
Doctoral Dissertations

Student Theses and Dissertations

Fall 2020

Time- and space-dependent uncertainty analysis and its application in lunar plasma environment modeling

Xinpeng Wei

Follow this and additional works at: https://scholarsmine.mst.edu/doctoral_dissertations



Part of the [Mechanical Engineering Commons](#)

Department: Mechanical and Aerospace Engineering

Recommended Citation

Wei, Xinpeng, "Time- and space-dependent uncertainty analysis and its application in lunar plasma environment modeling" (2020). *Doctoral Dissertations*. 2945.

https://scholarsmine.mst.edu/doctoral_dissertations/2945

This thesis is brought to you by Scholars' Mine, a service of the Missouri S&T Library and Learning Resources. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

TIME- AND SPACE-DEPENDENT UNCERTAINTY ANALYSIS AND ITS
APPLICATION IN LUNAR PLASMA ENVIRONMENT MODELING

by

XINPENG WEI

A DISSERTATION

Presented to the Graduate Faculty of the
MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

MECHANICAL ENGINEERING

2020

Approved by:

Dr. Daoru Han, Advisor
Dr. Xiaoping Du
Dr. K. Chandrashekhara
Dr. Serhat Hosder
Dr. Xiaoming He

© 2020

Xinpeng Wei

All Rights Reserved

PUBLICATION DISSERTATION OPTION

This dissertation consists of the following five articles, formatted in the style used by the Missouri University of Science and Technology:

Paper I, found on Pages 9-52, has been published in *Structural Safety*.

Paper II, found on Pages 53-79, has been submitted to *Journal of Computing and Information Science in Engineering*.

Paper III, found on Pages 80-104, has been published in *Journal of Mechanical Design*.

Paper IV, found on Pages 105-141, has been published in *Journal of Mechanical Design*.

Paper V, found on Pages 142-177, has been accepted to *Journal of Verification, Validation, and Uncertainty Quantification*.

ABSTRACT

During an engineering system design, engineers usually encounter uncertainties that ubiquitously exist, such as material properties, dimensions of components, and random loads. Some of these parameters do not change with time or space and hence are time- and space-independent. However, in many engineering applications, the more general time- and space-dependent uncertainty is frequently encountered. Consequently, the system exhibits random time- and space-dependent behaviors, which may result in a higher probability of failure, lower average lifetime, and/or worse robustness. Therefore, it is critical to quantify uncertainty and predict how the system behaves under time- and space-dependent uncertainty. The objective of this study is to develop accurate and efficient methods for uncertainty analysis. This study contains five works. In the first work, an accurate method based on the series expansion, Gauss-Hermite quadrature, and saddle point approximation is developed to calculate high-dimensional normal probabilities. Then the method is applied to estimate time-dependent reliability. In the second work, we develop an adaptive Kriging method to estimate product average lifetime. In the third work, a time- and space-dependent reliability analysis method based on the first-order and second-order methods is proposed. In the fourth work, we extend the existing robustness analysis to time- and space-dependent problems and develop an adaptive Kriging method to evaluate the time- and space-dependent robustness. In the fifth work, we develop an adaptive Kriging method to efficiently estimate the lower and upper bounds of the electric potentials of the photoelectron sheaths near the lunar surface.

ACKNOWLEDGMENTS

This dissertation was supported by the National Science Foundation under Grant No. 1923799 (formerly 1727329) and the Intelligent System Center, which are gratefully acknowledged.

I would like to express my sincere and deep appreciation to my current advisor, Dr. Daoru Han, and my former advisor, Dr. Xiaoping Du, for their valuable guidance, support, and encouragement during my Ph.D. study at Missouri University of Science and Technology. It has been my great honor to work with them. They taught me diligence and a rigorous attitude towards research. I have been benefiting from that a lot and will continue benefiting in my future research career.

I would like to extend my appreciation to my committee members, Dr. K. Chandrashekhara, Dr. Serhat Hosder, and Dr. Xiaoming He. Their valuable comments and guidance helped me a lot in completing this dissertation.

Besides, I would like to thank my labmates and friends, Dr. Zhen Hu, Dr. Zhengwei Hu, Dr. Zhangli Hu, Mr. Hao Wu, Mr. Jianxun Zhao, Mr. Joshua Burch, Mr. Terence McGarvey, Mr. David Lund, Mr. Blake Folta, Mr. Ross Haselhorst, and Ms. Sheri Haselhorst, ect., for their support and help.

Last but not least, I would like to express my deepest love and appreciation to my wife Jiaojiao Li, my son Leo Wei, my parents, parents-in-law, and sisters. They let my life be full of love, hope, and happiness!

TABLE OF CONTENTS

	Page
PUBLICATION DISSERTATION OPTION	iii
ABSTRACT.....	iv
ACKNOWLEDGMENTS	v
LIST OF ILLUSTRATIONS	xii
LIST OF TABLES	xv
 SECTION	
1. INTRODUCTION.....	1
1.1. BACKGROUND	1
1.1.1. Reliability Analysis.....	2
1.1.2. Robustness Analysis.....	4
1.2. RESEARCH OBJECTIVE	6
1.3. ORGANIZATION OF THE DISSERTATION	7
 PAPER	
I. APPROXIMATION TO MULTIVARIATE NORMAL INTEGRAL AND ITS APPLICATION IN TIME-DEPENDENT RELIABILITY ANALYSIS.....	9
ABSTRACT.....	9
1. INTRODUCTION.....	10
2. PROBLEM STATEMENT	12
3. REVIEW OF EXISTING METHODS.....	14
3.1. CRUDE MCS	14

3.2. SEQUENTIAL CONDITIONED IMPORTANCE SAMPLING (SCIS) METHOD.....	15
3.3. RANDOMIZED QUASI MCS.....	16
3.4. EQUIVALENT COMPONENT METHODS.....	17
4. OVERVIEW OF THE PROPOSED METHOD	18
5. FORMULATION OF THE PROPOSED METHOD	22
5.1. STEP 1: SCREENING RANDOM VARIABLES	22
5.2. STEP 2: SERIES EXPANSION WITH EOLE	24
5.3. STEP 3: CALCULATE MGF WITH GAUSS-HERMITE QUADRATURE ..	26
5.4. STEP 4: TRANSFORM MGF TO CDF USING SPA	29
6. APPLICATION IN TIME-DEPENDENT RELIABILITY ANALYSIS	31
7. NUMERICAL EXPERIMENTS.....	34
7.1. EXAMPLE 1: A MATH EXAMPLE WITH EXACT SOLUTION	35
7.2. EXAMPLE 2: A MATH EXAMPLE WITHOUT AN EXACT SOLUTION ..	38
7.3. EXAMPLE 3: A SLIDER-CRANK MECHANISM.....	40
7.4. EXAMPLE 4: A 52-BAR SPACE TRUSS	43
8. CONCLUSIONS.....	46
ACKNOWLEDGEMENTS	47
REFERENCES.....	47
II. PHYSICS-BASED GAUSSIAN PROCESS METHOD FOR PREDICTING AVERAGE PRODUCT LIFETIME IN DESIGN STAGE	53
ABSTRACT	53
1. INTRODUCTION.....	53

2. PROBLEM STATEMENT	56
3. INTRODUCTION TO GAUSSIAN PROCESS MODEL AND THE LEARNING FUNCTION U	57
4. THE PROPOSED METHOD.....	58
4.1. OVERVIEW OF THE PROPOSED METHOD.....	58
4.2. DESIGN OF EXPERIMENTS FOR INITIAL SURROGATE MODEL	60
4.3. ADAPTIVE TRAINING	61
4.4. ADAPTIVE SAMPLE SIZE	64
4.5. IMPLEMENTATION.....	66
5. EXTENSION TO PROBLEMS WITH INPUT RANDOM PROCESSES	67
6. EXAMPLES	69
6.1. EXAMPLE 1: A MATH EXAMPLE.....	70
6.2. EXAMPLE 2: A SIMPLY SUPPORTED BEAM	72
6.3. EXAMPLE 3: A 52-BAR SPACE TRUSS	74
7. CONCLUSIONS	76
ACKNOWLEDGEMENTS	77
REFERENCES.....	77
III. UNCERTAINTY ANALYSIS FOR TIME- AND SPACE-DEPENDENT RESPONSES WITH RANDOM VARIABLES	80
ABSTRACT.....	80
1. INTRODUCTION.....	80
2. PROBLEM STATEMENT	84
3. OVERVIEW.....	85

4. EXTREME VALUE ANALYSIS AT AN INSTANT OF TIME.....	87
4.1. EXTREME VALUE ANALYSIS USING FORM.....	87
4.2. EXTREME VALUES ANALYSIS USING SORM	89
5. PROCEDURE	91
6. EXAMPLES.....	95
6.1. A MATH EXAMPLE.....	95
6.2. A SLIDER MECHANISM.....	97
7. CONCLUSIONS	99
ACKNOWLEDGEMENTS	100
REFERENCES.....	100
IV. ROBUSTNESS METRIC FOR ROBUST DESIGN OPTIMIZATION UNDER TIME- AND SPACE-DEPENDENT UNCERTAINTY THROUGH MODELING	105
ABSTRACT.....	105
1. INTRODUCTION.....	105
2. REVIEW OF STATIC AND TIME-DEPENDENT ROBUSTNESS METRICS .	108
2.1. STATIC ROBUSTNESS METRIC.....	109
2.2. TIME-DEPENDENT ROBUSTNESS METRIC	109
3. A NEW ROBUSTNESS METRIC FOR TIME- AND SPACE-DEPENDENT QCS	110
4. A META-MODELING APPROACH TO ROBUSTNESS ANALYSIS	112
4.1. OVERVIEW OF THE PROPOSED ROBUSTNESS ANALYSIS	113
4.2. INITIAL TRAINING SET	114
4.3. EMPLOYMENT OF EGO	115

4.4. STOPPING CRITERIA	117
4.5. IMPLEMENTATION.....	120
4.6. EXTENSION TO PROBLEMS WITH INPUT RANDOM FIELDS	120
5. NUMERICAL EXAMPLES	124
5.1. A MATH PROBLEM.....	125
5.2. A SLIDER MECHANISM.....	130
5.3. A CANTILEVER BEAM.....	131
5.4. AN ELECTRON ACCELERATOR.....	133
6. CONCLUSIONS	135
ACKNOWLEDGEMENTS	136
REFERENCES.....	136
V. ADAPTIVE KRIGING METHOD FOR UNCERTAINTY QUANTIFICATION OF THE PHOTOELECTRON SHEATH AND DUST LEVITATION ON THE LUNAR SURFACE	142
ABSTRACT.....	142
1. INTRODUCTION.....	142
2. PROBLEM STATEMENT	145
2.1. 1-D PHOTOELECTRON SHEATH MODEL ON THE LUNAR SURFACE.....	145
2.2. DUST LEVITATION.....	146
2.3. OBJECTIVE.....	148
3. INTRODUCTION TO KRIGING MODEL AND AKM	150
3.1. OVERVIEW OF KRIGING MODEL.....	150
3.2. FORMULATION OF KRIGING MODEL	151

3.3. AN EXAMPLE OF KRIGING MODEL	152
3.4. AKM.....	152
4. THE PROPOSED METHOD.....	154
4.1. OVERVIEW OF THE PROPOSED METHOD.....	155
4.2. CANDIDATE SAMPLES AND INITIAL TRAINING SAMPLES	157
4.3. LEARNING FUNCTION.....	159
4.4. ERROR METRIC AND STOPPING CRITERION.....	162
4.5. IMPLEMENTATION.....	163
4.6. VALIDATION DISCUSSION.....	165
5. RESULTS.....	165
5.1. SHEATH PROFILE	166
5.2. DUST LEVITATION.....	170
6. CONCLUSIONS	173
ACKNOWLEDGMENT	174
REFERENCES.....	174
SECTION	
2. CONCLUSIONS	178
BIBLIOGRAPHY.....	181
VITA.....	188

LIST OF ILLUSTRATIONS

SECTION	Page
Figure 1.1. Organization of the dissertation.....	8
PAPER I	
Figure 1. Flowchart of SCIS	15
Figure 2. Random samples (left) and low-discrepancy samples (right)	17
Figure 3. Compounding procedures of the equivalent component method [42]	18
Figure 4. Functions that fully define the distribution of Z	19
Figure 5. Abstract flowchart of the proposed method	21
Figure 6. An example of the screening step.....	23
Figure 7. Variable screening for Example 2	38
Figure 8. A slider-crank mechanism [5]	41
Figure 9. Variable screening for Example 3	42
Figure 10. A 52-bar space truss [61].....	44
PAPER II	
Figure 1. A sample of the limit-state function	56
Figure 2. Brief flowchart of the proposed method.....	59
Figure 3. Detailed flowchart of the proposed method	66
Figure 4. Contours and training points	71
Figure 5. A simply supported beam [35]	72
Figure 6. A 52-bar truss [36].....	75

PAPER III

Figure 1. Flow chart of SSL.....	89
Figure 2. The procedure of updating $\beta(\mathbf{t})$ using SORM.....	90
Figure 3. Flow chart of the complete procedure.....	94
Figure 4. Probability of failure over different time intervals.....	96
Figure 5. A slider mechanism.....	97
Figure 6. Probability of failure over different time intervals.....	98

PAPER IV

Figure 1. Simplified flowchart.....	114
Figure 2. Detailed flowchart.....	123
Figure 3. Update of W	128
Figure 4. A slider mechanism [39].....	130
Figure 5. A cantilever beam.....	131
Figure 6. An electron accelerator.....	133

PAPER V

Figure 1. Three types of sheath potential profiles in the analytic 1-D photoelectron sheath model [2].....	145
Figure 2. A typical Type C sample of $E(Z; \mathbf{P})$	146
Figure 3. Method to solve for the equilibrium height of dust levitation.....	147
Figure 4. Original Kriging model: Prediction error is large near $x = 4$ and $x = 8$	153
Figure 5. Updated Kriging model with one more training sample added at $x = 8$: Overall prediction accuracy is improved significantly.....	154
Figure 6. Brief flowchart of AKM.....	155
Figure 7. Brief flowchart of the proposed method.....	156

Figure 8. Initial samples of $\phi(Z; \mathbf{P})$	168
Figure 9. Results by initial Kriging model: Predicted electric potential bounds are not accurate.....	169
Figure 10. Final result: Predicted electric potential bounds are accurate	170
Figure 11. Final result: Predicted electric field bounds are accurate	171
Figure 12. Dust levitation heights: The electric field bounds determines the dust levitation heights.....	172

LIST OF TABLES

PAPER I	Page
Table 1. Gauss-Hermite quadrature points and weights	27
Table 2. Modified Gauss-Hermite quadrature weights and points	28
Table 3. Results for Example 1	37
Table 4. Results for Case 1 of Example 2	39
Table 5. Results for Case 2 of Example 2	40
Table 6. Results for Case 3 of Example 2	40
Table 7. Variables and parameters of Example 3	42
Table 8. Results of Example 3	43
Table 9. Variables and parameters of Example 4	44
Table 10. Results for Example 4	45
PAPER II	
Table 1. Results of Example 1	72
Table 2. Variables of Example 2	73
Table 3. Results of Example 2	74
Table 4. Results of Example 3	76
PAPER III	
Table 1. Explanations for the complete procedure	93
Table 2. Probability of failure over different time intervals	96
Table 3. Probability of failure over different time intervals	98

PAPER IV

Table 1. Initial training points in hypercube space	125
Table 2. Initial training points in X-space and Z-space	126
Table 3. Added training points.....	127
Table 4. Robustness analysis results	128
Table 5. Robustness analysis results	129
Table 6. Robustness analysis results	131
Table 7. Robustness analysis results	133
Table 8. Robustness analysis results	135

PAPER V

Table 1. Variables of uncertainty	166
Table 2. Parameter values	166
Table 3. Samples generated by Hammersley sampling method	166
Table 4. Samples mapped into Ω	167
Table 5. Initial samples of \mathbf{P}	167
Table 6. Parameter values	170
Table 7. Parameters of Grains A and B	172
Table 8. Dust levitation heights: The proposed AKM obtained accurate levitation heights.....	173

SECTION

1. INTRODUCTION

1.1. BACKGROUND

Uncertainties widely exist in practical engineering. For example, there are uncertainties in material properties, dimensions of components, and random loads. Uncertainties can be roughly grouped into three categories: (a) static uncertainties, (b) time-dependent (TD) uncertainties, and (c) time- and space-dependent (TSD) uncertainties. Static uncertainties do not vary with time or space and are usually modeled as random variables. An example of static uncertainty is the manufacturing variation in dimensions. TD uncertainties vary with time and are usually modeled as random processes. Examples of TD uncertainties include wind loads and wave loads. TSD uncertainties vary with both time and space and are usually modeled as time-dependent random fields. An example is the river velocity. TSD uncertainties belong to the most general category since the other two categories are just special cases of the TSD category.

Subjected to the uncertain inputs, the output of a structure/product/system is generally also uncertain. Uncertainty analysis [1] is aimed at identifying the effect of uncertain inputs on the output. It is very important to do uncertainty analysis in a design stage to improve the design and avoid possible product failure as much as possible. Uncertainty analysis typically includes reliability analysis [2], robustness analysis [3], sensitivity analysis [4], and uncertainty propagation [5], etc. In this study, we mainly focus on reliability analysis and robustness analysis.

1.1.1. Reliability Analysis. Reliability is defined as the probability that a product performs its intended function over a specified period of time and under specified service conditions [6]. According to the involved uncertainties, reliability problems can be roughly grouped into three categories: (a) static problems, (b) TD problems, and (c) TSD problems.

Static reliability problems only involve static uncertainties. Reliability methods for static problems include, but are not limited to, analytical methods, surrogate model methods, moment methods, and simulation methods. Typical analytical methods include the first-order reliability method (FORM) and the second-order reliability method (SORM) [7-12]. FORM and SORM simplify a limit-state function, which specifies a functional relationship between a response and random input variables, using the first and second-order Taylor series expansions, respectively, at the so-called most probable point (MPP) [13]. FORM and SORM can obtain results with acceptable accuracy for many engineering problems and hence are widely used. However, when the limit-state functions are highly nonlinear, their accuracy may be poor. Surrogate model methods [14-16] use surrogate models, which are generally obtained from the design of experiments or variable screening using sensitivity analysis, to improve the computation efficiency. Surrogate model methods are generally efficient and if the surrogate models are well trained, accurate. Moment methods [13, 17] calculate the moments of the limit-state function and then approximate its distribution with the moments; and then the distribution is used to obtain the reliability. Usually, only the first four moments are used and poor accuracy may result if higher-order moments are also important. Simulation methods include the direct Monte Carlo simulation (MCS) [18], quasi-Monte Carlo simulation [19], importance sampling [20], and subset

simulation [21], etc. Usually, simulation methods are accurate but computationally expensive.

In TD problems, the limit-state function is a function of time and the inputs may or may not include TD uncertainties. For TD problems, many reliability methods are available, including upcrossing rate methods [22-24], surrogate model methods [25-28], simulation methods [29, 30], probability density evolution method [31], envelope function method [32], failure process decomposition-based method [33], and extreme value moment method [34]. Roughly speaking, upcrossing rate methods are the most dominant methods but the accuracy remains to be improved, surrogate methods can obtain accurate results if the surrogate models are well trained, and simulation methods are also accurate but computationally expensive.

TSD problems are the most general. The limit-state function is a function of both time and space and the inputs may or may not include TSD uncertainties. For TSD problems, only a few methods are available in the literature. Hu and Mahadevan [35, 36] developed a method based on adaptive surrogate modeling. Shi et al. [37] proposed two strategies. One strategy is combining the sparse grid technique with the fourth-moment method. And the other is combining the dimension reduction and maximum entropy method. Shi et al. [38] developed a transferred limit-state function technique to transform the TSD problem into a TSI counterpart. These methods still have limitations for wider applications. Efficiently and accurately dealing with TSD problems remains challenging.

In addition to the probability definition of reliability, average lifetime (or mean time to failure, MTTF) [39] is also used to quantify the reliability of a product. Statistics-based methods [40, 41] are widely used to estimate the MTTF. The methods need lifetime testing

on many products to obtain the lifetime samples, which are then used to estimate the average lifetime by statistical analysis. The methods are generally expensive and time-consuming since real products instead of numerical models are used for lifetime testing. Physics-based methods [18] then play an important role to deal with this problem. The methods use limit-state functions, which are computational models derived from physical principles, to predict the states of the components and subsystems of the product concerning potential failure modes [42]. With the computational models for the failure modes, physics-based methods are much more efficient than the statistics-based methods. They can predict reliability performance for a given design. However, efficient and accurate physics-based methods remain to be developed, when expensive limit-state functions are involved.

1.1.2. Robustness Analysis. Robust design optimization (RDO) [3] is an optimization design methodology for improving the quality of a product by minimizing the effect of the causes of variation without eliminating the causes [43]. It allows for the use of low-grade materials and reduces labor and material costs while improving reliability and reducing operating costs [43]. RDO has been used to improve product quality in industrial applications [44, 45]. Over the last three decades, it has gained much attention from many research fields, such as operations research [46-48], aerospace [49, 50], structural mechanics [51, 52], vibration control [53, 54], automobile [55-57], and fatigue analysis [58, 59]. Methods to solve RDO can be roughly grouped into three categories: probabilistic methods [60-62], deterministic methods [63-67], and metamodel-based methods [68-73]. Probabilistic methods perform robust optimization using the probability distributions of random variables. Deterministic methods incorporate a non-statistical index, such as the

gradient of a response, into the optimization problem to obtain a robust optimum [73]. Metamodel-based methods employ computationally cheap surrogate models to improve the efficiency of RDO.

Robustness analysis, which evaluates and predicts the robustness of a design, is repeated many times during RDO. Many metrics that measure the robustness exist in literature. The most common metric is Taguchi's quality loss function (QLF) [43]. This metric measures not only the distance between the average quality characteristics (QCs) and their targets but also the variation in the QCs [74]. There are also other robustness metrics, such as the signal-to-noise ratio [43], the percentile difference [75], and the worst-case QCs [76].

Most of the above robustness metrics are defined for static QCs that do not change over time and space. Some of the metrics could be used for dynamics problems, but they are only applicable for situations where the targets of QCs vary with signals [77, 78], instead of with time. To deal with problems involving time-dependent QCs, Goethals et al. [79] proposed to use the weighted sum of mean values of a QLF at discretized time instances to measure the robustness. The weighted-sum method, however, does not take into consideration of the autocorrelation of the time-dependent QLF, which is modeled as a stochastic process. To overcome this drawback, Du [74] proposed to use the maximum value of the time-dependent QLF to measure the time-dependent robustness and developed an MCS-based method to do time-dependent robustness analysis.

As for TSD problems, there is no method reported in the literature and there is a need to extend the static robustness and TD robustness to TSD robustness and develop corresponding methods to estimate the TSD robustness efficiently.

1.2. RESEARCH OBJECTIVE

The objective of this dissertation is to develop uncertainty analysis methods for TD problems and TSD problems and then apply uncertainty analysis to lunar plasma environment modeling. To achieve this objective, five research works are performed. The first three works deal with reliability analysis, the fourth work evaluates TSD robustness, and the last one applies uncertainty analysis to lunar plasma environment modeling.

Research work 1 is aimed at estimating high-dimensional normal probabilities and then applying it in TD reliability analysis. When FORM is used for TD reliability analysis, the limit-state function is transformed into an equivalent Gaussian process, which is then discretized into a large number of correlated Gaussian variables. A high-dimensional normal integral is therefore needed to calculate the TD reliability. An accurate method based on the combination of series expansion [80], Gaussian quadrature [81], and saddlepoint approximation [82, 83] is developed. The development of this method results in Paper I.

Research work 2 focuses on evaluating the average product lifetime, which is also a reliability metric of products. Currently, statistics-based methods [40, 41] are widely used to estimate the MTTF. The methods need lifetime testing on many products to obtain the lifetime samples, which are then used to estimate the average lifetime by statistical analysis. The methods are generally expensive and time-consuming since real products instead of numerical models are used for lifetime testing. A physics-based method is developed in this work. It uses limit-state functions derived from physical principles, to predict the states of the components and subsystems of the product concerning potential failure modes [42].

Without lifetime testing of real products, the proposed method is much more efficient. Details of this work are reported in Paper II.

Research work 3 is aimed at evaluating TSD reliability. For static and TD reliability problems, there are a lot of methods reported in the literature. However, for TSD problems, which are the most general, efficient, and accurate methods remain to be developed. In this work, a method based on FORM and SORM is developed to estimate TSD reliability efficiently. Details of this work are given in Paper III.

Research work 4 deals with TSD robustness analysis. For TSD robustness problems, few methods have been proposed so far. In this work, an adaptive training method based on the Kriging model [84] is proposed to estimate the TSD robustness accurately and efficiently. Details of this work are given in Paper IV.

Research work 5 applies the uncertainty analysis to lunar plasma environment modeling and is aimed at efficiently estimating the lower and upper bounds of the electric potentials of the photoelectron sheaths near the lunar surface [85]. To avoid evaluating the expensive black-box function for a large number of times, an adaptive training method based on the Kriging model is developed. Details of this work are given in Paper V.

This study is expected to help engineers better understand the effect of uncertainties on a product and then improve the reliability and/or robustness of the product at the design stage. If successful, it will contribute to higher product quality with reduced lifecycle costs.

1.3. ORGANIZATION OF THE DISSERTATION

The organization of this dissertation is shown in Figure 1.1. In the first paper, an accurate method based on the series expansion, Gauss-Hermite quadrature, and saddle

point approximation is developed to calculate high-dimensional normal probability. Then the method is applied to estimate structural time-dependent reliability. In the second paper, we develop an adaptive Kriging method to estimate product average lifetime. In the third paper, a time- and space-dependent reliability analysis method based on the widely used first-order and second-order methods is proposed. In the fourth paper, we extend the existing robustness analysis to time- and space-dependent problems and develop an adaptive Kriging method to efficiently evaluate the time- and space-dependent robustness. In the fifth paper, we apply the uncertainty analysis to lunar plasma environment modeling and develop an adaptive Kriging method to efficiently estimate the lower and upper bounds of the electric potentials of the photoelectron sheaths near the lunar surface.

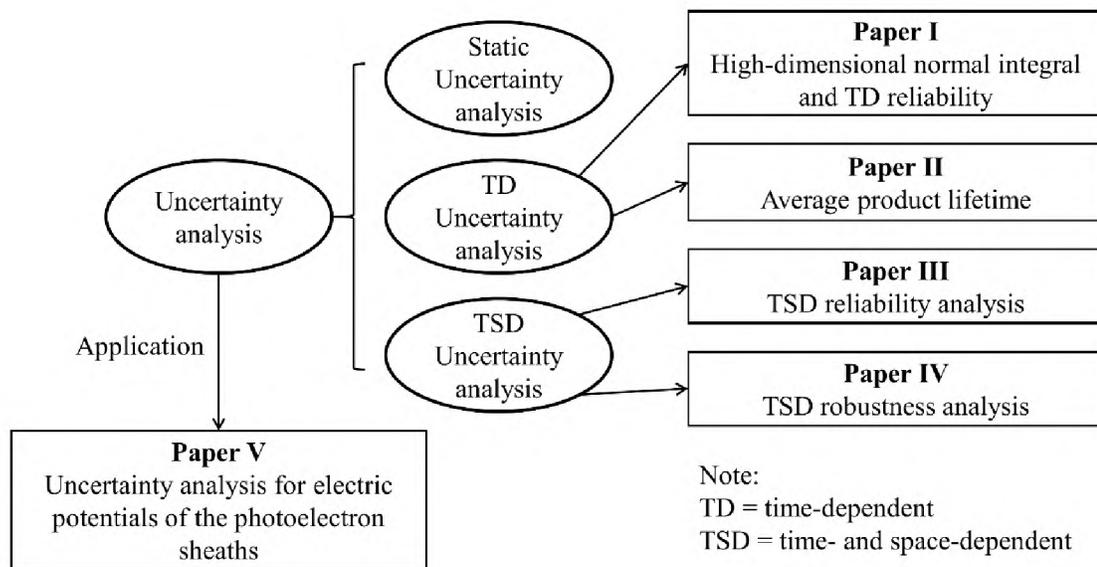


Figure 1.1. Organization of the dissertation

PAPER

I. APPROXIMATION TO MULTIVARIATE NORMAL INTEGRAL AND ITS APPLICATION IN TIME-DEPENDENT RELIABILITY ANALYSIS

ABSTRACT

It is common to evaluate high-dimensional normal probabilities in many uncertainty-related applications such as system and time-dependent reliability analysis. An accurate method is proposed to evaluate high-dimensional normal probabilities, especially when they reside in tail areas. The normal probability is at first converted into the cumulative distribution function of the extreme value of the involved normal variables. Then the series expansion method is employed to approximate the extreme value with respect to a smaller number of mutually independent standard normal variables. The moment generating function of the extreme value is obtained using the Gauss-Hermite quadrature method. The saddlepoint approximation method is finally used to estimate the cumulative distribution function of the extreme value, thereby the desired normal probability. The proposed method is then applied to time-dependent reliability analysis where a large number of dependent normal variables are involved with the use of the First Order Reliability Method. Examples show that the proposed method is generally more accurate and robust than the widely used randomized quasi Monte Carlo method and equivalent component method.

1. INTRODUCTION

Many uncertainty-related applications require the evaluation of multivariate normal probabilities, for instance, the system reliability analysis [1-3] and time-dependent reliability analysis [4-26]. Both analyses predict the reliability by integrating a multivariate normal density in the safe region if the First Order Reliability Method (FORM) [27] is employed. Other areas requiring a multivariate normal probability include the extreme value distribution [28], multivariate probit model [29], multiple comparisons [30], and multiple ordinal response models [31].

No methods exist for the exact computation of the multivariate normal probability, and many numerical and sampling methods have been developed to produce approximations [32]. The existing methods can be roughly grouped into two categories: random methods and deterministic methods.

Random methods generate a large number of samples of the involved random variables and then calculate the probability based on the statistical information of the samples. The most straightforward method is the crude Monte Carlo simulation (MCS) [33]. Other random methods are more or less based on the crude MCS. They include the quasi MCS [34, 35], the importance sampling method [36-38], the subset simulation method [39], and the Bayesian MCS [40]. The random methods are generally robust, easy to use, and accurate if the sample size is large enough. But they also have some shortcomings. First, samples are usually generated randomly and hence the result is not deterministic, resulting in unrepeatable results when different seed numbers, software, or computer platforms are used. Second, they are inefficient to estimate a small probability.

This makes reliability analysis difficult since engineering applications usually require a low probability of failure or high reliability. Note that some advanced random methods, such as the importance sampling method [36-38] and the subset simulation method [39], can get over this shortcoming to some extent.

Deterministic methods do not need random sampling. The equivalent component methods [41-43] are widely used. They sequentially compound two components, i.e., two of the involved normal variables, into an equivalent one, and the multivariate normal probability is eventually estimated by a univariate normal probability. The methods differ from one another mainly in the way of evaluating the correlation coefficients between the equivalent component and the other components. The correlation coefficients are determined by the sensitivity equivalency and the finite difference method [41, 43]. The finite difference method is replaced by an analytical approach [42], resulting in better accuracy and efficiency. The correlation coefficients can also be evaluated by conditional probabilities [3]. Generally, the equivalent component methods are efficient, even for high-dimensional problems. They may not be accurate when solving high-dimensional problems with small probabilities. One reason is that the equivalent component is not necessarily a normal variable and the error accumulates with the increase of the dimensionality. Besides, other deterministic methods are also available, including the first-order methods [44, 45], the product of conditional marginal probabilities [46, 47], and conditioning approximation methods [32, 48]. Their accuracy still needs to be improved when solving high-dimensional problems with small probabilities.

Overall, evaluating a multivariate normal probability is challenging in terms of accuracy and efficiency when the dimension is large and the probability is small. The

objective of this work is to develop a new method to improve both accuracy and efficiency. The proposed method involves the integration of dimension reduction, the expansion optimal linear estimation (EOLE) [49], the Gauss-Hermite quadrature method [50], and the saddlepoint approximation (SPA) [51, 52]. The proposed method is then applied to and evaluated by the time-dependent reliability analysis with a large number of dependent normal variables and small probabilities.

The remaining parts of the paper are organized as follows. Section 2 gives the problem statement. Section 3 reviews the existing methods. An overview of the proposed method is given in Section 4, followed by the detailed formulations in Section 5. Section 6 gives the application to time-dependent reliability analysis. Four examples are given in Section 7 to demonstrate the proposed method. Section 8 provides conclusions.

2. PROBLEM STATEMENT

Suppose \mathbf{Y} is a vector of N normal random variables with the mean vector $\boldsymbol{\mu}$ and the correlation matrix $\boldsymbol{\Sigma}$. The joint probability density function (PDF) $f_{\mathbf{Y}}(\mathbf{y})$ of \mathbf{Y} is given by

$$f_{\mathbf{Y}}(\mathbf{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^N |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{y} - \boldsymbol{\mu})\right) \quad (1)$$

The cumulative distribution function (CDF) $F_{\mathbf{Y}}(\hat{\mathbf{y}}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ of \mathbf{Y} is given by

$$F_{\mathbf{Y}}(\hat{\mathbf{y}}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \int_{-\infty}^{\hat{\mathbf{y}}} f_{\mathbf{Y}}(\mathbf{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) d\mathbf{y} \quad (2)$$

Note that Eq. (2) shows an N -dimensional integral.

Without losing generality, we assume that $\hat{\mathbf{y}} = \mathbf{0}$. We also assume that all components of \mathbf{Y} have a variance of 1. Then we only focus on calculating the following integral

$$F_Y(\mathbf{0}; \boldsymbol{\mu}, \mathbf{C}) = \int_{-\infty}^{\mathbf{0}} f_Y(\mathbf{y}; \boldsymbol{\mu}, \mathbf{C}) d\mathbf{y} \quad (3)$$

where \mathbf{C} is the correlation coefficient matrix of \mathbf{Y} . A general problem can be solved by Eq. (3) using the following transformation

$$F_Y(\hat{\mathbf{y}}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = F_Y(\mathbf{0}; (\boldsymbol{\mu} - \hat{\mathbf{y}}) ./ \boldsymbol{\sigma}, \mathbf{C}) \quad (4)$$

where $\boldsymbol{\sigma}$ is the standard deviation vector of \mathbf{Y} , and the operator $./$ represents the elementwise division. $F_Y(\mathbf{0}; (\boldsymbol{\mu} - \hat{\mathbf{y}}) ./ \boldsymbol{\sigma}, \mathbf{C})$ shares the same format with $F_Y(\mathbf{0}; \boldsymbol{\mu}, \mathbf{C})$. Introducing the indicator function $I(\cdot)$, Eq. (4) is written as

$$F_Y(\mathbf{0}; \boldsymbol{\mu}, \mathbf{C}) = \int_{-\infty}^{+\infty} I(\mathbf{y} < \mathbf{0}) f_Y(\mathbf{y}; \boldsymbol{\mu}, \mathbf{C}) d\mathbf{y} \quad (5)$$

where $I(\mathbf{y} < \mathbf{0}) = \begin{cases} 1, & \mathbf{y} < \mathbf{0} \\ 0, & \text{otherwise} \end{cases}$

In practical applications, high dimensions are commonly encountered. For example, in system reliability analysis, dimensionality may be dozens or hundreds. Many existing methods require \mathbf{C} to be full-rank. However, a non-full-rank \mathbf{C} is also frequently encountered in engineering problems. The objective of the study is to calculate the high-dimensional normal probabilities with a \mathbf{C} being full-rank or not.

3. REVIEW OF EXISTING METHODS

In this section, we briefly review four commonly used methods: the crude MCS, the sequential conditioned importance sampling method (SCIS) [38], the randomized quasi MCS method [35], and the equivalent component method [42]. The first three are random methods while the last one is a deterministic method.

3.1. CRUDE MCS

Crude MCS is the origin of other random methods. It first randomly generates n_s samples of \mathbf{Y} using $f_Y(\mathbf{y}; \boldsymbol{\mu}, \mathbf{C})$ and then approximates Eq. (5) by

$$F_Y(\mathbf{0}; \boldsymbol{\mu}, \mathbf{C}) \approx \tilde{F} = \frac{1}{n_s} \sum_{k=1}^{n_s} I(\mathbf{y}^k < \mathbf{0}) \quad (6)$$

where \tilde{F} represents the approximation, and \mathbf{y}^k is the k^{th} sample of \mathbf{Y} . \tilde{F} itself is a random variable. Therefore, different runs of crudes MCS lead to different realizations of \tilde{F} . This is known as a random error. The variation coefficient V_{MCS} of \tilde{F} is used to measure the random error and is given by

$$V_{\text{MCS}} = \sqrt{\frac{1 - \tilde{F}}{n_s \tilde{F}}} \quad (7)$$

It shows that the convergence rate of crude MCS is $O(1/\sqrt{n_s})$ [35], which is independent of N . With this feature, crude MCS does not suffer from the curse of dimensionality. The convergence rate, however, is thought to be low. For example, if the exact value of $F_Y(\mathbf{0}; \boldsymbol{\mu}, \mathbf{C})$ is 10^{-5} and V_{MCS} is required to be no more than 10^{-2} , then according to Eq. (7), the sample size n_s must be at least about 10^9 .

Despite its low convergence rate, MCS is widely used and is specially treated as a benchmark method for an accuracy comparison study when an exact solution is not available.

3.2. SEQUENTIAL CONDITIONED IMPORTANCE SAMPLING (SCIS) METHOD

SCIS is based on the importance sampling method and makes use of the property that conditioned on given values of arbitrary components of \mathbf{Y} , the remaining components also follow (univariate or multivariate) normal distribution [38].

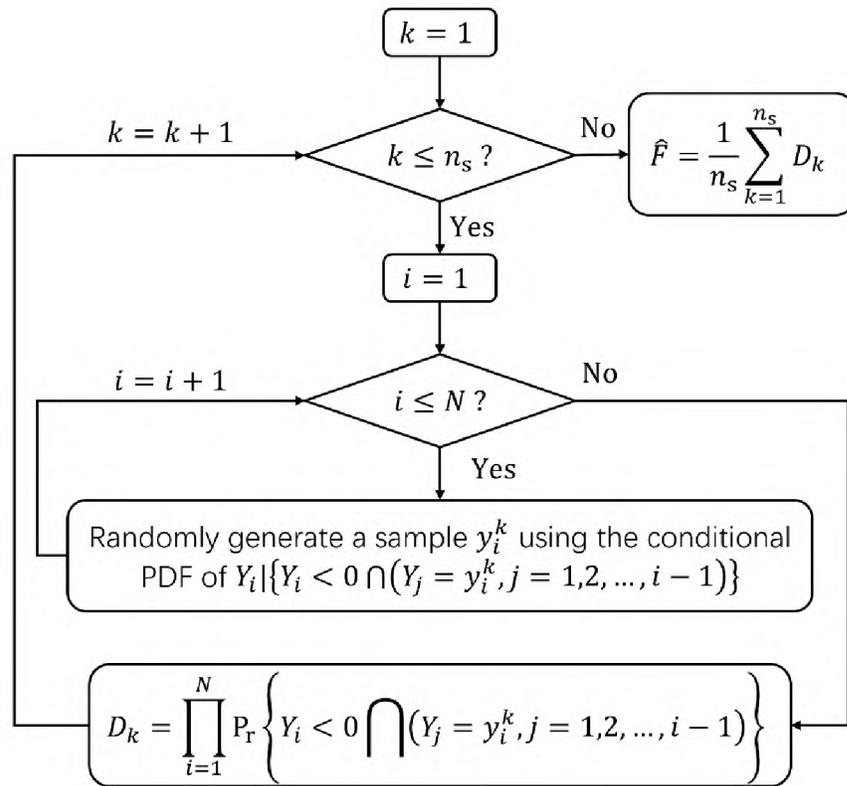


Figure 1. Flowchart of SCIS

The flowchart of SCIS is shown in Figure 1, where $P_r\{\cdot\}$ represents probability. Because of the property of multivariate normal variables, derivations of the conditional PDF and of D_k (in Figure 1) are obtained easily. More details are given in Ref. [38]. Similar to crude MCS, the approximation \tilde{F} calculated by SCIS is also a random variable, with its variation coefficient V_{SCIS} given by

$$V_{\text{SCIS}} = \frac{1}{n_s \tilde{F}} \sqrt{\sum_{k=1}^{n_s} (D_k - \tilde{F})^2} \quad (8)$$

Compared to Eq. (7), Eq. (8) shows that the convergence rate of SCIS is significantly better than that of crude MCS.

3.3. RANDOMIZED QUASI MCS

An effective way to improve the convergence rate of MCS is to replace the randomly generated samples by carefully selected, deterministic sequences of samples [35]. This approach is known as quasi MCS, and those samples are called low-discrepancy samples. Figure 2 shows 10^3 random samples and 10^3 low-discrepancy samples of \mathbf{Y} , given $\boldsymbol{\mu} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and $\mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. The low-discrepancy samples are generated by Halton sequences [53]. The low discrepancy samples are regularly even while the random samples have irregular clusters. The evenness improves the convergence rate of the quasi MCS.

A drawback of quasi MCS is that it is hard to estimate the error. To estimate the error with the way similar to crude MCS, the deterministic low-discrepancy samples are randomized and the randomized quasi MCS was developed [35, 54]. It is worth mentioning

that the quasi MCS developed by Genz and Bretz [35] is commonly used. This method has been coded into *pmvnorm()*, an R [55] function in R package *mvtnorm*.

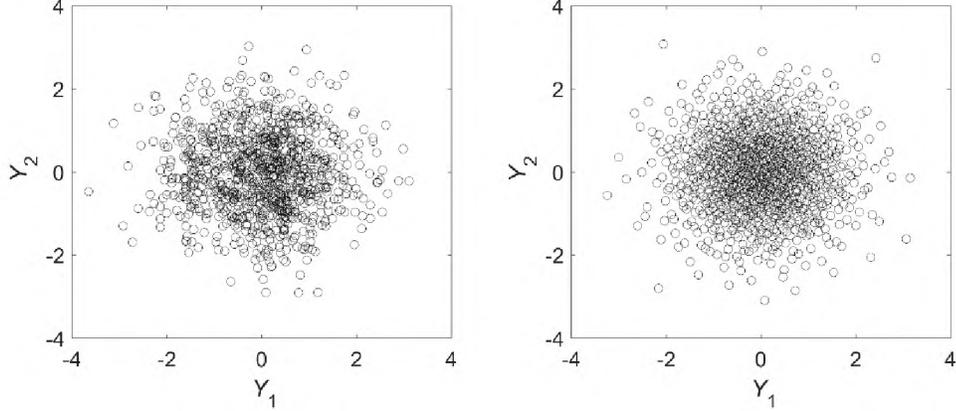


Figure 2. Random samples (left) and low-discrepancy samples (right)

3.4. EQUIVALENT COMPONENT METHODS

The equivalent component methods compound a pair of component normal variables into an equivalent normal variable sequentially so that the multivariate normal probability is finally estimated by a univariate normal probability. Figure 3 shows the compounding procedure. Y_{12}^e is the equivalent component obtained by compounding Y_1 and Y_2 . Then Y_{12}^e and Y_3 are compounded into Y_{123}^e . This process continues until N normal variables have been compounded into one equivalent normal variable $Y_{123\dots N}^e$. Eq. (3) is then approximated by

$$F_Y(\mathbf{0}; \boldsymbol{\mu}, \mathbf{C}) = \int_{-\infty}^0 f_e(y; \mu_e, \sigma_e^2) dy = \Phi\left(-\frac{\mu_e}{\sigma_e}\right) \quad (9)$$

where $f_e(y; \mu_e, \sigma_e^2)$, μ_e and σ_e^2 are the PDF, mean, and variance of $Y_{123\dots N}^e$, respectively, and $\Phi(\cdot)$ is the CDF of the standard normal variable.

The latest equivalent component method [42] has been proven to be effective for many problems. Assuming all the equivalent components to be normal variables, however, may introduce an unmeasurable error and hence compromise the accuracy of the method.

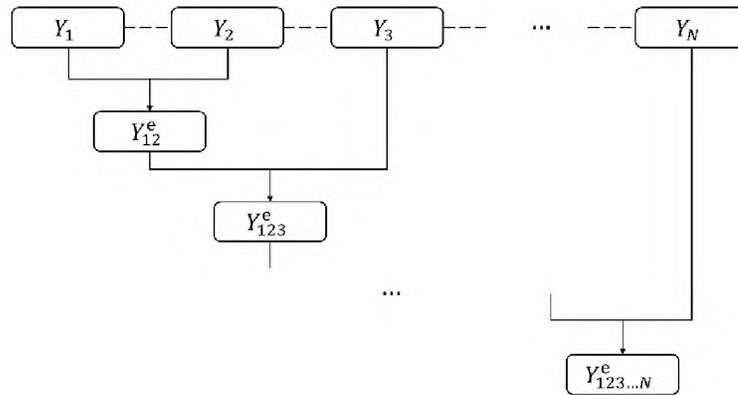


Figure 3. Compounding procedures of the equivalent component method [42]

4. OVERVIEW OF THE PROPOSED METHOD

The main idea of the proposed method is to convert the multidimensional probability into an equivalent extreme value probability. Eq. (10) is equivalent to

$$F_{\mathbf{Y}}(\mathbf{0}; \boldsymbol{\mu}, \mathbf{C}) = \Pr\left\{\bigcap_{i=1}^N Y_i < 0\right\} = \Pr\{\max(\mathbf{Y}) < 0\} = \Pr\{Z < 0\} = F_Z(0) \quad (10)$$

where $Z = \max(\mathbf{Y})$ is the maximum value of \mathbf{Y} . Note that Z itself is a random variable, and we denote it by $Z(\mathbf{Y})$ since it is a function of \mathbf{Y} .

The distribution of Z can be determined from its PDF $f_Z(z)$, CDF $F_Z(z)$, moment generating function (MGF) $M_Z(s)$, cumulant generating function (CGF) $K_Z(s)$, or characteristic function (CF) $C_Z(s)$. The relationships among those functions are shown in

Figure 4. A solid line means a theoretically rigorous transformation between the two functions connected by the line, while a dotted line means an approximate transformation. Theoretically, once any of the five functions is obtained, the other four can also be obtained using the transformation indicated above or below a line.

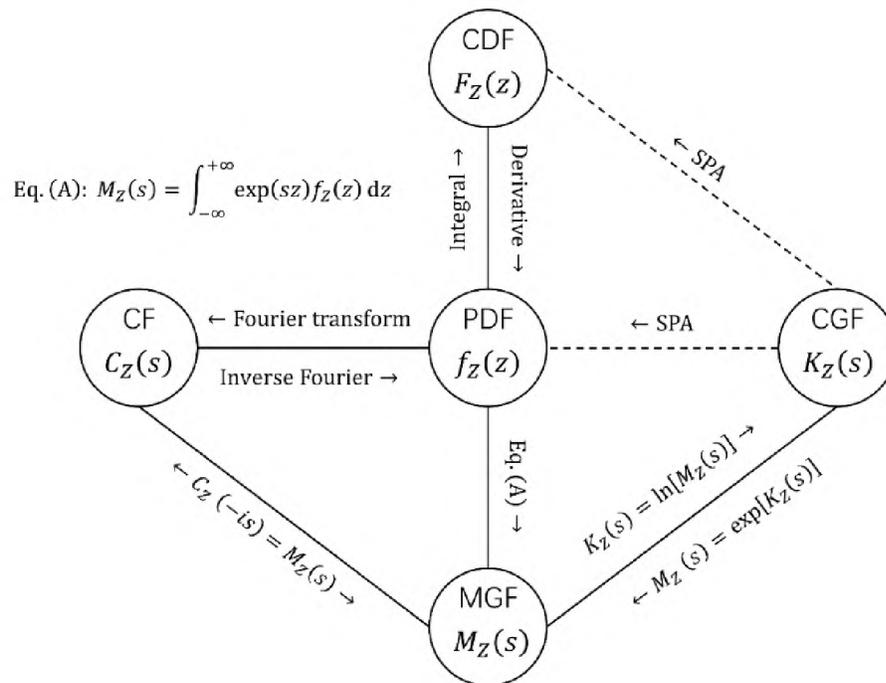


Figure 4. Functions that fully define the distribution of Z

The easiest starting point is the MGF given by

$$M_Z(s) = \int_{-\infty}^{+\infty} \exp(sz) f_Z(z) dz = \int_{-\infty}^{+\infty} \exp[sz(\mathbf{y})] f_Y(\mathbf{y}; \boldsymbol{\mu}, \mathbf{C}) d\mathbf{y} \quad (11)$$

Although Eq. (11) is also a high-dimensional integral similar to Eq. (5), it is much easier to calculate. The reason is that the integrand $\exp[sz(\mathbf{y})]$ is generally a continuous function, which can be calculated effectively using quadrature methods, while the integrand

$I(\mathbf{y} < \mathbf{0})$ in Eq. (5) is not. This is also the reason why we convert the multidimensional probability in Eq. (5) into the extreme value probability in Eq. (10).

Once $M_Z(s)$ is obtained, there are at least two ways to get $F_Z(z)$ or Eq. (10). As shown in Figure 4, the first way is $M_Z(s) \rightarrow C_Z(s) \rightarrow f_Z(z) \rightarrow F_Z(z)$ and the second way is $M_Z(s) \rightarrow K_Z(s) \rightarrow F_Z(z)$. The first way, however, is not practical, and there are two reasons. First, $M_Z(s)$ calculated by the quadrature method is not exact, and neither is $C_Z(s)$, which generally has complex output values and may suffer from large errors. Second, currently, there are no robust and effective methods to transform $C_Z(s)$ into $f_Z(z)$ using the inverse Fourier transform, especially when $C_Z(s)$ is not exact. In contrast, the second way is effective. The reason is that a simple logarithm is needed to obtain $K_Z(s)$ from $M_Z(s)$, and SPA is an accurate and efficient method to approximate $F_Z(z)$ from $K_Z(s)$, especially at the tails of $F_Z(z)$. Therefore, we use the latter way to calculate $F_Z(z)$.

Calculating Eq. (11), however, needs a heavy computational effort, since it may be a high-dimensional integral. To solve this problem, we propose two approaches to reduce the dimension of the integral. The first approach is to screen the random variables in \mathbf{Y} and remove the ones that barely contribute to the desired $F_Y(\mathbf{0}; \boldsymbol{\mu}, \mathbf{C})$. The second approach is to transform the integral from the \mathbf{Y} space, or physical space, into the eigenspace, using a truncated series expansion of \mathbf{Y} . With the latter approach, we can further reduce the dimension of the integral. This approach can usually reduce the dimension significantly because \mathbf{C} is a low-rank matrix in many engineering problems.

An abstract flowchart of the proposed method is given in Figure 5. Step 1 screens random variables in \mathbf{Y} . (See Subsection 5.1.) Note that after the screening step, we still use \mathbf{Y} for the remaining random variables for the convenience of presentation. In Step 2, we

use EOLE to expand \mathbf{Y} and then truncate the expansion to N' orders. This step transforms the integral in Eq. (11) from the \mathbf{Y} space into the \mathbf{U} space (the eigenspace). (See Subsection 5.2.) In Step 3, the Gauss-Hermite quadrature is used to calculate the MGF of Z in the eigenspace. (See Subsection 5.3.) In Step 4, SPA is employed to transform the MGF into CDF of Z , and finally the desired $F_Y(\mathbf{0}; \boldsymbol{\mu}, \mathbf{C})$ is obtained through Eq. (18). (See Subsection 5.4)

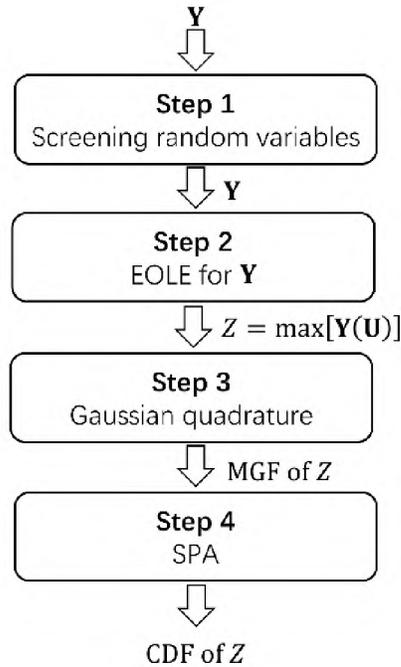


Figure 5. Abstract flowchart of the proposed method

There are four advantages of the proposed method. First, it can calculate multidimensional normal probabilities with arbitrary dimension N , as long as the dimension N' of the truncated eigenspace is not large. This is practical for dealing with engineering problems where the number of basic random variables is not large. Second,

the method is accurate even when calculating very small probabilities because SPA can recover CDF from MGF with sufficient accuracy, especially at tails of CDF. Third, it is generally more efficient than the abovementioned random methods, when $F_Y(\mathbf{0}; \boldsymbol{\mu}, \mathbf{C})$ is small, such as 10^{-5} and smaller. The reason is that random methods need a large sample size to guarantee accuracy when calculating small probabilities. Fourth, the result calculated by the proposed method is deterministic, instead of random by a random method.

5. FORMULATION OF THE PROPOSED METHOD

In this section, we give all details involved in the steps shown in Figure 5.

5.1. STEP 1: SCREENING RANDOM VARIABLES

The screening procedure is aimed at reducing the dimension by removing those components of \mathbf{Y} that are not important to $F_Y(\mathbf{0}; \boldsymbol{\mu}, \mathbf{C})$. If $\Pr\{Y_i < 0\}$ is almost equal to 1, or equivalently if $\Pr\{Y_i > 0\}$ is sufficiently small, then removing Y_i will not significantly affect the accuracy.

Since $\Pr\{Y_i > 0\}$ measures the importance of Y_i to $F_Y(\mathbf{0}; \boldsymbol{\mu}, \mathbf{C})$, the most important component Y_* is determined by

$$Y_* = \arg \max_i \Pr\{Y_i > 0\} \tag{12}$$

and $\Pr\{Y_* > 0\}$ is used as a benchmark to screen the other components of \mathbf{Y} . The screening criterion is given by

$$\Pr\{Y_i > 0\} < c\Pr\{Y_* > 0\} \quad (13)$$

where c is a hyperparameter taking a small value, such as 10^{-4} . Theoretically, the smaller is c , the higher accuracy will we have. However, if c is too small, the screening step will not effectively screen out unimportant normal random variables. If Y_i meets the criterion, it will be removed. Since \mathbf{Y} are normal variables, Eq. (13) is equivalent to

$$\Phi(\mu_i) < c\Phi(\mu_*) \quad (14)$$

where μ_* is the mean value of Y_* . Note that we have assumed in Section 2 that all components of \mathbf{Y} have a variance of 1, so Eq. (14) does not involve the variance of \mathbf{Y} . Figure 6 shows an example of the screening of \mathbf{Y} . Initially, there are $N = 300$ components in \mathbf{Y} . Only 68 components shown by the circles, however, do not satisfy the criterion in Eq. (13). Therefore, only 68 components are kept and the other 232 ones are removed, reducing the dimensionality from $N = 300$ to $N = 68$. Note that after the screening step, we also use \mathbf{Y} to denote the remaining random variables for convenience.

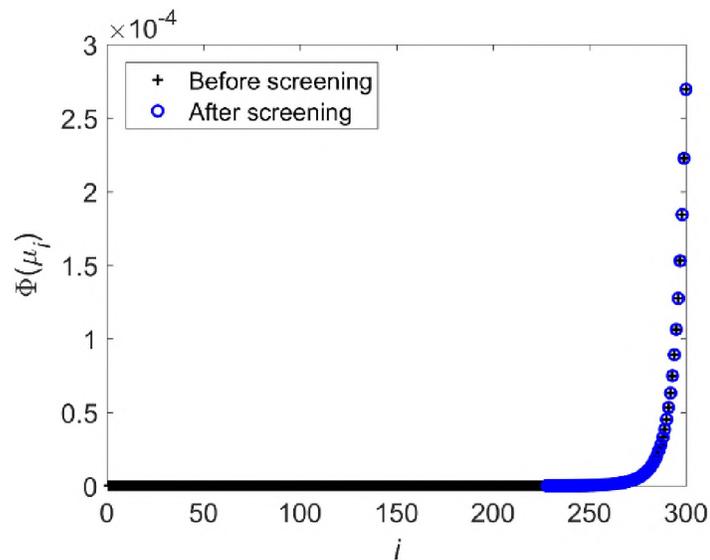


Figure 6. An example of the screening step

5.2. STEP 2: SERIES EXPANSION WITH EOLE

The purpose of EOLE [49] is to transform the integral in Eq. (11) from the \mathbf{Y} space into the eigenspace. A key step of EOLE is the eigendecomposition [56] of \mathbf{C} . In linear algebra, eigendecomposition, or spectral decomposition, is the factorization of a matrix into a canonical form. With the decomposition, a square matrix \mathbf{C} is represented in terms of its eigenvalues and eigenvectors. A (non-zero) vector \mathbf{V} is an eigenvector of \mathbf{C} if it satisfies the linear equation

$$\mathbf{C}\mathbf{V} = \lambda\mathbf{V} \quad (15)$$

where λ is the eigenvalue corresponding to \mathbf{V} . The eigenvalues are obtained though solving the following equation

$$\det(\mathbf{C} - \lambda\mathbf{I}) = 0 \quad (16)$$

where $\det(\cdot)$ represents determinant, and \mathbf{I} is an identity matrix with the same size as \mathbf{C} . The number of eigenvalues obtained by solving Eq. (16) is N_{rank} , the rank of \mathbf{C} . Once an eigenvalue is obtained, we can calculate its corresponding eigenvector by substituting it into Eq. (15).

With the eigendecomposition, we obtain N_{rank} eigenvalues λ and N_{rank} eigenvectors $\mathbf{V}_j, j = 1, 2, \dots, N_{\text{rank}}$. Note that the eigenvalues are sorted from the largest to the smallest. Then the EOLE expansion of \mathbf{Y} is given by

$$Y_i(\mathbf{U}) = \mu_i + \sum_{j=1}^{N_{\text{rank}}} \frac{U_j}{\sqrt{\lambda_j}} \mathbf{V}_j^T \mathbf{C}(:, j), i = 1, 2, \dots, N \quad (17)$$

where $\mathbf{U} = [U_1, U_2, \dots, U_j, \dots, U_{N_{\text{rank}}}]$ are N_{rank} mutually independent standard normal variables. λ_j is the j^{th} eigenvalue, and $\mathbf{C}(:, j)$ is the j^{th} column of \mathbf{C} . The j^{th} eigenvalue λ_j measures how sensitive \mathbf{Y} is to U_j .

For a full-rank \mathbf{C} , $N_{\text{rank}} = N$, and hence there are $N + 1$ terms in the expansion. For a non-full-rank \mathbf{C} , with $N_{\text{rank}} < N$, there are only N_{rank} non-zero eigenvalues, and therefore there are less than $N + 1$ terms in the expansion. In practical engineering, however, not all the N_{rank} eigenvalues are at the same level of magnitude. Excluding the μ_i term, we only keep the first N' terms that have large eigenvalues, because they contribute most to the expansion. Hereafter, we let \mathbf{U} denote $[U_1, U_2, \dots, U_j, \dots, U_{N'}]$. The uncertainty of \mathbf{Y} is mainly propagated from the uncertainty of \mathbf{U} , and hence we call \mathbf{U} significant basic random variables.

Specifically, N' is determined as the smallest integer that meets the criterion as follows

$$\left(\sum_{j=1}^{N'} \lambda_j \right) / \left(\sum_{j=1}^{N_{\text{rank}}} \lambda_j \right) \geq \eta \quad (18)$$

where η is a hyperparameter determining the accuracy of the expansion. It takes a value close to, but not larger than, 1. The smaller is η , the less accurate is the expansion. If $\eta = 1$, the expansion is exact. Typically, η is set to 0.9999. When N' has been determined by Eq. (18), the truncated EOLE expansion is given by

$$Y_i(\mathbf{U}) = \mu_i + \sum_{j=1}^{N'} \frac{U_j}{\sqrt{\lambda_j}} \mathbf{v}_j^T \mathbf{C}(:, j), i = 1, 2, \dots, N \quad (19)$$

With the truncated EOLE expansion, each Y_i is a function of \mathbf{U} , and hence $Z(\mathbf{Y}) = \max(\mathbf{Y})$ is also a function of \mathbf{U} . Then Eq. (11) is converted into

$$M_Z(s) = \int_{-\infty}^{+\infty} \exp[sz(\mathbf{u})] f_U(\mathbf{u}) d\mathbf{u} \quad (20)$$

where $f_U(\mathbf{u})$ is the PDF of \mathbf{U} , i.e., the PDF of N' -dimensional mutually independent standard normal variables.

Eq. (20) shows an N' -dimensional integral. Compared to Eq. (11) for an N -dimensional integral, Eq. (20) is more efficient because of the dimension reduction. With the dimension reduction, the efficiency of the proposed method mainly depends on N' instead of N . Intuitively, a larger N will lead to a larger N' . However, there is no direct relationship between N' and N . It is the number of significant eigenvalues of \mathbf{C} that directly determines N' . A \mathbf{C} with a dimension of 1,000 by 1,000 may have only two significant eigenvalues and hence $N' = 2$, while another \mathbf{C} with a dimension of 5 by 5 may have up to 5 significant eigenvalues and hence $N' = 5$.

5.3. STEP 3: CALCULATE MGF WITH GAUSS-HERMITE QUADRATURE

The purpose of this step is to calculate the multidimensional integral in Eq. (20) efficiently. Gauss-Hermite quadrature is a form of Gaussian quadrature for approximating the integrals with the following format

$$I = \int_{-\infty}^{+\infty} g(u) \exp(-u^2) du \quad (21)$$

where I is the integral result, $g(u)$ is a smooth and continuous function of u , and $\exp(-u^2)$ is called a weight function. With the Gauss-Hermite quadrature, Eq. (21) is approximated by

$$I = \sum_{q=1}^Q w^{(q)} g[u^{(q)}] \quad (22)$$

where Q , the quadrature order, is the number of quadrature points used, $w^{(q)}$ is the q^{th} weight, and $u^{(q)}$ is the q^{th} quadrature point. Table 1 shows the quadrature points and weights for some quadrature orders.

When the weight function is the PDF of the standard normal variable, i.e., $\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right)$, instead of $\exp(-u^2)$, the quadrature weights and points should be modified accordingly. The modification rule is simply multiplying the weights by $\frac{1}{\sqrt{\pi}}$, and the points by $\sqrt{2}$. For example, the weights and points in Table 1 are modified to that in Table 2.

Table 1. Gauss-Hermite quadrature points and weights

Quadrature order Q	Quadrature point $u^{(q)}$	Quadrature weight w_q
1	0	1.772453
2	± 0.707107	0.886227
3	0	1.81635
	± 1.22474	0.295409
4	± 0.524648	0.804914
	± 1.65068	0.081312

The integral in Eq. (20) is N' -dimensional, and the unidimensional formulation in Eq. (22) is extended to its multidimensional counterpart using the tensor product rule. The N' -dimensional Gauss-Hermite quadrature formulation is given by

$$I = \sum_{q_1=1}^{Q_1} \sum_{q_2=1}^{Q_2} \dots \sum_{q_{N'}=1}^{Q_{N'}} w_1^{(q_1)} w_2^{(q_2)} \dots w_{N'}^{(q_{N'})} g \left[u_1^{(q_1)}, u_2^{(q_2)}, \dots, u_{N'}^{(q_{N'})} \right] \quad (23)$$

where Q_j is the quadrature order in the j^{th} dimension. Therefore, Eq. (20) is approximated by

$$M_Z(s) = \sum_{q_1=1}^{Q_1} \sum_{q_2=1}^{Q_2} \dots \sum_{q_{N'}=1}^{Q_{N'}} w_1^{(q_1)} w_2^{(q_2)} \dots w_{N'}^{(q_{N'})} \exp \left\{ sZ \left[u_1^{(q_1)}, u_2^{(q_2)}, \dots, u_{N'}^{(q_{N'})} \right] \right\} \quad (24)$$

Note that the weight function $f_U(\mathbf{u})$ in Eq. (20) is the PDF of N' mutually independent standard normal variables and Eq. (24) should use the modified quadrature weights and points. The total number N_Q of quadrature points is equal to $\prod_{j=1}^{N'} Q_j$.

Table 2. Modified Gauss-Hermite quadrature weights and points

Quadrature order Q	Quadrature point $u^{(q)}$	Quadrature weight w_q
1	0	$1.772453/\sqrt{\pi}$
2	$\pm 0.707107\sqrt{2}$	$0.886227/\sqrt{\pi}$
3	0	$1.81635/\sqrt{\pi}$
	$\pm 1.22474\sqrt{2}$	$0.295409/\sqrt{\pi}$
4	$\pm 0.524648\sqrt{2}$	$0.804914/\sqrt{\pi}$
	$\pm 1.65068\sqrt{2}$	$0.081312/\sqrt{\pi}$

Generally, the higher are the quadrature orders $Q_j, j = 1, 2, \dots, N'$, the higher is the accuracy. Higher quadrature orders, however, mean lower efficiency. Therefore, a good tradeoff is needed. Since the j^{th} eigenvalue λ_j of \mathbf{C} measures how sensitive \mathbf{Y} is to U_j , as mentioned in Subsection 5.2, λ_j also measures how sensitive Z is to U_j . Hence, we assign values to $Q_j, j = 1, 2, \dots, N'$, according to the corresponding eigenvalues.

To determine $Q_j, j = 1, 2, \dots, N'$, we need the maximum and minimum allowable values Q_{\max} and Q_{\min} . Since λ_1 is the largest eigenvalue, we set Q_1 to Q_{\max} . $Q_j, j = 2, 3, \dots, N'$, are determined by

$$Q_j = \max \left\{ \text{round} \left(\frac{\lambda_j}{\lambda_1} Q_1 \right), Q_{\min} \right\} \quad (25)$$

where $\text{round}(\cdot)$ rounds its input value to the nearest integer. Eq. (25) shows that the larger is $\frac{\lambda_j}{\lambda_1}$, the larger is Q_j , but Q_j cannot be smaller than Q_{\min} . The specific values of the two hyperparameters Q_{\max} and Q_{\min} are dependent on the requirement of calculation accuracy and efficiency.

5.4. STEP 4: TRANSFORM MGF TO CDF USING SPA

SPA is a powerful tool to transform MGF to CDF as well as to PDF. Although the theory behind SPA is complicated, its implementation is straightforward.

First, the MGF $M_Z(s)$ is transformed to CGF $K_Z(s)$ through

$$K_Z(s) = \ln [M_Z(s)] \quad (26)$$

Then the first derivative $\dot{K}_Z(s)$ of $K_Z(s)$ is given by

$$\dot{K}_Z(s) = \frac{\dot{M}_Z(s)}{M_Z(s)} \quad (27)$$

where $\dot{M}_Z(s)$ is the first derivative of $M_Z(s)$ and is given by

$$\begin{aligned} \dot{M}_Z(s) = & \sum_{q_1=1}^{Q_1} \sum_{q_2=1}^{Q_2} \dots \sum_{q_{N'}=1}^{Q_{N'}} w_1^{(q_1)} w_2^{(q_2)} \dots w_{N'}^{(q_{N'})} Z \left[u_1^{(q_1)}, u_2^{(q_2)}, \dots, u_{N'}^{(q_{N'})} \right] \\ & * \exp \left\{ s Z \left[u_1^{(q_1)}, u_2^{(q_2)}, \dots, u_{N'}^{(q_{N'})} \right] \right\} \end{aligned} \quad (28)$$

The second derivative $\ddot{K}_Z(s)$ of $K_Z(s)$ is given by

$$\ddot{K}_Z(s) = -\frac{\dot{M}_Z^2(s)}{M_Z^2(s)} + \frac{\ddot{M}_Z(s)}{M_Z(s)} \quad (29)$$

where $\ddot{M}_Z(s)$ is the second derivative of $M_Z(s)$ and is given by

$$\begin{aligned} \ddot{M}_Z(s) = & \sum_{q_1=1}^{Q_1} \sum_{q_2=1}^{Q_2} \dots \sum_{q_{N'}=1}^{Q_{N'}} w_1^{(q_1)} w_2^{(q_2)} \dots w_{N'}^{(q_{N'})} z^2 \left[u_1^{(q_1)}, u_2^{(q_2)}, \dots, u_{N'}^{(q_{N'})} \right] \\ & * \exp \left\{ s z \left[u_1^{(q_1)}, u_2^{(q_2)}, \dots, u_{N'}^{(q_{N'})} \right] \right\} \end{aligned} \quad (30)$$

Daniels [57] derived the SPA to the PDF $f_Z(z)$ of Z as

$$f_Z(z) = \left[\frac{1}{2\pi \dot{K}_Z(s^*)} \right]^{\frac{1}{2}} \exp[K_Z(s^*) - s^* z] \quad (31)$$

where s^* , known as the saddlepoint, is the solution to the equation given by

$$\dot{K}(s) = z \quad (32)$$

The bisection method [58] is employed to solve Eq. (32). Apart from $f_Z(z)$, the CDF $F_Z(z)$ is given by

$$F_Z(z) = \Phi[w(z)] + \phi[w(z)] \left(\frac{1}{w(z)} - \frac{1}{v} \right) \quad (33)$$

where $\phi(\cdot)$ is the PDF of the standard normal variable,

$$w(z) = \text{sign}(s^*) \{ 2[s^* z - K_Z(s^*)] \}^{\frac{1}{2}} \quad (34)$$

and

$$v = s^* [\dot{K}_Z(s^*)]^{\frac{1}{2}} \quad (35)$$

Since we only need to calculate $F_Z(0)$, we can simply set $z = 0$ in Eqs. (32), (33) and (34).

Once $F_Z(0)$ is obtained, we also obtain the desired multivariate normal probability

$$F_Y(\mathbf{0}; \boldsymbol{\mu}, \mathbf{C}) = F_Z(0).$$

An important property of SPA is that it can convert MGF to CDF with sufficient accuracy, especially at the tails of CDF [51, 57]. Some studies showed that in some cases, SPA has tail exactness [59]. This property makes the proposed method able to calculate very small probabilities with high accuracy.

6. APPLICATION IN TIME-DEPENDENT RELIABILITY ANALYSIS

Time-dependent reliability measures the probability that a component or system does not fail within a given period of time. With different theories, existing methods to time-dependent reliability analysis are roughly grouped into simulation methods [16, 17, 21, 33], surrogate model methods [6, 11, 12, 18-20], extreme value methods [13, 22, 23, 25], outcrossing rate methods [4, 7, 10, 15], and equivalent Gaussian process methods [5, 8, 14], etc.

Simulation methods are straightforward. A large number of samples of Y are generated first, whose statistic information is then used to estimate the reliability or the probability of failure. This group of methods is generally accurate as long as the sample size is sufficiently large. Generating a large number of samples, however, is usually expensive or even unaffordable, especially when the limit-state function is an expensive black-box function. To deal with this problem, surrogate model methods train a computationally cheap surrogate model to replace the original expensive limit-state function. Once the surrogate model is well trained, the time-dependent reliability may be estimated accurately and efficiently. This group of methods, however, introduce some

additional issues, such as the design of experiment, training scheme, learning function, and convergence criteria, etc.

Extreme value methods convert the time-dependent problems into static ones by calculating the extreme values of the limit-state function with respect to time. Generally, the calculation of extreme values needs global optimization with respect to time. It limits the application of this group of methods since global optimization may not be efficient.

Outcrossing rate methods are traditional methods for time-dependent reliability analysis and are widely used. The methods are efficient if they are used with FORM. Their accuracy may not be good for problems with low reliability because the dependence among crossing events is neglected. On the contrary, the autocorrelation of the limit-state function is considered in equivalent Gaussian process methods, and hence more accurate results can be obtained. The procedures of equivalent Gaussian process methods are straightforward. FORM is first employed to convert the limit-state function into a Gaussian process whose discretization is a vector of correlated normal variables, and then a high-dimensional normal integral is used to calculate the reliability.

The existing equivalent Gaussian process methods mainly differ in the way the high-dimensional normal integral is estimated. Hu and Du [5] employed the crude MCS. Jiang et al. [14] employed the randomized quasi MCS [35]. Gong and Frangopol [8] employed the equivalent component method. In this study, we apply the proposed method to improve the accuracy of equivalent Gaussian process methods without a random sampling method.

The reliability is predicted by a limit-state function given by

$$Y = G(\mathbf{X}, \mathbf{P}(t), t) \quad (36)$$

where \mathbf{X} are the basic input random variables, $\mathbf{P}(t)$ are the input random processes, and t is time. Generally, Y is a random process. The time-dependent reliability R over the time interval $[\underline{t}, \bar{t}]$ is given by

$$R = \Pr\{Y < 0, \forall t \in [\underline{t}, \bar{t}]\} \quad (37)$$

To calculate R numerically, we need to discrete $[\underline{t}, \bar{t}]$ into N points $t_i, i = 1, 2, \dots, N$, where $t_1 = \underline{t}$ and $t_N = \bar{t}$. Then the random process Y is discretized into N random variables $Y_i = G(\mathbf{X}, \mathbf{P}(t_i), t_i), i = 1, 2, \dots, N$. With the discretization, Eq. (37) is rewritten as

$$R = \Pr\left\{\bigcap_{i=1}^N Y_i < 0\right\} \quad (38)$$

Although Y_i is in general, not a normal variable, we can use FORM to transform it into an equivalent normal variable with a unit variance [5]. Therefore, we always assume that Y_i is normally distributed with a unit variance without losing generality. Then Eq. (38) is equivalent to

$$R = F_Y(\mathbf{0}; \boldsymbol{\mu}, \mathbf{C}) \quad (39)$$

The details of how to calculate \mathbf{C} using FORM is given in [5]. The time-dependent probability of failure $P_f = 1 - R$.

For general time-dependent reliability problems, N can be hundreds. Although N is large, the number of significant basic random variables, i.e., N' , is not necessarily large. If there are no random processes in Eq. (36), N_{rank} will be exactly equal to the dimension of \mathbf{X} , i.e., the number of basic random variables. N' is no larger than N_{rank} . $N' = N_{\text{rank}}$

only if Y is sensitive to all the basic random variables. $N' < N_{\text{rank}}$ when Y is not sensitive to at least one basic random variable. If there are input random processes, N_{rank} is dependent on not only the number of basic random variables and random processes but also the autocorrelation functions of the input random processes.

From response Y , N' is generally determined by the correlation length l_Y of Y and the length $l_t = (\bar{t} - \underline{t})$ of time interval $[\underline{t}, \bar{t}]$. More specifically, the larger $\frac{l_Y}{l_t}$ is, the smaller N' will we have. For problems with small $\frac{l_Y}{l_t}$, N' is large and hence the proposed method may not be efficient or may even fail.

7. NUMERICAL EXPERIMENTS

In this section, we demonstrate the effectiveness of the proposed method using four time-dependent reliability analysis examples. The first example has the exact solution and hence we can easily test the accuracy of the proposed method. In the second example, the limit-state function is given as a Gaussian random process. The third example involves a mechanism whose inputs only contain several random variables without a random process. The last example has an implicit limit-state function, which is a black-box model evaluated by the finite element method (FEM) [60]. Exact solutions are not available for the last three examples, and hence we employ the crude MCS, using sufficiently large sample size, to obtain accurate results, which are treated as benchmarks. In all the examples, the hyperparameters c , η , Q_{max} , and Q_{max} are set to 10^{-4} , 0.9999, 35, and 5, respectively. Note that there are no criteria for selecting specific values for those hyperparameters. We

set those values based on our experience from many experiments. Note that all the reported results and errors are about the calculation of the multinormal probabilities, so the error due to FORM approximation is not included.

The proposed method is also compared with two widely used methods. The first one is the latest version of the equivalent component method [42], which is a deterministic method. For convenience, we denote this method by IECA (improved equivalent component method). The second one is the randomized quasi MCS developed by Genz and Bretz [35], which has been implemented in the *R* programming language and has been widely used to calculate the high-dimensional normal probabilities. We can simply call the *R* function *pmvnorm()* to calculate the desired probability. Since it is a random method whose result is dependent on the seed of the random number generator, we will run this method three times to see the differences. For convenience, we denote the three solutions from the method by RQ1, RQ2, and RQ3.

7.1. EXAMPLE 1: A MATH EXAMPLE WITH EXACT SOLUTION

The limit-state function $Y(t)$ is a stationary Gaussian process with mean value $\mu(t) = b$ and standard deviation $\sigma(t) = 1$. Its autocorrelation coefficient function $\rho(t_1, t_2)$ is given by

$$\rho(t_1, t_2) = \cos(t_1 - t_2) \quad (40)$$

The time interval $[t, \bar{t}] = [0, 2\pi]$ s. $Y(t)$ is a function of $\mathbf{U} = [U_1, U_2]$ given by

$$Y(t) = b + U_1 \cos(t) + U_2 \sin(t) = b + \sqrt{U_1^2 + U_2^2} \sin \left[t + \tan^{-1} \left(\frac{U_1}{U_2} \right) \right] \quad (41)$$

Therefore, the maximum value Z of $Y(t)$ is given by

$$Z = b + \sqrt{U_1^2 + U_2^2} \quad (42)$$

Since $U_1^2 + U_2^2$ is a chi-square variable with freedom 2, the exact R is given by

$$R = \Pr\{Z < 0\} = \text{Prob}\{U_1^2 + U_2^2 < b^2\} = \Psi(b^2, 2) \quad (43)$$

where $\Psi(\cdot, 2)$ represents the chi-square CDF with the degree of freedom being 2.

$[t, \bar{t}] = [0, 2\pi]$ is evenly discretized into $N = 500$ points, hence a 500-dimensional normal probability is to be calculated. With Eq. (40), we get the correlation coefficient matrix \mathbf{C} whose dimension is 500×500 . Since $Y(t)$ is a stationary Gaussian process, after discretization, $\mathbf{Y} = (Y_1, Y_2, \dots, Y_{500})$ share the same mean value b and standard deviation 1. As a result, no components in \mathbf{Y} are removed during the variable screening procedure.

Since there are only two input random variables in Eq. (41), $N_{\text{rank}} = 2$. The corresponding two eigenvalues of \mathbf{C} are 250.5 and 249.5, both of which are significant, and therefore there are $N' = 2$ significant basic random variables. Since $Q_{\text{max}} = 35$ and $Q_{\text{min}} = 5$, we use $Q_1 = 35$ and $Q_2 = 35$ quadrature points for U_1 and U_2 , respectively, and hence there are in total $N_Q = Q_1 Q_2 = 1225$ quadrature points. To test how the proposed method performs at different levels of P_f , we vary b . The values of P_f calculated by the proposed method, IECA, and RQ are given in Table 3. Note that the values in the parentheses under P_f are relative errors with respect to the accurate solutions and that the values in the square brackets are the estimated absolute errors (EAE) given by the RQ method.

When b takes -2 , -4 , -6 , and -8 , all the relative errors of the proposed method are less than 1%. It shows that the proposed method is accurate even when we calculate an

extremely small P_f , such as 1.27×10^{-14} . The reason for the high accuracy is that there are only two significant basic random variables, and hence the Gauss-Hermite quadrature can obtain accurate MGF using Eq. (24). SPA can also produce an accurate CDF, and hence accurate P_f . Besides, this example shows that although $N = 500$, N' is only 2.

Compared to the proposed method, IECA is less accurate. When b takes -2 , -4 , -6 , and -8 , the relative errors of IECA are 57.3%, 34.7%, 15.4%, and 4.4%, respectively. When $b = -2$, RQ gets stable and accurate results. However, when calculating small probabilities ($b = -4, -6, \text{ or } -8$), RQ1, RQ2, and RQ3 produce different results, showing instability. It is a typical feature of a random method.

Table 3. Results for Example 1

$b \rightarrow$ Methods \downarrow	-2	-4	-6	-8
Proposed	1.35×10^{-1} (0.0%)	3.34×10^{-4} (-0.5%)	1.52×10^{-8} (-0.2%)	1.27×10^{-14} (0.0%)
IECA	2.13×10^{-1} (57.3%)	4.52×10^{-4} (34.7%)	1.76×10^{-8} (15.4%)	1.32×10^{-14} (4.4%)
RQ1	1.35×10^{-1} (0.0%) [7.80×10^{-6}]	4.31×10^{-4} (28.5%) [2.92×10^{-4}]	1.60×10^{-8} (5.1%) [1.84×10^{-8}]	1.48×10^{-14} (16.7%) [1.18×10^{-14}]
RQ2	1.35×10^{-1} (0.0%) [6.77×10^{-6}]	3.39×10^{-4} (1.1%) [1.93×10^{-4}]	8.78×10^{-9} (-42.4%) [3.48×10^{-9}]	9.66×10^{-15} (-23.7%) [3.83×10^{-15}]
RQ3	1.35×10^{-1} (0.0%) [5.40×10^{-6}]	2.84×10^{-4} (-15.3%) [5.29×10^{-5}]	7.77×10^{-9} (-49.0%) [9.98×10^{-10}]	9.66×10^{-15} (-23.7%) [1.94×10^{-15}]
Exact	1.35×10^{-1}	3.35×10^{-4}	1.52×10^{-8}	1.27×10^{-14}

7.2. EXAMPLE 2: A MATH EXAMPLE WITHOUT AN EXACT SOLUTION

The limit-state function $Y(t)$ is a nonstationary Gaussian process. The standard deviation is $\sigma(t) = 1$ and the mean $\mu(t)$ is given by

$$\mu(t) = -6 - t\cos(t) \quad (44)$$

where $t \in [\underline{t}, \bar{t}] = [0, 5]$ s. We consider three different correlation coefficient functions, given by Eq. (45), Eq. (46), and Eq. (47).

$$\text{Case 1: } \rho(t_1, t_2) = \sin(\pi|t_1 - t_2|)/(\pi|t_1 - t_2|) \quad (45)$$

$$\text{Case 2: } \rho(t_1, t_2) = \exp[-0.25(t_1 - t_2)^2] \quad (46)$$

$$\text{Case 3: } \rho(t_1, t_2) = \exp(-0.25|t_1 - t_2|)(1 + 0.25|t_1 - t_2|) \quad (47)$$

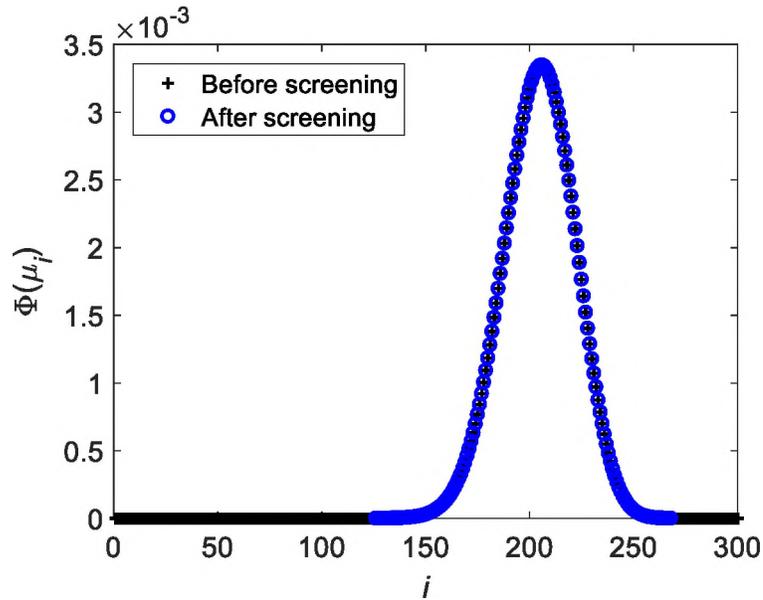


Figure 7. Variable screening for Example 2

For numerical calculation, $[\underline{t}, \bar{t}]$ is evenly discretized into $N = 300$ points, and hence the dimension is 300. Figure 7 shows the variable screening. 157 points among the

300 points do not contribute to P_f significantly and hence are removed. N is updated to 143. Note that the physical meaning of $\Phi(\mu_i)$ in Eq. (14) is the instantaneous probability of failure, and the variable screening procedure removes those time points with low instantaneous probabilities of failure.

In Case 1, there are $N' = 5$ significant basic random variables. The numbers of quadrature points for them are 35, 31, 15, 5, and 5, and hence there are in total 406,875 quadrature points. The results are given in Table 4, where ε represents the relative error with respect to MCS. The sample size of MCS is 8×10^6 .

P_f calculated by the proposed method is 6.42×10^{-3} with a relative error of -0.1% , while IECA yields a P_f value of 6.93×10^{-3} with a relative error of 7.9% . The proposed method is more accurate than IECA. RQ is more accurate than IECA, but not stable due to randomness.

Table 4. Results for Case 1 of Example 2

Methods	Proposed	IECA	RQ1	RQ2	RQ3	MCS
$P_f(\times 10^{-3})$	6.42	6.93	6.76	5.94	6.54	6.42
$\varepsilon(\%)$	-0.1	7.9	5.3	-7.5	1.8	-
EAE	-	-	6.37×10^{-4}	5.90×10^{-4}	3.40×10^{-4}	-

In Case 2, there are $N' = 4$ significant basic random variables. The numbers of quadrature points for them are 35, 7, 5, and 5, respectively, and hence there are in total 6125 quadrature points. The results are given in Table 5. The sample size of MCS is 1.2×10^7 . Again, the proposed method is more accurate than both IECA and RQ.

In Case 3, there are $N' = 4$ significant basic random variables. The numbers of quadrature points for them are 35, 5, 5, and 5, and hence there are in total 4,375 quadrature points. The results are given in Table 6. The sample size of MCS is 1.2×10^7 . All three methods are accurate, and the proposed method is slightly more accurate.

Table 5. Results for Case 2 of Example 2

Methods	Proposed	IECA	RQ1	RQ2	RQ3	MCS
$P_f(\times 10^{-3})$	3.96	3.60	3.76	4.17	4.17	3.99
$\varepsilon(\%)$	-0.8	-9.7	-5.8	4.6	4.5	-
EAE	-	-	4.49×10^{-4}	5.81×10^{-4}	4.69×10^{-4}	-

Table 6. Results for Case 3 of Example 2

Methods	Proposed	IECA	RQ1	RQ2	RQ3	MCS
$P_f(\times 10^{-3})$	3.42	3.35	3.48	3.35	3.48	3.43
$\varepsilon(\%)$	-0.2	-2.3	1.6	-2.3	1.7	-
EAE	-	-	2.31×10^{-4}	8.54×10^{-7}	2.34×10^{-4}	-

7.3. EXAMPLE 3: A SLIDER-CRANK MECHANISM

Shown in Figure 8 is a slider-crank mechanism [5]. The link with lengths R_1 and R_3 rotates with an angular velocity of $\omega = \pi$ rad/s. The motion output is the difference between the displacements of two sliders A and B . The mechanism is supposed to work with small motion errors during the time period $[t, \bar{t}] = [0, 2]$ seconds. The motion error is defined as the difference between the desired motion output and the actual motion output. A failure occurs when the motion error is larger than 0.94 mm. The actual motion output Δs_{actual} is given by

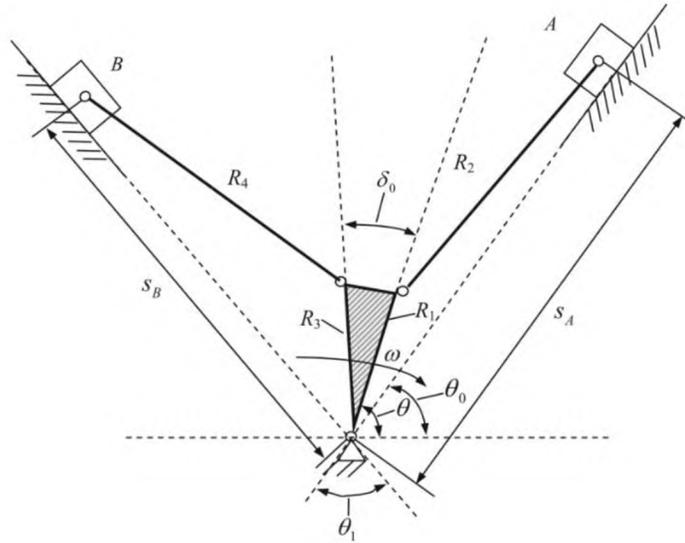


Figure 8. A slider-crank mechanism [5]

$$\begin{aligned} \Delta s_{\text{actual}} = & R_1 \cos(\theta - \theta_0) + \sqrt{R_2^2 - R_1^2 \sin^2(\theta - \theta_0)} \\ & - R_3 \cos(\theta_1 + \theta_0 - \theta - \delta_0) \\ & - \sqrt{R_4^2 - R_3^2 \sin^2(\theta_1 + \theta_0 - \theta - \delta_0)} \end{aligned} \quad (48)$$

where $\theta = \omega t$. The desired motion output $\Delta s_{\text{desired}}$ is given by

$$\begin{aligned} \Delta s_{\text{desired}} = & 108 \cos(\theta - \theta_0) + \sqrt{211^2 - 108^2 \sin^2(\theta - \theta_0)} \\ & - 100 \cos(\theta_1 + \theta_0 - \theta - \delta_0) \\ & - \sqrt{213^2 - 100^2 \sin^2(\theta_1 + \theta_0 - \theta - \delta_0)} \end{aligned} \quad (49)$$

Then the limit-state function $Y(t)$ is given by

$$Y(t) = (\Delta s_{\text{desired}} - \Delta s_{\text{actual}}) - 0.94 \quad (50)$$

Table 7 shows the random variables and other parameters.

Table 7. Variables and parameters of Example 3

Variable	Mean	Standard deviation	Distribution
R_1	108 mm	0.05 mm	Gaussian
R_2	211 mm	0.2 mm	Gaussian
R_3	100 mm	0.05 mm	Gaussian
R_4	213 mm	0.2 mm	Gaussian
θ_0	45°	0	Deterministic
θ_1	60°	0	Deterministic
δ_0	10°	0	Deterministic
ω	π rad/s	0	Deterministic

The time interval $[\underline{t}, \bar{t}]$ is evenly discretized into $N = 300$ points. Since $Y(t)$ is not a Gaussian random process, we need to transform it into an equivalent Gaussian process by applying FORM at each time point. After that, we need to calculate a 300-dimensional normal probability to obtain P_f . Figure 9 shows the variable screening step. No points among the 300 points are removed because the instantaneous probabilities of failure at all the 300 points contribute to P_f significantly.

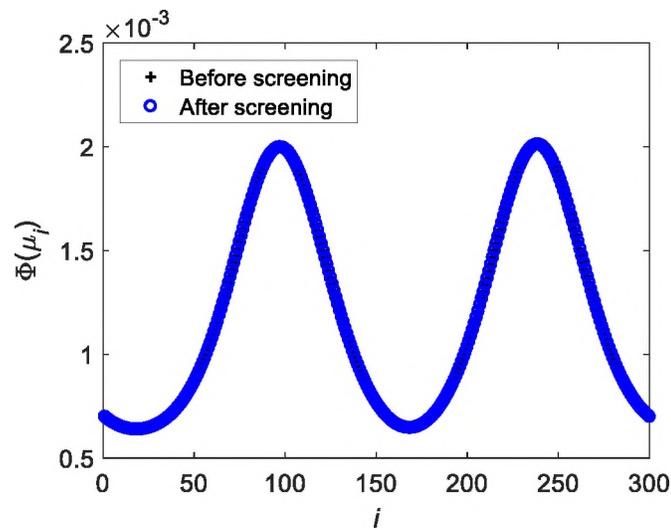


Figure 9. Variable screening for Example 3

There are four significant basic random variables in \mathbf{U} after the dimension reduction is performed. The numbers of quadrature points for \mathbf{U} are 35, 5, 5, and 5, and hence there are in total 4,375 quadrature points. The results are given in Table 8. The sample size of MCS is 1.8×10^7 .

P_f calculated by the proposed method is 2.38×10^{-3} with a relative error of 0.1%, while P_f calculated by IECA is 2.11×10^{-3} with a relative error of -11.4% . RQ is more accurate than IECA but less accurate than the proposed method.

Table 8. Results of Example 3

Methods	Proposed	IECA	RQ1	RQ2	RQ3	MCS
$P_f(\times 10^{-3})$	2.38	2.11	2.48	2.39	2.48	2.38
$\varepsilon(\%)$	0.1	-11.4	4.1	0.5	4.1	-
EAE	-	-	3.24×10^{-4}	3.82×10^{-4}	4.62×10^{-4}	-

Note that there is no input random process in this example and hence the number N' of significant basic random variables is at most the number of input random variables. If $Y(t)$ was not sensitive to some input random variables, N' would be less than the number of input random variables.

7.4. EXAMPLE 4: A 52-BAR SPACE TRUSS

This example is modified from an example in [61]. Shown in Figure 10 is a 52-bar space truss with 21 nodes. All the nodes are located on the surface of an imaginary hemisphere whose radius is $r = 240$ in. The cross-sectional areas of Bars 1~8 and 29~36 are 2 in^2 . The cross-sectional areas of Bars 9~16 and other bars are 1.2 in^2 and 0.6 in^2 ,

respectively. The Young's modulus of all bars is E . To distinguish the node numbers and the bar numbers, we add a decimal point after all node numbers in Figure 13. Nodes 1~13 are subjected to external loads $F_1 \sim F_{13}$, all in the $-z$ direction. F_1 is a stationary Gaussian process whose autocorrelation coefficient function is given by

$$\rho(t_1, t_2) = \exp[-0.25(t_1 - t_2)^2] \quad (51)$$

E and $F_2 \sim F_{13}$ are random variables, and their distributions are given in Table 9.

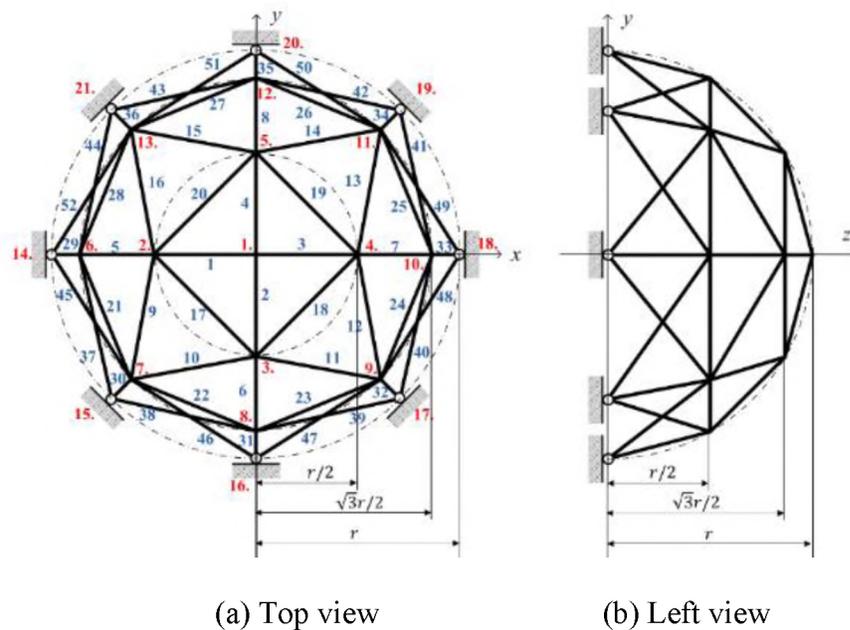


Figure 10. A 52-bar space truss [61]

Table 9. Variables and parameters of Example 4

Variable	Mean	Standard deviation	Distribution	Autocorrelation
E	2.5×10^4 ksi	2.5×10^2 ksi	Gaussian	N/A
$F_1(t)$	40 kip	4 kip	Nonstationary Gaussian process	Eq. (51)
$F_2 \sim F_5$	50 kip	5 kip	Lognormal	N/A
$F_6 \sim F_{13}$	60 kip	6 kip	Lognormal	N/A

A failure occurs when the displacement δ of Node 1 along $-z$ direction exceeds the threshold $\delta_0 = 1.3$ in at any instant of time in the period $[\underline{t}, \bar{t}] = [0, 5]$ years. The limit-state function is given by

$$Y(t) = \delta_0 - \delta(E, \mathbf{F}) \quad (52)$$

where $\mathbf{F} = [F_1(t), F_2, F_3, \dots, F_{13}]$ is the vector of all the loads. $\delta(E, \mathbf{F})$ is calculated by FEM. The linear bar element is used.

The time interval $[\underline{t}, \bar{t}]$ is evenly discretized into $N = 500$ points. Since $Y(t)$ is not a Gaussian random process, we need to transform it into an equivalent Gaussian process by applying FORM at each time point. After that, we need to calculate a 500-dimensional normal probability to obtain P_f . Since $Y(t)$ becomes a stationary Gaussian process after the transformation, $\mathbf{Y} = (Y_1, Y_2, \dots, Y_{500})$ share the same mean value and standard deviation. As a result, no components in \mathbf{Y} are removed during the variable screening procedure.

There are only $N' = 7$ significant basic random variables after the dimension reduction. The numbers of quadrature points for them are 35, 18, 6, 5, 5, 5, and 5, and hence there are in total 2,362,500 quadrature points. The sample size of MCS is 1.2×10^8 . The results are given in Table 10. The proposed method is significantly more accurate than both RQ and IECA.

Table 10. Results for Example 4

Methods	Proposed	IECA	RQ1	RQ2	RQ3	MCS
$P_f (\times 10^{-4})$	3.35	4.07	4.11	4.25	2.72	3.36
$\varepsilon (\%)$	-0.6	21.0	22.3	26.4	-19.1	-
EAE	-	-	2.51×10^{-4}	4.72×10^{-4}	2.13×10^{-4}	-

The four examples have demonstrated the high accuracy and robustness of the proposed method. IECA is accurate for some examples but less accurate for others, and RQ is not robust for some problems because of large randomness in the solutions with different sampling seeds. The proposed method works particularly well for a time-dependent reliability analysis for which the limit-state function has been approximated by a Gaussian process.

8. CONCLUSIONS

Evaluating a multivariate normal probability is widely encountered in many engineering problems. It is a challenging task when the dimension is high and the probability is low. The proposed method addresses the problem by using the extreme value of all the normal variables. Its moment generating function (MGF) is obtained by the Gauss-Hermite quadrature method, and the dimension is also reduced by screening out variables in both the physical space and the eigenspace. The saddlepoint approximation is used to recover the multivariate normal probability from MGF.

The main computational effort is the calculation of MGF by a multidimensional quadrature method. The efficiency depends on the dimension of the integral or the reduced dimension. Therefore, the efficiency of the proposed method mainly depends on the number of the significant basic random variables after the dimension reduction, instead of the dimension of the original normal variables. This is a good feature for many engineering problems where the dimension can be reduced significantly because not all normal

variables contribute significantly to the multivariate normal probability and the multivariate normal probability is not sensitive to all coordinates of the eigenspace.

Another advantage of the proposed method is its ability to calculate extremely small probabilities. The accuracy is achieved by the accurate generation of MGF, as well as saddlepoint approximation with its well-known accuracy for small probabilities. This feature makes the proposed method suitable for reliability applications where the probability of failure is inevitably small. The proposed method is also numerically stable, and the result is repeatable.

The method, however, may not work well if the reduced dimension is still high. For example, in time-dependent reliability problems, if the correlation length of the limit-state function is short and/or the time interval of interest is long, the reduced dimension will be high and the proposed method may not work well or may even fail. Our future work will focus on accommodating a larger dimension in the reduced space.

ACKNOWLEDGEMENTS

We would like to acknowledge the support from the National Science Foundation under Grant No 1923799 (formerly 1727329).

REFERENCES

- [1] Rausand M, Høyland A. System reliability theory: models, statistical methods, and applications. Hoboken: John Wiley & Sons; 2003.
- [2] Du X. System reliability analysis with saddlepoint approximation. Structural and Multidisciplinary Optimization. 2010;42:193-208.

- [3] Kang W-H, Song J. Evaluation of multivariate normal integrals for general systems by sequential compounding. *Structural Safety*. 2010;32:35-41.
- [4] Hu Z, Du X. Time-dependent reliability analysis with joint upcrossing rates. *Structural and Multidisciplinary Optimization*. 2013;48:893-907.
- [5] Hu Z, Du X. First order reliability method for time-variant problems using series expansions. *Structural and Multidisciplinary Optimization*. 2015;51:1-21.
- [6] Hu Z, Mahadevan S. A single-loop kriging surrogate modeling for time-dependent reliability analysis. *Journal of Mechanical Design*. 2016;138:061406.
- [7] Jiang C, Wei XP, Huang ZL, Liu J. An outcrossing rate model and its efficient calculation for time-dependent system reliability analysis. *Journal of Mechanical Design*. 2017;139:041402.
- [8] Gong C, Frangopol DM. An efficient time-dependent reliability method. *Structural Safety*. 2019;81:101864.
- [9] Wei X, Du X. Uncertainty Analysis for Time-and Space-Dependent Responses With Random Variables. *Journal of Mechanical Design*. 2019;141:021402.
- [10] Andrieu-Renaud C, Sudret B, Lemaire M. The PHI2 method: a way to compute time-variant reliability. *Reliability Engineering & System Safety*. 2004;84:75-86.
- [11] Wang Z, Chen W. Time-variant reliability assessment through equivalent stochastic process transformation. *Reliability Engineering & System Safety*. 2016;152:166-75.
- [12] Hu Z, Du X. Mixed efficient global optimization for time-dependent reliability analysis. *Journal of Mechanical Design*. 2015;137:051401.
- [13] Hu Z, Du X. A sampling approach to extreme value distribution for time-dependent reliability analysis. *Journal of Mechanical Design*. 2013;135:071003.
- [14] Jiang C, Wei XP, Wu B, Huang ZL. An improved TRPD method for time-variant reliability analysis. *Structural and Multidisciplinary Optimization*. 2018.
- [15] Sudret B. Analytical derivation of the outcrossing rate in time-variant reliability problems. *Structure and Infrastructure Engineering*. 2008;4:353-62.
- [16] Wang Z, Mourelatos ZP, Li J, Baseski I, Singh A. Time-dependent reliability of dynamic systems using subset simulation with splitting over a series of correlated time intervals. *Journal of Mechanical Design*. 2014;136:061008.

- [17] Singh A, Mourelatos Z, Nikolaidis E. Time-dependent reliability of random dynamic systems using time-series modeling and importance sampling. *SAE International Journal of Materials and Manufacturing*. 2011;4:929-46.
- [18] Wang Z, Wang P. A Nested Extreme Response Surface Approach for Time-Dependent Reliability-Based Design Optimization. *Journal of Mechanical Design*. 2012;134:121007-14.
- [19] Shi Y, Lu Z, Xu L, Chen S. An adaptive multiple-Kriging-surrogate method for time-dependent reliability analysis. *Applied Mathematical Modelling*. 2019;70:545-71.
- [20] Wang D, Jiang C, Qiu H, Zhang J, Gao L. Time-dependent reliability analysis through projection outline-based adaptive Kriging. *Structural and Multidisciplinary Optimization*. 2020:1-20.
- [21] Du W, Luo Y, Wang Y. Time-variant reliability analysis using the parallel subset simulation. *Reliability Engineering & System Safety*. 2019;182:250-7.
- [22] Shi Y, Lu Z, Cheng K, Zhou Y. Temporal and spatial multi-parameter dynamic reliability and global reliability sensitivity analysis based on the extreme value moments. *Structural and Multidisciplinary Optimization*. 2017;56:117-29.
- [23] Chen J-B, Li J. The extreme value distribution and dynamic reliability analysis of nonlinear structures with uncertain parameters. *Structural Safety*. 2007;29:77-93.
- [24] Yu S, Wang Z. A general decoupling approach for time-and space-variant system reliability-based design optimization. *Computer Methods in Applied Mechanics and Engineering*. 2019;357:112608.
- [25] Yu S, Wang Z, Meng D. Time-variant reliability assessment for multiple failure modes and temporal parameters. *Structural and Multidisciplinary Optimization*. 2018;58:1705-17.
- [26] Zhang D, Han X, Jiang C, Liu J, Li Q. Time-dependent reliability analysis through response surface method. *Journal of Mechanical Design*. 2017;139.
- [27] Madsen HO, Krenk S, Lind NC. *Methods of structural safety*. Englewood Cliffs: Prentice-Hall; 2003.
- [28] De Haan L, Ferreira A. *Extreme value theory: an introduction*. New York: Springer Science & Business Media; 2007.
- [29] Ochi Y, Prentice RL. Likelihood inference in a correlated probit regression model. *Biometrika*. 1984;71:531-43.

- [30] Dunnett CW. A multiple comparison procedure for comparing several treatments with a control. *Journal of the American Statistical Association*. 1955;50:1096-121.
- [31] Anderson JA, Pemberton J. The grouped continuous model for multivariate ordered categorical variables and covariate adjustment. *Biometrics*. 1985:875-85.
- [32] Trinh G, Genz A. Bivariate conditioning approximations for multivariate normal probabilities. *Statistics and Computing*. 2015;25:989-96.
- [33] Mooney CZ. Monte carlo simulation. Thousand Oaks: Sage Publications; 1997.
- [34] Sobol' I. Quasi-monte carlo methods. *Progress in Nuclear Energy*. 1990;24:55-61.
- [35] Genz A, Bretz F. Computation of multivariate normal and t probabilities. Heidelberg: Springer Science & Business Media; 2009.
- [36] Melchers R. Importance sampling in structural systems. *Structural Safety*. 1989;6:3-10.
- [37] Phinikettos I, Gandy A. Fast computation of high-dimensional multivariate normal probabilities. *Computational Statistics & Data Analysis*. 2011;55:1521-9.
- [38] Ambartzumian R, Der Kiureghian A, Ohanian V, Sukiasiana H. Multinormal probability by sequential conditioned importance sampling: theory and application. *Probabilistic Engineering Mechanics*. 1998;13:299-308.
- [39] Au S-K, Beck JL. Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic engineering mechanics*. 2001;16:263-77.
- [40] Cadini F, Gioietta A. A Bayesian Monte Carlo-based algorithm for the estimation of small failure probabilities of systems affected by uncertainties. *Reliability Engineering & System Safety*. 2016;153:15-27.
- [41] Roscoe K, Diermanse F, Vrouwenvelder T. System reliability with correlated components: Accuracy of the Equivalent Planes method. *Structural Safety*. 2015;57:53-64.
- [42] Gong C, Zhou W. Improvement of equivalent component approach for reliability analyses of series systems. *Structural Safety*. 2017;68:65-72.
- [43] Gollwitzer S, Rackwitz R. Equivalent components in first-order system reliability. *Reliability Engineering*. 1983;5:99-115.

- [44] Hohenbichler M, Rackwitz R. First-order concepts in system reliability. *Structural Safety*. 1982;1:177-88.
- [45] Tang L, Melchers R. Improved approximation for multinormal integral. *Structural Safety*. 1986;4:81-93.
- [46] Pandey M. An effective approximation to evaluate multinormal integrals. *Structural Safety*. 1998;20:51-67.
- [47] Yuan X-X, Pandey M. Analysis of approximations for multinormal integration in system reliability computation. *Structural Safety*. 2006;28:361-77.
- [48] Mendell NR, Elston R. Multifactorial qualitative traits: genetic analysis and prediction of recurrence risks. *Biometrics*. 1974:41-57.
- [49] Li C-C, Der Kiureghian A. Optimal discretization of random fields. *Journal of Engineering Mechanics*. 1993;119:1136-54.
- [50] Liu Q, Pierce DA. A note on Gauss—Hermite quadrature. *Biometrika*. 1994;81:624-9.
- [51] Butler RW. *Saddlepoint approximations with applications*. Cambridge: Cambridge University Press; 2007.
- [52] Hu Z, Du X. Saddlepoint approximation reliability method for quadratic functions in normal variables. *Structural Safety*. 2018;71:24-32.
- [53] Chi H, Mascagni M, Warnock T. On the optimal Halton sequence. *Mathematics and computers in simulation*. 2005;70:9-21.
- [54] Tuffin B. Randomization of quasi-monte carlo methods for error estimation: Survey and normal approximation. *Monte Carlo Methods and Applications*. 2004;10:617-28.
- [55] Ihaka R, Gentleman R. R: a language for data analysis and graphics. *Journal of Computational and Graphical Statistics*. 1996;5:299-314.
- [56] Abdi H, statistics. The eigen-decomposition: Eigenvalues and eigenvectors. *Encyclopedia of measurement and statistics*. 2007:304-8.
- [57] Daniels HE. Saddlepoint Approximations in Statistics. *The Annals of Mathematical Statistics*. 1954;25:631-50.
- [58] Press WH, Teukolsky SA, Vetterling WT, Flannery BP. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge: Cambridge university press; 2007.

- [59] Barndorff - Nielsen OE, Kluppelberg CJSjos. Tail exactness of multivariate saddlepoint approximations. 1999;26:253-64.
- [60] Zienkiewicz OC, Taylor RL, Nithiarasu P, Zhu J. The finite element method. London: McGraw-hill; 1977.
- [61] Zhang Z, Jiang C, Han X, Ruan X. A high-precision probabilistic uncertainty propagation method for problems involving multimodal distributions. Mechanical Systems & Signal Processing. 2019;126:21-41.

II. PHYSICS-BASED GAUSSIAN PROCESS METHOD FOR PREDICTING AVERAGE PRODUCT LIFETIME IN DESIGN STAGE

ABSTRACT

The average lifetime or the mean time to failure (MTTF) of a product is an important metric to measure product reliability. Current methods of evaluating MTTF are mainly based on statistics or data. They need lifetime testing on many products to get the lifetime samples, which are then used to estimate the MTTF. The lifetime testing, however, is expensive in terms of both time and cost. The efficiency is also low because it cannot be effectively incorporated in the early design stage where many physics-based models are available. We propose to predict the MTTF in the design stage using a physics-based Gaussian process method. Since the physics-based models are usually computationally demanding, we face a problem with both big data (on the model input side) and small data (on the model output side). The proposed adaptive supervised training method with the Gaussian process regression can quickly predict the MTTF with a minimized number of calling the physical models. The proposed method can enable the design to be continually improved by changing design variables until reliability measures, including the MTTF, are satisfied. The effectiveness of the method is demonstrated by three examples.

1. INTRODUCTION

In reliability engineering [1-5], the average lifetime, or the mean time to failure (MTTF), is an important metric of product reliability [1, 6]. Statistics-based methods [7, 8]

are widely used to estimate the MTTF. The methods need lifetime testing on many products to obtain the lifetime samples, which are then used to estimate the average lifetime by statistical analysis. The methods are generally expensive in three aspects. First, lifetime testing is time-consuming when the actual product lifetime is very long such as years. Although accelerated life testing [9] can reduce the testing time, the results may not reflect the reliability of the product in normal use conditions. Second, the cost of testing is usually high. Third, the testing is performed, and lifetime data are collected after the product was made. It is too late and more costly to fix reliability issues if the MTTF is shorter than expected. It is desirable to predict the MTTF during the early design stage.

Direct lifetime data, however, are rarely available during the design stage. Physics-based methods [10] then play an important role to deal with this problem. The methods use limit-state functions, which are computational models derived from physical principles, to predict the states of the components and subsystems of the product with respect to potential failure modes [11]. With the computational models for the failure modes, physics-based methods are much more efficient than the statistics-based methods. They can predict reliability performance for a given design. If the reliability measures, including the MTTF, do not meet the design requirements, design variables will be changed until the reliability requirements are met. Physics-based methods are therefore a powerful tool to support design for reliability [12-16].

Physics-based methods were originally developed for structural reliability analysis [10]. In the last decades, many new physics-based reliability methods have been developed. These methods cover a wide range of applications, from component reliability to system

reliability [10], and from time-independent reliability to time-dependent reliability [17-19] and time- and space-dependent reliability [20].

Computational models, such as a finite element analysis model [21], are usually computationally expensive. We usually know distributions of random input variables, and we can generate many random samples for the input variables. In this sense, we have big data. On the other hand, we can afford to run the computational models only a limited number of times, and then we have small data for the responses. For this reason, machine learning (ML) methodologies have been increasingly used for reliability analysis. For example, the Gaussian process (GP) method for quantifying model structure uncertainty [22, 23]; the support vector machine (SVM) method for estimating rare event probabilities [11], and other methods for predicting component and system reliability [24].

In this study, we extend the physics-based methods to predict the MTTF of a product. Since this task needs more calls of the computational model than a regular reliability analysis, we also rely on ML to maintain computational efficiency. Specifically, we employ the supervised machine learning method [25] and adaptively train a GP [26] to approximate the computational function with respect to the basic random input variables. A learning function is developed to guide adding training points. Once the learning is finished, the MTTF of the product is obtained.

The problem statement is given in Section 2. A brief introduction to GP is given in Section 3. The proposed method is discussed in Section 4. In Section 5, we extend the proposed method to deal with problems involving random processes. Three examples are provided in Section 6, followed by conclusions in Section 7.

2. PROBLEM STATEMENT

The computational function for reliability analysis is called a limit-state function, which is given by

$$Y = G(\mathbf{X}, t) \quad (1)$$

where $\mathbf{X} = (X_1, X_2, \dots, X_N)^T$ are N basic input random variables and t is time. Note that the input of $G(\cdot)$ may also include random processes, which can be transformed into functions with respect to random variables and t . Thus Eq. (1) does not lose generality. Y is in general a random process. The product fails once its response Y takes a negative value.

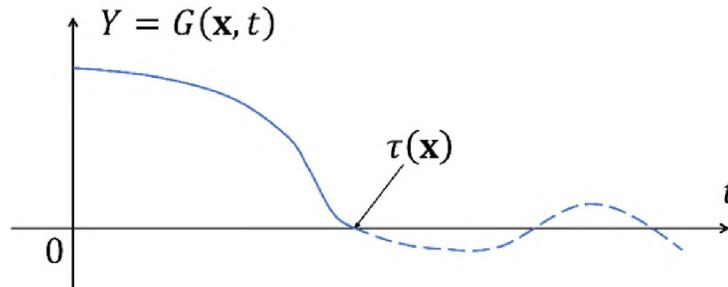


Figure 1. A sample of the limit-state function

Figure 1 shows a sample of Y when \mathbf{X} is fixed to a realization \mathbf{x} . When $t = \tau(\mathbf{x})$, Y takes a negative value for the first time, and hence $\tau(\mathbf{x})$ is called the first time to failure. If the product is non-repairable, $\tau(\mathbf{x})$ is the lifetime (given that $\mathbf{X} = \mathbf{x}$), and afterward $Y(\mathbf{x}, t), t > \tau$ has no physical meaning. Since $\tau(\mathbf{X})$ is dependent on the input random variables \mathbf{X} , it is also a random variable. The product's MTTF $\bar{\tau}$ is given by

$$\bar{\tau} = E[\tau(\mathbf{X})] \quad (2)$$

where $E(\cdot)$ represents an expectation.

The task of this study is to predict $\bar{\tau}$ efficiently and accurately. Mathematically, $\tau(\mathbf{X})$ is the first (or minimum) root of the following equation

$$G(\mathbf{X}, t) = 0 \quad (3)$$

Finding the minimum root of Eq. (3), however, may be computationally expensive when the limit-state function $G(\mathbf{X}, t)$ is an expensive black-box function. Therefore, developing an accurate and efficient first-root finder is a challenge.

3. INTRODUCTION TO GAUSSIAN PROCESS MODEL AND THE LEARNING FUNCTION U

Before presenting the proposed method, we briefly introduce GP [26] (or Kriging model [27]) and the learning function U [28], on which the proposed method is based.

A GP makes regression to a function $F(\mathbf{X})$ from a training sample set, or a design of experiment (DoE). The main idea of GP is to treat $F(\mathbf{X})$ as a realization of a Gaussian process $\hat{F}(\mathbf{X})$. The mean value function $\mu_{\hat{F}}(\mathbf{X})$, standard deviation function $\sigma_{\hat{F}}(\mathbf{X})$, and correlation coefficient function of $\hat{F}(\mathbf{X})$ are determined by using the maximum likelihood method [29]. Generally, $\mu_{\hat{F}}(\mathbf{X})$ is used as the deterministic prediction to $F(\mathbf{X})$, and $\sigma_{\hat{F}}(\mathbf{X})$ is used to measure the prediction uncertainty or prediction error. The prediction uncertainty comes from the fact that only a limit number of training points, and hence only part of the information in $F(\mathbf{X})$, are used to build $\hat{F}(\mathbf{X})$ and infer $\mu_{\hat{F}}(\mathbf{X})$. The missing information

leads to prediction uncertainty. The ability to measure the prediction error makes GP outweigh many other surrogate models. More details about GP are available in [26].

For a given specific point \mathbf{x} of input variables \mathbf{X} , the GP predicts $F(\mathbf{x})$ to be a normal variable $N(\mu_{\hat{F}}(\mathbf{x}), \sigma_{\hat{F}}^2(\mathbf{x}))$. In engineering problems where only the sign of $F(\mathbf{x})$ is of interest, such as reliability analysis where only the sign of the limit-state function is important, we need to measure how certain the sign of $F(\mathbf{x})$ has been predicted by $\text{sign}[\mu_{\hat{F}}(\mathbf{x})]$, the sign of $\mu_{\hat{F}}(\mathbf{x})$. If $\mu_{\hat{F}}(\mathbf{x}) > 0$, then the probability that $F(\mathbf{x}) > 0$ is $\Phi\left(\frac{\mu_{\hat{F}}(\mathbf{x})}{\sigma_{\hat{F}}(\mathbf{x})}\right)$, where $\Phi(\cdot)$ is the cumulative distribution function of a standard normal variable. Similarly, if $\mu_{\hat{F}}(\mathbf{x}) < 0$, then the probability that $F(\mathbf{x}) < 0$ is $\Phi\left(-\frac{\mu_{\hat{F}}(\mathbf{x})}{\sigma_{\hat{F}}(\mathbf{x})}\right)$. Combining the two cases, the probability that the sign of $F(\mathbf{x})$ has been correctly predicted by $\text{sign}[\mu_{\hat{F}}(\mathbf{x})]$ is $\Phi\left(\frac{|\mu_{\hat{F}}(\mathbf{x})|}{\sigma_{\hat{F}}(\mathbf{x})}\right)$, which is monotonous to $\Phi\left(\frac{|\mu_{\hat{F}}(\mathbf{x})|}{\sigma_{\hat{F}}(\mathbf{x})}\right)$ and known as the learning function U [28], is widely used to determine how correctly $\text{sign}[F(\mathbf{x})]$ has been predicted. In the following Section 4, we will show how GP and learning function U are used in the proposed method.

4. THE PROPOSED METHOD

4.1. OVERVIEW OF THE PROPOSED METHOD

The main idea of the proposed method is to adaptively train a GP $\hat{G}(\mathbf{X}, t)$ for $G(\mathbf{X}, t)$. With $\hat{G}(\mathbf{X}, t)$, we can obtain the surrogate model $\hat{\tau}(\mathbf{X})$ of $\tau(\mathbf{X})$ at the same time.

Since $\hat{t}(\mathbf{X})$ is computationally cheap, we can calculate $\bar{\tau}$ using Monte Carlo simulation (MCS) [30].

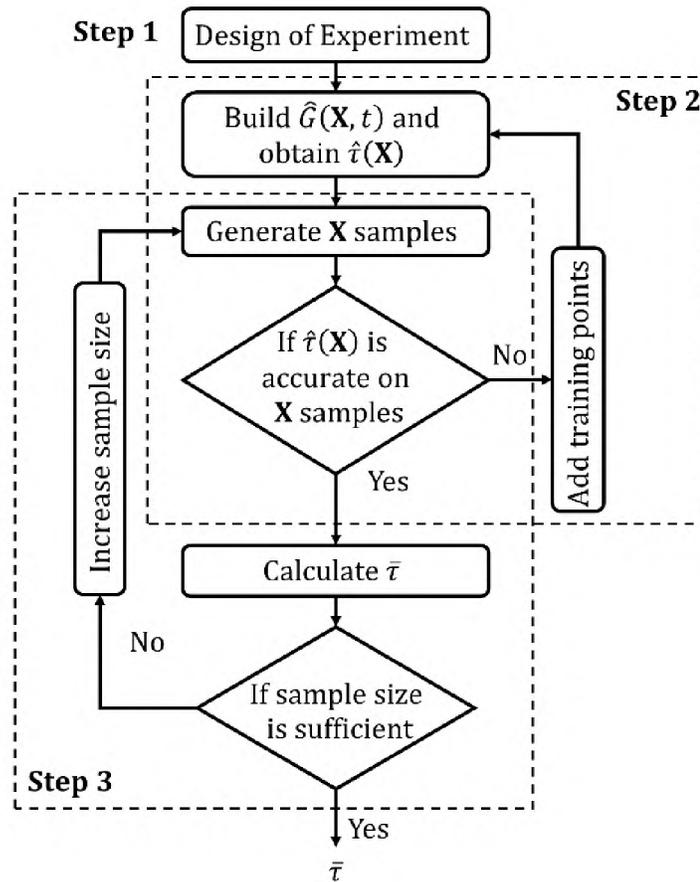


Figure 2. Brief flowchart of the proposed method

Training $\hat{G}(\mathbf{X}, t)$ should be task-oriented to improve efficiency. We develop a learning function and a stopping criterion to fulfill task-oriented training. Figure 2 shows a brief flowchart of the proposed method. There are mainly three steps. Step 1 is the design of experiments. It generates the initial training points for $\hat{G}(\mathbf{X}, t)$. In Step 2, $\hat{G}(\mathbf{X}, t)$ is adaptively refined by adding new training points. A learning function and a stopping

criterion are developed to find the new training points and determine when to terminate the training. In Step 3, the sample size of \mathbf{X} , and hence of $\hat{t}(\mathbf{X})$, is adaptively enlarged until $\bar{\tau}$ is estimated with sufficiently high fidelity. The three steps are discussed in detail in Subsections 4.2 through 4.4.

4.2. DESIGN OF EXPERIMENTS FOR INITIAL SURROGATE MODEL

The principle of the design of experiments for building a GP is to spread the initial training points evenly. Commonly used sampling methods include random sampling, Latin hypercube sampling, and Hammersley sampling [31]. In this study, we employ the Hammersley sampling method because it has better uniformity properties over a multidimensional space [32]. Since the dimension of the entire input vector (\mathbf{X}, t) is $N + 1$, the Hammersley sampling method generates initial training points in a hypercube $[0,1]^{N+1}$. To get initial training points of \mathbf{X} , we can simply use the inverse probability method to transform the training points from the hypercube space to the X-space. As for the initial training points of t , we treat t as if it was a uniform random variable and could also be transformed from the interval $[0,1]$ to the time interval $[0, T]$. We assume that T is sufficiently large so that Eq. (3) has at least a root in $[0, T]$. The initial training points \mathbf{x}^{in} of $\mathbf{X} = (X_1, X_2, \dots, X_N)^T$ are

$$\mathbf{x}^{\text{in}} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_N^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_N^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(n_{\text{in}})} & x_2^{(n_{\text{in}})} & \cdots & x_N^{(n_{\text{in}})} \end{bmatrix} \quad (4)$$

where n_{in} is the total number of initial training points. With \mathbf{x}^{in} and the initial training points \mathbf{t}^{in} of t , we then obtain initial training points \mathbf{y}^{in} of Y by evaluating Eq. (1) n_{in}

times. Finally, we get the initial training set $(\mathbf{x}^{\text{trn}}, \mathbf{t}^{\text{trn}}, \mathbf{y}^{\text{trn}}) = (\mathbf{x}^{\text{in}}, \mathbf{t}^{\text{in}}, \mathbf{y}^{\text{in}})$, where the superscript *trn* and *in* represents the general training points and initial training points, respectively.

4.3. ADAPTIVE TRAINING

With the initial training points $(\mathbf{x}^{\text{in}}, \mathbf{t}^{\text{in}}, \mathbf{y}^{\text{in}})$, we can build an initial GP $\hat{G}(\mathbf{X}, t)$ to approximate $G(\mathbf{X}, t)$. The initial $\hat{G}(\mathbf{X}, t)$ is generally not accurate. The task of adaptive training is to add training points to refine $\hat{G}(\mathbf{X}, t)$ sequentially and adaptively. Specifically, a task-oriented learning function and stopping criterion are developed.

For numerical computation, $[0, T]$ is evenly discretized into m points $\mathbf{t} = (t_1, t_2, \dots, t_m)^T$. Then $\tau(\mathbf{x})$ is approximated by

$$\hat{\tau}(\mathbf{x}) = \min\{t \in \mathbf{t} \mid \mu_{\hat{G}}(\mathbf{x}, t) \leq 0\} \quad (5)$$

To estimate $\bar{\tau}$, we first randomly generate n_s samples \mathbf{x}^s of \mathbf{X} . Then $\bar{\tau}$ is approximated by

$$\bar{\tau} = \frac{1}{n_s} \sum_{i=1}^{n_s} \hat{\tau}(\mathbf{x}^{(i)}) \quad (6)$$

where $\mathbf{x}^{(i)}$ is the i^{th} random sample of \mathbf{X} . Eq. (6) can yield accurate $\bar{\tau}$ when two conditions are satisfied. First, the sample size n_s is sufficiently large. How to determine n_s will be given in Subsection 4.3. Second, the model $\hat{\tau}(\mathbf{X})$ is accurate at all the samples \mathbf{x}^s . How to add training samples to refine $\hat{G}(\mathbf{x}, t)$ so that the second condition is satisfied is the key to the adaptive training.

Intuitively, $\hat{\tau}(\mathbf{X})$ is accurate as long as $\mu_{\hat{G}}(\mathbf{X}, t)$ approximates $G(\mathbf{X}, t)$ accurately. However, training $\hat{G}(\mathbf{X}, t)$ in this way is not efficient and it disobeys the task-oriented rule. In fact, $t^* \in \mathbf{t}$ is an accurate solution to Eq. (5) as long as the signs of

$\{G(\mathbf{x}, t) | t \in \mathbf{t}, t \leq t^*\}$ are predicted accurately. For example, if $\hat{G}(\mathbf{x}, t)$ can accurately predict the signs of $G(\mathbf{x}, t_j), j = 1, 2, 3, 4, 5$ as $(+, +, +, +, -)$, then t_5 is the accurate solution to Eq. (5). We do not need to care if $\hat{G}(\mathbf{x}, t)$ predicts the specific values of $G(\mathbf{x}, t_j), j = 1, 2, 3, 4, 5$ or the signs of $G(\mathbf{x}, t_j), j \geq 6$ accurately. Note that in this example the exact solution to Eq. (5) should be in the interval $[t_4, t_5]$, but we can all the same select t_5 as the solution without losing significant accuracy as long as m is sufficiently large.

The well-known learning function U [28] is used to measure how accurate the sign at a point is predicted. It is given by

$$U(\mathbf{x}, t) = \frac{|\mu_{\hat{G}}(\mathbf{x}, t)|}{\sigma_{\hat{G}}(\mathbf{x}, t)} \quad (7)$$

To refine $\hat{G}(\mathbf{X}, t)$, we should add training points where the accuracy is poor or U is small since a small U means that the chance of correctly predicting the sign of $G(\mathbf{x}, t)$ is small. If \mathbf{X} is fixed to \mathbf{x} , the next training point $(\mathbf{x}, t^{\text{next}})$ is determined by

$$(\mathbf{x}, t^{\text{next}}) = \arg \min_{t \in \mathbf{t}, t \leq \hat{\tau}(\mathbf{x})} U(\mathbf{x}, t) \quad (8)$$

Since there are n_s samples of \mathbf{X} , Eq. (8) determines n_s points. Among them, the point with minimal U is finally selected as the next training point $(\mathbf{x}^{\text{next}}, t^{\text{next}})$, which is determined by

$$(\mathbf{x}^{\text{next}}, t^{\text{next}}) = \arg \min_{\mathbf{x} \in \mathbf{x}^s, t \in \mathbf{t}, t \leq \hat{\tau}(\mathbf{x})} U(\mathbf{x}, t) \quad (9)$$

With the learning function given in Eq. (9), we can add training points to update $(\mathbf{x}^{\text{trn}}, \mathbf{t}^{\text{trn}}, \mathbf{y}^{\text{trn}})$ and $\hat{G}(\mathbf{X}, t)$ sequentially until a stopping criterion is satisfied.

The direct use of $U(\mathbf{x}, t)$ and hence Eq. (9), however, may result in duplicate training points. In other words, the next training point determined by Eq. (9) may be the

one among $(\mathbf{x}^{\text{trn}}, \mathbf{t}^{\text{trn}}, \mathbf{y}^{\text{trn}})$. Once this happens, the adaptive training fails. Theoretically, because GP is an exact interpolator, if a point (\mathbf{x}^*, t^*, y^*) is among the training set $(\mathbf{x}^{\text{trn}}, \mathbf{t}^{\text{trn}}, \mathbf{y}^{\text{trn}})$, $\hat{G}(\mathbf{X}, t)$ will predict $G(\mathbf{x}^*, t^*)$ exactly as y^* , i.e., $\mu_{\hat{G}}(\mathbf{x}^*, t^*) = y^*$ and $\sigma_{\hat{G}}(\mathbf{x}^*, t^*) = 0$. As a result, $U(\mathbf{x}^*, t^*) = +\infty$, (\mathbf{x}^*, t^*) will never be selected by Eq. (9) as the next training point, and the duplicate training points will never be encountered. However, due to the numerical error, $\sigma_{\hat{G}}(\mathbf{x}^*, t^*)$ is not exactly zero but a small positive number. In this case, if $\mu_{\hat{G}}(\mathbf{x}^*, t^*)$ is smaller than $\sigma_{\hat{G}}(\mathbf{x}^*, t^*)$, we will have $U(\mathbf{x}^*, t^*) < 1$, and Eq. (9) may select (\mathbf{x}^*, t^*) as the next training point, leading to duplicate training points.

Another problem caused by U is that added training points may cluster together [19]. It will make the correlation matrix of GP ill-conditioned. If this happens, some of the clustered training points will have a negligible effect on the refinement of $\hat{G}(\mathbf{X}, t)$, and adaptive training may not converge. Hu and Mahadevan [19] proposed to disqualify those points to be candidate training points if they are highly correlated with any one of the existing training points. Specifically, the candidate training points are shrunk from the point set $\mathbf{x}^s \times \mathbf{t}$ to $\left\{ (\mathbf{x}, t) \in \mathbf{x}^s \times \mathbf{t} \mid \max_{(\mathbf{x}', t') \in (\mathbf{x}^{\text{trn}}, \mathbf{t}^{\text{trn}})} \rho[(\mathbf{x}, t), (\mathbf{x}', t')] < \eta \right\}$, where $\rho(\cdot, \cdot)$ is the correlation coefficient used in GP to describe the correlation of two points, and η is a hyperparameter. It guarantees that the candidate training points are sufficiently far away from the current training points, and thereby that the newly selected training point will not overlap or cluster with any one of the current training points.

We employ this method and then the learning function proposed in Eq. (9) is updated to

$$(\mathbf{x}^{\text{next}}, t^{\text{next}}) = \arg \min_{t \leq \hat{\tau}(\mathbf{x}), (\mathbf{x}, t) \in \mathbf{C}} U(\mathbf{x}, t) \quad (10)$$

where $\mathbf{C} = \left\{ (\mathbf{x}, t) \in \mathbf{x}^s \times \mathbf{t} \mid \max_{(\mathbf{x}', t') \in (\mathbf{x}^{\text{trn}}, \mathbf{t}^{\text{trn}})} \rho[(\mathbf{x}, t), (\mathbf{x}', t')] < \eta \right\}$.

In addition to the learning function, the other important component of adaptive training is the stopping criterion. Since the learning function can add training points iteratively to update $\hat{G}(\mathbf{X}, t)$, and hence $\hat{\tau}(\mathbf{x})$ in Eq. (5), a stopping criterion is necessary to terminate the iteration. Once the model $\hat{\tau}(\mathbf{X})$ is accurate on all the samples \mathbf{x}^s , we no longer add new training points. Therefore, the iteration ends if the following condition is satisfied

$$W > w \quad (11)$$

where $W = \min_{t \leq \hat{\tau}(\mathbf{x}), (\mathbf{x}, t) \in \mathbf{C}} U(\mathbf{x}, t)$, and w is a hyperparameter and is recommended to set to

2. Generally, the larger is w , the more accurate will $\bar{\tau}$ be. Larger w , however, will lower the efficiency, so the selection of w needs a tradeoff. There is no rigorous theory to determine the best w , and we recommend 2 based on both our experience from many experiments and [28].

4.4. ADAPTIVE SAMPLE SIZE

Since the random sampling method is used to estimate $\bar{\tau}$ through Eq. (6), it is desirable to select a good sample size n_s . We use an initial sample size n_0 and then adaptively increase the sample size until $\bar{\tau}$ is obtained with sufficiently high fidelity [33].

Since $\tau(\mathbf{X})$ is a random variable, the sample size needed to estimate its mean value $\bar{\tau}$ is dependent on its standard deviation σ_τ . With the sample size n_s , the deviation coefficient Γ of $\bar{\tau}$ is given by

$$\Gamma = \frac{\sigma_\tau}{\bar{\tau}\sqrt{n_s}} \quad (12)$$

where $\bar{\tau}$ is estimated by Eq. (6) and σ_τ is estimated by

$$\sigma_\tau = \sqrt{\frac{1}{n_s - 1} \sum_{i=1}^{n_s} [\hat{t}(\mathbf{x}^{(i)}) - \bar{\tau}]^2} \quad (13)$$

Eq. (12) shows that the larger is n_s , the smaller Γ will we have. A smaller Γ means that $\bar{\tau}$ is more accurately estimated by Eq. (6). $\bar{\tau}$ is said to be accurate if the following condition is satisfied

$$\Gamma \leq \gamma \quad (14)$$

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

If the current n_s cannot satisfy Eq. (14), we should increase it. Combining Eq. (12) and Eq. (14), we have

$$n_s \geq \left(\frac{\sigma_\tau}{\bar{\tau}\gamma} \right)^2 \quad (15)$$

It means that at least a sample size of $\left(\frac{\sigma_\tau}{\bar{\tau}\gamma} \right)^2$ is necessary to guarantee Eq. (14). Let $n_1 = \text{ceil} \left[\left(\frac{\sigma_\tau}{\bar{\tau}\gamma} \right)^2 \right]$, where $\text{ceil}(\cdot)$ represents the operation to get the nearest larger integer. Then

the number n_{add} by which n_s should be increased is given by

$$n_{\text{add}} = n_1 - n_s \quad (16)$$

However, when $\hat{G}(\mathbf{X}, t)$ is too rough at the first several adaptive training iterations, both $\bar{\tau}$ and σ_τ may have poor accuracy, and n_{add} given in Eq. (16) may be misleading. To deal with this issue, we set a threshold \tilde{n}_{add} for n_{add} . Then Eq. (16) is updated to

$$n_{\text{add}} = \begin{cases} \tilde{n}_{\text{add}}, & \text{if } n_1 - n_s > \tilde{n}_{\text{add}} \\ n_1 - n_s, & \text{otherwise} \end{cases} \quad (17)$$

Since it is cheap to compute samples of $\hat{t}(\mathbf{X})$, \tilde{n}_{add} is not a key hyperparameter of the proposed method, and generally, it is good to set \tilde{n}_{add} to 1,000, according to our experience from many experiments.

4.5. IMPLEMENTATION

In this subsection, we give a detailed procedure of the proposed method. The full flowchart is shown in Figure 3. The total number n_e of function evaluations of $G(\mathbf{X}, t)$ is used to measure the main computational cost of the proposed method.

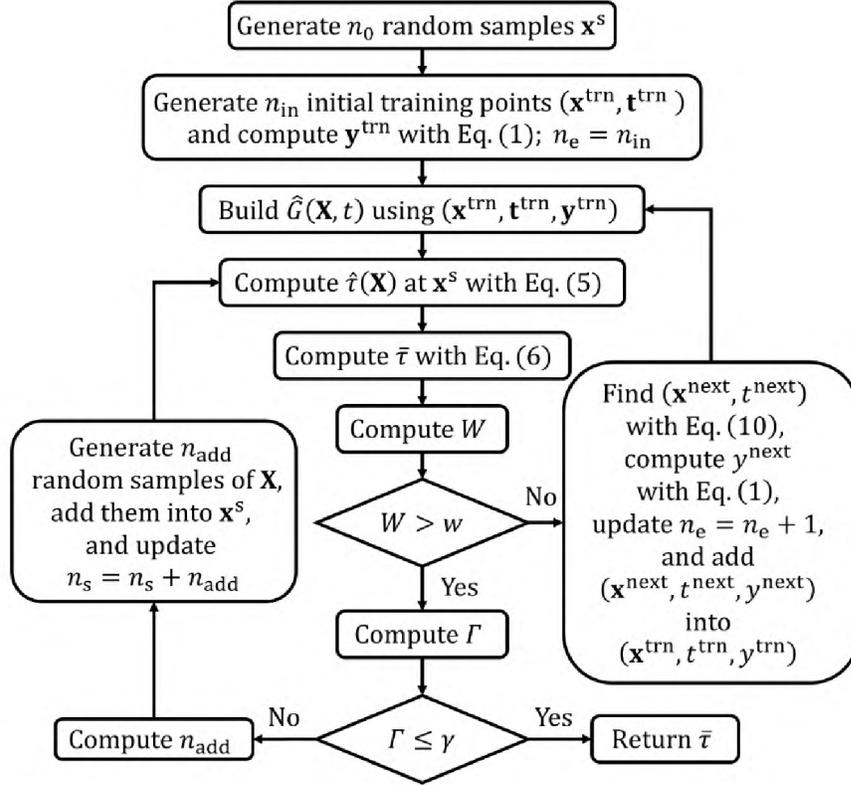


Figure 3. Detailed flowchart of the proposed method

5. EXTENSION TO PROBLEMS WITH INPUT RANDOM PROCESSES

When the limit-state function $G(\cdot)$ has input random processes, it is straightforward to employ the series expansion methods of the random processes so that the above implementation of the proposed method can still work.

Let $\mathbf{H}(t)$ represents a vector of random processes, then the limit-state function is given by

$$Y = G(\mathbf{X}, \mathbf{H}(t), t) \quad (18)$$

To easily present the idea, we assume there is only one random process $H(t)$. Widely used series expansions for random fields include the Karhunen-Loeve series expansion (K-L), the orthogonal series expansion (OSE), and the expansion optimal linear estimation method (EOLE) [34]. Since t is discretized into \mathbf{t} , the autocorrelation coefficient function of $H(t)$ is discretized into the autocorrelation coefficient matrix \mathbf{M}_H with dimension $m \times m$. Then the EOLE expansion $H(\boldsymbol{\xi}, t)$ of $H(t)$ is given by

$$H(\boldsymbol{\xi}, t) = \mu_H(t) + \sigma_H(t) \sum_{k=1}^m \frac{\xi_k}{\sqrt{\lambda_k}} \mathbf{v}_k \mathbf{M}_H(:, k), t \in \mathbf{t} \quad (19)$$

where $\mu_H(t)$ is the mean value function of $H(t)$, $\sigma_H(t)$ is the standard deviation function of $H(t)$, $\xi_k, k = 1, 2, \dots, m$ are m independent standard Gaussian variables, λ_k is the k -th eigenvalue of \mathbf{M}_H , \mathbf{v}_k is the k -th (row) eigenvector of \mathbf{M}_H , and $\mathbf{M}_H(:, k)$ is the k -th column of \mathbf{M}_H . Note that the eigenvalues are sorted from the largest to the smallest. Usually only the first m' ($m' \leq m$) eigenvalues are significant. Therefore, Eq. (19) is practically truncated, and only the first m' orders are kept:

$$H(\boldsymbol{\xi}, t) = \mu_H(t) + \sigma_H(t) \sum_{k=1}^{m'} \frac{\xi_k}{\sqrt{\lambda_k}} \mathbf{V}_k \mathbf{M}_H(:, k), t \in \mathbf{t} \quad (20)$$

With the truncated expansion in Eq. (20), Eq. (18) is rewritten as

$$Y = G(\mathbf{X}, H(\boldsymbol{\xi}, t), t) \quad (21)$$

or equivalently as

$$Y = G(\tilde{\mathbf{X}}, t) \quad (22)$$

where $\tilde{\mathbf{X}} = (\boldsymbol{\xi}, \mathbf{X})$. Eq. (22) shares the same format with Eq. (1) and hence the implementation given in Subsection 4.5 also works.

The direct implementation this way, however, may suffer from the curse of dimensionality. Since there are many random variables, i.e. $\boldsymbol{\xi}$, in the series expansion $H(\boldsymbol{\xi}, t)$, the dimension of $\boldsymbol{\xi}$ and hence that of $G(\tilde{\mathbf{X}}, t)$ is high. As a result, the dimension of $\hat{G}(\tilde{\mathbf{X}}, t)$ is also high. The high dimensionality has at least two drawbacks. First, it is not computationally cheap anymore, losing its expected advantages. Second, more training points are needed to train the GP. To overcome the drawbacks, we build a GP $\hat{G}(\mathbf{X}, H, t)$ with respect to \mathbf{X} , H , and t [19, 33]. Note that the entire random process H is treated as only one variable for $\hat{G}(\mathbf{X}, H, t)$. Then the surrogate model $\hat{G}(\tilde{\mathbf{X}}, t)$ with respect to $\tilde{\mathbf{X}}$ and t is obtained through

$$\hat{G}(\tilde{\mathbf{X}}, t) = \hat{G}[\mathbf{X}, H(\boldsymbol{\xi}, t), t] \quad (23)$$

Since the truncated series expansion $H(\boldsymbol{\xi}, \mathbf{Z})$ in Eq. (20) has a simple closed-form expression, if $\hat{G}(\mathbf{X}, H, t)$ is accurate and efficient, so will be $\hat{G}(\tilde{\mathbf{X}}, t)$ in Eq. (23). Since the dimension of $\hat{G}(\mathbf{X}, H, t)$ is $(m' - 1)$ lower than that of $\hat{G}(\tilde{\mathbf{X}}, t)$, it is more efficient to train $\hat{G}(\mathbf{X}, H, t)$. To build $\hat{G}(\mathbf{X}, H, t)$, we need the training points \mathbf{h}^{trn} of H . \mathbf{h}^{trn} can be obtained

simply by substituting $(\boldsymbol{\xi}^{\text{trn}}, \mathbf{t}^{\text{trn}})$ into Eq. (20). Similarly, when $(\tilde{\mathbf{x}}^{(\text{next})}, t^{(\text{next})})$ is determined by Eq. (10), the next training point $h^{(\text{next})}$ of H is obtained by substituting $(\boldsymbol{\xi}^{(\text{next})}, \mathbf{z}^{(\text{next})})$ into Eq. (20). Note that $\tilde{\mathbf{x}}^{(\text{next})} = (\boldsymbol{\xi}^{(\text{next})}, \mathbf{x}^{(\text{next})})$. When multiple input random processes are involved, the procedure of building and updating the surrogate model \hat{G} is similar.

6. EXAMPLES

In this section, we use three examples to illustrate the proposed method. The first one is a math example with only one input random variable. It is designed to graphically show the procedure of the proposed method. The second one is an engineering example with both input random variables and a random process. The third one is an engineering example where the limit-state function is a black box using the finite element method (FEM) and where there are five input random processes.

All the three examples share the same values of the following parameters: $m = 100$, $w = 2$, $\eta = 0.95$, $\gamma = 0.005$, and $\tilde{n}_{\text{add}} = 1,000$. MCS is also used to evaluate MTTF; it calls the original limit-state function in Eq. (1) directly to get samples of $\tau(\mathbf{X})$, and hence the mean lifetime $\bar{\tau}$. The sample size n_{MCS} of MCS is set to 10^5 . The results of MCS are treated as accurate solutions for the accuracy comparison. Both the proposed method and MCS share the same discretization of $t \in [0, T]$.

6.1. EXAMPLE 1: A MATH EXAMPLE

The limit-state function is given by

$$Y = \exp(-0.05t)\cos(0.25t + X), t \in [0,40] \quad (24)$$

where X is a standard uniform variable. With the Hammersley sampling method, we get $n_{\text{in}} = 5$ initial training points in $[0,1]^2$. They are assembled in a matrix \mathbf{M}

$$\mathbf{M} = \begin{bmatrix} 0 & 0.5 \\ 0.2 & 0.25 \\ 0.4 & 0.75 \\ 0.6 & 0.125 \\ 0.8 & 0.625 \end{bmatrix} \quad (25)$$

The first column of \mathbf{M} is mapped to the interval $[0, T]$ of t , and then we get the initial training points $\mathbf{t}^{\text{in}} = (0, 8, 16, 24, 32)^T$. The second column is mapped to the interval $[0, 1]$ of X , and then we get the initial training points $\mathbf{x}^{\text{in}} = (0.5, 0.25, 0.75, 0.125, 0.625)^T$. Substituting the five training points $(\mathbf{x}^{\text{in}}, \mathbf{t}^{\text{in}})$ into Eq. (1), we get five training points $\mathbf{y}^{\text{in}} = (0.8776, -0.4211, 0.0169, 0.2974, -0.1407)^T$ of Y .

Eq. (1) has been evaluated 5 times so far, and therefore currently $n_e = 5$. With the training points $(\mathbf{x}^{\text{trn}}, \mathbf{t}^{\text{trn}}, \mathbf{y}^{\text{trn}}) = (\mathbf{x}^{\text{in}}, \mathbf{t}^{\text{in}}, \mathbf{y}^{\text{in}})$, $\hat{G}(X, t)$ is built. Then more and more training points determined by the learning function in Eq. (10) are added one by one into the training set $(\mathbf{x}^{\text{trn}}, \mathbf{t}^{\text{trn}}, \mathbf{y}^{\text{trn}})$ to refine $\hat{G}(X, t)$. The sample size n_s is also increased adaptively from the initial value $n_0 = 1,000$. After the algorithm converges, eight training points are added, and n_e is finally updated to $5 + 6 = 11$. n_s is finally increased to 2,632.

Figure 4 shows the actual contours of the limit-state function, as well as the training points. Three contours are indicating $Y = 0$. For each value of X , Eq. (3) has three roots. However, we need only the first root. In other words, we need the GP to accurately predict only the first contour. With the proposed learning function in Eq. (10), almost all adaptive

training points are added near the first contour. It helps the GP efficiently find the first root, i.e., $\tau(X)$, without putting unnecessary computational effort in improving the GP in the unimportant area. This is an expected good property of the proposed task-oriented adaptive training.

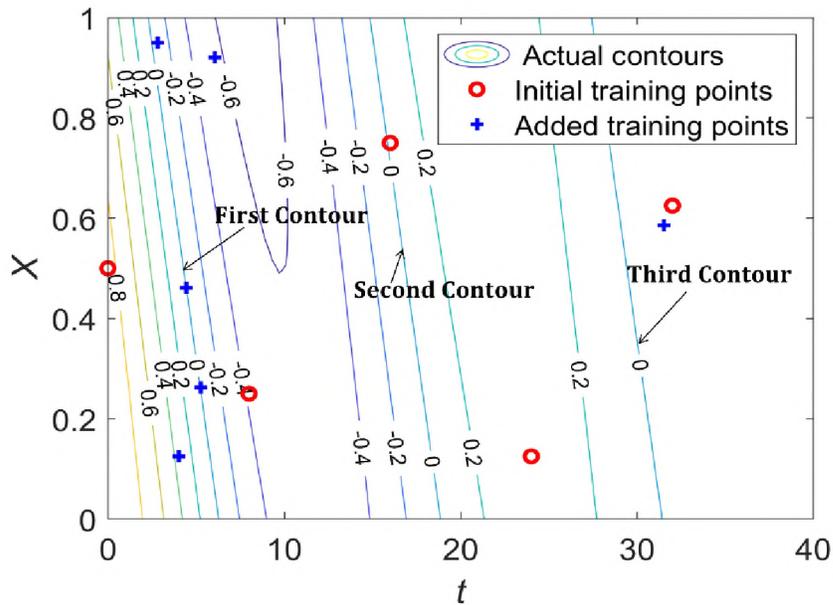


Figure 4. Contours and training points

The results are given in Table 1. The MTTF estimated by the proposed method is 4.48, and that estimated by MCS is 4.49. The relative error is -0.2% , showing the high accuracy of the proposed method. Besides, the proposed method only evaluates the limit-state function 11 times, far less than 10^7 times by MCS, showing the high efficiency of the proposed method.

Table 1. Results of Example 1

Methods	Proposed	MCS
$\bar{\tau}$	4.48	4.49
Relative error	-0.2%	-
n_e	11	10^7

6.2. EXAMPLE 2: A SIMPLY SUPPORTED BEAM

This example is modified from an example in [35]. Shown in Figure 5 is a simply supported beam subjected to two random loads. The cross-section A-A is rectangular with width a and height b . Due to corrosion, both a and b decrease with time t and are given by

$$a = a_0 \exp(-0.02t) \quad (26)$$

and

$$b = b_0 \exp(-0.02t) \quad (27)$$

where a_0 and b_0 are their initial values.

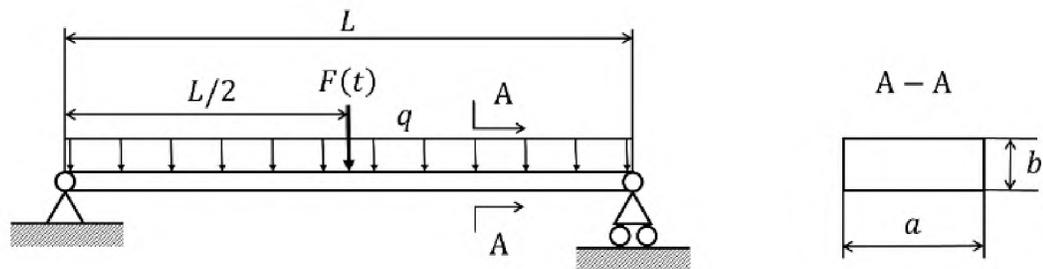


Figure 5. A simply supported beam [35]

A stationary random process load $F(t)$ acts at the midpoint of the beam. The beam is also subjected to a constant weight load and a load q , which is uniformly distributed on the top surface of the beam. The autocorrelation coefficient functions of $F(t)$ is given by

$$\rho(t_1, t_2) = \exp\left[-\left(\frac{t_1 - t_2}{5}\right)^2\right] \quad (28)$$

A failure occurs once the stress exceeds the ultimate strength. The limit-state function is given by

$$Y = -0.25F(t)L - 0.125qL^2 - 0.125\rho a_0 b_0 L^2 + 0.25(a_0 - 2kt)(a_0 - 2kt)^2\sigma \quad (29)$$

where σ is the ultimate strength, $\rho = 78.5 \text{ kg/m}^3$ is the density of the beam, $L = 5 \text{ m}$ is the length of the beam, and $t \in [0, 20]$ yr. Table 2 gives all random variables. n_{in} and n_0 are set to 10 and 1,000, respectively. We use six random variables for the EOLE expansion of $F(t)$.

Table 2. Variables of Example 2

Variable	Mean	Standard deviation	Distribution	Autocorrelation
a_0	0.2 m	0.002 m	Gaussian	N/A
b_0	0.04 m	0.004 m	Gaussian	N/A
σ	0.24 GPa	0.0024 GPa	Gaussian	N/A
$F(t)$	5,000 N	500 N	Stationary Gaussian process	Eq. (28)
q	450 N/m	50 N/m	Gaussian	N/A

The results are given in Table 3. The MTTF evaluated by the proposed method is 11.61 years, with a relative error of -0.4% . Besides, the proposed method only cost 23 limit-state function evaluations, which is much cheaper than MCS.

Table 3. Results of Example 2

Methods	Proposed	MCS
\bar{t}	11.61 yr	11.66 yr
Relative error	-0.4%	-
n_e	23	10^7

6.3. EXAMPLE 3: A 52-BAR SPACE TRUSS

This example is modified from an example in [36]. Shown in Figure 6 is a 52-bar space truss with 21 nodes. To distinguish the node numbers and the bar numbers, we add a decimal point after all node numbers in Figure 6. All the nodes are located on the surface of an imaginary hemisphere whose radius is $r = 240$ in. The cross-sectional areas of Bars 1~8 and 29~36 are 2 in^2 . The cross-sectional areas of Bars 9~16 and other bars are 1.2 in^2 and 0.6 in^2 , respectively. The Young's modulus of all bars is E , which is a lognormal random variable with mean and standard deviation being 25,000 ksi and 25 ksi, respectively. Nodes 1~5 are subjected to external loads $F_1(t) \sim F_5(t)$, all in the $-z$ direction. The five loads are Gaussian processes. They are independent of each other with the following autocorrelation coefficient function:

$$\rho(t_1, t_2) = \exp\left[-\left(\frac{t_1 - t_2}{5}\right)^2\right] \quad (30)$$

where $t_1, t_2 \in [0, 10]$ yr. $F_2(t) \sim F_5(t)$ are all stationary processes whose mean and standard deviation are 50 kip and 1 kip, respectively. $F_1(t)$ is nonstationary, with standard deviation being 1 kip, and mean value $\mu_1(t)$ given by

$$\mu_1(t) = 50\exp(0.02t) \text{ kip} \quad (31)$$

where $t \in [0, 10]$ yr.

A failure occurs when the displacement δ of Node 1 in $-z$ direction exceeds a threshold $\delta_0 = 1.3$ in. The limit-state function is given by

$$Y(t) = \delta_0 - \delta(E, \mathbf{F}) \quad (32)$$

where $\mathbf{F} = [F_1, F_2, F_3, \dots, F_5]$ is the vector of all loads. $\delta(E, \mathbf{F})$ is calculated by FEM, and the linear bar element is used.

n_{in} and n_0 are set to 10 and 1,000, respectively. We use six random variables in the EOLE expansion of each random load. The results are given in Table 4. The mean lifetime evaluated by the proposed method is 4.79 years with a relative error of 0.8%. Besides, the proposed method costs 56 limit-state function evaluations and is much more efficient than MCS.

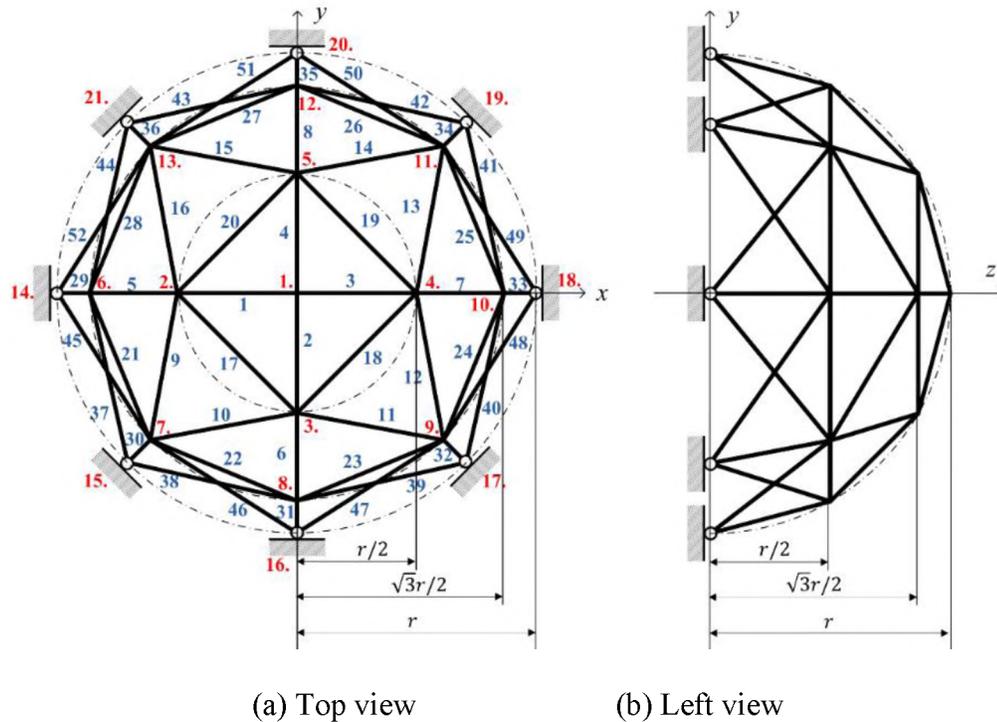


Figure 6. A 52-bar truss [36]

Table 4. Results of Example 3

Methods	Proposed	MCS
$\bar{\tau}$	4.79 yr	4.75 yr
Relative error	0.8%	-
n_e	56	10^7

7. CONCLUSIONS

The mean time to failure (MTTF) is an important measure of product reliability. This study demonstrates that MTTF can be predicted computationally by a physics-based method. If a failure mode of the product is well understood and can be modeled mathematically, a limit-state function is available, and the physics-based method can then be used. It is in general much more efficient and cheaper than statistics-based methods.

This study also demonstrates that ML is a powerful tool to assist the prediction of the MTTF, which requires a large number of calls of the limit-state function. The results indicate that the proposed Gaussian process-based adaptive training is effective to predict the MTTF. The key to the learning algorithm is the learning function that is specially designed for adaptive training. Three examples have shown the high accuracy and efficiency of the proposed method.

The proposed method can only accommodate one failure mode. If there are multiple failure modes, the MTTF will depend on the limit-state functions of the failure modes and their relationships, for instance, whether they are in parallel or series, and this will involve time-dependent system reliability analysis, where ML can play a more significant role. Our

future work will include developing physics-based ML algorithms for multiple Gaussian process responses so that multiple limit-state functions can be handled.

ACKNOWLEDGEMENTS

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

REFERENCES

- [1] O'Connor P, Kleyner A. Practical reliability engineering: John Wiley & Sons; 2012.
- [2] Henley EJ, Kumamoto H. Reliability engineering and risk assessment: Prentice-Hall Englewood Cliffs (NJ); 1981.
- [3] Birolini A. Reliability engineering: theory and practice: Springer Science & Business Media; 2013.
- [4] Zio E. Reliability engineering: Old problems and new challenges. Reliability Engineering & System Safety. 2009;94:125-41.
- [5] Pham H. Handbook of reliability engineering: Springer Science & Business Media; 2006.
- [6] Rausand M, Høyland A. System reliability theory: models, statistical methods, and applications: John Wiley & Sons; 2004.
- [7] Meeker WQ, Escobar LA. Statistical methods for reliability data: John Wiley & Sons; 2014.
- [8] Lawless J. Statistical methods in reliability. Technometrics. 1983;25:305-16.
- [9] Viertl R. Statistical methods in accelerated life testing: Vandenhoeck & Ruprecht; 1988.
- [10] Ditlevsen O, Madsen HO. Structural reliability methods. New York: Wiley; 1996.

- [11] Hu Z, Du X. Integration of Statistics-and Physics-Based Methods—A Feasibility Study on Accurate System Reliability Prediction. *Journal of Mechanical Design*. 2018;140.
- [12] Wang Z, Wang P. A Nested Extreme Response Surface Approach for Time-Dependent Reliability-Based Design Optimization. *Journal of Mechanical Design*. 2012;134:121007-14.
- [13] Du X, Chen W. Sequential optimization and reliability assessment method for efficient probabilistic design. *Journal of Mechanical Design*. 2004;126:225-33.
- [14] Dubourg V, Sudret B, Bourinet J-M. Reliability-based design optimization using kriging surrogates and subset simulation. *Structural and Multidisciplinary Optimization*. 2011;44:673-90.
- [15] Liu X, Wu Y, Wang B, Ding J, Jie H. An adaptive local range sampling method for reliability-based design optimization using support vector machine and Kriging model. *Structural and Multidisciplinary Optimization*. 2017;55:2285-304.
- [16] Papadrakakis M, Lagaros ND. Reliability-based structural optimization using neural networks and Monte Carlo simulation. *Computer Methods in Applied Mechanics and Engineering*. 2002;191:3491-507.
- [17] Hu Z, Du X. Time-dependent reliability analysis with joint upcrossing rates. *Structural and Multidisciplinary Optimization*. 2013;48:893-907.
- [18] Andrieu-Renaud C, Sudret B, Lemaire M. The PHI2 method: a way to compute time-variant reliability. *Reliability Engineering & System Safety*. 2004;84:75-86.
- [19] Hu Z, Mahadevan S. A single-loop kriging surrogate modeling for time-dependent reliability analysis. *Journal of Mechanical Design*. 2016;138:061406.
- [20] Wei X, Du X. Uncertainty Analysis for Time-and Space-Dependent Responses With Random Variables. *Journal of Mechanical Design*. 2019;141:021402.
- [21] Zienkiewicz OC, Taylor RL, Nithiarasu P, Zhu J. *The finite element method*. London: McGraw-hill; 1977.
- [22] Jiang Z, Li W, Apley DW, Chen W. A spatial-random-process based multidisciplinary system uncertainty propagation approach with model uncertainty. *Journal of Mechanical Design*. 2015;137.
- [23] Xi Z. Model-based reliability analysis with both model uncertainty and parameter uncertainty. *Journal of Mechanical Design*. 2019;141.

- [24] Stern RE, Song J, Work DB. Accelerated Monte Carlo system reliability analysis through machine-learning-based surrogate models of network connectivity. *Reliability Engineering & System Safety*. 2017;164:1-9.
- [25] Kotsiantis SB, Zaharakis I, Pintelas P. Supervised machine learning: A review of classification techniques. *Emerging Artificial Intelligence Applications in Computer Engineering*. 2007;160:3-24.
- [26] Williams CK, Rasmussen CE. *Gaussian processes for machine learning*: MIT Press Cambridge, MA; 2006.
- [27] Lophaven SN, Nielsen HB, Søndergaard J. *DACE: a Matlab kriging toolbox*: Citeseer; 2002.
- [28] Echard B, Gayton N, Lemaire M. AK-MCS: an active learning reliability method combining Kriging and Monte Carlo simulation. *Structural Safety*. 2011;33:145-54.
- [29] Eliason SR. *Maximum likelihood estimation: Logic and practice*: Sage; 1993.
- [30] Mooney CZ. *Monte carlo simulation*. Thousand Oaks: Sage Publications; 1997.
- [31] Chen W, Tsui K-L, Allen JK, Mistree F. Integration of the response surface methodology with the compromise decision support problem in developing a general robust design procedure. *ASME Design Engineering Technical Conference*. 1995:485-92.
- [32] Hosder S, Walters R, Balch M. Efficient sampling for non-intrusive polynomial chaos applications with multiple uncertain input variables. 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 2007. p. 1939.
- [33] Wei X, Du X. Robustness Metric for Robust Design Optimization Under Time-and Space-Dependent Uncertainty Through Metamodeling. *Journal of Mechanical Design*. 2020;142.
- [34] Sudret B, Der Kiureghian A. *Stochastic finite element methods and reliability: a state-of-the-art report*: Department of Civil and Environmental Engineering, University of California Berkeley; 2000.
- [35] Hu Z, Du X. First order reliability method for time-variant problems using series expansions. *Structural and Multidisciplinary Optimization*. 2015;51:1-21.
- [36] Zhang Z, Jiang C, Han X, Ruan X. A high-precision probabilistic uncertainty propagation method for problems involving multimodal distributions. *Mechanical Systems & Signal Processing*. 2019;126:21-41.

III. UNCERTAINTY ANALYSIS FOR TIME- AND SPACE-DEPENDENT RESPONSES WITH RANDOM VARIABLES

ABSTRACT

The performance of a product varies with respect to time and space if the associated limit-state function involves time and space. This study develops an uncertainty analysis method that quantifies the effect of random input variables on the performance (response) over time and space. The combination of the first-order reliability method (FORM) and the second-order reliability method (SORM) is used to approximate the extreme value of the response with respect to space at discretized instants of time. Then the response becomes a Gaussian stochastic process that is fully defined by the mean, variance, and autocorrelation functions obtained from FORM and SORM, and a sequential single-loop procedure is performed for spatial and random variables. The method is successfully applied to the reliability analysis of a crank-slider mechanism, which operates in a specified period of time and space.

1. INTRODUCTION

Uncertainty, which is a gap between the present state of knowledge and the complete knowledge [1], exists in all stages of product development and operation [2]. Examples of uncertainty include random material properties, random loading, random operation conditions; they also include random manufacturing imprecision, as well as the lack of knowledge, such as ignorance, assumptions, and simplifications [1]. Numerous

applications and studies have shown that not considering uncertainty properly during the design stage can lead to serious problems, such as low reliability, low robustness, low customer satisfaction, high risk, and high lifecycle cost [1, 3-5].

Reliability methods provide useful tools for uncertainty quantification and management. This is because reliability is not only an important quality characteristic of a product, but also related to other characteristics such as robustness, risk, safety, maintainability, and cost. Reliability is usually quantified by the probability that a product performs its intended function over a specified period of time and under specified service conditions [6]. Reliability problems can be roughly grouped into four categories: (a) time- and space-independent (TSI) problems, (b) space-dependent (SD) problems, (c) time-dependent (TD) problems, and (d) time- and space-dependent (TSD) problems. TSD problems belong to the most general category since the other three types are just special cases of the TSD category.

TSI problems are the most traditional problems. They involve only time- and space-independent random variables, such as the geometry or material properties of a structure and applied loads. The responses are also random variables. Reliability methods for TSI problems include, but are not limited to, analytical methods, surrogate model methods, moment methods, and simulation methods. Typical analytical methods include the first-order reliability method (FORM) and the second-order reliability method (SORM) [7-12]. FORM and SORM simplify a limit-state function, which specifies a functional relationship between a response and random input variables, using the first and second-order Taylor series expansions, respectively, at the so-called most probable point (MPP) [13]. Surrogate model methods [14-16] use simplified models, which are generally obtained from the

design of experiments or variable screening using sensitivity analysis, to improve the computation efficiency. Moment methods [13, 17] calculate the moments of the limit-state function and then approximate its distribution with the moments; and then the distribution is used to obtain the reliability. Simulation methods include the direct Monte Carlo simulation (MCS) [18], quasi-Monte Carlo simulation [19], importance sampling [20], and subset simulation [21]. Usually, simulation methods are accurate but computationally expensive.

SD problems have responses that are space dependent. This happens when either input variables are spatially distributed with random fields [22] or the response is a function of spatial variables. Structural reliability analysis for this kind of problem usually requires stochastic finite element methods [22, 23].

Another dimension on which the uncertainty may depend is time. This happens when the response is a function of time or input variables, such as material properties and loads, which are time-variant stochastic processes. For these TD problems, many methodologies are available, including upcrossing rate methods [24-26], surrogate model methods [27-30], simulation methods [31, 32], probability density evolution method [33], envelope function method [34], failure process decomposition-based method [35], and extreme value moment method [36]. Generally speaking, upcrossing rate methods are the most dominant methods, surrogate methods can obtain accurate results if the surrogate models are well trained, and simulation methods are also accurate but computationally expensive.

The combination of an SD problem and a TD problem leads to a TSD problem where the response is dependent on both space and time. For TSD problems, only a few

methods are available in the literature. Hu and Mahadevan [37, 38] developed a method based on adaptive surrogate modeling. Shi et al. [39] proposed two strategies. One strategy is combining the sparse grid technique with the fourth-moment method. And the other is combining the dimension reduction and maximum entropy method. Shi et al. [40] developed a transferred limit-state function technique to transform the TSD problem into a TSI counterpart. These methods still have limitations for wider applications. Efficiently and accurately dealing with TSD problems remains a challenging issue. There is a need to develop efficient, accurate, and robust methods for TSD problems.

In this work, we aim at developing an efficient and accurate method for a special TSD problem where the response is a function of temporal and spatial variables, as well as random variables. As a result, the response is a time-dependent random field. The main idea is to approximate the extreme value of the response with respect to space at discretized instants of time using the combination of FORM and SORM, thus transforming the TSD response into an equivalent Gaussian stochastic process. The transformation is performed by a sequential single-loop procedure [7, 41-43] so that high efficiency is maintained. The Kriging model method [44] is also employed. Then MCS is employed to estimate the reliability by sampling the Gaussian process.

The rest of the paper is organized as follows: Section 2 discusses the problem addressed in this study, and Section 3 provides an overview of the proposed method followed by the extreme value analysis and the general process in Sections 4 and 5, respectively. Two examples are given in Section 6, and conclusions are made in Section 7.

2. PROBLEM STATEMENT

In this work, we focus on a response that is a function of temporal variables, spatial variables, and random variables. The limit-state function is defined by

$$Y = g(\mathbf{X}, \mathbf{S}, t) \quad (1)$$

where Y is the response, $\mathbf{X} = [X_1, X_2, \dots, X_m]^T$ is an m -dimensional input random vector, $\mathbf{S} = [S_1, S_2, \dots, S_n]^T$ is an n -dimensional spatial variable vector bounded on $[\underline{\mathbf{S}}, \bar{\mathbf{S}}]$, and t is the time bounded on $[\underline{t}, \bar{t}]$.

When $Y < 0$, a failure occurs. The reliability in space $[\underline{\mathbf{S}}, \bar{\mathbf{S}}]$ and time span $[\underline{t}, \bar{t}]$ is then defined by

$$R = \Pr \left\{ g(\mathbf{X}, \mathbf{S}, t) > 0, \forall \mathbf{S} \in [\underline{\mathbf{S}}, \bar{\mathbf{S}}], \forall t \in [\underline{t}, \bar{t}] \right\} \quad (2)$$

where \forall means “for all”.

Since the response is a function of random variables and time, Y is a stochastic process, and it is also a random field because it is a function of random variables and space. As a result, Y is a general time-dependent random field. This kind of TSD problem is commonly encountered in engineering applications. For example, the performance of a mechanism, such as the motion error, is a stochastic process due to random mechanism dimensions and joint clearances. The mechanism may also operate in different locations, and the mechanism performance is also space dependent.

This kind of reliability problem is usually more complicated than TSI, SD, and TD problems since it involves both spatial and temporal variables. In this work, we develop a method to effectively perform uncertainty analysis for TSD problems.

3. OVERVIEW

As mentioned in Section 1, the main idea of the proposed method is to approximate the extreme value of the response with respect to space at discretized instants of time using FORM and SORM, thus transforming the TSD response into an equivalent Gaussian stochastic process. Eq. (2) is converted into

$$R = \Pr \left\{ Y_{\min}(\mathbf{X}, t) = \min_{\mathbf{s} \in [\underline{\mathbf{S}}, \bar{\mathbf{S}}]} g(\mathbf{X}, \mathbf{S}, t) > 0, \forall t \in [\underline{t}, \bar{t}] \right\} \quad (3)$$

where $Y_{\min}(\mathbf{X}, t)$ is the minimum value of $g(\mathbf{X}, \mathbf{S}, t)$ with respect to \mathbf{S} . $Y_{\min}(\mathbf{X}, t)$ is a general stochastic process, and Eq. (3) can be therefore regarded as the reliability of a TD problem. Since it is nearly impossible to simulate the stochastic process $Y_{\min}(\mathbf{X}, t)$ directly, we need to convert it into an equivalent Gaussian process $H(t)$ such that [45]

$$\begin{aligned} R &= \Pr \left\{ Y_{\min}(\mathbf{X}, t) = \min_{\mathbf{s} \in [\underline{\mathbf{S}}, \bar{\mathbf{S}}]} g(\mathbf{X}, \mathbf{S}, t) > 0, \forall t \in [\underline{t}, \bar{t}] \right\} \\ &\approx \Pr \left\{ H(t) > 0, \forall t \in [\underline{t}, \bar{t}] \right\} \end{aligned} \quad (4)$$

A possible way to convert $Y_{\min}(\mathbf{X}, t)$ into $H(t)$ is to employ FORM at every instant of time on $[\underline{t}, \bar{t}]$ as FORM is capable of transforming a non-Gaussian random variable into a Gaussian random variable [45]. However, FORM may result in poor

accuracy when $Y_{\min}(\mathbf{X}, t)$ is highly nonlinear. A better idea is to employ SORM to improve the accuracy, but SORM does not transform a non-Gaussian random variable into a Gaussian one, as what FORM does. To address this problem, we inversely convert the instantaneous reliability obtained by SORM to its equivalent reliability index with which an equivalent Gaussian variable, which is needed for $H(t)$, can be constructed. However, SORM is less efficient than FORM, especially when the dimension of \mathbf{X} is large. To balance the accuracy and efficiency, we use SORM only at time instants where the corresponding instantaneous reliability is relatively small because the accuracy of the instantaneous reliability at those instants is more important.

Calculating $Y_{\min}(\mathbf{X}, t)$ and performing FORM and SORM at every instant of time is impractical. We, therefore, create surrogate models to reduce the number of extreme value analyses and executions of FORM and SORM. Details will be given in Section 5.

After $H(t)$ is numerically obtained, MCS will be implemented to estimate R or the corresponding probability of failure

$$p_f = 1 - R \quad (5)$$

It is worth mentioning that Eq. (2) can also be rewritten as

$$R = \Pr \left\{ \min_{\mathbf{s} \in [\underline{\mathbf{s}}, \bar{\mathbf{s}}], t \in [L, \bar{t}]} g(\mathbf{X}, \mathbf{S}, t) > 0 \right\} \quad (6)$$

which means that the TSD problem can also be transformed into a TSI one, with the minimum value of $g(\mathbf{X}, \mathbf{S}, t)$ with respect to both spatial and temporal variables. But we do not do so for two reasons. First, in many engineering problems, the response $g(\mathbf{X}, \mathbf{S}, t)$

fluctuates significantly with respect to t and may not be a convex function of t . Thus, calculating the minimum value of $g(\mathbf{X}, \mathbf{S}, t)$ with respect to t will involve global optimization, which is in general less computationally efficient. Second, even if $\min_{\mathbf{s} \in [\underline{\mathbf{s}}, \bar{\mathbf{s}}], t \in [\underline{t}, \bar{t}]} g(\mathbf{X}, \mathbf{S}, t)$ can be obtained, the reliability function with respect to t may not be generated, and only the reliability at the end of the period of time under consideration can be obtained. The proposed method can easily produce the reliability function for the entire period of time. Details will be given in Section 6.

4. EXTREME VALUE ANALYSIS AT AN INSTANT OF TIME

In this section, we provide details about how to obtain $H(\tau), \tau \in [\underline{t}, \bar{t}]$. As mentioned in Section 3, to obtain $H(\tau)$, we need to calculate $Y_{\min}(\mathbf{X}, \tau)$ and perform FORM and SORM. In Subsection 4.1, the extreme value analysis using FORM will be given and then in Subsection 4.2 details on how to adaptively update the analysis result using SORM will be described.

4.1. EXTREME VALUE ANALYSIS USING FORM

The extreme value analysis at time instant τ using FORM can be modeled as the following optimization problem [7, 42, 43, 46]:

$$\begin{cases} \min \|\mathbf{U}\| \\ s.t. \min_{\mathbf{s} \in [\underline{\mathbf{s}}, \bar{\mathbf{s}}]} g(T(\mathbf{U}), \mathbf{S}, \tau) = 0 \end{cases} \quad (7)$$

where \mathbf{U} is the vector of standard Gaussian variables transformed from \mathbf{X} , and $T(\cdot)$ stands for the transformation. Eq. (7) indicates a two-layer optimization problem whose solution usually requires a double-loop optimization process. The outer loop is the FORM analysis, and the inner loop is the extreme value analysis. Usually, the double-loop optimization can lead to low efficiency. To improve the efficiency, Du et al. [7, 42, 43] developed a sequential single-loop (SSL) approach to decouple the two loops to a sequential single-loop process. The flow chart of employing SSL to solve the optimization problem in Eq. (7) is shown in Figure 1.

Step 5 involves major equations for the MPP search. β and α are the reliability index and sensitivity vector, respectively, and both are dependent on the specific instant of time τ . Once both $\beta(t)$ and $\alpha(t), t \in [\underline{t}, \bar{t}]$, are obtained, $H(t)$ is available and can then be used for the MCS process to estimate the reliability or the probability of failure.

Because $\|\alpha(t)\| = 1$ and \mathbf{U} is a vector of standard Gaussian variables, the mean of $H(t)$ is $\beta(t)$, the standard deviation of $H(t)$ is constantly 1, and the autocorrelation of $H(t)$ is [26, 45]

$$\rho(t_1, t_2) = \alpha^T(t_1)\alpha(t_2) \quad (8)$$

Note that although $\rho(t_1, t_2)$ is an important statistical characteristic of $H(t)$, it is not necessary for a sampling of $H(t)$. What we need are only the samples of \mathbf{U} , and the samples of $H(t)$ can be easily obtained via the following equation

$$H(t) = \beta(t) + \alpha^T(t)\mathbf{U} \quad (9)$$

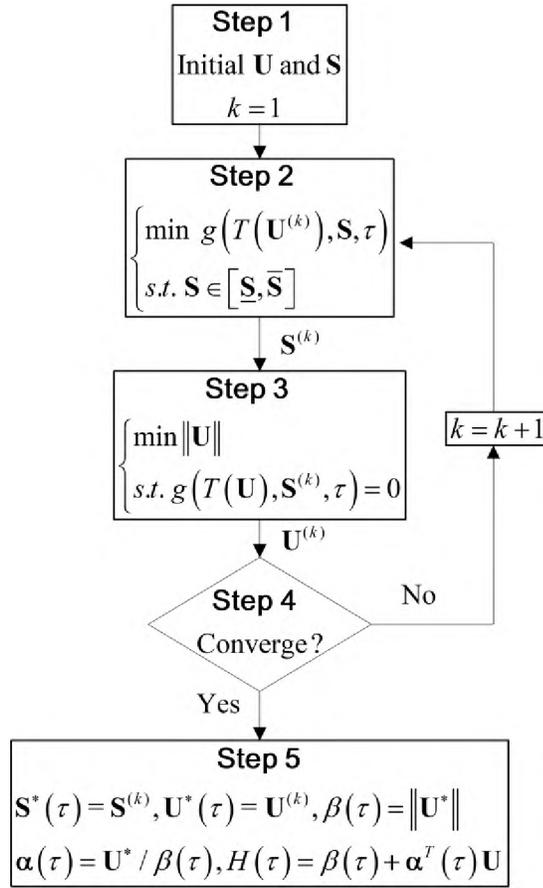


Figure 1. Flow chart of SSL

4.2. EXTREME VALUES ANALYSIS USING SORM

To improve the accuracy of Eq. (9), we also use SORM to update $\beta(t)$ if necessary. Since it is impossible to perform extreme value analyses at all time instants on $[\underline{t}, \bar{t}]$, we only do so at N instants of time denoted by $\mathbf{t} = (t_1, t_2, \dots, t_i, \dots, t_N)$, and hence what we need to update is $\beta(\mathbf{t}) = (\beta(t_1), \beta(t_2), \dots, \beta(t_i), \dots, \beta(t_N))$. However, SORM is more computationally expensive than FORM, especially when the number of dimensions of \mathbf{X} is large. Therefore, we propose to update only some key elements of $\beta(\mathbf{t})$ that influence

the target reliability R more than other elements. It is reasonable that those key elements have smaller values than others because a smaller instantaneous reliability index $\beta(t_i)$ contributes more to the failure event than a larger one.

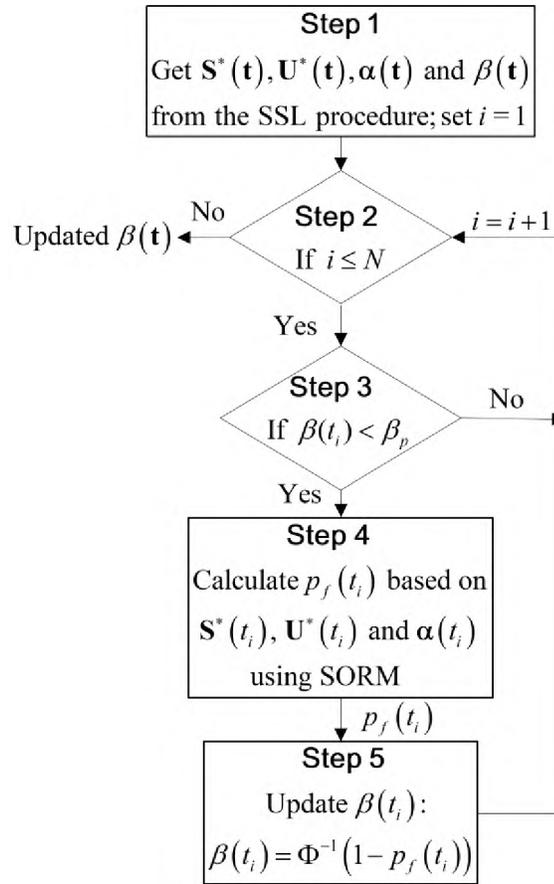


Figure 2. The procedure of updating $\beta(\mathbf{t})$ using SORM

Figure 2 shows the procedures to select the key elements of $\beta(\mathbf{t})$ and update them using SORM. In Step 1, $\mathbf{S}^*(\mathbf{t}) = (\mathbf{S}^*(t_1), \mathbf{S}^*(t_2), \dots, \mathbf{S}^*(t_i), \dots, \mathbf{S}^*(t_N))$, $\mathbf{U}^*(\mathbf{t}) = (\mathbf{U}^*(t_1), \mathbf{U}^*(t_2), \dots, \mathbf{U}^*(t_i), \dots, \mathbf{U}^*(t_N))$, and $\boldsymbol{\alpha}(\mathbf{t}) = (\boldsymbol{\alpha}(t_1), \boldsymbol{\alpha}(t_2), \dots, \boldsymbol{\alpha}(t_i), \dots, \boldsymbol{\alpha}(t_N))$.

In Step 3, β_p represents the p -th percentile of $\beta(\mathbf{t})$. For example, if $p = 30$, at 30% of the time instants SORM will be performed. Generally speaking, the larger is the value of p , the more accurate will R be, but with lower efficiency. In Step 4, since $\mathbf{S}^*(t_i)$, $\mathbf{U}^*(t_i)$, and $\boldsymbol{\alpha}(t_i)$ are already available from FORM in the SSL procedure, it is quite straightforward to calculate the corresponding instantaneous probability of failure $p_f(t_i)$ using SORM without searching for the MPP $\mathbf{U}^*(t_i)$.

5. PROCEDURE

In this section, the complete procedure of the proposed method is detailed. Overall, there are three main stages in the procedure. Stage 1 is the SSL procedure discussed in Subsection 4.1. Stage 2 updates $\beta(\mathbf{t})$ using SORM, as detailed in Subsection 4.2. Stage 3 calculates $\beta(t)$ and $\boldsymbol{\alpha}(t), t \in [\underline{t}, \bar{t}]$ with the employment of Kriging models. In the last stage, MCS is implemented to sample $H(t)$ and then estimate the probability of failure.

The flow chart is shown in Figure 3, and explanations are given in Table 1. In Figure 3, Steps 1, 2, and 6 are grouped into Stage 1; Steps 3, 8, and 9 are grouped into Stage 2; Stage 3 contains Steps 4 and 5; Stage 4 involves only Step 10. Since Stages 1 and 2 have been discussed in Section 4, and Stage 4 (i.e. the MCS procedure) is straightforward, herein we discuss mainly Stage 3, or the use of the Kriging model to approximate $\beta(t)$ and $\boldsymbol{\alpha}(t), t \in [\underline{t}, \bar{t}]$.

The Kriging model can provide not only predictions but also probabilistic error σ^2 (or the mean square error) of the predictions [44, 45]. Therefore, we can judge if the model is well trained with the error information. For a to-be-approximated function $F(\mathbf{v})$, the Kriging model is expressed as

$$\hat{F}(\mathbf{v}) = f(\mathbf{v}) + \varepsilon(\mathbf{v}) \quad (10)$$

where $f(\mathbf{v})$ includes polynomial terms with unknown coefficients, and $\varepsilon(\mathbf{v})$ is the error term assumed to be a Gaussian stochastic process with mean zero and variance σ^2 [44]. For the problem in this work, $F(\mathbf{v})$ may be $\alpha(t)$ or $\beta(t)$, and \mathbf{v} is t . This means that we build Kriging surrogate models for $\alpha(t)$ and $\beta(t)$ with respect to time. The Kriging models are denoted by $\hat{\beta}(t)$ and $\hat{\alpha}(t)$. We do not provide details about how to create the models, and interested readers can refer to reference [44].

Some initial samples of $\beta(t)$ and $\alpha(t)$ are generated after the SSL procedure has been performed at instants $\mathbf{t} = (\underline{t}, t_2, \dots, t_{N-1}, \bar{t})$. Then the samples of $\beta(t)$ and $\alpha(t)$ are used to train Kriging models, which are then used to approximate or predict $\alpha^T(t)$ and $\beta(t)$ at $\mathbf{t}_p = (\underline{t}, t_2, \dots, t_{N-1}, \bar{t})$. Since the dimension of $(\alpha^T(t), \beta(t))^T$ is $m+1$, with the Kriging prediction, a prediction matrix $\boldsymbol{\mu}$ and prediction error matrix $\boldsymbol{\sigma}^2$, whose dimensions are both $N_t \times (m+1)$, can be obtained. Then the prediction error coefficients $\boldsymbol{\gamma}$ are calculated by

$$\boldsymbol{\gamma} = \boldsymbol{\sigma} ./ \boldsymbol{\mu} \quad (11)$$

where “./” denotes an elementwise vector division.

Table 1. Explanations for the complete procedure

Steps	Explanations
1	$t_1 = \underline{t}, t_N = \bar{t}$.
2	The detailed procedure of SSL for a given instant of time is shown in Figure 1. Note that after $\mathbf{U}^*(t_i)$ has been obtained, it will be treated as the initial point when searching for $\mathbf{U}^*(t_{i+1})$. The reason is that usually $\mathbf{U}^*(t_{i+1})$ is to some extent close to $\mathbf{U}^*(t_i)$ and that taking $\mathbf{U}^*(t_i)$ as the initial point of $\mathbf{U}^*(t_{i+1})$ may reduce the cost of searching for $\mathbf{U}^*(t_{i+1})$. Similarly, $\mathbf{S}^*(t_i)$ is also treated as the initial point of $\mathbf{S}^*(t_{i+1})$.
3	Details of this step are given in Figure 2.
4	Kriging models $\hat{\beta}(t)$ and $\hat{\alpha}(t)$ are built. Additionally, the maximum prediction error coefficient γ_{\max} , and the instant t_{new} of time corresponding to γ_{\max} are also obtained.
5	If γ_{\max} is larger than the allowable value $\gamma_{allowable}$, the Kriging model is not well trained, and then a new training point at t_{new} is added. There is no rigorous method to determine the value of $\gamma_{allowable}$, but experiments show that 10^{-4} is a good one.
6	Details are given in Figure 1.
7	The set of training points is updated.
10	N_s samples of \mathbf{U} are generated first, and then N_s samples of $H(t)$ are obtained with $\hat{H}(t) = \hat{\beta}(t) + \hat{\alpha}^T(t)\mathbf{U}$. During the process, $[\underline{t}, \bar{t}]$ is evenly discretized into N_t points $(\underline{t}, t_2, \dots, t_{N_t-1}, \bar{t})$.

To make sure the Kriging models are well trained, the maximum γ_{\max} of $\boldsymbol{\gamma}$ should be smaller than the allowable value $\gamma_{allowable}$. If $\gamma_{\max} > \gamma_{allowable}$, a new instant t_{new} of time is selected through

$$t_{new} = \arg \max \boldsymbol{\gamma}(\mathbf{t}_p = \underline{t}, t_2, \dots, t_{N_t-1}, \bar{t}) \quad (12)$$

and $\beta(t_{new})$ and $\alpha(t_{new})$ are added to the training point set to refine the Kriging models. Usually, a smaller $\gamma_{allowable}$ leads to higher accuracy of p_f , but more training points are needed, thus resulting in lower efficiency.

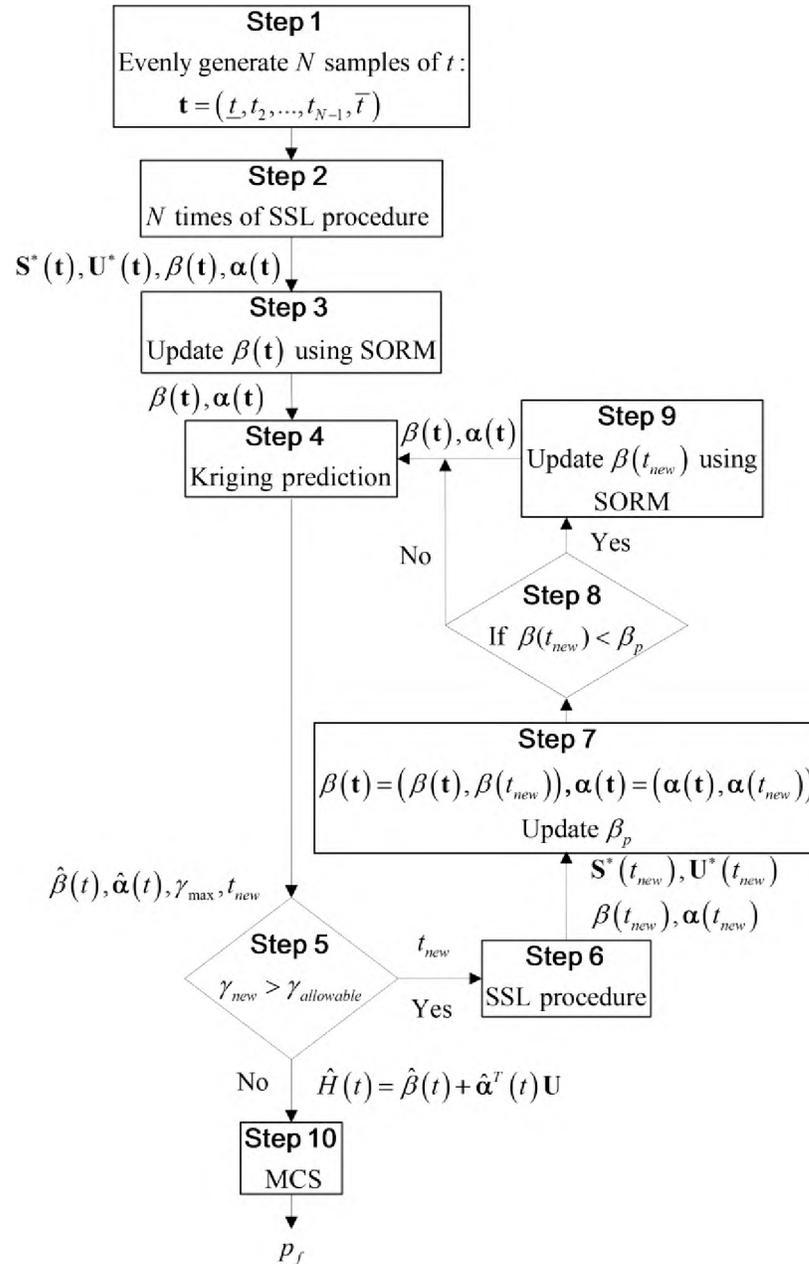


Figure 3. Flow chart of the complete procedure

6. EXAMPLES

In this section, two examples are used to demonstrate the proposed method. MCS is employed to provide accurate solutions for accuracy comparison.

6.1. A MATH EXAMPLE

In this mathematical example, the limit-state function is defined by

$$g(\mathbf{X}, \mathbf{S}, t) = 8 + 10x_1 + 12x_2 + x_1x_2 + 0.1s_1s_2x_1^2 - 0.2x_2^2 \cos(t + \pi/2) + \sin(t) \quad (13)$$

where $\mathbf{X} = (x_1, x_2)^T$ is the vector of two independent random variables $x_i \sim N(0, 0.2^2)$, $i = 1, 2$, $\mathbf{S} = (s_1, s_2)^T$, where $s_i \in [1.5, 2.5]$, $i = 1, 2$, is the spatial variable vector, and $t \in [0, 2\pi]$ rad is the temporal variable.

The probability of failure is computed over different time intervals with both MCS and the proposed method. In this example, the 50th percentile (i.e. $p = 50$) of $\beta(\mathbf{t})$ is used to determine which $\beta(t_i)$ should be updated using SORM, the allowable maximum prediction error coefficient is $\gamma_{allowable} = 10^{-4}$, the initial value of N is 5 (for Kriging models), the number of simulations for $H(t)$ is $N_s = 10^6$, and the number of discretized instants of time is $N_t = 126$, which gives a step size of the time 0.05. The number of simulations of MCS N_{MCS} is set to 10^6 , which is the same as N_s .

Theoretically, in MCS, for every given realization \mathbf{s} of \mathbf{S} , we need to generate N_{MCS} samples of stochastic process $g(\mathbf{X}, \mathbf{s}, t)$, leading to a heavy computational burden.

In this example, however, for every given realization \mathbf{x} of \mathbf{X} , $\min_{\mathbf{s} \in [1.5, 2.5]^2, t \in [0, 2\pi]} g(\mathbf{x}, \mathbf{S}, t)$ can always be obtained analytically, and so we use $g(\mathbf{X}) = \min_{\mathbf{s} \in [1.5, 2.5]^2, t \in [0, 2\pi]} g(\mathbf{X}, \mathbf{S}, t)$ to replace the limit-state function shown in Eq. (2) and then perform MCS to get accurate results. Results from the proposed method and MCS are listed in Table 2 and plotted in Figure 4.

Table 2. Probability of failure over different time intervals

$[0, t]$	$p_f(\text{proposed})$ (10^{-3})	$p_f(\text{MCS})$ (10^{-3})	Error (%)
$[0, 3.0]$	4.636	4.663	0.58
$[0, 3.5]$	6.602	6.617	0.23
$[0, 4.0]$	9.579	9.566	0.14
$[0, 4.5]$	11.666	11.581	0.73
$[0, 2\pi]$	11.902	11.808	0.80

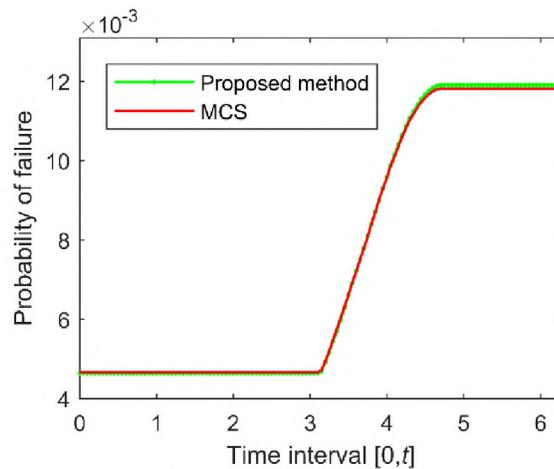


Figure 4. Probability of failure over different time intervals

As Table 2 and Figure 4 show, the proposed method has good accuracy. The error is mainly caused by the nonlinearity of the limit-state function. Besides, the number of

limit-state function calls by the proposed method is 217, far less than 33×10^6 , which is the total number of limit-state function calls by MCS, showing that the proposed method is quite efficient.

6.2. A SLIDER MECHANISM

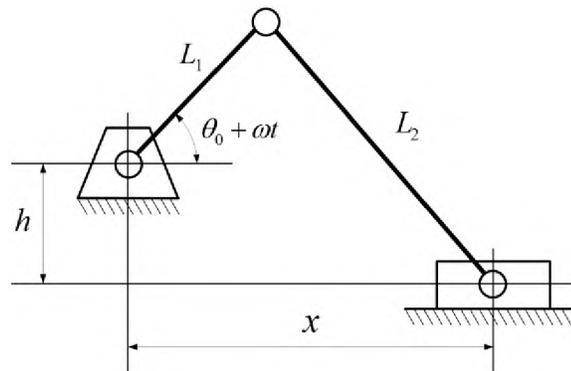


Figure 5. A slider mechanism

Shown in Figure 5 is a slider mechanism. It is used for difference applications (locations). The locations or spatial variables are the offset h and the initial angle θ_0 with the following ranges: $h \in [14.9, 15.1]$ m and $\theta_0 \in [0^\circ, 5^\circ]$; the spatial variable vector is then $\mathbf{S} = (h, \theta_0)^T$. The random variable vector is $\mathbf{X} = (L_1, L_2)^T$, which includes two independent random link lengths $L_1 \sim N(15, 0.15^2)$ m and $L_2 \sim N(35, 0.35^2)$ m. The time span is $t \in [0, 0.2\pi]$ s. The limit-state function is defined by

$$g = 1.1 - (x_{actual} - x_{required}) \quad (14)$$

in which the actual position x_{actual} and the required position $x_{required}$ of the slider are

$$x_{actual} = L_1 \cos(\theta_0 + \omega t) + \sqrt{L_2^2 - (h + L_1 \sin(\theta_0 + \omega t))^2} \quad (15)$$

$$x_{required} = 15 \cos(\omega t) + \sqrt{35^2 - (15 + 15 \sin(\omega t))^2} \quad (16)$$

respectively, where $\omega = 1 \text{ rad/s}$ is the angular velocity.

The probability of failure is computed over different time intervals with both MCS and the proposed method. In this example, $p = 50$, $\gamma_{allowable} = 10^{-4}$, $N_s = N_{MCS} = 10^6$, $N_t = 40$ (i.e., the time step of the discretization of $H(t)$ is 0.005π), and the initial value of N is 7.

Table 3. Probability of failure over different time intervals

$[0, t]$ (0.01π s)	p_f (proposed) (10^{-3})	p_f (MCS) (10^{-3})	Error (%)
[0,5]	6.765	6.729	0.53
[0,10]	8.750	8.729	0.24
[0,15]	11.930	11.811	1.01
[0,20]	16.975	17.015	0.24

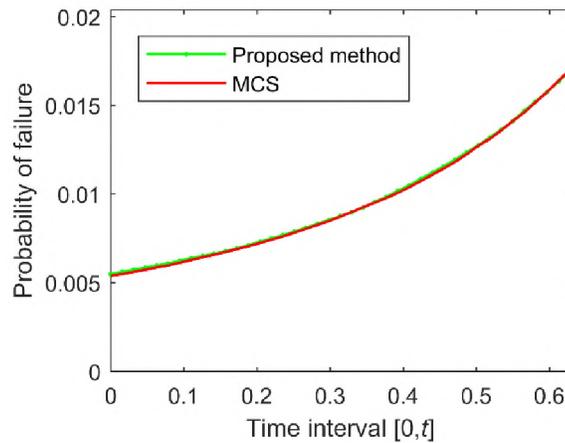


Figure 6. Probability of failure over different time intervals

Results from the proposed method and MCS are listed in Table 3 and are plotted in Figure 6. The proposed method obtains accurate results. As for the efficiency, the proposed method evaluates the limit-state function 214 times while MCS approximately 40.6×10^6 . This indicates that the proposed method is much more efficient.

7. CONCLUSIONS

In this work, a combination of the first-order and the second-order methods (FORM and SORM) is proposed to perform uncertainty analysis for a time- and space-dependent response with random input variables. With the employment of FORM, SORM, and the sequential single-loop method, we firstly transform the time- and space-dependent response into an equivalent Gaussian stochastic process, thus converting the time- and space-dependent reliability problem into an equivalent time-dependent reliability problem. Then the equivalent Gaussian process is simulated to estimate the time- and space-dependent probability of failure. To mitigate the computation burden, Kriging models are created to approximate the characteristics of the equivalent Gaussian stochastic process.

Transforming the time- and space-dependent response into an equivalent Gaussian stochastic process can avoid the global optimization process which aims at obtaining the minimum value of the limit-state function with respect to the temporal variable.

Numerical examples show that the proposed method has both good accuracy and efficiency. If the limit-state function, however, is a nonconvex function with respect to spatial variables, the true extreme value of the response may not be easily found, and in this case, the proposed method may result in large errors, or low efficiency, or both. The

extreme value of a limit-state function may not be differentiable, and in this case, the MPP search for both FORM and SORM may not converge if a gradient-based MPP search algorithm is used.

Future research may focus on two directions. The first direction is to develop efficient global optimization methods for the minimum response with respect to both special and temporal variables, thus transforming the time- and space-dependent problem into a traditional time- and space-independent problem. And the second one is to investigate optimization-free methods to efficiently deal with general problems where the limit-state function is highly nonlinear with respect to input random variables and nonconvex with respect to both spatial and temporal variables.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under grant CMMI 1727329. The support from the Intelligent Systems Center at the Missouri University of Science and Technology is also acknowledged.

REFERENCES

- [1] Du, X., 2015, "Development of Engineering Uncertainty Repository," (57106), p. V003T004A014.
- [2] Du, X., and Chen, W., 2001, "A Most Probable Point-Based Method for Efficient Uncertainty Analysis," *Journal of Design and Manufacturing Automation*, 4(1), pp. 47-66.
- [3] Hu, Z., and Du, X., 2012, "Reliability analysis for hydrokinetic turbine blades," *Renewable Energy*, 48, pp. 251-262.

- [4] Hu, Z., and Du, X., 2014, "Lifetime cost optimization with time-dependent reliability," *Engineering Optimization*, 46(10), pp. 1389-1410.
- [5] Hu, Z., Li, H., Du, X., and Chandrashekhara, K., 2013, "Simulation-based time-dependent reliability analysis for composite hydrokinetic turbine blades," *Structural and Multidisciplinary Optimization*, 47(5), pp. 765-781.
- [6] Choi, S. K., Canfield, R. A., and Grandhi, R. V., 2007, *Reliability-based Structural Design*, Springer London.
- [7] Du, X., Sudjianto, A., and Huang, B., 2005, "Reliability-based design with the mixture of random and interval variables," *Journal of mechanical design*, 127(6), pp. 1068-1076.
- [8] Huang, B., and Du, X., 2008, "Probabilistic uncertainty analysis by mean-value first order saddlepoint approximation," *Reliability Engineering & System Safety*, 93(2), pp. 325-336.
- [9] Zhang, J., and Du, X., 2010, "A second-order reliability method with first-order efficiency," *Journal of Mechanical Design*, 132(10), p. 101006.
- [10] Madsen, H. O., Krenk, S., and Lind, N. C., 2006, *Methods of structural safety*, Courier Corporation.
- [11] Banerjee, B., and Smith, B. G., "Reliability analysis for inserts in sandwich composites," *Proc. Advanced Materials Research, Trans Tech Publ*, pp. 234-238.
- [12] Kim, D.-W., Jung, S.-S., Sung, Y.-H., and Kim, D.-H., 2011, "Optimization of SMES windings utilizing the first-order reliability method," *The Transactions of The Korean Institute of Electrical Engineers*, 60(7), pp. 1354-1359.
- [13] Huang, B., and Du, X., 2006, "Uncertainty analysis by dimension reduction integration and saddlepoint approximations," *Journal of Mechanical Design*, 128(1), pp. 26-33.
- [14] Jin , R., Du , X., and Chen , W., 2003, "The use of metamodeling techniques for optimization under uncertainty," *Structural and Multidisciplinary Optimization*, 25(2), pp. 99-116.
- [15] Isukapalli, S., Roy, A., and Georgopoulos, P., 1998, "Stochastic response surface methods (SRSMs) for uncertainty propagation: application to environmental and biological systems," *Risk analysis*, 18(3), pp. 351-363.
- [16] Zhu, Z., and Du, X., 2016, "Reliability analysis with Monte Carlo simulation and dependent Kriging predictions," *Journal of Mechanical Design*, 138(12), p. 121403.

- [17] Xiong, F., Xiong, Y., Greene, S., Chen, W., and Yang, S., 2009, "A New Sparse Grid Based Method for Uncertainty Propagation," (49026), pp. 1205-1215.
- [18] Ditlevsen, O., and Madsen, H. O., 1996, *Structural reliability methods*, Wiley New York.
- [19] Sobol', I., 1990, "Quasi-monte carlo methods," *Progress in Nuclear Energy*, 24(1-3), pp. 55-61.
- [20] Dey, A., and Mahadevan, S., 1998, "Ductile structural system reliability analysis using adaptive importance sampling," *Structural safety*, 20(2), pp. 137-154.
- [21] Au, S.-K., and Beck, J. L., 2001, "Estimation of small failure probabilities in high dimensions by subset simulation," *Probabilistic engineering mechanics*, 16(4), pp. 263-277.
- [22] Der Kiureghian, A., and Ke, J.-B., 1988, "The stochastic finite element method in structural reliability," *Probabilistic Engineering Mechanics*, 3(2), pp. 83-91.
- [23] Ghanem, R. G., and Spanos, P. D., 1991, "Stochastic Finite Element Method: Response Statistics," *Stochastic finite elements: a spectral approach*, Springer, pp. 101-119.
- [24] Andrieu-Renaud, C., Sudret, B., and Lemaire, M., 2004, "The PHI2 method: a way to compute time-variant reliability," *Reliability Engineering & System Safety*, 84(1), pp. 75-86.
- [25] Hu, Z., and Du, X., 2013, "Time-dependent reliability analysis with joint upcrossing rates," *Structural and Multidisciplinary Optimization*, 48(5), pp. 893-907.
- [26] Jiang, C., Wei, X. P., Huang, Z. L., and Liu, J., 2017, "An Outcrossing Rate Model and Its Efficient Calculation for Time-Dependent System Reliability Analysis," *Journal of Mechanical Design*, 139(4), pp. 041402-041402-041410.
- [27] Hu, Z., and Du, X., 2015, "Mixed efficient global optimization for time-dependent reliability analysis," *Journal of Mechanical Design*, 137(5), p. 051401.
- [28] Hu, Z., and Mahadevan, S., 2016, "A single-loop kriging surrogate modeling for time-dependent reliability analysis," *Journal of Mechanical Design*, 138(6), p. 061406.
- [29] Wang, Z., and Chen, W., 2016, "Time-variant reliability assessment through equivalent stochastic process transformation," *Reliability Engineering & System Safety*, 152, pp. 166-175.

- [30] Wang, Z., and Wang, P., 2012, "A Nested Extreme Response Surface Approach for Time-Dependent Reliability-Based Design Optimization," *Journal of Mechanical Design*, 134(12), pp. 121007-121007-121014.
- [31] Wang, Z., Mourelatos, Z. P., Li, J., Baseski, I., and Singh, A., 2014, "Time-dependent reliability of dynamic systems using subset simulation with splitting over a series of correlated time intervals," *Journal of Mechanical Design*, 136(6), p. 061008.
- [32] Singh, A., Mourelatos, Z., and Nikolaidis, E., 2011, "Time-dependent reliability of random dynamic systems using time-series modeling and importance sampling," *SAE International Journal of Materials and Manufacturing*, 4(2011-01-0728), pp. 929-946.
- [33] Chen, J.-B., and Li, J., 2005, "Dynamic response and reliability analysis of non-linear stochastic structures," *Probabilistic Engineering Mechanics*, 20(1), pp. 33-44.
- [34] Du, X., 2014, "Time-dependent mechanism reliability analysis with envelope functions and first-order approximation," *Journal of Mechanical Design*, 136(8), p. 081010.
- [35] Yu, S., and Wang, Z., 2018, "A novel time-variant reliability analysis method based on failure processes decomposition for dynamic uncertain structures," *Journal of Mechanical Design*, 140(5), p. 051401.
- [36] Yu, S., Wang, Z., and Meng, D., 2018, "Time-variant reliability assessment for multiple failure modes and temporal parameters," *Structural and Multidisciplinary Optimization*, pp. 1-13.
- [37] Hu, Z., and Mahadevan, S., "Reliability Analysis of Multidisciplinary System with Spatio-temporal Response using Adaptive Surrogate Modeling," *Proc. 2018 AIAA Non-Deterministic Approaches Conference*, p. 1934.
- [38] Hu, Z., and Mahadevan, S., 2017, "A surrogate modeling approach for reliability analysis of a multidisciplinary system with spatio-temporal output," *Structural and Multidisciplinary Optimization*, 56(3), pp. 553-569.
- [39] Shi, Y., Lu, Z., Cheng, K., and Zhou, Y., 2017, "Temporal and spatial multi-parameter dynamic reliability and global reliability sensitivity analysis based on the extreme value moments," *Structural and Multidisciplinary Optimization*, 56(1), pp. 117-129.
- [40] Shi, Y., Lu, Z., Zhang, K., and Wei, Y., 2017, "Reliability analysis for structures with multiple temporal and spatial parameters based on the effective first-crossing point," *Journal of Mechanical Design*, 139(12), pp. 121403-121403.

- [41] Hu, Z., and Du, X., 2015, "A random field approach to reliability analysis with random and interval variables," *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering*, 1(4), p. 041005.
- [42] Hu, Z., and Du, X., "A Random Field Method for Time-Dependent Reliability Analysis With Random and Interval Variables," *Proc. ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers, pp. V02BT03A044-V002BT003A044.
- [43] Guo, J., and Du, X., 2010, "Reliability analysis for multidisciplinary systems with random and interval variables," *AIAA journal*, 48(1), pp. 82-91.
- [44] Lophaven, S. N., Nielsen, H. B., and Søndergaard, J., 2002, *DACE: a Matlab kriging toolbox*, Citeseer.
- [45] Hu, Z., and Du, X., 2015, "First order reliability method for time-variant problems using series expansions," *Structural and Multidisciplinary Optimization*, 51(1), pp. 1-21.
- [46] Jiang, C., Lu, G., Han, X., and Liu, L., 2012, "A new reliability analysis method for uncertain structures with random and interval variables," *International Journal of Mechanics and Materials in Design*, 8(2), pp. 169-182.

IV. ROBUSTNESS METRIC FOR ROBUST DESIGN OPTIMIZATION UNDER TIME- AND SPACE-DEPENDENT UNCERTAINTY THROUGH MODELING

ABSTRACT

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

1. INTRODUCTION

Robust design optimization (RDO) [1] is an optimization design methodology for improving the quality of a product by minimizing the effect of the causes of variation

without eliminating the causes [2]. It allows for the use of low-grade materials and reduces labor and material costs while improving reliability and reducing operating costs [2]. RDO has been used to improve product quality in industrial applications [3, 4]. Over the last three decades, it has gained much attention from many research fields, such as operations research [5-7], aerospace [8, 9], structural mechanics [10, 11], vibration control [12, 13], automobile [14-16], and fatigue analysis [17, 18]. Methods to solve RDO can be roughly grouped into three categories: probabilistic methods [19-21], deterministic methods [22-26], and metamodel-based methods [27-32]. Probabilistic methods perform robust optimization using the probability distributions of random variables. Deterministic methods incorporate a non-statistical index, such as the gradient of a response, into the optimization problem to obtain a robust optimum [32]. Metamodel-based methods employ computationally cheap surrogate models to improve the efficiency of RDO.

Robustness analysis, which evaluates and predicts the robustness of a design, is repeated many times during RDO. Many metrics that measure the robustness exist in literature. The most common metric is Taguchi's quality loss function (QLF) [2]. This metric measures not only the distance between the average quality characteristics (QCs) and their targets but also the variation in the QCs [33]. There are also other robustness metrics, such as the signal-to-noise ratio [2], the percentile difference [34], and the worst-case QCs [35].

Most of the above robustness metrics are defined for static QCs that do not change over time and space. Some of the metrics could be used for dynamics problems, but they are only applicable for situations where the targets of QCs vary with signals [36, 37], instead of with time. To deal with problems involving time-dependent QCs, Goethals et al.

[38] proposed to use the weighted sum of mean values of a QLF at discretized time instances to measure the robustness. The weighted-sum method, however, does not take into consideration of the autocorrelation of the time-dependent QLF, which is modeled as a stochastic process. To overcome this drawback, Du [33] proposed to use the maximum value of the time-dependent QLF to measure the time-dependent robustness.

In addition to the above static and time-dependent problems, more general is the time- and space-dependent (TSD) problem [39]. In many engineering applications, QCs vary with both time and space. There are at least two reasons for the TSD QCs. (1) A QC is a function of TSD variables, such as the wind load and road conditions. (2) The QC itself is a function of temporal and spatial variables. A typical example is a wind turbine. Since the wind speed varies with time and location, it is usually modeled as a TSD random field, subjected to which, the QC of the turbine is hence TSD.

There is a need to define a new robustness metric for the optimization involving TSD problems. The object of this work is to derive a robustness metric for TSD problems and develop a numerical method to evaluate it. We use the expectation of the maximum value of a TSD QLF to measure the robustness. For the former, we employ the same strategy in [33], and for the latter, we use a metamodeling method to manage the computational efficiency because of the involvement of the expensive multidimensional global optimization [40-43] with respect to temporal and spatial parameters. An efficient method based on the Gaussian process model [44-47] is then proposed. The contributions of this work are twofold. First, a TSD robustness metric is defined. It can take into consideration of all information of the TSD QLF, including its autocorrelation. Therefore,

it is mathematically a rigorous metric for TSD problems. Second, a Gaussian process-based method is developed to effectively compute the TSD robustness metric.

The proposed TSD robustness metric is an extension of the time-dependent robustness metric proposed in [33]. The similarity is that both the proposed TSD robustness metric and the time-dependent robustness metric use the maximum value of the QLF to measure the robustness. However, this study deals with a more general and complicated problem because the time-dependent problem is only a special case of the TSD problem. From the perspective of mathematical models, the new robustness metric needs the multidimensional global optimization with respect to both temporal and spatial parameters, while the time-dependent one involves unidimensional global optimizations with respect to only a temporal parameter. Besides, the new QLF may include random fields in its input.

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

2. REVIEW OF STATIC AND TIME-DEPENDENT ROBUSTNESS METRICS

Nominal-the-best, smaller-the-better, and larger-the-better are three types of QCs [33]. In this work, we only focus on the nominal-the-best type. The discussions, however, can be extended to the other two types.

2.1. STATIC ROBUSTNESS METRIC

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

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

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

$$L = A(Y - m)^2 \quad (2)$$

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

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

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

2.2. TIME-DEPENDENT ROBUSTNESS METRIC

A time-dependent QC is given by

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

Note that the input of $g(\cdot)$ may also include random processes, which can be transformed into functions with respect to random variables and t [48]. Thus Eq. (4) does not lose generality. At instant t , the QLF is given as

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

$L(t)$ can measure only the quality loss at a specific time instant t and is thus called point quality loss function (P-QLF). To measure the quality loss of a product over a time interval $[\underline{t}, \bar{t}]$, Du [33] proposed to use the extreme value or the worst-case value of $L(t)$ over $[\underline{t}, \bar{t}]$.

The worst-case quality loss is called interval quality loss function (I-QLF) and is given by

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

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

$$E_L(\underline{t}, \bar{t}) = E[L(\underline{t}, \bar{t})] \quad (7)$$

where $E(\cdot)$ stands for expectation. Minimizing $E_L(\underline{t}, \bar{t})$ reduces both the deviation of the QC from its target and the variation in the QC over the time interval $[\underline{t}, \bar{t}]$. When \mathbf{X} is fixed to a specific realization \mathbf{x} , Eq. (6) shows a unidimensional global optimization problem. Multiple samples of $L(\underline{t}, \bar{t})$ are necessary to calculate $E_L(\underline{t}, \bar{t})$ using Eq. (7), and hence multiple unidimensional global optimizations are required to obtain $E_L(\underline{t}, \bar{t})$.

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

In TSD problems, in addition to random variables and random processes, static random fields and time-dependent random fields are also involved. For convenience, we do not distinguish random processes, static random fields, or time-dependent random fields. In this paper, we generally call them random fields. Let $\mathbf{Z} = (S_1, S_2, S_3, t)$ be the vector comprising the three spatial parameters (x -, y -, and z -coordinates) and the time. Note that for problems in one-dimensional and two-dimensional space, $\mathbf{Z} = (S_1, t)$ and $\mathbf{Z} = (S_1, S_2, t)$, respectively. Also note that random fields can be transformed into functions with respect to random variables and \mathbf{Z} [48]. Without loss of generality, a TSD QC is then given by

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

With the TSD QC, the QLF is given by

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

$L(\mathbf{X}, \mathbf{Z})$ measures the quality loss at any specific point $\mathbf{z} \in \Omega$, where Ω is the domain of \mathbf{Z} , so it is also a P-QLF.

Before defining the TSD robustness metric, we propose some criteria of robustness metrics for the TSD problems, inspired by the criteria of the robustness metrics for time-dependent problems given in [33]. The criteria are as follows:

(a) The metric must represent the maximum quality loss over Ω . This reflects the fact that the quality loss is not reversible. If a quality loss, including the maximum quality loss, has occurred, there is no way to turn back.

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

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

(d) Minimizing the metric will lead to optimizing the mean QCs and minimizing the variations of the QCs over Ω . This criterion comes from the purpose of robust optimization [49].

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

$$E_L(\Omega) = E[L_{\max}(\mathbf{X}, \Omega)] \quad (10)$$

where

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

is the maximum value of $L(\mathbf{X}, \mathbf{Z})$ and is called the domain quality loss function (D-QLF).

The definition of $L_{\max}(\mathbf{X}, \Omega)$ ensures that $E_L(\Omega)$ meet Criterion (a) naturally. Let $\hat{\Omega} \subset \Omega$, then it is obvious that

$$L_{\max}(\mathbf{x}, \hat{\Omega}) \leq L_{\max}(\mathbf{x}, \Omega) \quad (12)$$

and hence $E_L(\hat{\Omega}) \leq E_L(\Omega)$. Therefore, $E_L(\Omega)$ meets Criterion (b). Since $L_{\max}(\mathbf{X}, \Omega)$ is the maximum value distribution [50, 51] of $L(\mathbf{X}, \mathbf{Z})$, the autocorrelation of $L(\mathbf{X}, \mathbf{Z})$ is necessary for computing $L_{\max}(\mathbf{X}, \Omega)$. Different autocorrelation functions of $L(\mathbf{X}, \mathbf{Z})$ will lead to different distributions of $L_{\max}(\mathbf{X}, \Omega)$, and hence $E_L(\Omega)$ can capture the autocorrelation of $L(\mathbf{X}, \mathbf{Z})$, indicating that $E_L(\Omega)$ meets Criterion (c). Since $L(\mathbf{X}, \mathbf{Z})$, and hence $L_{\max}(\mathbf{X}, \Omega)$ and $E_L(\Omega)$, are nonnegative, minimizing $E_L(\Omega)$ requires that the QC $g(\mathbf{X}, \mathbf{Z})$ gets close to its target $m(\mathbf{Z})$ as much as possible. Therefore, $E_L(\Omega)$ also meets Criterion (d).

4. A META-MODELING APPROACH TO ROBUSTNESS ANALYSIS

The robustness metric defined in Eq. (10) involves the extreme value of a general random field. It is not an easy task to evaluate the robustness metric for a given design. In this section, we discuss our proposal numerical method for the robustness analysis.

4.1. OVERVIEW OF THE PROPOSED ROBUSTNESS ANALYSIS

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

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

Then the Monte Carlo simulation (MCS) [52] is used to compute $E_L(\Omega)$ by

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

where n_{MCS} is the sample size, and $\mathbf{x}^{(i)}$ is the i -th sample of \mathbf{X} . Since $\hat{L}(\mathbf{X}, \mathbf{Z})$ is computationally much cheaper than $L(\mathbf{X}, \mathbf{Z})$, the proposed method can significantly improve efficiency. Generally, a larger number of training points of $L(\mathbf{X}, \mathbf{Z})$ is preferred to train $\hat{L}(\mathbf{X}, \mathbf{Z})$ for higher accuracy, but the efficiency will decrease because $L(\mathbf{X}, \mathbf{Z})$ in engineering applications is often a black-box function whose evaluation needs expensive numerical procedures or simulations [53].

To balance accuracy and efficiency, we do not require $\hat{L}(\mathbf{X}, \mathbf{Z})$ to be accurate globally. Instead, we only need it to be locally accurate at samples of \mathbf{X} in Eq. (14). To this end, we employ the efficient global optimization (EGO) [54, 55] to adaptively add training points to update $\hat{L}(\mathbf{X}, \mathbf{Z})$.

To have a quick overview of the proposed method, we give a simplified flowchart in Figure 1. There are in total eight steps in the proposed method. Details of Step 2 will be given in Subsection 4.2. The EGO, which comprises Steps 3 through 5, will be detailed in Subsection 4.3. We propose two stopping criteria in Steps 4 and 7, respectively. Detailed information is given in Subsection 4.4. The implementation of the algorithm and the

detailed flowchart will be given in Subsection 4.5. In Subsection 4.6, we discuss how to deal with a more general problem that involves random fields.

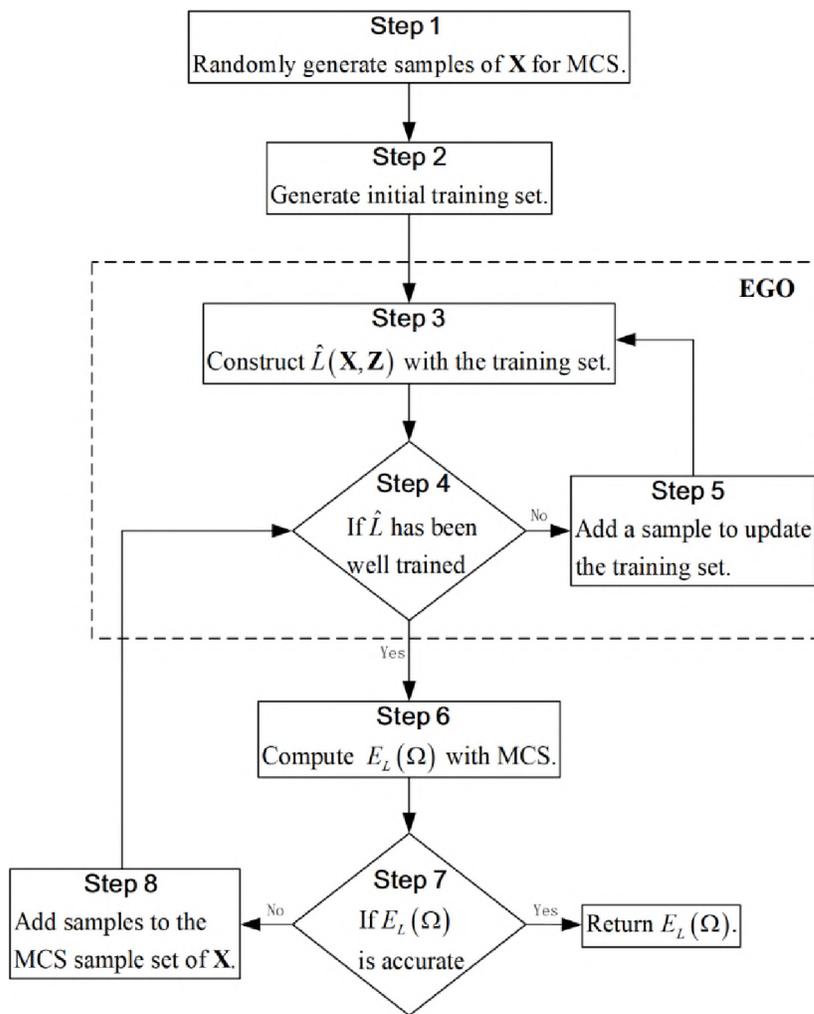


Figure 1. Simplified flowchart

4.2. INITIAL TRAINING SET

The principle of generating the initial training set for building a Gaussian process model is to spread the initial training points evenly. Commonly used sampling methods include random sampling, Latin hypercube sampling, and Hammersley sampling [56]. In

this study, we employ the Hammersley sampling method because it has better uniformity properties over a multidimensional space [57].

Since the dimension of the entire input vector (\mathbf{X}, \mathbf{Z}) is $N + N_Z$, where N_Z is the dimension of \mathbf{Z} , the Hammersley sampling method generates initial training points in a hypercube $[0,1]^{N+N_Z}$. To get initial training points of \mathbf{X} , we can simply use the inverse probability method to transform the training points from the hypercube space to the X-space. As for the initial training points of \mathbf{Z} , we treat all components of \mathbf{Z} as if they were independent uniform random variables and then also use the inverse probability method to transform the training points from the hypercube space to the Z-space.

Samples of a row random vector are assembled into a matrix. For example, the initial training points \mathbf{x}^{in} of $\mathbf{X} = (X_1, X_2 \dots, X_N)$ are

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

where n_{in} is the total number of initial training points. With \mathbf{x}^{in} and the initial training points \mathbf{z}^{in} of \mathbf{Z} , we then obtain initial training points \mathbf{l}^{in} of $L(\mathbf{X}, \mathbf{Z})$ by calling Eq. (9). Finally, we get the initial training set $(\mathbf{x}^{\text{trn}}, \mathbf{z}^{\text{trn}}, \mathbf{l}^{\text{trn}}) = (\mathbf{x}^{\text{in}}, \mathbf{z}^{\text{in}}, \mathbf{l}^{\text{in}})$, where the superscript *trn* represents training points.

4.3. EMPLOYMENT OF EGO

EGO is based on the Gaussian process model. With the training set $(\mathbf{x}^{\text{trn}}, \mathbf{z}^{\text{trn}}, \mathbf{l}^{\text{trn}})$ we can build $\hat{L}(\mathbf{X}, \mathbf{Z})$. Because only a limit number of training points are used, $\hat{L}(\mathbf{X}, \mathbf{Z})$ has model uncertainty (or epistemic uncertainty), which is measured by $\sigma(\mathbf{X}, \mathbf{Z})$.

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

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

Since $\hat{L}_{\max}(\mathbf{X}, \Omega)$ may not be the exact global maximum, we need to add training points of $(\mathbf{X}, \mathbf{Z}, L)$ to update $\hat{L}(\mathbf{X}, \mathbf{Z})$ so that the $\hat{L}_{\max}(\mathbf{X}, \Omega)$ will be more accurate. To determine how to add a new training point, we use the well-known expected improvement (EI) learning function [55] given by

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

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

$$(\mathbf{x}^{(\text{next})}, \mathbf{z}^{(\text{next})}) = \arg \max_{\mathbf{x} \in \mathbf{x}^{\text{MCS}}, \mathbf{z} \in \mathbf{z}^\Omega} \text{EI}(\mathbf{x}, \mathbf{z}) \quad (18)$$

where \mathbf{x}^{MCS} represents the MCS population of \mathbf{X} . Eq. (18) indicates a double-layer optimization. The reason is that whenever we want to optimize \mathbf{Z} in Eq. (18), we must fix

\mathbf{X} to a specific realization \mathbf{x} so that Eq. (16) can be used to calculate \hat{L}_{\max} . With Eq. (9), we can obtain the next training point $\mathbf{l}^{(\text{next})}$ of $L(\mathbf{X}, \mathbf{Z})$. Then the training set $(\mathbf{x}^{\text{trn}}, \mathbf{z}^{\text{trn}}, \mathbf{l}^{\text{trn}})$ is updated through

$$\begin{cases} \mathbf{x}^{\text{trn}} = \begin{bmatrix} \mathbf{x}^{\text{trn}} \\ \mathbf{x}^{(\text{next})} \end{bmatrix} \\ \mathbf{z}^{\text{trn}} = \begin{bmatrix} \mathbf{z}^{\text{trn}} \\ \mathbf{z}^{(\text{next})} \end{bmatrix} \\ \mathbf{l}^{\text{trn}} = \begin{bmatrix} \mathbf{l}^{\text{trn}} \\ \mathbf{l}^{(\text{next})} \end{bmatrix} \end{cases} \quad (19)$$

The updated training set $(\mathbf{x}^{\text{trn}}, \mathbf{z}^{\text{trn}}, \mathbf{l}^{\text{trn}})$ is used to refine $\hat{L}(\mathbf{X}, \mathbf{Z})$. Then $\hat{L}_{\max}(\mathbf{X}, \Omega)$ in Eq. (16) and hence $E_L(\Omega)$ in Eq. (14) are also updated. With similar procedures, training points are iteratively added into the training set, and $E_L(\Omega)$ is updated iteratively until stopping criteria are satisfied.

4.4. STOPPING CRITERIA

In this subsection, we discuss two stopping criteria in Steps 4 and 7 shown in Figure 1. The purpose of the stopping criterion in Step 4 is to judge whether more training points are necessary to update $\hat{L}(\mathbf{X}, \mathbf{Z})$. A straightforward stopping criterion is

$$\max_{\mathbf{x} \in \mathbf{x}^{\text{MCS}}, \mathbf{z} \in \mathbf{z}^{\Omega}} |\text{EI}(\mathbf{x}, \mathbf{z}) / \hat{L}_{\max}(\mathbf{x}, \Omega)| \leq c \quad (20)$$

where c is a threshold, which usually takes a small positive number, such as 0.005. This stopping criterion guarantees that for any $\mathbf{x} \in \mathbf{x}^{\text{MCS}}$, the absolute value of the expected improvement rate of $\hat{L}_{\max}(\mathbf{x}, \Omega)$ is small enough. In other words, this stopping criterion guarantees that the n_{MCS} samples of $\hat{L}_{\max}(\mathbf{X}, \Omega)$ are all accurate enough so that $E_L(\Omega)$ is accurate enough. The threshold c , however, does not directly measure the accuracy of $E_L(\Omega)$. As a result, it is hard to determine the proper value for c . If we set a too-small value

to c , it may result in unnecessary iterations and hence an unnecessary computational cost.

To resolve this problem, we propose a new stopping criterion

$$W = \left| \left\{ \text{mean}_{\mathbf{x} \in \mathbf{x}^{\text{MCS}}} \left[\max_{\mathbf{z} \in \mathbf{z}^{\Omega}} \text{EI}(\mathbf{x}, \mathbf{z}) \right] \right\} / E_L(\Omega) \right| \leq w \quad (21)$$

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

Since $\max_{\mathbf{z} \in \mathbf{z}^{\Omega}} \text{EI}(\mathbf{x}, \mathbf{z})$ is the maximum expected improvement of $\hat{L}_{\max}(\mathbf{x}, \Omega)$, $\text{mean}_{\mathbf{x} \in \mathbf{x}^{\text{MCS}}} \left[\max_{\mathbf{z} \in \mathbf{z}^{\Omega}} \text{EI}(\mathbf{x}, \mathbf{z}) \right]$ is the expected maximum improvement of $E_L(\Omega)$. Then, W is the absolute value of the expected improvement rate of $E_L(\Omega)$. W directly measures the accuracy of $E_L(\Omega)$, and so we can set the value of w according to specific engineering requirements. For example, if we set $w = 0.005$, no more training points will be added if adding more training points can change current $E_L(\Omega)$ by no more than 0.5%. As a result, if n_{MCS} is sufficiently large, the relative error of the obtained $E_L(\Omega)$ is expected to be between -0.5% and 0.5% .

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

$$\Gamma = \frac{\sigma(\Omega)}{E_L(\Omega) \sqrt{n_{\text{MCS}}}} \quad (22)$$

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

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

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

$$\Gamma \leq \gamma \quad (24)$$

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

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

$$n_{\text{MCS}} \geq \left[\frac{\sigma(\Omega)}{E_L(\Omega)\gamma} \right]^2 \quad (25)$$

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

$$n_{\text{add}} = n_0 - n_{\text{MCS}} \quad (26)$$

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

$$n_{\text{add}} = \begin{cases} \tilde{n}_{\text{add}}, & \text{if } n_0 - n_{\text{MCS}} > \tilde{n}_{\text{add}} \\ n_0 - n_{\text{MCS}}, & \text{otherwise} \end{cases} \quad (27)$$

4.5. IMPLEMENTATION

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

The strategy of the extreme value in this study is similar to what the nested extreme response surface approach [58] employs because both methods use the same EGO to solve the global optimization problem. But the problem in the former method is the multidimensional global optimization with respect to time and space while the problem in the latter method is a unidimensional one with respect to time. As a result, the learning functions and stopping criteria of the two methods are different.

4.6. EXTENSION TO PROBLEMS WITH INPUT RANDOM FIELDS

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

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

where $\mathbf{H}(\mathbf{Z})$ is a vector of random fields. To easily present the idea, we assume there is only one random field, given by $H(\mathbf{Z})$. Widely used series expansions for random fields include, but are not limited to, the Karhunen-Loeve series expansion (K-L), the orthogonal series expansion (OSE), and the expansion optimal linear estimation method (EOLE) [48]. Since Ω is discretized into \mathbf{z}^Ω , the autocorrelation coefficient function of $H(\mathbf{Z})$ is

discretized into the autocorrelation coefficient matrix \mathbf{M}_H with dimension $n_\Omega \times n_\Omega$. Then the EOLE expansion $H(\boldsymbol{\xi}, \mathbf{Z})$ of $H(\mathbf{Z})$ is given by

$$H(\boldsymbol{\xi}, \mathbf{z}) = \mu_H(\mathbf{z}) + \sigma_H(\mathbf{z}) \sum_{k=1}^{n_\Omega} \frac{\xi_k}{\sqrt{\lambda_k}} \mathbf{V}_k \mathbf{M}_H(:, k), \mathbf{z} \in \mathbf{z}^\Omega \quad (29)$$

where $\mu_H(\mathbf{z})$ is the mean value function of $H(\mathbf{Z})$, $\sigma_H(\mathbf{z})$ is the standard deviation function of $H(\mathbf{Z})$, $\xi_k, k = 1, 2, \dots, n_\Omega$ are n_Ω independent standard Gaussian variables, λ_k is the k -th eigenvalue of \mathbf{M}_H , \mathbf{V}_k is the k -th (row) eigenvector of \mathbf{M}_H , and $\mathbf{M}_H(:, k)$ is the k -th column of \mathbf{M}_H . Note that the eigenvalues are sorted from the largest to the smallest. Usually only the first p ($p \leq n_\Omega$) eigenvalues are significant. Therefore, Eq. (29) is practically truncated, and only the first p orders are kept:

$$H(\boldsymbol{\xi}, \mathbf{z}) = \mu_H(\mathbf{z}) + \sigma_H(\mathbf{z}) \sum_{k=1}^p \frac{\xi_k}{\sqrt{\lambda_k}} \mathbf{V}_k \mathbf{M}_H(:, k), \mathbf{z} \in \mathbf{z}^\Omega \quad (30)$$

Then the dimension of $\boldsymbol{\xi}$ is p . With the expansion, Eq. (28) is rewritten as

$$Y = g[\mathbf{X}, H(\boldsymbol{\xi}, \mathbf{Z}), \mathbf{Z}] \quad (31)$$

or equivalently as

$$Y = g(\tilde{\mathbf{X}}, \mathbf{Z}) \quad (32)$$

where $\tilde{\mathbf{X}} = (\boldsymbol{\xi}, \mathbf{X})$. Eq. (32) shares the same format with Eq. (8), and hence the above implementation in Subsection 4.5 is also applicable.

The direct implementation this way, however, may suffer from the curse of dimensionality. Since many random variables, i.e. $\boldsymbol{\xi}$, are in the series expansion $H(\boldsymbol{\xi}, \mathbf{Z})$, the dimension of $\boldsymbol{\xi}$ and hence that of $g(\tilde{\mathbf{X}}, \mathbf{Z})$ is high. As a result, the dimension of the surrogate model $\hat{L}(\tilde{\mathbf{X}}, \mathbf{Z})$ is also high. The high-dimensional surrogate model has as least

two drawbacks. First, it is not cheap anymore, losing its expected advantages. Second, more training points are needed for acceptable accuracy. To overcome the drawbacks, we build a surrogate model $\hat{L}(\mathbf{X}, H, \mathbf{Z})$ with respect to \mathbf{X}, H , and \mathbf{Z} . Note that the entire random field H is treated as only one variable for $\hat{L}(\mathbf{X}, H, \mathbf{Z})$. Then the surrogate model $\hat{L}(\tilde{\mathbf{X}}, \mathbf{Z})$ with respect to $\tilde{\mathbf{X}}$ and \mathbf{Z} is obtained through

$$\hat{L}(\tilde{\mathbf{X}}, \mathbf{Z}) = \hat{L}[\mathbf{X}, H(\boldsymbol{\xi}, \mathbf{Z}), \mathbf{Z}] \quad (33)$$

Since the truncated series expansion $H(\boldsymbol{\xi}, \mathbf{Z})$ in Eq. (30) has a simple closed-form expression, if $\hat{L}(\mathbf{X}, H, \mathbf{Z})$ is accurate and efficient, so will be $\hat{L}(\tilde{\mathbf{X}}, \mathbf{Z})$ in Eq. (33). Since the dimension of $\hat{L}(\mathbf{X}, H, \mathbf{Z})$ is $(p - 1)$ lower than that of $\hat{L}(\tilde{\mathbf{X}}, \mathbf{Z})$, it is more efficient to train $\hat{L}(\mathbf{X}, H, \mathbf{Z})$ with higher accuracy. To build $\hat{L}(\mathbf{X}, H, \mathbf{Z})$, we need the training points \mathbf{h}^{trn} of H . \mathbf{h}^{trn} can be obtained simply by substituting $(\boldsymbol{\xi}^{\text{trn}}, \mathbf{z}^{\text{trn}})$ into Eq. (30). Similarly, when $(\tilde{\mathbf{x}}^{(\text{next})}, \mathbf{z}^{(\text{next})})$ is determined by Eq. (18), the next training point $h^{(\text{next})}$ of H is obtained by substituting $(\boldsymbol{\xi}^{(\text{next})}, \mathbf{z}^{(\text{next})})$ into Eq. (30). Note that $\tilde{\mathbf{x}}^{(\text{next})} = (\boldsymbol{\xi}^{(\text{next})}, \mathbf{x}^{(\text{next})})$. When more than one input random fields are involved, the procedure of building and updating the surrogate model \hat{L} is similar.

However, it should be mentioned that Eq. (33) is not suitable for all problems involving input random fields. Roughly speaking, the problems involving input random fields, including random processes which are unidimensional random fields, can be grouped into two categories. To distinguish the two categories, we first need to make it clear that whenever \mathbf{Z} in Eq. (28) is fixed to a specific realization $\mathbf{z}^{(i)}$, $\mathbf{H}(\mathbf{z}^{(i)})$ and $g(\mathbf{X}, \mathbf{H}(\mathbf{z}^{(i)}), \mathbf{z}^{(i)})$ are a random vector and a random variable, respectively. If the randomness, or uncertainty, of the output random variable $g(\mathbf{X}, \mathbf{H}(\mathbf{z}^{(i)}), \mathbf{z}^{(i)})$ only comes

from the input random variables \mathbf{X} and $\mathbf{H}(\mathbf{z}^{(i)})$, then the problem belongs to Category 1. If the randomness of $g(\mathbf{X}, \mathbf{H}(\mathbf{z}^{(i)}), \mathbf{z}^{(i)})$ comes from not only \mathbf{X} and $\mathbf{H}(\mathbf{z}^{(i)})$ but also $\mathbf{H}(\mathbf{z}^{(j)}), j \neq i$, then the problem belongs to Category 2.

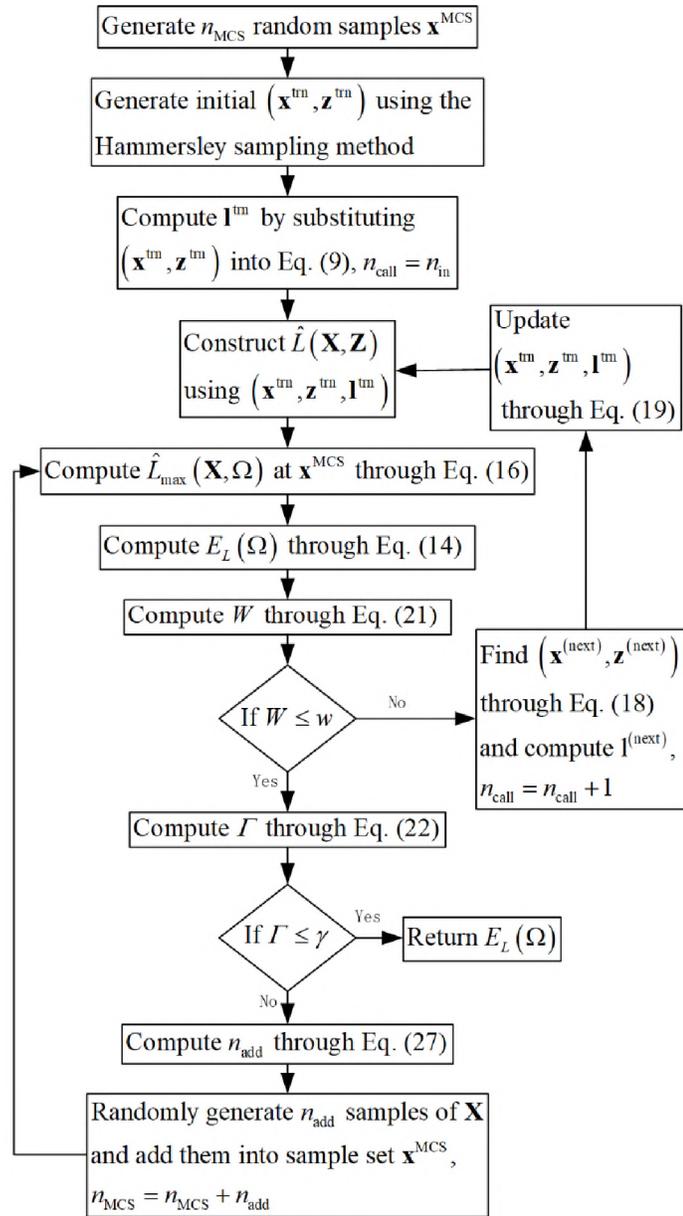


Figure 2. Detailed flowchart

Eq. (33) can only deal with problems in Category 1. When dealing with Category 2 problems, we cannot treat the entire random field as a single input variable to \hat{L} anymore. Instead, we must treat each component of ξ as an input variable to \hat{L} , resulting in a high-dimension Gaussian process model. Currently, the Gaussian process model cannot work well for high-dimensional problems. Therefore, we only consider Category 1 when input random fields are involved, and in the example section, both Example 3 and Example 4 belong to Category 1.

5. NUMERICAL EXAMPLES

In this section, we use four examples to test the proposed method. The first one is a mathematical example. With a low-dimension case in this example, we aim at clearly illustrating the detailed procedure of the proposed method. Then we test the method by setting a higher dimensionality for this example. The second one is an engineering example involving only random variables while the third one, also an engineering example, involves both random variables and unidimensional random fields. The last engineering example involves multidimensional random fields.

The direct MCS is also used to compute the TSD robustness metric. MCS calls the original QLF model in Eq. (11) directly. The sample size of MCS is set to 10^5 . The results of MCS are treated as the exact ones for the accuracy comparison. For all examples, the convergence thresholds w , γ , and n_{in} , are set to 0.005, 0.005, and 10, respectively. Both methods share the same discretization of Ω .

5.1. A MATH PROBLEM

The QC is given by

$$Y = \sum_{i=1}^N X_i^2 + 0.1(Z_1 + Z_2 + 5)^2 \sin(0.1Z_2) \prod_{i=1}^N X_i \quad (34)$$

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

$$m(\mathbf{Z}) = 0.1(Z_1 + Z_2 + 5)^2 \sin(0.1Z_2) \quad (35)$$

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

Table 1. Initial training points in hypercube space

Point number	Data			
1	0.0000	0.5000	0.3333	0.2000
2	0.1000	0.2500	0.6667	0.4000
3	0.2000	0.7500	0.1111	0.6000
4	0.3000	0.1250	0.4444	0.8000
5	0.4000	0.6250	0.7778	0.0400
6	0.5000	0.3750	0.2222	0.2400
7	0.6000	0.8750	0.5556	0.4400
8	0.7000	0.0625	0.8889	0.6400
9	0.8000	0.5625	0.0370	0.8400
10	0.9000	0.3125	0.3703	0.0800

For an easy demonstration, we first consider a low-dimension case and set $N = 2$.

Using the Hammersley sampling, we generate 10 initial training points in the hypercube

$[0,1]^{N+N_z} = [0,1]^4$. The initial training points are given in Table 1. Then we transform the first two data columns into Z-space, using the inverse probability method mentioned in Subsection 4.2, to get the initial training points \mathbf{z}^{in} . Note that each training point in \mathbf{z}^{in} is rounded to the nearest one in \mathbf{z}^{Ω} . The last two data columns are transformed into X-space to generate \mathbf{x}^{in} . Substituting those points in $(\mathbf{x}^{\text{in}}, \mathbf{z}^{\text{in}})$ one by one into Eq. (9), we obtain 10 initial training points \mathbf{l}^{in} of L . \mathbf{z}^{in} , \mathbf{x}^{in} and \mathbf{l}^{in} are given in Table 2. Since Eq. (9) is called 10 times, $n_{\text{call}} = 10$. Initially, we set $n_{\text{MCS}} = 400$.

Table 2. Initial training points in X-space and Z-space

Point number	\mathbf{z}^{in}		\mathbf{x}^{in}		\mathbf{l}^{in}
1	0.0000	2.5000	0.9914	0.9832	3664.4
2	0.2000	1.2500	1.0086	0.9949	4036.1
3	0.4000	3.7500	0.9756	1.0051	3618.5
4	0.6000	0.6250	0.9972	1.0168	4128.0
5	0.8000	3.1250	1.0153	0.9650	3657.3
6	1.0000	1.8750	0.9847	0.9858	3639.8
7	1.2000	4.3750	1.0028	0.9970	3993.8
8	1.4000	0.3125	1.0244	1.0072	4277.8
9	1.6000	2.8125	0.9643	1.0199	3722.3
10	1.8000	1.5625	0.9934	0.9719	3586.5

With the initial training points, we build the initial $\hat{L}(\mathbf{X}, \mathbf{Z})$. Then using Eq. (14) and Eq. (21) we obtain $E_L(\Omega) = \$ 4044.5$ and $W = 1.41\%$, respectively. $W = 1.41\%$ means that if we add more training points to update $\hat{L}(\mathbf{X}, \mathbf{Z})$, we expect to improve the current $E_L(\Omega)$ by 1.41%. Since 1.41% is larger than the threshold value 0.5%, more training points are needed. The learning function in Eq. (18) locates the next training point $(\mathbf{z}^{(\text{next})}, \mathbf{x}^{(\text{next})})$ at $(0, 0, 0.9263, 0.9630)$. Substituting $(\mathbf{z}^{(\text{next})}, \mathbf{x}^{(\text{next})})$ into Eq. (9), we

obtain $l^{(\text{next})} = 3188.0$ and then update $n_{\text{call}} = n_{\text{call}} + 1 = 11$. The new training point is added to the current training set to update $\hat{L}(\mathbf{X}, \mathbf{Z})$. With the process going on, more and more training points are added. In total 7 training points are added one by one, which are given in Table 3. n_{call} becomes $10 + 7 = 17$. The update of W is shown in Figure 3. In Iteration 8, the 7th training point shown in Table 3 is added to update $\hat{L}(\mathbf{X}, \mathbf{Z})$, resulting in $W = 0.45\% < 0.5\%$. Therefore, no more new training points are needed. Note that in all the eight iterations, $n_{\text{MCS}} = 400$ remains unchanged.

Table 3. Added training points

Iteration	$\mathbf{z}^{(\text{next})}$		$\mathbf{x}^{(\text{next})}$		$l^{(\text{next})}$
1	0.0000	0.0000	0.9263	0.9630	3188.0
2	2.0000	0.0000	0.9769	1.0544	4269.2
3	2.0000	5.0000	0.9263	0.9630	1082.4
4	0.0000	5.0000	1.0557	0.9859	5208.2
5	0.0000	5.0000	1.0222	1.0460	6107.1
6	2.0000	0.0000	1.0161	0.9451	3707.9
7	2.0000	0.0000	0.9330	0.9927	3444.1

To check if 400 samples are sufficient to obtain accurate $E_L(\Omega)$, we calculate Γ using Eq. (22), which results in $\Gamma = 0.0064$. Since 0.0064 is larger than the threshold $\gamma = 0.005$, the sample size $n_{\text{MCS}} = 400$ is not sufficiently large and hence we need to increase it. From Eqs. (25) and (26), we know that n_{MCS} should be increased by at least 247. However, according to Eq. (27), we only increase it by 100, because we set the hyperparameter $\tilde{n}_{\text{add}} = 100$. The reason for limiting the increasing step of n_{MCS} has been given in Subsection 4.4. Then with the updated $n_{\text{MCS}} = 400 + 100 = 500$ and updated \mathbf{x}^{MCS} , we calculate W again to check if $\hat{L}(\mathbf{X}, \mathbf{Z})$ is still accurate. Figure 3 shows that $W <$

0.5% in Iteration 9; hence $\hat{L}(\mathbf{X}, \mathbf{Z})$ is still accurate. We calculate Γ and then W again and repeat the process until both $W < 0.5\%$ and $\Gamma < 0.005$ are satisfied.

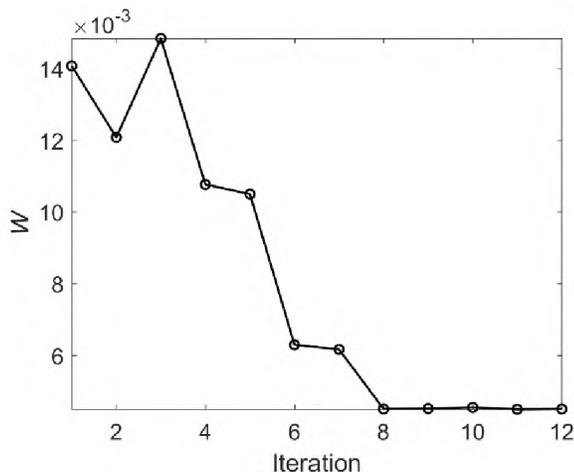


Figure 3. Update of W

Table 4. Robustness analysis results

Methods	Proposed method	MCS
$E_L(\Omega)(\$)$	4.28×10^3	4.35×10^3
Relative error (%)	-1.5	-
n_{MCS}	677	10^5
n_{call}	17	10^8

The final results, as well as the results obtained directly by MCS, are given in Table 4. The robustness computed by the proposed method is $\$4.28 \times 10^3$, and the robustness by MCS is $\$4.35 \times 10^3$. The proposed method is very accurate, with a small relative error of -1.5% . In addition to the 10 initial training points, 7 more training points are added adaptively to update the Gaussian process model, and hence the proposed method costs 17

function calls. The proposed method adaptively increases the sample size, obtaining accurate results with only 677 samples.

To test the proposed method with higher dimensionality, we set $N = 8$ while keeping other parameters unchanged. The results obtained from the proposed method and MCS are given in Table 5. The robustness computed by the proposed method is $\$6.70 \times 10^4$, and the robustness by MCS is $\$6.67 \times 10^4$. The relative error is 0.5%, and 70 function calls and only 400 samples are used by the proposed method. In this case, $N = 8$ and $N_Z = 2$, and hence the dimensionality of $\hat{L}(\mathbf{X}, \mathbf{Z})$ is 10. The two cases show that the proposed method works well for both low dimensions and moderate dimensions in this example problem.

Table 5. Robustness analysis results

Methods	Proposed method	MCS
$E_L(\Omega)(\$)$	6.70×10^4	6.67×10^4
Relative error (%)	0.5	-
n_{MCS}	400	10^5
n_{call}	70	10^8

Note that the second case needs a smaller sample size ($n_{\text{MCS}} = 400$) than that ($n_{\text{MCS}} = 677$) of the first case, although the dimensionality is higher in the second case.

The reason is that the deviation coefficient $\frac{\sigma(\Omega)}{E_L(\Omega)}$ of $L_{\max}(\mathbf{X}, \Omega)$ in the first case is larger than that in the second case.

5.2. A SLIDER MECHANISM

Shown in Figure 4 is a slider mechanism [39]. The spatial variables are the offset H and the initial angle θ_0 with the following ranges: $H \in [14.85, 15.15]$ m and $\theta_0 \in [-2^\circ, 2^\circ]$. The time span is $t \in [0, 0.1\pi]$ s. Then the \mathbf{Z} vector is (H, θ_0, t) . The random variable vector is $\mathbf{X} = (L_1, L_2)$, which includes two independent random link lengths $L_1 \sim N(15, 0.015^2)$ m and $L_2 \sim N(35, 0.035^2)$ m. The QC, or the actual position of the slider, is

$$Y = L_1 \cos(\theta_0 + \omega t) + \sqrt{L_2^2 - (H + L_1 \sin(\theta_0 + \omega t))^2} \quad (36)$$

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

$$m(\mathbf{Z}) = 15 \cos(\omega t) + \sqrt{35^2 - (15 + 15 \sin(\omega t))^2} \quad (37)$$

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

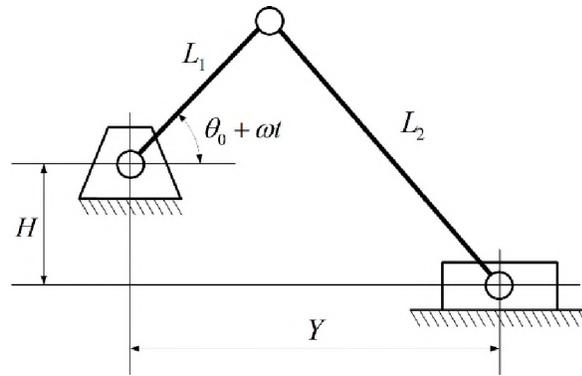


Figure 4. A slider mechanism [39]

Setting the initial value of n_{MCS} to 1000 and \tilde{n}_{add} to 100, the robustness analysis results are given in Table 6. The proposed method is accurate and efficient with 31 function calls.

Table 6. Robustness analysis results

Methods	Proposed method	MCS
$E_L(\Omega)$ (\$)	88.43	88.02
Relative error (%)	0.5	-
n_{MCS}	1492	10^5
n_{call}	31	8×10^8

5.3. A CANTILEVER BEAM

Shown in Figure 5 is a cantilever beam. Its span $L = 1$ m. Due to the machining error, the diameter of its cross-section is not deterministic. Instead, it is modeled as a one-dimensional stationary Gaussian random field $D(x)$. The mean value μ_D and standard deviation σ_D of $D(x)$ are 0.1 m and 0.001 m, respectively. Its autocorrelation coefficient function $\rho_D(x_1, x_2)$ is given as

$$\rho_D(x_1, x_2) = \exp[-(x_1 - x_2)^2] \quad (38)$$

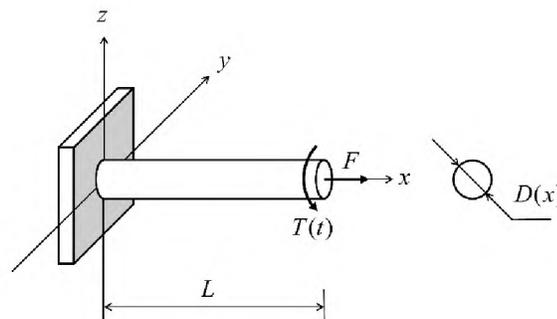


Figure 5. A cantilever beam

The beam is subjected to a torsion $T(t)$ and a tensile force F at the right endpoint. F is a normal variable with mean μ_F and standard deviation σ_F being 1000 N and 100 N, respectively. $T(t)$ is a stationary Gaussian process with mean μ_T and standard deviation σ_T being 200 N · m and 20 N · m, respectively. Its autocorrelation coefficient function $\rho_T(t_1, t_2)$ is given by

$$\rho_T(t_1, t_2) = \exp\left[-\left(\frac{t_1 - t_2}{2}\right)^2\right] \quad (39)$$

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

$$Y = \sqrt{\left[\frac{4F}{\pi D(x)^2}\right]^2 + 3\left[\frac{16T(t)}{\pi D(x)^3}\right]^2} \quad (40)$$

The target $m(\mathbf{Z}) = 0$ and $A(\mathbf{Z}) = \$1000/(\text{Mpa})^2$. The domain Ω of $\mathbf{Z} = (x, t)$ is $[0, 1] \text{ m} \times [0, 5] \text{ yr}$ and is evenly discretized into $n_\Omega = 20 \times 50 = 1000$ points.

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

Setting the initial value of n_{MCS} to 1000 and \tilde{n}_{add} to 1000, the robustness analysis results are given in Table 7. The robustness computed by the proposed method and by MCS are $\$3.85 \times 10^3$ and 3.88×10^3 , respectively. The relative error of the robustness computed by the proposed method is only -0.7% . The proposed method calls the original quality loss function 13 times, showing its high efficiency.

Table 7. Robustness analysis results

Methods	Proposed method	MCS
$E_L(\Omega)$ (\$)	3.85×10^3	3.88×10^3
Relative error (%)	-0.7	-
n_{MCS}	1000	10^5
n_{call}	13	10^8

5.4. AN ELECTRON ACCELERATOR

Shown in Figure 6 is an electron accelerator, which is used to accelerate electrons to a higher speed. Electrons are horizontally emitted from the electrode, then enter the electric field $E(w, h)$ in the accelerator, and finally fly out from the accelerator. The initial velocity of the electrons is a time-dependent stationary Gaussian random field $V_0(w, h, t)$, whose mean value μ_{V_0} and standard deviation σ_{V_0} are 500,000 m/s and 50,000 m/s, respectively. Its autocorrelation coefficient function $\rho_{V_0}(w_1, h_1, t_1; w_2, h_2, t_2)$ is given by

$$\rho_{V_0}(w_1, h_1, t_1; w_2, h_2, t_2) = \exp \left[- \left(\frac{w_1 - w_2}{0.2} \right)^2 - \left(\frac{h_1 - h_2}{0.2} \right)^2 - \left(\frac{t_1 - t_2}{4} \right)^2 \right] \quad (41)$$

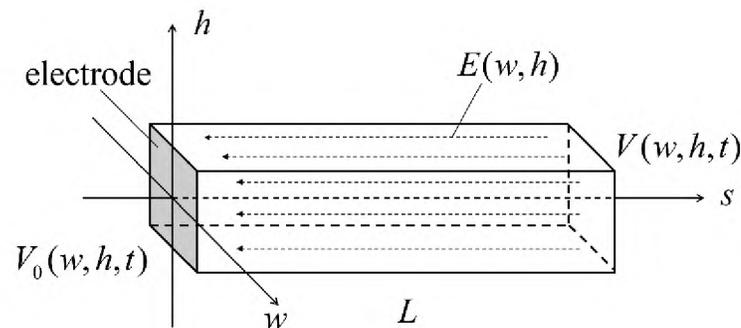


Figure 6. An electron accelerator

The length of the accelerator L is a random variable that follows a normal distribution $N(1, 0.01^2)$ m. The electric field $E(w, h)$ is a two-dimensional stationary Gaussian random field, whose mean value μ_E and standard deviation σ_E are 10 N/C and 1 N/C, respectively. Its autocorrelation coefficient function $\rho_E(w_1, h_1; w_2, h_2)$ is given by

$$\rho_E(w_1, h_1; w_2, h_2) = \exp \left[- \left(\frac{w_1 - w_2}{0.1} \right)^2 - \left(\frac{h_1 - h_2}{0.1} \right)^2 \right] \quad (42)$$

If the acceleration time and the interaction among the electrons are negligible, the velocity $V(w, h, t)$ of the electrons after acceleration is given by

$$V(w, h, t) = \sqrt{\frac{2qE(w, h)L}{m} + V_0^2(w, h, t)} \quad (43)$$

where $q = 1.6 \times 10^{-19}$ C and $m = 9.109 \times 10^{-31}$ kg are the electric quantity and mass of an electron, respectively. The target velocity V_t is given by

$$V_t = \sqrt{\frac{2q\mu_E L}{m} + \mu_{V_0}^2} \quad (44)$$

In this example, $\mathbf{Z} = (w, h, t) \in \Omega = [-0.05, 0.05] \text{ m} \times [-0.05, 0.05] \text{ m} \times [0, 10] \text{ s}$ and $m(\mathbf{Z}) = 10^{-8}/(\text{m/s})^2$. Ω is evenly discretized into $n_\Omega = 10 \times 10 \times 20 = 2000$ points. We also use EOLE to generate the series expansions of both $V_0(w, h, t)$ and $E(w, h)$, and the first 20 and 8 orders are kept, respectively.

Setting the initial value of n_{MCS} to 2×10^4 and \tilde{n}_{add} to 10^3 , the robustness analysis results are given in Table 8. The robustness computed by the proposed method is \$ 271.25. Again, the proposed method is both accurate, with a relative error being -0.1%, and efficient, with only 40 function evaluations. Although two multidimensional random fields are involved, the efficiency is still high. The high efficiency is achieved by using the

method described in Subsection 4.6, and the dimension of this problem is only three, including one random variable and two random fields.

Table 8. Robustness analysis results

Methods	Proposed method	MCS
$E_L(\Omega)$ (\$)	271.25	271.44
Relative error (%)	-0.1	-
n_{MCS}	22476	10^5
n_{call}	40	2×10^8

6. CONCLUSIONS

Existing robustness analysis methods only consider static or time-dependent problems. More general are time-and space-dependent problems. In this paper, a new robustness metric is proposed for time-and space-dependent problems. The new metric has the following features:

- The robustness is measured by the expected maximum quality loss over the domain of interest, which consists of the space and period of time under consideration.
- This metric can fully take into consideration of the autocorrelation of the time-and space-dependent quality loss function at all points of the domain of interest.
- Minimizing the expected maximum quality loss will reduce both the deviation of a quality characteristic (QC) from its target and the standard deviation of the QC in the domain of interest, thereby maximizing the robustness.

An efficient robustness analysis method is developed to quantify the robustness metric based on the Gaussian process model and efficient global optimization. The method can accommodate QCs that are general functions of random variables, random processes, and random fields, temporal variables, and spatial variables.

Possible future work includes the following tasks: Further improve the efficiency of the proposed robustness analysis method, develop other robustness analysis methods, and investigate possible robustness metrics when multiple time- and space-dependent QCs are considered.

ACKNOWLEDGEMENTS

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

REFERENCES

- [1] Taguchi, G., 1993, Taguchi on robust technology development: bringing quality engineering upstream, ASME Press, New York.
- [2] Phadke, M. S., 1995, Quality engineering using robust design, Prentice Hall PTR.
- [3] Du, X., and Chen, W., 2000, "Towards a better understanding of modeling feasibility robustness in engineering design," *Journal of Mechanical Design*, 122(4), pp. 385-394.
- [4] Huang, B., and Du, X., 2007, "Analytical robustness assessment for robust design," *Structural and Multidisciplinary Optimization*, 34(2), pp. 123-137.
- [5] Mulvey, J. M., Vanderbei, R. J., and Zenios, S. A., 1995, "Robust optimization of large-scale systems," *Operations research*, 43(2), pp. 264-281.

- [6] Goh, J., and Sim, M., 2010, "Distributionally robust optimization and its tractable approximations," *Operations research*, 58(4-part-1), pp. 902-917.
- [7] Fabozzi, F. J., Huang, D., and Zhou, G., 2010, "Robust portfolios: contributions from operations research and finance," *Annals of Operations Research*, 176(1), pp. 191-220.
- [8] Alexandrov, N. M., and Lewis, R. M., 2002, "Analytical and computational aspects of collaborative optimization for multidisciplinary design," *AIAA journal*, 40(2), pp. 301-309.
- [9] Sobieszczanski-Sobieski, J., and Haftka, R. T., 1997, "Multidisciplinary aerospace design optimization: survey of recent developments," *Structural optimization*, 14(1), pp. 1-23.
- [10] Doltsinis, I., Kang, Z., and engineering, 2004, "Robust design of structures using optimization methods," *Computer Methods in Applied Mechanics and Engineering*, 193(23-26), pp. 2221-2237.
- [11] Lagaros, N. D., Plevris, V., and Papadrakakis, M., 2007, "Reliability based robust design optimization of steel structures," *International Journal for Simulation Multidisciplinary Design Optimization*, 1(1), pp. 19-29.
- [12] Cheng, J., Liu, Z., Wu, Z., Li, X., and Tan, J., 2015, "Robust optimization of structural dynamic characteristics based on adaptive Kriging model and CNSGA," *Structural and Multidisciplinary Optimization*, 51(2), pp. 423-437.
- [13] Roy, B. K., and Chakraborty, S., 2015, "Robust optimum design of base isolation system in seismic vibration control of structures under random system parameters," *Structural Safety*, 55, pp. 49-59.
- [14] Fang, J., Gao, Y., Sun, G., and Li, Q., 2013, "Multiobjective reliability-based optimization for design of a vehicledoor," *Finite Elements in Analysis and Design*, 67, pp. 13-21.
- [15] Hwang, K., Lee, K., and Park, G., 2001, "Robust optimization of an automobile rearview mirror for vibration reduction," *Structural and Multidisciplinary Optimization*, 21(4), pp. 300-308.
- [16] Sun, G., Li, G., Zhou, S., Li, H., Hou, S., and Li, Q., 2011, "Crashworthiness design of vehicle by using multiobjective robust optimization," *Structural and Multidisciplinary Optimization*, 44(1), pp. 99-110.
- [17] Lee, T. H., and Jung, J., 2006, "Metamodel-based shape optimization of connecting rod considering fatigue life," *Key Engineering Materials*, 306-308, pp. 211-216.

- [18] Li, F., Meng, G., Sha, L., and Zhou, L., 2011, "Robust optimization design for fatigue life," *Finite Elements in Analysis and Design*, 47(10), pp. 1186-1190.
- [19] Carpinelli, G., Caramia, P., and Varilone, P., 2015, "Multi-linear Monte Carlo simulation method for probabilistic load flow of distribution systems with wind and photovoltaic generation systems," *Renewable Energy*, 76, pp. 283-295.
- [20] Li, F., Luo, Z., Sun, G., and Zhang, N., 2013, "An uncertain multidisciplinary design optimization method using interval convex models," *Engineering Optimization*, 45(6), pp. 697-718.
- [21] Du, X., Guo, J., and Beeram, H., 2008, "Sequential optimization and reliability assessment for multidisciplinary systems design," *Structural and Multidisciplinary Optimization*, 35(2), pp. 117-130.
- [22] Kim, N.-K., Kim, D.-H., Kim, D.-W., Kim, H.-G., Lowther, D. A., and Sykulski, J. K., 2010, "Robust optimization utilizing the second-order design sensitivity information," *IEEE transactions on Magnetics*, 46(8), pp. 3117-3120.
- [23] Papadimitriou, D., and Giannakoglou, K., 2013, "Third-order sensitivity analysis for robust aerodynamic design using continuous adjoint," *International Journal for Numerical Methods in Fluids*, 71(5), pp. 652-670.
- [24] Li, M., Hamel, J., and Azarm, S., 2010, "Optimal uncertainty reduction for multi-disciplinary multi-output systems using sensitivity analysis," *Structural and Multidisciplinary Optimization*, 40(1-6), p. 77.
- [25] Siddiqui, S., Azarm, S., and Gabriel, S., 2011, "A modified Benders decomposition method for efficient robust optimization under interval uncertainty," *Structural and Multidisciplinary Optimization*, 44(2), pp. 259-275.
- [26] Siddiqui, S., Gabriel, S. A., and Azarm, S., 2015, "Solving mixed-integer robust optimization problems with interval uncertainty using Benders decomposition," *Journal of the Operational Research Society*, 66(4), pp. 664-673.
- [27] Chatterjee, T., Chowdhury, R., and Ramu, P., 2019, "Decoupling uncertainty quantification from robust design optimization," *Structural and Multidisciplinary Optimization*, 59(6), pp. 1969-1990.
- [28] Chatterjee, T., Chakraborty, S., and Chowdhury, R., 2019, "A critical review of surrogate assisted robust design optimization," *Archives of Computational Methods in Engineering*, 26(1), pp. 245-274.

- [29] Steuben, J., and Turner, C. J., "Robust Optimization Exploration Using NURBs-Based Metamodeling Techniques," Proc. ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers Digital Collection, pp. 239-250.
- [30] Steuben, J. C., and Turner, C., 2012, "Robust optimization of mixed-integer problems using NURBs-based metamodels," Journal of Computing Information Science in Engineering, 12(4), p. 041010.
- [31] Steuben, J. C., Turner, C. J., and Crawford, R. H., 2013, "Robust engineering design optimization with non-uniform rational B-splines-based metamodels," Engineering Optimization, 45(7), pp. 767-786.
- [32] Zhou, H., Zhou, Q., Liu, C., and Zhou, T., 2018, "A kriging metamodel-assisted robust optimization method based on a reverse model," Engineering Optimization, 50(2), pp. 253-272.
- [33] Du, X., 2012, "Toward Time-Dependent Robustness Metrics," Journal of Mechanical Design, 134(1), p. 011004.
- [34] Du, X., Sudjianto, A., and Chen, W., 2004, "An integrated framework for optimization under uncertainty using inverse reliability strategy," Journal of Mechanical Design, 126(4), pp. 562-570.
- [35] Gu, X., Renaud, J. E., Batill, S. M., Brach, R. M., and Budhiraja, A. S., 2000, "Worst case propagated uncertainty of multidisciplinary systems in robust design optimization," Structural and Multidisciplinary Optimization, 20(3), pp. 190-213.
- [36] Tsui, K.-L., 1999, "Modeling and analysis of dynamic robust design experiments," IIE Transactions, 31(12), pp. 1113-1122.
- [37] Wu, F.-C., 2009, "Robust design of nonlinear multiple dynamic quality characteristics," Computers & Industrial Engineering, 56(4), pp. 1328-1332.
- [38] Goethals, P. L., and Cho, B. R., 2011, "The development of a robust design methodology for time-oriented dynamic quality characteristics with a target profile," Quality and Reliability Engineering International, 27(4), pp. 403-414.
- [39] Wei, X., and Du, X., 2019, "Uncertainty Analysis for Time-and Space-Dependent Responses With Random Variables," Journal of Mechanical Design, 141(2), p. 021402.
- [40] Törn, A., and Žilinskas, A., 1989, Global optimization, Springer.

- [41] Strongin, R. G., and Sergeyev, Y. D., 1992, "Global multidimensional optimization on parallel computer," *Parallel Computing*, 18(11), pp. 1259-1273.
- [42] Strongin, R. G., and Sergeyev, Y. D., 2013, *Global optimization with non-convex constraints: Sequential and parallel algorithms*, Springer Science & Business Media.
- [43] Sasena, M. J., Papalambros, P., and Goovaerts, P., 2002, "Exploration of metamodeling sampling criteria for constrained global optimization," *Engineering Optimization*, 34(3), pp. 263-278.
- [44] Williams, C. K., and Rasmussen, C. E., 2006, *Gaussian processes for machine learning*, MIT Press Cambridge, MA.
- [45] Lophaven, S. N., Nielsen, H. B., and Søndergaard, J., 2002, *DACE: a Matlab kriging toolbox*, Citeseer.
- [46] Zhu, Z., and Du, X., 2016, "Reliability analysis with Monte Carlo simulation and dependent Kriging predictions," *Journal of Mechanical Design*, 138(12), p. 121403.
- [47] Hu, Z., and Mahadevan, S., 2016, "A single-loop kriging surrogate modeling for time-dependent reliability analysis," *Journal of Mechanical Design*, 138(6), p. 061406.
- [48] Sudret, B., and Der Kiureghian, A., 2000, *Stochastic finite element methods and reliability: a state-of-the-art report*, Department of Civil and Environmental Engineering, University of California Berkeley.
- [49] Gabrel, V., Murat, C., and Thiele, A., 2014, "Recent advances in robust optimization: An overview," *European Journal of Operational Research*, 235(3), pp. 471-483.
- [50] Bovier, A., 2005, "Extreme values of random processes," *Lecture Notes Technische Universität Berlin*.
- [51] Kotz, S., and Nadarajah, S., 2000, *Extreme value distributions: theory and applications*, World Scientific, London.
- [52] Mooney, C. Z., 1997, *Monte carlo simulation*, Sage Publications.
- [53] Zienkiewicz, O. C., Taylor, R. L., Nithiarasu, P., and Zhu, J., 1977, *The finite element method*, McGraw-hill London.
- [54] Hu, Z., and Du, X., 2015, "Mixed efficient global optimization for time-dependent reliability analysis," *Journal of Mechanical Design*, 137(5), p. 051401.

- [55] Jones, D. R., Schonlau, M., and Welch, W. J., 1998, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, 13(4), pp. 455-492.
- [56] Chen, W., Tsui, K.-L., Allen, J. K., and Mistree, F., 1995, "Integration of the response surface methodology with the compromise decision support problem in developing a general robust design procedure," *ASME Design Engineering Technical Conference*, pp. 485-492.
- [57] Hosder, S., Walters, R., and Balch, M., 2007, "Efficient sampling for non-intrusive polynomial chaos applications with multiple uncertain input variables," *Proc. 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, p. 1939.
- [58] Wang, Z., and Wang, P., 2012, "A Nested Extreme Response Surface Approach for Time-Dependent Reliability-Based Design Optimization," *Journal of Mechanical Design*, 134(12), pp. 121007-121014.

V. ADAPTIVE KRIGING METHOD FOR UNCERTAINTY QUANTIFICATION OF THE PHOTOELECTRON SHEATH AND DUST LEVITATION ON THE LUNAR SURFACE

ABSTRACT

This paper presents an adaptive Kriging based method to perform uncertainty quantification (UQ) of the photoelectron sheath and dust levitation on the lunar surface. The objective of this study is to identify the upper and lower bounds of the electric potential and that of dust levitation height, given the intervals of model parameters in the 1-D photoelectron sheath model. To improve the calculation efficiency, we employ the widely used adaptive Kriging method (AKM). A task-oriented learning function and a stopping criterion are developed to train the Kriging model and customize the AKM. Experiment analysis shows that the proposed AKM is both accurate and efficient.

1. INTRODUCTION

The Moon is directly exposed to solar radiation and solar wind plasma (drifting protons and electrons) lacking an atmosphere and a global magnetic field. Consequently, the lunar surface is electrically charged by the bombardment of solar wind plasma and emission/collection of photoelectrons. Near the illuminated lunar surface, the plasma sheath is dominated by photoelectrons, thus usually referred to as “photoelectron sheath”. Additionally, dust grains on the lunar surface may get charged and levitated from the surface under the influence of the electric field within the plasma sheath as well as gravity.

This work is motivated by the high computational cost associated with uncertainty quantification (UQ) analysis of plasma simulations using high-fidelity kinetic models such as particle-in-cell (PIC). The main quantities of interest (QoI) of this study are the vertical structure of the photoelectron sheath and its effects on the levitation of dust grains with different sizes and electric charges.

Both the electric potential (ϕ) and the electric field (E) on the lunar surface are determined by many parameters, such as solar wind drifting velocity (v_d), electron temperature (T_e), photoelectron temperature (T_p), the density of ions at infinity ($n_{i,\infty}$), and density of photoelectrons (n_p), etc. Due to uncertain factors in the lunar environment, the electric potential, electric field, and the dust levitation height, etc., are also uncertain. While many sources of uncertainty may exist, they are generally categorized as either aleatory or epistemic. Uncertainties are characterized as epistemic if the modeler sees a possibility to reduce them by gathering more data or by refining models. Uncertainties are categorized as aleatory if the modeler does not foresee the possibility of reducing them [1]. An example of the aleatory uncertainty in the lunar environment is the solar wind parameters, and an example of the epistemic uncertainty is the photoelectron temperature which is obtained by limited measurement data from Apollo missions. For lunar landing missions, one needs to take into consideration the uncertainties of the electrostatic and dust environment near the lunar surface. For example, the upper and lower bounds of the electric field and dust grain levitation heights in the photoelectron sheath should be considered when determining whether it is safe for a certain area to land a spacecraft.

Determining the bounds of the electric potential, electric field, and dust levitation height, however, is computationally expensive, because the particle-based kinetic models

such as particle-in-cell simulations are time-consuming to evaluate. To address this issue, we develop an adaptive Kriging method (AKM) which can determine those bounds with a small number of calculations of the model. It is straightforward to train and obtain an accurate Kriging model [2] to replace the actual model and then calculate the bounds with the model. However, the Kriging model doesn't need to be accurate everywhere in its input space, because it will need more training samples and hence decrease the efficiency. Since the objective is to determine those bounds, we only need the Kriging model to be partially accurate near the regions of interest, as long as it can help find those bounds accurately. This way, we can save more computational efforts. To this end, we develop a task-oriented learning function and a stopping criterion to adaptively train the Kriging model. We start with an analytic model for the 1-D photoelectron sheath near the lunar surface [3, 4]. This model is computationally cheap and hence the accurate results can be obtained by brute force. With accurate results, we can test the accuracy of the proposed method. It is noted here that the ultimate application of this method is not the simple 1-D problem presented in this work, but more complicated or computationally expensive models such as 3-D fully kinetic particle-in-cell plasma simulations.

The rest of this paper is organized as follows. Section 2 presents the 1-D photoelectron sheath and dust levitation problem on the lunar surface, as well as the 1-D analytic model. Section 3 briefly introduces the Kriging method and general AKM. Section 4 presents the proposed AKM. Section 5 presents the results. Conclusions are given in Section 6.

2. PROBLEM STATEMENT

2.1. 1-D PHOTOELECTRON SHEATH MODEL ON THE LUNAR SURFACE

We employ the recently derived 1-D photoelectron sheath model for the lunar surface [3, 4]. As given in detail in [3, 4], there are three types of electric potential profiles [3-6] in the photoelectron sheath: Type A, Type B, and Type C, as shown in Figure 1, where ϕ is the electric potential and Z is the vertical coordinate. In this study, we focus on Type C sheath profile as it is expected at the polar regions of the Moon, where the next lunar landing mission will likely occur.

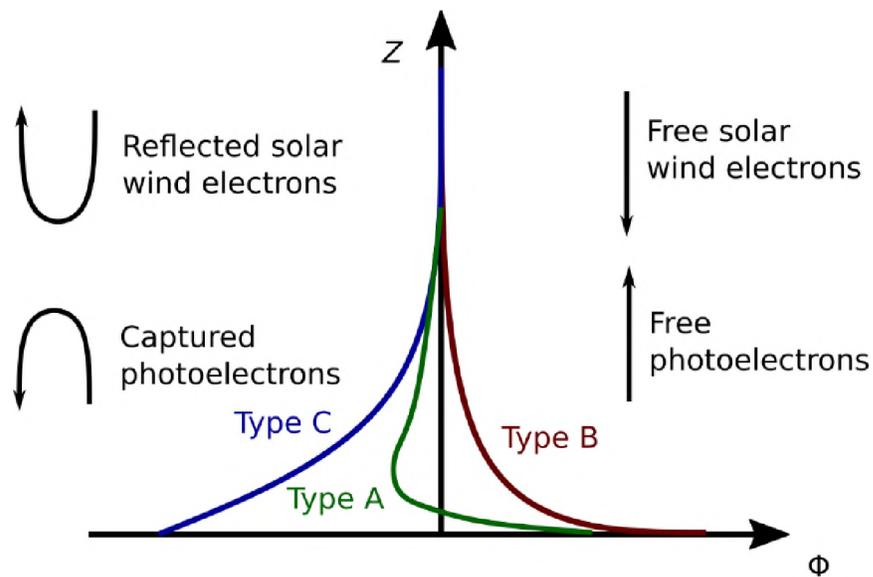


Figure 1. Three types of sheath potential profiles in the analytic 1-D photoelectron sheath model [2]

Both the electrical potential ϕ and corresponding electric field E are functions of Z with a series of parameters $\mathbf{P} = (v_d, T_e, T_p, n_{i,\infty}, n_p)$. To obtain $\phi(Z; \mathbf{P})$ and $E(Z; \mathbf{P})$, we

need to solve an ordinary differential equation (ODE) [3]. Once the potential profile ϕ is obtained, it is straightforward to calculate electric field E by

$$E(Z; \mathbf{P}) = -\frac{d\phi(Z; \mathbf{P})}{dZ} \quad (1)$$

A typical Type C sample curve of $E(Z; \mathbf{P})$ is shown in Figure 2. Note that both ϕ and E converge to zero at large values of Z where it is used as the electric potential reference (zero potential and zero field).

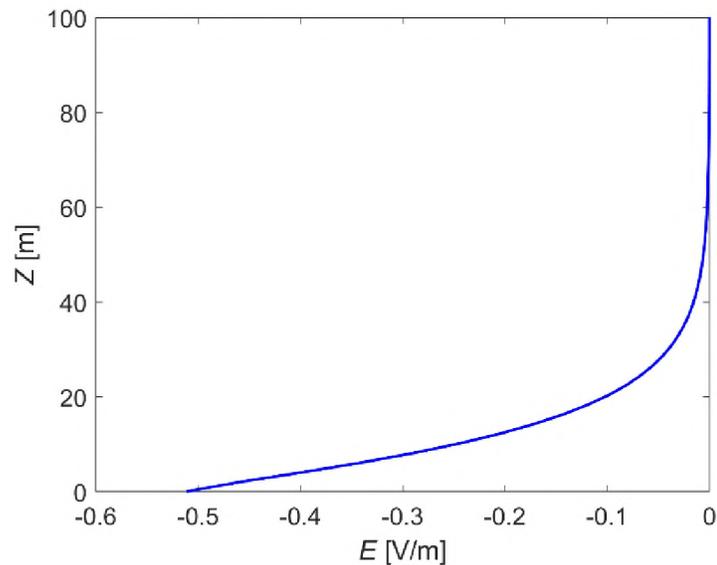


Figure 2. A typical Type C sample of $E(Z; \mathbf{P})$

2.2. DUST LEVITATION

Subjected to the electric field force, a charged dust on the lunar surface may be levitated [7, 8]. Above the lunar surface, there is a position where the upward electric field force balances the downward gravity [4]. This position is referred to as equilibrium

levitation height, denoted as Z^* . Z^* can be solved through the following equation of static equilibrium of a charged dust in an electric field:

$$qE(Z; \mathbf{P}) = mg \quad (2)$$

where q is the dust charge, m is the mass of the dust, and $g = 1.62 \text{ m/s}^2$ is the gravity acceleration on the lunar surface [9]. With the assumption of spherical dust grains, m is given by

$$m = \frac{4}{3} \pi r^3 \rho \quad (3)$$

where r is the radius of the lunar dust grain, and $\rho = 1.8 \text{ g/cm}^3$ is the mass density of dust grains [10]. For simplicity, Eq. (2) is rewritten as

$$E(Z; \mathbf{P}) = w \quad (4)$$

where $w = mg/q$. Once both $E(Z; \mathbf{P})$ and w have been given or determined, a root-finding scheme is employed to solve Eq. (4) for Z^* . Figure 3 shows an example of how to obtain Z^* graphically.

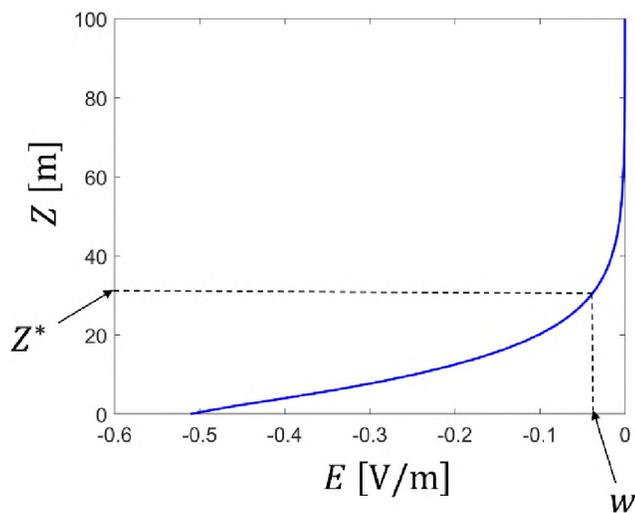


Figure 3. Method to solve for the equilibrium height of dust levitation

2.3. OBJECTIVE

Due to the lack of information, it is almost impossible to obtain the distribution functions of \mathbf{P} . The bounds of \mathbf{P} , however, are much easier to obtain. In some work designs on the lunar surface, we need to determine the bounds of $\phi(Z; \mathbf{P})$ and/or $E(Z; \mathbf{P})$, given the bounds of \mathbf{P} . In this study, all the parameters in \mathbf{P} are modeled as interval variables, whose domain is denoted as Ω . For a given realization \mathbf{p} of \mathbf{P} , both $\phi(Z; \mathbf{p})$ and $E(Z; \mathbf{p}), Z \in [Z_{\min}, Z_{\max}]$ are obtained by solving the ODE.

The upper bound $\bar{\phi}(Z)$ of the electric potential is defined as

$$\bar{\phi}(z) = \max_{\mathbf{p} \in \Omega} \phi(z; \mathbf{p}) \quad (5)$$

where z is a given value of variable Z . Note that the entire upper bound curve $\bar{\phi}(Z)$ is not necessarily determined by a specific \mathbf{p} . In other words, at different values of z , $\bar{\phi}(z)$ may be determined by different realizations of \mathbf{P} . Similarly, the lower bound $\underline{\phi}(Z)$ of the electric potential, the upper bound $\bar{E}(Z)$ of the electric field, and the lower bound $\underline{E}(Z)$ are defined as

$$\underline{\phi}(z) = \min_{\mathbf{p} \in \Omega} \phi(z; \mathbf{p}) \quad (6)$$

$$\bar{E}(z) = \max_{\mathbf{p} \in \Omega} E(z; \mathbf{p}) \quad (7)$$

$$\underline{E}(z) = \min_{\mathbf{p} \in \Omega} E(z; \mathbf{p}) \quad (8)$$

Since \mathbf{P} are modeled as interval variables and the intervals (lower and upper bounds) of output are desired, we cope with interval propagation problems in this work. The most straightforward method to determine $\bar{\phi}(Z), \underline{\phi}(Z), \bar{E}(Z)$ and $\underline{E}(Z)$ is through Monte Carlo

Simulation (MCS) [11] in the following steps. First, evenly generate a large number N_{MCS} of samples of \mathbf{P} . For convenience, we denote those samples as \mathbf{p}^{MCS} . Second, obtain the corresponding N_{MCS} samples of $\phi(Z; \mathbf{P})$ and $E(Z; \mathbf{P})$ by solving the ODE N_{MCS} times. Finally, calculate $\overline{\phi}(Z)$, $\underline{\phi}(Z)$, $\overline{E}(Z)$ and $\underline{E}(Z)$ using the N_{MCS} samples of $\phi(Z; \mathbf{P})$ and $E(Z; \mathbf{P})$:

$$\overline{\phi}(z) = \max_{\mathbf{p} \in \mathbf{p}^{\text{MCS}}} \phi(z; \mathbf{p}) \quad (9)$$

$$\underline{\phi}(z) = \min_{\mathbf{p} \in \mathbf{p}^{\text{MCS}}} \phi(z; \mathbf{p}) \quad (10)$$

$$\overline{E}(z) = \max_{\mathbf{p} \in \mathbf{p}^{\text{MCS}}} E(z; \mathbf{p}) \quad (11)$$

$$\underline{E}(z) = \min_{\mathbf{p} \in \mathbf{p}^{\text{MCS}}} E(z; \mathbf{p}) \quad (12)$$

However, this method is too expensive or even unaffordable. One reason is that solving the ODE a large number N_{MCS} of times is time-consuming, even when the analytic solution to the ODE is available for the 1-D problem. Another reason is that there is no analytic solution to complex 2-D or 3-D problems where kinetic particle-in-cell simulations are usually employed to solve the electrostatic field through Poisson's equation.

The objective of this study is to develop a method to determine $\overline{\phi}(Z)$, $\underline{\phi}(Z)$, $\overline{E}(Z)$ and $\underline{E}(Z)$ accurately and then calculate Z^* of dust grains. It is noted here that the ultimate application of this method is not the relatively simple 1-D problem presented in this work, but more complicated or computationally expensive models such as 3-D fully kinetic particle-in-cell plasma simulations. For computationally expensive models, evaluating the model consumes the majority of computational resource, so we will use the number N_{ODE} of ODEs that we need to solve as a measure of the computational cost.

3. INTRODUCTION TO KRIGING MODEL AND AKM

Before presenting the proposed method, we briefly introduce the Kriging model [12, 13] and AKM [13-28], on which the proposed method is based.

3.1. OVERVIEW OF KRIGING MODEL

Kriging model makes regression to a black-box function (BBF) using a training sample set, or a design of experiment (DoE). The main idea of Kriging is to treat the BBF as a realization of a Gaussian random field indexed by the input variables of the BBF. The theoretical foundation of the Kriging model is exactly the Bayesian inference [28]. From the perspective of the Bayesian interface, a prior parameterized Gaussian random field is trained by the DoE and hence a posterior Gaussian random field is generated. Then the mean value function, also indexed by the input variables of the BBF, of the posterior random field is the Kriging prediction to the BBF. Besides, the variance function, also indexed by the input variables of the BBF, of the posterior random field quantifies the local prediction uncertainty or prediction error.

The randomness, or uncertainty, of the posterior random field, mainly comes from the fact that only a limited number of samples of the BBF are used to train the prior random field. In other words, only part of the information of the BBF is available, and the missing part of information leads to the epistemic uncertainty in the random field. Generally, the more training samples we use, the less epistemic uncertainty will result, and with stronger confidence will we predict the BBF.

3.2. FORMULATION OF KRIGING MODEL

A simple yet widely used prior random field is the stationary Gaussian random field given by

$$K(\mathbf{X}) = \mu + \eta(\mathbf{X}; \xi^2, \boldsymbol{\theta}) \quad (13)$$

where μ is an unknown parameter representing the mean value of the random field $K(\mathbf{X})$ and $\eta(\mathbf{X}; \xi^2, \boldsymbol{\theta})$ is a zero-mean stationary Gaussian random field indexed by \mathbf{X} , the input variables of a BBF $k(\mathbf{X})$. Both the variance parameter ξ^2 and correlation parameters $\boldsymbol{\theta}$ of $\eta(\mathbf{X}; \xi^2, \boldsymbol{\theta})$ are unknown. The parameters μ , ξ^2 and $\boldsymbol{\theta}$ fully define the prior random field $K(\mathbf{X})$. A DoE, or a training sample set, of $k(\mathbf{X})$ is used to train $K(\mathbf{X})$ and then determine those parameters.

The correlation function $C(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ of $\eta(\mathbf{X}; \xi^2, \boldsymbol{\theta})$ is given by

$$C(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \xi^2 R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}; \boldsymbol{\theta}) \quad (14)$$

where $R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}; \boldsymbol{\theta})$ is the correlation coefficient function of $\eta(\mathbf{X}; \xi^2, \boldsymbol{\theta})$ at two points $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ of \mathbf{X} . There are many models for $R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}; \boldsymbol{\theta})$. A widely used model is known as the Gaussian model, or squared exponential model, given by

$$R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}; \boldsymbol{\theta}) = \prod_{d=1}^D \exp \left[-\theta_d \left(x_d^{(i)} - x_d^{(j)} \right)^2 \right] \quad (15)$$

where D is the dimension of \mathbf{X} , $x_d^{(i)}$ is the d^{th} component of $\mathbf{x}^{(i)}$, and θ_d is the d^{th} component of $\boldsymbol{\theta}$.

For a BBF $k(\mathbf{X})$, the Kriging model predicts $k(\mathbf{x})$ as $\tilde{k}(\mathbf{x})$, which is a normal variable whose mean value and variance are $\hat{k}(\mathbf{x})$ and $\sigma^2(\mathbf{x})$, respectively. Note that $\sigma^2(\mathbf{x})$ is also termed as the mean squared error (MSE). Generally, $\hat{k}(\mathbf{x})$ is regarded as the

deterministic prediction to $k(\mathbf{x})$, since a deterministic prediction is usually needed. $\sigma^2(\mathbf{x})$ measures the prediction uncertainty, or prediction error, and therefore we can validate a Kriging model simply using $\hat{k}(\mathbf{x})$ and $\sigma^2(\mathbf{x})$ without employing traditional validation methods, such as cross-validation [30]. Because of this advantage, many algorithms have been proposed to adaptively train a Kriging model for expensive BBFs [14-27, 31-36]. When sufficient training samples have been used for training, $\sigma^2(\mathbf{x})$ converges to 0 and the normal variable $\tilde{k}(\mathbf{x})$ degenerates to a deterministic value, i.e., the exact value of $k(\mathbf{x})$.

3.3. AN EXAMPLE OF KRIGING MODEL

Figure 4 shows a 1-D example of the Kriging model. In total five initial training samples are used to train the Kriging. The vertical distance between $\hat{k}(x) \pm \sigma(x)$ graphically quantify the prediction error at x . The larger the distance, the larger the prediction error. On interval $[0, 2]$, the training samples are denser than that on $[2, 10]$. Consequently, the prediction error is smaller on $[0, 2]$ than that on $[2, 10]$. It is noted that the prediction error is not only dependent on the density of the training samples but also on the nonlinearity of the BBF. With the prediction error shown in Figure 4, it is obvious that to improve the prediction accuracy, we need to add training samples somewhere near $x = 4$ and $x = 8$. Figure 5 shows the updated Kriging model with one more training sample added at $x = 8$. The overall prediction accuracy is improved significantly.

3.4. AKM

The main idea of AKM is to adaptively add training samples to update the Kriging model iteratively until an expected accuracy is achieved. Figure 6 shows a brief flowchart

of AKM. The QoI is what we aim to calculate, such as $\overline{\phi}(Z)$ and $\underline{\phi}(Z)$. Since the QoI is calculated through the Kriging model instead of the BBF itself, there is an inevitable error caused by the Kriging model. The error metric is used to measure the error. The stopping criterion, which is based on the error metric, is used to determine when to stop adding training samples. Once the error of QoI is sufficiently small, it is reasonable to return the QoI and stop the algorithm. If the error is large in an iteration, we must add one or more training samples to update the Kriging model. How to determine new training samples is the task of the learning function. A good learning function should be robust and lead to a high convergence rate.

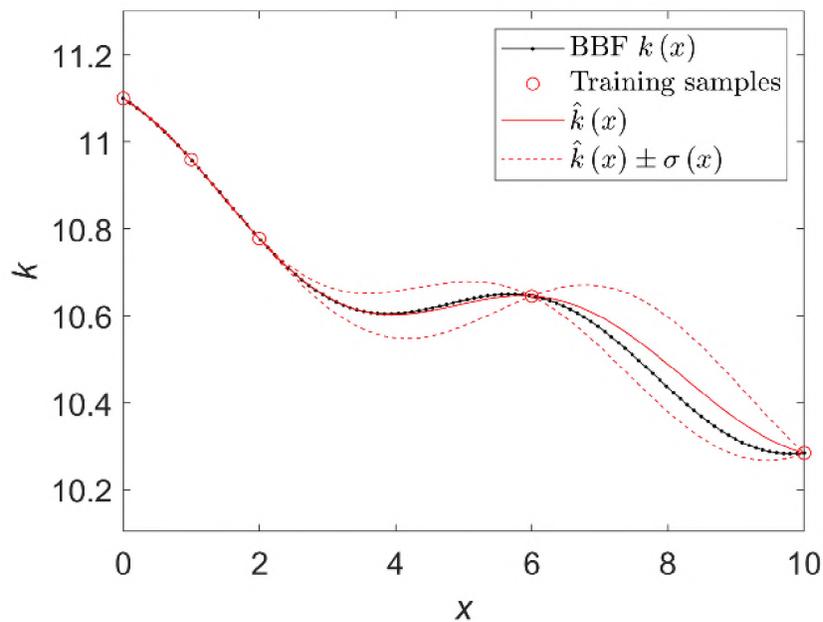


Figure 4. Original Kriging model: Prediction error is large near $x = 4$ and $x = 8$

Given a specific engineering problem, the key to employing an AKM is to make good use of all available information, such as the features of the BBF and QoI, and then design a customized or task-oriented error metric, stopping criterion, and learning function. In the UQ community, a great number of AKMs have been developed to solve various kinds of problems, such as reliability analysis [15, 17-24, 26, 31-33, 36], robustness analysis [14], sensitivity analysis [34], robust design [25, 35], and reliability-based design [16, 27], etc.

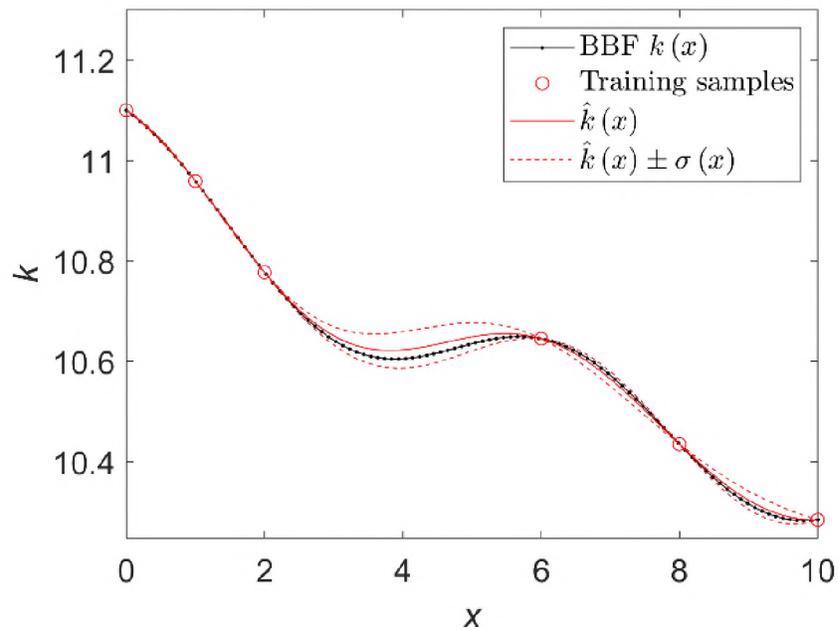


Figure 5. Updated Kriging model with one more training sample added at $x = 8$: Overall prediction accuracy is improved significantly

4. THE PROPOSED METHOD

In this section, we present detailed procedures for calculating $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$.

Similar procedures can also apply to calculate $\bar{E}(Z)$ and $\underline{E}(Z)$.

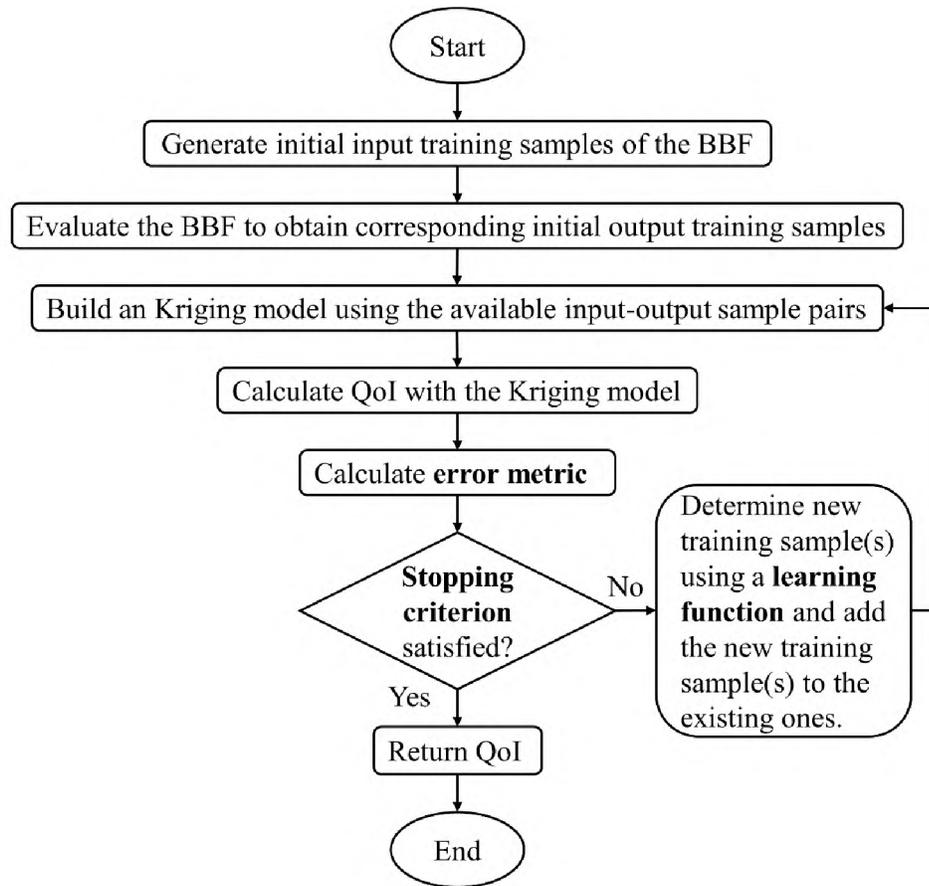


Figure 6. Brief flowchart of AKM

4.1. OVERVIEW OF THE PROPOSED METHOD

The main idea of the proposed method is to employ the framework of AKM and customize it to calculate $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$ (as well as $\bar{E}(Z)$ and $\underline{E}(Z)$). Figure 7 shows a brief flowchart of the proposed method. In Step 1, we evenly generate N_{in} initial samples of \mathbf{P} . Generally, N_{in} is much smaller than N_{MCS} . Details of this step will be given in Subsection 4.2. In Step 2, the ODE (1-D Poisson's equation) is solved N_{in} times, with the N_{in} samples of \mathbf{P} , to obtain N_{in} samples of $\phi(Z; \mathbf{P})$. In Step 3, the samples of $\phi(Z; \mathbf{P})$ are

used to build a Kriging model $\hat{\phi}(Z; \mathbf{P})$. Both Z and \mathbf{P} are treated as input variables so the dimension of $\hat{\phi}(Z; \mathbf{P})$ is $1 + 5 = 6$. In Step 4, $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$ are estimated through

$$\bar{\phi}(z) = \max_{\mathbf{p} \in \mathbf{P}^{\text{MCS}}} \hat{\phi}(z; \mathbf{p}) \quad (16)$$

$$\underline{\phi}(z) = \min_{\mathbf{p} \in \mathbf{P}^{\text{MCS}}} \hat{\phi}(z; \mathbf{p}) \quad (17)$$

In Step 5, an error metric is developed to measure the error of $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$ estimated by Eq. (16) and Eq. (17). Step 6 is about a stopping criterion. Details about Steps 5 and 6 will be given in Subsection 4.4. The learning function involved in Step 7 will be given in Subsection 4.3. The implementation of the proposed method will be given in Subsection 4.5.

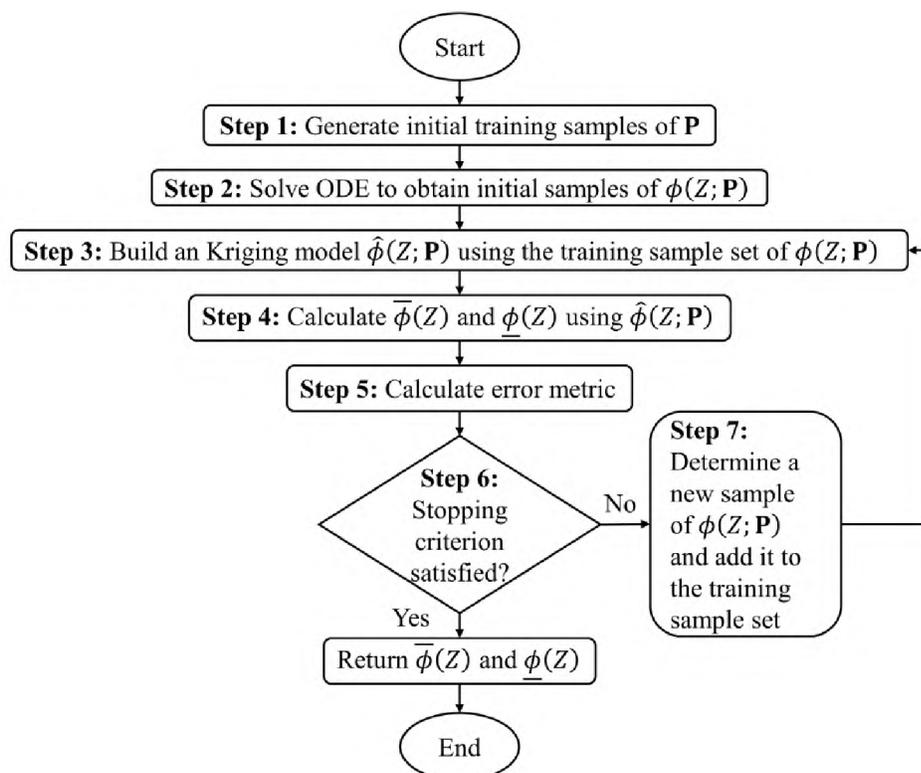


Figure 7. Brief flowchart of the proposed method

There are two significant differences between most existing AKMs and the proposed method. First, the former aims at estimating a constant value, such as the structural reliability and robustness, while the latter aims at estimating two functions, i.e., $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$. Second, when given a specific value of the input, the output of the BBFs involved in the former methods is a single value. However, in this work, with a given realization \mathbf{p} of \mathbf{P} , the output of solving the ODE is a function $\phi(Z; \mathbf{p})$. With those differences, we cannot use the existing error metrics, stopping criteria, or learning functions. Instead, we take into consideration those differences and design a new error metric, stopping criterion, and learning function to fit the problem. This is the main contribution of the proposed algorithm.

4.2. CANDIDATE SAMPLES AND INITIAL TRAINING SAMPLES

For numerical computation, we need to evenly discretize Ω into a few points. Suppose P_i , the i^{th} component of \mathbf{P} , is discretized into N_i points, then Ω will be discretized into in total $N_p = \prod_{i=1}^5 N_i$ points. For convenience, we denote the set of those points as \mathbf{p}^{MCS} . Similarly, Z is discretized into N_Z points (denoted as \mathbf{z}^{MCS}) in its range $[Z_{\min}, Z_{\max}]$. Theoretically, any $\mathbf{p} \in \Omega$ could be selected as a training sample for $\hat{\phi}(Z; \mathbf{P})$. However, we do not want any two training samples to be clustering together, because we use the exact interpolation in Kriging and clustered training samples may impact the training and the convergence rate of the proposed AKM. Therefore, we only select training samples of \mathbf{P} from \mathbf{p}^{MCS} and call \mathbf{p}^{MCS} candidate samples or candidate points.

The N_{in} initial training samples \mathbf{p}^{in} of \mathbf{P} are selected such that they are distributed in Ω as even as possible. Commonly used sampling methods include random sampling,

Latin hypercube sampling, and Hammersley sampling [37]. In this study, we employ the Hammersley sampling method because it has better uniformity properties over a multidimensional space [38]. The Hammersley sampling method firstly generates initial training samples in a 5-dimensional hypercube $[0,1]^5$ and then they are mapped into Ω to get the initial training samples of \mathbf{P} . Note that the five dimensions of the hypercube are assumed to be independent, with the assumption that all variables in \mathbf{P} are independent. Those training samples, however, are not necessarily among \mathbf{p}^{MCS} , so we need to round them to the nearest ones in \mathbf{p}^{MCS} . Since the components of \mathbf{P} do not necessarily share the same dimension unit, the distances which we use to find the nearest samples should be normalized. For example, the distance d between a sample $\mathbf{p}^{(h)}$ generated by Hammersley and a candidate sample $\mathbf{p}^{(c)}$ in \mathbf{p}^{MCS} is given by

$$d(\mathbf{p}^{(h)}, \mathbf{p}^{(c)}) = \sqrt{\sum_{i=1}^5 \left(\frac{p_i^{(h)} - p_i^{(c)}}{P_{i,\max} - P_{i,\min}} \right)^2} \quad (18)$$

where $p_i^{(h)}$ is the i^{th} component of $\mathbf{p}^{(h)}$, $p_i^{(c)}$ is the i^{th} component of $\mathbf{p}^{(c)}$, $P_{i,\max}$ is the maximal value of P_i which is the i^{th} component of \mathbf{P} , and $P_{i,\min}$ is the minimal value of P_i . Then $\mathbf{p}^{(h)}$ is rounded to $\mathbf{p}^* = \arg \min_{\mathbf{p} \in \mathbf{p}^{\text{MCS}}} d(\mathbf{p}^{(h)}, \mathbf{p}^{(c)})$. When all the initial training samples generated by Hammersley have been rounded to the nearest ones in \mathbf{p}^{MCS} , we get the initial training sample set $\mathbf{p}^{\text{in}} \subset \mathbf{p}^{\text{MCS}}$ of \mathbf{P} .

Solving the ODE N_{in} times, each with a sample in \mathbf{p}^{in} , we get N_{in} samples of $\phi(Z; \mathbf{P})$. Note that each sample of $\phi(Z; \mathbf{P})$ has N_Z points, since we discretized Z into N_Z points. Then we have $N_Z N_{\text{in}}$ input training points $\mathbf{z}^{\text{MCS}} \times \mathbf{p}^{\text{in}}$. Except the N_{in} points at Z_{\max} , we select the other $(N_Z - 1)N_{\text{in}}$ points to form the first part of the input training

sample set of $\hat{\phi}(Z; \mathbf{P})$. We denote those $(N_Z - 1)N_{\text{in}}$ input training points as \mathbf{x}^{inp1} , where superscript *inp* of \mathbf{x} represents the input, and the superscript *1* means that \mathbf{x}^{inp1} is only the first part of the entire input training sample set. The other part \mathbf{x}^{inp2} is given below.

Since for any $\mathbf{p} \in \mathbf{p}^{\text{MCS}}$, it is known that $\phi(Z_{\text{max}}; \mathbf{p}) \equiv 0$ (Figure 1), theoretically we also need to add all the N_P points $Z_{\text{max}} \times \mathbf{p}^{\text{MCS}}$ as input training samples so that we make good use of all known information. However, it is not practical to do so. For example, if $N_i = 10, i = 1, 2, \dots, 5$, we need to add $N_P = 10^5$ points as input training samples. So many training samples will make $\hat{\phi}(Z; \mathbf{P})$ complex, expensive, and not accurate, losing its expected properties. To balance the need to add them and the drawback of adding all of them, we add part of them. Specifically, we evenly generate N'_P samples \mathbf{p}' of \mathbf{P} using procedures similar to what we used to generate \mathbf{p}^{in} . Then \mathbf{x}^{inp2} is given by

$$\mathbf{x}^{\text{inp2}} = \{(Z_{\text{max}}, \mathbf{p}) | \mathbf{p} \in \mathbf{p}'\} \quad (19)$$

The input training sample set $\mathbf{x}^{\text{inp}} = \mathbf{x}^{\text{inp1}} \cup \mathbf{x}^{\text{inp2}}$. Denote the corresponding electric potential ϕ at \mathbf{x}^{inp} as ϕ^{out} . The input-output training sample pairs $(\mathbf{x}^{\text{inp}}, \phi^{\text{out}})$ are used to build the initial $\hat{\phi}(Z; \mathbf{P})$. More training samples will be added to update $\hat{\phi}(Z; \mathbf{P})$ later.

4.3. LEARNING FUNCTION

Generally, the initial Kriging model is not accurate enough to get $\overline{\phi}(Z)$ or $\underline{\phi}(Z)$ accurately through Eq. (5) and Eq. (6). To improve the accuracy of $\hat{\phi}(Z; \mathbf{P})$ and hence of $\overline{\phi}(Z)$ and $\underline{\phi}(Z)$, we need to add training samples of $\phi(Z; \mathbf{P})$ to refine $\hat{\phi}(Z; \mathbf{P})$. A learning function is used to determine which sample of \mathbf{P} , and hence of $\phi(Z; \mathbf{P})$, should be added.

In our previous work [3], we used the learning function given by

$$\mathbf{p}^{(\text{next})} = \arg \max_{\mathbf{p} \in \mathbf{p}^{\text{MCS}}} \sum_{z \in Z^{\text{MCS}}} \left| \frac{\sigma(z; \mathbf{p})}{\hat{\phi}(z; \mathbf{p})} \right| \quad (20)$$

where $\mathbf{p}^{(\text{next})}$ is the next to-be-added sample of \mathbf{P} , $\hat{\phi}(z; \mathbf{p})$ is the predicted value of $\phi(z; \mathbf{p})$ by the Kriging model $\hat{\phi}(Z; \mathbf{P})$, and $\sigma(z; \mathbf{p})$ is the standard deviation of the prediction. Both $\hat{\phi}(z; \mathbf{p})$ and $\sigma(z; \mathbf{p})$ are calculated by the Kriging toolbox. $\frac{\sigma(z^{(j)}; \mathbf{p})}{\hat{\phi}(z^{(j)}; \mathbf{p})}$ is the deviation coefficient of the prediction at $(z; \mathbf{p})$, and thus the learning function in Eq. (20) determines the training sample $\mathbf{p}^{(\text{next})}$ at which the summation of the absolute deviation coefficients of the predictions along Z coordinate is maximal. The summation $\sum_{z \in Z^{\text{MCS}}} \left| \frac{\sigma(z; \mathbf{p})}{\hat{\phi}(z; \mathbf{p})} \right|$ measures the overall prediction error at \mathbf{p} . Adding a sample of $\phi(Z; \mathbf{P})$ at \mathbf{p} to update $\hat{\phi}(Z; \mathbf{P})$ will let $\sum_{z \in Z^{\text{MCS}}} \left| \frac{\sigma(z; \mathbf{p})}{\hat{\phi}(z; \mathbf{p})} \right|$ become zero, and therefore adding a sample of $\phi(Z; \mathbf{P})$ at $\mathbf{p}^{(\text{next})}$ to update $\hat{\phi}(Z; \mathbf{P})$ will decrease the overall prediction error of $\phi(Z; \mathbf{P})$ by the largest extent. This is the basic mechanism of the learning function in Eq. (20).

However, we do not necessarily need $\phi(Z; \mathbf{P})$ to be overall accurate. Since the objective is to estimate $\overline{\phi}(Z)$ and $\underline{\phi}(Z)$ accurately and efficiently, we only need $\hat{\phi}(Z; \mathbf{P})$ to be partially or locally accurate enough so that it can help estimate $\overline{\phi}(Z)$ and $\underline{\phi}(Z)$ accurately. With this idea, we can further improve the efficiency of updating $\hat{\phi}(Z; \mathbf{P})$ by adding training samples more skillfully.

A widely used learning function in an AKM that aims at calculating extreme values is the expected improvement function [27]. The expected improvement function $\bar{\xi}(z, \mathbf{p})$ of $\bar{\phi}(z)$ is given by

$$\begin{aligned} \bar{\xi}(z, \mathbf{p}) = & \left(\hat{\phi}(z; \mathbf{p}) - \bar{\phi}(z) \right) \Phi \left(\frac{\hat{\phi}(z; \mathbf{p}) - \bar{\phi}(z)}{\sigma(z; \mathbf{p})} \right) \\ & + \sigma(z; \mathbf{p}) \varphi \left(\frac{\hat{\phi}(z; \mathbf{p}) - \bar{\phi}(z)}{\sigma(z; \mathbf{p})} \right) \end{aligned} \quad (21)$$

where $\Phi(\cdot)$ and $\varphi(\cdot)$ are the cumulative distribution function and probability density function of the standard Gaussian variable, respectively. A simple explanation of the expected improvement function $\bar{\xi}(z, \mathbf{p})$ is that if we added a training point at (z, \mathbf{p}) , we could expect to improve current $\bar{\phi}(z)$ to $\bar{\phi}(z) + \bar{\xi}(z, \mathbf{p})$, with an improvement rate of $\bar{\xi}(z, \mathbf{p})/\bar{\phi}(z)$. If the objective is to estimate $\bar{\phi}(z)$, which is a maximal value, instead of $\bar{\phi}(Z)$, which is an entire function, we can determine the next training sample $\mathbf{p}^{(\text{next})}$ of \mathbf{P} using the learning function given by

$$\mathbf{p}^{(\text{next})} = \arg \max_{\mathbf{p} \in \mathbf{p}^{\text{MCS}}} |\bar{\xi}(z, \mathbf{p})/\bar{\phi}(z)| \quad (22)$$

However, since the objective is to determine the entire function $\bar{\phi}(Z)$ and one ODE solution has N_Z training points, we must have a learning function which aims at improving the calculation accuracy of the entire function $\bar{\phi}(Z)$. Therefore, we propose a learning function given by

$$\mathbf{p}^{(\text{next})} = \arg \max_{\mathbf{p} \in \mathbf{p}^{\text{MCS}}} \sum_{z \in Z^{\text{MCS}}} |\bar{\xi}(z, \mathbf{p})/\bar{\phi}(z)| \quad (23)$$

where we sum up the absolute values of the improvement rate.

This learning function means that if we added a training sample $\phi(Z; \mathbf{p}^{(\text{next})})$, which has N_Z points, to update $\hat{\phi}(Z; \mathbf{P})$, we could expect to get the best improvement of $\bar{\phi}(Z)$. Similarly, the expected improvement function $\underline{\xi}(z, \mathbf{p})$ of $\underline{\phi}(z)$ is given by

$$\begin{aligned} \underline{\xi}(z, \mathbf{p}) = & \left(\underline{\phi}(z) - \hat{\phi}(z; \mathbf{p}) \right) \Phi \left(\frac{\phi(z) - \hat{\phi}(z; \mathbf{p})}{\sigma(z; \mathbf{p})} \right) \\ & + \sigma(z; \mathbf{p}) \varphi \left(\frac{\phi(z) - \hat{\phi}(z; \mathbf{p})}{\sigma(z; \mathbf{p})} \right) \end{aligned} \quad (24)$$

To estimate $\underline{\phi}(Z)$, we also propose a learning function given by

$$\mathbf{p}^{(\text{next})} = \arg \max_{\mathbf{p} \in \mathbf{p}^{\text{MCS}}} \sum_{z \in Z^{\text{MCS}}} \left| \underline{\xi}(z, \mathbf{p}) / \underline{\phi}(z) \right| \quad (25)$$

To estimate both $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$ simultaneously, we combine Eq. (24) and Eq. (25) to propose a learning function given by

$$\mathbf{p}^{(\text{next})} = \arg \max \left\{ \max_{\mathbf{p} \in \mathbf{p}^{\text{MCS}}} \sum_{z \in Z^{\text{MCS}}} \left| \frac{\bar{\xi}(z, \mathbf{p})}{\bar{\phi}(z)} \right|, \max_{\mathbf{p} \in \mathbf{p}^{\text{MCS}}} \sum_{z \in Z^{\text{MCS}}} \left| \frac{\underline{\xi}(z, \mathbf{p})}{\underline{\phi}(z)} \right| \right\} \quad (26)$$

Once $\mathbf{p}^{(\text{next})}$ has been determined, we solve the ODE to numerically get a function $\phi(Z; \mathbf{p}^{(\text{next})})$, from which we get $(N_Z - 1)$ points (the remaining one at Z_{max} , where $\phi \equiv 0$, is excluded) and add them into $(\mathbf{x}^{\text{inp}}, \boldsymbol{\phi}^{\text{out}})$ to enrich the training samples.

4.4. ERROR METRIC AND STOPPING CRITERION

Since Eq. (16) and Eq. (17) cannot obtain absolutely accurate $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$ due to the prediction error of $\hat{\phi}(Z; \mathbf{P})$, we need an error metric to measure the error of currently estimated $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$. Since $\left| \frac{\bar{\xi}(z, \mathbf{p})}{\bar{\phi}(z)} \right|$ measures the absolute expected improvement rate

of $\bar{\phi}(z)$, if $\left| \frac{\bar{\xi}(z, \mathbf{p})}{\bar{\phi}(z)} \right|$ is small for any $z \in \mathbf{z}^{\text{MCS}}$ and $\mathbf{p} \in \mathbf{p}^{\text{MCS}}$, $\bar{\phi}(Z)$ is expected to be sufficiently accurate. Therefore, we propose to use $\max_{z \in \mathbf{z}^{\text{MCS}}, \mathbf{p} \in \mathbf{p}^{\text{MCS}}} \left| \frac{\bar{\xi}(z, \mathbf{p})}{\bar{\phi}(z)} \right|$ to quantify the error of $\bar{\phi}(Z)$. Similarly, $\max_{z \in \mathbf{z}^{\text{MCS}}, \mathbf{p} \in \mathbf{p}^{\text{MCS}}} \left| \frac{\xi(z, \mathbf{p})}{\underline{\phi}(z)} \right|$ is used to quantify the error of $\underline{\phi}(Z)$. Combining both, we have the error metric Γ , which measures the error of both $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$, given by

$$\Gamma = \max \left\{ \max_{z \in \mathbf{z}^{\text{MCS}}, \mathbf{p} \in \mathbf{p}^{\text{MCS}}} \left| \frac{\bar{\xi}(z, \mathbf{p})}{\bar{\phi}(z)} \right|, \max_{z \in \mathbf{z}^{\text{MCS}}, \mathbf{p} \in \mathbf{p}^{\text{MCS}}} \left| \frac{\xi(z, \mathbf{p})}{\underline{\phi}(z)} \right| \right\} \quad (27)$$

Once Γ is small enough, the estimated $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$ are expected to be sufficiently accurate. Therefore, the stopping criterion shown in Figure 7 is defined as

$$\Gamma < \gamma \quad (28)$$

where γ is a threshold that controls the efficiency and accuracy of the proposed AKM. Generally speaking, a smaller γ will lead to higher accuracy but lower efficiency.

4.5. IMPLEMENTATION

As shown in Figure 1, $\phi(Z; \mathbf{P})$ approaches zero when Z takes large value. As a result, $\bar{\phi}(z)$ and $\underline{\phi}(z)$ in Eq. (26) and Eq. (27) are likely to take very small values close to zero. It leads to the singularity of the calculation of Eq. (26) and Eq. (27), harming the robustness of the proposed algorithm. To solve this issue, we translate all training samples of $\phi(Z; \mathbf{P})$ simply by adding a negative constant ϵ . This way, the translated $\phi(Z; \mathbf{P})$ will never approach zero and the singularity issue is evitable. Trained by the translated samples

of $\phi(Z; \mathbf{P})$, the Kriging model $\hat{\phi}(Z; \mathbf{P})$ will also lead to the translation of $\overline{\phi}(Z)$ and $\underline{\phi}(Z)$.

We can translate $\overline{\phi}(Z)$ and $\underline{\phi}(Z)$ back simply by subtracting ϵ from them. Note that there is no rigorous theory to quantify how ϵ affects the properties of the proposed AKM. We suggest determining ϵ using

$$\epsilon = \text{mean}\{\phi(0; \mathbf{p}) | \mathbf{p} \in \mathbf{p}^{\text{in}}\} \quad (29)$$

where $\text{mean}(\cdot)$ represents mean value. Based on all the procedures given above, we generate the pseudo codes of the proposed AKM given in Algorithm 1.

Algorithm 1. Pseudo codes of the proposed method

Row	Pseudo codes
1	Evenly discretize Ω into N_p points \mathbf{p}^{MCS} .
2	Evenly discretize interval $[Z_{\min}, Z_{\max}]$ into N_Z points \mathbf{z}^{MCS} .
3	Generate N_{in} samples \mathbf{p}^{in} of \mathbf{P} with procedures given in Subsection 4.2.
4	Solve ODE N_{in} times to get N_{in} samples $\phi(Z; \mathbf{p})$, $\mathbf{p} \in \mathbf{p}^{\text{in}}$ of $\phi(Z; \mathbf{P})$; Calculate ϵ with Eq. (29); $N_{\text{ODE}} = N_{\text{in}}$.
5	Determine $(\mathbf{x}^{\text{inp}}, \boldsymbol{\phi}^{\text{out}})$ with procedures given in Subsection 4.2; $\boldsymbol{\phi}^{\text{out}} = \boldsymbol{\phi}^{\text{out}} + \epsilon$.
6	WHILE TRUE DO
7	Build Kriging model $\phi(Z; \mathbf{P})$ using $(\mathbf{x}^{\text{inp}}, \boldsymbol{\phi}^{\text{out}})$.
8	Calculate $\overline{\phi}(Z)$ and $\underline{\phi}(Z)$ with Eq. (16) and Eq. (17); $\overline{\phi}(Z) = \overline{\phi}(Z) - \epsilon$; $\underline{\phi}(Z) = \underline{\phi}(Z) - \epsilon$.
9	Calculate Γ with Eq. (27)
10	IF $(\Gamma \geq \gamma)$ DO
11	Solve Eq. (20) for $\mathbf{p}^{(\text{next})}$; $N_{\text{ODE}} = N_{\text{ODE}} + 1$. Solve ODE to get a new sample $\phi(Z; \mathbf{p}^{(\text{next})})$; $\phi(Z; \mathbf{p}^{(\text{next})}) =$
12	$\phi(Z; \mathbf{p}^{(\text{next})}) + \epsilon$; All points of $\phi(Z; \mathbf{p}^{(\text{next})})$ excluding the one at Z_{\max} are added into $(\mathbf{x}^{\text{inp}}, \boldsymbol{\phi}^{\text{out}})$.
13	ELSE
14	BREAK WHILE
15	END IF
16	END WHILE
17	RETURN $\overline{\phi}(Z)$, $\underline{\phi}(Z)$, and N_{ODE} .

4.6. VALIDATION DISCUSSION

Theoretically, it is vital to validate the Kriging model to make sure that it has been trained accurately. An explicit validation, however, is not involved in the proposed AKM. There are two main reasons. First, the adaptive training focuses on the accuracy of QoI instead of the accuracy of the Kriging model. Once there is an indication that the accuracy of QoI in current training iteration is sufficient, i.e., the stopping criterion in Eq. (27) is satisfied, the algorithm stops adding more training samples, no matter the Kriging model is globally accurate or not. As a result, when the algorithm has converged, the Kriging model is likely accurate only in some subdomains but not accurate in other domains. Therefore, it is not suitable to do explicit cross-validation. Second, the error metric Γ can measure the accuracy of QoI, and therefore we do validation implicitly. As long as the accuracy of QoI is sufficient, it does not matter if the Kriging model is or not accurate in the entire domain.

5. RESULTS

In this section, we illustrate the proposed AKM. MCS is used to solve the same problems with brute force. Results by MCS are treated as standard to verify the proposed AKM. We build the Kriging model and calculate the Kriging predictions using the DACE toolbox [39]. The anisotropic Gaussian kernel is used.

5.1. SHEATH PROFILE

We consider the Type C of the 1-D photoelectron sheath problem discussed in Section 2. The sun elevation angle is given as 9 degrees. The maximal and minimal values of $\mathbf{P} = (v_d, T_e, T_p, n_{i,\infty}, n_p)$ are given in Table 1. We use both MCS and the proposed AKM to estimate $\overline{\phi}(Z)$ and $\underline{\phi}(Z)$. The values of all involved parameters are given in Table 2.

Table 1. Variables of uncertainty

Variables	$v_d(\text{m/s})$	$T_e(\text{eV})$	$T_p(\text{eV})$	$n_{i,\infty}(\text{cm}^{-3})$	$n_p(\text{cm}^{-3})$
Minimum	421,200	10.8	1.8	7.83	57.6
Maximum	414,800	13.2	2.2	9.57	70.4

Table 2. Parameter values

Parameters	$N_1 \sim N_5$	N_p	N_{in}	N'_p	N_Z	γ
Values	5	5^5	5	100	50	0.01

Table 3. Samples generated by the Hammersley sampling method

Sample number	Dimension 1	Dimension 2	Dimension 3	Dimension 4	Dimension 5
1	0	0.5000	0.3333	0.2000	0.1429
2	0.2	0.2500	0.6667	0.4000	0.2857
3	0.4	0.7500	0.1111	0.6000	0.4286
4	0.6	0.1250	0.4444	0.8000	0.5714
5	0.8	0.6250	0.7778	0.0400	0.7143

The domain Ω of \mathbf{P} is discretized into $N_p = 5^5$ points, which are assembled into \mathbf{p}^{MCS} . The $N_{\text{in}} = 5$ samples in hypercube space $[0,1]^5$, generated by the Hammersley sampling method, are given in Table 3. Then the 5 samples are mapped into Ω , as given in

Table 4. Rounding the 5 samples in Ω to the nearest ones in \mathbf{p}^{MCS} , we get the initial samples \mathbf{p}^{in} of \mathbf{P} , as given in Table 5. Solving the ODE five times, each with a sample in \mathbf{p}^{in} , we get five samples of $\phi(Z; \mathbf{P})$ as shown in Figure 8.

Table 4. Samples mapped into Ω

Sample number	$v_d(\text{m/s})$	$T_e(\text{eV})$	$T_p(\text{eV})$	$n_{i,\infty}(\text{cm}^{-3})$	$n_p(\text{cm}^{-3})$
1	421,200	12.0000	1.9333	8.1780	59.4286
2	439,920	11.4000	2.0667	8.5260	61.2571
3	458,640	12.6000	1.8444	8.8740	63.0857
4	477,360	11.1000	1.9778	9.2220	64.9143
5	496,080	12.3000	2.1111	7.8996	66.7429

Table 5. Initial samples of \mathbf{P}

Sample number	$v_d(\text{m/s})$	$T_e(\text{eV})$	$T_p(\text{eV})$	$n_{i,\infty}(\text{cm}^{-3})$	$n_p(\text{cm}^{-3})$
1	421,200	12.0000	1.9000	8.2650	60.8000
2	444,600	11.4000	2.1000	8.7000	60.8000
3	468,000	12.6000	1.8000	8.7000	64.0000
4	468,000	11.4000	2.0000	9.1350	64.0000
5	491,400	12.0000	2.1000	7.8300	67.2000

Each sample of $\phi(Z; \mathbf{P})$ contains $N_Z = 50$ numerical points. Excluding the five points at Z_{max} , we have $N_Z N_{\text{in}} - 5 = 245$ training points in $(\mathbf{x}^{\text{inp1}}, \boldsymbol{\phi}^{\text{out1}})$. With the Hammersley sampling method, we generate $N'_p = 100$ samples of \mathbf{P} and hence 100 training points in $(\mathbf{x}^{\text{inp2}}, \boldsymbol{\phi}^{\text{out2}})$. Note that all points in $(\mathbf{x}^{\text{inp2}}, \boldsymbol{\phi}^{\text{out2}})$ have $Z = Z_{\text{max}}$ and $\phi = 0$. Combining $(\mathbf{x}^{\text{inp1}}, \boldsymbol{\phi}^{\text{out1}})$ and $(\mathbf{x}^{\text{inp2}}, \boldsymbol{\phi}^{\text{out2}})$, we have 345 training points in $(\mathbf{x}^{\text{inp}}, \boldsymbol{\phi}^{\text{out}})$. To do the translation mentioned in Subsection 4.5, we update $\boldsymbol{\phi}^{\text{out}}$ simply by $\boldsymbol{\phi}^{\text{out}} = \boldsymbol{\phi}^{\text{out}} + \epsilon$, where $\epsilon = -6.97 \text{ V}$ is obtained with Eq. (29). With the updated

$(\mathbf{x}^{\text{inp}}, \boldsymbol{\phi}^{\text{out}})$, we build an initial Kriging model and then estimate $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$ through Eq. (16) and Eq. (17). Finally, we translate $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$ back by $\bar{\phi}(Z) = \bar{\phi}(Z) - \epsilon$ and $\underline{\phi}(Z) = \underline{\phi}(Z) - \epsilon$. Figure 9 shows the $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$ estimated by both MCS and the proposed AKM (with the initial Kriging model). It shows that the initial Kriging model is not able to predict $\bar{\phi}(Z)$ or $\underline{\phi}(Z)$ with sufficient accuracy.

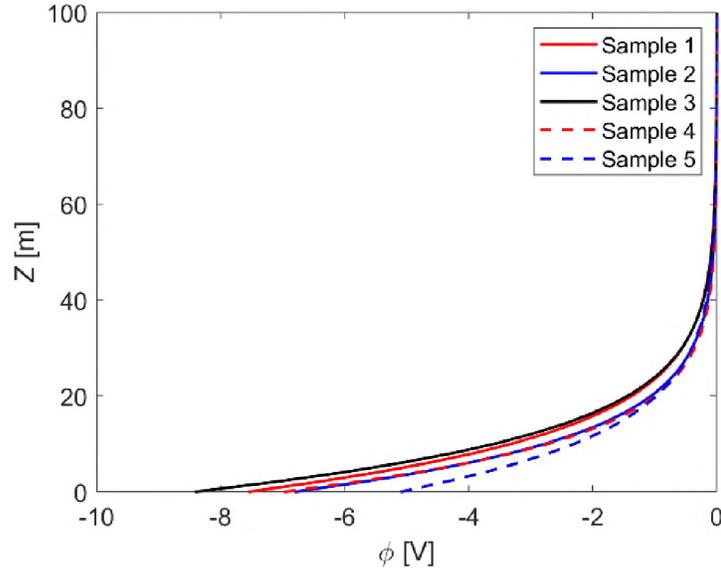


Figure 8. Initial samples of $\phi(Z; \mathbf{P})$

To improve accuracy, the proposed method indicates adding a sample at $\mathbf{p}^{(\text{next})} = (514800, 13.2, 2.2, 9.57, 57.6)$. With the $\mathbf{p}^{(\text{next})}$, we solve the ODE and get a new sample of $\phi(Z; \mathbf{P})$. This sample contains $N_Z = 50$ numerical points. We translate all the numerical points and add them, excluding the one at Z_{max} , to update $(\mathbf{x}^{\text{inp}}, \boldsymbol{\phi}^{\text{out}})$. The reason why we abandon the point at Z_{max} is that there are already enough points at Z_{max} in

$(\mathbf{x}^{\text{inp2}}, \boldsymbol{\phi}^{\text{out2}})$. With the updated $(\mathbf{x}^{\text{inp}}, \boldsymbol{\phi}^{\text{out}})$, we build a new $\hat{\phi}(Z; \mathbf{P})$. With the new $\hat{\phi}(Z; \mathbf{P})$ another $\mathbf{p}^{(\text{next})}$ is indicated. With similar procedures, more and more samples of $\phi(Z; \mathbf{P})$ are added to refine $\hat{\phi}(Z; \mathbf{P})$ until the stopping criterion given in Eq. (28) is satisfied.

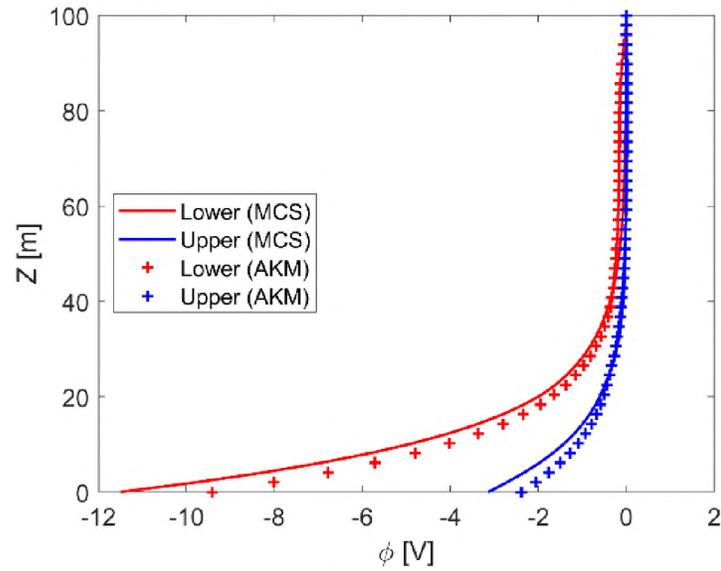


Figure 9. Results by initial Kriging model: Predicted electric potential bounds are not accurate

The final estimation of $\overline{\phi}(Z)$ and $\underline{\phi}(Z)$ is shown in Figure 10. It shows that the proposed AKM can estimate $\overline{\phi}(Z)$ and $\underline{\phi}(Z)$ very accurately. 19 more samples of $\phi(Z; \mathbf{P})$ have been added to refine $\hat{\phi}(Z; \mathbf{P})$, and therefore in total $N_{\text{ODE}} = N_{\text{in}} + 16 = 21$ ODE solutions are needed. Compared to $N_p = 3,125$ ODE solutions needed in MCS, the proposed method is very efficient.

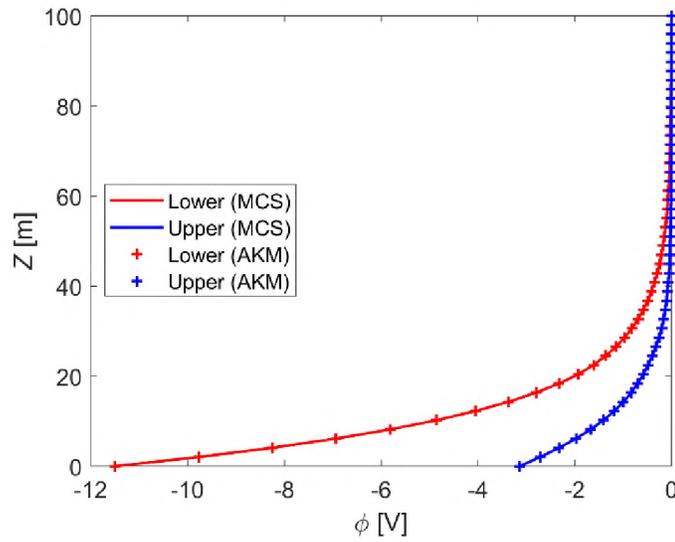


Figure 10. Final result: Predicted electric potential bounds are accurate

5.2. DUST LEVITATION

In this example, we still consider the same 1-D photoelectron sheath problem in Subsection 5.1, but the objective is to estimate $\bar{E}(Z)$ and $\underline{E}(Z)$ and then calculate the dust levitation height. The values of all involved parameters are given in Table 6.

Table 6. Parameter values

Parameters	$N_1 \sim N_5$	N_p	N_{in}	N'_p	N_Z	γ
Values	5	5^5	5	100	50	0.01

The procedures used to estimate $\bar{E}(Z)$ and $\underline{E}(Z)$ are almost the same as that used to estimate $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$. The only difference is that the samples of $E(Z; \mathbf{P})$ instead of $\phi(Z; \mathbf{P})$ are used. The final estimation of $\bar{E}(Z)$ and $\underline{E}(Z)$ is shown in Figure 11. It shows that the proposed AKM method is very accurate. As for efficiency, the proposed method

needs only $N_{\text{ODE}} = N_{\text{in}} + 18 = 23$ ODE solutions. Compared to $N_p = 3,125$ ODE solutions needed in MCS, the proposed method is very efficient.

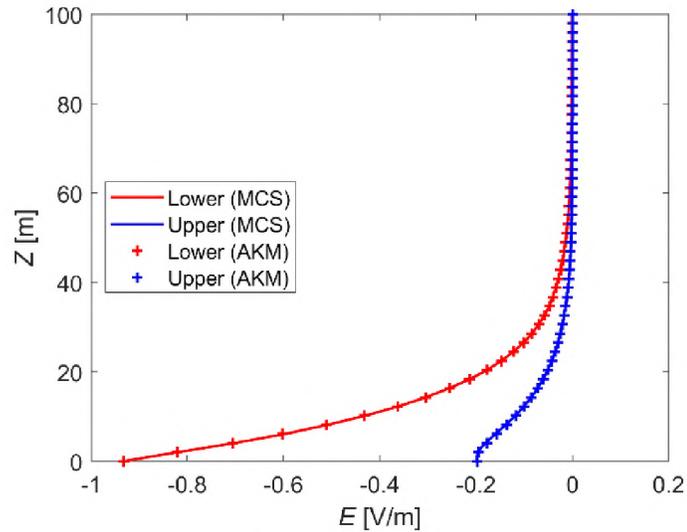


Figure 11. Final result: Predicted electric field bounds are accurate

When the upper and lower bounds of the electric field have been determined, we can use them to determine the levitation heights of the dust grains. Assuming there are two types of dust grains, A and B, in the electric field. The relevant parameters of the grains are given in Table 7, where $e = 1.062 \times 10^{-19}\text{C}$ is the electric charge of an electron. The dust levitation heights are shown in Figure 12 and given in Table 8. Due to the uncertainty of \mathbf{P} , the levitation heights of both A and B are also uncertain. The levitation height of A may be any value in the interval $[0 \text{ m}, 9.33 \text{ m}]$, which is estimated by the proposed method. The interval determined by MCS is $[0 \text{ m}, 9.26 \text{ m}]$. It shows that the proposed method can

estimate the levitation height of Grain A with sufficient accuracy. A similar conclusion applies to the levitation height of Grain B.

Given any dust grain with known w value, we can easily determine its levitation height interval using the method shown in Figure 12. This will help to evaluate the risk or damage caused by the levitated dust grains for lunar exploration missions.

Table 7. Parameters of Grains A and B

Grains	r (μm)	m (g)	q/e	w (V/m)
A	0.5	1.5268×10^{-12}	50,000	-0.4658
B	0.3	3.2979×10^{-13}	45,000	-0.1118

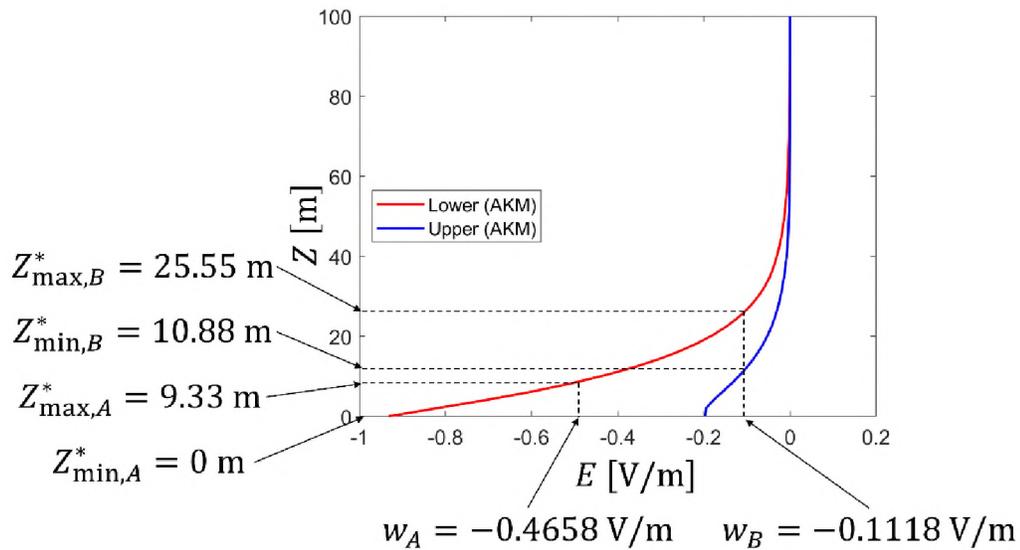


Figure 12. Dust levitation heights: The electric field bounds determines the dust levitation heights

Table 8. Dust levitation heights: The proposed AKM obtained accurate levitation heights

Grains		AKM	MCS	Relative error (%)
A	Z_{\min}^* (m)	0.00	0.00	0.0
	Z_{\max}^* (m)	9.33	9.26	0.8
B	Z_{\min}^* (m)	10.88	11.00	-1.1
	Z_{\max}^* (m)	25.55	25.55	0.0

6. CONCLUSIONS

We presented an adaptive Kriging based method to perform UQ analysis of the 1-D photoelectron sheath and dust levitation on the lunar surface. A recently derived 1-D photoelectron sheath model was used as the high-fidelity physics-based model and the black-box function. The adaptive Kriging method, with a task-oriented learning function and stopping criterion, was utilized to improve the efficiency in calculating the upper and lower bounds of electric potential as well as dust levitation height, given the intervals of model parameters. Experiment analysis shows that the proposed AKM method is both accurate and efficient. Current and ongoing efforts are focused on building an adaptive Kriging model for 2-D and 3-D kinetic particle simulations of the lunar plasma/dust environment and perform UQ analysis.

ACKNOWLEDGMENT

We would like to acknowledge the support from NASA-Missouri Space Grant Consortium through NASA-EPSCoR Missouri and NSF-CMMI #1923799.

REFERENCES

- [1] Der Kiureghian, A., and Ditlevsen, O., 2009, "Aleatory or epistemic? Does it matter?," *Structural safety*, 31(2), pp. 105-112.
- [2] Williams, C. K., and Rasmussen, C. E., 2006, *Gaussian processes for machine learning*, MIT Press Cambridge, MA.
- [3] Zhao, J., Wei, X., Hu, Z., He, X., Han, D., Hu, Z., and Du, X., "Photoelectron Sheath near the Lunar Surface: Fully Kinetic Modeling and Uncertainty Quantification Analysis," *Proc. AIAA Scitech 2020 Forum*, p. 1548.
- [4] Zhao, J., Wei, X., Du, X., He, X., and Han, D., 2020, "Photoelectron Sheath on the Lunar Surface: Analytic Solutions and Fully-Kinetic Particle-in-Cell Simulations for Maxwellian and Kappa Distributions," submitted.
- [5] Nitter, T., Havnes, O., and Melandsø, F., 1998, "Levitation and dynamics of charged dust in the photoelectron sheath above surfaces in space," *Journal of Geophysical Research: Space Physics*, 103(A4), pp. 6605-6620.
- [6] Fu, J. H., 1971, "Surface potential of a photoemitting plate," *Journal of Geophysical Research*, 76(10), pp. 2506-2509.
- [7] Poppe, A., and Horányi, M., 2010, "Simulations of the photoelectron sheath and dust levitation on the lunar surface," *Journal of Geophysical Research: Space Physics*, 115(A8).
- [8] Wang, J., He, X., and Cao, Y., 2008, "Modeling electrostatic levitation of dust particles on lunar surface," *IEEE transactions on plasma science*, 36(5), pp. 2459-2466.
- [9] Hirt, C., and Featherstone, W., 2012, "A 1.5 km-resolution gravity field model of the Moon," *Earth and Planetary Science Letters*, 329, pp. 22-30.

- [10] Snelling, A. A., and Rush, D. E., 1993, "Moon dust and the age of the solar system," *Creation Ex Nihilo Technical Journal*, 7(1), pp. 2-42.
- [11] Mooney, C. Z., 1997, *Monte carlo simulation*, Sage Publications, Thousand Oaks.
- [12] Rasmussen, C. E., 2004, "Gaussian processes in machine learning," *Advanced lectures on machine learning*, Springer, pp. 63-71.
- [13] Jeong, S., Murayama, M., and Yamamoto, K., 2005, "Efficient optimization design method using kriging model," *Journal of aircraft*, 42(2), pp. 413-420.
- [14] Wei, X., and Du, X., 2020, "Robustness Metric for Robust Design Optimization Under Time-and Space-Dependent Uncertainty Through Metamodeling," *Journal of Mechanical Design*, 142(3).
- [15] Shi, Y., Lu, Z., Xu, L., and Chen, S., 2019, "An adaptive multiple-Kriging-surrogate method for time-dependent reliability analysis," *Applied Mathematical Modelling*, 70, pp. 545-571.
- [16] Moustapha, M., and Sudret, B., 2019, "Surrogate-assisted reliability-based design optimization: a survey and a unified modular framework," *Structural and Multidisciplinary Optimization*, pp. 1-20.
- [17] Yun, W., Zhenzhou, L., Zhou, Y., and Jiang, X., 2018, AK-SYSi: an improved adaptive Kriging model for system reliability analysis with multiple failure modes by a refined U learning function.
- [18] Sun, Z., Wang, J., Li, R., and Tong, C., 2017, "LIF: A new Kriging based learning function and its application to structural reliability analysis," *Reliability Engineering & System Safety*, 157, pp. 152-165.
- [19] Ma, J., Ren, Z., Zhao, G., Zhang, Y., and Koh, C.-S., 2017, A New Reliability Analysis Method Combining Adaptive Kriging With Weight Index Monte Carlo Simulation.
- [20] Zhu, Z., and Du, X., "A System Reliability Method With Dependent Kriging Predictions," *Proc. ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, p. V02BT03A043.
- [21] Zhu, Z., and Du, X., 2016, "Reliability analysis with Monte Carlo simulation and dependent Kriging predictions," *Journal of Mechanical Design*, 138(12), p. 121403.
- [22] Hu, Z., and Mahadevan, S., 2016, "A single-loop kriging surrogate modeling for time-dependent reliability analysis," *Journal of Mechanical Design*, 138(6), p. 061406.

- [23] Zhang, L., Lu, Z., and Wang, P., 2015, "Efficient structural reliability analysis method based on advanced Kriging model," *Applied Mathematical Modelling*, 39(2), pp. 781-793.
- [24] Lv, Z., Lu, Z., and Wang, P., 2015, "A new learning function for Kriging and its applications to solve reliability problems in engineering," *Computers & Mathematics with Applications*, 70(5), pp. 1182-1197.
- [25] Cheng, J., Liu, Z., Wu, Z., Li, X., and Tan, J., 2015, "Robust optimization of structural dynamic characteristics based on adaptive Kriging model and CNSGA," *Structural and Multidisciplinary Optimization*, 51(2), pp. 423-437.
- [26] Echard, B., Gayton, N., and Lemaire, M., 2011, "AK-MCS: an active learning reliability method combining Kriging and Monte Carlo simulation," *Structural Safety*, 33(2), pp. 145-154.
- [27] Dubourg, V., Sudret, B., and Bourinet, J.-M., 2011, "Reliability-based design optimization using kriging surrogates and subset simulation," *Structural and Multidisciplinary Optimization*, 44(5), pp. 673-690.
- [28] Jones, D. R., Schonlau, M., and Welch, W. J., 1998, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, 13(4), pp. 455-492.
- [29] Box, G. E., and Tiao, G. C., 2011, *Bayesian inference in statistical analysis*, John Wiley & Sons.
- [30] Browne, M. W., 2000, "Cross-validation methods," *Journal of Mathematical Psychology*, 44(1), pp. 108-132.
- [31] Huang, X., Chen, J., and Zhu, H., 2016, "Assessing small failure probabilities by AK-SS: An active learning method combining Kriging and Subset Simulation," *Structural Safety*, 59, pp. 86-95.
- [32] Wen, Z., Pei, H., Liu, H., and Yue, Z., 2016, "A Sequential Kriging reliability analysis method with characteristics of adaptive sampling regions and parallelizability," *Reliability Engineering & System Safety*, 153, pp. 170-179.
- [33] Yu, Z., Sun, Z., Wang, J., and Chai, X., 2018, A new Kriging-based DoE strategy and its application to structural reliability analysis.
- [34] Cheng, K., Lu, Z., Ling, C., Zhou, S. J. S., and Optimization, M., 2020, "Surrogate-assisted global sensitivity analysis: an overview," pp. 1-27.

- [35] Zhou, H., Zhou, Q., Liu, C., and Zhou, T., 2018, "A kriging metamodel-assisted robust optimization method based on a reverse model," *Engineering Optimization*, 50(2), pp. 253-272.
- [36] Wu, H., Zhu, Z., and Du, X., 2020, "System Reliability Analysis With Autocorrelated Kriging Predictions," *Journal of Mechanical Design*, 142(10).
- [37] Chen, W., Tsui, K.-L., Allen, J. K., and Mistree, F., 1995, "Integration of the response surface methodology with the compromise decision support problem in developing a general robust design procedure," *ASME Design Engineering Technical Conference*, pp. 485-492.
- [38] Hosder, S., Walters, R., and Balch, M., 2007, "Efficient sampling for non-intrusive polynomial chaos applications with multiple uncertain input variables," *Proc. 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, p. 1939.
- [39] Lophaven, S. N., Nielsen, H. B., and Søndergaard, J., 2002, *Aspects of the matlab toolbox DACE*, Citeseer, Technical University of Denmark.

SECTION

2. CONCLUSIONS

The main objective of this study is to develop accurate and efficient methods for uncertainty analysis. This study contains five works. In the first work, an accurate method based on the series expansion, Gauss-Hermite quadrature, and saddle point approximation is developed to calculate high-dimensional normal probabilities. Then the method is applied to estimate structural time-dependent reliability. In the second work, we develop an adaptive Kriging method to estimate product average lifetime. In the third work, a time- and space-dependent reliability analysis method based on the widely used first-order and second-order methods is proposed. In the fourth work, we extend the existing robustness analysis to time- and space-dependent problems and develop an adaptive Kriging method to efficiently evaluate the time- and space-dependent robustness. In the fifth work, we apply the uncertainty analysis to lunar plasma environment modeling and develop an adaptive Kriging method to efficiently estimate the lower and upper bounds of the electric potentials of the photoelectron sheaths near the lunar surface. Based on the above research works, the following conclusions are drawn.

(1) It shows in the first work that the proposed method based on the series expansion, Gauss-Hermite quadrature, and saddle point approximation can calculate high-dimensional normal probabilities accurately, even when the normal probabilities are very small. In all examples, its accuracy is overall significantly better than the widely used randomized quasi

Monte Carlo method and improved equivalent component method. However, the proposed method requires the reduced dimension to not be large.

(2) Statistics-based methods are very expensive to evaluate average product lifetime since they need lifetime testing of many real products. Physics-based methods can dramatically improve efficiency because they only need numerical models of the products. Besides, physics-based methods can predict the average product lifetime in a design stage. The proposed method in the second work, which is based on adaptive training of the Kriging model, can estimate the average product lifetime with a small number of evaluations of the numerical models.

(3) The time- and space-dependent reliability problem is the most complicated and general reliability problem. Efficient and accurate methods remain to be developed. The proposed method in the third work is only a beginning because it can only deal with problems that involve random variables instead of random processes or random fields.

(4) Similar to the time- and space-dependent reliability problem, the time- and space-dependent robustness is also the most complicated and general robustness problem. Examples in the fourth work show that the proposed adaptive training method is a promising method to estimate time- and space-dependent robustness.

(5) When planning an exploratory task on the lunar surface, it is vital to take into consideration the electric potential caused by the lunar plasma. Due to uncertainties and lack of information, it is almost impossible to determine the exact value of the electric potential. However, determining its bounds is sometimes feasible when bounds of the uncertain parameters are given. Determining the output bounds given the input bounds is

modeled as interval propagation. The proposed adaptive training-based method in the fifth work shows good accuracy and efficiency in computing the interval propagation.

Future work includes applying the developed reliability and robustness analysis methods in reliability-based optimal design and robust design.

BIBLIOGRAPHY

- [1] Smith, R. C., 2013, *Uncertainty quantification: theory, implementation, and applications*, Siam.
- [2] Melchers, R. E., 1999, *Structural reliability analysis and prediction*, Wiley, Chichester.
- [3] Taguchi, G., 1993, *Taguchi on robust technology development: bringing quality engineering upstream*, ASME Press, New York.
- [4] Saltelli, A., Chan, K., and Scott, E. M., 2000, *Sensitivity analysis*, Wiley New York.
- [5] Lee, S. H., and Chen, W., 2009, "A comparative study of uncertainty propagation methods for black-box-type problems," *Structural Multidisciplinary Optimization*, 37(3), p. 239.
- [6] Choi, S. K., Canfield, R. A., and Grandhi, R. V., 2007, *Reliability-based Structural Design*, Springer London.
- [7] Du, X., Sudjianto, A., and Huang, B., 2005, "Reliability-based design with the mixture of random and interval variables," *Journal of mechanical design*, 127(6), pp. 1068-1076.
- [8] Huang, B., and Du, X., 2008, "Probabilistic uncertainty analysis by mean-value first order saddlepoint approximation," *Reliability Engineering & System Safety*, 93(2), pp. 325-336.
- [9] Zhang, J., and Du, X., 2010, "A second-order reliability method with first-order efficiency," *Journal of Mechanical Design*, 132(10), p. 101006.
- [10] Madsen, H. O., Krenk, S., and Lind, N. C., 2006, *Methods of structural safety*, Courier Corporation.
- [11] Banerjee, B., and Smith, B. G., "Reliability analysis for inserts in sandwich composites," *Proc. Advanced Materials Research*, Trans Tech Publ, pp. 234-238.
- [12] Kim, D.-W., Jung, S.-S., Sung, Y.-H., and Kim, D.-H., 2011, "Optimization of SMES windings utilizing the first-order reliability method," *The Transactions of The Korean Institute of Electrical Engineers*, 60(7), pp. 1354-1359.
- [13] Huang, B., and Du, X., 2006, "Uncertainty analysis by dimension reduction integration and saddlepoint approximations," *Journal of Mechanical Design*, 128(1), pp. 26-33.

- [14] Jin , R., Du , X., and Chen, W., 2003, "The use of metamodeling techniques for optimization under uncertainty," *Structural and Multidisciplinary Optimization*, 25(2), pp. 99-116.
- [15] Isukapalli, S., Roy, A., and Georgopoulos, P., 1998, "Stochastic response surface methods (SRSMs) for uncertainty propagation: application to environmental and biological systems," *Risk analysis*, 18(3), pp. 351-363.
- [16] Zhu, Z., and Du, X., 2016, "Reliability analysis with Monte Carlo simulation and dependent Kriging predictions," *Journal of Mechanical Design*, 138(12), p. 121403.
- [17] Xiong, F., Xiong, Y., Greene, S., Chen, W., and Yang, S., 2009, "A New Sparse Grid Based Method for Uncertainty Propagation," (49026), pp. 1205-1215.
- [18] Ditlevsen, O., and Madsen, H. O., 1996, *Structural reliability methods*, Wiley, New York.
- [19] Sobol', I., 1990, "Quasi-monte carlo methods," *Progress in Nuclear Energy*, 24(1-3), pp. 55-61.
- [20] Dey, A., and Mahadevan, S., 1998, "Ductile structural system reliability analysis using adaptive importance sampling," *Structural Safety*, 20(2), pp. 137-154.
- [21] Au, S.-K., and Beck, J. L., 2001, "Estimation of small failure probabilities in high dimensions by subset simulation," *Probabilistic engineering mechanics*, 16(4), pp. 263-277.
- [22] Andrieu-Renaud, C., Sudret, B., and Lemaire, M., 2004, "The PHI2 method: a way to compute time-variant reliability," *Reliability Engineering & System Safety*, 84(1), pp. 75-86.
- [23] Hu, Z., and Du, X., 2013, "Time-dependent reliability analysis with joint upcrossing rates," *Structural and Multidisciplinary Optimization*, 48(5), pp. 893-907.
- [24] Jiang, C., Wei, X. P., Huang, Z. L., and Liu, J., 2017, "An Outcrossing Rate Model and Its Efficient Calculation for Time-Dependent System Reliability Analysis," *Journal of Mechanical Design*, 139(4), pp. 041402-041402-041410.
- [25] Hu, Z., and Du, X., 2015, "Mixed efficient global optimization for time-dependent reliability analysis," *Journal of Mechanical Design*, 137(5), p. 051401.
- [26] Hu, Z., and Mahadevan, S., 2016, "A single-loop kriging surrogate modeling for time-dependent reliability analysis," *Journal of Mechanical Design*, 138(6), p. 061406.

- [27] Wang, Z., and Chen, W., 2016, "Time-variant reliability assessment through equivalent stochastic process transformation," *Reliability Engineering & System Safety*, 152, pp. 166-175.
- [28] Wang, Z., and Wang, P., 2012, "A Nested Extreme Response Surface Approach for Time-Dependent Reliability-Based Design Optimization," *Journal of Mechanical Design*, 134(12), pp. 121007-121014.
- [29] Wang, Z., Mourelatos, Z. P., Li, J., Baseski, I., and Singh, A., 2014, "Time-dependent reliability of dynamic systems using subset simulation with splitting over a series of correlated time intervals," *Journal of Mechanical Design*, 136(6), p. 061008.
- [30] Singh, A., Mourelatos, Z., and Nikolaidis, E., 2011, "Time-dependent reliability of random dynamic systems using time-series modeling and importance sampling," *SAE International Journal of Materials and Manufacturing*, 4(2011-01-0728), pp. 929-946.
- [31] Chen, J.-B., and Li, J., 2005, "Dynamic response and reliability analysis of non-linear stochastic structures," *Probabilistic Engineering Mechanics*, 20(1), pp. 33-44.
- [32] Du, X., 2014, "Time-dependent mechanism reliability analysis with envelope functions and first-order approximation," *Journal of Mechanical Design*, 136(8), p. 081010.
- [33] Yu, S., and Wang, Z., 2018, "A novel time-variant reliability analysis method based on failure processes decomposition for dynamic uncertain structures," *Journal of Mechanical Design*, 140(5), p. 051401.
- [34] Yu, S., Wang, Z., and Meng, D., 2018, "Time-variant reliability assessment for multiple failure modes and temporal parameters," *Structural and Multidisciplinary Optimization*, pp. 1-13.
- [35] Hu, Z., and Mahadevan, S., "Reliability Analysis of Multidisciplinary System with Spatio-temporal Response using Adaptive Surrogate Modeling," *Proc. 2018 AIAA Non-Deterministic Approaches Conference*, p. 1934.
- [36] Hu, Z., and Mahadevan, S., 2017, "A surrogate modeling approach for reliability analysis of a multidisciplinary system with spatio-temporal output," *Structural and Multidisciplinary Optimization*, 56(3), pp. 553-569.
- [37] Shi, Y., Lu, Z., Cheng, K., and Zhou, Y., 2017, "Temporal and spatial multi-parameter dynamic reliability and global reliability sensitivity analysis based on the extreme value moments," *Structural and Multidisciplinary Optimization*, 56(1), pp. 117-129.

- [38] Shi, Y., Lu, Z., Zhang, K., and Wei, Y., 2017, "Reliability analysis for structures with multiple temporal and spatial parameters based on the effective first-crossing point," *Journal of Mechanical Design*, 139(12), pp. 121403-121403.
- [39] Rausand, M., and Høyland, A., 2004, *System reliability theory: models, statistical methods, and applications*, John Wiley & Sons.
- [40] Meeker, W. Q., and Escobar, L. A., 2014, *Statistical methods for reliability data*, John Wiley & Sons.
- [41] Lawless, J., 1983, "Statistical methods in reliability," *Technometrics*, 25(4), pp. 305-316.
- [42] Hu, Z., and Du, X., 2018, "Integration of Statistics-and Physics-Based Methods—A Feasibility Study on Accurate System Reliability Prediction," *Journal of Mechanical Design*, 140(7).
- [43] Phadke, M. S., 1995, *Quality engineering using robust design*, Prentice Hall PTR.
- [44] Du, X., and Chen, W., 2000, "Towards a better understanding of modeling feasibility robustness in engineering design," *Journal of Mechanical Design*, 122(4), pp. 385-394.
- [45] Huang, B., and Du, X., 2007, "Analytical robustness assessment for robust design," *Structural and Multidisciplinary Optimization*, 34(2), pp. 123-137.
- [46] Mulvey, J. M., Vanderbei, R. J., and Zenios, S. A., 1995, "Robust optimization of large-scale systems," *Operations research*, 43(2), pp. 264-281.
- [47] Goh, J., and Sim, M., 2010, "Distributionally robust optimization and its tractable approximations," *Operations research*, 58(4-part-1), pp. 902-917.
- [48] Fabozzi, F. J., Huang, D., and Zhou, G., 2010, "Robust portfolios: contributions from operations research and finance," *Annals of Operations Research*, 176(1), pp. 191-220.
- [49] Alexandrov, N. M., and Lewis, R. M., 2002, "Analytical and computational aspects of collaborative optimization for multidisciplinary design," *AIAA journal*, 40(2), pp. 301-309.
- [50] Sobieszczanski-Sobieski, J., and Haftka, R. T., 1997, "Multidisciplinary aerospace design optimization: survey of recent developments," *Structural optimization*, 14(1), pp. 1-23.

- [51] Doltsinis, I., Kang, Z., and engineering, 2004, "Robust design of structures using optimization methods," *Computer Methods in Applied Mechanics and Engineering*, 193(23-26), pp. 2221-2237.
- [52] Lagaros, N. D., Plevris, V., and Papadrakakis, M., 2007, "Reliability based robust design optimization of steel structures," *International Journal for Simulation Multidisciplinary Design Optimization*, 1(1), pp. 19-29.
- [53] Cheng, J., Liu, Z., Wu, Z., Li, X., and Tan, J., 2015, "Robust optimization of structural dynamic characteristics based on adaptive Kriging model and CNSGA," *Structural and Multidisciplinary Optimization*, 51(2), pp. 423-437.
- [54] Roy, B. K., and Chakraborty, S., 2015, "Robust optimum design of base isolation system in seismic vibration control of structures under random system parameters," *Structural Safety*, 55, pp. 49-59.
- [55] Fang, J., Gao, Y., Sun, G., and Li, Q., 2013, "Multiobjective reliability-based optimization for design of a vehicledoor," *Finite Elements in Analysis and Design*, 67, pp. 13-21.
- [56] Hwang, K., Lee, K., and Park, G., 2001, "Robust optimization of an automobile rearview mirror for vibration reduction," *Structural and Multidisciplinary Optimization*, 21(4), pp. 300-308.
- [57] Sun, G., Li, G., Zhou, S., Li, H., Hou, S., and Li, Q., 2011, "Crashworthiness design of vehicle by using multiobjective robust optimization," *Structural and Multidisciplinary Optimization*, 44(1), pp. 99-110.
- [58] Lee, T. H., and Jung, J., 2006, "Metamodel-based shape optimization of connecting rod considering fatigue life," *Key Engineering Materials*, 306-308, pp. 211-216.
- [59] Li, F., Meng, G., Sha, L., and Zhou, L., 2011, "Robust optimization design for fatigue life," *Finite Elements in Analysis and Design*, 47(10), pp. 1186-1190.
- [60] Carpinelli, G., Caramia, P., and Varilone, P., 2015, "Multi-linear Monte Carlo simulation method for probabilistic load flow of distribution systems with wind and photovoltaic generation systems," *Renewable Energy*, 76, pp. 283-295.
- [61] Li, F., Luo, Z., Sun, G., and Zhang, N., 2013, "An uncertain multidisciplinary design optimization method using interval convex models," *Engineering Optimization*, 45(6), pp. 697-718.
- [62] Du, X., Guo, J., and Beeram, H., 2008, "Sequential optimization and reliability assessment for multidisciplinary systems design," *Structural and Multidisciplinary Optimization*, 35(2), pp. 117-130.

- [63] Kim, N.-K., Kim, D.-H., Kim, D.-W., Kim, H.-G., Lowther, D. A., and Sykulski, J. K., 2010, "Robust optimization utilizing the second-order design sensitivity information," *IEEE transactions on Magnetics*, 46(8), pp. 3117-3120.
- [64] Papadimitriou, D., and Giannakoglou, K., 2013, "Third-order sensitivity analysis for robust aerodynamic design using continuous adjoint," *International Journal for Numerical Methods in Fluids*, 71(5), pp. 652-670.
- [65] Li, M., Hamel, J., and Azarm, S., 2010, "Optimal uncertainty reduction for multi-disciplinary multi-output systems using sensitivity analysis," *Structural and Multidisciplinary Optimization*, 40(1-6), p. 77.
- [66] Siddiqui, S., Azarm, S., and Gabriel, S., 2011, "A modified Benders decomposition method for efficient robust optimization under interval uncertainty," *Structural and Multidisciplinary Optimization*, 44(2), pp. 259-275.
- [67] Siddiqui, S., Gabriel, S. A., and Azarm, S., 2015, "Solving mixed-integer robust optimization problems with interval uncertainty using Benders decomposition," *Journal of the Operational Research Society*, 66(4), pp. 664-673.
- [68] Chatterjee, T., Chowdhury, R., and Ramu, P., 2019, "Decoupling uncertainty quantification from robust design optimization," *Structural and Multidisciplinary Optimization*, 59(6), pp. 1969-1990.
- [69] Chatterjee, T., Chakraborty, S., and Chowdhury, R., 2019, "A critical review of surrogate assisted robust design optimization," *Archives of Computational Methods in Engineering*, 26(1), pp. 245-274.
- [70] Steuben, J., and Turner, C. J., "Robust Optimization Exploration Using NURBs-Based Metamodeling Techniques," *Proc. ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers Digital Collection, pp. 239-250.
- [71] Steuben, J. C., and Turner, C., 2012, "Robust optimization of mixed-integer problems using NURBs-based metamodels," *Journal of Computing Information Science in Engineering*, 12(4), p. 041010.
- [72] Steuben, J. C., Turner, C. J., and Crawford, R. H., 2013, "Robust engineering design optimization with non-uniform rational B-splines-based metamodels," *Engineering Optimization*, 45(7), pp. 767-786.
- [73] Zhou, H., Zhou, Q., Liu, C., and Zhou, T., 2018, "A kriging metamodel-assisted robust optimization method based on a reverse model," *Engineering Optimization*, 50(2), pp. 253-272.

- [74] Du, X., 2012, "Toward Time-Dependent Robustness Metrics," *Journal of Mechanical Design*, 134(1), p. 011004.
- [75] Du, X., Sudjianto, A., and Chen, W., 2004, "An integrated framework for optimization under uncertainty using inverse reliability strategy," *Journal of Mechanical Design*, 126(4), pp. 562-570.
- [76] Gu, X., Renaud, J. E., Batill, S. M., Brach, R. M., and Budhiraja, A. S., 2000, "Worst case propagated uncertainty of multidisciplinary systems in robust design optimization," *Structural and Multidisciplinary Optimization*, 20(3), pp. 190-213.
- [77] Tsui, K.-L., 1999, "Modeling and analysis of dynamic robust design experiments," *IIE Transactions*, 31(12), pp. 1113-1122.
- [78] Wu, F.-C., 2009, "Robust design of nonlinear multiple dynamic quality characteristics," *Computers & Industrial Engineering*, 56(4), pp. 1328-1332.
- [79] Goethals, P. L., and Cho, B. R., 2011, "The development of a robust design methodology for time-oriented dynamic quality characteristics with a target profile," *Quality and Reliability Engineering International*, 27(4), pp. 403-414.
- [80] Li, C.-C., and Der Kiureghian, A., 1993, "Optimal discretization of random fields," *Journal of Engineering Mechanics*, 119(6), pp. 1136-1154.
- [81] Liu, Q., and Pierce, D. A., 1994, "A note on Gauss—Hermite quadrature," *Biometrika*, 81(3), pp. 624-629.
- [82] Butler, R. W., 2007, *Saddlepoint approximations with applications*, Cambridge University Press, Cambridge.
- [83] Daniels, H. E., 1954, "Saddlepoint Approximations in Statistics," *The Annals of Mathematical Statistics*, 25(4), pp. 631-650.
- [84] Lophaven, S. N., Nielsen, H. B., and Søndergaard, J., 2002, *Aspects of the matlab toolbox DACE*, Citeseer, Technical University of Denmark.
- [85] Zhao, J., Wei, X., Hu, Z., He, X., Han, D., Hu, Z., and Du, X., "Photoelectron Sheath near the Lunar Surface: Fully Kinetic Modeling and Uncertainty Quantification Analysis," *Proc. AIAA Scitech 2020 Forum*, p. 1548.

VITA

Xinpeng Wei received his bachelor's degree in Vehicle Engineering in 2013 from Hunan University, Changsha, Hunan, China. In December 2020, He received his Ph.D. degree in Mechanical Engineering from Missouri University of Science and Technology, Rolla, Missouri, USA.