta-modeling and dimension reduction techniques. Two steps are typically involved. A low dimensional latent subspace is identified by the sliced inverse regression (SIR) [48], which is a linear sufficient dimension reduction (SDR) technique, or other dimension reduction methods [78], [103], [104] using training points generated by DoE. A surrogate model of the performance function is then constructed in the low dimensional latent subspace and is refined by cross validation. Since the training points are pre-defined by DoE in the first step, there is no guarantee that 1) a suitable latent subspace exists, and 2) the accuracy of the surrogate model is satisfactory. An active learning based meta-modeling approach combined with dimension reduction is reported in [79]. It combines AK-MCS [30] with a dimension reduction technique called active subspace (AS) [104] to iteratively select the optimum training points in the original high dimensional space, and good accuracy and efficiency are achieved. SIR and AS, however, are both linear dimension reduction techniques, and they may not work well for problems that need nonlinear dimension reduction.

It is desirable to use nonlinear dimension reduction approaches for high dimensional reliability analysis. Nonlinear dimension reduction techniques can be classified into two groups, supervised nonlinear dimension reduction [105], [106] and unsupervised nonlinear dimension reduction [107], [108]. Similar to PCA, unsupervised nonlinear dimension reduction, such as Kernel PCA [107], autoencoder [35], and diffusion maps [109], do not make use of the information of the model response or labels in the dimension reduction process. For the supervised dimension reduction methods, studies in [50], [110], [111] combine the so-called kernel trick [112], [113] with SDR to overcome the limitation of linear SDR, making supervised nonlinear sufficient dimension reduction feasible. The approaches include the kernel canonical correlation analysis (KCCA) [111], kernel SIR (KSIR) [110], and generalized SIR (GSIR) [50]. GSIR not only relaxes the stringent conditions required by linear SDR where the reduced subspace is the linear combination of the original random variables, but also relieves the assumption of KSIR that the subspace is the linear combination of a set of nonlinear functions. Given the advantages of GSIR, it is worth investigating its use in high dimensional reliability analysis.

This work develops a high dimensional reliability method that combines GSIR with GP, IS, and active learning. The proposed method inherits the advantage of GSIR, which is more general and robust no matter if the linear or nonlinear dimension reduction is required. The computational cost of constructing the surrogate model used for reliability analysis is decreased drastically due to the dimension reduction by GSIR. Since the use of IS requires less computational effort than MCS based methods, the proposed method can also handle small probabilities of failure.

The rest of the paper is organized as follows. Section 4.2.2 reviews the related methodologies used in this paper. Section 3.3 presents the details of the proposed method followed by four examples in Section 3.4. Concluding remarks are provided in Section 3.5.

3.2 Literature Review

3.2.1 Generalized Sliced Inverse Regression (GSIR)

GSIR is an approach belonging to sufficient dimension reduction (SDR). Given input variables $\mathbf{X} \in \mathbb{R}^{n \times p}$ and the response $Y \in \mathbb{R}^{n \times 1}$ that depends on $\mathbf{X}$, SDR seeks a function $R(\mathbf{X})$ to map $\mathbf{X}$ to a subspace. The sufficiency is achieved when the distribution of Y given $\mathbf{X}$ is the same as that given $R(\mathbf{X})$, where n is the number of training points, and p is the dimension of $\mathbf{X}$. For linear SDR, $R(\mathbf{X})$ contains one or more linear combinations of $\mathbf{X}$, and the task is to find a matrix $\beta \in \mathbb{R}^{p \times d}$ such that

$$Y \perp\!\!\!\perp \mathbf{X} \mid \beta^{\mathrm{T}}\mathbf{X} \tag{3.2}$$

where d is the dimension of the subspace, and $d < p$; $\perp\!\!\!\perp$ denotes independence, meaning that the distribution of Y is conditionally independent of $\mathbf{X}$ given $\beta^{\mathrm{T}}\mathbf{X}$. Different from SDR, nonlinear SDR searches for a set of nonlinear functions $f_1(\mathbf{X}), \ldots, f_d(\mathbf{X})$ such that

$$Y \perp\!\!\!\perp \mathbf{X} \mid f_1(\mathbf{X}), \ldots, f_d(\mathbf{X}) \tag{3.3}$$

Since $d < p$, the dimension is reduced from p to d.

The nonlinear functions may be hard to define in practice. But the use of the kernel trick could allow dimension reduction to proceed without defining the nonlinear function. This is done by projecting $\mathbf{X}$ and Y to the kernel space.

GSIR [50] is a nonlinear dimension reduction method that stems from the nonlinear SDR theory. The conditional expectation of $\mathbf{X}$ given Y is denoted by

$$E_{\mathbf{X}|Y} = \Sigma_{YY}^{-1/2} R_{Y\mathbf{X}} \Sigma_{\mathbf{XX}}^{1/2} \tag{3.4}$$

where $R_{Y\mathbf{X}}$ is called the correlation operator denoted by

$$R_{Y\mathbf{X}} = \Sigma_{YY}^{-1/2} \Sigma_{Y\mathbf{X}} \Sigma_{\mathbf{XX}}^{-1/2} \tag{3.5}$$

and Σ is the covariance operator.

If a data set of training points are available with $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ and $(y_1, \ldots, y_n)$, then

$$\Sigma_{\mathbf{XX}} = \Sigma_{Y\mathbf{X}} = \frac{1}{n} G_{\mathbf{X}} = \frac{1}{n} Q K_{\mathbf{X}} Q = \frac{1}{n} Q \begin{bmatrix} K(\mathbf{x}_1, \mathbf{x}_1) & \cdots & K(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ K(\mathbf{x}_n, \mathbf{x}_1) & \cdots & K(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix} Q \tag{3.6}$$

$$\Sigma_{YY} = \frac{1}{n} G_Y = \frac{1}{n} Q K_Y Q = \frac{1}{n} Q \begin{bmatrix} K(y_1, y_1) & \cdots & K(y_1, y_n) \\ \vdots & \ddots & \vdots \\ K(y_n, y_1) & \cdots & K(y_n, y_n) \end{bmatrix} Q \tag{3.7}$$

where $Q = I_n - 1_n 1_n^T / n$; $G_{\mathbf{X}}$ and G_Y are the centered versions of the kernel matrixes $K_{\mathbf{X}}$ and K_Y; I_n is an $n \times n$ identity matrix; and 1_n is an $n \times 1$ vector with all elements being 1. The kernel function used in this paper is the anisotropic squared-exponential function and is defined by

$$K(\mathbf{x}_{\mathrm{i}}, \mathbf{x}_{\mathrm{j}}) = \exp\left(-\theta_{\mathbf{x}} (\mathbf{x}_{\mathrm{i}} - \mathbf{x}_{\mathrm{j}})^2\right), \mathrm{i.j} = 1, \ldots, n \tag{3.8}$$

$K\left(y_\mathrm{i}, y_\mathrm{j}\right)$ is obtained by replacing $\theta_\mathbf{X}\left(\mathbf{x}_\mathrm{i}-\mathbf{x}_\mathrm{j}\right)^2$ with $\theta_y\left(y_\mathrm{i}-y_\mathrm{j}\right)^2$ in Eq. (3.8). $\theta_\mathbf{X}$ is computed by

$$\frac{1}{\theta_\mathrm{X}}=\begin{pmatrix} n \\ 2 \end{pmatrix}^{-1} \sum_{\mathrm{i}<\mathrm{j}} \left|\mathbf{x}_\mathrm{i}-\mathbf{x}_\mathrm{j}\right|^2 \tag{3.9}$$

θ_Y is obtained by replacing $\left|\mathbf{x}_\mathrm{i}-\mathbf{x}_\mathrm{j}\right|$ with $\left|y_\mathrm{i}-y_\mathrm{j}\right|$ in Eq. (3.9). Similar to other kernel-based methods, such as GP and SVM, we can choose a kernel from several well established options [114]. Anisotropic squared-exponential function or squared-exponential function is a good starting point when we are short of knowledge about a problem.

Substituting Σ_{YY}, $\Sigma_{\mathbf{XX}}$, and $\Sigma_{Y\mathbf{X}}$ into Eqs. (3.4) and (3.5) yields the correlation operator and conditional expectation.

$$R_{Y\mathbf{X}}=G_Y^{\dagger 1/2} G_\mathbf{X} G_\mathbf{X}^{\dagger 1/2} \tag{3.10}$$

$$E_{\mathbf{X}|Y}=G_Y^{\dagger} G_\mathbf{X} G_\mathbf{X}^{\dagger 1/2} G_\mathbf{X}^{1/2} \tag{3.11}$$

where † means the Moore–Penrose inverse [115] of a matrix in a general sense. In the numerical computation, the Moore-Penrose inverses $G_\mathbf{X}^\dagger$ and $G_Y^\dagger$ are replaced by the ridge-regression-type regularized inverses $\left(G_\mathbf{X}+\epsilon_\mathbf{X} I_n\right)^{-1}$ and $\left(G_Y+\epsilon_Y I_n\right)^{-1}$, respectively, where $\epsilon_\mathbf{X}$ and ϵ_Y are the penalty terms. The first d eigenvectors $\xi_1, \xi_2, \ldots, \xi_d$ used to form the sufficient predictors are obtained by performing the eigen-analysis of the matrix in Eq. (12).

$$\begin{aligned} G_\mathbf{X}^\dagger \left[E_{\mathbf{X}|Y}\right]^\mathrm{T} G_Y^2 \left[E_{\mathbf{X}|Y}\right] G_\mathbf{X}^\dagger = & \left(G_\mathbf{X}+\epsilon_\mathbf{X} I_n\right)^{-3/2} G_\mathbf{X}^{3/2} \left(G_Y+\epsilon_Y I_n\right)^{-1} \\ & G_Y^2 \left(G_Y+\epsilon_Y I_n\right)^{-1} G_\mathbf{X}^{3/2} \left(G_\mathbf{X}+\epsilon_\mathbf{X} I_n\right)^{-3/2} \end{aligned} \tag{3.12}$$

After the dimension reduction (training) is complete, predictions of new input variables can be made. Given a new set of input variables $\left(\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_m\right)$, denote their responses by $\hat{Y}=\left(\hat{y}_1, \ldots, \hat{y}_m\right)$, and their predictors can be obtained as follow.

The kernel matrix of the training points and new points are obtained by

$$K_{\mathbf{X}\hat{\mathbf{X}}} = \begin{bmatrix} K\left(\mathbf{x}_1, \hat{\mathbf{x}}_1\right) & \cdots & K\left(\mathbf{x}_1, \hat{\mathbf{x}}_m\right) \\ \vdots & \ddots & \vdots \\ K\left(\mathbf{x}_n, \hat{\mathbf{x}}_1\right) & \cdots & K\left(\mathbf{x}_n, \hat{\mathbf{x}}_m\right) \end{bmatrix} \tag{3.13}$$

Then, the sufficient predictor $\hat{f}_{\mathrm{i}}$ is given by

$$\hat{f}_{\mathrm{i}} = \xi_{\mathrm{i}}^T Q K_{\mathrm{X}\widehat{\mathrm{X}}}, \mathrm{i} = 1, \ldots, d \tag{3.14}$$

The corresponding eigenvalues of the eigenvectors $(\xi_1, \xi_2, \ldots, \xi_d)$ are sorted in a descending order, as is the importance of the corresponding eigenvectors. The first sufficient predictor $\hat{f}_1$ is therefore the most important predictor. As indicated in [50], the relationship between $\hat{f}_1$ and the response $\hat{Y}$ is usually monotonic, and Spearman's correlation is used to measure the monotonic relationship. The monotonicity is an advantage of GSIR over many other dimension reduction methods [50] since the monotonic relationship can clearly classify a training point into either the safe region or the failure region for the reliability prediction. The advantage is also demonstrated in this study as will be shown in Section 3.4. It is therefore possible to reduce the original dimension p to 1 because $\hat{f}_1$ is in a one-dimensional space.

The GSIR algorithm is summarized as follows.

Algorithm 1 Generalized sliced inverse regression [50]

```
Collect training points $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ and $(y_1, \ldots, y_n)$.
Select the ridge parameters $\epsilon_{\mathbf{X}}$ and $\epsilon_Y$ and compute $\theta_{\mathbf{X}}$, $\theta_Y$ by Eq. (3.9).
Solve for the first $d$ eigenvectors $\xi_1, \xi_2, \ldots, \xi_d$ of the matrix in Eq. (3.12).
Form the sufficient predictors by Eq. (3.14).
```

3.2.2 Importance Sampling (IS)

Importance sampling (IS) is a sampling method that approximates a mathematical expectation with respect to a target distribution by a weighted average of random draws from another distribution (called an importance distribution). For high reliability problem, if samples are drawn from the joint (target) distribution of the original random variables, the chance of getting samples in the failure region is low. Such a chance will be much higher if the samples are drawn from a suitable importance distribution, thereby increasing the computational efficiency. Therefore, it is desirable to use an importance distribution that is centered in the region where the failure is most likely. In risk analysis literature, the Most Probable Point (MPP) [99] is usually used as the center of the importance distribution. The MPP belongs to the limit state surface, and this point has the highest probability density in the standard normal space (U-space).

To solve for MPP, we first transformed $\mathbf{X}$ to $\mathbf{U}$, whose components are independent standard normal variables [45]. The transformation is denoted by $\mathbf{X} = T(\mathbf{U})$. The performance function then becomes $Y = g(T(\mathbf{U})) = G(\mathbf{U})$. The next step is to obtain the IS center. There is no need to search for the true MPP in practice. We can use the point from the one-iteration FORM as the IS center to reduce the computation time. Although it may not be close to the true MPP, the one-iteration MPP allows the IS samples centered around it to cover a sufficiently large area of failure region if a proper sample size is used. The one-iteration MPP is obtained by

$$\mathbf{u}^* = -\frac{G(\mathbf{u}_0)\nabla G(\mathbf{u}_0)}{\|\nabla G(\mathbf{u}_0)\|^2} \tag{3.15}$$

where $\mathbf{u}_0 = (0, \ldots, 0)^{\mathrm{T}}$ is the origin of the U-space. For the highly nonlinear problems, more iterations of the MPP search may be needed to approach the failure boundary.

With the approximate MPP $\mathbf{u}^*$, we shift the center of the probability density to $\mathbf{u}^*$, resulting in importance probability density $\varphi_{\mathbf{U}}(\cdot)$, represented by the new distribution $\widehat{U}_{\mathrm{i}} \sim N(u_{\mathrm{i}}^*, 1^2)$, where u_{i}^* is the i-th component of $\mathbf{u}^*$. In this paper, we use the same standard deviations of standard normal variables. The probability of failure is estimated with density $\phi_{\mathbf{U}}(\cdot)$ of U and the importance density $\varphi_{\mathbf{U}}(\cdot)$.

$$p_f = \int I_F(\mathbf{u}) \frac{\phi_{\mathbf{U}}(\mathbf{u})}{\varphi_{\mathbf{U}}(\mathbf{u})} \varphi_{\mathbf{U}}(\mathbf{u}) d\mathbf{u} \tag{3.16}$$

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

$$I_F(\mathbf{u}) = \begin{cases} 0, G(\mathbf{U}) > 0 \\ 1, G(\mathbf{U}) \leq 0 \end{cases} \tag{3.17}$$

With the samples drawn from the importance density $\phi_{\mathbf{U}}(\mathbf{u})$, p_f in Eq. (3.16) is estimated by

$$p_f \approx \hat{p}_f = \frac{1}{N_{IS}} \sum_{\mathrm{i}=1}^{N_{IS}} I_F\left(\hat{\mathbf{u}}_{\mathrm{i}}\right) \frac{\phi_{\mathrm{U}}\left(\hat{\mathbf{u}}_{\mathrm{i}}\right)}{\varphi_{\mathrm{U}}\left(\hat{\mathbf{u}}_{\mathrm{i}}\right)} \tag{3.18}$$

where $\hat{\mathbf{u}}_{\mathrm{i}}, \mathrm{i} = 1, \ldots, N_{IS}$, are the samples generated from $\phi_{\mathbf{U}}(\mathbf{u})$. The variance of the probability of failure is estimated by

$$\operatorname{Var}\left(\hat{p}_f\right) = \frac{1}{N_{IS}} \left(\frac{1}{N_{IS}} \sum_{\mathrm{i}=1}^{N_{IS}} \left(I_F\left(\hat{\mathbf{u}}_{\mathrm{i}}\right) \left(\frac{\phi_{\mathbf{U}}\left(\hat{\mathbf{u}}_{\mathrm{i}}\right)}{\varphi_{\mathbf{U}}\left(\hat{\mathbf{u}}_{\mathrm{i}}\right)} \right)^2 \right) - \hat{p}_f^2 \right) \tag{3.19}$$

If a proper importance distribution is chosen, $\operatorname{Var}\left(\hat{p}_f\right)$ is less than the variance of MCS, therefore increasing the computational efficiency.

The coefficient of variation δ_{IS} of $\hat{p}_f$ is calculated by

$$\delta_{IS} = \frac{\sqrt{\operatorname{Var}\left(\hat{p}_f\right)}}{\hat{p}_f} \tag{3.20}$$

3.2.3 Gaussian Process (GP) Modeling

GP modeling [116] views a function $G(\mathbf{U})$ as a realization of a Gaussian process. Given a set of training points, a GP model is obtained by

$$\hat{G}(\mathbf{u}) = \mathbf{f}(\mathbf{u})^{\mathrm{T}} \boldsymbol{\beta} + Z(\mathbf{u}) \tag{3.21}$$

where $\mathbf{f}(\mathbf{u})^{\mathrm{T}} \boldsymbol{\beta}$ is a deterministic term, providing the trend and the mean response; $\mathbf{f}(\mathbf{u}) = (f_1(\mathbf{u}), f_2(\mathbf{u}), \ldots, f_p(\mathbf{u}))^{\mathrm{T}}$ is a vector of regression functions; $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)^{\mathbf{T}}$ is a vector of regression coefficients; $Z(\cdot)$ is a stationary Gaussian process with zero mean and covariance. The covariance is denoted by

$$\mathrm{Cov}\left(Z\left(\mathbf{u}_{\mathrm{i}}\right), Z\left(\mathbf{u}_{\mathrm{j}}\right)\right)=\sigma_{Z}^{2} R\left(\mathbf{u}_{\mathrm{i}}, \mathbf{u}_{\mathrm{j}}\right) \tag{3.22}$$

where σ_Z^2 is the process variance, $R(\cdot,\cdot)$ is the correlation function, specifically the squared-exponential kernel used in this work. GP can also provide the variance of the prediction as the GP predictor $\hat{G}(\mathbf{u})$ follows a normal distribution, denoted by

$$\hat{G}(\mathbf{u}) \sim N\left(\mu_{G}(\mathbf{u}), \sigma_{G}^{2}(\mathbf{u})\right) \tag{3.23}$$

where $\mu_G(\cdot)$ and $\sigma_G^2(\cdot)$ represent the mean GP prediction and GP variance, respectively.

3.2.4 AK-IS

AK-IS [99] is an active learning method combining Kriging (GP modeling) with IS for reliability analysis. AK-IS at first uses the MPP-centered importance distribution to generate samples, called the IS population. It then constructs the GP model by the point used for solving the MPP and refines the model by adding training points selected from the IS population. A new training point is selected by a learning function and is added to the set of training points, which allows for an update of the GP model. The process stops once the desired accuracy is achieved. The size of the IS population will be increased if a target coefficient of variation is unsatisfied. The learning function is defined by

$$\mathbb{U}(\widehat{\mathbf{u}})=\frac{\left|\mu_{G}(\widehat{\mathbf{u}})\right|}{\sigma_{G}^{2}(\widehat{\mathbf{u}})} \tag{3.24}$$

where $\widehat{\mathbf{u}}$ is a point in the IS population, and $\mu_G(\cdot)$ and $\sigma_G^2(\cdot)$ are given in Eq. (3.23). A lower $\mathbb{U}(\widehat{\mathbf{u}})$ means a higher probability that the point is misclassified. Then the point with the minimum $\mathbb{U}(\widehat{\mathbf{u}})$ in the IS population is selected as the new training point. The learning process stops when $\min \mathbb{U}(\widehat{\mathbf{u}}) \geq 2$.

3.3 Methodology

The purpose of this study is to explore the use of GSIR in high dimensional reliability analysis to reduce computational efforts. The central strategy is reducing the dimension of input variables by GSIR so that a GP model can be constructed in a low dimensional subspace. The following three steps are involved and are illustrated in Figure 3.1, where TP stands for training point.

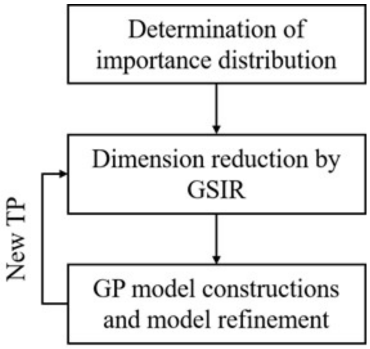

Determination of importance distribution
Dimension reduction by GSIR
GP model constructions and model refinement
New TP

Figure 3.1. Schematic of the proposed method

1) Determine the importance distribution: we first obtain the one-iteration MPP which is the importance distribution center. Samples are generated from the importance distribution as described in Section 3.3.1 to form the importance population.

2) Initialization and dimension reduction by GSIR. Initially, training points $\mathbf{U}$ are generated by Latin Hypercube sampling, centered at the origin of the U-space, and the corresponding responses are obtained by calling the performance function $Y = G(\mathbf{U})$. GSIR then trains sufficient predictors using the set of training points and associated responses. In subsequent iterations, new training points are selected by active learning from the IS population.

3) Surrogate model creation in subspace: the GP model is constructed in the one-dimensional space of the sufficient predictor (Eq. (3.14)). The input of the model is the first sufficient predictor from step 2), and the output is the prediction of the response Y for the GP model.

Since the sufficient predictor and GP model may not be accurate, steps 2 and 3 are performed iteratively to refine the GP model by selecting new training points from the IS population through an active learning strategy. In each iteration, only one new training point is added. The model update completes once the convergence criterion is met. When the algorithm converges, the probability of failure is obtained by the IS estimation method discussed in Section 3.3.1. Next, we provide detailed descriptions of the three major steps.

3.3.1 Importance Distribution

The first step of the proposed method is to generate a sample population that supplies candidate training points during active learning. As discussed in Section , if the sample population covers both safe and failure regions where the probability density is high, the variance of the estimated probability of failure will be reduced, thereby increasing computational efficiency.

We first transform random variables $\mathbf{X}$ in the X-space to $\mathbf{U}$ in the U-space. The performance function then becomes $Y = G(\mathbf{U})$. Then all the derivations will be performed with respect to $\mathbf{U}$. After the transformation, the one-iteration MPP $\mathbf{u}^*$ is obtained by Eq. (3.15) and serves as the IS center [99]. The computational cost for the one-iteration MPP is $n+1$ evaluations of the performance function.

As mentioned in Section , the importance probability density $\varphi_{\mathbf{U}}(\cdot)$ results in new distributions $\widehat{U}_{\mathrm{i}} \sim N(u_{\mathrm{i}}^*, 1^2)$, $\mathrm{i} = 1, \ldots, n$, where u_{i}^* is the i-th component of $\mathbf{u}^*$. We then draw samples $\widehat{\mathbf{U}} = (\widehat{\mathbf{u}}_1, \ldots, \widehat{\mathbf{u}}_{N_{IS}})$ from $\varphi_{\mathbf{U}}(\cdot)$ to establish an IS population denoted by $\mathbb{P}_{IS}$. The IS population can cover both safety and failure regions with balanced samples in both regions. It is recommended that the size of IS population should be sufficiently large (e.g., 10^4), especially for high dimensional problems. If the coefficient of variation in Eq. (3.20) is large, the population size should be increased accordingly. If the one-iteration MPP is far away from the true MPP, we can also increase the importance sampling size or increase the standard deviations of $\mathbf{U}$ to cover the critical failure region. The added training points during the active learning stage are selected from the IS population.

3.3.2 Initialization and Dimension Reduction by GSIR

The initial training points are generated by Latin Hypercube sampling and are centered at the origin of the U-space, which is denoted by $\mathbf{U}_t = (\mathbf{u}_{t1}, \ldots, \mathbf{u}_{tn})\,, \mathbf{u}_{t1} \in \mathbb{R}^{p\times 1}$. Then, the corresponding responses $Y_t = (y_{t1}, \ldots, y_{tn})\,, y_{t1} \in \mathbb{R}^{1\times 1}$ are obtained by calling the performance function at $\mathbf{U}_t$. It is recommended that the sample size of initial training points is three to five times of the input dimension. This number of initial training points can help create an accurate initial model and can therefore reduce the number of new training points in the subsequent iterations. It is also possible to use fewer initial training points, and the number of new training points will be likely increase.

Once the training points $(\mathbf{U}_t, Y_t)$ are available, GSIR is used to reduce the dimension of input variables such that the GP model can be constructed in a low dimensional space. We first obtained the kernel matrices $K_{\mathbf{U}_t}$ and K_{Y_t} by

$$K_{\mathbf{U}_t} = \begin{bmatrix} K(\mathbf{u}_{t1}, \mathbf{u}_{t1}) & \cdots & K(\mathbf{u}_{t1}, \mathbf{u}_{tn}) \\ \vdots & \ddots & \vdots \\ K(\mathbf{u}_{tn}, \mathbf{u}_{t1}) & \cdots & K(\mathbf{u}_{tn}, \mathbf{u}_{tn}) \end{bmatrix} \tag{3.25}$$

$$K_{Y_t} = \begin{bmatrix} K(y_{t1}, y_{t1}) & \cdots & K(y_{t1}, y_{tn}) \\ \vdots & \ddots & \vdots \\ K(y_{tn}, y_{t1}) & \cdots & K(y_{tn}, y_{tn}) \end{bmatrix} \tag{3.26}$$

where $K(\cdot,\cdot)$ is the kernel function defined in Eq. (3.8). The centered kernel matrices of $K_{\mathbf{U}_t}$ and K_{Y_t} are obtained by $G_{\mathbf{U}_t} = QK_{\mathbf{U}_t}Q$ and $G_{Y_t} = QK_{Y_t}Q$. Then the correlation operator and conditional expectation are obtained by

$$R_{Y_t\mathbf{U}_t} = G_{Y_t}^{\dagger 1/2} G_{\mathbf{U}_t} G_{\mathbf{U}_t}^{\dagger 1/2} \tag{3.27}$$

$$E_{\mathbf{U}_t|Y_t} = G_{Y_t}^{\dagger} G_{\mathbf{U}_t} G_{\mathbf{U}_t}^{\dagger 1/2} G_{\mathbf{U}_t}^{1/2} \tag{3.28}$$

The first d eigenvectors $\xi_1, \xi_2, \ldots, \xi_d$, which are used to form sufficient predictors, are calculated by performing eigen-analysis to the following matrix:

$$\begin{aligned} G^{\dagger}_{\mathbf{U}_t} \left[E_{\mathbf{U}_t|Y_t}\right]^T G^2_{Y_t} \left[E_{\mathbf{U}_t|Y_t}\right] G^{\dagger}_{\mathbf{U}_t} = & \left(G_{\mathbf{U}_t} + \epsilon_{\mathbf{U}_t} I_n\right)^{-3/2} G^{3/2}_{\mathbf{U}_t} \left(G_{Y_t} + \epsilon_{Y_t} I_n\right)^{-1} \\ & G^2_{Y_t} \left(G_{Y_t} + \epsilon_{Y_t} I_n\right)^{-1} G^{3/2}_{\mathbf{U}_t} \left(G_{\mathbf{U}_t} + \epsilon_{\mathbf{U}_t} I_n\right)^{-3/2} \end{aligned} \tag{3.29}$$

As mentioned in Section 3.2.1, the first sufficient predictor $\left(\hat{f}_1\right)$ and the real response Y_t are in a good monotonic relationship. We now denote $\left(\hat{f}_1\right)$ by f_{GSIR}. Then the training points $\mathbf{U}_t$ are mapped to a one-dimensional space through the sufficient predictor

$$f_{GSIR}\left(\mathbf{U}_t\right) = \xi_1^T Q K_{\mathbf{U}_t} \tag{3.30}$$

Predictions of the sufficient predictor can be made for new untried points $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_l)$ by

$$f_{GSIR}(\mathbf{U}) = \xi_1^T Q K_{\mathbf{U}_t \mathbf{U}} \tag{3.31}$$

where

$$K_{\mathbf{U}_t \mathbf{U}} = \begin{bmatrix} K\left(\mathbf{u}_{t1}, \mathbf{u}_1\right) & \cdots & K\left(\mathbf{u}_{t1}, \mathbf{u}_l\right) \\ \vdots & \ddots & \vdots \\ K\left(\mathbf{u}_{tn}, \mathbf{u}_1\right) & \cdots & K\left(\mathbf{u}_{tn}, \mathbf{u}_l\right) \end{bmatrix} \tag{3.32}$$

The next step is to construct a functional relationship between the real response Y and the sufficient predictor f_{GSIR}.

3.3.3 GP Modeling in Subspace

As discussed in Section 3.3.2, the sufficient predictor f_{GSIR} does not provide the prediction of the true response Y, but both have a monotonic relationship. The task now is to transform the sufficient predictor f_{GSIR} to the response Y. Despite the monotonicity feature, the relationship may be nonlinear. Many regression techniques can be used for this task, such as GP, support vector machine (SVM), polynomial chaos expansions (PCEs), and Neural Networks (NNs). In this study, we use GP modeling as an example to illustrate the process. The GP method can not only handle nonlinearity well but also provide the uncertainty estimate of the prediction, enabling an active learning process.

Given a set of training points $(\mathbf{U}_t, Y_t)$, the sufficient predictor $f_{GSIR}(\mathbf{U}_t)$, which is a one-dimensional variable, is obtained by Eq. (3.30) as discussed in Section 3.3.2. Then we construct a one-dimensional GP model using the low dimensional training points $(f_{GSIR}(\mathbf{U}_t), Y_t)$ denoted by

$$Y = \widehat{G}\left(f_{GSIR}\right) = \mathbf{f}\left(f_{GSIR}\right)^{\mathrm{T}} \boldsymbol{\beta} + Z\left(f_{GSIR}\right) \tag{3.33}$$

Since f_{GSIR} is obtained from training points $(\mathbf{U}_t, Y_t)$, it is a function of $\mathbf{U}$. Eq. (3.33) can be rewritten as

$$Y = \widehat{G}\left(f_{GSIR}(\mathbf{U})\right) = \mathbf{f}\left(f_{GSIR}(\mathbf{U})\right)^{\mathrm{T}} \boldsymbol{\beta} + Z\left(f_{GSIR}(\mathbf{U})\right) \tag{3.34}$$

For an untried point $\mathbf{u}$, $f_{GSIR}(\mathbf{u})$ is obtained by Eq. (3.32), and the Gaussian predictor $\hat{G}\left(f_{GSIR}(\mathbf{u})\right)$ follows a normal distribution as follow.

$$Y = \hat{G}\left(f_{GSIR}(\mathbf{u})\right) \sim N\left(\mu_G\left(f_{GSIR}(\mathbf{u})\right), \sigma_G^2\left(f_{GSIR}(\mathbf{u})\right)\right) \tag{3.35}$$

where $\mu_G\left(f_{GSIR}(\mathbf{u})\right)$ is the prediction of the mean value of Y at $\mathbf{u}$, and $\sigma_G\left(f_{GSIR}(\mathbf{u})\right)$ measures the uncertainty in the prediction. The accuracy of the prediction will be gradually improved during the learning process discussed in Section 3.3.4.

3.3.4 Active Learning

The dimension reduction and regression discussed above are executed iteratively to improve the accuracy of the GP model. The accuracy depends on the size and location of the training points, which can be hard to be determined beforehand. To have the best balance between accuracy and efficiency, we gradually improve the accuracy of the regression model by an active learning strategy that adds new training points one by one selected from the IS population. The GSIR dimension reduction and the GP model are updated and refined until the convergence criterion is met. Next, we discuss how to select a new training point and how to measure the accuracy. We adopt the $\mathbb{U}$-learning function **2RN8** to select the next training point at each iteration. Since the GP model is created in the space of sufficient predictor $f_{GSIR}(\mathbf{U})$, given the IS population $\widehat{\mathbf{U}} = (\widehat{\mathbf{u}}_1, \ldots, \widehat{\mathbf{u}}_{N_{IS}})$, the sufficient predictor $f_{GSIR}(\widehat{\mathbf{U}}) = (f_{GSIR}(\widehat{\mathbf{u}}_1), \ldots, f_{GSIR}(\widehat{\mathbf{u}}_{N_{IS}}))$ is obtained by Eq. (3.32). Then the learning function is denoted by

$$\mathbb{U}\left(f_{GSIR}\left(\widehat{\mathbf{u}}_{\mathrm{i}}\right)\right) = \frac{\left|\mu_G\left(f_{GSIR}\left(\widehat{\mathbf{u}}_{\mathrm{i}}\right)\right)\right|}{\sigma_G^2\left(f_{GSIR}\left(\widehat{\mathbf{u}}_{\mathrm{i}}\right)\right)} \tag{3.36}$$

As discussed in Section 3.2.4, the value of the learning function indicates the probability of misclassification of the GP model. The smaller is $\mathbb{U}\left(f_{GSIR}\left(\widehat{\mathbf{u}}_{\mathrm{i}}\right)\right)$, the higher chance the point is misclassified. Therefore, the next training point is the point that $\mathbb{U}\left(\mu_G\left(f_{GSIR}\left(\widehat{\mathbf{u}}_{\mathrm{i}}\right)\right)\right)$ is the smallest and is therefore found by

$$f_{GSIR}\left(\mathbf{u}_{new}\right) = \min_{\widehat{\mathbf{u}}_{\mathrm{i}} \in \mathbb{P}_{IS}} \mathbb{U}\left(f_{GSIR}\left(\widehat{\mathbf{u}}_{\mathrm{i}}\right)\right) \tag{3.37}$$

Then, the corresponding response Y_{new} is available by evaluating the performance function $Y_{new} = G(\mathbf{u}_{new})$. $(\mathbf{u}_{new}, Y_{new})$ is then added to the existing training points. The indicator function in the low dimensional space is $I_F\left(f_{GSIR}(\mathbf{u})\right) = 0$ if $\widehat{G}\left(f_{GSIR}(\mathbf{u})\right) > 0$ or 1 if $\widehat{G}\left(f_{GSIR}(\mathbf{u})\right) < 0$. The probability of failure (p_f) is obtained by Eq. (2.18). Here we use the original joint PDF $\phi_{\mathbf{U}}(\cdot)$ of $\mathbf{U}$ and the importance density $\varphi_{\mathbf{U}}(\mathbf{u})$ to estimate the probability of failure instead of using the joint PDF in the subspace after dimension reduction. First, it is difficult or almost impossible to estimate the joint PDF of the variables in the subspace. Second, the sufficiency maintained by the sufficient dimension reduction means that the information in the original space is preserved after dimension reduction. Based on the two reasons, we use the original joint PDF $\phi_{\mathbf{U}}(\cdot)$ of $\mathbf{U}$ and the importance density $\varphi_{\mathbf{U}}(\mathbf{u})$.

The $\mathbb{U}$-learning function is adapted from the lower confidence bounding (*lcb*) function [117]. The value of $\mathbb{U}\left(f_{GSIR}\left(\widehat{\mathbf{u}}_{\mathrm{i}}\right)\right)$ reflects the least confidence level that the indicator function $I_F\left(f_{GSIR}\left(\widehat{\mathbf{u}}_{\mathrm{i}}\right)\right)$ is classified into the correct group (safe or failure). Thus, the stopping criterion is set to be $\min \mathbb{U}\left(f_{GSIR}\left(\widehat{\mathbf{u}}_{\mathrm{i}}\right)\right) \geq 2$, which means that, at the lowest confidence level, the probability of $I_F\left(f_{GSIR}\left(\widehat{\mathbf{u}}_{\mathrm{i}}\right)\right)$ being accurately classified is $\Phi(2) = 97.7\%$, where $\Phi(\cdot)$ is the cumulative density function (CDF) of a standard normal variable. The iterative process terminates until the stopping criterion is satisfied.

Since the probability of failure is calculated in every iteration with the updated GP model, the final probability of failure is obtained from the last iteration. It is recommended that if the coefficient of variation in Eq. (3.20) is high, for example, 5%, the IS population size should be increased.

3.3.5 Numerical Procedure

The numerical procedures are summarized below.

Algorithm 2 GSIR-GP-IS

```
**Initialization**
Determine the approximate MPP and importance distribution..
Generate IS population $\widehat{\mathbf{U}} = (\hat{\mathbf{u}}_{\mathrm{i}})_1^{N_{IS}}$.
Select ridge parameters $\epsilon_{\mathrm{X}}$, $\epsilon_Y$.
Define initial TPs $(\mathbf{U}_t, Y_t)$ by Latin hypercube sampling.
**while** Convergence is false **do**
    Perform dimension reduction GSIR, and obtain the first eigenvector $(\xi_1)$ of the sufficient predictor in Eq. (3.30) and $f_{GSIR}(\mathbf{U}_t) = \xi_1^T Q K_{\mathrm{U}_t}$.
    Construct the GP model $\widehat{G}(f_{GSIR}(\mathbf{U}))$ using the low-dimensional training points $(f_{GSIR}(\mathbf{U}), Y_t)$.
    Obtain the sufficient predictor at $\widehat{\mathbf{U}}$: $f_{GSIR}(\widehat{\mathbf{U}}) = \xi_1^T Q K_{\mathbf{U}_t \widehat{\mathbf{U}}}$.
    Run the GP model at $f_{GSIR}(\widehat{\mathbf{U}})$ to have $\widehat{G}\left(f_{GSIR}(\widehat{\mathbf{U}})\right)$, the probability of failure $p_f$ is obtained by Eq. (3.18).
    **if** $\min \mathbb{U}\left(f_{GSIR}\left(\hat{\mathbf{u}}_{\mathrm{i}}\right)\right) < 2$ **then**
        Find the next training point $(u_{new})$ using Eqs. (3.36) and (3.37); obtain $Y_{new} = G(\mathbf{u}_{new})$.
        Add $(\mathbf{u}_{new}, Y_{new})$ to TPs.
    **else if** $\min \mathbb{U}\left(f_{GSIR}\left(\hat{\mathbf{u}}_{\mathrm{i}}\right)\right) \geq 2$ **then**
        Stop.
**if** $\delta_{IS}$ (Eq. (3.20)) and accuracy of $p_f$ is satisfied **then**
    Stop.
**else**
    Go to **Initialization** and increase $N_{IS}$ (the size of IS population).
**Output:** $p_f$ and associated error.
```

3.4 Examples

In this section, four examples are provided to demonstrate the proposed method. The first example is a mathematical problem followed by three engineering examples. We compare the proposed method with MCS, FORM, the second order saddlepoint approximation (SOSPA), and SIR-GP-IS. SOSPA [115] is a second-order approximation method based on SORM and saddlepoint approximation (SPA). SIR-GP-IS uses the same settings as the proposed method but using the linear dimension reduction method. Since the first example has been studied by SS-SVM [116] and SIR-SPCE [81], we also compare the two methods for the first example. Since constructing a GP model in the high dimensional space is more expansive than in a subspace, we only use GP-IS, the algorithm without dimension reduction, to evaluate the first case of example 1 to show the necessity of dimension reduction for high-dimensional problems.

A weak penalty is applied for the four examples, and the ridge parameters $\epsilon_{\mathbf{X}}$ and ϵ_Y of GSIR are set to be 10^{-5}. Since the proposed method is a sampling-based meta-modeling method, we run the method 20 times to assess its performance. We then report the medians of the results, including the probability of failure, the error, and the number of function calls. The accuracy of different methods is assessed by the error relative to MCS. The relative error is defined by

$$\varepsilon = \left| \frac{p_f - p_{f,\mathrm{MCS}}}{p_{f,\mathrm{MCS}}} \right| \times 100\% \tag{3.38}$$

where p_f is the probability obtained by a non-MCS method. The efficiency is measured by the number of function calls and the coefficient of efficiency (CoE). CoE describes the efficiency with respect to the dimension and is defined by

$$\mathrm{CoE} = \frac{\text{The number of performance function calls}}{\text{The dimension of input random variables}} \tag{3.39}$$

3.4.1 A Mathematical Problem

The mathematical example is given in [118] and is further studied in [81], [119]. The performance function is defined by

$$g(\mathbf{X}) = n + 3\sigma\sqrt{n} - \sum_{\mathrm{i}=1}^{n} x_{\mathrm{i}} \tag{3.40}$$

in which $x_{\mathrm{i}}, \mathrm{i} = 1, \ldots, n$, are independent and lognormally distributed with means and standard deviations being 1 and 0.2, respectively. We study three cases that n is equal to 40, 100, and 250. The corresponding initial DoE sample sizes are 200, 400, and 1000. The IS population sizes N_{IS} are 10^4, 10^4, and 3×10^4 for the three cases. The IS population sizes are large enough to cover the major failure boundaries. If this condition was not satisfied, a larger sample size (e.g.,10^5 or 10^6) would be needed.

The results of dimension reduction are presented in Figures. 3.2, 3.3, and 3.4, in which TPs means training points. The sufficient predictor, which is obtained by GSIR using Eq. (3.32), is in the one-dimensional space. The three cases show that the sufficient predictor and the real response have perfect monotonicity, for which Spearman's correlations are 0.9993, 0.9999, and 1.0 for the three cases. It is found that the added learning points are concentrated on the failure boundary or the limit state, which means that the GSIR-aided dimension reduction method can identify points in the vicinity of the failure boundary and alleviate the curse of dimensionality.

In addition to the proposed method (GSIR-GP-IS), other methods are also performed, including MCS, FORM, and SOSPA, GP-IS, SIR-GP-IS. We only run MCS for one time with sufficient samples. FORM and SOSPA are also run for one time since these approximation methods are not influenced by randomness. The results of SS-SVM and SIR-SPCE are directly from the literatures as mentioned previously. The results are summarized in Table 3.1.

FORM is the most efficient, but least accurate method. When $n = 40$, the proposed method outperforms SOSPA, SS-SVM, and SIR-SPCE with respect to both accuracy and efficiency. Its error and number of function calls are 0.37% and 401.7, respectively. For the

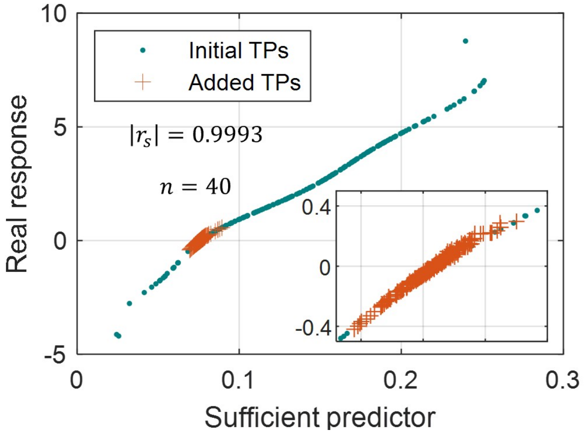


Figure 3.2. Sufficient predictor versus real response of Case 1

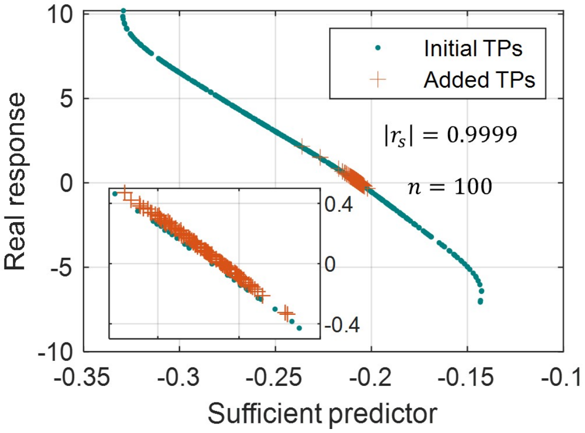


Figure 3.3. Sufficient predictor versus real response of Case 2

case $n = 100$, GSIR-GP-IS obtains an error of 1.03% with 715.4 function calls. Although SS-SVM has a slightly smaller error (0.58%), it calls the performance function 3,729 times. The proposed method performs well for the high dimensional case ($n = 250$). Although SS-SVM and SIR-SPCE are more accurate than the proposed method, their efficiency is much poorer with CoEs of 40.8 and 40, respectively. The proposed method produces an accurate solution (2.25% error) with much fewer function calls (CoE = 6.2). GP-IS, the algorithm without dimension reduction, cannot converge within the prespecified number of iterations (1000) for the case with the lowest dimension ($n = 40$). SIR-GP-IS, the algorithm

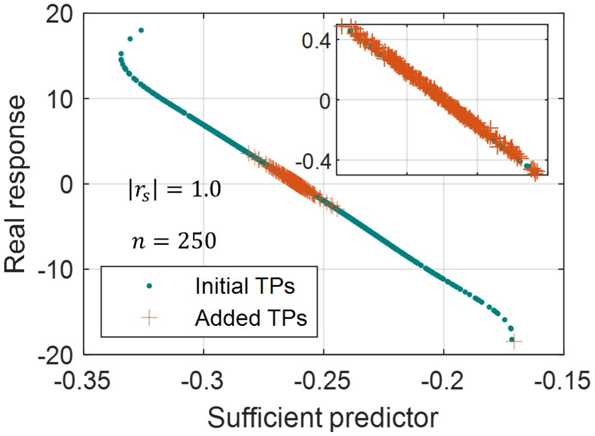


Figure 3.4. Sufficient predictor versus real response of Case 3

Table 3.1. Results of different methods for Example 1

n	Methods	p_f	Error(%)	FCs	CoE
40	MCS	1.97×10^{-3}	-	10^7	2.5×10^5
	FORM	2.152×10^{-4}	89.1	164	4.1
	SOSPA	2.028×10^{-3}	2.96	1,025	25.6
	GP-IS	7.41×10^{-2}	3600	1241	31.0
	SIR-GP-IS	1.14×10^{-3}	42.04	1241	31.0
	SS-SVM	1.95×10^{-3}	1.5	3,729	93.2
	SIR-SPCE	1.88×10^{-3}	3.3	1,200	30
	GSIR-GP-IS	1.98×10^{-3}	0.37	400.65	10.0
100	MCS	1.72×10^{-3}	-	10^7	10^5
	FORM	4.204×10^{-5}	97.55	404	4.0
	SOSPA	1.796×10^{-3}	4.42	5,555	5.6
	SIR-GP-IS	0.1614	9283	559	5.59
	SS-SVM	1.74×10^{-3}	0.58	6036	60.4
	SIR-SPCE	1.63×10^{-3}	5.6	3,000	30
	GSIR- GP-IS	1.74×10^{-3}	1.03	715.4	7.2
250	MCS	1.56×10^{-3}	-	10^7	4×10^4
	FORM	2.82×10^{-6}	99.82	1004	4.0
	SOSPA	1.673×10^{-3}	7.24	32,630	130.5
	SIR-GP-IS	0.1487	9431	1281	5.1
	SS-SVM	1.61×10^{-3}	1.26	10,707	42.8
	SIR-SPCE	1.59×10^{-3}	0.6	10,000	40
	GSIR-GP-IS	1.60×10^{-3}	2.25	1,548.6	6.2

with the linear dimension reduction, cannot converge either within 1000 iterations for the same case. The results reported in Table 3.1 are from the last iteration of the two methods. Although SIR-GP-IS displays a fast convergence rate for the other two cases, the algorithm does not work as the errors are 9283% and 9341%, respectively. Since the surrogate model construction in a high dimensional space is time-consuming, we only run GP-IS for $n = 40$ to demonstrate the necessity of dimension reduction.

To analyze the uncertainty of the result from the proposed method, we also provide box plots in Figure 3.5 for the probabilities of failure and errors from the 20 runs. The medians of the probability of failure and the corresponding errors are $(1.98 \times 10^{-3}, 0.37\%)$, $(1.74 \times 10^{-3}, 1.03\%)$, and $(1.60 \times 10^{-3}, 2.25\%)$ for the three cases, where the errors here are obtained by comparing the median probability of failure with the MCS by Eq. (3.38). The error plot is from the 20 simulations whose median errors are 1.06%, 2.98%, and 2.41%. The standard deviations of errors are 1.77%, 2.80%, and 3.01%. The highest error is smaller than 6%.

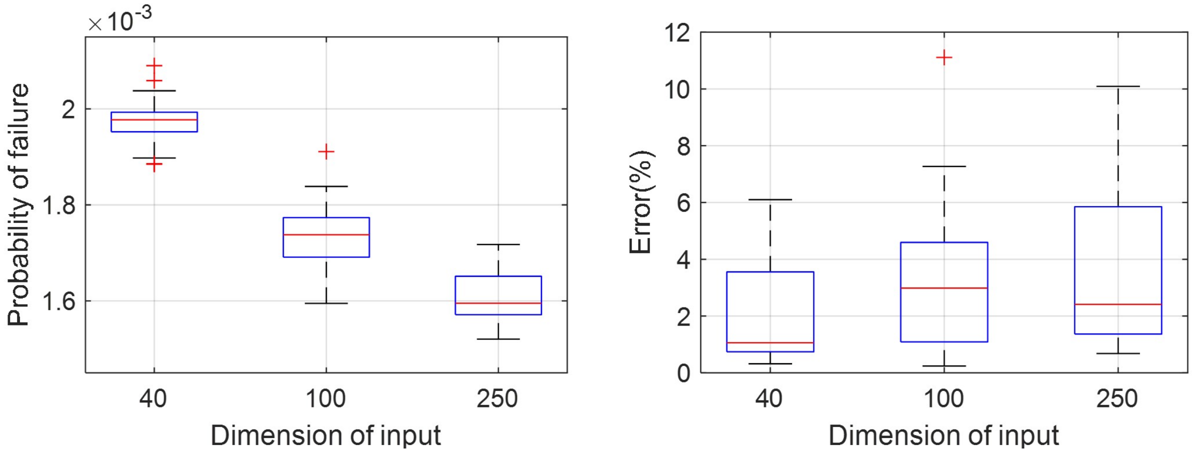


Figure 3.5. Box plot of 20 simulations

As mentioned in Section 3.2.1, the nonlinearity between the sufficient predictors and real response is ascending. Figure 3.6 shows the first ten sufficient predictors versus the real response based on the result of case 1 ($n = 40$). Since the other cases and the other three examples also have the same pattern, we provide the figure for illustration for only case 1 in this example. For GSIR, the first d sufficient predictors are obtained, and we use the first

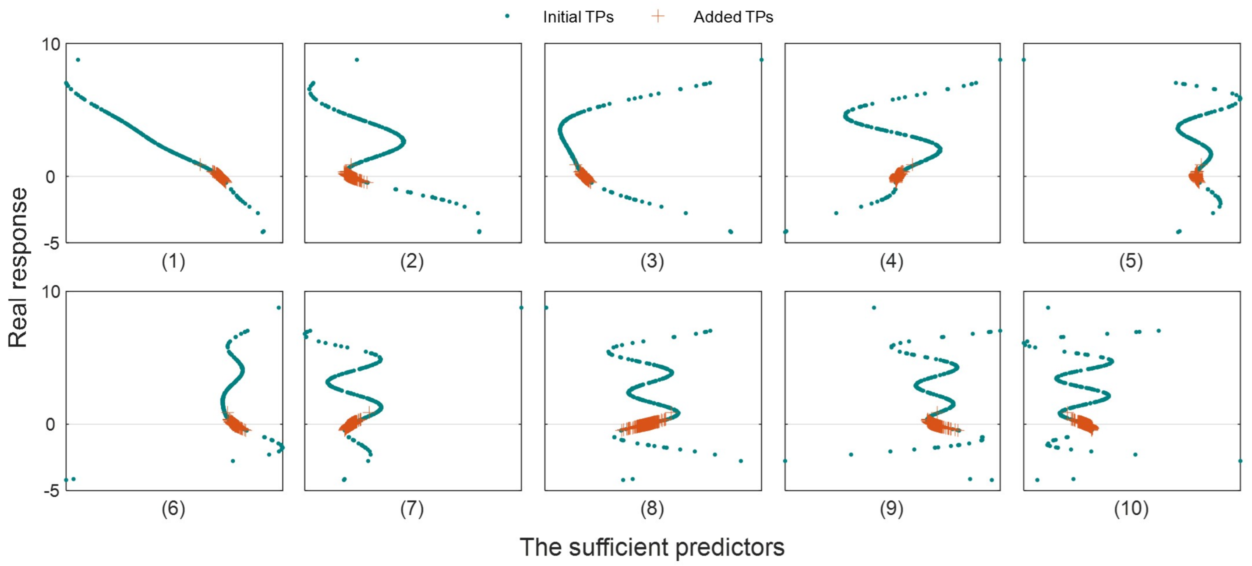


Figure 3.6. The relationship between sufficient predictors and real response

one for GP modeling and active learning. After the algorithm converges, we plot the first 10 sufficient predictors versus the real response. It is clear that the added training points by active learning cluster at the failure boundary for all sufficient predictors. Therefore, we need to use only the first sufficient predictor.

3.4.2 A Cantilever Beam

The second example is a beam (Figure 3.7) that is subjected to 106 random forces on the top. Six forces ($F_1, \ldots, F_6$) are lognormally distributed, and the rest of the forces ($F_7, \ldots, F_{106}$) are normally distributed. The locations ($l_{F_1}, \ldots, l_{F_{106}}$) that the forces, the width (w), height (h), and the yield strength (S_y) are also normally distributed. All the 215 random variables are independent. Their distributions are given in Table 3.2.

A failure would occur if the yield strength S_y is smaller than the maximum stress, and the performance function is therefore given by

$$g(\mathbf{X}) = S_y - \frac{6\sum_{\mathrm{i}=1}^{106} F_{\mathrm{i}} l_{F_{\mathrm{i}}}}{wh^2} \tag{3.41}$$

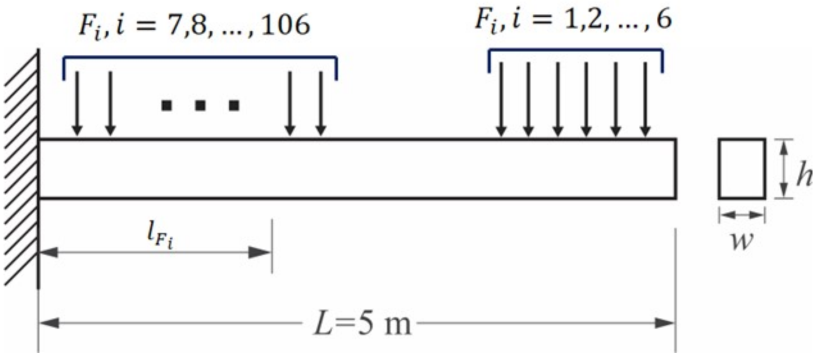


Figure 3.7. A cantilever beam

Table 3.2. Distributions of random variables in Example 2

Random variables	Distribution	Mean	Standard deviation
S_y (MPa)	Normal	720	60
w (m)	Normal	0.2	0.001
h (m)	Normal	0.4	0.001
$F_{\mathrm{i}}, \mathrm{i} = 1, 2, \dots, 6$ (kN)	Lognormal	30 + 5i	2.4 + 0.4i
$l_{F_{\mathrm{i}}}, \mathrm{i} = 1, 2, \dots, 6$ (m)	Normal	4.3 + 0.1i	0.01
$F_{\mathrm{i}}, \mathrm{i} = 7, 8, \dots, 106$ (kN)	Normal	10	1
$l_{F_{\mathrm{i}}}, \mathrm{i} = 7, 8, \dots, 106$ (m)	Normal	0.02i	0.01

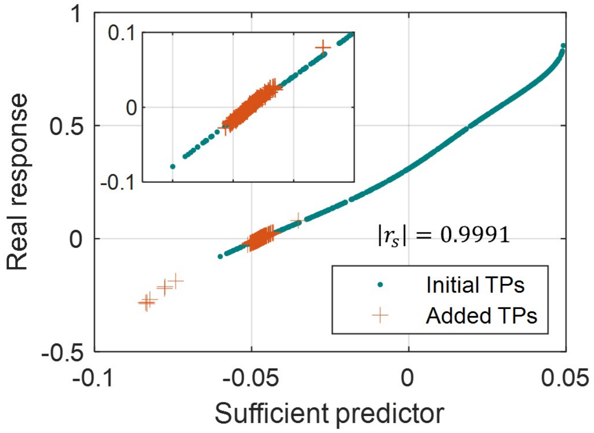


Figure 3.8. Sufficient predictor versus real response of Example 2

There are 600 initial training points in this example. We project the 215 input random variables in the high dimensional space to the sufficient predictor, and the relationship between the sufficient predictor and the real response is shown in Figure 3.8. A good monotonic relationship is obtained with the Spearman's correlation being 0.9991. The added training points are concentrated on the failure boundary.

The results of the proposed method with 20 runs are provided in Figure 3.9. The probability of failure and the corresponding error are within the intervals $[1.85 \times 10^{-6}, 2.08 \times 10^{-6}]$ and $[0.02\%, 2.08\%]$, respectively. And the median failure probability is 1.9448×10^{-6} with the corresponding error of 2.83%. In addition to the proposed method, MCS, FORM, SOSPA, and SIR-GP-IS are also used. The results are listed in Table 3.3. Although FORM calls the performance function only 648 times, its error is 5.9%. SOSPA is the most accurate method, but its computational cost is extremely high, with 24,084 function calls and a CoE of 112. SIR-GP-IS cannot converge within the maximum of 1,000 iterations. The results

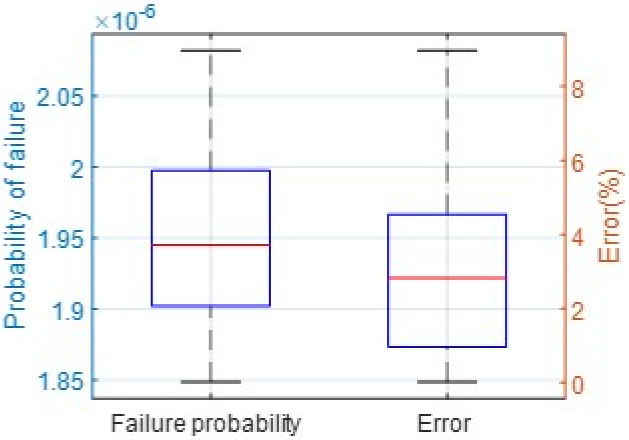


Figure 3.9. Statistical results of Example 2

Table 3.3. Results of different methods for Example 2

Methods	p_f	Error (%)	FC	CoE
MCS	1.9106×10^{-6}	-	1.6×10^{9}	7.4×10^{6}
FORM	1.7964×10^{-6}	6.0	648	3.0
SOSPA	1.9200×10^{-6}	0.5	24,084	112.0
SIR-GP-IS	2.3097×10^{-6}	20.89	1816	8.45
GSIR-GP-IS	1.9448×10^{-6}	2.83	1040.4	4.84

shown in Table 3 are from the last iteration. GSIR-GP-IS maintains a good balance between accuracy and efficiency with an error of 2.83% and a CoE of 4.84.

3.4.3 A Truss System

A dome-truss [120] consists of 52 bars with 21 nodes as shown in Figure 3.10, where numbers without dots represent bars and the others with dots mean nodes. All the nodes lie on an imaginary hemisphere with a radius of 240 in. The cross-section areas and the young's moduli of the bars are normally distributed. Six random forces ($F_1, \dots, F_6$) that point to the center of the imaginary hemisphere are applied to nodes 1-13. The forces are applied as follows: F_1 to node 1, F_2 to nodes 2, 4, F_3 to nodes 3, 5, F_4 to nodes 6, 10, F_5 to nodes 8, 12, and F_6 to nodes 7, 9, 11, and 13. The random variables are independent and their distributions are summarized in Table 3.4.

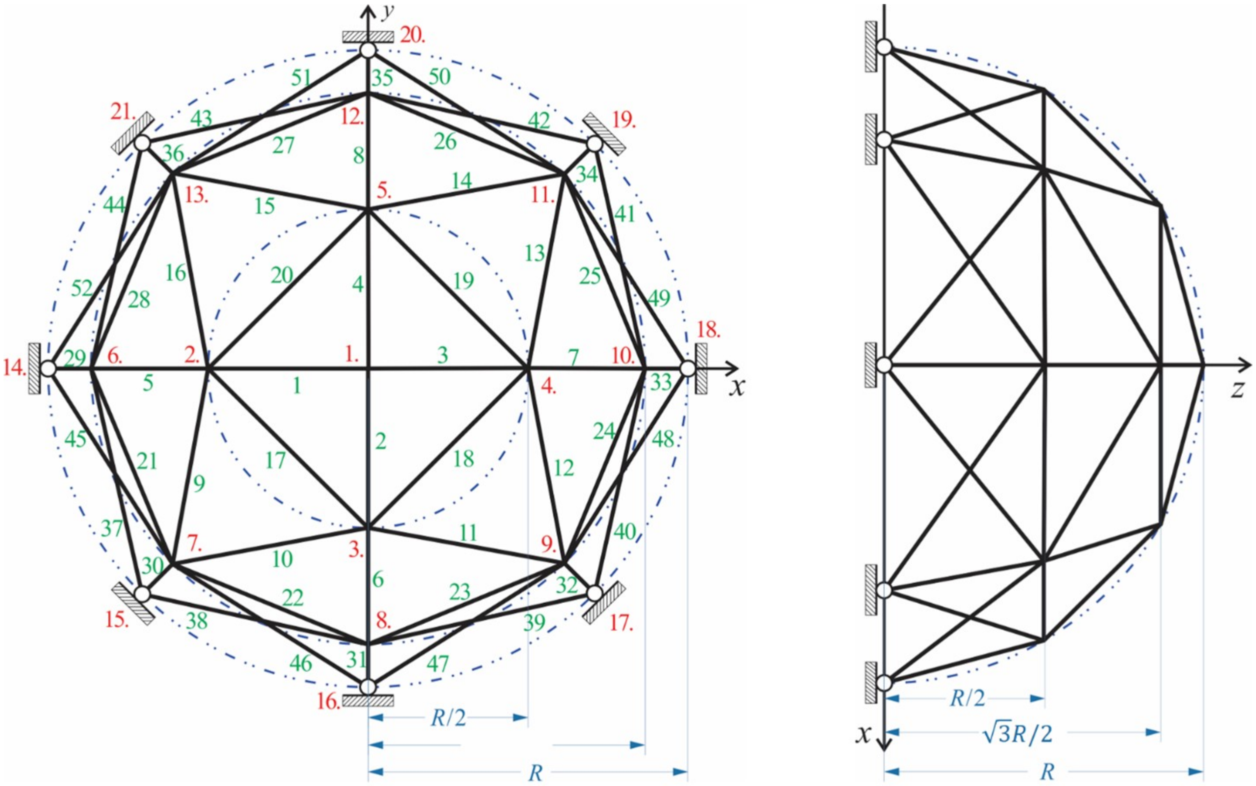
Table 3.4. Distributions of random variables in Example 3

Random variables	Distribution	Mean	Standard deviation
$E_{\mathrm{i}}, \mathrm{i} = 1 \sim 50$ (ksi)	Normal	2.5×10^4	1000
$A_{\mathrm{i}}, \mathrm{i} = 1 \sim 8$, and $29 \sim 36$ (in^2)	Normal	2	0.001
$A_{\mathrm{i}}, \mathrm{i} = 9 \sim 16$ (in^2)	Normal	1.2	0.0006
$A_{\mathrm{i}}, \mathrm{i} = 17 \sim 28$, and $37 \sim 52$ (in^2)	Normal	0.6	0.0003
F_1 (kip)	Normal	45	3.6
F_2 (kip)	Extreme	40	6.0
F_3 (kip)	Extreme	35	5.25
F_4 (kip)	Normal	30	2.4
F_5 (kip)	Normal	25	2.0
F_6 (kip)	Normal	20	1.6

The performance function is given in Eq. (3.42).

$$Y = g(\mathbf{X}) = \delta_0 - \delta(\mathbf{E}; \mathbf{A}; \mathbf{F}) \tag{3.42}$$

where $\delta_0 = 0.7$ in is the allowed maximum displacement of node 1, and δ_0 is the actual displacement of the same node, which is obtained by the finite element method (FEM).



(a) Top view

(b) Side view

Figure 3.10. A 52-bars truss system

$\mathbf{E} = [E_1, E_2, \ldots, E_{52}]^{\mathrm{T}}$ and $\mathbf{A} = [A_1, A_2, \ldots, A_{52}]^{\mathrm{T}}$ are vectors of the young's moduli and cross-sectional areas, respectively, $\mathbf{F} = [F_1, F_2, \ldots, F_6]^{\mathrm{T}}$ is a force vector.

We have Figure 3.11 shows that the sufficient predictor is monotonic to the real response that the Spearman's correlation is 0.9996. The failure boundary is identified by making use of the monotonic relationship through active learning.

The statistical results of the proposed method are given in Figure 3.12. For the 20 runs, most of the errors are smaller than 6%, and the maximum error is about 8.5%.

The results of all the methods are given in Table 3.5. GSIR-GP-IS is less accurate than SOSPA but is far more efficient than SOSPA. GSIR-GP-IS has only 764.4 function calls and a CoE of 6.95 while SOSPA has 6,771 function calls with a CoE of 61.55. The accuracy of FORM is poor, and its error is 15.77%. SIR-GP-IS cannot converge with a subspace of 1 in 1000 iterations and the results reported are from the last iteration.

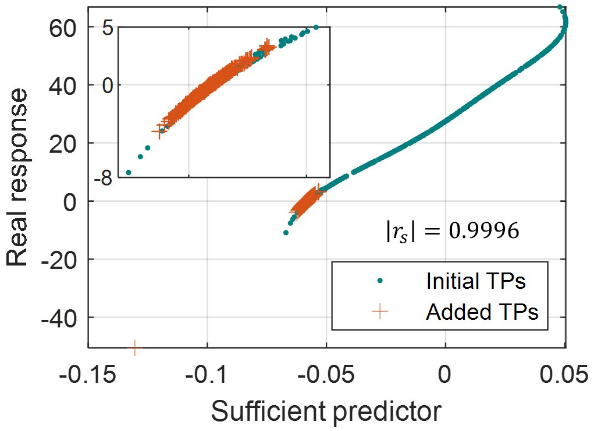


Figure 3.11. Sufficient predictor versus real response of Example 3

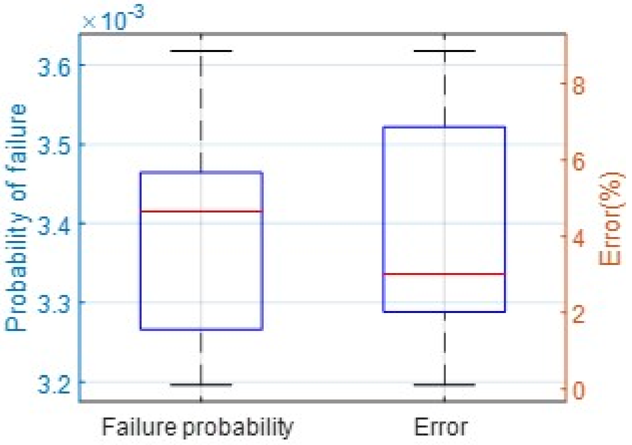


Figure 3.12. Statistical results of Example 3

Table 3.5. Results of different methods for Example 3

Methods	p_f	Error (%)	FC	CoE
MCS	3.506×10^{-3}	-	10^7	9.09×10^4
FORM	4.059×10^{-3}	15.77	555	5.05
SOSPA	3.529×10^{-3}	0.65	6,771	61.55
SIR-GP-IS	3.046×10^{-3}	13.11	1,511	13.7
GSIR-GP-IS	3.4148×10^{-3}	3.02	764.4	6.95

3.4.4 Nonlinear Seismic Dynamic Analysis of A Shear Frame

This example involves a 25-story shear frame structure (Figure 3.13) under stochastic seismic excitation. The masses ($m_1, \ldots, m_{25}$) of all stories are normal variables with means of 3×10^5 kg and coefficients of variation (C.O.V) of 0.05. The inter-story stiffnesses ($k_1, \ldots, k_{25}$) follow lognormal distributions with means of 1.2×10^8 N/m and C.O.V. of 0.1. The motion of the shear frame under seismic ground motions is characterized by the extended Bouc-Wen model [121] given by

$$\boldsymbol{M}\ddot{\boldsymbol{X}} + \boldsymbol{C}\dot{\boldsymbol{X}} + \alpha_h \boldsymbol{K}\boldsymbol{X} + (1-\alpha_h)\,\boldsymbol{K}\boldsymbol{Z} = -\boldsymbol{M}\ddot{u}_g(t) \tag{3.43}$$

where $\ddot{\boldsymbol{X}}$, $\dot{\boldsymbol{X}}$, and $\boldsymbol{X}$ are vectors of acceleration, velocity, and displacement, respectively; $\boldsymbol{M}$, $\boldsymbol{C}$ and $\boldsymbol{K}$ are the mass matrix, damping matrix, and stiffness matrix, respectively; α_h is a weighting parameter regarding hysteresis; $\boldsymbol{Z}$ is a vector of hysteretic displacement; $\ddot{u}_g(t)$ denotes the random ground motion and is given by

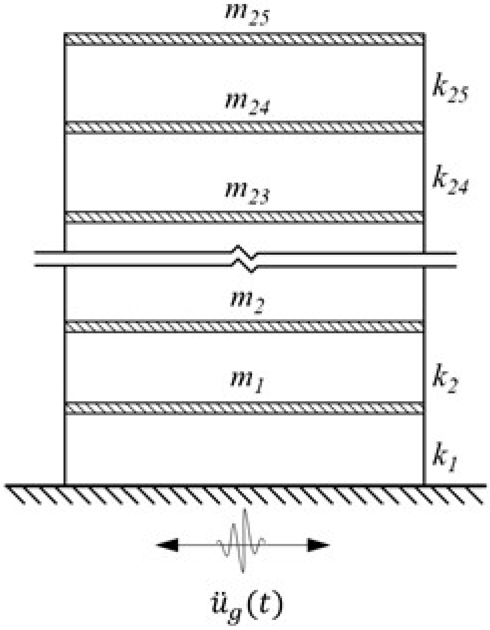


Figure 3.13. Schematic of a shear frame

$$\ddot{u}_g(t) = \xi_{NS}\ddot{u}_{NS}(t) + \xi_{WE}\ddot{u}_{WE}(t) \tag{3.44}$$

where ξ_{NS} and ξ_{WE} are independent extreme-value variables whose means and C.O.V both are 1 and 0.1, respectively; $\ddot{u}_{NS}(t)$ and $\ddot{u}_{WE}(t)$ are accelerations in the N-S and W-E directions, respectively, obtained from the EI Centro Earthquake [122]. There are 52 random variables in this example. The 13 parameters of the extended Bouc-Wen model used in this paper are $A = 1$, $\alpha_h = 0.04$, $\beta_h = 30$, $\gamma_h = 10$, $n = 1$, $\delta_\nu = 2000$, $\delta_\eta = 2000$, $\zeta_s = 0.99$, $q = 0.25$, $p = 1000$, $\psi = 0.05$, δ_ψ=5, $\lambda = 0.5$.

The damping matrix is $\boldsymbol{C} = \alpha\boldsymbol{M} + \beta\boldsymbol{K}$, where α and β are the damping coefficients, which are given by $\alpha = 0.02$ and $\beta = 0.01$. The maximum displacement of the first floor d_{max} is obtained by solving the nonlinear Ordinary Differential Equations system in Eq. (3.43).

$$d_{\max} = \max_{t\in[0,T]} \psi_1\left(m_1, \ldots, m_{25}, k_1, \ldots, k_{25}, \xi_{NS}, \xi_{WE}\right) \tag{3.45}$$

where $\psi_1(\cdot)$ denotes the function of the displacement of the first floor over time. When the maximum displacement exceeds a threshold ($d_{thres} = 32mm$), the shear frame fails. The performance function of the shear frame is defined by

$$Y = d_{\text{thres}} - d_{\max} \tag{3.46}$$

For this example, we have 200 initial training points and the IS population is 1×10^4. As shown in Figure 3.14, the proposed method can successfully identify the failure boundary for the nonlinear system, although some points are not at the failure boundary at the beginning stage of active learning. Figure 3.15 shows that the proposed method maintains a good accuracy for most of the 20 simulations.

The results of all the methods are provided in Table 3.6. GSIR-GP-IS outperforms other methods with an error of 2.39%, and 367.25 average function calls, and CoE of 6.93. SIR-GP-IS has a slightly larger average error of 6.02%, but its efficiency is much worse since it needs 938 function calls. The MPP search of FORM cannot converge in 50 iterations with

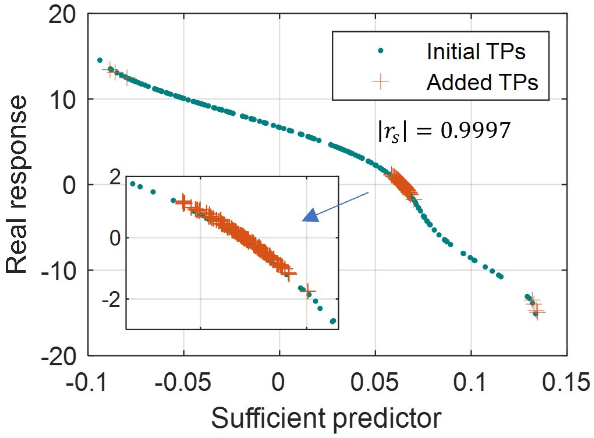


Figure 3.14. Sufficient predictor versus real response of Example 4

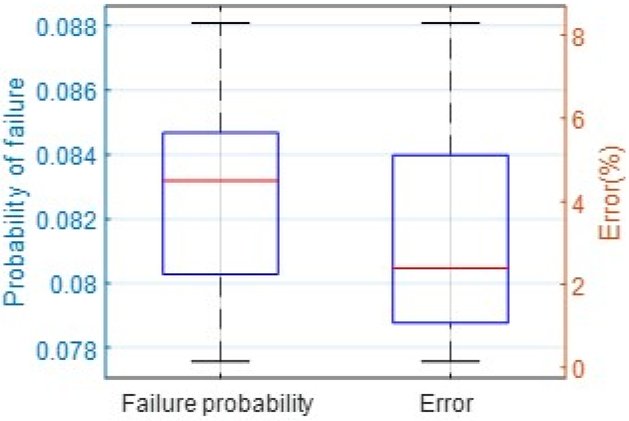


Figure 3.15. Statistical results of Example 4

Table 3.6. Results of different methods for Example 4

Methods	p_f	Error (%)	FC	CoE
MCS	8.46×10^{-2}	-	10^5	1.92×10^3
FORM*	0.4869	476	2,650	50.96
SOSPA*	0.8878	949	4,081	78.48
SIR-GP-IS	8.97×10^{-2}	6.02	938	18.04
GSIR-GP-IS	8.32×10^{-2}	2.39	367.25	7.06

*The MPP search does not converge in 50 iterations. Results are reported based on the MPP obtained at the 50^{th} iteration.

2650 function calls. The reported p_f for FORM is from the last iteration. Since SOSPA is based on the result of the MPP from FORM, SOSPA also has a large error for this example.

3.5 Summary

The proposed method combines the generalized sliced inverse regression (GSIR), importance sampling (IS), Gaussian process (GP), and active learning to relieve the curse of dimensionality of high dimensional reliability analysis. A GP model is constructed in a subspace after dimension reduction by GSIR. Then, active learning is used to refine the GP model. By iteratively adding new training points to the training set, the failure boundary is identified, which results in an accurate probability of failure. The four examples demonstrate that GSIR can successfully relieve the curse of dimensionality. The proposed method has a good potential to predict the reliability of high dimensional problems accurately and efficiently.

The proposed method has some limitations. It requires a sufficient number of initial training points, and this may not be computationally efficient for large-scale problems. It is possible that the use of a univariate subspace (the first sufficient predictor) may not be accurate enough for highly nonlinear problems. The proposed method may also produce a large error if multiple failure regions exist. To address the first two limitations, we will study the optimal balance between the number of initial training points and the number of added training points; we will also investigate the use of multiple predictors. For the third limitation, we will explore the possibility of using importance sampling centered at the most probable points of the multiple failure regions.

4. UNCERTAINTY QUANTIFICATION BY CONVOLUTIONAL NEURAL NETWORK GAUSSIAN PROCESS REGRESSION WITH IMAGE AND NUMERICAL DATA

Jianhua Yin[1], Xiaoping Du[2]

[1]School of Mechanical Engineering, Purdue University, West Lafayette, IN, 47907, United States

[2]Department of Mechanical and Energy Engineering, Indiana University - Purdue University Indianapolis, 723 W. Michigan Street, Indianapolis, IN 46202-5195

The preliminary work of this paper is presented in *2022 AIAA SciTech Forum*

Author Contributions

The authors confirm contribution to the paper as follows: Jianhua Yin and Xiaoping Du designed the study and contributed to the writing of the manuscript. Jianhua Yin developed detailed methodology, testing problems, and the code.

4.1 Background

Computer simulation models, derived from physical laws and principles, play a key role in engineering analysis and design. Their typical applications include prediction, sensitivity analysis, uncertainty quantification (UQ), what-if analysis, optimization, design space exploration, and systems design, which need to run the simulation many times. The models representing sophisticated physical details across wide spatial and time domains, however, are usually computationally demanding.

Models built from regressions are increasingly used as surrogates for computationally expensive computational models [123]–[128]. A computational model is run for a limited number of times, producing a set of labeled training points, based on which a surrogate model is built. Surrogate models can be built using the traditional Response Surface Modeling (RSM) [129] and can also be built with Machine Learning (ML) methods [130], such as Gaussian Process (GP) [131], [132], Support Vector Machines (SVM) [39], and Artificial Neural Networks (ANN) [36]. However, the computational cost soars with the increase of dimensionality.

Several methods [91], [133]–[136] have been proposed to improve the efficiency of UQ for high-dimensional problems. For example, a high dimensional reliability analysis method differentiates the contribution of important and unimportant variables to ensure high efficiency [91]. After dimension reduction, computations are performed in the reduced space of the important variables. Yin and Du [133] developed an active learning approach with dimension reduction by generalized sliced inverse regression (GSIR) to mitigate the curse of dimensionality. A deep neural network approach was developed in [35] to handle high-dimensional problems using auto-encoder and GP. These methods, however, are not efficient or applicable when image data exist.

In many applications, such as medicine, computational mechanics, material design, and additive manufacturing, both image and numerical data coexist. For example, in the severity assessment of stenosis [42]–[44], in addition to the patient image data, other numerical data, such as material properties and boundary conditions, are also inputted to computational fluid dynamics (CFD) simulation. The simulation can then replace a painful and costly invasive pressure measurement. As shown in Figure 4.1, the patient CT image is converted into a computer geometric model, and the geometry is meshed into a number of discrete elements for the subsequent hemodynamics (CFD) simulation, which then yields simulated velocity and pressure fields for decisions on diagnosis and potential treatment. Since there are at least thousands of meshed elements, the dimension of the input data to the CDF simulation is very high. Another example is the multidisciplinary optimization design of aircraft wings [86], [137] by finite element analysis (FEA). The structure or the geometry of the wings can be considered as image data; loading, material properties, and boundary conditions are numerical data.

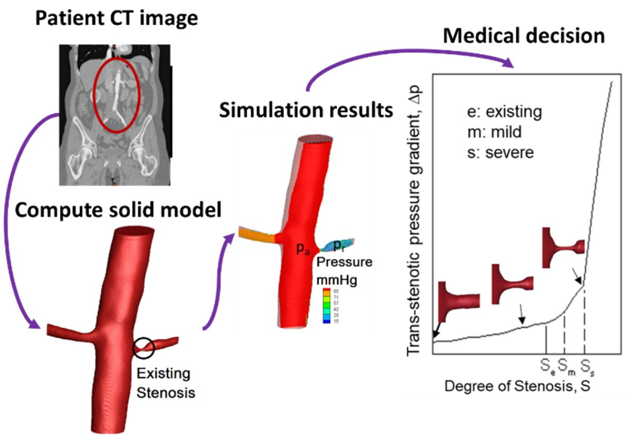


Figure 4.1. CDF simulation for stenosis severity assessment [43], [44].

If an accurate surrogate model is built with a limited number of expensive simulations (e.g. CFD, FEA), the inexpensive surrogate model can replace the original model for analysis and design. However, it is impossible or inconvenient to use typical surrogate modeling methods (RSM, GP, and SVM) to accommodate image data. Although the mixed input problem with both image and numerical data can be handled by a multi-input network [138], the implementation is complicated, and the efficiency may not be satisfactory.

Convolutional Neural Networks (CNN) [51]–[53] is a deep learning method that can deal with the high dimensional image input data. It is specifically designed to handle image data. CNN can recognize built-in features of the image directly through several convolution layers without relying on manual feature selection. It has achieved extraordinary successes in computer vision, image recognition, speech recognition, and engineering applications. Although CNN is designed to deal with only image-like data, recent studies [139] have shed light on the new use of CNN with numerical data. These methods convert numerical data into image data and enable a new capability of treating pure numerical data. Inspired by the studies, we develop a new concept to explore possible ways of using CNN for both image and numerical data. Also, motivated by the existing studies of GP [140], [141] and the mixed CNN [142], we propose to combine CNN with GP to overcome the shortcoming of CNN: it cannot provide the epistemic uncertainty to quantify the model error for the predicted response.

This work develops a CNN-based framework for high-dimensional UQ, which can accommodate both image and numerical data. The proposed method combines CNN with GP for UQ, providing the epistemic uncertainty associated with the prediction. Both epistemic uncertainty and aleatory uncertainty are considered when the CNN based surrogate model is used for uncertainty propagation, which results in more accurate statistical moments of the prediction. Besides, the CNN-based supervised dimension reduction technique avoids the curse of dimensionality of GP.

The rest of the paper is organized as follows. Section 4.2 reviews the basics of CNN and GP. Section 4.3 provides the procedure of converting numerical data into image data and merging converted images with existing images. In Section 4.4, we show how to integrate GP with CNN and perform UQ. In Section 4.5, we present three examples followed by conclusions in Section 4.6.

4.2 Literature Review

4.2.1 Convolutional Neural Networks

CNN [51], [52] is a class of ANN, designed for dealing with images and can be used for classification and regression. Unique features are extracted from images by the Convolution-Pooling operation that the dimension of images is reduced and optimal hyper-parameters of CNN are obtained. Fully connected layers are added to the end of convolutional and pooling layers to project the features to the output space. At the end of the fully connected layers, there is a classification layer or regression layer depending on the task.

The main layers of CNN are briefly discussed below.

- **Convolutional layer**: This layer is for extracting the features of input images. With a set of filters that scan across the image, the convolved feature is obtained by computing the dot product of filter and image elements. The extracted feature depends on the choice of filter. Different filters will result in different feature maps, which may influence the accuracy of the CNN model. Note that it is unnecessary to define the components of the filters beforehand; the parameters of filters are learned at the end of the training process.

- **Pooling layer**: This layer is for downsampling to reduce the dimension of the feature maps but keep the most important information. In this process, the number of parameters in the training process is decreased. Common pooling methods include average pooling, max pooling, and sum pooling. Average pooling means that only the average values of every sub-region are retained. Max pooling only keeps the maximum values of the sub-regions. Sum pooling is to sum all the elements of the sub-regions.

- **Fully connected layer**: This layer is after several folds of the convolution-pooling process. The output features of the last convolution-pooling, which is the input of the fully connected layer, are usually flattened into a one-dimensional array (a vector).

4.2.2 Gaussian Process Regression

Gaussian process regression (GPR) [114] is a kernel-based probabilistic method. A general form of the regression model is given by

$$G(\mathbf{x}) = \mathbf{f}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\beta} + \xi \tag{4.1}$$

where $G(\mathbf{x})$ is the response to be predicted; $\mathbf{x}$ is a vector of input variables; $\mathbf{f}(\mathbf{x})$ is a vector of the basis functions; $\boldsymbol{\beta}$ is a vector of coefficients; ξ is the noise term and follows a Gaussian distribution with the mean of 0 and the variance of σ^2, namely, $\xi \sim N(0, \sigma^2)$.

Given the training data $\mathbf{X} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\}$, the prior distribution of observed values $G(\mathbf{X})$ is

$$G(\mathbf{X}) \sim N\left(0, K(\mathbf{X}, \mathbf{X}) + \sigma^2 I_n\right) \tag{4.2}$$

where $K(\mathbf{X}, \mathbf{X})$ is a $n \times n$ symmetric covariance matrix; I_n is an n-dimensional unit matrix; and n is the number of training points. Since it is assumed that the data can be represented as a sample from a multivariate Gaussian distribution, the joint prior distribution of the training data $\mathbf{X}$ and the predictions at m test points $\mathbf{X}^* = \{\mathbf{x}^{(1*)}, \ldots, \mathbf{x}^{(m*)}\}$ is given by

$$\begin{bmatrix} G(\mathbf{X}) \\ G(\mathbf{X}^*) \end{bmatrix} \sim N\left(0, \begin{bmatrix} K_{\mathbf{XX}} + \sigma^2 I_n & K_{\mathbf{XX}^*}^{\mathrm{T}} \\ K_{\mathbf{XX}^*} & K_{\mathbf{X}^*\mathbf{X}^*} \end{bmatrix}\right) \tag{4.3}$$

where $K_{\mathbf{XX}} = K(\mathbf{X}, \mathbf{X})$ denotes the covariance matrix of training points; $K_{\mathbf{XX}^*} = K_{\mathbf{XX}^*}^{\mathrm{T}}$ is the covariance matrix between test points $\mathbf{X}^*$ and training points $\mathbf{X}$; and $K_{\mathbf{X}^*\mathbf{X}^*}$ is the covariance matrix of the test points. In this work, the Gaussian correlation function [114] is used as the kernel function. Then, the posterior distribution of a test point is given by

$$G(\mathbf{x}^*) \sim N\left(\mu_G(\mathbf{x}^*), \sigma_G^2(\mathbf{x}^*)\right) \tag{4.4}$$

where

$$\mu_G(\mathbf{x}^*) = K_* \left[K + \sigma^2 I_n\right]^{-1} G(\mathbf{x}) \tag{4.5}$$

$$\sigma_G^2(\mathbf{x}^*) = K_{**} - K_* \times \left[K + \sigma^2 I_n\right]^{-1} K_*^{\mathrm{T}} \tag{4.6}$$

Therefore, the mean $\mu_G(\mathbf{x}^*)$ and variance $\sigma_G^2(\mathbf{x}^*)$ of the prediction are obtained. It is assumed that $\mu_G(\mathbf{x}^*)$ provides the best estimate of the prediction while $\sigma_G^2(\mathbf{x}^*)$ indicates the model uncertainty of the prediction.

4.3 Convolutional Neural Network with Mixed Data

As mentioned previously, image data and numerical data could coexist in many applications. CNN is designed for dealing with images. In this section, we introduce how to convert numerical data into image data and how to merge the converted numerical data to exiting image data.

4.3.1 Overview

A model with mixed inputs is constructed when numerical and image data coexit, which is given by

$$y = g(\mathbf{x}, \mathbf{im}) \tag{4.7}$$

where $\mathbf{x}$ is a vector of numerical input variables or a 1D array, and $\mathbf{x} \in \mathbb{R}^{n_x \times 1}$ with n_x rows and 1 column; **im** is the input image, and $\mathbf{im} \in \mathbb{R}^{n_{\mathrm{im}} \times m_{\mathrm{im}} \times c_{\mathrm{im}}}$ with n_{im} rows, m_{im} columns, and c_{im} channels which demotes the depth of the image; y is the model response. The model in Eq. (4.7), however, is usually computationally expensive, and we then build its surrogate model, given by

$$y = \hat{g}(\mathbf{x}, \mathbf{im}) \tag{4.8}$$

The central strategy of the proposed method is to convert the numerical data $\mathbf{x}$ into images. Mathematically, it is a task to transform a 1D array into a 3D array. After the transformation, numerical data $\mathbf{x}$ becomes an image $\mathbf{im}_x$ as a 3D array. Denote the transformation by $T(\cdot)$; namely

$$\mathbf{x} = T\left(\mathbf{im}_x\right) \tag{4.9}$$

Then the new image $\mathbf{im}_x$ is merged with image $\mathbf{im}$. Denote the aggregated image by $\mathbf{Im} = (\mathbf{im}_x, \mathbf{im})$, and the input is now $\mathbf{Im}$. The surrogate model in Eq. (4.8) is usually built by CNN. After the conversion in Eq. (4.9), CNN can be used without any modifications, producing a surrogate or regression model

$$y = \hat{g}_{\text{CNN}}(\mathbf{Im}) \tag{4.10}$$

Once the surrogate model is built, a prediction can be made for a test point $(\mathbf{x}^*, \mathbf{im}^*)$ as follows

$$y = \hat{g}_{\text{CNN}}\left(\mathbf{x}^*, \mathbf{im}^*\right) = \hat{g}_{\text{CNN}}\left(T\left(\mathbf{im}_x^*\right), \mathbf{im}^*\right) = \hat{g}_{\text{CNN}}\left(\mathbf{Im}^*\right) \tag{4.11}$$

where $\mathbf{Im}^* = (\mathbf{im}_x^*, \mathbf{im}^*)$ and $\mathbf{im}_x^* = T^{-1}(\mathbf{x}^*)$, and $T^{-1}(\cdot)$ denotes the inverse transformation. Next, we introduce how to convert numerical data into image data.

4.3.2 Conversion of Numerical Data into Image Data

There are many ways to convert a 1D array of numerical data into a 3D array (an image). Herein we discuss two of them. One is the bar graph, and the other is the grayscale graph. Both ways can represent the measurable features of the numerical data. The dataset of $\mathbf{x}$ is first normalized to $[0, 1]$.

For the bar graph [139], there are many possible ways of transformation (numerical data to image data) for a given dataset. An example is shown in Figure 4.2 using a data point

$\mathbf{u} = (0.2, 0.5, 0.3, 0.5, 0.8)$. The height of the image in pixels is $h = \beta n_x + \gamma(n_x + 1)$, where β is the width of a single bar, and γ is the width of a gap between bars. We denote the maximum length of the bar by w, which is the width of transformed image. The actual heights of the bars are $\mathbf{u}w$, where $\mathbf{u} = (u_1, u_2, \dots, u_{n_x})$ are the normalized numerical data in a general sense.

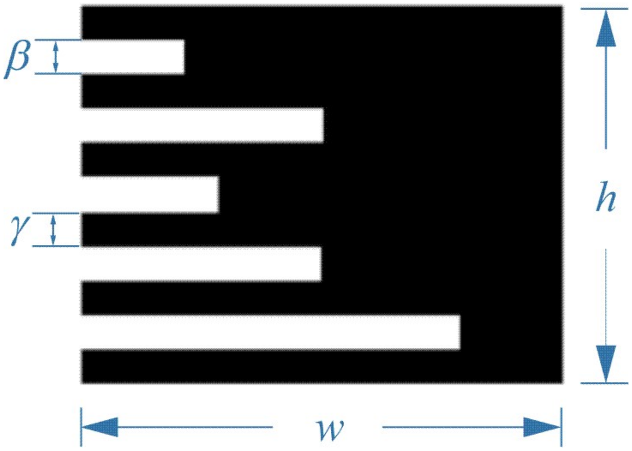


Figure 4.2. A converted bar graph.

The width of the image w influences the resolution of numerical features since the number of pixels is an integer. A continuous variable u is discretized into w intervals. The larger is w, the higher is the resolution of numerical features, but a longer time is needed to process the image. In practice, we set w to be the maximum value of $(n_{\text{im}}, m_{\text{im}})$.

$$w = \max\left(n_{\text{im}}, m_{\text{im}}\right) \tag{4.12}$$

If we have higher-dimensional numerical features, it is possible to merge the converged image using the height (h). Then, we can set w to be a larger value to preserve more accurate numerical feature.

The grayscale method is to transform the normalized data into grayscale images. The normalized value decides how dark or bright the pixels are. A pixel is black when the normalized value is 0 and white when the normalized value is 1. There are several ways to transform the normalized data into images. An example is shown in Figure 4.3 with the same data point $\mathbf{u} = (0.2, 0.5, 0.3, 0.5, 0.8)$. The image height is $h = \beta n_x$, and the width still is w, We convert the numerical data as a band image, where β is the width of a band.

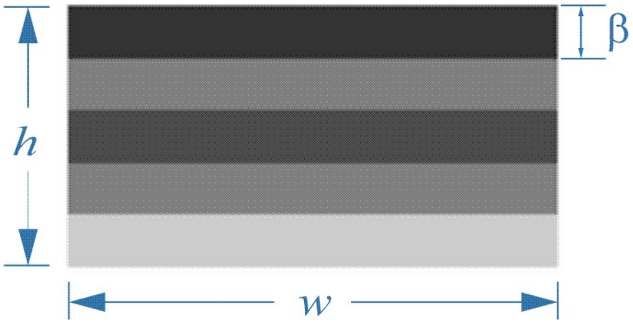


Figure 4.3. A converted grayscale graph.

The grayscale transformation is insensitive to the width of the image (w) since numerical features are embedded in the gray levels but are not impacted by the converted image size. However, this transformation method is limited by the gray level. For the commonly used 8-bit color format of grayscale images, the color from black to white is discretized to 256 different shades of color whose range is 0-255. Black is 0 and white is 255. The numerical features are, therefore, discretized to 256 intervals, which means the resolution is fixed. Also, the numerical features are not influenced by the orientation of the image for both transformation methods.

The width of bars or bands for the two conversion methods used in Figures 4.2 and 4.3 are large for a demonstration purpose. Based on our experience, the accuracy of CNN is insensitive to the value of the width, and a smaller value is preferred for a lower computational cost. Now we have converted images and original image data. Next, we discuss how to merge them to serve as the input of CNN.

4.3.3 Combination of Images

After numerical features $\mathbf{x}$ are converted into image $\mathbf{im}_x$, where $\mathbf{im}_x \in \mathbb{R}^{h \times w \times c_{\mathrm{i}}}$, we merge it into the existing image $\mathbf{im}$, where $\mathbf{im} \in \mathbb{R}^{n_{\mathrm{im}} \times m_{\mathrm{im}} \times c_{\mathrm{im}}}$. Letting $w = \max\left(n_{\mathrm{im}}, m_{\mathrm{im}}\right) = m_{\mathrm{im}}$, we merge the two images vertically, which results in an aggregated image $\mathbf{Im} = (\mathbf{im}_x; \mathbf{im})$, where $\mathbf{Im} \in \mathbb{R}^{(n_{\mathrm{im}}+w) \times m_{\mathrm{im}} \times c_{\mathrm{im}}}$. We can merge the transformed images in Figures 4.2 and

4.3 to an existing image as shown in Figure 4.4. Figure 4.4(a) indicates that the converted image $\mathbf{im}_x \in \mathbb{R}^{22\times 28\times 1}$ is merged to the existing image $\mathbf{im} \in \mathbb{R}^{28\times 28\times 1}$ vertically, which results in the aggregated image $\mathbf{Im} \in \mathbb{R}^{50\times 28\times 1}$. Similarly, as shown in Figure 4.4(b), we combine $\mathbf{im}_x \in \mathbb{R}^{15\times 28\times 1}$ and $\mathbf{im} \in \mathbb{R}^{28\times 28\times 1}$, resulting in $\mathbf{Im} \in \mathbb{R}^{43\times 28\times 1}$.

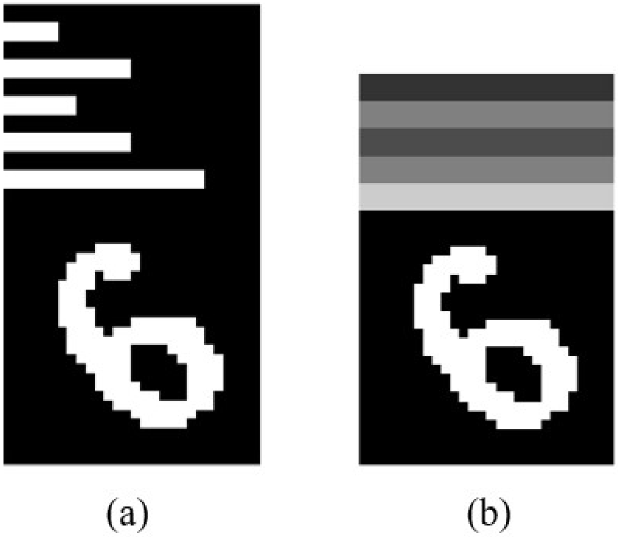


Figure 4.4. An example of aggregated images.

Alternatively, the transformed images can also be merged to other sides of the existing image if they are rotated to suitable orientations. The orientations of the transformed images do not have a strong influence on the result of the CNN training.

After the transformed images are merged to the existing images, the aggregated images are loaded to CNN for regression. However, given an untried image, it is difficult to know the prediction error (model uncertainty) by calling the trained CNN model. In the next section, we present the detailed procedures of combining CNN with GP to estimate the model error by epistemic uncertainty.

4.4 Uncertainty Quantification by CNN-GP with Mixed Data

The CNN training process is straightforward. Once the CNN structure is determined, the CNN model $\hat{g}_{\text{CNN}}$ is obtained using a set of training data $\left\{\left(\mathbf{Im}^{(1)}, y^{(1)}\right); \ldots; (\mathbf{Im}^{(n)}, y^{(n)})\right\}$. Given a new untried image $\mathbf{Im}^{(*)}$, we can predict its response $\hat{y}^{(*)}$. However, we do not know how accurate the prediction is unless we run the original computational model, which is against the purpose of regression. The discrepancy between the true response and predicted response is the model error whose uncertainty is called epistemic uncertainty. We can use epistemic uncertainty to estimate the model error. In this study, we combine CNN with GP to quantify the epistemic uncertainty of the surrogate model.

The purpose of regression is to build a surrogate model that can accurately replace the computational model. For the training data from physics-based simulations, the input data are assumed perfect without data uncertainty, and only epistemic uncertainty exists in the surrogate model. When the surrogate model is used to make predictions, however, the model inputs are random in most applications, such as stochastic loading, material properties, and other random parameters. This kind of uncertainty is called aleatory uncertainty or data uncertainty. As a result, we should quantify the effects of both types of uncertainty on the model output.

4.4.1 CNN-GP

The objective of this task is to quantify the epistemic uncertainty of the CNN prediction. However, CNN is not able to provide the model uncertainty of the prediction. This study estimates the model uncertainty by combining CNN with GPR because GPR is capable of quantifying epistemic uncertainty [57], [143]–[145].

Since GPR suffers from the curse of dimensionaltiy, dimension reduction is needed. As illustrated in Figure 4.5, we treat CNN as a supervised dimension reduction process by several folds of convolution, pooling, and activation layers. The high dimensional image is projected to the Z-latent space (one of the FC layers) with n_z latent variables. The latent variables are denoted by $\mathbf{z}$. Once the CNN model is built, we obtain the relationship between $\mathbf{z}$ and image input. Then we use GP to obtain the relationship between the response y and $\mathbf{z}$. Thus, the combination of CNN and GP yields the estimate of the prediction error by quantifying the epistemic uncertainty at a test point.

Recall that we use the training data $\left\{\left(\mathbf{Im}^{(1)}, y^{(1)}\right); \ldots; (\mathbf{Im}^{(n)}, y^{(n)})\right\}$ to construct the CNN model ($\hat{g}_{\text{CNN}}$). Once the model is obtained, we can easily project the image from high

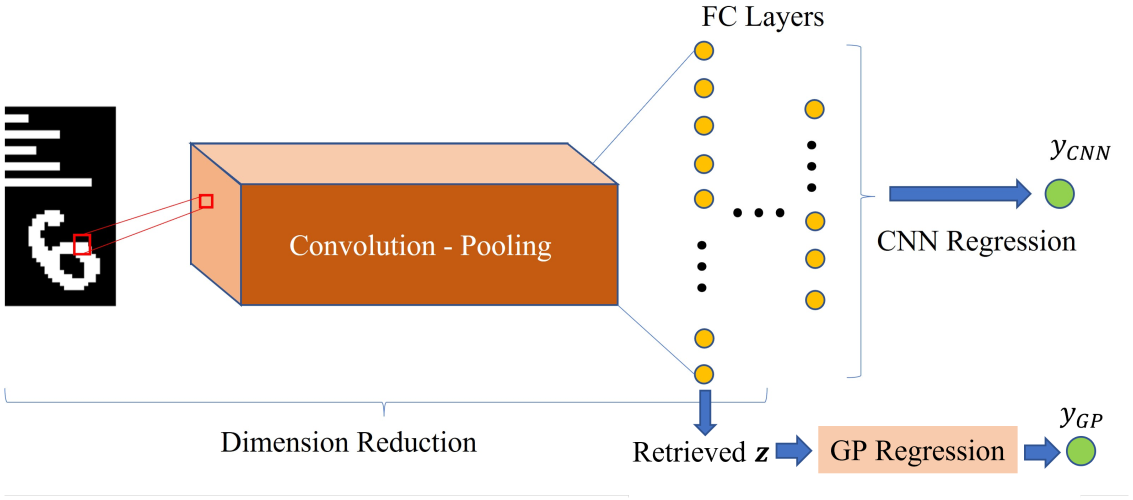


Figure 4.5. Illustration of CNN-GP.

dimensional space to the Z-latent space by retrieving the hyperparameters of CNN. The projection function is given by

$$\mathbf{z} = A_{\text{retri}}\left(\mathbf{Im}, \boldsymbol{\theta}_{\boldsymbol{h}}\right) \tag{4.13}$$

where $\boldsymbol{\theta}_{\boldsymbol{h}}$ denotes the hyperparameters of the trained CNN model. Therefore, the original high dimension images $\mathbf{IM} = \left\{\mathbf{Im}^{(1)}, \ldots, \mathbf{Im}^{(n)}\right\}$ that are used to train the CNN model is projected to the Z-latent space.

Using the latent space variables $\mathbf{Z} = \{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(n)}\}$ and the corresponding labels $Y = \{y^{(1)}, \ldots, y^{(n)}\}$ as the training data, we construct a GP model which is denoted by

$$y = G(\mathbf{z}) = G\left(A_{\text{retri}}\left(\mathbf{Im}, \boldsymbol{\theta}_{\boldsymbol{h}}\right)\right) \tag{4.14}$$

And the prior distribution of observed values $G(\mathbf{z})$ is given by

$$G(\mathbf{Z}) \sim N\left(0, K(\mathbf{Z}, \mathbf{Z}) + \sigma^2 I_n\right) \tag{4.15}$$

where $K(\mathbf{Z}, \mathbf{Z})$ is a $n \times n$ symmetric covirance matrix of $\mathbf{Z}$, which is simplified as $K_{\mathbf{ZZ}}$; the other parameters are introduced in Section 4.2.2. Given m untried points $\mathbf{Z}^* = \{\mathbf{z}^{(1*)}, \ldots, \mathbf{z}^{(m*)}\}$

which are obtained by Eq. (4.13) based on m new images $\mathbf{IM}^* = \{\mathbf{Im}^{(1*)}, \ldots, \mathbf{Im}^{(m*)}\}$, the joint distribution of the training points and the untried points is given by

$$\begin{bmatrix} G(\mathbf{Z}) \\ G(\mathbf{Z}^*) \end{bmatrix} \sim N\left(0, \begin{bmatrix} K_{\mathbf{ZZ}} + \sigma^2 I_n & K_{\mathbf{ZZ}^*}^{\mathrm{T}} \\ K_{\mathbf{ZZ}^*} & K_{\mathbf{Z}^*\mathbf{Z}^*} \end{bmatrix}\right) \tag{4.16}$$

where $K_{\mathbf{ZZ}^*}$ is the covariance matrix between the training points and untried points; $K_{\mathbf{Z}^*\mathbf{Z}^*}$ is the covariance matrix of the untried points. The posterior distribution of an untried point is given by

$$G(\mathbf{z}^*) \sim N\left(\mu_G(\mathbf{z}^*), \sigma_G^2(\mathbf{z}^*)\right) \tag{4.17}$$

where $\mu_G(\mathbf{z}^*)$ is the prediction mean and $\sigma_G^2(\mathbf{z}^*)$ is the variance of the prediction and they are obtained by

$$\mu_G(\mathbf{z}^*) = K_{\mathbf{ZZ}^*}\left[K_{\mathbf{ZZ}} + \sigma^2 I_n\right]^{-1} Y \tag{4.18}$$

$$\sigma_G^2(\mathbf{z}^*) = K_{\mathbf{Z}^*\mathbf{Z}^*} - K_{\mathbf{ZZ}^*} \times \left[K_{\mathbf{ZZ}} + \sigma^2 I_n\right]^{-1} K_{\mathbf{ZZ}^*}^{\mathrm{T}} \tag{4.19}$$

After dimension reduction using Eq. (4.13), $\mathbf{z}$ variables preserve the most important features of image data $\mathbf{Im}$. Therefore, we have following approximations,

$$K_{\mathbf{II}} \approx K_{\mathbf{ZZ}} \tag{4.20}$$

$$K_{\mathbf{II}^*} \approx K_{\mathbf{ZZ}^*} \tag{4.21}$$

$$K_{\mathbf{I}^*\mathbf{I}^*} \approx K_{\mathbf{Z}^*\mathbf{Z}^*} \tag{4.22}$$

where $K_{\mathbf{II}}$ is the covariance matrix between the training images $\mathbf{IM}$; $K_{\mathbf{II}^*}$ is the covariance matrix between the training images and test images $\mathbf{IM}^*$; $K_{\mathbf{I}^*\mathbf{I}^*}$ is the covariance matrix between the test images. Then, we can obtain the prediction mean of CNN-GP and its variance in terms of **Im** which are given by

$$\mu_G(\mathbf{Im}^*) = K_{\mathbf{II}^*}\left[K_{\mathbf{II}} + \sigma^2 I_n\right]^{-1} Y \approx K_{\mathbf{ZZ}^*}\left[K_{\mathbf{ZZ}} + \sigma^2 I_n\right]^{-1} Y \tag{4.23}$$

$$\sigma_G^2(\mathbf{Im}^*) = K_{\mathbf{I}^*\mathbf{I}^*} - K_{\mathbf{II}^*} \times \left[K_{\mathbf{II}} + \sigma^2 I_n\right]^{-1} K_{\mathbf{II}^*}^{\mathrm{T}} \approx K_{\mathbf{Z}^*\mathbf{Z}^*} - K_{\mathbf{ZZ}^*} \times \left[K_{\mathbf{ZZ}} + \sigma^2 I_n\right]^{-1} K_{\mathbf{ZZ}^*}^{\mathrm{T}} \tag{4.24}$$

It is known that $\mu_G(\mathbf{Im}^*)$ provides the best prediction and that $\sigma_G^2(\mathbf{Im}^*)$ represents the epistemic uncertainty which can used to estimate the error of CNN-GP at test points.

4.4.2 Uncertainty Quantification

As mentioned previously, epistemic uncertainty exists in surrogate models. When the surrogate model is used to propagate the uncertainty from input space to output space in a real application, there is aleatory uncertainty in the model input and epistemic uncertainty in the model itself. If the two types of uncertainty are not properly considered in the prediction, the results may not be reliable. Therefore, it is important to quantify both types of uncertainty.

In real applications, users are interested in knowing the mean (average) and the variance (uncertainty) of the response. Now, we discuss how to obtain the first two moments (the mean and variance) of the prediction. The use of the proposed approach is not limited to GP-based methods; it can be used for other meta-modeling techniques, such as SVM, polynomial chaos expansion (PCE), and ANN.

Recall that the prediction of CNN-GP has two parts, the mean prediction ($\mu_G(\mathbf{Im})$) and its variance ($\sigma_G^2(\mathbf{Im})$). When CNN-GP is used to make predictions at test points, there is epistemic uncertainty and aleatory uncertainty since the input may not be deterministic and CNN-GP may not be accurate at the test points. Next, we discuss how to integrate the two types of uncertainty in the quantification of the first two moments.

The mean and variance are given by

$$\mu_y = \int yp(y)dy \tag{4.25}$$

$$\sigma_y^2 = E\left((y-\mu_y)^2\right) = \int (y-\mu_y)^2\, p(y)dy \tag{4.26}$$

where $p(\cdot)$ is the probability density function (PDF) of y.

The response y is obtained by evaluating the expensive computational model with high dimensional input (numerical and image input) as shown in Eq. (4.7). It is expensive or impossible to obtain the first two moments using the integration method. Since we already have the computationally inexpensive CNN-GP model, we can use MCS to find the mean and variance. Given the MCS samples $\left\{\mathbf{Im}^{(1)},\dots,\mathbf{Im}^{(N)}\right\}$, we have the model responses $\left\{y^{(1)},\dots,y^{(N)}\right\}$ by calling the CNN-GP model in Eq. (4.14). The mean and variance can be estimated by

$$\mu_y = \int yp(y)dy \approx \frac{1}{N}\sum_{\mathrm{i}=1}^{N} y^{(\mathrm{i})} \tag{4.27}$$

$$\sigma_y^2 = E\left((y-\mu_y)^2\right) = \int (y-\mu_y)^2\, p(y)dy \approx \frac{1}{N-1}\sum_{\mathrm{i}=1}^{N}\left(y^{(\mathrm{i})}-\mu_y\right)^2 \tag{4.28}$$

The mean from Eq. (4.27) is with respect to aleatory uncertainty, which is rewritten as

$$E_A(y) = \mu_y = \frac{1}{N}\sum_{\mathrm{i}=1}^{N} y^{(\mathrm{i})} \tag{4.29}$$

As shown in Eqs. (4.23) and (4.24), the predicted response by CNN-GP follows a normal distribution, $y^{(\mathrm{i})} \sim N\left(\mu_G(\mathbf{Im}^{(\mathrm{i})}), \sigma_G^2(\mathbf{Im}^{(\mathrm{i})})\right)$. The mean prediction $\mu_G(\mathbf{Im}^{(\mathrm{i})})$ represents the best estimate of the prediction, and $\sigma_G^2(\mathbf{Im}^{(\mathrm{i})})$ represents the epistemic uncertainty at $\mathbf{Im}^{(\mathrm{i})}$. Taking the epistemic uncertainty into account, we obtain the mean response with both aleatory uncertainty and epistemic uncertainty.

$$E(y) = E_A(E_E(y)) = \frac{1}{N}\sum_{\mathrm{i}=1}^{N} E_E(y^{(\mathrm{i})}) = \frac{1}{N}\sum_{\mathrm{i}=1}^{N} u_G(\mathbf{Im}^{(\mathrm{i})}) \tag{4.30}$$

where $E(\cdot)$, $E_E(\cdot)$, and $E_A(\cdot)$ are all expectation operations. The subscripts E and A denote epistemic and aleatory, respectively.

The variance with respect to aleatory uncertainty is given by

$$V_A(y) = \frac{1}{N-1} \sum_{\mathrm{i}=1}^{N} \left(y^{(\mathrm{i})} - E_A(y)\right)^2 \tag{4.31}$$

where $V_A(\cdot)$ represents the variance operation. $V_A(y)$ is random due to the epistemic uncertainty of $y^{(\mathrm{i})}$. The mean of $V_A(y)$ with respect to epistemic uncertainty is given by

$$\begin{aligned} E_E\left(V_A(y)\right) &= \frac{1}{N-1} \sum_{\mathrm{i}=1}^{N} E_E\left(y^{(\mathrm{i})} - E(y)\right)^2 \\ &= \frac{1}{N-1} \sum_{\mathrm{i}=1}^{N} \left(E_E(y^{(\mathrm{i})}) - E(y)\right)^2 + \frac{1}{N-1} \sum_{\mathrm{i}=1}^{N} V_E\left(y^{(\mathrm{i})} - E(y)\right)^2 \end{aligned} \tag{4.32}$$

where

$$\left(E_E(y^{(\mathrm{i})}) - E(y)\right)^2 = \left(u_G(\mathbf{Im}^{(\mathrm{i})}) - \frac{1}{N} \sum_{\mathrm{i}=1}^{N} u_G(\mathbf{Im}^{(\mathrm{i})})\right)^2 \tag{4.33}$$

$$V_E\left(y^{(\mathrm{i})} - E(y)\right)^2 = \sigma_G^2(\mathbf{Im}^{(\mathrm{i})}) \tag{4.34}$$

Plugging Eqs. (4.30), (4.33) and (4.34) into Eq. (4.32), the expected variance of the prediction is given by.

$$V(y) = E_E(V_A(y)) = \frac{1}{N-1} \sum_{\mathrm{i}=1}^{N} \left(u_G(\mathbf{Im}^{(\mathrm{i})}) - \frac{1}{N} \sum_{\mathrm{i}=1}^{N} u_G(\mathbf{Im}^{(\mathrm{i})})\right)^2 + \frac{1}{N-1} \sum_{\mathrm{i}=1}^{N} \sigma_G^2(\mathbf{Im}^{(\mathrm{i})}) \tag{4.35}$$

Now we obtained the first two moments for the model response, which provides the best estimate of the prediction and the uncertainty of the prediction.

4.5 Examples

We provide three examples in this section. In the first example, we use the well-known MNIST dataset [146] to test the concept and show the detailed procedure. The second and the third examples are real engineering problems, which involve steady-state nonlinear heat transfer and transient nonlinear heat transfer. The accuracy of UQ is evaluated by comparisons with the results of MCS. For comparison with other machine learning methods, we provide the results of GP [147] and SVM [148] for the second and third examples.

4.5.1 Dynamics Problem using MNIST

MNIST is a database containing grayscale images of handwritten digits, which are commonly used for training and testing image processing systems. We change the problem to a dynamics problem so that we relate it with potential engineering applications.

The size of each image in MNIST is 28 × 28 pixels, and each digit has an associated label, which is the angle θ that the digit rotates. We convert the original problem into a dynamics problem. Since the MNIST dataset contains only image input, we modify it by adding numerical input variables. We assume that each digit is a rigid body and that its position in MNIST is its initial position. We use digit ”6” as an example for demonstration. As shown in Figure 4.6, initially the rigid body rotates about its center of mass at an initial angular velocity of ω_0 (rad/s), and the initial angular displacement of the body with respect to the vertical axis is θ (rad). To rotate the body back so that it is aligned with the vertical axis, we apply a moment M (N·m) in the opposite direction of ω_0. The body reaches an angular velocity ω (rad/s) in its final position when $\theta = 0$.

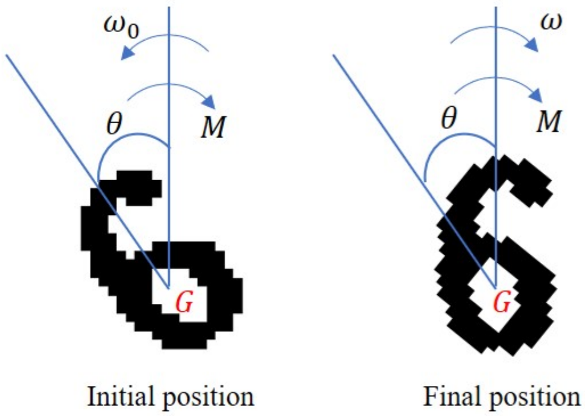


Figure 4.6. Illustration of the dynamics problem using a rigid body of a digit.

The task is as follows: Given the shape of the rigid body, its initial position θ, its initial angular velocity ω_0, and the external moment M, we find its final angular velocity ω. The input therefore includes $\mathbf{im} \in \mathbb{R}^{28\times28\times1}$ in MNIST and numerical variables $\mathbf{x} = (\omega_0,\ M)^{\mathrm{T}}$, and the output is ω. Then, we have both image and numerical data in the input. The data of ω_0 and M are generated randomly with a uniform distribution in the range of $[0, 1]$.

We now discuss how to generate data for label y. At first, we need to extract a rigid body from an image to calculate the inertia properties of the body. This is the task of segmentation. We define a threshold value for the segmentation. After testing with different threshold values, we found that a threshold of 0.2 is the best to keep the shapes of the extracted bodies smooth. The digit in Figure 4.6 is an example of the extracted body.

Assume the rigid body is placed on a smooth horizontal surface. From the law of energy conservation, we have

$$\frac{1}{2}I\omega_0^2 + M\theta = \frac{1}{2}I\omega^2 \tag{4.36}$$

where I is the moment of inertia about the center of mass, $\frac{1}{2}I\omega_0^2$ and $\frac{1}{2}I\omega^2$ are the initial and final kinetic energy, respectively, and $M\theta$ is the work functioned by the moment. The final angular velocity ω is obtained by solving Eq. (4.36). In other words, the numerical inputs are ω_0 and θ, and the model output is ω.

We next follow the proposed strategy to transform the numerical vector $\mathbf{x} = (\omega_0, M)^{\mathrm{T}}$ into image $\mathbf{im}_x$ and then merge $\mathbf{im}_x$ into the extracted rigid body $\mathbf{im}$. Using the bar graph transformation, we set $\beta = 1$ and $\gamma = 0$, and $w = 28$; thereby we have $\mathbf{im}_x \in \mathbb{R}^{2\times28\times1}$ and $\mathbf{Im} \in \mathbb{R}^{30\times28\times1}$.

The merged image data ($\mathbf{Im}$) is fed to CNN to perform the regression task. There are 10,000 samples of which half of them are for training and the rest are for validation. We employ four-fold convolution layers. The CNN structure and parameters are listed in Table 4.1.

The accuracy of the CNN regression with the bar graph transformation is shown in Figure 4.7. The scatter plot shows that the predictions against the true values are distributed around the 45° line, which indicates that the proposed method can well handle the mixed numerical and image data. The root-mean-square error (RMSE) of the validation is 0.67 rad/s.

Table 4.1. CNN structure for Example 1

Layer	Filter size	Filter number	Stride	Activation
Convolution layer 1	3×3	8	-	relu
Average pooling	2×2	-	2	-
Convolution layer 2	3×3	16	-	relu
Average pooling	2×2	-	2	-
Convolution layer 3	3×3	32		relu
Convolution layer 4	3×3	32	-	tanh
Fully connected layer 1	8 neurons	-	-	-
Fully connected layer 2	1 neuron	-	-	-

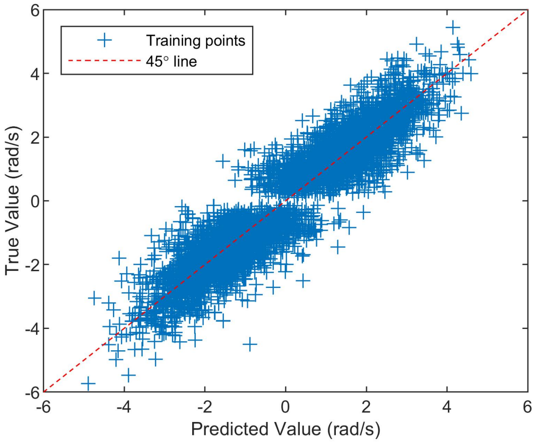


Figure 4.7. CNN regression accuracy of Example 1 with the bar transformation.

Following the procedure in Section 4.4.1, we estimate the model error using epistemic uncertainty. We first retrieve the variables in a latent space ($\mathbf{z}$), then the dimension of original high-dimension images is reduced. According to the CNN structure in Table 4.1, there are eight neurons in the first fully connected layer. Thus, we reduce the dimension of the images (30×28) to eight variables. Then, a GP model is constructed with respect to the eight variables in $\mathbf{z}$ and the model response y. Once we have the CNN-GP model, we can easily obtain the mean response and its standard deviation at a test point. We use the CNN-GP model and obtain the mean responses and standard deviations at 20 test points. The results are shown in Figure 4.8, where the circles represent the mean responses. The length of an error bar gives a 95% confidence interval (CI). Therefore, we know the confidence of the predictions with the epistemic uncertainty at the test points. The true responses or labels from MNIST are also plotted in Figure 4.8.

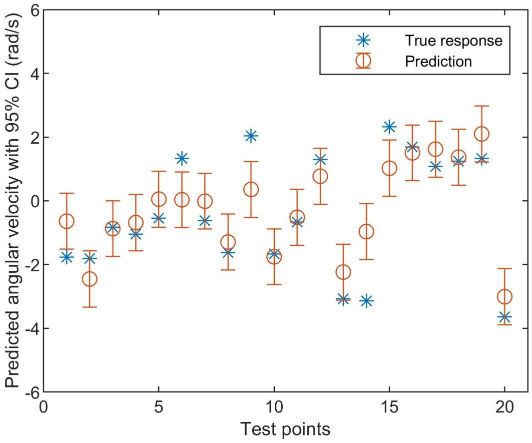


Figure 4.8. Model responses with epistemic uncertainty of Example 1.

This example is for only a demonstration of how to quantify model uncertainty. In the following two examples, we show the proposed method can be used in engineering applications with both epistemic uncertainty and aleatory uncertainty.

4.5.2 Steady-state Nonlinear Heat Transfer

This example involves a heat transfer analysis of a thin rectangle plate. The spatial domain is shown in Figure 4.9. The temperature is fixed along the left boundary (b_1) and is a random variable following a uniform distribution. The other three boundaries are thermal isolation without heat transfer. The temperature of point p_1 is the quantity of interest or the output.

The nonlinear heat transfer is governed by a partial differential equation (PDE) given by

$$-kt_z\nabla^2 T + 2h_c T + 2\epsilon\sigma T^4 = 2h_c T_a + 2\epsilon\sigma T_a^4 \tag{4.37}$$

where k is the thermal conductivity and is an uncertain parameter; t_z is the plate thickness; h_c is the convection coefficient; ϵ is the emissivity of the plate surface; σ is the Stefan-Boltzmann constant; T_a is the ambient temperature. Except for k, the other parameters

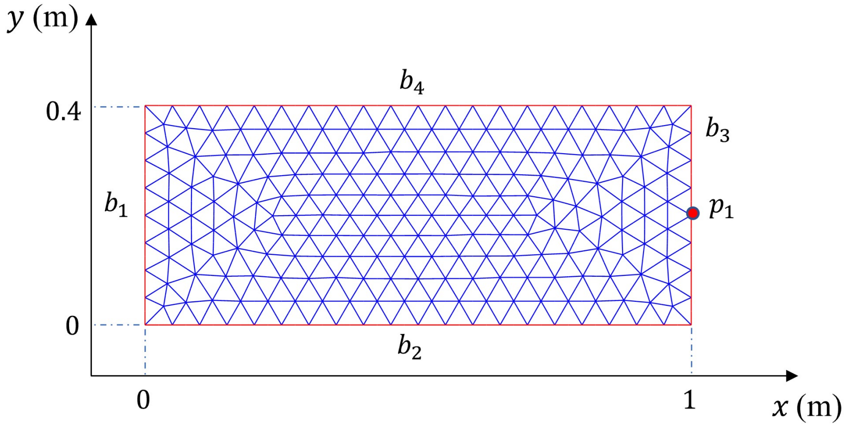


Figure 4.9. The spatial domain of the heat transfer problem.

are all constant. The randomness of k is characterized by a Gaussian random field, which is modeled by the truncated Karhunen-Loeve (K-L) expansion. The K-L expansion is given by

$$H(\boldsymbol{X}) = \mu(\boldsymbol{X}) + \sum_{\mathrm{i}=1}^{m} \sqrt{\lambda_{\mathrm{i}}}\varphi_{\mathrm{i}}(\boldsymbol{X})\xi_{\mathrm{i}} \tag{4.38}$$

where $\mu(\cdot)$ is the mean function of the random field; X is a 2D vector representing the spatial location; λ_{i} and φ_{i} are the eigenvalues and eigenfunctions of the auto-correlation function, respectively; ξ_{i} is a set of independent standard normal random variables; m is the truncation number. In this work, we use the squared exponential kernel for the auto-correlation function. The correlation between two arbitrary points is given by

$$\rho_{\mathrm{i,j}} = \exp\left\{-\left(\frac{\|\boldsymbol{X}_{\mathrm{i}} - \boldsymbol{X}_{\mathrm{j}}\|}{d}\right)^2\right\} \tag{4.39}$$

where d is the correlation length; $\boldsymbol{X}_{\mathrm{i}}$ and $\boldsymbol{X}_{\mathrm{j}}$ are two arbitrary points; and $\|\cdot\|$ represents the norm of a vector.

Once the mesh and random variables are known, we can generate realizations of the random field, which are represented by images. Boundary conditions are given as numerical data. We convert the numerical data into images and merge them to the corresponding realizations of the random field, which results in the merged image data for CNN.

We generate 10,000 samples, among which 1,600 samples are used for training, and the rest are for testing. Four examples of the combined images are shown in Figure 4.10. The dark bars represent different boundary conditions with different gray scales, and the rest are different realizations of the random field. All the labels (the temperature at p_1) are obtained by the Stochastic Finite Element Method (SFEM). The CNN structure is given in Table 4.2.

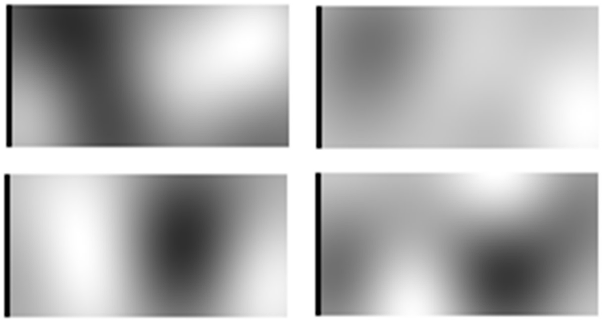


Figure 4.10. Examples of merged image data in Example 2.

Table 4.2. CNN structure of Example 2

Layer	Filter size	Filter number	Stride	Activation
Convolution layer 1	5×5	32	-	relu
Average pooling	2×2	-	2	-
Convolution layer 2	3×3	28	-	relu
Average pooling	2×2	-	2	-
Convolution layer 3	3×3	24	-	relu
Convolution layer 4	3×3	16	-	relu
Convolution layer 5	2×2	8	-	tanh
Fully connected layer 1	8 neurons	-	-	-
Fully connected layer 2	1 neuron	-	-	-

The regression result of CNN is shown in Figure 4.11. The predictions and true labels of test points scatter around the 45° line compactly. Recall that a mixed network CNN-GP is used to quantify the prediction uncertainty. A GP model is constructed with respect to the output of the first fully connected layer of the CNN. Since the layer has eight neurons, the dimension of the GP model input is eight. In other words, the CNN model serves as

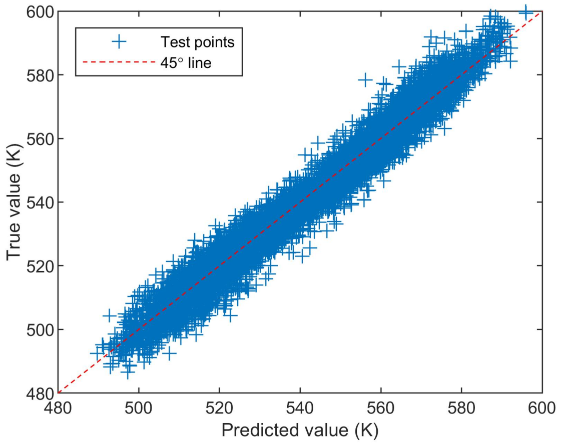


Figure 4.11. True label versus the predicted label by CNN of Example 2.

a supervised dimension reduction process, and the reduced dimension is eight. After the CNN-GP model is obtained, we have the mean prediction with uncertainty information. We randomly evaluate 20 test points using the CNN-GP model. We obtain the epistemic uncertainty at the test points as shown in Figure 4.12.

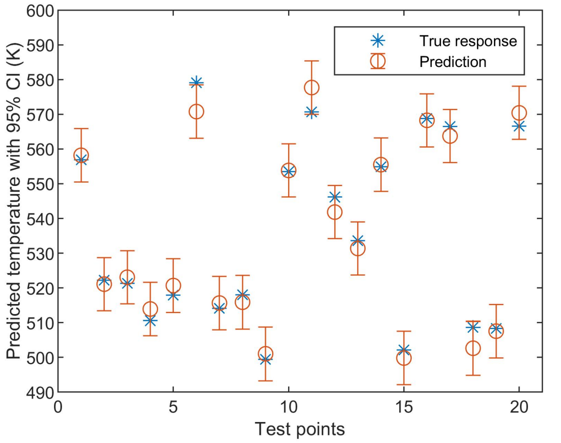


Figure 4.12. Mean predictions with epistemic uncertainty of Example 2.

We compare the accuracy of different methods using relative errors between predictions and true responses, which are given in Table 4.3. In the all tables, Std means standard deviation. In general, CNN-GP is more accurate than CNN. The average error and maximum error of CNN-GP are reduced compared with those of CNN. Besides, the standard deviations of the errors decrease as well for both training and test points. The GP model has the best accuracy for the training points, but the maximum error at the test points is the largest (6.42%). SVR has the largest standard deviations of errors.

Table 4.3. Regression accuracy of different methods of Example 2.

Methods	Samples	Error (%)		
		Average	Max	Std
GP	Training points	0	0	0
	Test points	0.14	6.42	0.26
SVM	Training points	0.46	3.35	0.67
	Test points	0.70	5.38	0.65
CNN	Training points	0.68	3.55	0.53
	Test points	0.70	5.64	0.57
CNN-GP	Training points	0.65	3.12	0.51
	Test points	0.67	5.48	0.55

We also compare the results of uncertainty propagation due to aleatory uncertainty (the random field k) and epistemic uncertainty for the GP model and CNN-GP using the 8,400 test points as shown in Table 4.4. The results of SVM and CNN only contain the effect of aleatory uncertainty because these two methods cannot quantify epistemic uncertainty. The results of the GP model and CNN-GP contain the effect of both types of uncertainty. The results from MCS are the ground truth because we use SFEM to obtain true response. It is shown that CNN-GP is more accurate than CNN and has a subtle difference compared with MCS. CNN is also accurate enough with the errors of mean and standard deviation being 0.11% and 3.79%, respectively. GP and SVM have better accuracy for the mean but with a larger error for the standard deviation compared with CNN-GP.

Table 4.4. Results of uncertainty propagation of Example 2.

Methods	Mean	Error (%)	Std	Error (%)
MCS	542.87	-	23.18	-
GP	542.92	0.01	22.78	1.74
SVM	542.91	0.01	20.15	13.06
CNN	542.28	0.11	22.30	3.79
CNN-GP	542.79	0.02	23.10	0.35

4.5.3 Transient Nonlinear Heat Transfer

The third example is concerned with transient nonlinear heat transfer in a thin plate. The domain is the same as Example 2 shown in Figure 4.9. The boundary conditions are the same: the temperature at b_1 follows a uniform distribution and other boundaries are thermal isolation. Still, the temperature at point p_1 is the quantity of interest. The difference between this example and Example 2 is that the heat transfer is transient or time dependent. The governing equation is given by

$$\rho C_p t_z \frac{\partial T}{\partial t} - k t_z \nabla^2 T + 2h_c T + 2\epsilon\sigma T^4 = 2h_c T_a + 2\epsilon\sigma T_a^4 \tag{4.40}$$

where ρ is the material density; C_p is the heat capacity with randomness that is characterized by a Gaussian random field; t is the time. Other parameters have been introduced in Example 2. Therefore, we have two random fields and one random variable in this example. Each random field is realized using 20 independent standard normal random variables.

Similarly, we generate random samples and mesh to have image data using Eqs. (4.38) and (4.39). Since we have two random fields, the images are with two channels. Using the same methods, we convert the temperature of the left boundary into images and merge with the existing images (random field realizations). One merged image data with two channels is shown in Figure 4.13.

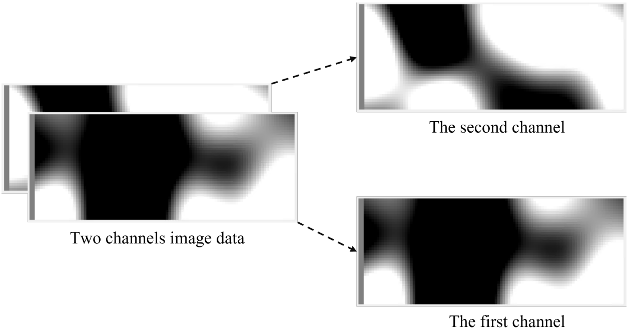


Figure 4.13. An example of the merged two-channel image data.

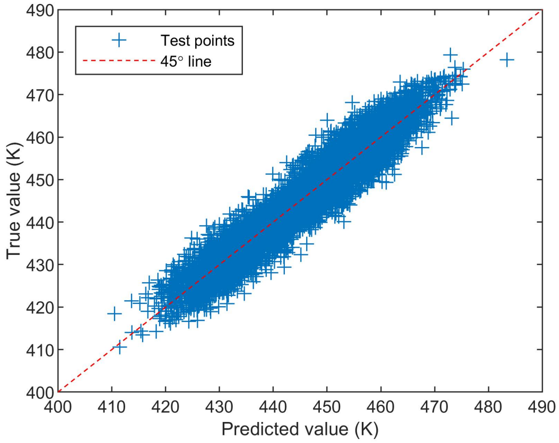


Figure 4.14. True label versus the predicted label by CNN of Example 3.

Table 4.5. Regression accuracy of different methods of Example 3.

Methods	Samples	Error (%)		
		Average	Max	Std
GP	Training points	0	0	0
	Test points	0.43	7.62	0.57
SVM	Training points	0.29	1.47	0.18
	Test points	0.46	2.55	0.36
CNN	Training points	0.42	2.10	0.31
	Test points	0.45	2.38	0.34
CNN-GP	Training points	0.36	2.05	0.28
	Test points	0.39	2.15	0.30

Table 4.6. Results of uncertainty propagation

Methods	Mean	Error (%)	Std	Error (%)
MCS	446.61	-	11.94	-
GP	446.55	0.01	11.18	6.37
SVM	446.32	0.07	10.34	13.40
CNN	446.90	0.06	11.54	3.35
CNN-GP	446.54	0.02	11.83	0.89

There are 10,000 samples generated in total. We use 2,500 samples for CNN-GP model training and 7,500 samples left are for testing. The true labels are obtained by the SFEM. This example uses the same CNN structure in Table 4.2.

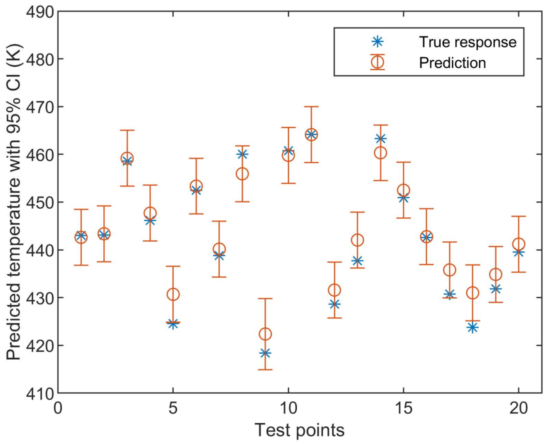


Figure 4.15. Mean predictions with epistemic uncertainty of Example 3.

The regression accuracy is shown in Figure 4.14. The true response and predicted response at test points are located around the 45° line. We quantitatively compare the accuracy of different methods in Table 4.5 using the percentage error. The GP model has the best accuracy for training points. Although CNN-GP has larger errors than the GP model for training points, it has the best accuracy at test points compared with other methods. The UQ results in Table 4.6 indicate that CNN-GP performs better for standard deviation with an error of 0.89%. The results of the mean for all methods are accurate enough. Overall, CNN-GP has the best performance.

4.6 Summary

This study proposes a regression method combining Convolutional Neural Network (CNN) and Gaussian Process Regression (GPR) for mixed numerical and image data. It also quantifies the epistemic uncertainty of the regression model. The strategy is to first transform numerical data into image data and then merge the converted images to existing images. Thus, the model input becomes pure image data that can then be fed into CNN for regression without any further modifications. The quantification of the model uncertainty is fulfilled by the integration of CNN and GPR. As a result, the output of the regression model contains the mean prediction and standard deviation of the prediction, which provides an estimate of the model error. With this distinctive feature, the proposed method can be potentially applied to engineering problems where numerical and image data coexist and can be employed for non-deterministic analysis and design where aleatory uncertainty exists in the model input. Potential applications include reliability-based design, robust design, and risk-based design.

In the future, we will use CNN-GP for reliability analysis (extreme events) and perform regression with the image-like model response. In this work, we use regular CNN without including the filter size, filter number, and the convolutional layer numbers of CNN as learnable parameters. The reasons are as follows: 1) How to efficiently include those parameters in optimization is still a challenging problem and is another ongoing research work. 2) The computational complexity will be increased when the parameters are added to the optimization as design variables. 3) The results of CNN constructed manually have good accuracy.

5. CONCLUSIONS

5.1 Contributions of The Dissertation

Uncertainty Quantification (UQ) is critical for engineering designs. High-dimensional problems are inevitably encountered in scientific and engineering fields. Current UQ methods are suffered from the curse of dimensionality which makes UQ computationally expensive in high-dimensional space. Besides, current UQ methods are not applicable when more complicated high-dimensional data (e.g. images) exist. To address those issues, three methodologies are proposed in this dissertation.

In the first methodology, we focus on the problems with high-dimensional numerical input which can be divided into important and unimportant variables according to their influence on the system. The unimportant variables are fixed at their first step MPP components so that the dimension is reduced. An accurate reliability analysis method is utilized in the reduced space. The final reliability is obtained by accounting for the contribution of both important and unimportant variables.

In the second methodology, we address one issue of the first method that the dimension is not reduced using the first method since the most of the variables are important or variables equally contribute to the system. Generalized Sliced Inverse Regression (GSIR) is used to reduce the dimension of numerical input variables. Combined with Gaussian Process (GP) modeling, importance sampling, and active learning, an accurate surrogate model for reliability analysis is built with limited performance function evaluations.

In the third methodology, we propose a UQ method based on Convolutional Neural Networks (CNN), which can accommodate both image and numerical data. We first convert numerical data into image data and merge the converted image data with existing images. Then, the aggregated image data are fed to CNN for training a CNN surrogate model. After the training process is finished, a Gaussian Process Regression (GPR) model is built with respect to the latent variables of CNN and true response (label) for UQ.

Based on the experiment results, we reach the following conclusions.

- In high dimensional problems, the contributions of random variables are different. We can divide them into important and unimportant variables according to their contributions. When most of the variables are unimportant variables, the dimensionality can be significantly reduced by fixing the unimportant variables to their percentiles, thereby with improve the efficiency. High accuracy reliability is obtained by counting the influence of both important and unimportant variables.

- When most of the variables are important or variables have similar contributions, GSIR can be used for dimension reduction to alleviate the curse of dimensionality.
- After dimension reduction by GSIR, active learning executed in the subspace largely improves the efficiency of meta-modeling methods with satisfying accuracy.
- The proposed CNN-based surrogate model, CNN-GP, can accommodate both image and numerical data with satisfying regression accuracy. This is achieved by first converting numerical data into image data and then merging the converted images to the existing images. The aggregated images serve as the input of CNN.
- The epistemic uncertainty is quantified to estimate the model error by integrating CNN with GPR. The accuracy of Uncertainty Propagation from input (mixed image and numerical input) to output is improved by accounting for both epistemic uncertainty and aleatory uncertainty.

5.2 Recommendations of Future Research

This dissertation provides three cost-efficient UQ methods to deal with uncertainty in high dimensional space. Several research topics could be further investigated to refine the tools developed.

The first research issue is the highly nonlinear problems for Method 1. Method 1 uses the first-step most probable point (MPP) components to determine the importance of variables. However, when the performance function is high nonlinear, the first-step MPP may not be accurate to determine the importance of variables. Although more MPP search iterations can potentially alleviate the problem, the efficiency will deteriorate. Other advanced sensitivity analysis methods may eligible to replace the first-step MPP approach.

The second research issue is that multiple failure regions exist in the design space. It is assumed that only one MPP exists in the first two methods. When multiple failure regions exist, there will be multiple MPPs, which could introduce errors. In system reliability analysis and highly nonlinear problems, it is common to have multiple MPPs. The advanced sampling methods instead of MPP-based importance sampling may be helpful to overcome this problem.

The third research issue is to apply CNN-GP to extreme events UQ, reliability analysis. The current UQ by CNN-GP focuses on the whole probability space to obtain the first two moments of the quantity of interest (QoI). When we perform reliability analysis, our focus will be the subdomain close to the failure boundary. Current CNN-GP may waste computational resources on unnecessary subdomains. To improve efficiency, integrating active learning is a possible solution for reliability analysis by CNN-GP.

In addition, the proposed methods could be applied to engineering design and optimization where uncertainty exists. For example, the proposed methods could be used for Reliability-based Design Optimization (RBDO) and Robust Design (RD). The efficient UQ methods in this dissertation could be used to evaluate the reliability with reduced computational cost in RBDO and RD until the reliability constraints are satisfied.

REFERENCES

[1] D. S. Moore, "Uncertainty," *On the shoulders of giants: New approaches to numeracy*, pp. 95–137, 1990.

[2] K. C. Kapur and L. R. Lamberson, "Reliability in engineering design," *New York*, 1977.

[3] C. Yu, "Literature review on aircraft structural risk and reliability analysis," *DSTO Aeronautical and Maritime Research Laboratory, DSTO-TR-1110*, 2001.

[4] J.-N. Yang and W. Trapp, "Reliability analysis of aircraft structures under random loading and periodic inspection," *AIAA Journal*, vol. 12, no. 12, pp. 1623–1630, 1974.

[5] Y. W. Liu and F. Moses, "A sequential response surface method and its application in the reliability analysis of aircraft structural systems," *Structural safety*, vol. 16, no. 1-2, pp. 39–46, 1994.

[6] T. Zou, S. Mahadevan, Z. Mourelatos, and P. Meernik, "Reliability analysis of automotive body-door subsystem," *Reliability Engineering & System Safety*, vol. 78, no. 3, pp. 315–324, 2002.

[7] J. P. Bliss and S. A. Acton, "Alarm mistrust in automobiles: How collision alarm reliability affects driving," *Applied Ergonomics*, vol. 34, no. 6, pp. 499–509, 2003.

[8] D. Hirschmann, D. Tissen, S. Schroder, and R. W. De Doncker, "Reliability prediction for inverters in hybrid electrical vehicles," *IEEE Transactions on Power Electronics*, vol. 22, no. 6, pp. 2511–2517, 2007.

[9] D. Shen, J. Yin, X. Du, and L. Li, "Distributed nonlinear model predictive control for heterogeneous vehicle platoons under uncertainty," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2021, pp. 3596–3603.

[10] J. Yin, D. Shen, X. Du, and L. Li, "Distributed stochastic model predictive control with taguchi's robustness for vehicle platooning," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[11] M. A. Hariri-Ardebili, "Risk, reliability, resilience (R3) and beyond in dam engineering: A state-of-the-art review," *International Journal of Disaster Risk Reduction*, vol. 31, pp. 806–831, 2018.

[12] K. Pietrzyk and C.-E. Hagentoft, "Reliability analysis in building physics design," *Building and Environment*, vol. 43, no. 4, pp. 558–568, 2008.

[13] F. Biondini, F. Bontempi, and P. G. Malerba, "Fuzzy reliability analysis of concrete structures," *Computers & Structures*, vol. 82, no. 13-14, pp. 1033–1052, 2004.

[14] J. Yin and X. Du, "A safety factor method for reliability-based component design," *Journal of Mechanical Design*, vol. 143, no. 9, 2021.

[15] G. Bird, "Monte-carlo simulation in an engineering context," *Progress in Astronautics and Aeronautics*, vol. 74, pp. 239–255, 1981.

[16] C. Z. Mooney, *Monte carlo simulation*, 116. Sage, 1997.

[17] R. Melchers, "Importance sampling in structural systems," *Structural safety*, vol. 6, no. 1, pp. 3–10, 1989.

[18] R. M. Neal, "Annealed importance sampling," *Statistics and Computing*, vol. 11, no. 2, pp. 125–139, 2001.

[19] S.-K. Au and J. L. Beck, "Estimation of small failure probabilities in high dimensions by subset simulation," *Probabilistic Engineering Mechanics*, vol. 16, no. 4, pp. 263–277, 2001.

[20] S. Song, Z. Lu, and H. Qiao, "Subset simulation for structural reliability sensitivity analysis," *Reliability Engineering & System Safety*, vol. 94, no. 2, pp. 658–665, 2009.

[21] A. M. Hasofer and N. C. Lind, "Exact and invariant second-moment code format," *Journal of the Engineering Mechanics division*, vol. 100, no. 1, pp. 111–121, 1974.

[22] M. Hohenbichler, S. Gollwitzer, W. Kruse, and R. Rackwitz, "New light on first-and second-order reliability methods," *Structural safety*, vol. 4, no. 4, pp. 267–284, 1987.

[23] K. Breitung, "Asymptotic approximations for multinormal integrals," *Journal of Engineering Mechanics*, vol. 110, no. 3, pp. 357–366, 1984.

[24] H. Wu, Z. Hu, and X. Du, "Time-dependent system reliability analysis with second-order reliability method," *Journal of Mechanical Design*, vol. 143, no. 3, 2021.

[25] X. Du and A. Sudjianto, "First order saddlepoint approximation for reliability analysis," *AIAA Journal*, vol. 42, no. 6, pp. 1199–1207, 2004.

[26] Z. Hu and X. Du, "Saddlepoint approximation reliability method for quadratic functions in normal variables," *Structural safety*, vol. 71, pp. 24–32, 2018.

[27] H. Wu and X. Du, "System reliability analysis with second-order saddlepoint approximation," *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering*, vol. 6, no. 4, p. 041 001, 2020.

[28] R. Jin, X. Du, and W. Chen, "The use of metamodeling techniques for optimization under uncertainty," *Structural and Multidisciplinary Optimization*, vol. 25, no. 2, pp. 99–116, 2003.

[29] S. Isukapalli, A. Roy, and P. Georgopoulos, "Stochastic response surface methods (SRSMs) for uncertainty propagation: Application to environmental and biological systems," *Risk Analysis*, vol. 18, no. 3, pp. 351–363, 1998.

[30] B. Echard, N. Gayton, and M. Lemaire, "AK-MCS: An active learning reliability method combining kriging and monte carlo simulation," *Structural Safety*, vol. 33, no. 2, pp. 145–154, 2011.

[31] H. Wu, Z. Zhu, and X. Du, "System reliability analysis with autocorrelated kriging predictions," *Journal of Mechanical Design*, vol. 142, no. 10, 2020.

[32] M. das Chagas Moura, E. Zio, I. D. Lins, and E. Droguett, "Failure and reliability prediction by support vector machines regression of time series data," *Reliability Engineering & System Safety*, vol. 96, no. 11, pp. 1527–1534, 2011.

[33] A. H. Elhewy, E. Mesbahi, and Y. Pu, "Reliability analysis of structures using neural network method," *Probabilistic Engineering Mechanics*, vol. 21, no. 1, pp. 44–53, 2006.

[34] C. Chen, Y. Liu, X. Sun, *et al.*, "Reliability analysis using deep learning," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers, vol. 51739, 2018, V01BT02A040.

[35] M. Li and Z. Wang, "Deep learning for high-dimensional reliability analysis," *Mechanical Systems and Signal Processing*, vol. 139, p. 106 399, 2020.

[36] H. Li, J. Yin, and X. Du, "Label free uncertainty quantification," in *AIAA SCITECH 2022 Forum*, 2022, p. 1097.

[37] I. Kaymaz, "Application of kriging method to structural reliability problems," *Structural Safety*, vol. 27, no. 2, pp. 133–151, 2005.

[38] Z. Zhu and X. Du, "Reliability analysis with monte carlo simulation and dependent kriging predictions," *Journal of Mechanical Design*, vol. 138, no. 12, 2016.

[39] Y. Wang, X. Yu, and X. Du, "Improved reliability-based optimization with support vector machines and its application in aircraft wing design," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[40] N. A. Shrivastava, A. Khosravi, and B. K. Panigrahi, “Prediction interval estimation of electricity prices using pso-tuned support vector machines,” *IEEE Transactions on Industrial Informatics*, vol. 11, no. 2, pp. 322–331, 2015.

[41] J. E. Hurtado and D. A. Alvarez, “Neural-network-based reliability analysis: A comparative study,” *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 1-2, pp. 113–132, 2001.

[42] H. Wu, M. Khan, X. Du, A. P. Sawchuk, and H. W. Yu, “Reliability analysis for image-based non-invasive pressure quantification in aortorenal artery systems,” *Circulation Research*, vol. 125, no. Suppl_1, A122–A122, 2019.

[43] H. Yu, C. Rong, X. Jin, *et al.*, “Fast and noninvasive evaluation of in vivo pressure in stenosed aortoiliac arteries,” *Journal of Vascular Surgery*, vol. 72, no. 3, e308–e309, 2020.

[44] H. Yu, M. Khan, H. Wu, *et al.*, “Inlet and outlet boundary conditions and uncertainty quantification in volumetric lattice boltzmann method for image-based computational hemodynamics,” *Fluids*, vol. 7, no. 1, 2022.

[45] M. Rosenblatt, “Remarks on a multivariate transformation,” *The annals of mathematical statistics*, vol. 23, no. 3, pp. 470–472, 1952.

[46] Y.-T. Wu, “Computational methods for efficient structural reliability and reliability sensitivity analysis,” *AIAA Journal*, vol. 32, no. 8, pp. 1717–1723, 1994.

[47] A. Saltelli, “Sensitivity analysis for importance assessment,” *Risk Analysis*, vol. 22, no. 3, pp. 579–590, 2002.

[48] K.-C. Li, “Sliced inverse regression for dimension reduction,” *Journal of the American Statistical Association*, vol. 86, no. 414, pp. 316–327, 1991.

[49] R. D. Cook and L. Ni, “Sufficient dimension reduction via inverse regression: A minimum discrepancy approach,” *Journal of the American Statistical Association*, vol. 100, no. 470, pp. 410–428, 2005.

[50] K.-Y. Lee, B. Li, and F. Chiaromonte, “A general theory for nonlinear sufficient dimension reduction: Formulation and estimation,” *The Annals of Statistics*, vol. 41, no. 1, pp. 221–249, 2013.

[51] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.

[52] Y. LeCun, B. Boser, J. S. Denker, *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[53] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, IEEE, 2017, pp. 1–6.

[54] X. Du and Z. Hu, "First order reliability method with truncated random variables," *Journal of Mechanical Design*, vol. 134, no. 9, 2012.

[55] D. I. Papadimitriou and Z. P. Mourelatos, "Reliability-based topology optimization using mean-value second-order saddlepoint approximation," *Journal of Mechanical Design*, vol. 140, no. 3, 2018.

[56] Z. Hu and X. Du, "Mixed efficient global optimization for time-dependent reliability analysis," *Journal of Mechanical Design*, vol. 137, no. 5, 2015.

[57] Y. Jung, K. Kang, H. Cho, and I. Lee, "Confidence-based design optimization for a more conservative optimum under surrogate model uncertainty caused by gaussian process," *Journal of Mechanical Design*, vol. 143, no. 9, p. 091 701, 2021.

[58] M. Papadrakakis and N. D. Lagaros, "Reliability-based structural optimization using neural networks and monte carlo simulation," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 32, pp. 3491–3507, 2002.

[59] M. Moarefzadeh and R. Melchers, "Directional importance sampling for ill-proportioned spaces," *Structural Safety*, vol. 21, no. 1, pp. 1–22, 1999.

[60] A. Dey and S. Mahadevan, "Ductile structural system reliability analysis using adaptive importance sampling," *Structural Safety*, vol. 20, no. 2, pp. 137–154, 1998.

[61] A. L. Kaminsky, Y. Wang, and K. Pant, "An efficient batch k-fold cross-validation voronoi adaptive sampling technique for global surrogate modeling," *Journal of Mechanical Design*, vol. 143, no. 1, 2020.

[62] R. Y. Rubinstein and D. P. Kroese, *Simulation and the Monte Carlo method.* John Wiley Sons, 2016, vol. 10.

[63] S. Engelund and R. Rackwitz, "A benchmark study on importance sampling techniques in structural reliability," *Structural Safety*, vol. 12, no. 4, pp. 255–276, 1993.

[64] W. Chen, M. Fuge, and J. Chazan, "Design manifolds capture the intrinsic complexity and dimension of design spaces," *Journal of Mechanical Design*, vol. 139, no. 5, 2017.

[65] S. Sarkar, S. Mondal, M. Joly, *et al.*, "Multifidelity and multiscale bayesian framework for high-dimensional engineering design and calibration," *Journal of Mechanical Design*, vol. 141, no. 12, 2019.

[66] P. Pandita, I. Bilionis, and J. Panchal, "Extending expected improvement for high-dimensional stochastic optimization of expensive black-box functions," *Journal of Mechanical Design*, vol. 138, no. 11, 2016.

[67] N. Knerr and D. Selva, "Cityplot: Visualization of high-dimensional design spaces with multiple criteria," *Journal of Mechanical Design*, vol. 138, no. 9, 2016.

[68] T. H. Ha, K. Lee, and J. T. Hwang, "Large-scale multidisciplinary optimization under uncertainty for electric vertical takeoff and landing aircraft," in *AIAA SCITECH 2020 Forum*, p. 0904.

[69] I. M. Sobol', "Theorems and examples on high dimensional model representation," *Reliability Engineering & System Safety*, vol. 79, no. 2, pp. 187–193, 2003.

[70] S. Rahman and H. Xu, "A univariate dimension-reduction method for multi-dimensional integration in stochastic mechanics," *Probabilistic Engineering Mechanics*, vol. 19, no. 4, pp. 393–408, 2004.

[71] S. Rahman, "Global sensitivity analysis by polynomial dimensional decomposition," *Reliability Engineering & System Safety*, vol. 96, no. 7, pp. 825–837, 2011.

[72] S. Xie, B. Pan, and X. Du, "High dimensional model representation for hybrid reliability analysis with dependent interval variables constrained within ellipsoids," *Structural and Multidisciplinary Optimization*, vol. 56, no. 6, pp. 1493–1505, 2017.

[73] K. H. Hajikolaei and G. Gary Wang, "High dimensional model representation with principal component analysis," *Journal of Mechanical Design*, vol. 136, no. 1, 2013.

[74] X. Yue, J. Zhang, W. Gong, M. Luo, and L. Duan, "An adaptive PCE-HDMR metamodeling approach for high-dimensional problems," *Structural and Multidisciplinary Optimization*, pp. 1–22, 2021.

[75] J. W. Park, H. Cho, and I. Lee, "Selective dimension reduction method (DRM) to enhance accuracy and efficiency of most probable point (MPP)–based DRM," *Structural and Multidisciplinary Optimization*, vol. 61, no. 3, pp. 999–1010, 2020.

[76] K. Kang and I. Lee, "Efficient high-dimensional metamodeling strategy using recursive decomposition coupled with sequential sampling method," *Structural and Multidisciplinary Optimization*, vol. 63, no. 1, pp. 375–390, 2021.

[77] W. Li, G. Lin, and B. Li, “Inverse regression-based uncertainty quantification algorithms for high-dimensional models: Theory and practice,” *Journal of Computational Physics*, vol. 321, pp. 259–278, 2016.

[78] R. Tripathy and I. Bilionis, “Deep active subspaces: A scalable method for high-dimensional uncertainty propagation,” in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 59179, American Society of Mechanical Engineers, V001T02A074.

[79] T. Zhou and Y. Peng, “Structural reliability analysis via dimension reduction, adaptive sampling, and monte carlo simulation,” *Structural and Multidisciplinary Optimization*, pp. 1–23, 2020.

[80] L. Condra, *Reliability improvement with design of experiment.* Crc Press, 2001.

[81] Q. Pan and D. Dias, “Sliced inverse regression-based sparse polynomial chaos expansions for reliability analysis in high dimensions,” *Reliability Engineering & System Safety*, vol. 167, pp. 484–493, 2017.

[82] R. Tripathy, I. Bilionis, and M. Gonzalez, “Gaussian processes with built-in dimensionality reduction: Applications to high-dimensional uncertainty propagation,” *Journal of Computational Physics*, vol. 321, pp. 191–223, 2016.

[83] P. Jiang, S. Missoum, and Z. Chen, “Optimal svm parameter selection for non-separable and unbalanced datasets,” *Structural and Multidisciplinary Optimization*, vol. 50, no. 4, pp. 523–535, 2014.

[84] G. H. Dunteman, *Principal components analysis.* Sage, 1989.

[85] F. B. Bryant and P. R. Yarnold, “Principal-components analysis and exploratory and confirmatory factor analysis,” in *Reading and understanding multivariate statistics.* Washington, DC, US: American Psychological Association, 1995, pp. 99–136.

[86] X. Yu and X. Du, “Reliability-based multidisciplinary optimization for aircraft wing design,” *Structures and Infrastructure Engineering*, vol. 2, no. 3-4, pp. 277–289, 2006.

[87] H. E. Daniels, “Saddlepoint approximations in statistics,” *The Annals of Mathematical Statistics*, pp. 631–650, 1954.

[88] R. Lugannani and S. Rice, “Saddle point approximation for the distribution of the sum of independent random variables,” *Advances in Applied Probability*, vol. 12, no. 2, pp. 475–490, 1980.

[89] X. Du and W. Chen, “A most probable point-based method for efficient uncertainty analysis,” *Journal of Design and Manufacturing Automation*, vol. 4, no. 1, pp. 47–66, 2001.

[90] Z. Zhang, C. Jiang, X. Han, and X. Ruan, “A high-precision probabilistic uncertainty propagation method for problems involving multimodal distributions,” *Mechanical Systems and Signal Processing*, vol. 126, pp. 21–41, 2019.

[91] J. Yin and X. Du, “High-dimensional reliability method accounting for important and unimportant input variables,” *Journal of Mechanical Design*, pp. 1–28, 2021.

[92] X. Huang, J. Chen, and H. Zhu, “Assessing small failure probabilities by AK–SS: An active learning method combining kriging and subset simulation,” *Structural Safety*, vol. 59, pp. 86–95, 2016.

[93] Q. Pan and D. Dias, “An efficient reliability method combining adaptive support vector machine and monte carlo simulation,” *Structural Safety*, vol. 67, pp. 85–95, 2017.

[94] N. Lelièvre, P. Beaurepaire, C. Mattrand, and N. Gayton, “AK-MCSi: A kriging-based method to deal with small failure probabilities and time-consuming models,” *Structural Safety*, vol. 73, pp. 1–11, 2018.

[95] R. Teixeira, M. Nogal, and A. O’Connor, “Adaptive approaches in metamodel-based reliability analysis: A review,” *Structural Safety*, vol. 89, p. 102 019, 2021.

[96] J. E. Ramirez-Marquez and D. W. Coit, “A monte-carlo simulation approach for approximating multi-state two-terminal reliability,” *Reliability Engineering & System Safety*, vol. 87, no. 2, pp. 253–264, 2005.

[97] V. Dubourg, B. Sudret, and J.-M. Bourinet, “Reliability-based design optimization using kriging surrogates and subset simulation,” *Structural and Multidisciplinary Optimization*, vol. 44, no. 5, pp. 673–690, 2011.

[98] S.-K. Au and J. L. Beck, “A new adaptive importance sampling scheme for reliability calculations,” *Structural Safety*, vol. 21, no. 2, pp. 135–158, 1999.

[99] B. Echard, N. Gayton, M. Lemaire, and N. Relun, “A combined importance sampling and kriging reliability method for small failure probabilities with time-demanding numerical models,” *Reliability Engineering & System Safety*, vol. 111, pp. 232–240, 2013.

[100] K. Binder, D. Heermann, L. Roelofs, A. J. Mallinckrodt, and S. McKay, “Monte carlo simulation in statistical physics,” *Computers in Physics*, vol. 7, no. 2, pp. 156–157, 1993.

[101] P. W. Glynn and D. L. Iglehart, “Importance sampling for stochastic simulations,” *Management Science*, vol. 35, no. 11, pp. 1367–1392, 1989.

[102] L. Hawchar, C.-P. El Soueidy, and F. Schoefs, “Principal component analysis and polynomial chaos expansion for time-variant reliability problems,” *Reliability Engineering & System Safety*, vol. 167, pp. 406–416, 2017.

[103] J. Xu and D. Wang, “Structural reliability analysis based on polynomial chaos, voronoi cells and dimension reduction technique,” *Reliability Engineering & System Safety*, vol. 185, pp. 329–340, 2019.

[104] P. G. Constantine, E. Dow, and Q. Wang, “Active subspace methods in theory and practice: Applications to kriging surfaces,” *SIAM Journal on Scientific Computing*, vol. 36, no. 4, A1500–A1524, 2014.

[105] A. Chaudhuri, B. Kramer, and K. E. Willcox, “Information reuse for importance sampling in reliability-based design optimization,” *Reliability Engineering & System Safety*, p. 106 853, 2020.

[106] C. Lataniotis, S. Marelli, and B. Sudret, “Extending classical surrogate modeling to high dimensions through supervised dimensionality reduction: A data-driven approach,” *International Journal for Uncertainty Quantification*, vol. 10, no. 1, 2020.

[107] B. Schölkopf, A. Smola, and K.-R. Müller, “Kernel principal component analysis,” in *International conference on artificial neural networks*, Springer, pp. 583–588.

[108] Y. Peng, T. Zhou, and J. Li, “Surrogate modeling immersed probability density evolution method for structural reliability analysis in high dimensions,” *Mechanical Systems and Signal Processing*, vol. 152, p. 107 366, 2021.

[109] R. R. Coifman and S. Lafon, “Diffusion maps,” *Applied and computational harmonic analysis*, vol. 21, no. 1, pp. 5–30, 2006.

[110] H.-M. Wu, “Kernel sliced inverse regression with applications to classification,” *Journal of Computational and Graphical Statistics*, vol. 17, no. 3, pp. 590–610, 2008.

[111] S. Akaho, “A kernel method for canonical correlation analysis,” *arXiv preprint cs/0609071*, 2006.

[112] M. Aiserman, È. M. Braverman, and L. Rozonoer, “Theoretical foundations of the potential function method in pattern recognition,” *Avtomat. i Telemeh*, vol. 25, no. 6, pp. 917–936, 1964.

[113] V. Vapnik, *The nature of statistical learning theory*. Springer Science & Business Media, 2013.

[114] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*, Springer, 2003, pp. 63–71.

[115] K. M. Prasad and R. Bapat, "The generalized moore-penrose inverse," *Linear Algebra and its Applications*, vol. 165, pp. 59–69, 1992.

[116] C. Lataniotis, D. Wicaksono, S. Marelli, and B. Sudret, "Uqlab user manual – kriging (gaussian process modeling)," Chair of Risk, Safety and Uncertainty Quantification, Report UQLab-V1.4-105, 2021.

[117] D. D. Cox and S. John, "A statistical method for global optimization," in *1992 IEEE International Conference on Systems, Man, and Cybernetics*, IEEE, pp. 1241–1246.

[118] R. Rackwitz, "Reliability analysis—a review and some perspectives," *Structural Safety*, vol. 23, no. 4, pp. 365–395, 2001.

[119] J.-M. Bourinet, F. Deheeger, and M. Lemaire, "Assessing small failure probabilities by combined subset simulation and support vector machines," *Structural Safety*, vol. 33, no. 6, pp. 343–353, 2011.

[120] H. M. Gomes, "Truss optimization with dynamic constraints using a particle swarm algorithm," *Expert Systems with Applications*, vol. 38, no. 1, pp. 957–968, 2011.

[121] F. Ma, H. Zhang, A. Bockstedte, G. C. Foliente, and P. Paevere, "Parameter analysis of the differential model of hysteresis," *Journal of Applied Mechanics*, vol. 71, no. 3, pp. 342–349, 2004.

[122] Pacific Earthquake Engineering Research Center, *Time history data files from el centro site imperial valley irrigation district*, Available at http://www.vibrationdata.com/elcentro.htm (2021/09/01).

[123] S. Wojtkiewicz, M. Eldred, R. Field Jr, A. Urbina, and J. Red-Horse, "Uncertainty quantification in large computational engineering models," in *19th AIAA Applied Aerodynamics Conference*, 2001, p. 1455.

[124] C. Roy and W. Oberkampf, "A complete framework for verification, validation, and uncertainty quantification in scientific computing," in *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, 2010, p. 124.

[125] P. Acar and V. Sundararaghavan, "Uncertainty quantification of microstructural properties due to experimental variations," *AIAA Journal*, vol. 55, no. 8, pp. 2824–2832, 2017.

[126] H.-R. Bae, R. V. Grandhi, and R. A. Canfield, "Uncertainty quantification of structural response using evidence theory," *AIAA Journal*, vol. 41, no. 10, pp. 2062–2068, 2003.

[127] P. S. Palar, K. Zakaria, L. R. Zuhal, K. Shimoyama, and R. P. Liem, "Gaussian processes and support vector regression for uncertainty quantification in aerodynamics," in *AIAA SCITECH 2021 Forum*, 2021, p. 0181.

[128] B. D. Youn, K. K. Choi, and L. Du, "Enriched performance measure approach for reliability-based design optimization," *AIAA Journal*, vol. 43, no. 4, pp. 874–884, 2005.

[129] M. Papila and R. Haftka, "Uncertainty and response surface approximations," in *19th AIAA Applied Aerodynamics Conference*, 2001, p. 1680.

[130] J. Weinmeister, X. Gao, and S. Roy, "Analysis of a polynomial chaos-kriging metamodel for uncertainty quantification in aerodynamics," *AIAA Journal*, vol. 57, no. 6, pp. 2280–2296, 2019.

[131] T. W. Simpson, T. M. Mauery, J. J. Korte, and F. Mistree, "Kriging models for global approximation in simulation-based multidisciplinary design optimization," *AIAA Journal*, vol. 39, no. 12, pp. 2233–2241, 2001.

[132] B. J. Bichon, M. S. Eldred, L. P. Swiler, S. Mahadevan, and J. M. McFarland, "Efficient global reliability analysis for nonlinear implicit performance functions," *AIAA Journal*, vol. 46, no. 10, pp. 2459–2468, 2008.

[133] J. Yin and X. Du, "Active learning with generalized sliced inverse regression for high-dimensional reliability analysis," *Structural Safety*, vol. 94, p. 102 151, 2022.

[134] A. Hadidi, B. F. Azar, and A. Rafiee, "Efficient response surface method for high-dimensional structural reliability analysis," *Structural Safety*, vol. 68, pp. 15–27, 2017.

[135] C. Bucher, "Asymptotic sampling for high-dimensional reliability analysis," *Probabilistic Engineering Mechanics*, vol. 24, no. 4, pp. 504–510, 2009.

[136] T. Zhou and Y. Peng, "Active learning and active subspace enhancement for pdem-based high-dimensional reliability analysis," *Structural Safety*, vol. 88, p. 102 026, 2021.

[137] A. B. Lambe and J. R. Martins, "Matrix-free aerostructural optimization of aircraft wings," *Structural and Multidisciplinary Optimization*, vol. 53, no. 3, pp. 589–603, 2016.

[138] A. Rosebrock, *Keras: Multiple inputs and mixed data*, Available at https://www.pyimagesearch.com/2019/02/04/keras-multiple-inputs-and-mixed-data/ (2021/02/15).

[139] A. Sharma, E. Vans, D. Shigemizu, K. A. Boroevich, and T. Tsunoda, "Deepinsight: A methodology to transform a non-image data to an image for convolution neural network architecture," *Scientific reports*, vol. 9, no. 1, pp. 1–7, 2019.

[140] P. D. Arendt, D. W. Apley, and W. Chen, “Quantification of model uncertainty: Calibration, model discrepancy, and identifiability,” *Journal of Mechanical Design*, vol. 134, no. 10, 2012.

[141] S. Bae, C. Park, and N. H. Kim, “Estimating effect of additional sample on uncertainty reduction in reliability analysis using gaussian process,” *Journal of Mechanical Design*, vol. 142, no. 11, 2020.

[142] J. Bradshaw, A. G. d. G. Matthews, and Z. Ghahramani, “Adversarial examples, uncertainty, and transfer testing robustness in gaussian process hybrid deep networks,” *arXiv preprint arXiv:1707.02476*, 2017.

[143] D. Rajaram, T. G. Puranik, A. Renganathan, *et al.*, “Deep gaussian process enabled surrogate models for aerodynamic flows,” in *AIAA SCITECH 2020 Forum*, 2020, p. 1640.

[144] P. Satria Palar, L. Rizki Zuhal, and K. Shimoyama, “Gaussian process surrogate model with composite kernel learning for engineering design,” *AIAA Journal*, vol. 58, no. 4, pp. 1864–1880, 2020.

[145] X. Wei and X. Du, “Robustness metric for robust design optimization under time- and space-dependent uncertainty through metamodeling,” *Journal of Mechanical Design*, vol. 142, no. 3, 2019.

[146] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[147] S. N. Lophaven, H. B. Nielsen, J. Søndergaard, *et al.*, *DACE: a Matlab kriging toolbox*. Citeseer, 2002, vol. 2.

[148] I. Steinwart and A. Christmann, *Support vector machines*. Springer Science & Business Media, 2008.

VITA

Jianhua Yin was born in Shuozhou, Shanxi Province, People's Republic of China. In 2014 and 2017, he earned his Bachelor's and Master's degree in Geological Engineering from Chang'an University, Xi'an, and Nanjing University, Nanjing, respectively. In Jan 2018, he was enrolled in the Ph.D. program of Geotechnical Engineering at Missouri University of Science and Technology and was focusing on the study of multi-physics simulation. Then, he transferred to Purdue University to pursue his Ph.D. degree in Mechanical Engineering in May 2019. His research interests include dimension reduction, machine learning, uncertainty quantification, and design optimization. In May 2022, he received his Ph.D. degree in Mechanical Engineering from Purdue University, West Lafayette, Indiana, USA.

PUBLICATIONS

Journal Papers

Yin, J., Du, X., "Uncertainty Quantification by Convolutional Neural Network Gaussian Process Regression with Image and Numerical Data," In preparation for submission.

Shen, D., **Yin, J.**, Li, L., Du, X., "Comparative Study of Heterogeneous Vehicle Platoon via Distributed Model Predictive Control and Distributed H-infinity State Feedback Control," In preparation for submission.

Yin, J., Shen, D., Du, X., and Li, L., "Distributed Stochastic Model Predictive Control with Taguchi's Robustness for Vehicle Platooning," *IEEE Transactions on Intelligent Transportation System*, 2022.

Yin, J., and Du, X., "Active Learning with Generalized Sliced Inverse Regression for High-dimensional Reliability Analysis," *Structural Safety*, 2022.

Yin, J., and Du, X., "High-Dimensional Reliability Method Accounting for Important and Unimportant Input Variables," *ASME Journal of Mechanical Design*, 2022.

Yin, J., and Du, X., "A Safety Factor Method for Reliability-Based Component Design," *ASME Journal of Mechanical Design*, 2021.

Conference Papers

Du, X., and **Yin, J.**, "Teaching Machine Learning for Analysis and Design to Mechanical Engineering Students," Accepted by *ASME 2022 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference.*

Li, H., **Yin, J.**, Du, X., "Uncertainty Quantification of Physics-based Label-free Deep Learning and Probabilistic Prediction of Extreme Events," Accepted by *ASME 2022 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference.*

Yin, J., Du, X., "Uncertainty Quantification by Convolutional Neural Network Gaussian Process Regression with Image and Numerical Data," *AIAA SciTech 2022 Forum*, 2022.

Li, H., **Yin, J.**, Du, X., “Label Free Uncertainty Quantification,” *AIAA SciTech 2022 Forum*, 2022.

Shen, D., **Yin, J.***, Du, X., and Li, L., “Distributed Nonlinear Model Predictive Control for Heterogeneous Vehicle Platoons Under Uncertainty,” *In 2021 IEEE 24th International Conference on Intelligent Transportation Systems (ITSC)*, Indianapolis, IN, USA, Sep. 19-22, 2021. (* denotes equal contribution)

Yin, J., and Du, X., “High-Dimensional Reliability Method Accounting for Important and Unimportant Input Variables,” *Proc. ASME 2021 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Virtual, Aug. 17-19, 2021.

Yin, J., and Du, X., “A Practical Safety Factor Method for Reliability-Based Component Design,” *Proc. ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Virtual, Aug. 17-19, 2020.