
Theses and Dissertations

Summer 2011

Reliability-based design optimization using surrogate model with assessment of confidence level

Liang Zhao
University of Iowa

Copyright 2011 Liang Zhao

This dissertation is available at Iowa Research Online: <http://ir.uiowa.edu/etd/1194>

Recommended Citation

Zhao, Liang. "Reliability-based design optimization using surrogate model with assessment of confidence level." PhD (Doctor of Philosophy) thesis, University of Iowa, 2011.
<http://ir.uiowa.edu/etd/1194>.

Follow this and additional works at: <http://ir.uiowa.edu/etd>



Part of the [Mechanical Engineering Commons](#)

RELIABILITY-BASED DESIGN OPTIMIZATION USING SURROGATE MODEL
WITH ASSESSMENT OF CONFIDENCE LEVEL

by
Liang Zhao

An Abstract

Of a thesis submitted in partial fulfillment
of the requirements for the Doctor of
Philosophy degree in Mechanical Engineering
in the Graduate College of
The University of Iowa

July 2011

Thesis Supervisor: Professor Kyung K. Choi

ABSTRACT

The objective of this study is to develop an accurate surrogate modeling method for construction of the surrogate model to represent the performance measures of the compute-intensive simulation model in reliability-based design optimization (RBDO). In addition, an assessment method for the confidence level of the surrogate model and a conservative surrogate model to account the uncertainty of the prediction on the untested design domain when the number of samples are limited, are developed and integrated into the RBDO process to ensure the confidence of satisfying the probabilistic constraints at the optimal design. The effort involves: (1) developing a new surrogate modeling method that can outperform the existing surrogate modeling methods in terms of accuracy for reliability analysis in RBDO; (2) developing a sampling method that efficiently and effectively inserts samples into the design domain for accurate surrogate modeling; (3) generating a surrogate model to approximate the probabilistic constraint and the sensitivity of the probabilistic constraint with respect to the design variables in most-probable-point-based RBDO; (4) using the sampling method with the surrogate model to approximate the performance function in sampling-based RBDO; (5) generating a conservative surrogate model to conservatively approximate the performance function in sampling-based RBDO and assure the obtained optimum satisfy the probabilistic constraints.

In applying RBDO to a large-scale complex engineering application, the surrogate model is commonly used to represent the compute-intensive simulation model of the performance function. However, the accuracy of the surrogate model is still challenging for highly nonlinear and large dimension applications. In this work, a new method, the

Dynamic Kriging (DKG) method is proposed to construct the surrogate model accurately. In this DKG method, a generalized pattern search algorithm is used to find the accurate optimum for the correlation parameter, and the optimal mean structure is set using the basis functions that are selected by a genetic algorithm from the candidate basis functions based on a new accuracy criterion. Plus, a sequential sampling strategy based on the confidence interval of the surrogate model from the DKG method, is proposed. By combining the sampling method with the DKG method, the efficiency and accuracy can be rapidly achieved.

Using the accurate surrogate model, the most-probable-point (MPP)-based RBDO and the sampling-based RBDO can be carried out. In applying the surrogate models to MPP-based RBDO and sampling-based RBDO, several efficiency strategies, which include: (1) using local window for surrogate modeling; (2) adaptive window size for different design candidates; (3) reusing samples in the local window; (4) using violated constraints for surrogate model accuracy check; (3) adaptive initial point for correlation parameter estimation, are proposed.

To assure the accuracy of the surrogate model when the number of samples is limited, and to assure the obtained optimum design can satisfy the probabilistic constraints, a conservative surrogate model, using the weighted Kriging variance, is developed, and implemented for sampling-based RBDO.

Abstract Approved: _____
Thesis Supervisor

Title and Department

Date

RELIABILITY-BASED DESIGN OPTIMIZATION USING SURROGATE MODEL
WITH ASSESSMENT OF CONFIDENCE LEVEL

by
Liang Zhao

A thesis submitted in partial fulfillment
of the requirements for the Doctor of
Philosophy degree in Mechanical Engineering
in the Graduate College of
The University of Iowa

July 2011

Thesis Supervisor: Professor Kyung K. Choi

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

PH.D. THESIS

This is to certify that the Ph.D. thesis of

Liang Zhao

has been approved by the Examining Committee
for the thesis requirement for the Doctor of Philosophy
degree in Mechanical Engineering at the July 2011 graduation.

Thesis Committee: _____
Kyung K. Choi, Thesis Supervisor

Yong Chen

Ikjin Lee

Jia Lu

Shaoping Xiao

Olesya Zhupanska

To my parents and my wife

ACKNOWLEDGMENTS

First and foremost I want to thank my advisor Professor Kyung K. Choi. It has been a great honor to be his Ph.D. student. He has taught me, both consciously and unconsciously, how excellent thoroughly research is done. I appreciate all his contributions of time, ideas, and funding to make my Ph.D. experience productive and stimulating. The joy and enthusiasm he has for his research was contagious and motivational for me, even during tough times in the Ph.D. pursuit.

The members of the Choi group have contributed immensely to my personal and professional time at Iowa. The group has been a source of friendships as well as good advice and collaboration. I would like to acknowledge Dr. Liu Du who helped me going through the intensity of research at my early year. I would also like to acknowledge Dr. Ikjin Lee for the co-advising and co-working with me that made this research to be the integrity of the achievements from several former students. I would specially thank Mr. Nicholas Gaul and Dr. Edwin Hardee for their strong background on the finite element analysis and software development that made this research to be applicable for the engineering applications. Other past and present group members that I have had the pleasure to work with are Dr. Yoojeong Noh, Mr. Hyeongjin Song, Mr. Hyunkyoo Cho, Mr. Weifei Hu and Mr. Sangjune Bae.

In regards to the committee members, I would like to thank all of them for the great advice for the research study as well as thesis writing. I would like to thank Professor Yong Chen for his advice for the basis function selection in Dynamic Kriging method. I would like to thank Professor Jia Lu for teaching me the finite element method course as well as the thoroughly coding for it. I would like to thank Professor Shaoping

Xiao for his great comments and suggestions for my thesis and the help for my qualifying exam. I would also like to thank Professor Olesya Zhupanska for her great comments and effort for my thesis as well as her great effort for my defense exam even if she was on the visit.

For the financial support, this research is primarily supported by the ARO Project W911NF-09-1-0250 and the Automotive Research Center which is sponsored by the U.S. Army Tank Automotive Research, Development and Engineering Center (TARDEC). This research was also partially supported by the World Class University Program through the National Research Foundation of Korea (NRF) grant funded by the Ministry of Education, Science and Technology (Grant Number R32-2008-000-10161-0 in 2009). These supports are greatly appreciated.

TABLE OF CONTENTS

LIST OF TABLES	viii
LIST OF FIGURES	x
CHAPTER	
I. INTRODUCTION	1
1.1 Background and Motivation	2
1.1.1 Reliability-Based Design Optimization.....	2
1.1.2 Surrogate Modeling Methods	4
1.2 Objectives of the Proposed Study.....	9
1.3 Organization of Thesis.....	11
II. FUNDAMENTAL CONCEPTS IN DESIGN UNDER UNCERTAINTY	14
2.1 Introduction.....	14
2.2 Reliability Analysis	14
2.2.1 Transformation	15
2.2.2 First Order Reliability Method (FORM).....	17
2.2.3 Second Order Reliability Method (SORM).....	17
2.3 Inverse Reliability Analysis.....	18
2.4 Most Probable Point (MPP) Based RBDO	21
2.4.1 MPP-Based RBDO Using FORM.....	21
2.4.2 MPP-Based RBDO Using Dimension Reduction Method (DRM)	23
2.5 Sampling-Based RBDO.....	24
2.5.1 Reliability and Statistical Moments.....	24
2.5.2 Stochastic Sensitivity Analysis.....	25
2.5.3 Formulation of Sampling-Based RBDO	27
III. SURROGATE MODELING METHODS.....	29
3.1 Introduction.....	29
3.2 Existing Surrogate Modeling Methods	29
3.2.1 Polynomial Response Surface Method.....	29
3.2.2 Radial Basis Function Method	30
3.2.3 Support Vector Regression Method	31
3.2.4 Universal Kriging Method.....	33
3.2.5 Blind Kriging Method	36
3.3 Dynamic Kriging Method (DKG)	38
3.3.1 Using Pattern Search Method for Correlation Coefficient Estimation.....	38
3.3.2 Effect of Using Different Basis Function for Kriging Method	47
3.3.3 Using Genetic Algorithm (GA) for Basis Function Selection.....	50
3.3.4 Performance and Robustness of GA-Based Basis Function Selection	54

3.3.5	Confidence Level of Dynamic Kriging (DKG).....	57
3.4	Comparison Study	58
3.5	Numerical Examples for Surrogate Modeling.....	60
3.5.1	2-D Branin-Hoo Example.....	60
3.5.2	12-D M1A1 Tank Roadarm Example	60
3.6	Deterministic Design Optimization Using Dynamic Kriging Method.....	64
3.6.1	Local Window for Surrogate Modeling	65
3.6.2	Sampling Strategy	67
3.6.3	Adaptive Initial Point for Pattern Search and Basis Function Selection	69
3.6.4	Numerical Example.....	71
IV.	DESIGN OF EXPERIMENT	75
4.1	Introduction.....	75
4.2	Sequential Sampling Method Using DKG	76
4.3	Comparison of Sequential Sampling and Latin Hypercube Sampling.....	80
V.	MPP-BASED RBDO USING SURROGATE MODEL	84
5.1	Introduction.....	84
5.2	Sequential Sampling Surrogate Modeling for MPP-Based RBDO	84
5.2.1	Center Fixed Local Support for Surrogate Model in Reliability Analysis	85
5.2.2	Adjustment of Insertion Criterion and Error Definition for Surrogate Modeling.....	88
5.2.3	Adaptive Local Window Size in Deterministic Design Optimization.....	89
5.2.4	Overall Process of MPP-Based RBDO Using Surrogate Modeling.....	93
5.3	Numerical Examples.....	95
5.3.1	2-D Mathematical Example.....	96
5.3.2	M1A1 Tank Roadarm Example.....	97
VI.	SAMPLING-BASED RBDO USING SURROGATE MODEL.....	101
6.1	Introduction.....	101
6.2	Sampling-Based RBDO Using Sequential-Sampling-Based Dynamic Kriging	102
6.3	Efficiency Strategies for Multiple Constraints and Optimization	107
6.3.1	Sample Reuse Strategy	107
6.3.2	Using Violated Constraints for Accuracy Check	108
6.3.3	Adaptive Initial Point for Correlation Parameter Estimation	109
6.4	Numerical Examples.....	110
6.4.1	2-D Mathematical Example.....	110
6.4.2	M1A1 Tank Roadarm Example.....	113
VII.	PARALLELIZATION OF RBDO USING SURROGATE MODEL.....	115

7.1	Introduction.....	115
7.2	Parallel Computing Toolbox in Matlab	116
7.3	Parallelization in RBDO	118
7.3.1	Parallelization of Genetic Algorithm in DKG.....	118
7.3.2	Parallelization of Surrogate Models for Multiple Constraints in RBDO.....	119
7.3.3	Parallelization of MCS in Reliability Analysis	119
7.3.4	Parallelization of Computer Aided Engineering Analysis (CAE)	120
7.3.5	Summary of Parallelization in Sample-Based RBDO Using SS-DKG.....	121
7.4	Numerical Examples.....	121
VIII.	CONSERVATIVE SURROGATE MODEL USING WEIGHTED KRIGING VARIANCE FOR SAMPLING-BASED RBDO	123
8.1	Introduction.....	123
8.2	Sampling-Based RBDO using Conservative Surrogate Model	125
8.2.1	Corrected Akaike Information Criterion (AICc).....	125
8.2.2	Weighted Kriging Prediction Variance for Conservative Surrogate Model	126
8.2.3	Sampling-Based RBDO Using the Conservative Surrogate Model.....	132
8.3	Numerical Example	134
IX.	CONCLUSTION, RESEARCH PROGRESS AND FUTURE WORK.....	138
9.1	Conclusion	138
9.2	Research Progress	140
9.3	Future Work.....	141
	REFERENCES	142

LIST OF TABLES

Table

2.1	Probability Distribution and Its Transformation between X and U -space	16
3. 1	Comparison between Four Optimization Methods.....	44
3.2	Frequency of Rank of $\psi(\theta)$ Function Value at Optimum θ	45
3.3	Frequency of Rank of <i>rRMSE</i> Values by Different Optimization Methods.....	45
3.4	Comparison between Four Optimization Methods.....	46
3.5	Frequency of Rank of $\psi(\theta)$ Function Value at Optimum θ	47
3.6	Frequency of Rank of <i>rRMSE</i> Values by Different Optimization Methods.....	47
3.7	<i>rRMSE</i> of Kriging Methods with Different Basis Functions.....	48
3.8	R^2 Median History (Branin-Hoo, 50 trials)	61
3.9	Frequency of Rank of Five Methods (Branin-Hoo, 18 pts, 50 trials).....	61
3.10	Mean <i>rRMSE</i> Values for Each Method (Branin-Hoo, 18 pts, 50 trials).....	61
3.11	Frequency of Rank of Five Methods (Roadarm, 50 pts, 50 trials)	65
3.12	Mean <i>rRMSE</i> Values for Each Method (Roadarm, 50 pts, 50 trials)	65
3.13	Design Variables and Design Domain.....	71
3.14	Comparison between Sensitivity-based and Surrogate-based Optimum.....	73
4.1	Comparison of Three Sampling Methods	81
4.2	Error Comparison of Three Sampling Methods	82
5.1	Comparison Between Fixed & Adaptive Support Size	93
5.2	Properties of Random Variables	96
5.3	Comparison Between Two RBDO Methods	98
5.4	Properties of Random Variables of Roadarm	99
5.5	Comparison Between Two RBDO Methods	100
6.1	First-Order Score Function for μ_i for Independent Random Variables	104
6.2	Log-derivative of Copula Density Function	105

6.3	Improvement of Efficiency Using Different Initial Searching Points	110
6.4	Comparison of Various RBDOs ($P_F^{Tar} = 2.275\%$)	112
6.5	Comparison Between Different RBDO Methods	114
7.1	Comparison of Computational Time	122
8.1	Optimum Designs and the Probability of Failure at the Optimum	136
8.2	Cost Function and Probability of Failure at Optimum Design (50 trials)	136

LIST OF FIGURES

Figure

2.1	MPP and Reliability Index β_{HL} in U -space [Wei, 2006]	18
2.2	Difference Between Reliability Analysis and Inverse Reliability Analysis	20
3.1	The Effect of Different θ for the Accuracy of Kriging Prediction	40
3.2	Branin-Hoo Example	43
3.3	The 2-D Example for Basis Function Selection	49
3.4	Contours Using Different Basis Function Selections	56
3.5	Comparison between GA and EHA Selections	57
3.6	Finite Element Model of Roadarm	62
3.7	Fatigue Life Contour and Critical Nodes of Roadarm.....	62
3.8	Shape Design Variables of Roadarm	63
3.9	Local Window for Surrogate Model.....	66
3.10	Contour of True Response at Different Local Windows	67
3.11	Flowchart of Design Optimization Using The DKG Method.....	70
3.12	Mean Value of Time Spent for the DKG Method	74
4.1	Surrogate Model Using Sequential Sampling Method	78
4.2	Entire Process of Sequential-Sampling-Based DKG for Surrogate Modeling	79
5.1	Local Support Size of Surrogate Model for Reliability Analysis in U -Space	86
5.2	Demonstration of Ill-conditioned Matrix Problem in U -Space	87
5.3	Neighboring Area of the Point of Interest for Error Checking	90
5.4	Relation Between Support Size and Design Movement.....	91
5.5	Contour of Constraint and Cost Functions	93
5.6	Flowchart of SS-DKG Toolbox.....	94
5.7	Flowchart of MPP-based RBDO Using SS-DKG	95
6.1	Sampling-Based RBDO Using SS-DKG.....	106

6.2	The Local Window for Surrogate Model.....	107
6.3	RBDO History and Optimum Design.....	112
7.1	Parallel Computing Toolbox in Matlab.....	117
7.2	Workflow of Parallelization in RBDO.....	121
8.1	1-D Problem with 7 Samples.....	127
8.2	1-D Problem with 6 Samples.....	128
8.3	Leave-one-out Kriging Prediction.....	130
8.4	Conservative Surrogate Model Using Weighted Kriging Variance (7 samples)....	133
8.5	Conservative Surrogate Model Using Weighted Kriging Variance (6 samples)....	133
8.6	Different Conservative Surrogate Models and Optimum Designs.....	135

CHAPTER I

INTRODUCTION

This study presents new methods in reliability-based design optimization (RBDO) using the surrogate model with associated confidence level. First, a new dynamic Kriging (DKG) surrogate modeling method, which accurately represents the true unknown performance function by using the pattern search for correlation parameter estimation and a genetic algorithm for basis functions selection in the Kriging method, is proposed. Second, based on the proposed surrogate modeling method, a confidence interval of the surrogate model is obtained and used to develop a sequential sampling method. Third, the sequential-sampling-based dynamic Kriging method is applied to both the most probable point based RBDO and sampling-based RBDO to achieve the RBDO optimum design. Fourth, to ensure the efficiency of the sampling-based RBDO using surrogate model, a parallel processing method is applied by using the high performance computing system (HPC) for surrogate modeling and the Monte Carlo simulation for reliability analysis. In the end, an assessment method for the confidence of the surrogate model is developed in RBDO to ensure the confidence level of the obtained optimum design in the presence of the uncertainty from the surrogate modeling.

Section 1.1 presents background and motivation of the proposed research; Section 1.2 provides objectives of the proposed research; and Section 1.3 describes the thesis organization.

1.1 Background and Motivation

1.1.1 Reliability-Based Design Optimization

Reliability-based design optimization (RBDO) has been widely used recently for design of engineering applications. To achieve the optimal design that satisfies the target reliability of performance function, the optimization algorithm and reliability analysis are implemented. In recent years, there have been various attempts to develop enhanced reliability analysis methods to accurately compute the probability of failure of a performance function.

The most common reliability analysis methods are (1) analytical methods and (2) simulation or sampling methods. The analytical methods have two different branches. One branch includes the Most-Probable-Point (MPP)-based method, which includes the first-order reliability method (FORM) [Haldar and Mahadevan, 2000; Hasofer and Lind, 1974; Tu and Choi, 1999; Tu et al., 2001], the second-order reliability method (SORM) [Hohenbichler and Rackwitz, 1986; Breitung, 1984], and the MPP-based dimension reduction method (DRM) [Rahman and Wei, 2006; Lee et al., 2008]. The other branch includes the probability density function (PDF) approximation method [Rosenblueth, 1975; Huang and Du, 2006; Youn et al., 2008]. Among the MPP-based methods, FORM and SORM first use first or second order Taylor series expansion to approximate the performance function at the MPP, respectively; and then calculate the probability of failure using the approximated performance function, respectively. The MPP-based DRM approximates the multi-dimensional performance function by using the sum of lower dimensional functions and then calculates the probability of failure. The PDF approximation method evaluates the PDF of the performance function by assuming a

general output distribution type and then, using the approximated PDF, evaluates the probability of failure of the performance function.

RBDO using MPP-based reliability analysis is a gradient-based design optimization, which requires the sensitivity of the probabilistic constraint at MPP with respect to the design variable, which is the mean value of input random variable. Many works have been conducted to study the sensitivities of the probabilistic constraints [Haldar and Mahadevan, 2000; Tu and Choi, 1999; Ditlevsen and Madsen, 1996; Hou, 2004; Gumbert et al., 2003; Hohenbichler and Rackwitz, 1986; Rahman and Wei, 2007]. The sensitivity using the Reliability Index Approach (RIA) for the FORM-based RBDO [Haldar and Mahadevan, 2000; Tu and Choi, 1999; Ditlevsen and Madsen, 1996; Hou, 2004; Gumbert et al., 2003; Hohenbichler and Rackwitz, 1986] and the DRM-based RBDO [Rahman and Wei, 2007] is derived in detail. Also, the sensitivity using the Performance Measure Approaches (PMA) for the DRM-based RBDO [Tu and Choi, 2001; Youn et al., 2005; Lee et al., 2009] is derived in detail. However, in most complex engineering applications, the sensitivity of the probabilistic constraints is unavailable or extremely difficult to obtain. In such cases, surrogate model is commonly used to approximate the performance function and the sensitivity of the performance function with respect to the design variable by evaluating the constraint function at limited number of sample points.

The simulation or sampling methods for reliability analysis usually use Monte Carlo simulation (MCS) for the evaluation of the probabilistic constraints. In such cases, MCS generates a large number of samples, e.g., one million, and evaluates the constraint at these MCS samples to check if the constraint is violated or not. In the end, the

probability of failure is obtained as the number of samples at which the constraint is violated divided by the total number of MCS samples. For the sensitivity analysis, score function [Rahman, 2009; Lee et al, 2010] is used to analytically calculate the sensitivity of the probabilistic constraint based on the MCS result.

RBDO using the sampling method for reliability analysis can avoid the approximation made in FORM/SORM/DRM for the probabilistic constraints and the system reliability can be directly calculated accurately. However, the directly use of this approach is prohibited in complex engineering applications as MCS requires a very large number of samples for reliability analysis. In such cases, an accurate surrogate model would be desirable to represent the performance function. With the accurate surrogate model, probability of failure and the sensitivity calculation are obtained by carrying out MCS based on the prediction from the surrogate model instead of evaluating the performance function directly.

1.1.2 Surrogate Modeling Methods

Surrogate modeling techniques have been developed over several decades in use of simulation based design optimization. The polynomial response surface method [Box and Wilson, 1951] is commonly considered as the origin of surrogate modeling methods, which uses a set of polynomials to represent the true response by minimizing the mean squared error (MSE). Kriging method [Matheron, 1963; Sacks et al., 1989; Cressie, 1993; Martin and Simpson, 2005; Joseph, 2008], which uses the random field to handle the residue, has become popular due to its flexibility and accuracy for highly nonlinear problems. Multivariate adaptive regression splines (MARS) [Friedman and Silverman, 1989; Friedman, 1991], is a linear model with a forward stepwise algorithm to select

model terms followed by a backward procedure to prune the model. Artificial neural networks (ANNs) [Lippmann, 1987; Haykin, 1999; White, 1989 Barron et al. 1992; Ripley, 1993], which are represented by a diagram of nodes in various layers with weighted connections between nodes in different layers, have been very popular for modeling a variety of physical relationships. Radial basis functions (RBF) method [Powell, 1987; Meckesheimer et al., 2001], uses a linear combination of basis functions to represent the response. Support vector regression (SVR) [Cortes and Vapnik, 1995] extends the RBF and specifies a margin as the acceptance for errors in the samples without affecting the prediction of the surrogate model.

Among the existing methods, Kriging method is one of the most popular methods due to its flexibility for highly nonlinear problems. However, during the practical use of the Kriging method, two problems have been discovered. The first problem is that the existing optimization methods used to find the optimal correlation parameter do not provide the global optimum. Usually this correlation parameter is obtained by applying the maximum likelihood estimator (MLE). The commonly used DACE toolbox for Kriging method [Lophaven, 2002] uses the modified Hooke and Jeeves (H-J) algorithm to find the optimum solution. Martin [Martin, 2009] uses Levenberg-Marquardt (L-M) method with using the scoring method to efficiently and accurately calculate the Hessian matrix for optimization. Forrester et al [Forrester et al, 2009] instead use the genetic algorithm which is a gradient-free method to find the optimum. All these methods have their own advantages and disadvantages. The modified H-J method is efficient but unable to provide the true optimum. The L-M method is a gradient-based method. Although it is efficient, it can only find a local optimum, and thus the obtained optimum is affected by

the initial search point. Moreover, due to the large flat region and multiple local minima of the objective function, the L-M method often stops prematurely before converging to a true optimum. The GA method is supposed to be able to find the global optimum, but it is less efficient, and the obtained optimum varies due to the randomness within a genetic algorithm.

In this research, a generalized pattern search algorithm is used to find the optimal correlation parameter for the Kriging method accurately and efficiently based on the maximum likelihood estimator (MLE). The second problem is that neither the ordinary Kriging (OKG) nor the universal Kriging (UKG) can adaptively fit the mean structure of the Kriging model for highly nonlinear functions; and fails to characterize the local nonlinearity of the true function in different design areas. It is shown that different basis functions may yield different surrogate models with the same sample profile. This is especially the case when a local window is used to generate the surrogate model in the design optimization process. That is, different basis functions may need to be used at on different local windows. Therefore, a new method that optimally selects basis functions to represent the mean structure based on current samples within the local window is desirable.

One method of adjusting the mean structure was proposed by Joseph et al. [Joseph et al., 2008] by using a Bayesian framework to identify the mean structure for the Kriging method. They use a Bayesian forward variable selection, which can be trapped into a local optimum and prevents itself from finding a global optimal subset of the basis functions. In this study, a new method is proposed to find the pseudo-global optimal basis

functions by applying a genetic algorithm (GA) for the selection procedure by minimizing the process variance in Kriging model.

Another crucial issue in surrogate modeling is sampling strategy. The one-step sampling method, i.e. Latin hypercube sampling method (LHS) [Stein, 1987; Goel et al., 2008] has been widely used in surrogate modeling. This method tries to occupy the entire design domain most evenly and gain as much information about the true model as it can. However, it is not a problem-adaptive method, which means that it always gives us a similar sample profile that occupies the entire domain evenly regardless of the distribution of the nonlinearity of the true response. This could be a critical problem if the distribution of the nonlinear area is aggregating in a particular region of the domain. Another sampling technique, importance sampling [Dey et al., 1998], samples around the limit state area and predicts the response accurately around the limit state. This importance sampling method also only gives a good local surrogate model around the limit state area and usually does not represent the true model accurately enough in other areas of the domain. Wang [Wang, 2003] used a sequential sampling strategy to achieve optimal design and Xiong et al. [Xiong et al., 2008] used an objective-oriented sequential sampling method to insert sample in a sequential manner. In this study, it is proposed to use a sequential sampling strategy integrated with the proposed DKG method. By coupling the sampling method with the DKG method, the efficiency and accuracy can be rapidly achieved.

When the surrogate model is not accurate enough, new samples are sequentially inserted to improve the prediction accuracy. However, inserting a very large number of new samples sequentially may not be desirable when the samples are from the physical

experiments or when the computational resource for simulation is limited. Therefore, the improvement of the surrogate model cannot be obtained from the increase of the sample numbers. When applying the surrogate model for reliability-based design optimization when the number of samples is small, to assure the obtained optimum design will satisfy the probabilistic constraints, a conservative surrogate model is needed. Picheny [Picheny, 2009] used both safety margin and safety factor approaches to construct the conservative surrogate model and concluded that, when the Kriging method is used, both methods provide similar performance in terms of the conservativeness and the accuracy. Hertog et al. [Hertog et al., 2006] and Luna and Young [Luna and Young, 2003] used the bootstrapping method to estimate the Kriging prediction interval to construct the conservative surrogate model. The bootstrapping variance is larger than the traditional Kriging prediction variance by considering the uncertainty from the correlation parameter in the Kriging method. However, the bootstrapping procedure is time-consuming and is not applicable for high-dimensional problems.

Viana et al. [Viana et al, 2010] used cross-validation to estimate the safety margin for the conservative surrogate model. While the cross-validation error is widely used to estimate the prediction error, this constant safety margin approach does not distinguish the prediction error at different sample locations and often obtains an over-conservative surrogate model where the samples are aggregated around and under-conservative surrogate model where the samples are sparse.

In this study, a weighted Kriging variance is proposed to construct a more accurate conservative surrogate model based the sample locations. To evaluate the importance from each sample and determine the weight for it, an accuracy measure of the

surrogate model is needed first. The corrected Akaike information criterion (AICc) is used to assess the accuracy changes in the surrogate model during the cross-validation process, and an importance function value is assigned to each sample according to the relative changes in AICc of the surrogate model and the weight is also assigned to each sample according to the importance function values. Then a weighted Kriging variance is calculated based on the weight values for each sample. By applying this weighted Kriging variance, one can construct the conservative surrogate model for dynamic Kriging and use it for sampling-based RBDO to assure the obtained optimum design can satisfy the probabilistic constraint.

1.2 Objectives of the Proposed Study

The first objective of this study is to propose the DKG surrogate modeling method that improves the traditional Kriging method by using the pattern search for correlation parameter estimation and genetic algorithm for selection of basis functions. To show advantages of the proposed method, comparison studies will be carried out. For the comparison study, popular existing surrogate modeling methods, such as polynomial response surface, radial basis function, support vector regression, universal Kriging and blind Kriging, are used to compare with the proposed DKG method.

The second objective of this study is to propose a sequential sampling method that is inherently derived from the proposed DKG method. This sequential sampling efficiently locates the position in the design domain where it has the least prediction confidence and insert new sample respectively. To demonstrate the efficiency of the proposed sampling method, a comparison study will be carried out. For the comparison study, one-step sampling method, i.e. Latin hypercube sampling method, is compared

with the sequential sampling method to see if the sequential sampling method can provide a better sample profile to generate an accurate surrogate model.

The third objective of this study is to apply the proposed sequential-sampling based dynamic Kriging method to both MPP-based RBDO in which both the performance function value and the sensitivity of the performance function are approximated from the surrogate models; and the sampling-based RBDO in which the surrogate model is used to represent the performance function only. The comparison study will be carried out to see if the RBDO optimum design obtained by using the surrogate model is accurate or not compared with the optimum design obtained by using the analytical probabilistic constraints. Then, to ensure the practical use of using the surrogate model for both MPP-based RBDO and sampling-based RBDO for large-scale complex engineering applications, a high performance computing improvement is developed by parallelizing the surrogate modeling the MCS calculation for multiple constraints.

The last objective of the study is to propose conservative surrogate model that can take into account the uncertainty of the surrogate model due to limited number of samples in the RBDO framework. To see the effect on the accuracy of the RBDO optimum design, the obtained RBDO optimum using the conservative surrogate model is verified for the probabilistic constraints. A statistical study is also carried out to show if the conservative surrogate model performs robustly regardless of the sample profile.

1.3 Organization of Thesis

Chapter 2 presents fundamental concepts for the reliability analysis and reliability-based design optimization, which are helpful to better understand the proposed methods.

Chapter 3 presents the popular existing surrogate modeling methods and then proposes the new dynamic Kriging method. The new dynamic Kriging method is compared with the other five existing surrogate modeling methods through three numerical examples.

Chapter 4 presents the sequential sampling method derived from the DKG method using the bandwidth of the prediction interval of the surrogate model. The sequential sampling method is compared with the Latin hypercube sampling method by using two numerical examples.

Chapter 5 presents application of sequential sampling based DKG (SS-DKG) method to MPP-based RBDO. The way of using SS-DKG to approximate the performance function value and its sensitivity are described. Three strategies, which are using local window for surrogate modeling, using neighboring region of the point of interest for accuracy check, and adaptive window size depending on the movement of the design change, are proposed later to improve both the accuracy and the efficiency of surrogate modeling. Two numerical examples, which are 2-D mathematical example and 12-D engineering example, are used to demonstrate the accuracy of using SS-DKG for MPP-based RBDO.

Chapter 6 presents application of the SS-DKG method to sampling-based RBDO. The way of using SS-DKG to represent the performance function in sampling-based

RBDO and the stochastic sensitivity analysis using the surrogate model are described in detail. Three strategies, which are (1) adaptively reusing existing samples, (2) using violated constraints for surrogate model generation, and (3) adaptively using optimal correlation parameter value obtained from previous design iteration, are proposed. The same two numerical examples, which are 2-D mathematical example and 12-D engineering example, are used to demonstrate the accuracy of using SS-DKG for sampling-based RBDO.

Chapter 7 proposes the parallelization of RBDO using the surrogate model for high performance computing in large-scale engineering application. The concept and toolbox of parallel computing is introduced. Then, three parts of parallelization, which are parallelization of the genetic algorithm in DKG, parallelization of surrogate models for multiple constraints, and parallelization of MCS prediction using the surrogate model, are explained in detail. Finally, a numerical example is used to demonstrate the improvement on the efficiency after applying parallelization for RBDO using surrogate model.

Chapter 8 proposes the conservative surrogate model for sampling-based RBDO. To construct a desirable conservative surrogate model that does not change the optimum region of the true response and adaptively identifies the uncertainty from the surrogate model when the number of samples is small, a weighted Kriging variance that uses the relative changes in AICc of the Kriging model to quantify the importance of each sample point and assign the weight to each sample point, is proposed. Numerical example is carried out to compare the optimum design obtained by using the conservative surrogate model with the weighted Kriging variance and the conservative surrogate model with the

constant safety margin method. The result shows that the proposed method can adaptively construct the conservative surrogate model and assure the obtained RBDO optimum satisfy the probabilistic constraints.

Chapter 9 provides conclusions from the study carried out and proposes additional study on the efficiency of the surrogate model in RBDO for future research.

CHAPTER II

FUNDAMENTAL CONCEPTS IN DESIGN UNDER UNCERTAINTY

2.1 Introduction

This chapter presents review of fundamental concepts in design under uncertainty. Section 2.2 and 2.3 discuss the basic idea of reliability analysis and inverse reliability analysis which are necessary for RBDO that will be explained in Section 2.4. Section 2.5 discuss the recently developed RBDO method that uses the Monte Carlo Simulation to calculate the stochastic sensitivity analysis and therefore does not require any approximation in calculating the probability of failure for the performance function.

2.2 Reliability Analysis

A reliability analysis entails calculation of probability of failure, denoted by P_F , which is defined using a multi-dimensional integral [Madsen et al., 1986]

$$P_F \equiv P[G(\mathbf{X}) > 0] = \int_{G(\mathbf{x}) > 0} f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} \quad (2.1)$$

where $\mathbf{X} = \{X_1, X_2, \dots, X_N\}^T$ is an N -dimensional random vector, $G(\mathbf{X})$ is the performance function such that $G(\mathbf{X}) > 0$ is defined as failure, and $f_{\mathbf{x}}(\mathbf{x})$ is a joint probability density function (PDF) of the random input variable \mathbf{X} . In most practical engineering applications, the exact evaluation of Eq. (2.1) is very difficult or often impossible to obtain since $f_{\mathbf{x}}(\mathbf{x})$ is generally non-Gaussian and $G(\mathbf{X})$ is highly nonlinear. To handle the non-Gaussian $f_{\mathbf{x}}(\mathbf{x})$, a transformation from the original \mathbf{X} -space into the independent standard normal \mathbf{U} -space is introduced. In addition, $G(\mathbf{X})$ is approximated using the first-order Taylor series expansion in the First Order Reliability

Method (FORM) or the second-order Taylor series expansion in the Second Order Reliability Method (SORM) if $G(\mathbf{X})$ is highly nonlinear.

2.2.1 Transformation

Consider an N -dimensional random vector \mathbf{X} with a joint cumulative distribution function (CDF) $F_{\mathbf{X}}(\mathbf{x})$. Let $T: \mathbf{X} \rightarrow \mathbf{U}$ denote a transformation from \mathbf{X} -space to \mathbf{U} -space that is defined by Rosenblatt transformation [Rosenblatt, 1952] as

$$T: \begin{cases} u_1 = \Phi^{-1} [F_{X_1}(x_1)] \\ u_2 = \Phi^{-1} [F_{X_2}(x_2|x_1)] \\ \vdots \\ u_N = \Phi^{-1} [F_{X_N}(x_N|x_1, x_2, \dots, x_{N-1})] \end{cases} \quad (2.2)$$

where $F_{X_i}(x_i|x_1, x_2, \dots, x_{i-1})$ is the conditional CDF given by

$$F_{X_i}(x_i|x_1, x_2, \dots, x_{i-1}) = \frac{\int_{-\infty}^{x_i} f_{X_1 X_2 \dots X_i}(x_1, x_2, \dots, x_{i-1}, \xi) d\xi}{f_{X_1 X_2 \dots X_{i-1}}(x_1, x_2, \dots, x_{i-1})} \quad (2.3)$$

and $\Phi(\bullet)$ is the standard normal CDF given by

$$\Phi(u) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^u \exp\left(-\frac{1}{2}\xi^2\right) d\xi \quad (2.4)$$

The inverse transformation can be obtained from Eq. (2.2) as

$$T^{-1}: \begin{cases} x_1 = F_{X_1}^{-1} [\Phi(u_1)] \\ x_2 = F_{X_2}^{-1} [\Phi(u_2|x_1)] \\ \vdots \\ x_N = F_{X_N}^{-1} [\Phi(u_N|x_1, x_2, \dots, x_{N-1})] \end{cases} \quad (2.5)$$

If the N -dimensional random vector \mathbf{X} is independent, that is, the joint PDF is given by

$$f_{\mathbf{X}}(\mathbf{x}) = f_{x_1}(x_1) \times f_{x_2}(x_2) \times \cdots \times f_{x_N}(x_N) \quad (2.6)$$

where $f_{x_i}(x_i)$ are the marginal PDFs, then Rosenblatt transformation and the inverse transformation are simplified as

$$u_i = \Phi^{-1}\left[F_{x_i}(x_i)\right] \quad \text{and} \quad x_i = F_{x_i}^{-1}\left[\Phi(u_i)\right] \quad (2.7)$$

where $F_{x_i}(x_i)$ are the marginal CDFs. In this study, the input random variables are assumed to be independent for the simplicity of calculation and the study with dependent input random variables will be the future research topic. Table 2.1 shows five representative distributions and their transformations assuming random variables are independent.

Table 2.1 Probability Distribution and Its Transformation between \mathbf{X} and \mathbf{U} -space

	Parameters	PDF	Transformation
Normal	$\mu = \text{mean}$ $\sigma = \text{standard deviation}$	$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-0.5\left[\frac{x-\mu}{\sigma}\right]^2}$	$X = \mu + \sigma U$
Log-normal	$\bar{\sigma}^2 = \ln\left[1 + \left(\frac{\sigma}{\mu}\right)^2\right]$, $\bar{\mu} = \ln(\mu) - 0.5\bar{\sigma}^2$	$f(x) = \frac{1}{\sqrt{2\pi x\bar{\sigma}}} e^{-0.5\left[\frac{\ln x - \bar{\mu}}{\bar{\sigma}}\right]^2}$	$X = \exp(\bar{\mu} + \bar{\sigma}U)$
Weibull	$\mu = v\Gamma\left(1 + \frac{1}{k}\right)$, $\sigma^2 = v^2\left[\Gamma\left(1 + \frac{2}{k}\right) - \Gamma^2\left(1 + \frac{1}{k}\right)\right]$	$f(x) = \frac{k}{v} \left(\frac{x}{v}\right)^{k-1} e^{-\left(\frac{x}{v}\right)^k}$	$X = v[-\ln(\Phi(-U))]^{\frac{1}{k}}$
Gumbel	$\mu = v + \frac{0.577}{\alpha}$, $\sigma = \frac{\pi}{\sqrt{6}\alpha}$	$f(x) = \alpha e^{-\alpha(x-v)} e^{-e^{-\alpha(x-v)}}$	$X = v - \frac{1}{\alpha} \ln[-\ln(\Phi(U))]$
Uniform	$\mu = \frac{a+b}{2}$, $\sigma = \frac{b-a}{\sqrt{12}}$	$f(x) = \frac{1}{b-a}$, $a \leq x \leq b$	$X = a + (b-a)\Phi(U)$

2.2.2 First Order Reliability Method (FORM)

To calculate the probability of failure of the performance function $G(\mathbf{x})$ using FORM and SORM, it is necessary to find the most probable point (MPP), which is defined as the point \mathbf{u}^* on the limit state function ($g(\mathbf{u}) = 0$) closest to the origin in the standard normal \mathbf{U} -space as shown in Fig. 2.1. In this study, the performance function in \mathbf{U} -space is defined as $g(\mathbf{u}) \equiv G(\mathbf{x}(\mathbf{u})) = G(\mathbf{x})$ using the Rosenblatt transformation. Hence, MPP can be found by solving the following optimization problem to

$$\begin{aligned} & \text{minimize} && \|\mathbf{u}\| \\ & \text{subject to} && g(\mathbf{u}) = 0 \end{aligned} \quad (2.8)$$

After finding MPP, the distance from MPP to the origin is commonly called the Hasofer-Lind reliability index (Hasofer and Lind, 1974) and denoted by β_{HL} , that is, $\beta_{\text{HL}} = \|\mathbf{u}^*\|$. Using the reliability index β_{HL} , FORM can approximate the probability of failure using a linear approximation of the performance function as

$$P_F^{\text{FORM}} \cong \Phi(-\beta_{\text{HL}}) \quad (2.9)$$

2.2.3 Second Order Reliability Method (SORM)

The MPP obtained by solving Eq. (2.8) is also used for the probability of failure calculation using SORM. Using a quadratic approximation of the performance function in \mathbf{U} -space and the rotational transformation from \mathbf{U} -space to \mathbf{V} -space, the probability of failure can be obtained using SORM as (Breitung, 1984; Hohenbichler and Rackwitz, 1988; Rahman and Wei, 2006)

$$P_F^{\text{SORM}} \cong \Phi(-\beta_{\text{HL}}) \left| \mathbf{I}_{N-1} - 2 \frac{\phi(\beta_{\text{HL}})}{\Phi(-\beta_{\text{HL}})} \tilde{\mathbf{A}}_{N-1} \right|^{\frac{1}{2}} \quad (2.10)$$

where $\mathbf{A} = \begin{bmatrix} \tilde{\mathbf{A}}_{N-1} & \tilde{\mathbf{A}}_{1N} \\ \tilde{\mathbf{A}}_{N1} & \tilde{\mathbf{A}}_{NN} \end{bmatrix} = \frac{1}{2\|\nabla g_U\|} \mathbf{R}^T \mathbf{H} \mathbf{R}$, \mathbf{H} is the Hessian matrix evaluated at the MPP

in \mathbf{X} -space, \mathbf{R} is the rotation matrix such that $\mathbf{u} = \mathbf{R}\mathbf{v}$, and $\phi(\bullet)$ is the PDF of a standard Gaussian random variable.

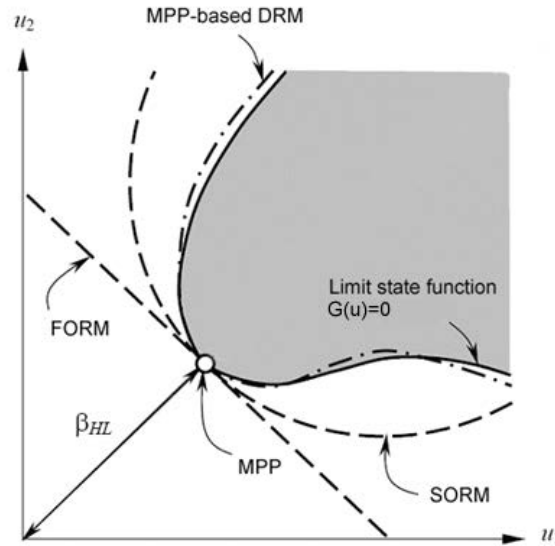


Figure 2. 1 MPP and Reliability Index β_{HL} in \mathbf{U} -space [Wei, 2006]

2.3 Inverse Reliability Analysis

The reliability analysis presented in Section 2.2 is called the reliability index approach (RIA) (Tu and Choi, 1999) since it finds the reliability index β_{HL} using Eq. (2.8). The advantage of RIA is that the probability of failure for the performance function can be calculated at a given design, for example, using Eqs. (2.9) and (2.10).

However, it is well known that the inverse reliability analysis in the performance measure approach (PMA) (Tu and Choi, 1999; Tu et al., 2001; Choi et al., 2001; Youn et al., 2003) is much more robust and efficient than the reliability analysis in RIA. PMA does not calculate the probability of failure directly. Instead, PMA judges whether or not a given design satisfies the probabilistic constraint for a given target probability of failure P_F^{Tar} . Using FORM, the target reliability index β_t can be calculated as $\beta_t = -\Phi^{-1}(P_F^{\text{Tar}})$ using Eq. (2.9), and then the feasibility of the given design can be checked by solving the following optimization problem to

$$\begin{aligned} & \text{maximize} && g(\mathbf{u}) \\ & \text{subject to} && \|\mathbf{u}\| = \beta_t \end{aligned} \quad (2.11)$$

Since Eq. (2.11) is the inverse problem of Eq. (2.8), this is called the inverse reliability analysis. The optimum point of Eq. (2.11) is also called the MPP and denoted by \mathbf{u}^* . If the constraint function value at the MPP, $g(\mathbf{u}^*)$, is less than zero ($G(\mathbf{X}) < 0$ is defined as safe), then the probabilistic constraint is satisfied for the given target reliability β_t and target probability of failure. The inverse reliability analysis using SORM is much more difficult and has not been developed yet. Moreover, it requires the second order sensitivity. We can compare the difference between the reliability analysis and inverse reliability analysis graphically using Fig. 2.2.

To find the MPP using the inverse reliability analysis with the given target reliability index β_t , several methods have been developed including the mean value (MV) method, advanced mean value (AMV) method [Wu et al., 1990; Wu, 1994], hybrid mean value (HMV) method [Youn et al., 2003], and enhanced hybrid mean value (HMV+) method [Youn et al., 2005a].

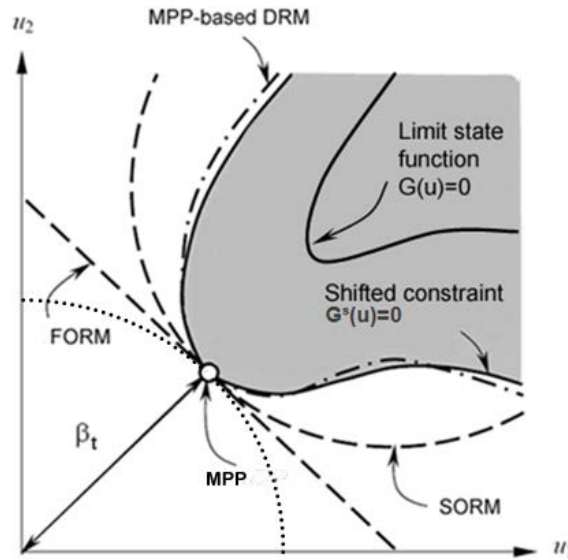


Figure 2. 2 Difference Between Reliability Analysis and Inverse Reliability Analysis

The MV method linearly approximates the performance function using the function and gradient information at the mean value in \mathbf{U} -space as

$$g(\mathbf{U}) \cong g(\boldsymbol{\mu}_{\mathbf{U}}) + \sum_{i=1}^N \frac{\partial g}{\partial U_i} \bigg|_{\mathbf{U}=\boldsymbol{\mu}_{\mathbf{U}}} (U_i - \mu_{U_i}) \quad (2.12)$$

Then, MPP of the inverse reliability analysis using MV can be obtained as

$$\mathbf{u}_{\text{MV}}^* = \beta_t \boldsymbol{\alpha}(\boldsymbol{\mu}_{\mathbf{U}}) \quad (2.13)$$

where $\boldsymbol{\alpha}(\boldsymbol{\mu}_{\mathbf{U}})$ is the normalized gradient vector evaluated at the mean value and written as

$$\boldsymbol{\alpha}(\boldsymbol{\mu}_{\mathbf{U}}) \cong \frac{\nabla_{\mathbf{U}} g(\boldsymbol{\mu}_{\mathbf{U}})}{\|\nabla_{\mathbf{U}} g(\boldsymbol{\mu}_{\mathbf{U}})\|} \quad (2.14)$$

where $\nabla_{\mathbf{U}} = \left\{ \frac{\partial}{\partial U_1}, \dots, \frac{\partial}{\partial U_N} \right\}^T$. This MV method is a crude method to find MPP of the inverse reliability analysis.

However, since it does not require further function evaluation and sensitivity analysis, MPP by MV method can be a good approximation to judge which constraint is active or not when a constraint function is far from the design point.

The MPP obtained by the MV method can be considered as the first iteration of the AMV method. AMV uses the gradient at MPP obtained by the MV method to find the next MPP candidate and the iteration will continue to perform until the approximate MPP converges to the correct MPP. Hence, the AMV method can be formulated as

$$\mathbf{u}_{AMV}^{(1)} = \mathbf{u}_{MV}^*, \quad \mathbf{u}_{AMV}^{(k+1)} = \beta_t \boldsymbol{\alpha}(\mathbf{u}_{AMV}^{(k)}) \quad (2.15)$$

This AMV method is known as an efficient method when the constraint function is convex. A constraint function is defined as convex around the MPP if FORM-based reliability analysis underestimates the probability of failure and vice versa for concave. For example, the constraint function in Fig. 2.2 is concave around the MPP since FORM-based reliability analysis overestimates the probability of failure.

To resolve the weakness of AMV for a concave function, the HVM method uses AMV method when a constraint function is convex and the conjugate mean value (CMV) (Youn et al., 2003) method when a constraint function is concave. HVM+ method uses an interpolation between two previous MPP candidate points if the constraint function is concave instead of using the CMV method.

2.4 Most Probable Point (MPP) Based RBDO

2.4.1 MPP-Based RBDO Using FORM

In general, the RBDO model can be formulated to

$$\begin{aligned}
& \text{minimize} && \text{Cost}(\mathbf{d}) \\
& \text{subject to} && P[G_i(\mathbf{X}) > 0] \leq P_{F_i}^{\text{Tar}}, \quad i = 1, \dots, nc \\
& && \mathbf{d}^L \leq \mathbf{d} \leq \mathbf{d}^U, \quad \mathbf{d} \in \mathbb{R}^{ndv} \text{ and } \mathbf{X} \in \mathbb{R}^{nrv}
\end{aligned} \tag{2.16}$$

where $\mathbf{d} = \{d_i\}^T = \boldsymbol{\mu}(\mathbf{X})$ is the design vector; $\mathbf{X} = \{X_i\}^T$ is the random vector; and nc , ndv and nrv are the number of probabilistic constraints, design variables, and random variables, respectively. Using the inverse reliability analysis, the i th probabilistic constraint can be rewritten as

$$P[G_i(\mathbf{X}) > 0] - P_{F_i}^{\text{Tar}} \leq 0 \Rightarrow G_i(\mathbf{x}^*) \leq 0 \tag{2.17}$$

where $G_i(\mathbf{x}^*)$ is the i^{th} probabilistic constraint evaluated at the MPP \mathbf{x}^* in \mathbf{X} -space.

Using FORM, Eq. (2.16) can be reformulated to

$$\begin{aligned}
& \text{minimize} && \text{Cost}(\mathbf{d}) \\
& \text{subject to} && P[G_i(\mathbf{X}) > 0] \leq P_{F_i}^{\text{Tar}} = \Phi(-\beta_i), \quad i = 1, \dots, nc \\
& && \mathbf{d}^L \leq \mathbf{d} \leq \mathbf{d}^U, \quad \mathbf{d} \in \mathbb{R}^{ndv} \text{ and } \mathbf{X} \in \mathbb{R}^{nrv}
\end{aligned} \tag{2.18}$$

where β_i is the target reliability index for the i th constraint and the probabilistic constraint can be changed into

$$P[G_i(\mathbf{X}) > 0] - \Phi(-\beta_i) \leq 0 \Rightarrow G_i(\mathbf{x}_{\text{FORM}}^*) \leq 0 \tag{2.19}$$

where $\mathbf{x}_{\text{FORM}}^*$ is the FORM-based MPP which can be obtained by solving Eq. (2.11) and transformation $T^{-1}(\mathbf{u}^*) = \mathbf{x}^*$ in Eq. (2.7). For the simplicity, \mathbf{x}^* means the FORM-based MPP hereafter.

To solve the formulation in Eq. (2.18), it is required to calculate the sensitivity of the probabilistic constraint in Eq. (2.19) with respect to a design parameter $d_i = \mu(X_i)$. The sensitivity of the probabilistic constraint with respect to the design parameter is written using the chain rule as

$$\frac{\partial G(\mathbf{x}^*)}{\partial \mathbf{d}} = \frac{\partial G}{\partial \mathbf{d}} \Big|_{\mathbf{x}=\mathbf{x}^*} = \sum_{i=1}^N \frac{\partial G}{\partial x_i} \Big|_{\mathbf{x}=\mathbf{x}^*} \frac{\partial x_i}{\partial \mathbf{d}} \Big|_{\mathbf{x}=\mathbf{x}^*} = \left[\frac{\partial \mathbf{x}}{\partial \mathbf{d}} \right]_{\mathbf{x}=\mathbf{x}^*}^T \frac{\partial G}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\mathbf{x}^*} \quad (2.20)$$

and Eq. (2.20) can be further simplified as (Gumbert et al., 2003; Hou, 2004)

$$\frac{\partial G(\mathbf{x}^*)}{\partial \mathbf{d}} = \left[\frac{\partial \mathbf{x}}{\partial \mathbf{d}} \right]_{\mathbf{x}=\mathbf{x}^*}^T \frac{\partial G}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\mathbf{x}^*} = \frac{\partial G}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\mathbf{x}^*} \quad (2.21)$$

2.4.2 MPP-Based RBDO Using Dimension Reduction

Method (DRM)

The dimension reduction method [Xu and Rahman, 2004a; Xu and Rahman, 2004b] is a newly developed technique to approximate the multi-dimensional integration of a performance function using a function with reduced dimension. There are several DRMs depending on the level of dimension reduction: (1) univariate dimension reduction, which is an additive decomposition of N -dimensional performance function into one-dimensional functions; (2) bivariate dimension reduction, which is an additive decomposition of N -dimensional performance function into at most two-dimensional functions; (3) multivariate dimension reduction, which is an additive decomposition of N -dimensional performance function into at most S -dimensional functions, where $S \leq N$.

In the univariate DRM, which is most commonly used, an N -dimensional performance function $G(\mathbf{X})$ can be additively decomposed into one-dimensional functions at the MPP of the random vector \mathbf{X} as

$$G(\mathbf{X}) \cong \hat{G}(\mathbf{X}) \equiv \sum_{i=1}^N G(x_1^*, \dots, x_{i-1}^*, X_i, x_{i+1}^*, \dots, x_N^*) - (N-1)G(\mathbf{x}^*) \quad (2.22)$$

where $\mathbf{x}^* = \{x_1^*, x_2^*, \dots, x_N^*\}^T$ is the FORM-based MPP of the performance function $G(\mathbf{X})$ obtained from Eq. (2.11) and N is the number of random variables. For example, if $G(\mathbf{X}) = G(X_1, X_2)$ with $N = 2$, then the univariate additive decomposition of $G(\mathbf{X})$ is

$$G(\mathbf{X}) \cong \hat{G}(\mathbf{X}) \equiv G(X_1, x_2^*) + G(x_1^*, X_2) - G(x_1^*, x_2^*) \quad (2.23)$$

This MPP-based univariate DRM is used for more accurate reliability analysis than FORM later in this work for comparison study purpose.

2.5 Sampling-Based RBDO

In the MPP-based RBDO, the probability of failure of the performance function is approximated by FORM, SORM or DRM. For highly nonlinear problems, this approximation may not be accurate enough and lead to inaccurate probability calculation. Therefore, Lee et al., [Lee et al., 2010] proposed another sampling-based RBDO, which uses the score function by Monte Carlo Simulation to calculate the probability of failure and the sensitivity of the probabilistic constraint in RBDO.

2.5.1 Reliability and Statistical Moments

In this sampling-based RBDO, the reliability analysis, for both the component and the system level, involves calculation of the probability of failure, denoted by P_F , which is defined using a multi-dimensional integral

$$P_F(\boldsymbol{\psi}) \equiv P[\mathbf{X} \in \Omega_F] = \int_{\mathbf{R}^N} I_{\Omega_F}(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}; \boldsymbol{\psi}) d\mathbf{x} = E[I_{\Omega_F}(\mathbf{X})] \quad (2.24)$$

where $\boldsymbol{\psi}$ is a vector of distribution parameters, which usually includes the mean ($\boldsymbol{\mu}$) and/or standard deviation ($\boldsymbol{\sigma}$) of the random input $\mathbf{X} = \{X_1, \dots, X_N\}^T$, $P[\bullet]$ represents a probability measure, Ω_F is the failure set, $f_{\mathbf{X}}(\mathbf{x}; \boldsymbol{\psi})$ is a joint probability density function

(PDF) of \mathbf{X} , and $E[\bullet]$ represents the expectation operator. The failure set is defined as $\Omega_F \equiv \{\mathbf{x} : G_i(\mathbf{x}) > 0\}$ for component reliability analysis of the i^{th} constraint function $G_i(\mathbf{x})$, $\Omega_F \equiv \{\mathbf{x} : \bigcup_{i=1}^{NC} G_i(\mathbf{x}) > 0\}$ and $\Omega_F \equiv \{\mathbf{x} : \bigcap_{i=1}^{NC} G_i(\mathbf{x}) > 0\}$ for series system and parallel system reliability analysis of NC performance functions, respectively. $I_{\Omega_F}(\mathbf{x})$ in Eq. (2.24) is called an indicator function and defined as

$$I_{\Omega_F}(\mathbf{x}) \equiv \begin{cases} 1, & \mathbf{x} \in \Omega_F \\ 0, & \text{otherwise} \end{cases} \quad (2.25)$$

In this work, since the mean of \mathbf{X} , $\boldsymbol{\mu} = \{\mu_1, \dots, \mu_N\}^T$, is used as a design vector, the vector of distribution parameters $\boldsymbol{\psi}$ is simply replaced with $\boldsymbol{\mu}$ for the derivation of the sensitivity.

In a fashion similar to Eq. (2.24), the q^{th} statistical moment of a performance function $H(\mathbf{x})$ is defined as

$$m_q(\boldsymbol{\mu}) \equiv E[H^q(\mathbf{X})] = \int_{\mathbf{R}^N} H^q(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}; \boldsymbol{\mu}) d\mathbf{x} \quad (2.26)$$

Thus, Eqs. (2.24) and (2.26) can be written in a generalized form as

$$h(\boldsymbol{\mu}) \equiv E[g(\mathbf{X})] = \int_{\mathbf{R}^N} g(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}; \boldsymbol{\mu}) d\mathbf{x} \quad (2.27)$$

which is called a probabilistic response. In Eq. (2.27), $h(\boldsymbol{\mu})$ and $g(\mathbf{x})$ represent $P_F(\boldsymbol{\mu})$ and $I_{\Omega_F}(\mathbf{x})$, respectively, for reliability analysis.

2.5.2 Stochastic Sensitivity Analysis

The sensitivity of the probabilistic response $h(\boldsymbol{\mu})$ with respect to μ_i is considered in this section. For the derivation of the sensitivity, the following four assumptions, which are known as the regularity conditions, are required.

- The joint PDF $f_{\mathbf{x}}(\mathbf{x}; \boldsymbol{\mu})$ is continuous.
- The mean $\mu_i \in M_i \subset \mathbf{R}$, $i = 1, \dots, N$, where M_i is an open interval on \mathbf{R} .
- The partial derivative $\frac{\partial f_{\mathbf{x}}(\mathbf{x}; \boldsymbol{\mu})}{\partial \mu_i}$ exists and is finite for all \mathbf{x} and μ_i . In addition, $h(\boldsymbol{\mu})$ is a differentiable function of $\boldsymbol{\mu}$.
- There exists a Lebesgue integrable dominating function $r(\mathbf{x})$ such that

$$\left| g(\mathbf{x}) \frac{\partial f_{\mathbf{x}}(\mathbf{x}; \boldsymbol{\mu})}{\partial \mu_i} \right| \leq r(\mathbf{x}) \quad (2.28)$$

for all $\boldsymbol{\mu}$.

With the four assumptions satisfied, taking the partial derivative of Eq. (2.27)

with respect to μ_i yields

$$\frac{\partial h(\boldsymbol{\mu})}{\partial \mu_i} = \frac{\partial}{\partial \mu_i} \int_{\mathbf{R}^N} g(\mathbf{x}) f_{\mathbf{x}}(\mathbf{x}; \boldsymbol{\mu}) d\mathbf{x} \quad (2.29)$$

and the differential and integral operators can be interchanged due to assumption 4 and the Lebesgue dominated convergence theorem, giving

$$\begin{aligned} \frac{\partial h(\boldsymbol{\mu})}{\partial \mu_i} &= \int_{\mathbf{R}^N} g(\mathbf{x}) \frac{\partial f_{\mathbf{x}}(\mathbf{x}; \boldsymbol{\mu})}{\partial \mu_i} d\mathbf{x} \\ &= \int_{\mathbf{R}^N} g(\mathbf{x}) \frac{\partial \ln f_{\mathbf{x}}(\mathbf{x}; \boldsymbol{\mu})}{\partial \mu_i} f_{\mathbf{x}}(\mathbf{x}; \boldsymbol{\mu}) d\mathbf{x} \\ &= E \left[g(\mathbf{x}) \frac{\partial \ln f_{\mathbf{x}}(\mathbf{x}; \boldsymbol{\mu})}{\partial \mu_i} \right] \end{aligned} \quad (2.30)$$

since $g(\mathbf{x})$ is not a function of μ_i . The partial derivative of the log function of the joint PDF in Eq. (2.30) with respect to μ_i is known as the first-order score function for μ_i and is denoted as

$$s_{\mu_i}^{(1)}(\mathbf{x}; \boldsymbol{\mu}) \equiv \frac{\partial \ln f_{\mathbf{x}}(\mathbf{x}; \boldsymbol{\mu})}{\partial \mu_i}. \quad (2.31)$$

Therefore, it is required to know the first-order score function to derive the sensitivity of the probabilistic response, which is either the reliability or the statistical moments. The derivation of the first-order score function for independent and correlated random variables will be shown in the subsequent section.

2.5.3 Formulation of Sampling-Based RBDO

The mathematical formulation of a general RBDO problem is expressed as

$$\begin{aligned} & \text{minimize} && \text{Cost}(\mathbf{d}) \\ & \text{subject to} && P[G_i(\mathbf{X}) > 0] \leq P_{F_i}^{\text{Tar}}, \quad i = 1, \dots, NC \\ & && \mathbf{d}^L \leq \mathbf{d} \leq \mathbf{d}^U, \quad \mathbf{d} \in \mathbb{R}^{ndv} \text{ and } \mathbf{X} \in \mathbb{R}^{nrv} \end{aligned} \quad (2.32)$$

where $\mathbf{d} = \{d_i\}^T = \boldsymbol{\mu}(\mathbf{X})$ is the design vector, which is the mean value of the N -dimensional random vector $\mathbf{X} = \{X_1, X_2, \dots, X_N\}^T$; $P_{F_i}^{\text{Tar}}$ is the target probability of failure for the i^{th} constraint; and NC , ndv , and nrv are the number of probabilistic constraints, design variables, and random variables, respectively.

Then, by applying MCS to the constraint function $G_i(\mathbf{X})$, the probabilistic constraints in Eq. (2.32) can be approximated as

$$P_{F_i} \equiv P[G_i(\mathbf{X}) > 0] \cong \frac{1}{K} \sum_{k=1}^K I_{\Omega_F}(\mathbf{x}^{(k)}) \leq P_{F_i}^{\text{Tar}} \quad (2.33)$$

where K is the MCS sample size, $\mathbf{x}^{(k)}$ is the k^{th} realization of \mathbf{X} , and the failure set Ω_F for the constraint is defined as $\Omega_F \equiv \{\mathbf{x}; G_i(\mathbf{x}) > 0\}$. Sensitivity of the probabilistic constraint in Eq. (2.32) is obtained using the score function as

$$\frac{\partial P_{F_i}}{\partial \mu_i} \cong \frac{1}{K} \sum_{k=1}^K I_{\Omega_F}(\mathbf{x}^{(k)}) s_{\mu_i}^{(1)}(\mathbf{x}^{(k)}; \boldsymbol{\mu}) \quad (2.34)$$

where $s_{\mu_i}^{(1)}(\mathbf{x}^{(k)}; \boldsymbol{\mu})$ is obtained using Eq. (2.31).

As shown in Eq. (2.34), the sensitivity calculation using the score function and MCS does not require the sensitivity of the constraint function, which is known to be extremely difficult to obtain in a large-scale engineering application. Furthermore, the computation of the sensitivity using the score function does not include any approximation except the statistical noise due to MCS, which can be avoided using a sufficiently large MCS sample size. In addition, this sensitivity analysis does not require the transformation from the original design space to the standard normal space, which usually makes the performance function become highly nonlinear, especially when the random input follows non-Gaussian marginal distribution and is correlated. Therefore, the sensitivity analysis using the score function and MCS is very accurate and computationally efficient for engineering applications with correlated random input.

CHAPTER III

SURROGATE MODELING METHODS

3.1 Introduction

Surrogate modeling has been widely used in engineering to approximate the compute-intensive simulation model for design purpose. Over the last several decades, researchers have developed a number of methods for surrogate model generation. Each of these methods has its own advantage and disadvantage with respect to a number of issues, such as the accuracy, efficiency and ease of use. This chapter first reviews the most popular existing methods of surrogate modeling and later proposes a new surrogate modeling method that improves the accuracy of Kriging method. In the last, three numerical examples are used to compare the accuracy between the proposed method and other existing methods.

3.2 Existing Surrogate Modeling Methods

3.2.1 Polynomial Response Surface Method

The classic polynomial response surface (PRS) method is the original and the most widely used form of surrogate model in engineering design. According to the most popular texts [Box and Draper, 1987], a polynomial response surface of order p of a function $y(x)$ of dimension $k=1$ with samples $\{x^{(1)}, x^{(2)}, \dots, x^{(n)}\}$ and $\mathbf{y} = (y^{(1)}, y^{(2)}, \dots, y^{(n)})$, can be written as

$$\hat{y}(x) = a_0 + a_1x + a_2x^2 + \dots + a_px^p = \sum_{i=0}^p a_i x^i \quad (3.1)$$

The coefficients $\mathbf{a} = (a_0, a_1, \dots, a_p)^T$ are estimated by the least squares solutions of $\Phi \mathbf{a} = \mathbf{y}$

where Φ is the Vandermonde matrix

$$\Phi = \begin{pmatrix} 1 & x^{(1)} & (x^{(1)})^2 & \dots & (x^{(1)})^p \\ 1 & x^{(2)} & (x^{(2)})^2 & \dots & (x^{(2)})^p \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x^{(n)} & (x^{(n)})^2 & \dots & (x^{(n)})^p \end{pmatrix} \quad (3.2)$$

and \mathbf{y} is the vector of response at the samples. The least square solution, which is equivalent to the maximum likelihood estimate of \mathbf{a} is thus

$$\mathbf{a} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{y} \quad (3.3)$$

The highest order m is usually obtained by minimizing

$$\sigma_p^2 = \frac{MSE}{n - p - 1} \quad (3.4)$$

where $MSE = \sum_{i=1}^n (y(x^{(i)}) - \hat{y}(x^{(i)}))^2$ is the so called mean squared error (MSE).

3.2.2 Radial Basis Function Method

Radial basis functions (RBF) [Broomhead and Loewe, 1988] use a weighted sum of simple functions to approximate the true response. In RBF, consider n samples $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}\}$ with n response $\mathbf{y} = (y^{(1)}, y^{(2)}, \dots, y^{(n)})$, where $\mathbf{x} \in \mathbb{R}^m$. The approximation of the response is under the form as

$$\hat{y}(\mathbf{x}) = \mathbf{w}^T \Psi = \sum_{i=1}^{n_c} w_i \psi(\|\mathbf{x} - \mathbf{c}^{(i)}\|) \quad (3.5)$$

where $\mathbf{c}^{(i)}$ denotes the i th of n_c basis function center and $\boldsymbol{\psi}$ is an n_c vector containing the values of the basis function $\psi(\cdot)$ evaluated at the L_2 norm of $\mathbf{x} - \mathbf{c}^{(i)}$. Indeed, RBF is recognized as a single-layer neural network with radial coordinate neurons, featuring an input \mathbf{x} , hidden units $\psi(\cdot)$, weights \mathbf{w} , linear output transfer functions and output $\hat{y}(\mathbf{x})$. The basis function can take simple forms as: linear $\psi(r) = r$, cubic $\psi(r) = r^3$ and thin plate spline $\psi(r) = r^2 \ln r$. Moreover, more complex parametric forms are also applicable, such as: Gaussian $\psi(r) = e^{-r^2/2\sigma^2}$, multi-quadric $\psi(r) = (r^2 + \sigma^2)^{1/2}$ and inverse multi-quadric $\psi(r) = (r^2 + \sigma^2)^{-1/2}$. With the basis function setup, the weight \mathbf{w} is obtained by solving the linear equations, as

$$\mathbf{w}^T \boldsymbol{\Psi} = \mathbf{y} \quad (3.6)$$

where $\boldsymbol{\Psi}$ is the so-called Gram matrix and it is defined as $\Psi_{ij} = \psi(\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|)$.

3.2.3 Support Vector Regression Method

Support vector regression (SVR) origins from support vector machines (SVM) in AT&T Bell laboratories. The key attribute of SVR is that it allows us to specify or calculate a margin (ε) within which we are willing to accept errors in the sample data without affecting the surrogate prediction. In SVR, consider n samples $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}\}$ with n response $\mathbf{y} = (y^{(1)}, y^{(2)}, \dots, y^{(n)})$, where $\mathbf{x} \in \mathbf{R}^m$. The prediction of the response is defined as

$$\hat{y}(\mathbf{x}) = \mu + \langle \mathbf{w}, \mathbf{x} \rangle \quad (3.7)$$

where $\langle \cdot, \cdot \rangle$ denotes the dot product, μ is a constant and \mathbf{w} is the weight needs to be determined. To produce a prediction which generalizes well, it is desirable to find a function with at most ε deviation from \mathbf{y} and at the same time minimize the model complexity $|\mathbf{w}|^2$. Therefore, the problem becomes

$$\begin{aligned} & \text{minimize: } \frac{1}{2} |\mathbf{w}|^2 \\ & \text{s.t. } \quad -\varepsilon \leq y^{(i)} - \langle \mathbf{w}, \mathbf{x}^{(i)} \rangle - \mu \leq \varepsilon \end{aligned} \quad (3.8)$$

The assumption behind the constraints is that it demands that a function $\hat{y}(\mathbf{x})$ exists which approximates all $y^{(i)}$ with precision $\pm\varepsilon$. Such a solution may not actually exist and it is also likely that better predictions will be obtained if we allow for the possibility of outliers. This is achieved by introducing slack variables, ξ^+ for $\hat{y}(\mathbf{x}^{(i)}) - y(\mathbf{x}^{(i)}) > \varepsilon$ and ξ^- for $y(\mathbf{x}^{(i)}) - \hat{y}(\mathbf{x}^{(i)}) > \varepsilon$. Thus, the problem becomes

$$\begin{aligned} & \text{minimize: } \frac{1}{2} |\mathbf{w}|^2 + C \frac{1}{n} \sum_{i=1}^n (\xi^{+(i)} + \xi^{-(i)}) \\ & \text{s.t. } \quad \begin{cases} y^{(i)} - \langle \mathbf{w}, \mathbf{x}^{(i)} \rangle - \mu \leq \varepsilon + \xi^{+(i)}, \\ \langle \mathbf{w}, \mathbf{x}^{(i)} \rangle + \mu - y^{(i)} \leq \varepsilon + \xi^{-(i)}, \\ \xi^{+(i)}, \xi^{-(i)} \geq 0 \end{cases} \end{aligned} \quad (3.9)$$

The constrained problem of Eq. (3.9) is solved by introducing the Lagrange multipliers, $\eta^{+(i)}$, $\eta^{-(i)}$, $\alpha^{+(i)}$ and $\alpha^{-(i)}$ to give the Lagrangian,

$$\begin{aligned} L = & \frac{1}{2} |\mathbf{w}|^2 + C \frac{1}{n} \sum_{i=1}^n (\xi^{+(i)} + \xi^{-(i)}) - \sum_{i=1}^n (\eta^{+(i)} \xi^{+(i)} + \eta^{-(i)} \xi^{-(i)}) \\ & - \sum_{i=1}^n a^{+(i)} (\varepsilon + \xi^{+(i)} - y^{(i)} + \langle \mathbf{w}, \mathbf{x}^{(i)} \rangle + \mu) \\ & - \sum_{i=1}^n a^{-(i)} (\varepsilon + \xi^{-(i)} + y^{(i)} - \langle \mathbf{w}, \mathbf{x}^{(i)} \rangle - \mu) \end{aligned} \quad (3.10)$$

By solving $\frac{\partial L}{\partial \mathbf{w}} = 0$, $\frac{\partial L}{\partial \mu} = 0$, $\frac{\partial L}{\partial \xi^+} = 0$, $\frac{\partial L}{\partial \xi^-} = 0$, we obtain

$$\mathbf{w} = \sum_{i=1}^n (\alpha^{+(i)} - \alpha^{-(i)}) \mathbf{x}^{(i)} \quad (3.11)$$

and therefore the prediction is

$$y(\mathbf{x}) = \mu + \sum_{i=1}^n (\alpha^{+(i)} - \alpha^{-(i)}) \langle \mathbf{x}, \mathbf{x}^{(i)} \rangle \quad (3.12)$$

In the above procedure, the $\langle \cdot, \cdot \rangle$ can be replaced by a generalized kernel function

$\psi(\mathbf{x}_i, \mathbf{x}_j)$, and popular choice for ψ are:

$$\begin{aligned} \psi(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) &= (\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) && \text{(linear)} \\ \psi(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) &= (\mathbf{x}^{(i)}, \mathbf{x}^{(j)})^d && (d \text{ degree homogeneous polynomial}) \\ \psi(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) &= (\mathbf{x}^{(i)}, \mathbf{x}^{(j)} + c)^d && (d \text{ degree inhomogeneous polynomial}) \\ \psi(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) &= \exp\left(\frac{-|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}|^2}{\sigma^2}\right) && \text{(Gaussian)} \end{aligned} \quad (3.13)$$

3.2.4 Universal Kriging Method

The Kriging method has gained large interest for constructing the surrogate model in recent years. In the Kriging method, the outcomes are considered as a realization of a stochastic process. Consider n samples $\mathbf{x} = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)})^T$ with n response $\mathbf{y} = (y^{(1)}, y^{(2)}, \dots, y^{(n)})^T$, where $\mathbf{x} \in \mathbf{R}^m$. In the Kriging method, the response at the samples consists of a summation of two parts as

$$\mathbf{y} = \mathbf{F}\boldsymbol{\beta} + \mathbf{e} \quad (3.14)$$

The first part of the right-hand side of Eq. (3.14), $\mathbf{F}\boldsymbol{\beta}$, is the mean structure of the response, where $\mathbf{F} = [f_k(\mathbf{x}^{(i)})]$, $i = 1, \dots, n$, $k = 1, \dots, K$ is an $n \times K$ design matrix, and $f_k(\mathbf{x})$

represent user-defined basis functions, which are usually in a simple polynomial form, such as $1, x, x^2, \dots$. In Eq. (3.14), $\boldsymbol{\beta} = [\beta_1, \beta_2, \dots, \beta_k]^T$ are the regression coefficients from the generalized least square regression method. The second part of the right-hand side of Eq. (3.14), $\mathbf{e} = [e(\mathbf{x}^{(1)}), e(\mathbf{x}^{(2)}), \dots, e(\mathbf{x}^{(n)})]^T$, is a realization of the stochastic process $e(\mathbf{x})$ that is assumed to have zero mean and covariance structure $E[e(\mathbf{x}^{(i)})e(\mathbf{x}^{(j)})] = \sigma(\boldsymbol{\theta})^2 R(\boldsymbol{\theta}, \mathbf{x}^{(i)}, \mathbf{x}^{(j)})$, where $\sigma(\boldsymbol{\theta})^2$ is the process variance, $\boldsymbol{\theta}$ is the process parameter that has to be estimated by applying the maximum likelihood estimator (MLE), and $R(\boldsymbol{\theta}, \mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ is the correlation function of the stochastic process. Usually in engineering problems the correlation function is set to Gaussian form, expressed as

$$R(\boldsymbol{\theta}, \mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \prod_{k=1}^m \exp(-\theta_k (\mathbf{x}_k^{(i)} - \mathbf{x}_k^{(j)})^2) \quad (3.15)$$

where $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_m)$ and $\mathbf{x}_k^{(i)}$ is the k^{th} dimension of variable $\mathbf{x}^{(i)}$. The optimal choice of $\boldsymbol{\theta}$ is defined as the maximum likelihood estimator (MLE), which is the maximizer of the likelihood function, expressed as

$$\min \psi(\boldsymbol{\theta}) = |\mathbf{R}(\boldsymbol{\theta})|^{1/n} \sigma(\boldsymbol{\theta})^2 \quad (3.16)$$

where \mathbf{R} is the symmetric correlation matrix with i - j^{th} component

$$(R)_{ij} = R(\boldsymbol{\theta}, \mathbf{x}^{(i)}, \mathbf{x}^{(j)}), \quad i, j = 1, \dots, n \quad \text{and} \quad \sigma(\boldsymbol{\theta})^2 = \frac{1}{n} (\mathbf{Y} - \mathbf{F}\boldsymbol{\beta})^T \mathbf{R}^{-1} (\mathbf{Y} - \mathbf{F}\boldsymbol{\beta}).$$

Under the general decomposition of Eq. (3.14) and the obtained $\boldsymbol{\theta}$ from Eq. (3.16), the objective is to predict the noise-free unbiased response at a new point of interest \mathbf{x} . In the Kriging method, this prediction of response is written as a linear predictor as

$$\hat{y}(\mathbf{x}) = \mathbf{w}^T \mathbf{y} \quad (3.17)$$

where $\mathbf{w} = [w_1(\mathbf{x}), w_2(\mathbf{x}), \dots, w_n(\mathbf{x})]^T$ denotes the $n \times 1$ weight vector for prediction at \mathbf{x} .

Using Eq. (3.17), the unbiased prediction condition $E[\hat{y}(\mathbf{x})] = E[y(\mathbf{x})]$ is expressed as

$$\begin{aligned}
 E[\hat{y}(\mathbf{x}) - y(\mathbf{x})] &= E[\mathbf{w}^T \mathbf{y} - y(\mathbf{x})] \\
 &= E[\mathbf{w}^T (\mathbf{F}\boldsymbol{\beta} + \mathbf{e}) - (\mathbf{f}\boldsymbol{\beta} + e(\mathbf{x}))] \\
 &= E[\mathbf{w}^T \mathbf{e} - e(\mathbf{x}) + (\mathbf{F}^T \mathbf{w} - \mathbf{f}^T)^T \boldsymbol{\beta}] \\
 &= E[(\mathbf{F}^T \mathbf{w} - \mathbf{f}^T)^T \boldsymbol{\beta}] = \mathbf{0}
 \end{aligned} \tag{3.18}$$

where $\mathbf{f} = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})]$. Therefore, the unbiased condition is ensured by imposing the constraint $\mathbf{F}^T \mathbf{w} = \mathbf{f}^T$ on the prediction weights for each point of interest.

Under this constraint, \mathbf{w} is obtained as

$$\mathbf{w} = \mathbf{R}^{-1}(\mathbf{r} - \mathbf{F}\boldsymbol{\lambda}) \tag{3.19}$$

by solving the Lagrangian first-order necessary conditions of minimizing the mean squared error (MSE) of prediction, where $\mathbf{r} = [R(\boldsymbol{\theta}, \mathbf{x}^{(1)}, \mathbf{x}), \dots, R(\boldsymbol{\theta}, \mathbf{x}^{(n)}, \mathbf{x})]^T$ is the correlation vector between the prediction location \mathbf{x}_0 and all n samples $\mathbf{x}^{(i)}$, $i=1, \dots, n$.

Hence the prediction is expressed as

$$\begin{aligned}
 \hat{y}(\mathbf{x}) &= \mathbf{w}^T \mathbf{y} = (\mathbf{r} - \mathbf{F}\boldsymbol{\lambda})^T \mathbf{R}^{-1} \mathbf{y} \\
 &= \mathbf{f}^T \boldsymbol{\beta} + \mathbf{r}^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{F}\boldsymbol{\beta})
 \end{aligned} \tag{3.20}$$

Under the assumption of the Gaussian process, the α -level prediction interval of response is written as

$$\hat{y}(\mathbf{x}) - Z_{1-\alpha/2} \sigma(\mathbf{x}) \leq y(\mathbf{x}) \leq \hat{y}(\mathbf{x}) + Z_{1-\alpha/2} \sigma(\mathbf{x}) \tag{3.21}$$

where $Z_{1-\alpha/2}$ is the α -level quantile of standard normal distribution and $\sigma^2(\mathbf{x})$ is the predicted variance at \mathbf{x} . Therefore, the bandwidth of the prediction interval at a point of interest \mathbf{x} is

$$d(\mathbf{x}) = 2Z_{1-\alpha/2} \sigma(\mathbf{x}) \tag{3.22}$$

3.2.5 Blind Kriging Method

Blind Kriging (BKG) is a method by which the basis functions in Kriging model are identified through a Bayesian forward variable selection procedure proposed by Joseph et al [Joseph et al, 2008]. In BKG, the candidate basis functions are set to be linear, quadratic, linear-by-linear, linear-by-quadratic, quadratic-by-linear and quadratic-by-quadratic functions. The linear and quadratic functions are defined using orthogonal polynomial functions, expressed as

$$\begin{aligned} x_{lin,j} &= \frac{\sqrt{3}}{\sqrt{2}} 2(x_j - 0.5) \text{ and} \\ x_{quad,j} &= \frac{1}{\sqrt{2}} [6(x_j - 0.5)^2 - 2] \end{aligned} \quad (3.23)$$

for $j = 1, 2, \dots, m$. To find the most important candidates, we need to find the maximum of the vector

$$\mathbf{\Lambda} = \mathbf{D}\mathbf{U}^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{F}_k\hat{\mathbf{\beta}}_k) \quad (3.24)$$

where \mathbf{D} is a $(2m^2 + 1) \times (2m^2 + 1)$ diagonal matrix.

$$\mathbf{D} = \text{diag}(1, r_{lin,1}, r_{quad,1}, r_{lin,2}, \dots, r_{quad,m-1}, r_{quad,m}) \quad (3.25)$$

and

$$\begin{aligned} r_{lin,j} &= \frac{3 - 3R_j(1)}{3 + 4R_j(0.5) + 2R_j(1)} \text{ and} \\ r_{quad,j} &= \frac{3 - 4R_j(0.5) + R_j(1)}{3 + 4R_j(0.5) + 2R_j(1)} \end{aligned} \quad (3.26)$$

$R_j(r) = \exp(-\theta_j |r|^2)$ and θ_j is the j^{th} value of the correlation parameter $\boldsymbol{\theta}$ and \mathbf{F}_k is a $n \times k$ matrix containing the k basis functions which have been determined. \mathbf{U} is an $n \times (2m^2 + 1)$

matrix whose first column is 1, with subsequent columns given by the interactions of the sample data:

$$\mathbf{U} = \begin{pmatrix} 1 & x_{lin,1}(x_1^{(1)}) & x_{quad,1}(x_1^{(1)}) & x_{lin,2}(x_2^{(1)}) & \dots & x_{quad,m-1}(x_{m-1}^{(1)})x_{quad,m}(x_m^{(1)}) \\ \cdot & \cdot & \cdot & & & \cdot & \cdot \\ \cdot & \cdot & \cdot & & & \cdot & \cdot \\ 1 & x_{lin,1}(x_1^{(n)}) & x_{quad,1}(x_1^{(n)}) & x_{lin,2}(x_2^{(n)}) & \dots & x_{quad,m-1}(x_{m-1}^{(n)})x_{quad,m}(x_m^{(n)}) \end{pmatrix} \quad (3.27)$$

$$\hat{\boldsymbol{\beta}}_k = (\mathbf{F}_k^T \mathbf{R}^{-1} \mathbf{F}_k)^{-1} (\mathbf{F}_k^T \mathbf{R}^{-1} \mathbf{y}) \quad (3.28)$$

where \mathbf{R} is the covariance matrix. In BKG, the correlation parameter $\boldsymbol{\theta}$ is estimated as per ordinary Kriging. Then the Λ is calculated from Eq. (3.24) using $\mathbf{F}_0 = \mathbf{1}$ and $\hat{\boldsymbol{\mu}}_0 = \hat{\mu}$ from ordinary Kriging. Next set $k = 1$ and choose the basis function corresponding to the maximum value of Λ . Then we again computer Λ and now \mathbf{F}_k is an $n \times (k+1)$ matrix whose first column is 1 and m^{th} column is the column of \mathbf{U} corresponding to the index of the maximum value of Λ . Then the BKG correlation parameter $\boldsymbol{\theta}$ is estimated by maximizing the concentrated ln-likelihood

$$\ln(L) \approx -\frac{n}{2} \ln(\hat{\sigma}_k^2) - \frac{1}{2} \ln(|\mathbf{R}|) \quad (3.29)$$

Therefore, the BKG prediction at any new point \mathbf{x} is

$$\hat{y}(\mathbf{x}) = \mathbf{f}(x)^T \hat{\boldsymbol{\beta}}_k + \mathbf{r}^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{F}_k \hat{\boldsymbol{\beta}}_k) \quad (3.30)$$

where $\mathbf{f}(x)$ is the selected basis function vector and $\mathbf{r} = [R(\boldsymbol{\theta}, \mathbf{x}^{(1)}, \mathbf{x}), \dots, R(\boldsymbol{\theta}, \mathbf{x}^{(n)}, \mathbf{x})]$ is the correlation between \mathbf{x} and the sample points.

Predictions are then used to calculate a cross-validation error and the above process iterated to reduce this error up to $t = 2m^2$ times. Joseph et al. stop iterating when the cross-validation error begins to rise consistently and the t basis functions

corresponding to the smallest error. They also arguably note that it is not necessary to estimate θ at every step, just the first and the last.

3.3 Dynamic Kriging Method (DKG)

3.3.1 Using Pattern Search Method for Correlation

Coefficient Estimation

As discussed in Section 1.1.2, the first critical issue that affects the accuracy of the surrogate model generated by Kriging method is the optimization method used to estimate the correlation parameter θ as shown in Eq. (3.16). To show how the optimal θ affects the final accuracy of the Kriging prediction in Eq. (3.20), consider one simple revised example based on Forrester's work1 [Forrester et al., 2009]. In this example, the true function is expressed as

$$y = (6x - 2)^2 \sin(12x - 4) + 10 \quad x \in [0,1] \quad (3.31)$$

and five evenly distributed samples along the x -axis are used to generate the Kriging prediction. Then, the accuracy of the Kriging prediction is tested using different θ , and the relative root mean square error ($rRMSE$) is used as the accuracy measurement. In particular, the $rRMSE$ is defined as

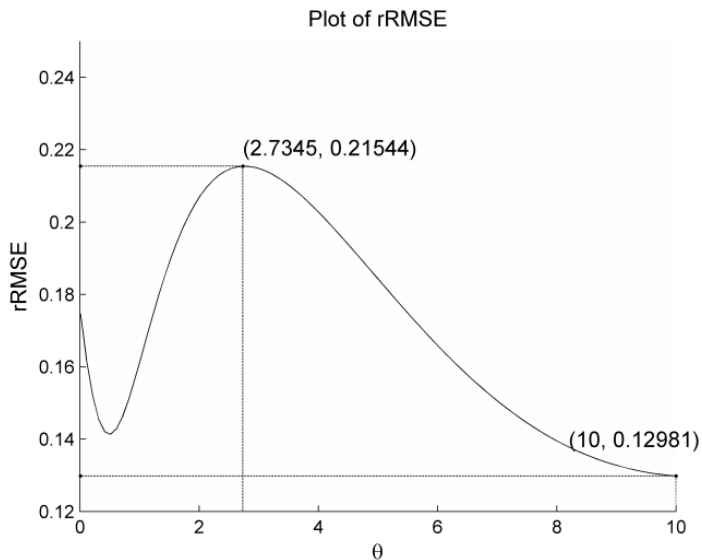
$$rRMSE = \sqrt{\frac{1}{NTS} \sum_{i=1}^{NTS} \left(\frac{\hat{y}(x_i) - y(x_i)}{y(x_i)} \right)^2} \quad (3.32)$$

where NTS is the number of testing points, and $\hat{y}(x_i)$ and $y(x_i)$ are the Kriging prediction and the true response at testing point x_i , respectively. In this example, NTS is 100 and all the testing points are evenly distributed along the x -axis. As shown in Fig. 3.1(a), the $rRMSE$ value changes significantly as θ changes. As θ increases from 2.7345 to 10, the

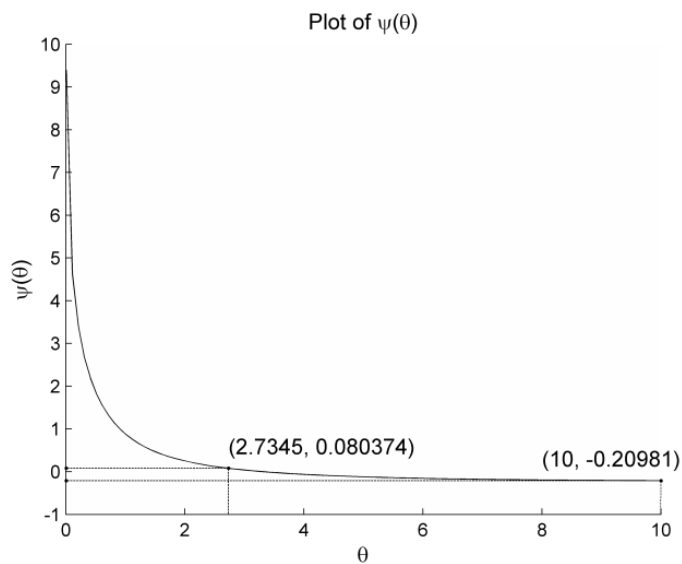
$rRMSE$ decreases from its maximum (0.21544) to the minimum (0.12981), whereas in Fig. 3.1(b), the $\psi(\boldsymbol{\theta})$ function value only changes from 0.080374 to -0.20981 . This behavior shows that it is very important to accurately solve the minimization problem of Eq. (3.16) and find the true optimal $\boldsymbol{\theta}$ to generate an accurate Kriging prediction.

Therefore, to accurately solve Eq. (3.16), it is proposed to use the generalized pattern search (GPS) method. The reason for using the GPS method is that the $\psi(\boldsymbol{\theta})$ function in Eq. (3.16) usually has a highly nonlinear region near the origin and a large flat region elsewhere as shown in Fig. 3.1(b). What is more, the $\psi(\boldsymbol{\theta})$ usually contains multiple local minima for high-dimension problems. For such a minimization problem, a gradient-based optimization algorithm often prematurely converges to a local minimum if the initial $\boldsymbol{\theta}$ value is close to the origin; or prematurely stops in the large flat region if the initial $\boldsymbol{\theta}$ value is close to the upper bound of the $\boldsymbol{\theta}$ domain. Among all the non-gradient-based optimization algorithms, the GA is considered time-consuming and unreliable for such a continuous optimization problem, whereas the GPS method is not affected by the initial search point and accurately converges to the optimal $\boldsymbol{\theta}$. The global convergence of the GPS method has been proven by Lewis and Torczon [Lewis and Torczon, 1999]. It is worth mentioning that since the $\psi(\boldsymbol{\theta})$ is always highly nonlinear near the origin and large flat elsewhere, the initial search point is set to be the lower bound of the $\boldsymbol{\theta}$ domain for the optimization method to converge quickly.

In the GPS method, a pattern is a set of vectors $\{\mathbf{v}_i\}$ that the pattern search algorithm uses to determine which points to search at each iteration. The set $\{\mathbf{v}_i\}$ is defined by the number of independent variables in the objective function $\psi(\boldsymbol{\theta})$, and the positive basis set.



(a) Plot of $rRMSE$ for Kriging Prediction Using Different θ



(b) Plot of $\psi(\theta)$ in Kriging Model Using Different θ

Figure 3. 1 The Effect of Different θ for the Accuracy of Kriging Prediction

Two commonly used positive basis sets in pattern search algorithms are the maximal basis, with $2m$ vectors, and the minimal basis, with $m + 1$ vectors.

The collections of vectors that form the pattern are fixed-direction vectors. For example, if there are three independent variables in the optimization problem, the default for a $2m$ positive basis consists of the following pattern vectors:

$$\begin{aligned} \mathbf{v}_1 &= [1 \ 0 \ 0] & \mathbf{v}_2 &= [0 \ 1 \ 0] & \mathbf{v}_3 &= [0 \ 0 \ 1] \\ \mathbf{v}_4 &= [-1 \ 0 \ 0] & \mathbf{v}_5 &= [0 \ -1 \ 0] & \mathbf{v}_6 &= [0 \ 0 \ -1] \end{aligned} \quad (3.33)$$

An $m+1$ positive basis consist of the following default pattern vector:

$$\mathbf{v}_1 = [1 \ 0 \ 0] \quad \mathbf{v}_2 = [0 \ 1 \ 0] \quad \mathbf{v}_3 = [0 \ 0 \ 1] \quad \mathbf{v}_4 = [-1 \ -1 \ -1] \quad (3.34)$$

At each step, the pattern search method searches a set of points, called a mesh, for a point that improves the objective function $\psi(\boldsymbol{\theta})$. The pattern search method forms the mesh in the following way:

- Generating a set of vectors $\{\mathbf{d}_i\}$ by multiplying each pattern vector \mathbf{v}_i by a scalar Δ . Δ is called the mesh size.
- Adding the $\{\mathbf{d}_i\}$ to the current point – the point with the best objective function value found at the previous step.

Then pattern search algorithm will polls the points in the current mesh by computing their objective function values. Then it compares the mesh point with the smallest objective function value to the current point. If that mesh point has a smaller value than the current point, the poll is successful. After polling, the algorithm changes the value of the mesh size Δ . It multiplies Δ by 2 after a successful poll and by 0.5 after an unsuccessful poll. The pattern search continues on until any one of the convergence

criteria, which are the change of mesh size, change of best objective function value and number of iteration, is achieved.

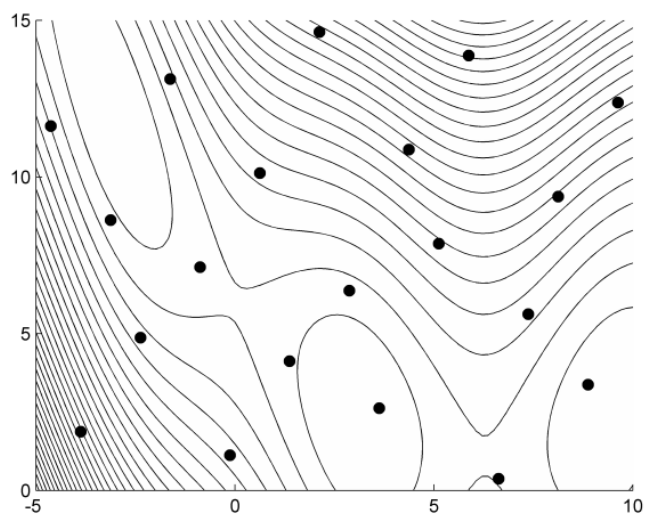
To more clearly demonstrate the challenge in this minimization problem and how the pattern search works, an illustrative example, which is Branin-Hoo problem

$$f(x_1, x_2) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10 \quad (3.35)$$

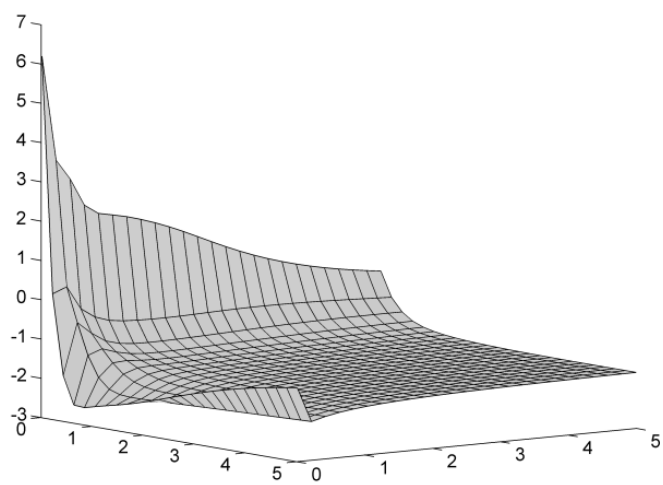
$$x_1 \in [-5, 10], x_2 \in [0, 15]$$

is used. With 20 Latin hypercube samples generated, the sample profile and the true function contour are first shown in Fig. 3.2(a). The associated $\psi(\boldsymbol{\theta})$ plot can be viewed in Fig. 3.2(b) and it can be seen that $\psi(\boldsymbol{\theta})$ has a sharp corner region near the origin and long flat region in the rest of the feasible domain. The initial search point is set to be the center of the feasible domain.

As shown in Table. 3.1, the four optimization methods that are discussed in the introduction give four different optimum points for $\boldsymbol{\theta}$. Among all four methods, the generalized pattern search method finds the best optimum point which is indeed the global optimum after verification. To verify the effect of accurate $\boldsymbol{\theta}$ estimation on the accuracy of the Kriging surrogate model, 100×100 testing points are used to calculate the $rRMSE$ of the surrogate model. Using the best optimum $\boldsymbol{\theta}$ found by the pattern search method, the associated $rRMSE$ from the Kriging model becomes the smallest among four methods. This result indeed shows that using the generalized pattern search method can achieve the best optimum for $\boldsymbol{\theta}$, and the best Kriging model for surrogate modeling thereafter.



(a) Contour of True Function and Samples



(b) Plot of $\psi(\theta)$ and Different Optimum Results

Figure 3. 2 Branin-Hoo Example

Table 3. 1 Comparison between Four Optimization Methods

	H-J	L-M	GA	GPS
Optimal θ_{opt}	(1.7679, 0.2628)	(0.7897, 0.0108)	(0.7158, 0.0088)	(0.7168, 0.0087)
$\psi(\theta_{\text{opt}})$	0.1169	0.0763	0.0758	0.0750
$rRMSE$	1.9223	0.1072	0.0825	0.0755

To demonstrate performance of the GPS for finding a better optimum of the $\psi(\theta)$ function compared with the other three optimization methods in a general way, a statistical study is conducted using the Branin-Hoo example again. In this statistical study, 100 randomly generated sets of 20-LHS samples are used. For each sample set, the ordinary Kriging is applied to generate the prediction. The four optimization methods discussed above are applied to solve Eq. (3.16), and the $\psi(\theta)$ function values at the optimum of θ are ranked from the smallest to the largest order. After the 100 trials, the frequency of the rank for four methods is shown in Table 3.2, where it shows that the GPS found the best optimal θ 92 times out of 100. Even though it is hard to claim that one optimization algorithm performance is better than the others all the time, to solve this particular bounded constrained problem of Eq. (3.16), it is clearly shown that the GPS method can obtain the best results in finding the accurate optimal θ , while the L-M method and the GA method have comparable performance thereafter.

To show the accuracy of the Kriging models based on different optimal θ values from four optimization methods, 100×100 grid testing points are evaluated to calculate the $rRMSE$ values and the rank of $rRMSE$ values associated with each optimization method from the smallest to the largest is shown in Table 3.3. It shows that the Kriging model using the GPS method achieves a better accuracy than the other three methods. It

is noted that that the difference among the four methods in Table 3.3 is not as significant as the one in Table 3.2.

Table 3. 2 Frequency of Rank of $\psi(\theta)$ Function Value at Optimum θ

	H-J	L-M	GA	GPS
1 st	0	4	4	92
2 nd	0	71	24	5
3 rd	2	23	72	3
4 th	98	2	0	0

Table 3. 3 Frequency of Rank of $rRMSE$ Values by Different Optimization Methods

	H-J	L-M	GA	GPS
1st	0	30	31	39
2nd	3	43	14	40
3rd	13	22	48	17
4th	84	5	7	4

As the dimension of the design variables increases, the difference of using four optimization methods to solve Eq. (3.16) is becoming more significant. Consider a 12-D mathematical example, express as

$$y = (x_i - 1)^2 + \sum_{i=2}^{12} i(2x_i^2 - x_{i-1})^2, \quad -10 \leq x_i \leq 10 \quad (3.36)$$

This function is called Dixon-Price function. With 60 samples generated using Latin hypercube sampling method, the ordinary Kriging prediction using four different optimization methods for finding optimal θ is generated. Table 3.4 shows the optimal θ obtained using four different optimization methods and the associated objective function

values and the $rRMSE$ values. The GPS method finds the best optimal θ with the smallest value and generates the most accurate Kriging prediction.

Table 3. 4 Comparison between Four Optimization Methods

	H-J	L-M	GA	GPS
Optimal θ_{opt}	0.0100	0.0197	0.0306	0.0100
	0.0100	0.0100	0.0185	0.0100
	0.0133	0.2439	0.4174	0.0555
	0.0100	0.0100	0.0275	0.0100
	0.0157	0.2727	0.1990	0.0688
	0.0100	0.0100	0.0950	0.0100
	0.0115	0.1024	0.1316	0.0687
	0.0100	0.0100	0.1988	0.0306
	0.0170	0.1275	0.1524	0.0795
	0.0124	0.0100	0.1859	0.1122
	0.0234	0.1583	0.3100	0.1308
	0.0189	1.1422	0.9055	0.1680
	$\psi(\theta_{opt})$	1.1193	0.7067	0.8201
$rRMSE$	0.2443	0.1552	0.2095	0.1234

Like the previous example, to exclude the effect from the sample position and show the general performance of four optimization methods, 100 randomly generated sets of 60-LHS samples are used. The four optimizations are applied to find the optimal θ values to generate Kriging prediction. The $rRMSE$ values are calculated based on a fixed set of 1000-LHS samples. Table 3.5 shows that the GPS method finds the best optimal θ value in 83 times; and the L-M method finds the best optimal θ in 17 times. The GA method and H-J method fails to find the best optimal θ value. At the same time, the associated $rRMSE$ values are ranked as well, as shown in Table 3.6. The Kriging model with the GPS method generates the most accurate surrogate model in 80 times; followed

by the one with the L-M method of 18 times. Tables 3.5 and 3.6 indeed show that the GPS method outperforms other three optimization methods for this high dimension problem.

Table 3. 5 Frequency of Rank of $\psi(\theta)$ Function Value at Optimum θ

	H-J	L-M	GA	GPS
1 st	0	17	0	83
2 nd	12	16	55	17
3 rd	37	23	40	0
4 th	51	44	5	0

Table 3. 6 Frequency of Rank of $rRMSE$ Values by Different Optimization Methods

	H-J	L-M	GA	GPS
1 st	0	18	2	80
2 nd	16	55	12	17
3 rd	61	20	16	3
4 th	70	7	70	0

3.3.2 Effect of Using Different Basis Function for Kriging

Method

For the UKG method, the basis function $f_k(\mathbf{x})$ in \mathbf{F} which is used in Eq. (3.14) is fixed during the entire surrogate modeling process, and it usually takes up to the second order polynomial. However, it is obvious that higher-order terms can predict nonlinear mean structure, which may vary for different problems. Hence, in general, for highly nonlinear cases, fixed-order basis functions may not be suitable to describe the nonlinearity of the mean structure. On the other hand, Martin and Simpson [Martin and

Simpson, 2005] pointed out that, in some cases, the accuracy of the surrogate model may not be enhanced by using higher-order terms. That is, the surrogate model may become even worse when some particular higher-order terms are used.

The impact of selection of basis functions can be shown using the following illustrative example,

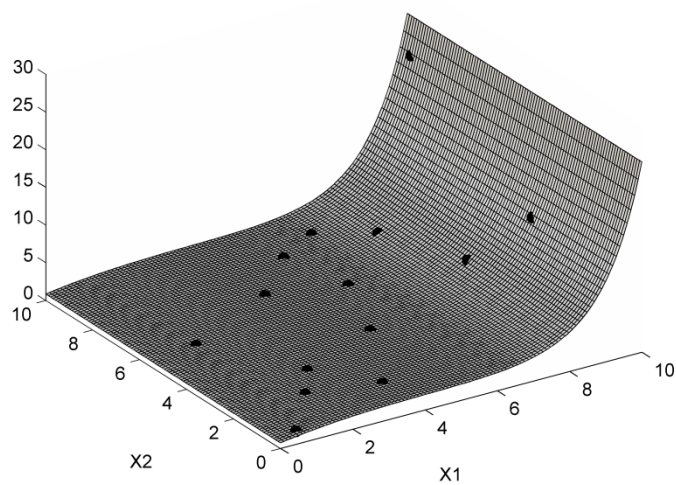
$$f(x_1, x_2) = \frac{(x_1 - 5)^3 + \frac{e^{x_1}}{10} + x_2 + 10}{100} \quad x_1, x_2 \in [0, 10] \quad (3.37)$$

where the true function plot is shown in Fig. 3.3(a).

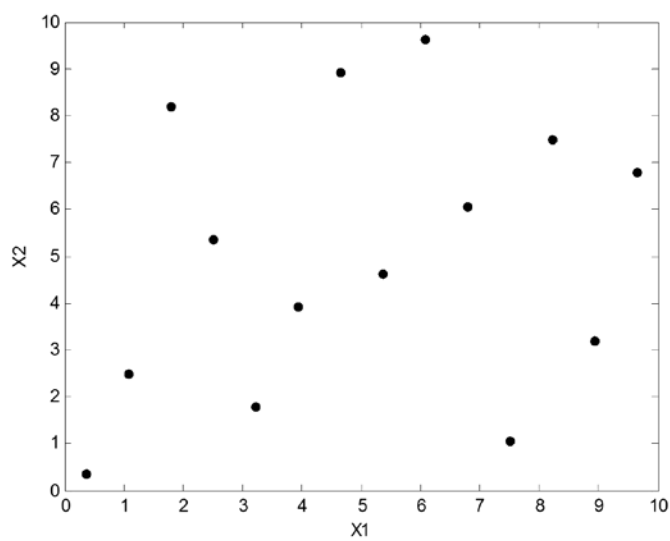
The Kriging method with different basis functions is applied to this problem using the 14 samples obtained from LHS as shown in Fig. 3.3, and the *rRMSE* values that are calculated from 100×100 grid testing points are compared. Table 3.7 shows that the *rRMSE* value decreases from 0th-order to 1st-order polynomials but it increases from 1st-order to 3rd-order polynomials, which implies that increasing order does not improve the accuracy of the surrogate model. Moreover, if unnecessary basis functions are excluded to obtain the customized-order Kriging, the Kriging prediction becomes more accurate.

Table 3. 7 *rRMSE* of Kriging Methods with Different Basis Functions

Kriging Methods	Basis Functions	<i>rRMSE</i>
OKG	1	0.1343
1 st -order UKG	1, x_1, x_2	0.0819
2 nd -order UKG	1, $x_1, x_1, x_1x_2, x_1^2, x_2^2$	0.1137
3 rd -order UKG	1, $x_1, x_2, x_1x_2, x_1^2, x_2^2, x_1^2x_2, x_1x_2^2, x_1^3, x_2^3$	0.1671
Customized-order Kriging	1, x_1, x_1^2, x_1^3, x_2	0.0810



(a). The True Function Surface



(b). 14 Samples

Figure 3. 3 The 2-D Example for Basis Function Selection

3.3.3 Using Genetic Algorithm (GA) for Basis Function

Selection

Based on the effect of the basis function shown in Section 3.3.2, the problem is how to find the optimal subset of the basis functions such that the Kriging prediction would have the best accuracy. The objective is to find a subset of the basis functions such that the obtained Kriging prediction can have the smallest *rRMSE* value. However, since the *rRMSE* value is not available unless the true function is explicitly known, it is proposed to use the Kriging process variance σ^2 as the estimator of the accuracy for the Kriging prediction in this paper. Therefore, the formulation for this problem becomes

$$\begin{aligned} & \text{Find a subset of the basis functions} \\ & \text{to minimize } \sigma^2 = \frac{1}{n} (\mathbf{y} - \mathbf{F}\boldsymbol{\beta})^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{F}\boldsymbol{\beta}) \end{aligned} \quad (3.38)$$

It is noted that different types of the candidate of basis functions, such as Hermit polynomials, trigonometric functions, and exponential functions, have been tested in this study, and it is found that the simple polynomial forms perform efficiently and effectively without losing accuracy. Thus, in this paper, all the candidate basis functions are assumed to be polynomials and in the form of their multiplications $x_1^{p_1} x_2^{p_2} \dots x_m^{p_m}$, where m is the number of design variables, $p_i \in [0, P]$ is an integer power of x_i , $\sum_{i=1}^m p_i \leq P$, and P is the highest order of the mean structure in the Kriging model. The total number of possible candidate basis functions is C_{m+P}^P . Therefore, the full set \mathbf{f} becomes

$$\mathbf{f} = [1, x_1, x_2, \dots, x_m, x_1^2, \dots, x_m^2, x_1 x_2, \dots, x_{m-1} x_m, \dots, x_1^P, x_1^{P-1} x_2, \dots, x_m^P]_{1 \times C_{m+P}^P} \quad (3.39)$$

In Eq. (3.39), one constraint needs to be satisfied first. That is, the total number of possible candidate basis functions cannot be larger than the number of samples to

generate the Kriging prediction. Therefore, by finding the largest P such that $C_{m+P}^P \leq n-1$, the highest order P is determined. The reason for using $n-1$ instead of n is that it is known that when the number of basis functions equals the number of samples, it could cause an over-fitting problem in the Kriging prediction. Therefore, to make the Kriging method work robustly, it is recommended to use $n-1$ instead of n to find the highest order P . After P is determined according to the number of samples, Eq. (3.38) becomes a classic variable selection problem, expressed as

$$\begin{aligned} & \text{Find the subset of } \mathbf{f} \\ & \text{to minimize } \sigma^2 = \frac{1}{n} (\mathbf{y} - \mathbf{F}\boldsymbol{\beta})^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{F}\boldsymbol{\beta}) \end{aligned} \quad (3.40)$$

It is obvious that the global optimal subset of these candidate basis functions can be guaranteed only by applying the exhaustive algorithm (EHA), which evaluates all possible $2^{C_{m+P}^P}$ subsets of the basis functions. Consequently, the computational expense of EHA increases rapidly and becomes unaffordable when C_{m+P}^P is large. Therefore, an alternative method to solve Eq. (3.38) needs to be applied such that the Kriging prediction based on this alternative optimal subset is accurate enough and close to the result obtained using the true optimal subset with less computational expense.

As discussed in previous section, many research works have been carried out for the variable selection problems. In an area related to Kriging modeling, the Blind Kriging method uses a Bayesian forward variable selection to find the significant coefficients in $\boldsymbol{\beta}$ by using the cross-validation error as the objective function. In this Blind Kriging framework, the forward selection scheme can make the optimization process prematurely converge to a local optimal selection.

In this study, the genetic algorithm (GA) is applied to find the optimal selection for basis functions. A main concern of using the GA method is the number of iterations and convergence time. This is true for using the GA to solve a continuous problem. However, in this particular basis function selection problem there are several reasons that the GA method can be efficient and attractive. The main reason is that the GA intends to find the global optimum instead of the local optimum, which leads to a more accurate Kriging prediction compared with the result using a forward selection scheme. The second reason is that it is a discrete minimization problem in Eq. (3.40) with limited C_{m+p}^P possible basis functions. Unlike the encoding or decoding computation for the solution in a continuous problem, the selection of the basis function itself can be directly expressed in genetic form, where 1 means selected and 0 means non-selected. The third reason is that, with selection of complementary (i.e. opposite) subsets for the initial generation, the GA can converge quickly. The fourth reason is that to restrict the total computational time, one can set the maximum number of iterations and modify the highest order of P for the GA method for complex engineering application and yet obtains a satisfactory result. Efficiency strategies for how to apply the GA method for solving Eq. (3.40) are discussed in detail in the following sections.

Step. 1. Initial Generation. The GA procedure starts with an initial generation, called the 0th-generation. In this paper, the 0th-generation includes both the single basis functions solutions and almost-full basis function solutions. The C_{m+p}^P single basis function solutions are defined as $[1,0,0,\dots,0]$, $[0,1,0,0,\dots,0],\dots,[0,0,\dots,0,1]$, which indicate that the single basis function will be considered first. Another C_{m+p}^P almost-full basis function solutions are defined as $[0,1,1,\dots,1],[1,0,1,\dots,1],\dots,[1,1,\dots,1,0]$, which are

complementary to the single basis function solutions, will also be considered first. Other than these solutions, the full basis function solution of each order (from 0th order to P th order) will also be included, which will add $P+1$ solutions into the initial generation. All together, there are $2C_{m+P}^P + P+1$ solutions in the 0th-generation. It is worth mentioning that the single basis function solutions and almost-full basis function solutions are used to avoid reaching the local optimum for basis selection. Based on numerous examples tested during this study, if the GA starts only at one side, either single basis function solutions or almost-full basis function solutions, it is found that there is a very good chance that it will converge to a local optimum and fail to find the global optimum.

Step. 2. Convergence Criteria. As discussed earlier, the convergence criterion of the GA needs to be carefully set to have the GA converge efficiently. In this work, the convergence conditions are chosen as

$$\begin{aligned}
 & \text{(a) Number of Stalled Iteration} = 2 \\
 & \text{(b) } \left| \frac{\sigma_{k-1}^2 - \sigma_k^2}{\sigma_{k-1}^2} \right| \leq 1\% \text{ and } \left| \frac{\sigma_{k-2}^2 - \sigma_{k-1}^2}{\sigma_{k-2}^2} \right| \leq 1\% \\
 & \text{(c) Maximum Number of Iteration} = \binom{P}{n+P}
 \end{aligned} \tag{3.41}$$

Condition (a) means the GA stops if the process variances σ^2 in any two consecutive iterations are the same. Condition (b) means the GA stops if the absolute relative change of σ^2 between two consecutive iterations is less than 1%. Condition (c) means the GA stops if the maximum number of iteration reaches the number of the candidate basis functions. The total stopping criterion is that the GA stops if any of these three conditions is satisfied. After testing a number of different problems, it is found that condition (a) is the most frequent, which indeed indicates the GA stops very quickly.

Step. 3. Additional Efficiency Strategy in Basis Function Selection. When evaluating the process variance σ^2 using different subsets, the optimization problem for θ search needs to be solved every time, which results in a significant computational time if the GPS method is used. With a number of testing problems, it is found that the optimal θ does not significantly affect the result of the basis function selection. Therefore, in this paper, the θ search in evaluating each subset during the GA process is removed; instead, the ordinary Kriging model is generated first and the obtained optimal θ by the GPS method is used during the GA process for basis function selection. Only after the optimal subset of the basis functions is found by the GA method and used to generate the final Kriging prediction, Eq. (3.16) is solved again by using the GPS method to find the optimal θ . Therefore, the GPS method is used twice in the dynamic Kriging process.

3.3.4 Performance and Robustness of GA-Based Basis Function Selection

In this section, the GA-based optimal subset \mathbf{f}_{GA} is compared to the global optimal subset \mathbf{f}_{EHA} obtained from the exhaustive algorithm (EHA) for a small-scale problem to demonstrate the accuracy and the robustness of the GA-based selection method. The Branin-Hoo testing problem with the same 20 LHS samples used by Forrester and Keane is used here. The EHA is first applied to find the true global optimal subset \mathbf{f}_{EHA} . Since the total number of samples is 20, the highest possible order P is found to be 4. Therefore, the total number of possible candidate basis functions is $C_{4+2}^4 = 15$.

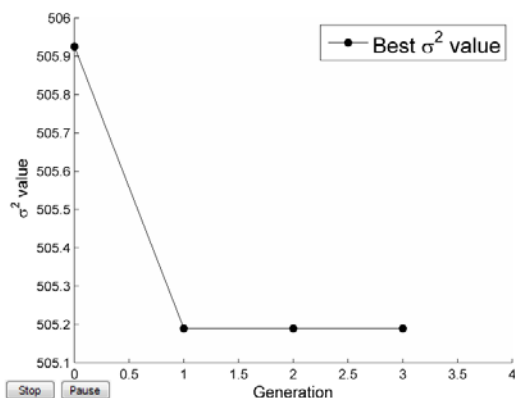
The \mathbf{f}_{EHA} is obtained by running the EHA procedure, which evaluates the σ^2 values for all 2^{15} possible subsets, and the \mathbf{f}_{EHA} is found to be $[1, x_2, x_1x_2, x_2^2, x_1^3, x_1^2x_2, x_1x_2^2, x_2^3, x_1^4, x_2^4]$ with the $rRMSE$ of 0.04 based on 100×100 grid

testing points. When the GA-based basis selection is applied, the \mathbf{f}_{GA} is $[1, x_1, x_2, x_1x_2, x_1^2, x_2^2]$ with the $rRMSE$ of 0.07. The $rRMSE$ obtained by the Blind Kriging method is 0.19, which is not as good as the result of the GA-based basis selection. Figure 4a shows that the optimization process using the GA method, which converges after 4 iterations. The contours of the Kriging prediction based on three different basis selection methods are shown in Figs. 4b – 4d. For computational efficiency, in the GA-based basis selection, only $4 \times (2C_{m+p}^P + 4) = 136$ subsets have been evaluated, whereas in the EHA-based selection process, all $2^{15} = 32768$ subsets have to be evaluated to find \mathbf{f}_{EHA} . Thus, the GA-based selection requires only about $136/32768 = 0.42\%$ of the computational time spent by the EHA to find a solution for this problem. Specifically, the clock time spent on the GA-based selection is 604 ms, whereas the clock time spent on the EHA-based selection is 152370 ms, and the clock time spent on the Blind Kriging is 374 ms on the Intel P8700 CPU computer.

To verify whether the GA-based selection algorithm is robust, the robustness study is carried out as follows. First, a performance is defined as

$$\text{Performance} = \frac{\text{Number of } rRMSE_{GA} < rRMSE_i}{\text{Number of All Possible Solutions}} \times 100\% \quad (3.42)$$

where $rRMSE_i$ is calculated using the Kriging prediction with the i^{th} subset out of the total subsets during the EHA selection process. The performance in Eq. (3.42) indicates the percentile of the better accuracy of the subset of the basis functions obtained by the GA method among all possible subsets. Since the sample position has an influence on the result, ten consecutive trials with different sample sets from LHS are carried out to see if the GA-based selection method is robust.



(a) The GA Optimization History

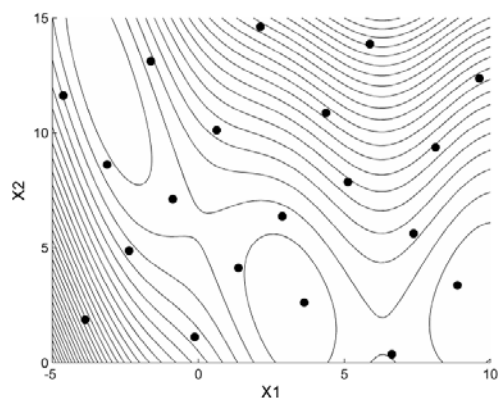
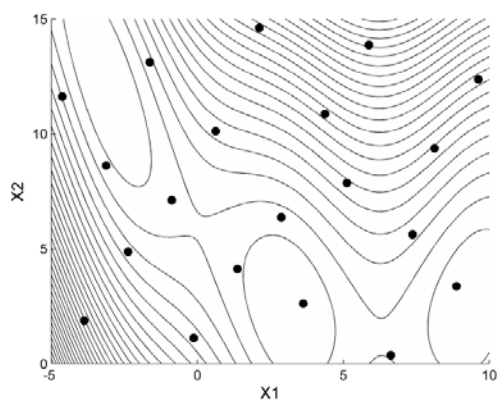
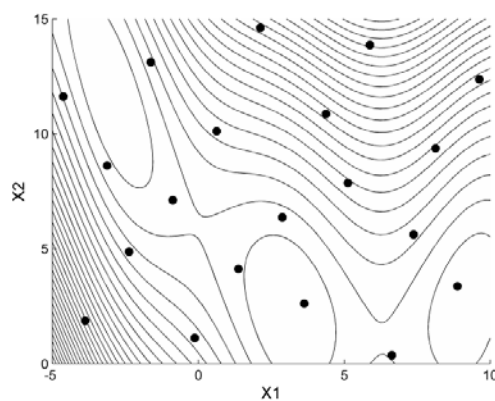
(b) GA-based Prediction ($rRMSE=0.07$)(c) EHA Prediction ($rRMSE=0.04$)(d) Blind Kriging Prediction ($rRMSE=0.19$)

Figure 3. 4 Contours Using Different Basis Function Selections

In Fig. 3.5, the solid line is the $rRMSE$ values obtained by using the GA method for basis selection, the dashed line is the $rRMSE$ values obtained by the optimal basis selection using the EHA method, and the dotted line is the performance as defined in Eq. (3.42) for each trial. It shows that the GA-based selection process can find a very good

subset of basis functions, which is better than about 97% of other subsets in the EHA process, while only using 0.5% of the computational time used by the EHA method.

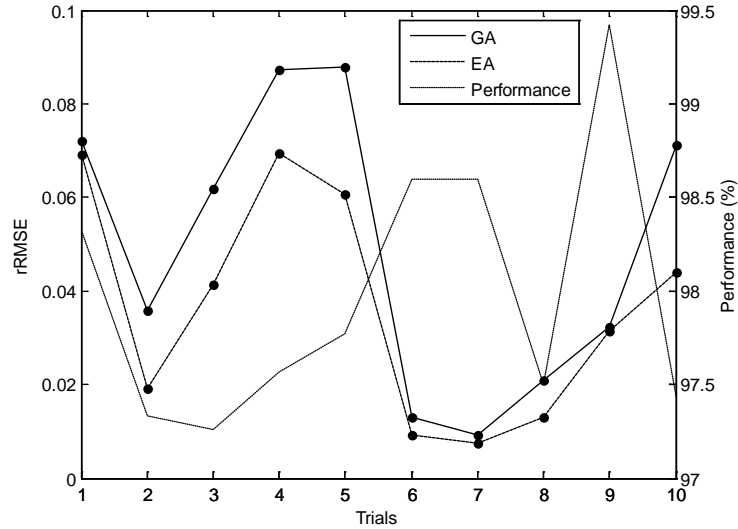


Figure 3. 5 Comparison between GA and EHA Selections

With the optimal \mathbf{f}_{GA} obtained using the GA method by solving Eq. (3.40) and the optimal $\boldsymbol{\theta}$ obtained using the GPS method by solving Eq. (3.16), the dynamic Kriging (DKG) method is formed and compared with other existing surrogate modeling methods in the following sections.

3.3.5 Confidence Level of Dynamic Kriging (DKG)

With the optimum $\boldsymbol{\theta}$ found by pattern search method and the optimum basis function selection Ω_{GA} found by GA, we finalize the proposed DKG method. The prediction from DKG can be constructed in the same procedure shown in Section 3.2.4. Therefore, the prediction at any point \mathbf{x} is expressed as

$$\hat{y}(\mathbf{x}) = (\mathbf{r}(\boldsymbol{\theta}) - \mathbf{F}(\Omega_{GA})\boldsymbol{\lambda}(\boldsymbol{\theta}, \Omega_{GA}))^T \mathbf{R}(\boldsymbol{\theta})^{-1} \mathbf{Y} |_{\boldsymbol{\theta}, \Omega_{GA}} \quad (3.43)$$

Again, the α -level prediction interval of response is written as

$$\hat{y}(\mathbf{x}) - Z_{1-\alpha/2}\sigma(\mathbf{x}) \leq y(\mathbf{x}) \leq \hat{y}(\mathbf{x}) + Z_{1-\alpha/2}\sigma(\mathbf{x}) |_{\boldsymbol{\theta}, \Omega_{GA}} \quad (3.44)$$

Equation (3.44) shows that the confidence level of the prediction from dynamic Kriging method, which is a statistical quantification saying that

$$\text{Prob}\left\{\hat{y}(\mathbf{x}) - Z_{1-\alpha/2}\sigma(\mathbf{x}) \leq y(\mathbf{x}) \leq \hat{y}(\mathbf{x}) + Z_{1-\alpha/2}\sigma(\mathbf{x}) |_{\boldsymbol{\theta}, \Omega_{GA}}\right\} = 1 - \alpha \quad (3.45)$$

3.4 Comparison Study

To compare the performance of the DKG method against other surrogate modeling methods, we selected the four most widely used surrogate modeling methods [Forrester and Keane, 2009], which are the UKG method, the polynomial response surface (PRS) method, the radial basis function (RBF) method, and the blind Kriging method (BKG). To make a fair comparison, we first need to specify how these methods are optimally used in this study.

For the UKG method, the mean structure is set to be second-order polynomials. For the PRS method, the response y is considered as the linear combination of regression basis functions. The predicted response is expressed as

$$y = a_0 + \sum_{i=1}^m a_i x_i + \sum_{i,j=1}^m a_{ij} x_i x_j + \dots + \sum_{\substack{i,j,\dots,k=1 \\ P}}^m a_{ij\dots k} x_i x_j \dots x_k \quad (3.46)$$

where the P is the highest order. In this comparison study, P is decided by minimizing the cross-validation error. For the RBF method, the response y is considered as a linear combination of basis functions, expressed as

$$y = \mathbf{w}^T \boldsymbol{\Psi} = \sum_{i=1}^n w_i \psi(\|\mathbf{x} - \mathbf{c}_i\|) \quad (3.47)$$

where \mathbf{c}_i is the center of the i^{th} basis function. In this paper, $\psi(r) = e^{-r^2/2\sigma^2}$ is used. The σ value is also determined by minimizing the cross-validation error. For the SVR method, the response y is expressed as

$$y = \mu + \sum_{i=1}^n w_i \psi(\mathbf{x}, \mathbf{x}_i) \quad (3.48)$$

In this study, the kernel function is $\psi(\mathbf{x}, \mathbf{x}_i) = e^{-\|\mathbf{x}_i - \mathbf{x}\|^2/2\sigma^2}$. The σ value is determined by minimizing the cross validation error.

The SURROGATES toolbox [Viana et al, 2009] is used to conduct the first four methods with the modification of using cross validation to find the best value of the parameter for each method. For the blind Kriging method, the SUMO toolbox [Gorissen et al. 2009] which is referred by the authors of blind Kriging [Joseph et al. 2008] is used.

The comparison procedure is carried out as follows. First, n samples are generated by the LHS method. Secondly, five surrogate models are generated based on the given samples. After constructing the surrogate models from six methods, the function values from the surrogate model at S evenly distributed testing points are predicted and the $rRMSE$ values are calculated as the error measurement. Then, a rank is determined for these five methods in terms of the accuracy of the generated surrogate model based on the MSE values from each method. To eliminate the influence of the sample profile, the comparison is conducted for 50 trials, and the frequency of being identified as the best surrogate model is counted to find the method that performs the best.

For comparison of these methods, one important issue that needs to be pointed out is “at what level of accuracy these surrogate models should be compared?” That is,

comparing performance of surrogate modeling methods when none of the surrogate model achieved an appropriate level of accuracy for the purpose of applications is meaningless. Therefore, one first needs to set the level of accuracy at which the surrogate model will be used for the comparison study. In this study, the coefficient of determination (R^2) is used as the normalized accuracy measurement to check if the surrogate model is acceptable or not. The surrogate model is defined as accurate when median of the R^2 value is larger than 0.99 for 50 trials. The rank of the performance of each surrogate modeling method is compared at the sample size when at least one method can generate a surrogate model with R^2 larger than 0.99.

3. 5 Numerical Examples for Surrogate Modeling

3.5.1 2-D Branin-Hoo Example

The first problem tested is the Branin-Hoo problem, which is expressed in Eq. (3.35). In this problem the true function is a combination of polynomial and cosine function. Therefore it is not in favor any of the five methods and can be viewed as an unbiased problem for all five methods. As shown in Table 3.8, the comparison started with 16-sample case and the DKG method again achieved the acceptable surrogate model in first place at 18-sample case where the DKG method has been identified as the best for 44 times, as shown in Table 3.9. The mean values of the $rRMSE$ for five methods are shown in Table 3.10.

3.5.2 12-D M1A1 Tank Roadarm Example

The second problem used for comparison study is an engineering application of an M1A1 tracked vehicle roadarm problem.

Table 3. 8 R^2 Median History (Branin-Hoo, 50 trials)

Sample Size	UKG	RBF	PRS	BKG	DKG
16	0.784	0.586	0.961	0.901	0.975
17	0.839	0.612	0.967	0.928	0.989
18	0.857	0.731	0.970	0.979	0.992
19	0.872	0.789	0.952	0.982	0.996
20	0.911	0.805	0.975	0.993	0.999

Table 3. 9 Frequency of Rank of Five Methods (Branin-Hoo, 18 pts, 50 trials)

Rank	UKG	RBF	PRS	BKG	DKG
1st	0	0	0	7	43
2nd	15	8	0	23	4
3rd	18	14	2	14	2
4th	16	20	8	5	1
5th	1	8	40	1	0

Table 3. 10 Mean $rRMSE$ Values for Each Method (Branin-Hoo, 18 pts, 50 trials)

	UKG	RBF	PRS	BKG	DKG
Mean $rRMSE$	2.8321	4.5357	6.3349	1.7615	0.6810

The roadarm is modeled using 1572 eight-node isoparametric finite elements (SOLID45) and four beam elements (BEAM44) of Ansys, as shown in Fig. 3.7, and is made of S4340 steel with Young's modulus $E=3.0 \times 10^7$ psi and Poisson's ratio $\nu=0.3$. The durability analysis of the roadarm is carried out using the Durability and Reliability Analysis Workspace (DRAW) [DRAW Manual, 1999] to obtain the fatigue life. The fatigue lives at the 13 critical nodes shown in Fig. 3.7 are chosen as the design constraints.

In Fig. 3.8, the shape design variables consist of four cross-sectional shapes of the roadarm where the widths (x_1 -direction) of the cross-sectional shapes are defined as design variables d_1 , d_3 , d_5 , and d_7 at intersections 1, 2, 3, and 4, respectively, and the heights (x_3 -direction) of the cross-sectional shapes are defined as design variables d_2 , d_4 , d_6 , and d_8 .

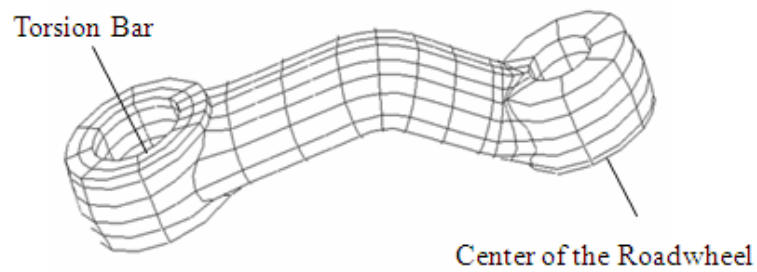


Figure 3. 6 Finite Element Model of Roadarm

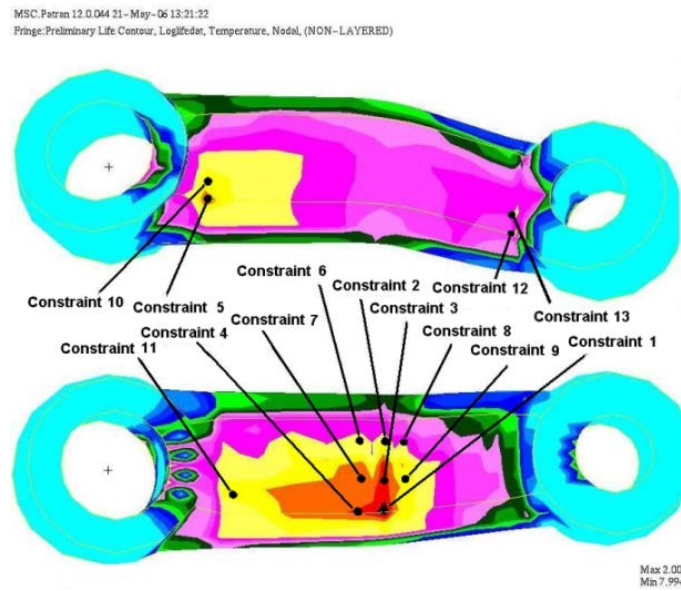


Figure 3. 7 Fatigue Life Contour and Critical Nodes of Roadarm

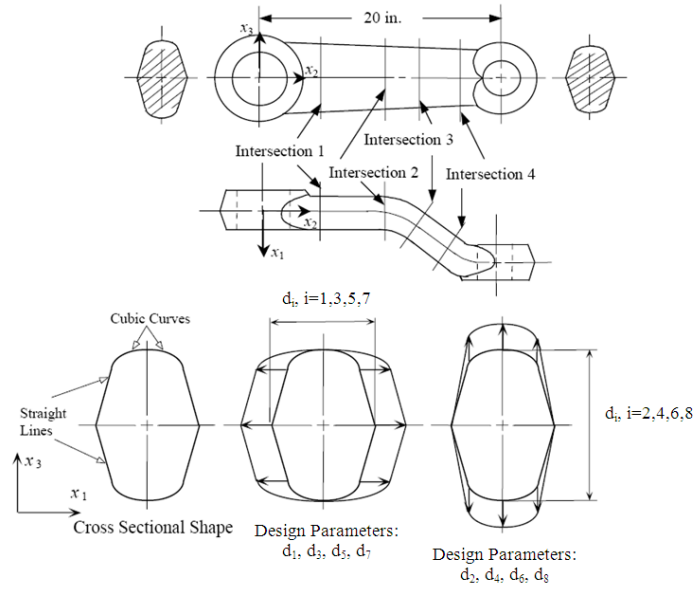


Figure 3. 8 Shape Design Variables of Roadarm

Since the finite element analysis and fatigue analysis are time-consuming, the surrogate model is needed when carrying out design optimization. For comparison of the DKG and other metamodeling methods, the normalized fatigue life at the first critical node is used as the response, and the surrogate model is to be generated for

$$G(\mathbf{d}) = 1 - \frac{L(\mathbf{d})}{L_t}, \quad (3.49)$$

where $L(\mathbf{d})$ is the crack initiation fatigue life at the first critical node, and L_t is the crack initiation target fatigue life, which is 5 years for this example. The domain for generating the surrogate model is defined as a hyper-sphere with a radius of $5\% \times \|\mathbf{d}_0\|$ where $\mathbf{d}_0 = [1.750 \ 3.250 \ 1.750 \ 3.170 \ 1.756 \ 3.038 \ 1.752 \ 2.908]$. The same 50-trial statistical study as conducted in the previous example is carried out. In each trial, 50 LHS samples within the hyper-sphere are randomly generated. The surrogate models are generated by each of

the five metamodeling methods. 2000 LHS samples are first evaluated using the finite element analysis and used to calculate the $rRMSE$ value for each surrogate model.

After 50 trials, the rank of the $rRMSE$ values of the surrogate models using different metamodeling methods is calculated and shown in Table 3.11. Again, the DKG method performs the best in 28 times for generating the most accurate surrogate model, followed by the blind Kriging method. Table 3.12 shows the mean $rRMSE$ values for each of the surrogate modeling methods where the $rRMSE$ from the DKG method is smallest and followed by the blind Kriging method.

It is noted that the fatigue life response for the roadarm problem is mildly nonlinear. After checking the selected optimal subset of the basis functions from both the dynamic Kriging method and blind Kriging method, it is found that the optimal subsets from two methods are almost the same for this example due to the mild nonlinearity. The main difference in prediction accuracy comes from the optimal θ values when the two methods use different optimization methods to solve Eq. (3.16).

3.6 Deterministic Design Optimization Using Dynamic Kriging Method

In simulation-based design optimization, surrogate models are widely used to efficiently approximate the compute-intensive physical responses. In this section, a detailed practical use of the DKG method for design optimization is discussed. After the explanation of several efficiency strategies of how to use the DKG method, an engineering design optimization problem is used to demonstrate the overall performance of the DKG method for the simulation-based design optimization.

Table 3. 11 Frequency of Rank of Five Methods (Roadarm, 50 pts, 50 trials)

Rank	UKG	RBF	PRS	BKG	DKG
1st	0	0	0	22	28
2nd	6	1	0	21	22
3rd	38	2	4	6	0
4th	6	11	32	1	0
5th	0	36	14	0	0

Table 3. 12 Mean $rRMSE$ Values for Each Method (Roadarm, 50 pts, 50 trials)

	UKG	RBF	PRS	BKG	DKG
Mean $rRMSE$	1.4461	3.1603	2.2046	0.6518	0.5845

3.6.1 Local Window for Surrogate Modeling

Since the DKG method selects the best basis functions subset according to the nonlinearity of the response, it is better to generate the surrogate model on a local window than to generate a global surrogate model on the entire design domain. When the candidate design point moves at each iteration, the sample set within the local window changes; therefore the DKG method will choose different basis functions subset according to the local nonlinearity of the response to generate the most accurate surrogate model locally. This local window concept is visualized in Fig. 3.19. The hypersphere used to define the local window is expressed as

$$\sum_{i=1}^m (x_i - d_i)^2 \leq R^2 \quad (3.50)$$

where $\mathbf{d} = [d_1, d_2, \dots, d_m]$ is the current design point and R is the radius, respectively. In this paper, the R value is set as $5\% \times \|\mathbf{d}\|$.

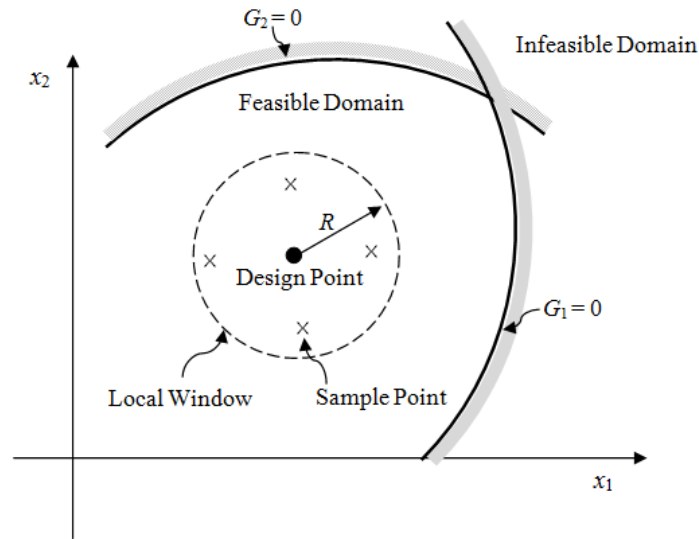


Figure 3. 9 Local Window for Surrogate Model

To see how the DKG method works effectively in the local window, consider a 2-D highly nonlinear polynomial function expressed as

$$G(\mathbf{X}) = -1 + (0.9063X_1 + 0.4226X_2 - 6)^2 + (0.9063X_1 + 0.4226X_2 - 6)^3 - 0.6 \times (0.9063X_1 + 0.4226X_2 - 6)^4 - (-0.4226X_1 + 0.9063X_2) \quad (3.51)$$

and generate surrogate models for Eq. (3.51) at three different design points (\mathbf{d}_1 , \mathbf{d}_2 , \mathbf{d}_3) using the DKG method within each local window. The location of three design points and local windows at each point are shown in Fig 3.10.

Nine initial samples are randomly generated within each local window, and basis functions up to the second-order polynomial function are used for the DKG method. After applying the DKG method at each local window, $\{x_1, x_2, x_1x_2\}$, $\{1, x_1, x_2, x_1^2, x_2^2\}$, and $\{1, x_1, x_2, x_1x_2, x_1^2\}$ are selected as the best basis functions at local windows at \mathbf{d}_1 , \mathbf{d}_2 , and \mathbf{d}_3 , respectively, to accurately describe the true function.

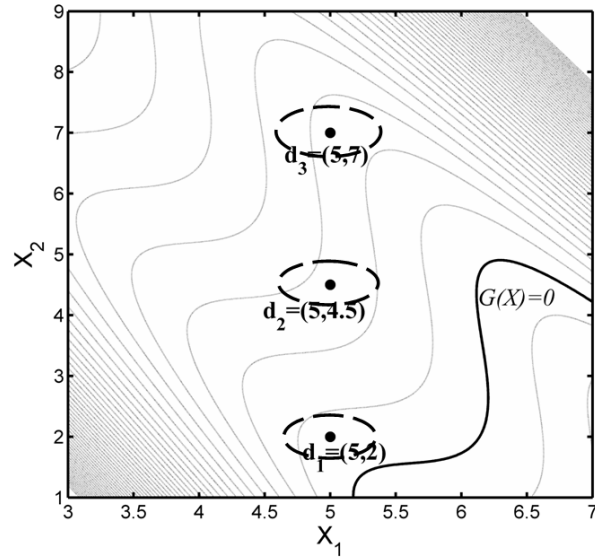


Figure 3. 10 Contour of True Response at Different Local Windows

The fact that different basis function sets are selected at different design points means that one basis function set cannot best describe the local nonlinearity of the true function since the local nonlinearity changes as the design point moves. Furthermore, the fact that all six possible basis terms are not selected shows that it is not necessarily good to use all available terms for the generation of surrogate models; this indeed shows the effectiveness of the DKG method.

3.6.2 Sampling Strategy

After deciding the local window for the surrogate model generation, N_r initial samples are generated on the local window using the Latin Centroidal Voronoi Tessellations (LCVT) for evenly distributed samples, and then surrogate models are generated on the local window. The minimum number of the initial samples is decided by $\binom{P}{m+P}$. For example, for an 8-D example with up to second-order polynomial basis

functions, the minimum number of Nr will be $\binom{2}{8+2} = 45$. However, for high-dimensional problems, the minimum number of initial samples may not be sufficient to generate accurate surrogate models; more samples may need depending on the accuracy of the surrogate model. The accuracy of the surrogate model generated with the initial samples Nr can be estimated using

$$\eta = \frac{\text{mean}(M\hat{S}E(\mathbf{x}_i))}{\text{Var}(y(\mathbf{x}_j))}, \text{ for } i = 1 \sim Nr \text{ and } j = 1 \sim n \quad (3.52)$$

where $\text{Var}(y(\mathbf{x}_j))$ is the variance of n true responses at the sample points and is used to normalize the accuracy measure, Nr is the total number of testing points generated using LCVT, and $M\hat{S}E$ is the predicted mean square error (MSE) from the DKG model. The physical meaning of the accuracy measure in Eq. (3.52) is related to the prediction variance of the Kriging model. Hence, the smaller the prediction variance is, the more accurate the surrogate model is. If the accuracy of a surrogate model is satisfactory, which is defined as $\eta \leq 1\%$ in this study, the surrogate model can be used for optimization. However, if the accuracy does not satisfy the target, more samples are sequentially inserted within the local window until the surrogate model satisfies the target accuracy condition. The new inserting point \mathbf{x}_{new} is chosen as the one that has the largest $M\hat{S}E$ value among the testing points in the local window. For a typical design optimization problem, multiple constraints are usually involved. In this case, the accuracy measure in Eq. (3.52) needs to be modified to reflect the effect of multiple surrogate models. In this study, the maximum value of accuracy measures for each

surrogate model is used as the overall accuracy measure for multiple surrogate models, and thus the accuracy measure is given by

$$\eta_{\max} = \max \left\{ \frac{\text{mean}(\hat{MSE}_k(\mathbf{x}_i))}{\text{Var}(y_k(\mathbf{x}_j))} \right\}, \text{ for } i = 1 \sim Nr, j = 1 \sim n, k = 1 \sim nc \quad (3.53)$$

where $y_k(\mathbf{x}_j)$ and \hat{MSE}_k are the variance of n true responses and the MSE for the k^{th} surrogate model, and nc is the number of surrogate models. Correspondingly, the new inserting point \mathbf{x}_{new} is chosen as the point that has the largest \hat{MSE} among all constraints, expressed as $\arg \max_{\mathbf{x}_i} \{ \hat{MSE}_k(\mathbf{x}_i) \}, \text{ for } i = 1 \sim Nr, j = 1 \sim n, k = 1 \sim nc$.

3.6.3 Adaptive Initial Point for Pattern Search and Basis

Function Selection

When applying the DKG method to a complex engineering design optimization problem, the number of variables used for surrogate modeling is usually large. In such cases, the pattern search algorithm to find the optimal correlation parameter θ and the genetic algorithm to find the optimal basis function subset may become computationally expensive. It is known that the computational efficiency of the pattern search algorithm and the genetic algorithm is strongly affected by the initial search point. If the initial search starts from the neighboring area of the true optimum, it can find the optimum θ within a remarkably shorter time than if it starts from a point far away from the optimum. Moreover, along the optimization history, if the design movement is small, which means that the current design is near the optimum design, the surrogate model generated at the current design will be very similar to the one generated at the previous design. This

means that two optimum θ will be close. Therefore, we can adaptively use the optimal θ obtained in the previous iteration as the initial point for the pattern search of the current iteration instead of using arbitrary initial point. Similarly, the optimal basis function subset $\mathbf{f}_{k-1}^{\text{opt}}$ found in the previous iteration will be included in the initial generation in GA for the current iteration. This will save computational time for the DKG method in particular when the current design is in the neighborhood of the optimum design. The overall flowchart of using the DKG method for the design optimization is shown in Fig. 3.11.

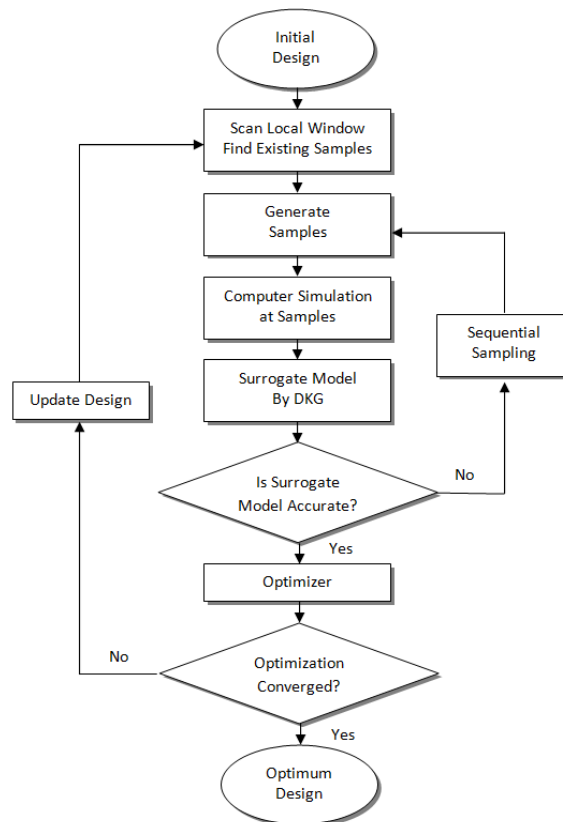


Figure 3. 11 Flowchart of Design Optimization Using The DKG Method

3.6.4 Numerical Example

This section uses the M1A1 tracked vehicle roadarm problem to carry out the design optimization using the DKG method with the efficiency strategies discussed in previous sections. The finite element model, the critical points, and the design variables definition are the same as shown in Figs. 3.6-8. Table 3.13 shows the initial design and the design domain for each variable.

Table 3. 13 Design Variables and Design Domain

Design Variables	Lower Bound \mathbf{d}^L	Initial Design \mathbf{d}^o	Upper Bound \mathbf{d}^U
d_1	1.350	1.750	2.150
d_2	2.650	3.250	3.750
d_3	1.350	1.750	2.150
d_4	2.570	3.170	3.670
d_5	1.356	1.756	2.156
d_6	2.438	3.038	3.538
d_7	1.352	1.752	2.152
d_8	2.508	2.908	3.408

The design optimization for the M1A1 tracked vehicle roadarm is formulated to

$$\begin{aligned}
 & \text{minimize} && \text{Cost}(\mathbf{d}) \\
 & \text{subject to} && G_j(\mathbf{d}) < 0, \quad j = 1, \dots, 13 \\
 & && \mathbf{d}^L \leq \mathbf{d} \leq \mathbf{d}^U, \quad \mathbf{d} \in \mathbf{R}^8
 \end{aligned} \tag{3.54}$$

where

Cost(\mathbf{d}): Weight of Roadarm

$$G_j(\mathbf{d}) = 1 - \frac{L(\mathbf{d})}{L_t}, \quad j = 1 \sim 13 \quad (3.55)$$

$L(\mathbf{d})$: Crack Initiation Fatigue Life,

L_t : Crack Initiation Target Fatigue Life (=5 years)

The constraint value $G_j(\mathbf{d})$ and the sensitivity of $G_j(\mathbf{d})$ with respect to the design variables are predicted by $\hat{G}_j(\mathbf{d})$ and $\partial\hat{G}_j(\mathbf{d})/\partial d_i$ where $\hat{G}_j(\mathbf{d})$ is the prediction generated by the DKG method. The sequential quadratic programming algorithm (SQP) is used to solve the optimization problem of Eq. (3.54). At each iteration, 15 samples are used as the initial samples in the local window and new samples are sequentially inserted if the accuracy of the surrogate model is not achieved. After 11 iterations, the optimization process converged to the optimum design, using 146 samples altogether. To verify the accuracy of the optimum result, the sensitivity-based design optimization is carried out too.

In the sensitivity-based design optimization, the sensitivity information $\partial G_i / \partial d_j, i = 1 \sim 13$ and $j = 1 \sim 8$ is available through the design sensitivity analysis method and used to solve Eq. (3.54); therefore no surrogate model is needed in this procedure. Table 3.14 shows the comparison results for the two approaches. The optimum design obtained using the surrogate-based design optimization is almost identical with the optimum design obtained using the sensitivity-based one except the small difference in d_3 and d_6 . The sensitivity-based design optimization requires 11 function and 11 sensitivity evaluations. One sensitivity evaluation includes sensitivity calculations for all design variables, so it requires 8 sensitivity calculations in this

example, whereas the sampling-based design optimization requires a total of 146 samples for the surrogate model generation using the DKG method.

Table 3. 14 Comparison between Sensitivity-based and Surrogate-based Optimum

Design Variable	Initial	Sensitivity-Based	Surrogate-Based
d ₁	1.750	1.653	1.653
d ₂	3.250	2.650	2.650
d ₃	1.750	1.922	1.911
d ₄	3.170	2.570	2.570
d ₅	1.756	1.478	1.478
d ₆	3.038	3.287	3.297
d ₇	1.752	1.630	1.630
d ₈	2.908	2.508	2.508
# of F.E.		11+11×8	146
Iteration		11	11
Active Constraints		1,3,5,8,12	1,3,5,8,12
Cost	515.09	466.80	466.93

In Fig. 3.12, the average computational time to generate each surrogate model is shown as the solid line with squares. The average computational time to generate each surrogate model without applying the efficiency strategy is also shown as the solid line with stars. It is shown that with applying the efficiency strategy, the average computational time is reduced from 4 seconds to 1 second. What is more, the most reduction occurs at the second iteration due to the improvement of the initial theta point and basis function selection.

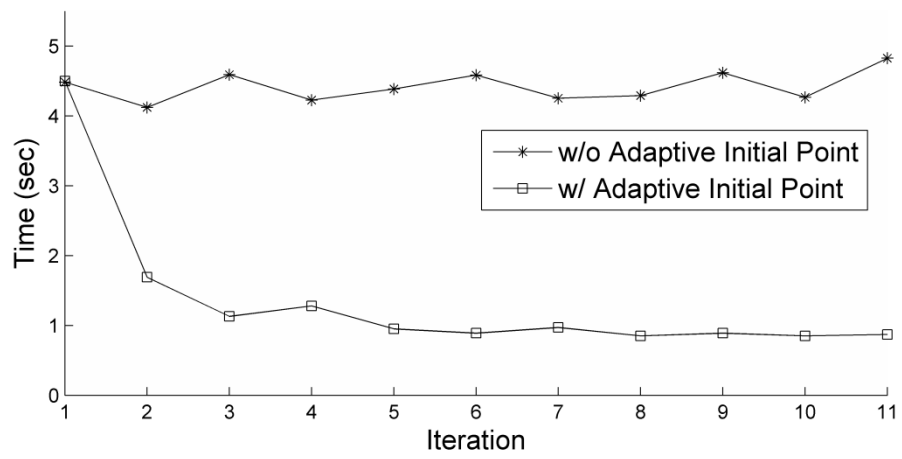


Figure 3. 12 Mean Value of Time Spent for the DKG Method

CHAPTER IV

DESIGN OF EXPERIMENT

4.1 Introduction

The design of experiment (DoE) is the one of the key issue for the success of the surrogate modeling. Over the last decades, researchers developed number of sampling techniques to meet different purpose in surrogate modeling for accuracy and efficiency issue. Based on their underlying principles, current DoE approaches can be roughly categorized into two groups: one-stage sampling and sequential sampling. Comprehensive reviews of DoE approaches can be found in the literatures [Queipo et al., 2005; Simpson et al., 2002; Wang and Shan, 2007].

One-stage sampling approaches generate all sample points at one time to maximize the observed information. Typical one-stage sampling approaches include orthogonal array design [Hedayat et al., 1999; Owen, 1992], maximum entropy design [Shewry and Wynn, 1987; Currin et al., 1991], minmax and maxmin distance design [Johnson et al., 1990], Latin hypercube design [Sacks et al., 1989], Latin Centroidal Voronoi Tessellation [Burkardt et al., 2002] and the D-optimal design [Goel et al., 2008]. Having one form or another, these one-stage sampling approaches occupy the design domain uniformly and do not account the region of nonlinearity of the true response, which makes it hard to use these sampling methods to effectively occupy the highly nonlinear region and achieve accuracy rapidly.

The sequential sampling approaches, on the other hand, sample points into the design space successively. The locations of the new samples are identified based on those of existing samples and the infill criteria. Several infill criteria are developed and applied

into surrogate modeling based design optimization. Typical sequential sampling approaches include statistical lower bound [Lin et al., 2004], probability of improvement and expected improvement [McKay et al., 1979],

4.2 Sequential Sampling Method Using DKG

To conduct the DKG method for generating a surrogate model, one crucial problem is the design of experiment (DoE). As shown in the previous section, the performance of the DKG method improves as the number of samples increases. However, it is hard to tell how many samples are enough in the beginning. Moreover, traditional LHS, grid sampling and Latin Centroidal Voronoi Tessellation (LCVT) are occupying the design domain uniformly and do not account the region of nonlinearity of the true response which makes hard to use these sampling methods to effectively occupy the highly nonlinear region and achieve accuracy rapidly. Therefore, a technique that can identify the highly nonlinear region and how many samples should be used to obtain an accurate surrogate model is needed. In this study, a sequential sampling technique is developed using the bandwidth of the prediction interval determined by Eq. (3.22). This sampling technique is integrated with the DKG method to find the region where the surrogate model has the least prediction confidence, which means that the region is highly nonlinear. This sampling technique is conducted in a sequential manner. Thus, it can tell whether or not the existing samples are enough by verifying the accuracy of the surrogate model.

Consider n existing samples, where the DKG method is conducted first. The sampling insertion criterion $IC(\mathbf{x})$ for finding the next sample position is defined as

$$IC(\mathbf{x}) = d(\mathbf{x}) \quad (4.1)$$

where $d(\mathbf{x})$ is the prediction interval bandwidth by the DKG method from Eq. (3.22) at any given point \mathbf{x} . The next sample point \mathbf{x}_{new} is identified by

$$\arg\{\max_{\mathbf{x} \in \mathbf{D}}(IC(\mathbf{x}))\} \quad (4.2)$$

where \mathbf{D} is the design domain. By enforcing \mathbf{x}_{new} to be the solution of Eq. (4.2), the “weakest point” in the domain is identified where we have the least confidence in the prediction.

To demonstrate the process of this sequential sampling method, first consider a 1- D problem expressed as

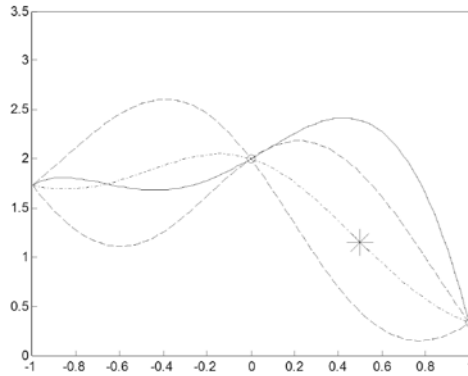
$$y(x) = 0.5x^5 - 1.5x^4 - 2.5x^3 + 0.53x^2 + 1.3x + 2.0 \quad (4.3)$$

where $x \in [-1, 1]$. The sequential sampling method is initiated with three evenly distributed samples as shown in Fig. 4.1(a). In the figures, the solid line, the solid-dash line, and the two dashed lines are the true response given by Eq. (4.3), the predicted surrogate model by Eq. (3.21), and the 95% prediction interval by Eq. (3.22), respectively. The star is the identified next insertion point within the domain.

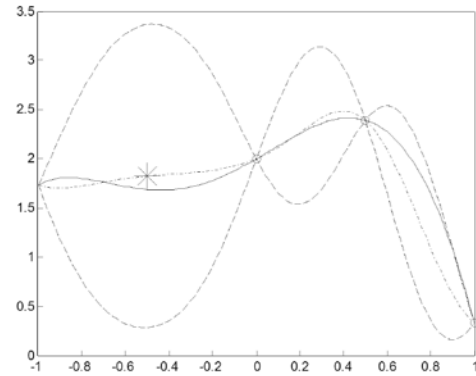
Figures 4.1(a) ~ (d) show that the bandwidth of the prediction interval decreases fast and the surrogate model from Kriging converges to the true response rapidly. A region where there is a large discrepancy between the true model and the surrogate model is correctly identified during the iterations by applying the IC criterion.

To assess the accuracy of the surrogate model, an error definition is given by the average of relative error within the domain as

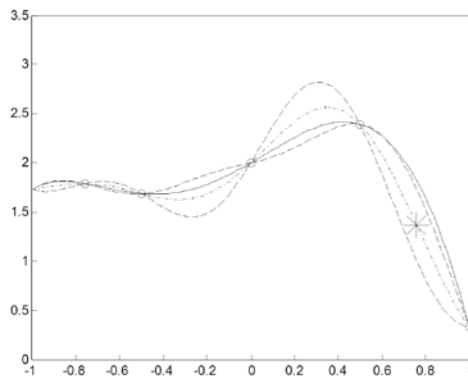
$$Err = \frac{1}{S} \sum_{i=1}^S \left| \frac{d(\mathbf{x}_i)}{\hat{y}(\mathbf{x}_i)} \right| \quad (4.4)$$



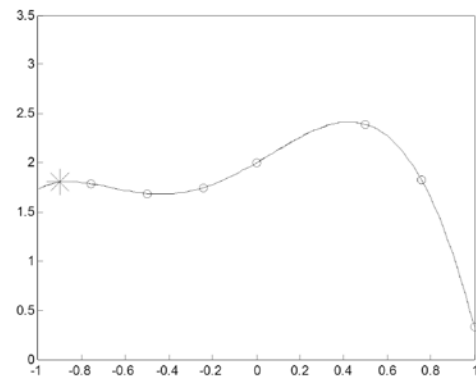
(a) Initial Stage: 3 Samples



(b) After 1 More Sample Inserted



(c) After 3 More Samples Inserted



(d) After 5 More Samples Inserted

Figure 4. 1 Surrogate Model Using Sequential Sampling Method

where $d(\mathbf{x})$ is the prediction interval bandwidth from Eq. (3.22), $\hat{y}(\mathbf{x})$ is the predicted value by the DKG method at any given point \mathbf{x} , and S is the number of checking points over the entire domain, which takes a larger number, 10^4 in this study. With this accuracy definition of the surrogate model, the decision of how many samples are needed to

generate an accurate surrogate model can be made. The sequential sampling process will be continued until it meets the accuracy tolerance Tol , given as

$$RErr = \left| \frac{Err_{k-1} - Err_k}{Err_k} \right| < Tol \quad (4.5)$$

where Err_k is the error at the k^{th} iteration and $RErr$ is the relative error. Overall, the entire process of conducting this sequential-sampling-based DKG (SS-DKG) method is shown in Fig. 4.2.

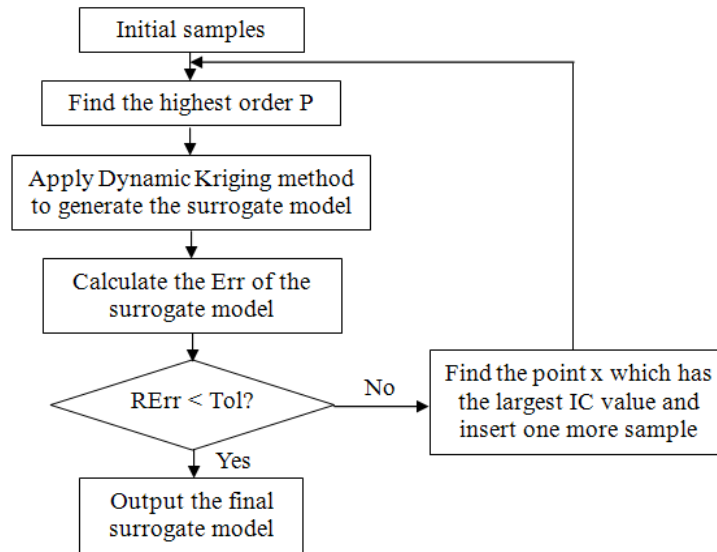


Figure 4. 2 Entire Process of Sequential-Sampling-Based DKG for Surrogate Modeling

4.3 Comparison of Sequential Sampling and Latin

Hypercube Sampling

An error convergence comparison study is carried out between sequential sampling, grid sampling and Latin hypercube sampling for a complex 2-D example. The 2-D function is

$$\begin{aligned}
 Y(\mathbf{x}) = & -1 + (0.9063x_1 + 0.4226x_2 - 6)^2 + (0.9063x_1 + 0.4226x_2 - 6)^3 \\
 & - 0.6 \times (0.9063x_1 + 0.4226x_2 - 6)^4 - (-0.4226x_1 + 0.9063x_2) \quad (4.6) \\
 & \text{where } x_1 \in [5, 9], x_2 \in [1, 5.5]
 \end{aligned}$$

Among the three sampling methods, the grid sampling method samples each design parameter at N points evenly along its axis. Hence, the number of entire samples used is N^m where m is the number of design parameters. In traditional Latin hypercube sampling, N points in an m -dimensional Latin hypercube are to be selected where each of the m coordinate dimensions is discretized to the values 1 through N . The points are to be chosen in such a way that no two points have any coordinate value in common. A so-called improved Latin hypercube sampling (IHS) method was proposed [Beachkofski, 2002]; it attempts to spread out the samples over the entire domain as evenly as possible. In this study, IHS is used for comparison purposes. Since each realization of Latin hypercube sampling is random and different from others, we took 100 trials for IHS, and the maximum, minimum, median and mean of the errors are recorded. The MSE value is calculated at 1000×1000 evenly distributed testing points over the entire domain after using the Kriging method for each sampling method at different sample sizes.

The comparison in Table 4.1 shows that the MSE by using sequential sampling converges more quickly to zero than by using the grid sampling. When compared with IHS, even though in the best case (Min) IHS is more accurate, sequential sampling

converges more quickly in the probability sense (Mean and Median) and yields more accurate results as the samples size increases.

Table 4. 1 Comparison of Three Sampling Methods

# of Sample	Grid Sampling	LHS (100 trials)				Sequential Sampling
		Min	Max	Mean	Median	
9	45.7700	4.5585	45.7662	17.6739	13.5772	28.3378
10	-	1.3421	42.9771	14.0425	10.9951	20.4632
11	-	2.5313	37.7024	10.5840	8.2951	11.1661
12	-	1.4786	28.2741	6.8540	4.8491	7.6797
13	-	0.9482	16.9488	4.5887	3.4631	2.4228
14	-	0.6957	14.2426	3.3310	2.9982	2.9324
15	-	0.5573	10.7053	2.6185	2.0486	1.6182
16	1.1469	0.3461	7.6221	1.7053	1.3589	1.1815
17	-	0.3961	11.9416	1.4349	0.9891	0.7090
18	-	0.3243	7.8473	1.1978	0.7965	0.4948
19	-	0.3398	3.0215	0.8427	0.6542	0.3464
20	-	0.1627	1.7970	0.7403	0.5485	0.3046
25	0.9424	0.0112	0.9794	0.1850	0.1043	0.0326
36	0.0150	0.0004	1.9020	0.0035	0.0021	0.0019

Another widely used 2-D mathematical example for testing surrogate modeling is used to demonstrate the fast convergence of the sequential sampling. This 2-D function is the so-called Branin-Hoo problem given in Eq. (3.35) and shown here

$$Y(\mathbf{x}) = (x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6)^2 + 10(1 - \frac{1}{8\pi}) \cos(x_1) + 10 \quad (4.7)$$

$$-5 \leq x_1 \leq 10, \quad 0 \leq x_2 \leq 15$$

The comparison study is the same as the previous 2-D example, carried out between grid sampling, Latin hypercube sampling and the sequential sampling. Table 4.2

shows the similar result compared with Table 4.1. The sequential sampling method converges faster than Latin hypercube sampling in probability sense.

Table 4. 2 Error Comparison of Three Sampling Methods

# of Samples	Grid Sampling	LHS (100 trials)				Sequential Sampling
		Min	Max	Mean	Median	
9	1.5324	0.9335	10.2490	2.2859	2.0126	3.3253
10	-	0.6862	6.9577	2.0105	1.7270	3.2142
11	-	0.6119	3.6918	1.7331	1.5889	1.8793
12	-	0.5758	3.5160	1.5714	1.4669	0.9074
13	-	0.5486	12.0986	1.6700	1.5116	0.7909
14	-	0.4769	3.7629	1.4600	1.3876	1.0941
15	-	0.1429	3.9855	1.4394	1.3072	0.8423
16	1.1243	0.1506	4.2011	1.3024	1.3808	0.8460
17	-	0.0711	2.5922	1.2321	1.2981	0.7655
18	-	0.0706	3.0256	1.2700	1.3242	0.6200
19	-	0.0239	3.4103	1.0452	1.1424	0.4419
20	-	0.0306	2.3604	1.1194	1.1461	0.4583
25	0.2600	0.0090	2.1187	0.7206	0.7230	0.0123
36	0.5630	0.0018	1.6306	0.5484	0.6692	0.0023

In addition, considering for multidimensional design cases, IHS is computationally far more expensive and not even affordable to apply. Another disadvantage of IHS is that the user does not know how many samples are enough for response surface until all samples are evaluated. If the response surface does not achieve the required accuracy based on the current samples, the user has to increase the sample size and use the Latin hypercube sampling method again. In this case, those previously evaluated samples may not be used at all which leads to a very large waste of sampling cost. On the other hand, in the sequential sampling method, the user increases the sample

size step by step and can stop when the accuracy of response surface is achieved. All of previous samples contribute to the response surface; hence no sample is wasted.

CHAPTER V

MPP-BASED RBDO USING SURROGATE MODEL

5.1 Introduction

When the sensitivity of the performance function with respect to the design variable is not available in complex engineering applications, surrogate model is commonly used to approximate both the function and the sensitivities in RBDO process.

In this study, we propose a new MPP-based RBDO using the proposed SS-DKG method. We use FORM for reliability analysis and apply the SS-DKG to generate the surrogate model for function evaluation and sensitivity calculation. In this way, we keep the efficiency and accuracy of sensitivity-based reliability analysis for probability of failure and offer a way to calculate the function evaluation and sensitivity that is more efficient and accurate compared to the existing global surrogate model methods.

Numerical examples demonstrate that the proposed MPP-based RBDO using SS-DKG can achieve similar accuracy and similar or better results in terms of computational efficiency when compared with the traditional sensitivity-based RBDO. At the same time, it offers a greater capability for dealing with diverse and practical engineering applications where sensitivity is not feasible.

5.2 Sequential Sampling Surrogate Modeling for MPP- Based RBDO

As shown in Chapters 3 and 4, the SS-DKG method is used to generate the surrogate model. In MPP-based RBDO using FORM, the surrogate model from SS-DKG

will be both used to construct the surrogate model and calculate the sensitivities of the surrogate model.

Consider n samples $\{\mathbf{x}^{(i)}\}, i=1,2,\dots,n$ with their performance function values $y(\mathbf{x}^{(i)})$ given within a local reliability analysis window at the given design \mathbf{d} . Based on the DKG method, the approximation of the performance function and its sensitivities at the point of interest \mathbf{x} are expressed as

$$\begin{aligned}\hat{y}(\mathbf{x}) &= \mathbf{r}^T \mathbf{R}^{-1} \mathbf{Y} - (\mathbf{F}^T \mathbf{R}^{-1} \mathbf{r} - \mathbf{f})^T (\mathbf{F}^T \mathbf{R}^{-1} \mathbf{F})^{-1} \mathbf{F}^T \mathbf{R}^{-1} \mathbf{Y} \\ &= \mathbf{f}^T \boldsymbol{\beta}^* + \mathbf{r}^T \boldsymbol{\gamma}^*\end{aligned}\quad (5.1)$$

$$\nabla \hat{y}(\mathbf{x}) = \left[\frac{\partial y}{\partial x_1}, \dots, \frac{\partial y}{\partial x_m} \right]^T = J_f(\mathbf{x})^T \boldsymbol{\beta}^* + J_r(\mathbf{x})^T \boldsymbol{\gamma}^* \quad (5.2)$$

where $\boldsymbol{\beta}^* = (\mathbf{F}^T \mathbf{R}^{-1} \mathbf{F})^{-1} \mathbf{F}^T \mathbf{R}^{-1} \mathbf{Y}$, $\boldsymbol{\gamma}^* = \mathbf{R}^{-1} (\mathbf{Y} - \mathbf{F} \boldsymbol{\beta}^*)$, $(J_f(\mathbf{x}))_{ij} = \frac{\partial f_i(\mathbf{x})}{\delta x_j}$ and

$$(J_r(\mathbf{x}))_{ij} = \frac{\partial R(\boldsymbol{\theta}, \mathbf{x}_i, \mathbf{x})}{\partial x_j}.$$

To successfully conduct the MPP-based RBDO as discussed in Section 2.4, a number of strategies for both efficiency and accuracy are applied, which are explained in the following sections.

5.2.1 Center Fixed Local Support for Surrogate Model in Reliability Analysis

During one design iteration of RBDO the performance function value and sensitivity with respect to design variables are required to find the MPP, which can be obtained using the DKG method. Since the MPP search is conducted on a hyper-sphere in \mathbf{U} -space where the current design is the origin, the surrogate model domain should be a

hyper-sphere that covers the design domain with the radius larger than the target reliability index β_t . In this study, as shown in Fig. 5.1, the local support for the reliability analysis is defined as a hyper-sphere whose radius is $c\beta_t$, where c is a scaling factor which is 1.2~1.5 depending on the nonlinearity of the performance functions.

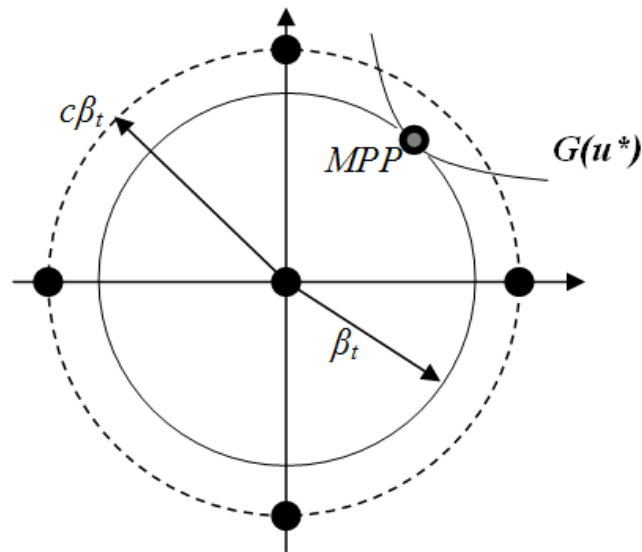


Figure 5.1 Local Support Size of Surrogate Model for Reliability Analysis in U-Space

It is possible to generate a new local surrogate model with new samples for each MPP candidate. That is, new samples around the current MPP candidate are evaluated and a new local surrogate model is generated each time. However this approach would cause two problems. The first problem is that it wastes samples. In fact, for reliability analysis, since the MPP will always be located within a relatively small area around the origin, one center-fixed local surrogate model is enough. The second problem is, if a new surrogate model is generated every time with new samples for a new MPP candidate, the use of the previously evaluated samples for generation of the current surrogate model

may cause the estimated correlation \mathbf{R} matrix to be ill-conditioned and consequently fail to predict a surrogate model. That is, \mathbf{R} may become ill-conditioned if the samples are too aggregated within a narrow space. Figure 5.2 shows how the ill-conditioned matrix happens if we generate samples and new local surrogate model for each MPP candidate search. As shown in Fig. 5.2, after two iterations of the MPP search, ten samples are generated along the axis; then, the current MPP candidate III will evaluate additional five samples. Within the local support of MPP candidate III, there are eight samples that aggregate in two parallel lines. In this situation, the estimated \mathbf{R} would become ill-conditioned. This situation would occur more often in high-dimensional design problems. Because of these two major reasons, it is better to have the local support fixed as a hypersphere with the center at the origin in \mathbf{U} -space for reliability analysis.

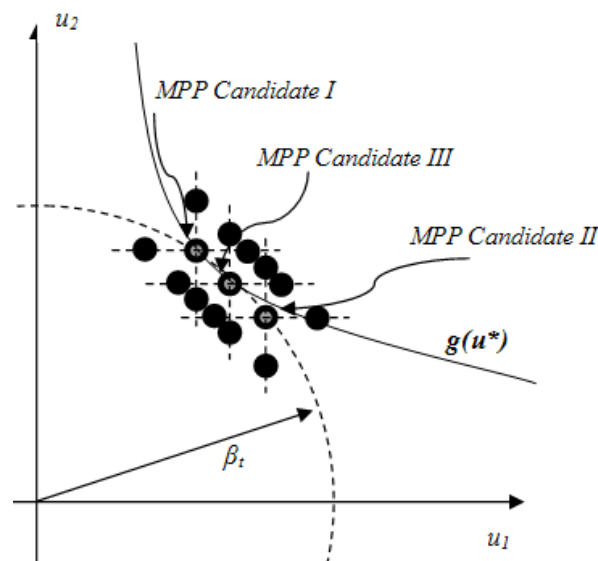


Figure 5.2 Demonstration of Ill-conditioned Matrix Problem in \mathbf{U} -Space

5.2.2 Adjustment of Insertion Criterion and Error

Definition for Surrogate Modeling

As explained in Section 4.2, the sequential sampling method is to identify the point that has the largest IC among the entire design domain. However in the RBDO procedure, we are focusing more on the neighborhood of the point of interest. Hence an adjustment is done to search the “weakest point” around the point of interest, which is described below:

Step 1. Initial samples are placed by grid sampling into the local window, and the responses are evaluated at these samples.

Step 2. Apply the Kriging method based on current existing samples.

Step 3. Check the criterion $IC(\mathbf{x}_i)$ within the domain D , which is defined as

$$D = \{ \mathbf{x} \mid \| y(\mathbf{x}) - \hat{y}(\mathbf{x}) \| \leq Z_{(1-\alpha)/2} \sigma_p(\mathbf{x}) \} \quad (5.3)$$

where \mathbf{x} is the point of interest.

Step 4. Find the point \mathbf{x}^* such that $IC(\mathbf{x}^*)$ takes the largest value and insert one more sample at \mathbf{x}^* .

Step 5. Check the error; if error is larger than tolerance, repeat Step 2 to Step 4.

Otherwise, stop sampling.

It is noted that the error has to reflect the discrepancy of the predicted response and the true response at the point of interest, while we only have the prediction interval bandwidth as the measure of accuracy of response. Hence the error here is defined as the average of the normalized prediction standard deviation as

$$Err = \frac{1}{N} \sum_{i=1}^N \left| \frac{d(\mathbf{x}_i)}{\hat{y}(\mathbf{x}_i)} \right| \quad (5.4)$$

where $d(\mathbf{x}_i)$ is the bandwidth of the prediction interval at \mathbf{x}_i obtained by Eq. (3.22), N is the number of checking points within a small neighborhood of the point of interest $\{\mathbf{x}_i \mid \|\mathbf{x}_i - \mathbf{x}_0\| < d_0\}$, and d_0 is the threshold. From Eq. (5.4) we can see that the error of the current point of interest \mathbf{x}_i is defined as the average error around it; this can be graphically understood in Fig. 5.3. Recall that the purpose of the surrogate model is to predict the function value and sensitivity accurately at the current point of interest; hence there is no need to check the entire support domain, only the neighborhood around the point of interest has to be assured to be accurate. Moreover, checking the entire support domain would make the accuracy criteria harder to achieve and therefore would lead to more sample insertion which is an unnecessary waste for function evaluation.

On the other hand, if we check only the error at the point of interest, there is a chance that the error could be zero if the point of interest is also a sample that has been evaluated. Since the error is used as the criteria to check that whether or not the surrogate model is accurate enough, zero error at the point of interest cannot provide any confidence in the accuracy for predicted sensitivity of the performance function. Therefore, a small neighborhood around the point of interest is most desirable for error checking. In this study, the neighboring area is defined as 10% of the current local window size.

5.2.3 Adaptive Local Window Size in Deterministic Design Optimization

In the RBDO algorithm, PMA+ requires to carry out deterministic optimization first and then start the RBDO process. In surrogate-model-based deterministic

optimization, there is a tradeoff between the support size and the accuracy of the surrogate model. As is well known, a smaller support can provide a more accurate surrogate model than a larger support does. At the same time, using more of previously evaluated samples is also desirable to save computational time, and a larger support can always include more of previously evaluated samples than a smaller support can. Hence it is important to decide a suitable support size to generate the surrogate model. In this study, we propose an adaptive support size method that can resolve this problem for the deterministic optimization process.

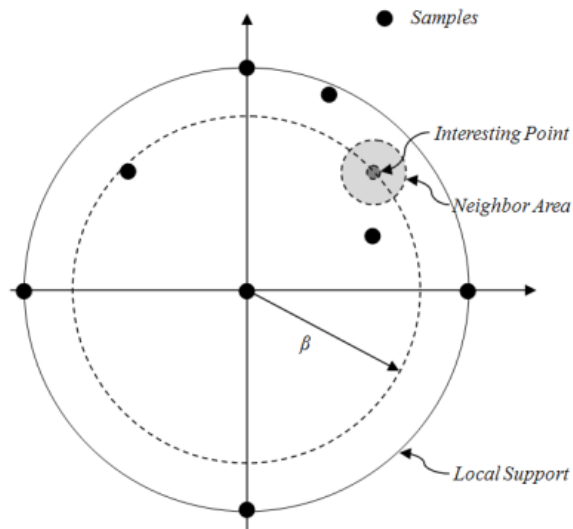


Figure 5.3 Neighboring Area of the Point of Interest for Error Checking

Starting from the initial design point, the first priority is to find the correct direction for the iteration. Hence a smaller support that can give more accurate sensitivity is desirable. As the design moves to the next position, the support size would be decided by the distances from the previous designs as

$$R_k = \min(R_{k-1}\sqrt{d_{k-1}/d_k}, R_{\max}) \quad (5.5)$$

where R_k is the k^{th} support size, R_{\max} is the maximum threshold, and d_k is the distance between the $(k-1)^{\text{th}}$ design and the k^{th} design as shown in Fig. 5.4.

Equation 5.5 implies that when the design movement is large, which means two designs are quite different, the support size should be decreased to yield a more accurate surrogate model to determine the next direction. On the other hand, when the design movement is small, the support size should be increased to include more previously evaluated samples to save computational time.

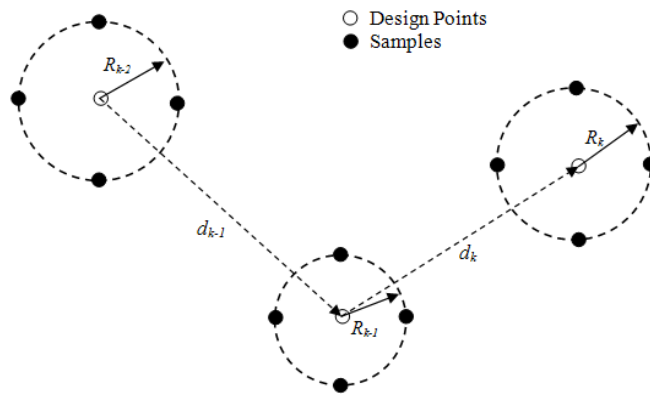


Figure 5.4 Relation Between Support Size and Design Movement

A comparison study shows the efficiency of this adaptive support size in saving function evaluation. A deterministic optimization problem is defined as

$$\begin{aligned}
& \text{minimize} && C(\mathbf{X}) \\
& \text{subject to} && G_i(\mathbf{X}) < 0, \quad i = 1, \dots, nc \\
& \text{where} && \\
& C(\mathbf{X}) = && -\frac{(X_1 + X_2 - 10)^2}{30} - \frac{(X_1 - X_2 + 10)^2}{120} \\
& G_1(\mathbf{X}) = && 1 - \frac{X_1^2 X_2}{20} \\
& G_2(\mathbf{X}) = && -1 + (0.9063X_1 + 0.4226X_2 - 6)^2 + (0.9063X_1 \\
& && + 0.4226X_2 - 6)^3 - 0.6 \times (0.9063X_1 + 0.4226X_2 \\
& && - 6)^4 - (-0.4226X_1 + 0.9063X_2) \\
& G_3(\mathbf{X}) = && 1 - \frac{80}{(X_1^2 + 8X_2 + 5)} \\
& \mathbf{X}^{\text{initial}} = && [5, 5]^T, \quad X_1 \in [0, 10], \quad X_2 \in [0, 10]
\end{aligned} \tag{5.6}$$

The three constraints and the cost function contours are plotted in Fig. 5.5 First, the Kriging method using the fixed support size is applied to solve the deterministic optimization problem. Second the adaptive support size efficiency strategy is applied to solve the same problem. Finally these two results are compared with the result solved by using the sensitivity-based deterministic optimization algorithm. As shown in Table 5.1, the number of function evaluations (NFE) consists of the function evaluation (FE) and the sensitivity calculation (SC). The time for function evaluation is the time used to evaluate the function value at a given sample, and the time for sensitivity calculation is the time used to calculate the first-order derivative of the function respect to design variables and random parameters. The DKG method by using the adaptive support size efficiency strategy not only saves number of function evaluations, but also yields a more accurate optimum design.

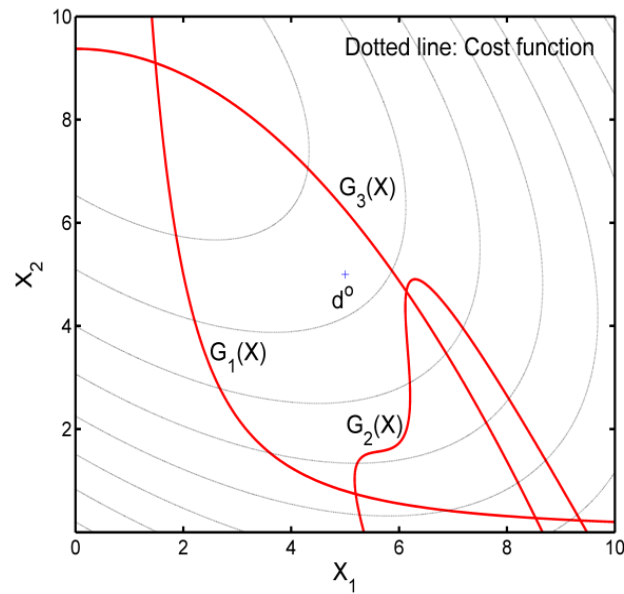


Figure 5.5 Contour of Constraint and Cost Functions

Table 5.1 Comparison Between Fixed & Adaptive Support Size

Case	Optimum Design		Cost	NFE
	X_1	X_2		
Fixed Support	5.2415	0.7278	-2.297	23FE
Adaptive Support	5.1949	0.7414	-2.291	18FE
Sensitivity-Based	5.1969	0.7404	-2.292	9 FE+9*2 SC

5.2.4 Overall Process of MPP-Based RBDO Using Surrogate Modeling

Summarizing results of the three sub-sections above, we can now apply the SS-DKG method for the MPP-based RBDO problem. The SS-DKG, shown as Fig. 5.6, is applied to carry out the function evaluation and sensitivity calculation. For each point of interest, the support size of current surrogate model is decided first, and then the previously evaluated samples within the local support are found. The sequential sampling

strategy is carried out to insert more sample points until the surrogate model achieves the accuracy criterion. In the end the function value and sensitivity value are given by the Kriging method. This SS-DKG is encapsulated as a black-box in the overall RBDO procedure and is called whenever the function value and sensitivity value are required, as shown in Fig. 5.7.

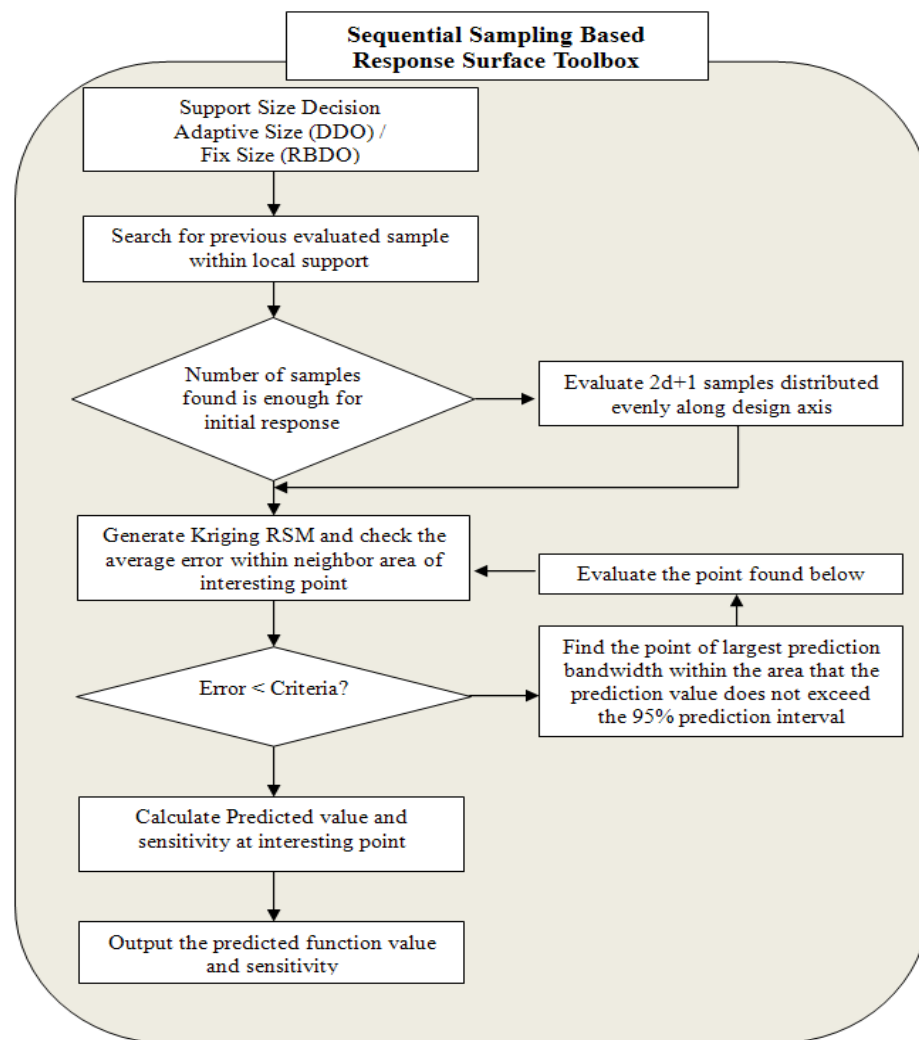


Figure 5.6 Flowchart of SS-DKG Toolbox

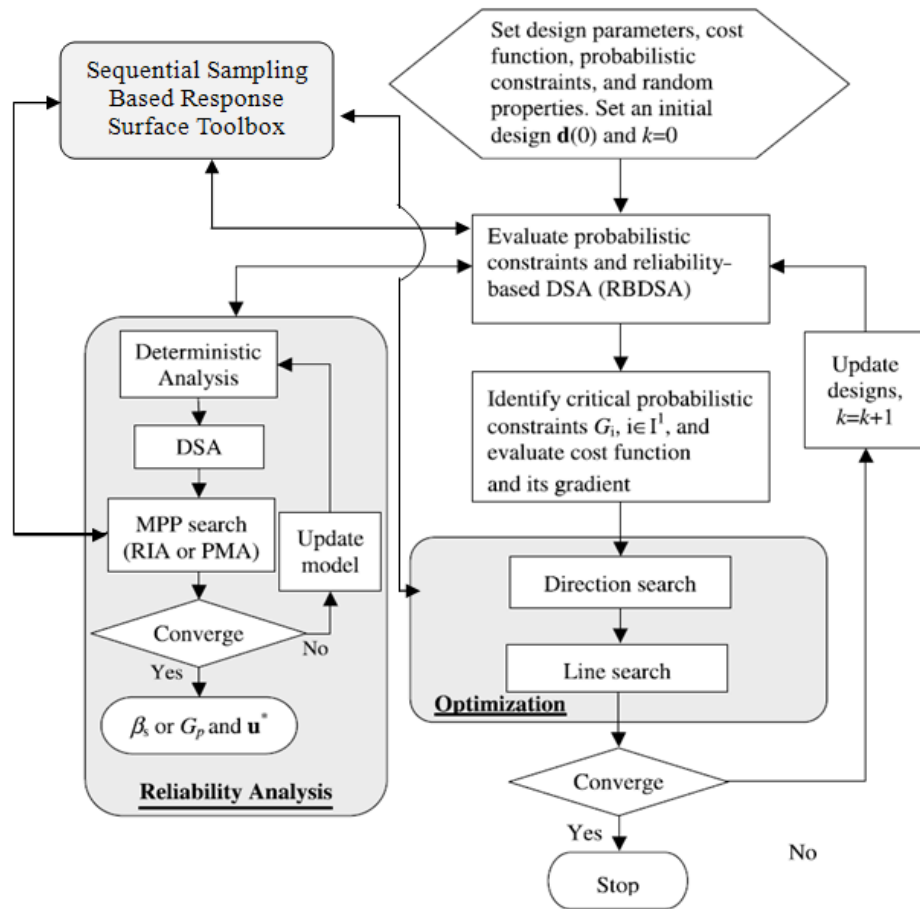


Figure 5.7 Flowchart of MPP-based RBDO Using SS-DKG

5.3 Numerical Examples

In this section, the accuracy and efficiency of the SS-DKG for RBDO is verified by comparing it with the sensitivity-based RBDO. A 2-D highly nonlinear mathematical problem and a 12-D multidimensional engineering vehicle problem are carried out by both the MPP-based RBDO using surrogate model and the MPP-based RBDO using analytical sensitivity for comparison purposes.

5.3.1 2-D Mathematical Example

A two-dimensional mathematical RBDO problem is formulated to

$$\begin{aligned}
 & \text{minimize} \quad C(\mathbf{d}) \\
 & \text{subject to} \quad P(G_i(\mathbf{X}(\mathbf{d})) > 0) \leq P_{F_i}^{\text{Tar}}, \quad i = 1, 2, 3 \\
 & \mathbf{d}^L \leq \mathbf{d} \leq \mathbf{d}^U, \quad \mathbf{d} \in \mathbb{R}^2 \text{ and } \mathbf{X} \in \mathbb{R}^2 \\
 & \text{where} \\
 & C(\mathbf{d}) = -\frac{(d_1 + d_2 - 10)^2}{30} - \frac{(d_1 - d_2 + 10)^2}{120} \\
 & G_1(\mathbf{X}) = 1 - \frac{X_1^2 X_2}{20} \\
 & G_2(\mathbf{X}) = -1 + (0.9063X_1 + 0.4226X_2 - 6)^2 + (0.9063X_1 + 0.4226X_2 - 6)^3 \\
 & \quad - 0.6(0.9063X_1 + 0.4226X_2 - 6)^4 - (-0.4226X_1 + 0.9063X_2) \\
 & G_3(\mathbf{X}) = 1 - \frac{80}{X_1^2 + 8X_2 + 5} \\
 & \mathbf{d}^L = [0, 0]^T \text{ and } \mathbf{d}^U = [10, 10]^T, \quad \mathbf{d}^{\text{initial}} = [5, 5]^T \\
 & X_i \sim N(d_i, 0.5) \text{ for } i=1, 2
 \end{aligned} \tag{5.7}$$

where the target probability of failure for each constraint is $P_{F_i}^{\text{Tar}} = 2.275\%$, $i=1\sim 3$. Since the given target probability of failure is 2.275%, the reliability index of $\beta = -\Phi^{-1}(P_{F_i}^{\text{Tar}}) = 2.0$ is selected. The properties of the random variables are shown in Table 5.2.

Table 5.2 Properties of Random Variables

Random Variables	Distribution	\mathbf{d}^L	\mathbf{d}^0	\mathbf{d}^U	Standard Deviation
X_1	Normal	0.0	5.0	10.0	0.3
X_2	Normal	0.0	5.0	10.0	0.3

In Table 5.3, it shows that both the MPP-based RBDO using analytical sensitivity and the MPP-based RBDO using surrogate model converge to the optimal design after two iterations. Numbers shown in column “Iteration” means the current design iteration and the current line search number. For example, “D.O.” means the deterministic optimum, and “1, 2” means that it is the first design with the second line search in RBDO. According to Table 5.3, we can see that the MPP-based RBDO using surrogate model can achieve the same optimum design as MPP-based RBDO using analytical sensitivity. Moreover, in this case, the number of function evaluations used in surrogate model method is smaller than the number used in MPP-based RBDO using analytical sensitivity because when the design is getting close to the optimum point more of the previously evaluated samples are used to save computational time. Hence the MPP-based RBDO using surrogate model is more efficient than MPP-based RBDO using analytical sensitivity in this case.

5.3.2 M1A1 Tank Roadarm Example

The roadarm of the M1A1 Tank is used to demonstrate the applicability of the SS-DKG-based RBDO. The roadarm is modeled as the same one in Chapter III, as shown in Fig. 3.7.

As described in Section 3.5.2, the shape design variables consist of four cross-sectional shapes of the roadarm where the widths (x_1 -direction) of the cross-sectional shapes are defined as design variables, d_1 , d_3 , d_5 , and d_7 , at intersections 1, 2, 3, and 4, respectively, and the heights (x_3 -direction) of the cross-sectional shapes are defined as design variables, d_2 , d_4 , d_6 , and d_8 . Eight shape design random variables and four random parameters for the fatigue material properties are listed in Table 5.4.

Table 5.3 Comparison Between Two RBDO Methods

Iteration	Cost	x_1	x_2	NFE
MPP-based RBDO using analytical sensitivity				
Initial	-0.8333	5.0000	5.0000	-
D.O.	-2.2918	5.1969	0.7404	9FE+9*2 SC
0,1	-2.2918	5.1969	0.7404	14FE+14*2 SC
0,2	-1.9940	4.5288	1.3894	24FE+24*2 SC
1,1	-1.8904	4.6433	1.5952	31FE+31*2 SC
1,2	-1.8957	4.6372	1.5842	34FE+34*2 SC
2,1	-1.9019	4.6706	1.5685	37FE+37*2 SC
Opt.	-1.9019	4.6706	1.5685	37FE+37*2 SC
MPP-based RBDO using Surrogate Model				
Initial	-0.8333	5.0000	5.0000	-
D.O.	-2.2913	5.1949	0.7414	18
0,1	-2.2913	5.1949	0.7414	18
0,2	-1.9517	4.8707	1.4510	21
1,1	-1.8967	4.5731	1.5873	22
1,2	-1.9018	4.6021	1.5740	22
2,1	-1.8829	4.6862	1.6084	23
2,2	-1.8866	4.6694	1.6015	24
Opt.	-1.9013	4.6692	1.5702	24

The RBDO problem for the roadarm is formulated to

$$\begin{aligned}
 & \text{minimize} && C(\mathbf{d}) \\
 & \text{subject to} && P(G_i(\mathbf{d}) > 0) \leq P_{F_i}^{\text{Tar}}, \quad i = 1, \dots, 13 \\
 & && \mathbf{d}^L \leq \mathbf{d} \leq \mathbf{d}^U
 \end{aligned}$$

where

$C(\mathbf{d})$: Weight of Roadarm (5.8)

$$G_i(\mathbf{d}) = \frac{L(\mathbf{d})}{L_t} - 1, \quad i = 1, \dots, 13$$

$L(\mathbf{d})$: Crack Initiation Fatigue Life,

L_t : Crack Initiation Target Fatigue Life: 5 years

$$P_{F_i}^{\text{Tar}} = \Phi^{-1}(-\beta_i) = \Phi^{-1}(-2), \quad i = 1, \dots, 13$$

Table 5.4 Properties of Random Variables of Roadarm

Random Variables	Lower Bound \mathbf{d}^L	Initial Design \mathbf{d}^0	Upper Bound \mathbf{d}^U	COV	Distribution Type
d_1	1.3500	1.7500	2.1500	1%	Normal
d_2	2.6496	3.2496	3.7496	1%	Normal
d_3	1.3500	1.7500	2.1500	1%	Normal
d_4	2.5703	3.1703	3.6703	1%	Normal
d_5	1.3563	1.7563	2.1563	1%	Normal
d_6	2.4377	3.0377	3.5377	1%	Normal
d_7	1.3517	1.7517	2.1517	1%	Normal
d_8	2.5085	2.9085	3.4085	1%	Normal
Fatigue Material Properties					
Non-design Uncertainties			Mean	COV	Distribution Type
Fatigue Strength Coefficient, σ			177000	3%	Log-normal
Fatigue Strength Exponent, b			-0.0730	3%	Normal
Fatigue Ductility Coefficient, ε_f			0.4100	3%	Log-normal
Fatigue Ductility Exponent, c			-0.6000	3%	Normal

For the deterministic design optimization process, 8-D surrogate model is used since the other 4 parameters are not changing in this process. Table 5.4 shows the final RBDO optimum results as well as the computational time. The computational time consists of two parts: function evaluation and sensitivity calculation. For sensitivity-based RBDO, the sensitivity calculation per design variable takes about 20% of the time that the function evaluation does in finite element analysis. Hence, the overall computational time is calculated as

$$\text{Comp. Time} = \text{Num}(\text{FE}) + 0.5 \times \text{Num}(\text{FE}) \times \text{Num}(\text{Variables}) \quad (5.9)$$

Table 5.5 shows that the MPP-based RBDO using surrogate model used more computational time to achieve both the deterministic optimum and the RBDO optimum, compared with the sensitivity-based one. The optimum designs from each method are about the same. This example shows that the MPP-based RBDO using surrogate model can achieve same accuracy as the sensitivity-based one and therefore it will be more applicable for complex practical engineering applications where the sensitivity information are not available.

Table 5.5 Comparison Between Two RBDO Methods

Design Parameters	Initial Design	Deterministic Optimization		RBDO	
		Analytical	Surrogate	Analytical	Surrogate
d_1	1.7500	1.6739	1.6552	1.7396	1.7442
d_2	3.2496	2.6496	2.6496	2.6496	2.6496
d_3	1.7500	1.9293	1.9225	1.9555	1.9601
d_4	3.1703	2.5703	2.5703	2.5703	2.5703
d_5	1.7563	1.4923	1.4771	1.5392	1.5421
d_6	3.0377	3.2992	3.2862	3.3605	3.3616
d_7	1.7517	1.6569	1.6303	1.7218	1.7291
d_8	2.9085	2.5085	2.5091	2.5085	2.5085
Cost	515.09	469.445	466.746	477.923	478.649
Active Constraints		1,3,5,8,12	1,3,5,8,12	1,3,5,8,12	1,3,5,8,12
NFE		11FE +11*8SC	200FE	317FE+ 317*12SC	1988FE
Comp. Time		28.6	200	1077.8	1988

CHAPTER VI

SAMPLING-BASED RBDO USING SURROGATE MODEL

6.1 Introduction

As shown in chapter V, when the sensitivity of the performance function in RBDO is not available, the MPP-based RBDO uses the sensitivity predicted by surrogate model. However, even though the prediction of the function value from SS-DKG is accurate enough, the sensitivity prediction could be still a challenging issue. The simple way to ensure the accuracy of the sensitivity is increasing the number of samples used for surrogate modeling, which is not cost-effective and may be even unaffordable for large dimensional applications. Therefore, another RBDO methodology that does not require the sensitivity from the surrogate model is needed. Lee [Lee et al., 2010] proposed the sampling-based RBDO which only uses the performance function value for RBDO. This sampling-based RBDO is desirable to integrate with surrogate model for complex engineering problems when the sensitivity of the performance is not available.

As briefly discussed in chapter II, this sampling-based RBDO does not require obtaining the sensitivities of the performance functions and their sensitivities from the surrogate model. Instead, stochastic sensitivity analysis using the score function, which was proposed for the independent random variables or correlated Gaussian random variables, is used to derive the sensitivities of the probabilistic. The sensitivity analysis does not require the transformation from original design space to standard normal space, which means that there is no approximation or restriction in calculating the sensitivities of the probabilistic response. The surrogate model from the SS-DKG method is used to represent the performance function accurately. Once an accurate surrogate model is

available, the Monte Carlo simulation (MCS) is applied to the surrogate model to estimate the reliability or statistical moments of the system with negligible computational burden.

In this chapter, the formulation of sampling-based RBDO using SS-DKG is explained first. Then the efficiency strategies applied in this RBDO procedure are explained in detail. Finally, two numerical examples are used to demonstrate the accuracy and efficiency of the sampling-based RBDO using surrogate model.

6.2 Sampling-Based RBDO Using Sequential-Sampling-Based Dynamic Kriging

The mathematical formulation of a general RBDO problem is expressed as

$$\begin{aligned} & \text{minimize} && \text{Cost}(\mathbf{d}) \\ & \text{subject to} && P[G_i(\mathbf{X}) > 0] \leq P_{F_i}^{\text{Tar}}, \quad i = 1, \dots, NC \\ & && \mathbf{d}^L \leq \mathbf{d} \leq \mathbf{d}^U, \quad \mathbf{d} \in \mathbf{R}^m \text{ and } \mathbf{X} \in \mathbf{R}^{nrv} \end{aligned} \quad (6.1)$$

where $\mathbf{d} = \{d_i\}^T = \boldsymbol{\mu}(\mathbf{X})$ is the design vector, which is the mean value of the m -dimensional random vector $\mathbf{X} = \{X_1, X_2, \dots, X_m\}^T$; $P_{F_i}^{\text{Tar}}$ is the target probability of failure for the i^{th} constraint; and NC , m , and nrv are the number of probabilistic constraints, design variables, and random variables, respectively.

A reliability analysis, which is used to evaluate the probabilistic constraints in Eq. (6.1), involves calculation of the probability of failure, denoted by P_F , which is defined using a multi-dimensional integral

$$P_F(\boldsymbol{\mu}) \equiv P[\mathbf{X} \in \Omega_F] = \int_{\mathbf{R}^N} I_{\Omega_F}(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}; \boldsymbol{\mu}) d\mathbf{x} = E[I_{\Omega_F}(\mathbf{X})] \quad (6.2)$$

where $\boldsymbol{\mu}$ is the mean of the random input $\mathbf{X} = \{X_1, \dots, X_N\}^T$, $P(\cdot)$ represents a probability measure, Ω_F is the failure set, $f_{\mathbf{X}}(\mathbf{x}; \boldsymbol{\mu})$ is a joint probability density function (PDF) of \mathbf{X} , and $E[\bullet]$ represents the expectation operator. The failure set is defined as $\Omega_F \equiv \{\mathbf{x} : G_i(\mathbf{x}) > 0\}$ for component reliability analysis of the i^{th} constraint function $G_i(\mathbf{x})$. As explained in Section 2.5.1, $I_{\Omega_F}(\mathbf{x})$ in Eq. (6.2) is called an indicator function and is defined as

$$I_{\Omega_F}(\mathbf{x}) \equiv \begin{cases} 1, & \mathbf{x} \in \Omega_F \\ 0, & \text{otherwise} \end{cases} \quad (6.3)$$

Following the same procedure as in Section 2.5.2, the sensitivity of the probability of failure P_F with respect to a design variable μ_i can be obtained as

$$\begin{aligned} \frac{\partial P_F(\boldsymbol{\mu})}{\partial \mu_i} &= \int_{\mathbb{R}^N} I_{\Omega_F}(\mathbf{x}) \frac{\partial f_{\mathbf{X}}(\mathbf{x}; \boldsymbol{\mu})}{\partial \mu_i} d\mathbf{x} = \int_{\mathbb{R}^N} I_{\Omega_F}(\mathbf{x}) \frac{\partial \ln f_{\mathbf{X}}(\mathbf{x}; \boldsymbol{\mu})}{\partial \mu_i} f_{\mathbf{X}}(\mathbf{x}; \boldsymbol{\mu}) d\mathbf{x} \\ &= E \left[I_{\Omega_F}(\mathbf{x}) \frac{\partial \ln f_{\mathbf{X}}(\mathbf{x}; \boldsymbol{\mu})}{\partial \mu_i} \right] \end{aligned} \quad (6.4)$$

since $I_{\Omega_F}(\mathbf{x})$ is not a function of μ_i . The partial derivative of the log function of the joint PDF in Eq. (6.4) with respect to μ_i is known as the first-order score function for μ_i and is denoted as

$$s_{\mu_i}^{(1)}(\mathbf{x}; \boldsymbol{\mu}) \equiv \frac{\partial \ln f_{\mathbf{X}}(\mathbf{x}; \boldsymbol{\mu})}{\partial \mu_i}. \quad (6.5)$$

For statistically independent random variables, the first-order score function for μ_i is expressed as

$$s_{\mu_i}^{(1)}(\mathbf{x}; \boldsymbol{\mu}) \equiv \frac{\partial \ln f_{\mathbf{X}}(\mathbf{x}; \boldsymbol{\mu})}{\partial \mu_i} = \frac{\partial \ln f_{X_i}(x_i; \mu_i)}{\partial \mu_i} \quad (6.6)$$

and, for a correlated bivariate random input, it is expressed as

$$s_{\mu_i}^{(1)}(\mathbf{x}; \boldsymbol{\mu}) \equiv \frac{\partial \ln f_{\mathbf{X}}(\mathbf{x}; \boldsymbol{\mu})}{\partial \mu_i} = \frac{\partial \ln c(u, v; \theta)}{\partial \mu_i} + \frac{\partial \ln f_{X_i}(x_i; \mu_i)}{\partial \mu_i} \quad (6.7)$$

where c is a copula density function, $u = F_{X_i}(x_i; \mu_i)$ and $v = F_{X_j}(x_j; \mu_j)$ are cumulative distribution functions (CDFs) for X_i and X_j , respectively, and θ is the correlation coefficient. Each term in Eqs. (6.6) and (6.7) is listed in Tables 6.1 and 6.2, respectively.

Table 6. 1 First-Order Score Function for μ_i for Independent Random Variables

Marginal Distribution	First-Order Score Function, $s_{\mu_i}^{(1)}(\mathbf{x}; \boldsymbol{\mu})$
Normal	$\frac{x_i - \mu_i}{\sigma_i^2}$
Log-normal	$-\frac{1}{\bar{\sigma}_i} \frac{\partial \bar{\sigma}_i}{\partial \mu_i} + \frac{1}{\bar{\sigma}_i^2} \left(\frac{\ln x_i - \bar{\mu}_i}{\bar{\sigma}_i} \right) \times \left[\bar{\sigma}_i \frac{\partial \bar{\mu}_i}{\partial \mu_i} + (\ln x_i - \bar{\mu}_i) \frac{\partial \bar{\sigma}_i}{\partial \mu_i} \right]$
Gumbel	$\alpha_i - \alpha_i e^{-\alpha_i(x_i - v_i)}$
Weibull	$\frac{1}{k_i} \frac{\partial k_i}{\partial \mu_i} - \frac{1}{v_i} \frac{\partial v_i}{\partial \mu_i} + \frac{\partial k_i}{\partial \mu_i} \ln \frac{x_i}{v_i} - \frac{(k_i - 1)}{v_i} \frac{\partial v_i}{\partial \mu_i} - \left(\frac{x_i}{v_i} \right)^{k_i} \left(\frac{\partial k_i}{\partial \mu_i} \ln \frac{x_i}{v_i} - \frac{k_i}{v_i} \frac{\partial v_i}{\partial \mu_i} \right)$

Since the SS-DKG method is used to generate surrogate models for $G_i(\mathbf{X})$, the MCS can be used for both probability of failure and its sensitivity estimation with negligible computational burden. By applying the MCS to the surrogate model denoted by $\hat{G}_i(\mathbf{X})$, the probabilistic constraints in Eq. (6.1) can be approximated as

$$P_{F_i}(\boldsymbol{\mu}) \equiv P[G_i(\mathbf{X}) > 0] \cong \frac{1}{K} \sum_{k=1}^K I_{\hat{\Omega}_F}(\mathbf{x}^{(k)}) \leq P_{F_i}^{\text{Tar}} \quad (6.8)$$

Table 6. 2 Log-derivative of Copula Density Function

Copula Type	$\frac{\partial \ln c(u, v; \theta)}{\partial \mu_i}$
Clayton	$\left(-\frac{1+\theta}{u} + \frac{(2\theta+1)u^{-(1+\theta)}}{u^{-\theta} + v^{-\theta} - 1} \right) \frac{\partial u}{\partial \mu_i}$
AMH	$\left[\frac{\theta^2(1-v) + \theta(v+1)}{1 - \theta^2(1-u)(1-v) - \theta(2-u-v-uv)} - \frac{3\theta(1-v)}{1 - \theta(1-u)(1-v)} \right] \frac{\partial u}{\partial \mu_i}$
Frank	$\theta \left[\frac{2(e^{\theta(1+u)} - e^{\theta(u+v)})}{e^\theta - e^{\theta(1+u)} - e^{\theta(1+v)} + e^{\theta(u+v)}} + 1 \right] \frac{\partial u}{\partial \mu_i}$
FGM	$\left[\frac{2\theta(2v-1)}{1 + \theta(1-2u)(1-2v)} \right] \frac{\partial u}{\partial \mu_i}$
Gaussian	$\left[\frac{\Phi^{-1}(u)}{\phi(\Phi^{-1}(u))} + \frac{\theta\Phi^{-1}(v) - \Phi^{-1}(u)}{\phi(\Phi^{-1}(u))(1-\theta^2)} \right] \frac{\partial u}{\partial \mu_i}$
Independent	0

where K is the MCS sample size, $\mathbf{x}^{(k)}$ is the k^{th} realization of \mathbf{X} , and the failure set $\hat{\Omega}_F$ for the surrogate model is defined as $\hat{\Omega}_F \equiv \{\mathbf{x} : \hat{G}_i(\mathbf{x}) > 0\}$. Sensitivity of the probabilistic constraint is obtained using the score function as

$$\frac{\partial P_{F_i}}{\partial \mu_i} \cong \frac{1}{K} \sum_{k=1}^K I_{\hat{\Omega}_F}(\mathbf{x}^{(k)}) s_{\mu_i}^{(1)}(\mathbf{x}^{(k)}; \boldsymbol{\mu}) \quad (6.9)$$

where $s_{\mu_i}^{(1)}(\mathbf{x}^{(k)}; \boldsymbol{\mu})$ is obtained using Eqs. (6.6) and (6.7) for independent and correlated random input variables, respectively. From Eq. (6.9), it can be seen that the stochastic sensitivity analysis does not require sensitivities of performance functions, which are represented by surrogate models.

For the sampling-based RBDO, the local window is used to generate a surrogate model instead of the global window. The SS-DKG method is well suited for the local

window by selecting an optimum set of basis functions to represent the response. That is, a different set of basis functions is selected to represent the response at different local window. For the initial sampling, Latin Centroidal Voronoi Tessellations (LCVT) is utilized to uniformly fill the local window, and then, additional samplings to satisfy the accuracy condition of the surrogate model are inserted using the sequential sampling explained in Chapter IV. The entire flowchart of sampling-based RBDO using SS-DKG is shown in Fig.6.1 .

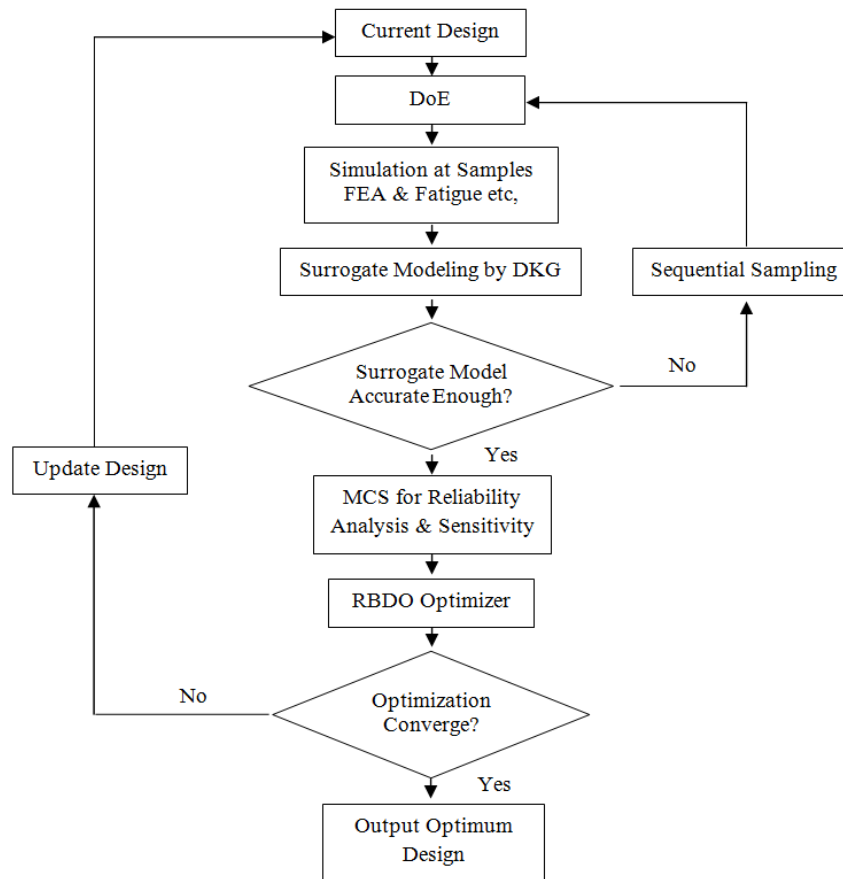


Figure 6. 1 Sampling-Based RBDO Using SS-DKG

6.3 Efficiency Strategies for Multiple Constraints and Optimization

For a typical RBDO problem, usually there are a number of constraints, and the FEA simulation at samples is compute-intensive. Therefore, several efficiency strategies are applied to reduce the total computational time for RBDO procedure.

6.3.1 Sample Reuse Strategy

To ensure efficiently obtaining an accurate surrogate model from SS-DKG method, the surrogate model is always generated within a local design window instead of the global design domain as shown in Fig. (6.2). The local window is defined as

$$\{\mathbf{x}(x_1, x_2, \dots, x_m) \mid |x_i - d_i| \leq c\sigma_i\beta_i^i, i = 1, \dots, m\} \quad (6.10)$$

where c is a constant between 1.2~1.5, σ_i and β_i^i are the standard deviation and the reliability index of the design variable x_i , respectively.

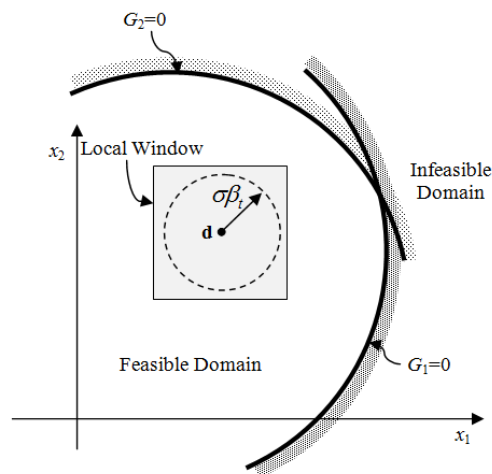


Figure 6. 2 The Local Window for Surrogate Model

To generate the surrogate model, N_r initial samples are needed. However, as the RBDO procedure is continuing, there might be some existing samples within the current local window when the design moves. The sample reuse strategy will first search for the local window to see if there is enough number of existing samples or not. If the number of existing samples N_e is enough, which means $N_e > N_r$, then there will be no additional sampling for this local window; otherwise, $N_r - N_e$ samples will be inserted into the local window.

Moreover, when the current candidate design is close to the previous candidate design, usually there are a number of samples that are surrounding but not within the current local window. It is obvious that these existing samples are also valuable and should be used for the surrogate modeling for current design. In such cases, it is proposed to adjust the current local window size to include more existing samples. In this study, it is decided to increase the local window size by increasing c value in Eq. (6.10) from current value to 1.5 if $N_e > 0.9 \times N_r$.

6.3.2 Using Violated Constraints for Accuracy Check

As shown in Fig. (6.1), the accuracy check for the surrogate model will be carried out first to ensure there is enough number of samples within the local window to generate the accurate surrogate models. However, in practical engineering applications, there might be several constraints that are satisfied since the current design is far away from those limit states. In this case, generating the surrogate models for them would waste the computational time. Therefore, these “safe” constraints are ignored in the accuracy check and not used in sensitivity analysis and only those violated constraints are used.

Since the true responses of the constraints within current local window are only available at the sample points, a constraint is defined as violated if it is larger than zero at any sample point, expressed as

$$\text{there exist } i \in \{1, 2, \dots, N_d\}, \text{ such that } G(\mathbf{x}^{(i)}) > 0 \quad (6.11)$$

where $\mathbf{x}^{(i)}$ is the i^{th} samples in the local window.

It is worth mentioning that the status of the violated constraint may change as the design point moves during iteration. Some inactive constraints could become active or violated during the next iteration.

6.3.3 Adaptive Initial Point for Correlation Parameter

Estimation

When applying the sampling-based RBDO to complex engineering problems, the number of variables used for surrogate modeling could be large. In such cases, the pattern search for DKG may become time-consuming. Therefore, how to efficiently carry out the pattern search process for correlation parameter needs to be found. It is noted that the computational time of pattern search is strongly affected by the initial search point. If the pattern search starts from the neighboring area of the true optimum, it can find the optimum within much shorter time compared with starting from a point far away from the optimum. Moreover, it is also obvious that even when the candidate design point moves during the RBDO procedure, the surrogate models are generated for the same performance functions. Therefore, one can assume that the optimal correlation parameter found from DKG in the previous iteration is close to the current optimal one and thus a good the initial point for the current pattern search.

One engineering example is used to demonstrate the improvement of using the previous optimum value as the initial searching point for pattern search in the current local window. The example is the same 12-D M1A1 roadarm example used in Chapter V. Constraint #1 is used for surrogate modeling generation. 400 samples are used in the current local window. As shown in Table 6.3, if the lower bound of the correlation parameter θ which is $\theta_L = [0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01]^T$ is used as the initial searching point, the pattern search uses 1333 function calls to find the optimum value and the time spent is 123.4s. If the optimal θ_{prev}^{opt} value from previous iteration is used as the initial searching point, the pattern search uses 763 function calls to find the optimum and the computational time is reduced to 70.8s.

Table 6. 3 Improvement of Efficiency Using Different Initial Searching Points

Init θ	Optimum θ	$\psi(\theta)$	No. of FE	Time (s)
θ_L	[0.01, 0.01, 0.095938, 0.1018, 0.01, 0.086172, 0.01, 0.01, 0.13891, 0.27172, 0.01, 0.01]	0.010806	1333	123.4
θ_{prev}^{opt}	[0.011953, 0.01, 0.09594, 0.1018, 0.01, 0.088125, 0.01, 0.01, 0.1389, 0.27563, 0.01, 0.01]	0.010941	763	70.8

6.4 Numerical Examples

6.4.1 2-D Mathematical Example

To see how the proposed sampling-based RBDO using SS-DKG works, consider a 2-D mathematical RBDO problem, which is formulated as Eq. (5.7). The properties of two random variables are shown in Table 5.2, and they are assumed to be independent.

As shown in Eq. (6.12), the target probability of failure (P_F^{Tar}) is 2.275% for all constraints.

The design optimization initiates from the initial design (5, 5). As shown in Fig. 6.3, with 3 iterations and 40 samples, the deterministic design optimization is completed first. Then the sampling-based RBDO process starts. Without additional samples, the RBDO optimum is successfully obtained. Table 6.4 compares the numerical results of five different RBDO methods. The first three results are obtained from the so-called MPP-based RBDO using analytical sensitivity, which requires the sensitivity of the constraint functions for the MPP search and design optimization. The MPP-based RBDO includes the FORM, the DRM with three quadrature points, which is denoted as DRM3, and the DRM with five quadrature points, which is denoted as DRM5 in Table 6.4. The last two results are obtained from the sampling-based RBDO, which uses the MCS for the estimation of the probability of failure and its sensitivity. The sampling-based RBDO using the SS-DKG method is the proposed method, and for the comparison of the accuracy of the proposed method, the result of the sampling-based RBDO using the true functions given in Eq. (6.11) is also shown in the table.

From Table 6.4, it can be seen that the probability of failure of the second constraint (1.3438%) at the optimum design obtained using the FORM is not close to the target probability of failure (2.275%). This is because the second constraint is highly nonlinear as shown in Fig. 6.3 and the FORM cannot accurately estimate the probability of failure of highly nonlinear functions. To improve the accuracy of the probability of failure at the optimum design, the MPP-based DRM with three or five quadrature points can be used.

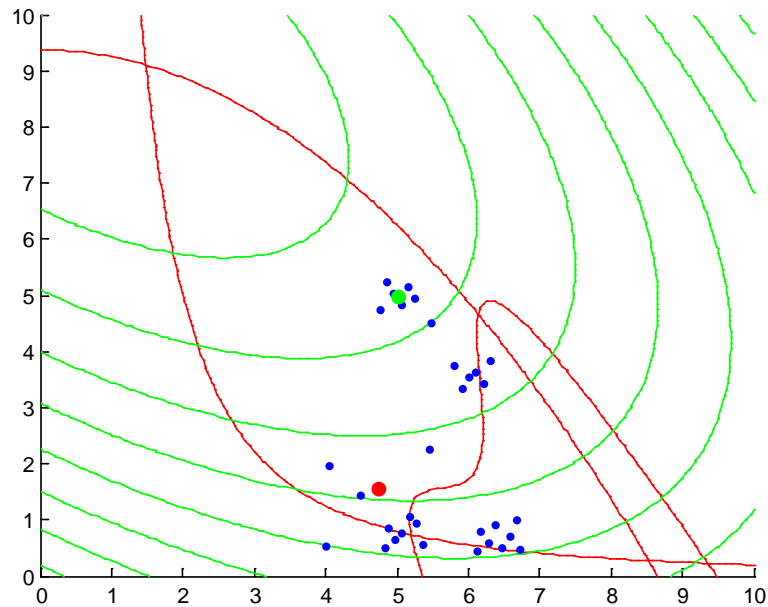


Figure 6. 3 RBDO History and Optimum Design

Table 6.4 shows that the MPP-based DRM indeed improves the accuracy of the probability of failure at the optimum design with more function evaluations.

Table 6. 4 Comparison of Various RBDOs ($P_F^{Tar} = 2.275\%$)

Methods	Cost	Optimum Design	MCS		Number of Function Evaluations	
			P_{F_1}	P_{F_2}		
MPP-Based RBDO	FORM	-1.9018	4.6709, 1.5689	2.4886	1.3438	52+52
	DRM3	-1.9048	4.7188, 1.5594	2.2934	1.9318	128+106
	DRM5	-1.9070	4.7307, 1.5539	2.2868	2.1620	146+102
Sampling-Based RBDO	SS-DKG	-1.9077	4.7316, 1.5530	2.2618	2.1835	50
	Anal.	-1.9081	4.7385, 1.5512	2.2482	2.3471	N.A.

To obtain the optimum design, the FORM uses 52 function evaluations and 52 sensitivity calculations, whereas the MPP-based DRM with five quadrature points uses 146 function evaluations and 102 sensitivity calculations,. The number of function evaluations for the MPP-based DRM will be increased as the number of quadrature points increases.

On the other hand, the sampling-based RBDO, which uses the SS-DKG method and stochastic sensitivity analysis, requires only 33 samples for the accurate optimum design. Without the sensitivity of the performance functions, the sampling-based RBDO can obtain a very accurate optimum design, and the optimum design is very close to the optimum design obtained using the true functions. This means that the SS-DKG method generates very accurate surrogate models for the true functions. From this example, it can be said that once accurate surrogate models are available, the proposed sampling-based RBDO yields very accurate optimum designs with good efficiency.

6.4.2 M1A1 Tank Roadarm Example

The roadarm of a tracked vehicle, as described in Section 5.3.2, is used to demonstrate the applicability of the SS-DKG for sampling-based RBDO again.

From Table 6.5, it is found that the sampling-base RBDO using SS-DKG can obtain the same optimum design compared with the MPP-based RBDO using the analytical sensitivity information. Moreover, the sampling-based RBDO using SS-DKG uses less number of samples as used in Section 5.3.2.

Table 6. 5 Comparison Between Different RBDO Methods

Design Parameters	Initial Design	Deterministic Optimization		MPP-Based RBDO	Sampling-Based RBDO
		SEN	SS-DKG	SEN	SS-DKG
d_1	1.750	1.653	1.655	1.747	1.745
d_2	3.250	2.650	2.650	2.650	2.650
d_3	1.750	1.922	1.922	1.955	1.954
d_4	3.170	2.570	2.570	2.570	2.570
d_5	1.756	1.478	1.477	1.543	1.540
d_6	3.038	3.287	3.288	3.362	3.367
d_7	1.752	1.630	1.630	1.729	1.722
d_8	2.908	2.508	2.508	2.508	2.508
Cost	515.09	466.80	466.84	478.59	478.18
Active Constraints		1,3,5,8,12	1,3,5,8,12	1,3,5,8,12	1,3,5,8,12
NFE		11FE +11*8SC	200FE	317FE+ 317*12SC	1857FE
Comp. Time		28.6	200	1077.8	1857

CHAPTER VI

PARALLELIZATION OF RBDO USING SURROGATE MODEL

7.1 Introduction

With the successfully implement of using SS-DKG for sampling-based RBDO to engineering application as shown in Section 6.4, the methodology can be used for practical engineering application problems. However, there is one critical issue found during the testing problem in Section 6.4. That is, the computational time increases dramatically when the number of samples evaluated for FEA and fatigue analysis increases. Moreover, the total computational time for surrogate modeling and Monte Carlo simulation may also be very large or even unaffordable if the number of design variables and constraints are large for complex applications. Therefore, a high performance computing strategy needs to be implemented into the surrogate model based RBDO procedure to ensure it is applicable for large-scale complex applications.

During the last two decades, the high performance computing (HPC) technology has been rapidly developed. Usually HPC refers to the engineering application of cluster-based computing or multi-processor based computing. In the cluster-based computing process, a computing task is submitted into a master machine. The master machine converts the task into several sub-tasks and sends them to the machines on the cluster network which are called “clients”. Each of the client machines computes the sub-task simultaneously and returns the results after the computing to the master machine. The master machine finalizes the results from the client machine and outputs the final result to user in the end. In the multi-processor based computing process, there is only one machine that contains multiple processors and memory units. The main task is converted

into several sub-tasks by the master processor, computed by other processors simultaneously and finalized by the master processor in the end.

Usually, two different ways of HPC are applied to different problems depending on the cost of computing resources. The cluster-based HPC can be cost-effective since the cost of the client machines can be relatively cheaper than the cost of a multi-processor machine at the same computing performance. However, the multi-processor based HPC can be more efficient since all data are transferred within the same machine whereas the cluster-based one is transferring the data on the network where communication time between the master and the client machines is also a concern.

For the sampling-based RBDO using SS-DKG, both of the two HPC methods mentioned above are needed to handle different parallel computing process. In this chapter, the toolbox used for HPC in Matlab is introduced first; the parallelization of the sampling-based RBDO using SS-DKG is explained in details; and finally a numerical example is used to show the improvement on efficiency by using the HPC.

7.2 Parallel Computing Toolbox in Matlab

Parallel Computing Toolbox in Matlab lets users solve computationally- and data-intensive problems using Matlab® and Simulink® on cluster-based and multiprocessor-based computers. Parallel processing constructs such as parallel for-loops, distributed arrays, parallel numerical algorithms, and message-passing functions to let user implement task- and data-parallel algorithms in Matlab at a high level language without programming for specific hardware and network architectures. As a result, converting serial Matlab applications to parallel Matlab applications requires few code modifications

and no programming in a low-level language is required. Users can run their applications interactively or offline batch environments.

Users can use the toolbox to execute applications on a single multiprocessor desktop. Without changing the code, users can run the same application on a computer cluster, grid, or cloud computing service (using Matlab Distributed Computing Server). Parallel Matlab applications can be distributed as executables or shared libraries (using Matlab Compiler) that can access Matlab Distributed Computing Server. The workflow of using the parallel computing toolbox of Matlab is shown in Fig. 7.1.

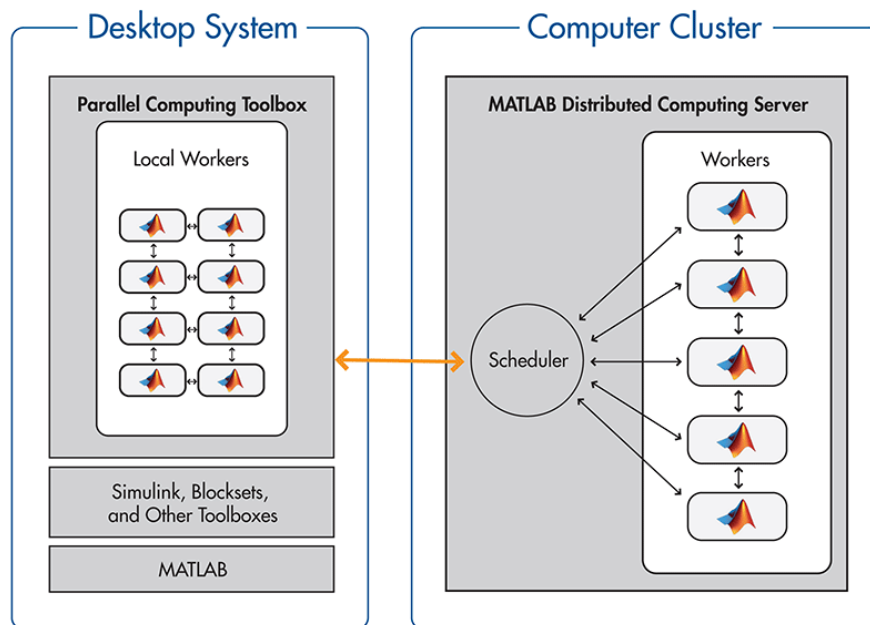


Figure 7. 1 Parallel Computing Toolbox in Matlab

In summary, it has the key features that are useful for a quick setup to implement the HPC as follows:

- Support data-parallel and task-parallel application development;

- Ability to annotate code segments with parfor (parallel for-loops) and spmd (single program multiple data) for implementing task- and data-parallel algorithms, respectively;
- Have high-level constructs such as distributed arrays, parallel algorithms, and message-passing functions for processing large data sets on multiple processors;
- Ability to run eight workers locally on a multicore desktop;
- Integration with MATLAB Distributed Computing Server for cluster- and grid-based applications that use any scheduler or any number of workers;
- Interactive and batch execution modes.

7.3 Parallelization in RBDO

With the parallel computing toolbox available, we will discuss the details about where the parallelization is needed in the proposed sampling-based RBDO using SS-DKG method.

7.3.1 Parallelization of Genetic Algorithm in DKG

As discussed in Section 3.3, the DKG uses a genetic algorithm to select the best selection of the basis functions. In each generation during the GA process, there are $2C_{m+P}^P + P + 1$ candidate selections that need to be evaluated for the fitness value. When the number of samples increases, the highest order number P increases and $2C_{m+P}^P + P + 1$ increases in a faster way and may become compute-intensive for large-scale problems. Therefore, parallelization of the evaluation for the fitness value in each generation is needed. Matlab provides a built-in option for the built-in genetic algorithm if the parallel computing toolbox is available. User can turn on that option for parallel computing

without any additional effort. Since the cluster-based HPC have additional time spent on the communication between the master machine and the client machines, to ensure the computing efficiency, this parallelization for GA is conducted on a 32-cores Linux server.

7.3.2 Parallelization of Surrogate Models for Multiple Constraints in RBDO

As discussed in Section 6.2, a typical RBDO problem contains a number of constraints. Since the surrogate model from SS-DKG is required for each violated constraint and the pattern search procedure may become compute-intensive for large-scale applications, it is desirable to carry out the surrogate modeling for all active or violated constraints simultaneously. Therefore, the parallelization of surrogate modeling for multiple constraints is needed, especially for large-scale applications.

Since Matlab requires license on each of the client machine for cluster-based HPC, to carry out the parallelization cost-effectively, the parallelization of the surrogate modeling is conducted on the 32-cores Linux server too in this study. It is worth mentioning that the Matlab parallel computing toolbox allows only 8 cores working simultaneously, 8 surrogate models for the active or violated constraints are generated at the same time and the total computational time will still be significantly reduced.

7.3.3 Parallelization of MCS in Reliability Analysis

As discussed in Section 6.2, the Monte Carlo simulation (MCS) is used to calculate the probability of failure of the performance functions as well as the sensitivity of probabilistic constraints. For accuracy, the number of MCS sample usually takes a

large number, i.e. one million. Moreover, since the prediction from DKG at the MCS samples are not explicit according to Eq. (3.43), which means that to calculate the prediction at a point of interest, a large dimensional matrix calculation is involved. When the number of samples used for surrogate modeling is large, the total computational time for the prediction at the MCS samples becomes large too. Therefore, the parallelization of the MCS procedure is also needed to reduce the large computational time. This parallelization is also conducted on the 32-cores Linux server with 8 cores working simultaneously by using the Matlab parallel computing toolbox.

7.3.4 Parallelization of Computer Aided Engineering Analysis (CAE)

To generate the surrogate models by SS-DKG for the constraints in RBDO, it is required to evaluate the performance function at the samples, which is usually conducted by computer aided engineering analysis (CAE) such as finite element analysis, fatigue analysis, etc. This procedure is usually compute-intensive too. If the number of samples is large for a large-scale engineering application, the total computational time of conducting CAE for all samples may become unaffordable. Therefore, the parallelization is necessary for this procedure. Usually the CAE is carried out by the general CAE commercial software. The parallelization can be done on the cluster-based HPC by the parallel computing platform (LSF-Platform) in Linux/Unix server directly. Unlike the 8-max-cores restriction for a multi-processor computer in Matlab parallel computing toolbox, the number of parallel computing session controlled by LSF-Platform under Linux/Unix is not restricted. Therefore, one can use as many machines on the cluster network as possible.

7.3.5 Summary of Parallelization in Sample-Based RBDO

Using SS-DKG

With the discussions from Sections 7.3.1~7.3.5, we can obtain the overall workflow of the parallelization in the sampling-based RBDO using SS-DKG which is shown in Fig. 7.2.

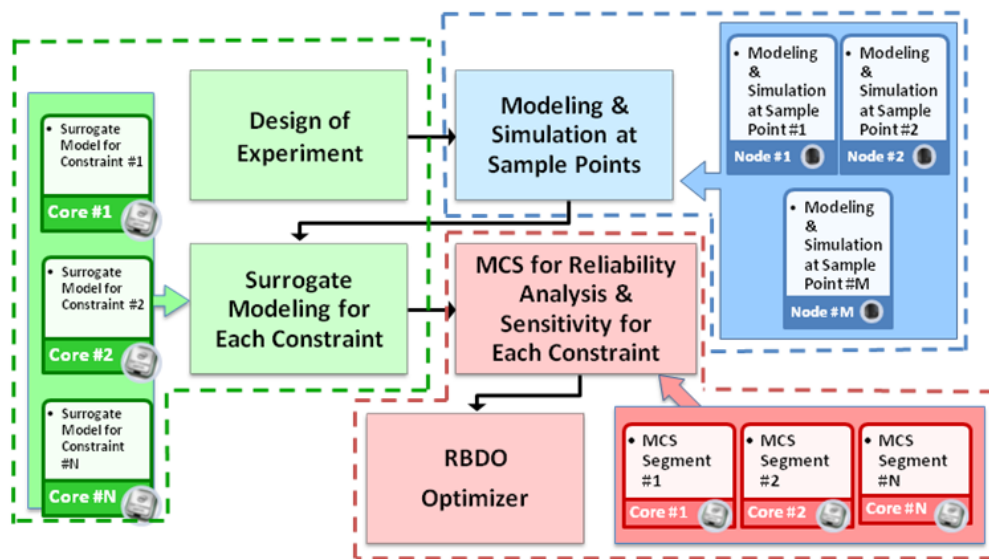


Figure 7. 2 Workflow of Parallelization in RBDO

7.4 Numerical Examples

To demonstrate the improvement of the efficiency by applying the parallelization to the sampling-based RBDO using SS-DKG, the same 12-D M1A1 tank roadarm example is used. The comparison between the parallel computing and the original serial computing is carried out by running the reliability analysis at the initial design point with 400 points in the local window. Nine constraints are identified as active or violated

constraints and surrogate models are generated for these 9 constraints. MCS with 500000 samples are evaluated based on the surrogate models for these 9 constraints. The Matlab parallel computing toolbox is used on the 32-cores Linux server and the maximum 8 workers are used for parallel computing.

As shown in Table 7.1, the parallel computing indeed reduces the computational time by 64.6% for surrogate modeling and 89.3% for the MCS. The reason that the reduction of the computational time for the surrogate modeling is not exact 8 times compared with the serial one is that there are nine constraints are used to generate the surrogate models while eight cores are used. Therefore, it includes 2 iteration for the parallel computing, 1st iteration for constraint 1~8 and 2nd iteration for constraint 9 only. On the other hand, the MCS is parallelized based on the number of MCS samples; therefore, the reduction of the computational time is approximately proportional to the number of cores used.

Table 7. 1 Comparison of Computational Time

	No. of Sample	No. of Constraint	Surrogate Modeling (s)	MCS (s)	Total Time (s)
Parallel	400	9	872.7	89.2	965.9
Serial			2468.3	838.1	3322.4

CHAPTER VIII
CONSERVATIVE SURROGATE MODEL USING WEIGHTED
KRIGING VARIANCE FOR SAMPLING-BASED RBDO

8.1 Introduction

In sampling-based RBDO, Monte Carlo simulation (MCS) is used to carry out both the probability of failure and the stochastic sensitivity analysis. To carry out MCS efficiently in complex engineering applications, a surrogate model is used to predict the performance function at the MCS samples. When the surrogate model is not accurate enough, new samples are sequentially inserted to improve the prediction accuracy. However, inserting a large number of new samples sequentially may not be applicable when the samples are from the physical experiments or when the computational resource for simulation is limited. Therefore, the improvement of the surrogate model cannot be obtained by simply increasing the sample numbers. When applying the surrogate model for reliability-based design optimization when the number of samples is small, to assure the obtained optimum design can satisfy the probabilistic constraints, a conservative surrogate model is needed. Picheny [Picheny, 2009] used both the safety margin and safety factor approaches to construct conservative surrogate models and concluded that when the Kriging method is used, both methods provide similar performance in terms of the conservativeness and the accuracy. Hertog et al. [Hertog, et al., 2006], Luna and Young [Luna and Young, 2003] used the bootstrapping method to estimate the Kriging prediction interval to construct a conservative surrogate model. The bootstrapping variance is larger than the traditional Kriging prediction variance by considering the uncertainty from the correlation parameter in the Kriging method. However, the

bootstrapping procedure is time-consuming and is not applicable for high-dimensional problems. Viana et al. [Viana et al, 2010] used cross-validation to estimate the safety margin for the conservative surrogate model. While the cross-validation error is widely used to estimate the prediction error, this constant safety margin approach does not distinguish the prediction error at different sample locations and often yields an over-conservative surrogate model where the samples are aggregated around and under-conservative surrogate model where the samples are sparse.

In this study, a weighted Kriging variance is considered to construct a more accurate conservative surrogate model based the sample locations. To evaluate the importance from each sample and determine the weight for it, an accuracy measure of the surrogate model is needed first. Under the Kriging framework, the Akaike information criterion (AIC) [Akaike, 1974] is a reliable indicator to assess the accuracy of the surrogate model. Hurvich and Tsai [Hurvich and Tsai, 1989] proposed a corrected AIC (AICc) to correct the bias in AIC when the number of samples is small. Burnham and Anderson [Burnham and Anderson, 2002] recommended using AICc rather than AIC when the sample size is small and showed that the AICc converges to AIC as the number of sample increases. Martin and Simpson [Martin and Simpson, 2005] compared the performance of the AICc assessment with other methods and concluded that AICc provides the best accuracy assessment for surrogate model accuracy. In this study, the AICc is used to assess the accuracy changes in the surrogate model during the cross-validation process, and an importance function value is assigned to each sample according to the relative changes in AICc of the surrogate model and the weight is also assigned to each sample according to the importance function values. Then a weighted

Kriging variance is calculated based on the weight values for each sample. By applying this weighted Kriging variance, one can construct the conservative surrogate model for dynamic Kriging and use it for sampling-based RBDO to assure the obtained optimum design will satisfy the probabilistic constraint.

The remainder of this chapter is organized in three sections. First, the background of the AICc is briefly summarized. Then, the weighted Kriging variance using the weight from the relative change in AICc is introduced to construct the conservative surrogate model. Third, the conservative surrogate model is applied to sampling-based RBDO and the optimum design is verified for the probabilistic constraint using numerical examples and compared with the results using the constant safety margin approach.

8.2 Sampling-Based RBDO using Conservative Surrogate Model

8.2.1 Corrected Akaike Information Criterion (AICc)

The AIC is originally proposed to evaluate the quality of a model in statistics based on the log-likelihood function. It is a measure of the relative goodness of fit of a statistical model, and in general expressed as

$$AIC = 2k - 2\ln(L) \quad (8.1)$$

where k is the number of parameters in the statistical model and L is the maximized value of the likelihood function for the estimated model. In the Kriging framework discussed in Chapter 3, the k is the number of dimension of \mathbf{X} and L is the likelihood function. When the sample size is small, i.e., $n/k < 40$ (n is the number of sample), which is often the case

in engineering problems, the AICc is used instead to provide a unbiased estimation, which is expressed as

$$AICc = AIC + \frac{2k(k+1)}{n-k-1} \quad (8.2)$$

8.2.2 Weighted Kriging Prediction Variance for Conservative Surrogate Model

In Eq. (3.22), the Kriging prediction variance σ_p^2 is calculated and the $C\%$ upper bound of the prediction interval is

$$y_{\text{var}} = y_{\text{krig}} + \Phi^{-1}\left(\frac{C}{100}\right)\sigma_p \quad (8.3)$$

where $\Phi^{-1}(\bullet)$ is the inverse CDF of standard normal random variable. This prediction upper is usually used as the $C\%$ level conservative surrogate model for Kriging and often times it is considered as a variable safety margin approach for constructing the conservative surrogate model. Since Kriging is an interpolation method, the prediction variance σ_p^2 becomes zero at the sample point. As a result, the conservative surrogate model y_{var} is the same as the y_{krig} at the sample points. This interpolation property of Kriging causes trouble when the conservative surrogate model is used for optimization problem. Consider a 1-D example,

$$\min y = -(6x-2)^2 \sin(12x-4) \quad x \in [0,1] \quad (8.4)$$

The function plot and the Kriging prediction using seven points are shown in Fig. 8.1.

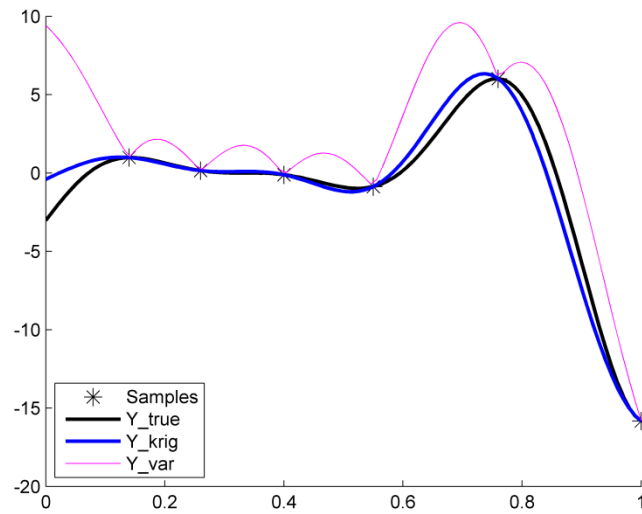


Figure 8.1 1-D Problem with 7 Samples

When the upper bound is used as the conservative surrogate model and used for optimization, the optimum x could be easily converged to the sample points, which are the local minima of y_{var} . Therefore, this upper bound of the Kriging prediction interval is not applicable for optimization. Another constant safety margin approach is also often used for the conservative surrogate model, where the safety margin is decided based on the empirical CDF of the cross-validation error. In this constant safety margin approach, the conservative surrogate model is obtained by shifting the Kriging prediction to certain amount and is often expressed as

$$y_{\text{xv}} = y_{\text{krig}} + e_{\text{xv},\%C} \quad (8.5)$$

where $e_{\text{xv},\%C}$ is the $C\%$ percentile of the cross-validation error. This conservative surrogate model has the same discrepancy from the Kriging prediction regardless of the sample position. This raises a problem that the conservative surrogate model cannot

count in the different uncertainty of the discrepancy due to different sample locations. Therefore, this constant safety margin approach may become over-conservative where the samples are dense and under-conservative where the samples are sparse. Consider the same example as shown in Fig. 8.2. If there is no sample at $x = 0.76$, the safety margin $e_{xv, \%C}$ in Eq. (8.5) is 1.454 and the conservative surrogate model is shown in Fig. 8.2. The conservative surrogate model is over-conservative in the region of $[0, 0.56]$ and under-conservative in the region of $[0.56, 1]$.

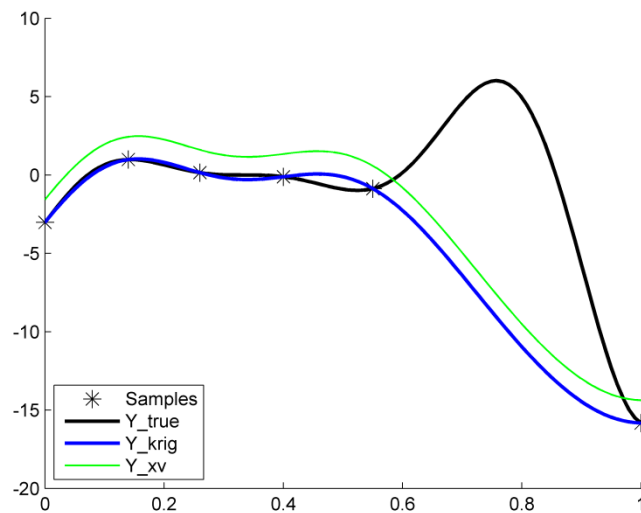


Figure 8. 2 1-D Problem with 6 Samples

According to the two examples discussed above, one can see that when a conservative surrogate model from Kriging is used for optimization problem, a variable safety margin approach that has zero margin at the sample points is not desirable because it generate unnecessary local optima regions; whereas a constant safety margin may not be desirable because it does not count in the affect from the sample position. What is

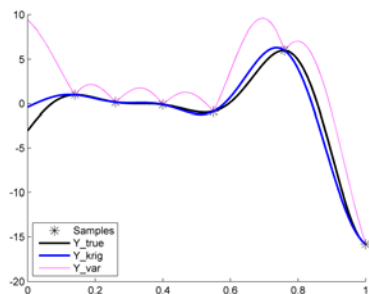
needed is a conservative surrogate model that uses a variable safety margin which does not generate local optima and counts in the effect from the sample position.

In this study, a new conservative surrogate model that combines the variable safety factor approach from Eq. (8.3) and the constant safety margin approach from Eq. (8.5) is proposed. First, to count in the effect from the sample position, an importance measure for each sample is needed. In this study, the AICc is chosen to quantify how important a sample is for the Kriging prediction. The importance function for sample x_i is defined as

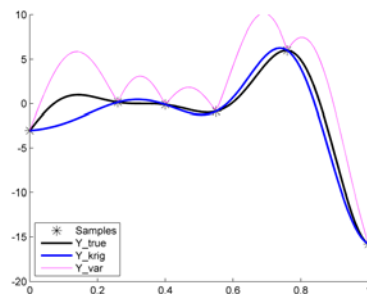
$$\text{Im}(x_i) = \left| \frac{\text{AICc} - \text{AICc}^{(-i)}}{\text{AICc}} \right|, \quad i = 1, \dots, n \quad (8.6)$$

where AICc is calculated using all n samples and $\text{AICc}^{(-i)}$ is calculated by omitting x_i . From Eq. (8.6), it can be seen that the larger $\text{Im}(x_i)$ is, the more important that x_i is. Figs. 8.3(a)-(g) show the Kriging prediction when each of the samples is omitted and the associated importance function values are shown in Fig. 8.3(h) as well. The sample point 6 at $x = 0.76$ is indeed around the highly nonlinear region, and its importance function value is the largest among all 7 samples. The second most important sample point 5 is at $x = 0.56$ which is a local minimum. Therefore, this importance function by the relative change in AICc values can characterize how important one sample is for the Kriging prediction accuracy.

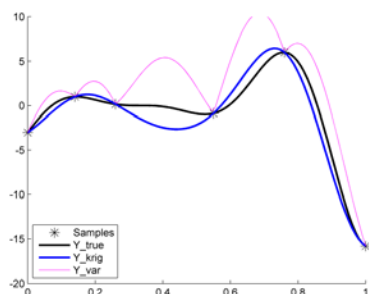
After the importance value for each sample is calculated, to construct a weighted Kriging variance, a leave-one-out Kriging variance $(\sigma_p^2)^{(-i)}$ is calculated using Eq. (3.22) where i^{th} sample x_i is omitted in generating the Kriging model, and then the weighted Kriging variance is expressed as



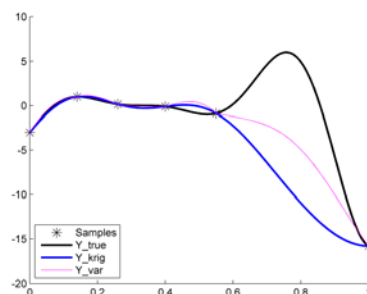
(a) Kriging Prediction w/o Sample #1



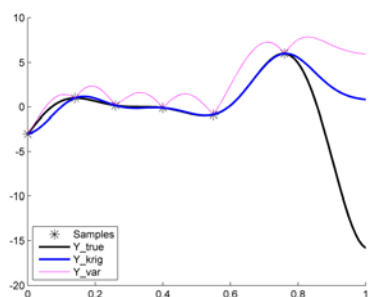
(b) Kriging Prediction w/o Sample #2



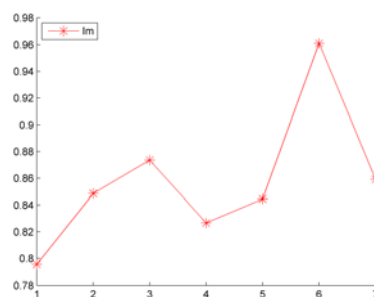
(c) Kriging Prediction w/o Sample #4



(f) Kriging Prediction w/o Sample #6



(g) Kriging Prediction w/o Sample #7



(h) $Im(x)$ Function Values at Each Sample

Figure 8. 3 Leave-one-out Kriging Prediction

$$\sigma_{p,weighted}^2 = \sum_{i=1}^n w(x_i) (\sigma_p^2)^{(-i)} \quad (8.7)$$

where the weight function is defined as

$$w(x_i) = \frac{1/\text{Im}(x_i)}{\sum_{i=1}^n 1/\text{Im}(x_i)} \quad (8.8)$$

The weight function decides how much the leave-one-out Kriging variance $(\sigma_p^2)^{(-i)}$ contributes to the total Kriging variance when x_i is missing according to the importance function value. Finally, the conservative surrogate model based on the weighted Kriging variance is

$$y_{con} = y_{krig} + \Phi^{-1}\left(\frac{C}{100}\right) \sigma_{p,weighted} \quad (8.9)$$

According to Eq. (8.9), one can see that when x_i is omitted, the leave-one-out Kriging variance $(\sigma_p^2)^{(-i)}$ would not become zero at x_i and therefore the total Kriging variance would not be zero at x_i as well, and indeed smoothes the conservative surrogate model eventually. On the other hand, since the $(\sigma_p^2)^{(-i)}$ will be larger at the place where no sample is nearby, it will make the total weighted Kriging variance become variable on the entire domain. Consider the same example shown in Figs. 8.4 and 8.5. In Fig. 8.4, the lines labeled as Y_true, Y_krig, Y_xv, Y_var and Y_con are the true response, the Kriging response, the conservative surrogate model using constant safety margin, the upper bound of Kriging prediction interval, and the conservative surrogate model using the weighted Kriging variance, respectively.

When all seven samples are used, the conservative surrogate model using weighted Kriging variance is smoother than the upper bound of Kriging prediction interval and closer to the true response than the conservative surrogate model using the safety margin. Moreover, if one of the important sample, (i.e., $x = 0.76$ in this case), is missing, the conservative surrogate model using the weighted Kriging variance would have a larger discrepancy than the conservative surrogate model using the constant safety margin approach in the region of $[0.56, 1]$ and a smaller discrepancy in the region of $[0, 0.56]$, as shown in Fig. 8.5. These two cases indeed show that the conservative surrogate model using the weighted Kriging variance can adaptively identify the conservativeness according to sample locations and provide a smooth surrogate model that does not change the optimum region of the original response function.

8.2.3 Sampling-Based RBDO Using the Conservative Surrogate Model

When a limited number of samples are used to generate the surrogate model, to assure the optimal design will satisfy the probabilistic constraints, the surrogate model needs to be replaced by the conservative surrogate model. Therefore, the conservative surrogate model using the weighted Kriging variance from Eq. (8.9) is used to represent the original performance function. The formulation of sampling-based RBDO becomes

$$\begin{aligned}
 & \text{minimize} && \text{Cost}(\mathbf{d}) \\
 & \text{subject to} && P[\hat{G}_j^c(\mathbf{X}) > 0] \leq P_{F_j}^{\text{Tar}}, \quad j = 1, \dots, nc \\
 & && \mathbf{d}^L \leq \mathbf{d} \leq \mathbf{d}^U, \quad \mathbf{d} \in \mathbf{R}^{nd} \text{ and } \mathbf{X} \in \mathbf{R}^{nr}
 \end{aligned} \tag{8.10}$$

where $\hat{G}_j^c(\mathbf{X})$ is the conservative surrogate model for performance function $G(\mathbf{X})$.

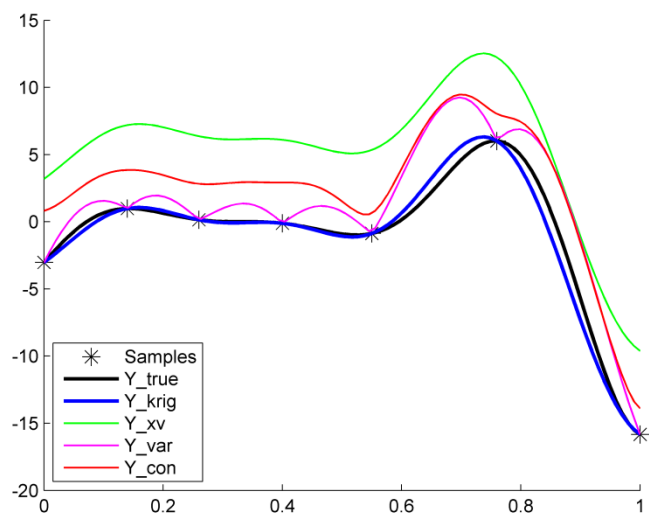


Figure 8. 4 Conservative Surrogate Model Using Weighted Kriging Variance (7 samples)

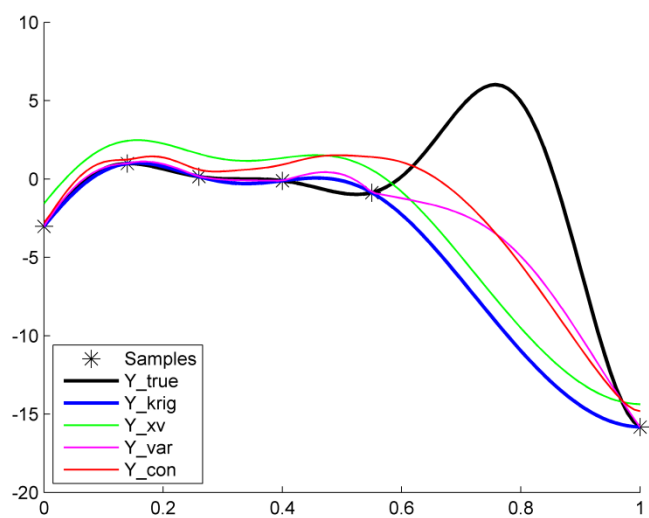


Figure 8. 5 Conservative Surrogate Model Using Weighted Kriging Variance (6 samples)

8.3 Numerical Example

Consider the 2-D RBDO problem with three probabilistic constraints used in Section 5.3.1. The initial design point is $\mathbf{d}_0 = [5 \ 5]$. First, it is worth mentioning that there is no need to construct the conservative surrogate model from the initial design point. In a typical RBDO process, the deterministic design optimization is carried out first. The RBDO process starts from the deterministic optimum thereafter. The conservative surrogate model will be constructed for the second stage for RBDO process only. In this example, the deterministic optimum is $\mathbf{x} = [5.19, 0.74]$ (which is the magenta cross in Fig. 8.6) and the number of sample in the local window is assumed to be fixed to 10 samples from Latin hypercube sampling method. Since 10 samples may not be enough to construct accurate surrogate model, the conservative surrogate models using the proposed weighted Kriging variance are generated for two active constraints G_1 and G_2 . For comparison study, the conservative surrogate models using the constant safety margin are also generated. The sampling-based RBDO is carried out using these two different conservative surrogate models and the optimum designs are compared in Table 8.2 and plotted in Fig. 8.6. The $C\%$ level is set to be 90% in this example.

In Fig. 8.6, it can be seen that the surrogate model from dynamic Kriging model itself, which is the blue line, underestimates the true response and results in unreliable for the obtained optimum design $\mathbf{d} = [4.7432, 1.5391]$ (blue cross in Fig. 8.6) and the probability of failure for G_2 is 2.4992% which is larger than the target probability of failure 2.275%, as shown in Table 8.1. Therefore the conservative surrogate model is indeed necessary for counting in the uncertainty from the surrogate model and assuring the optimum design can satisfy the probabilistic constraints. By using the weighted

Kriging variance for conservative surrogate model, the obtained RBDO optimum design is $\mathbf{d} = [4.7020, 1.5801]$, which is the red cross in Fig. 8.7, and the cost function value, the probability of failure for G_1 and G_2 are -1.8956 , 2.0654% and 1.6313% , respectively. As a comparison, the constant safety margin approach gives a more conservative optimum design $\mathbf{d} = [4.6510, 1.6205]$ (the green cross in Fig. 8.6) where the probability of failure for G_1 and G_2 are 1.8354% and 0.9455% , and the cost function value is -1.8783 . From the results in Table 8.1, one can see that the conservative surrogate model using the weighted Kriging variance can assure the optimum design satisfying the probabilistic constraints and has a better optimum design in terms of cost function compared to the conservative surrogate model using the constant safety margin approach.

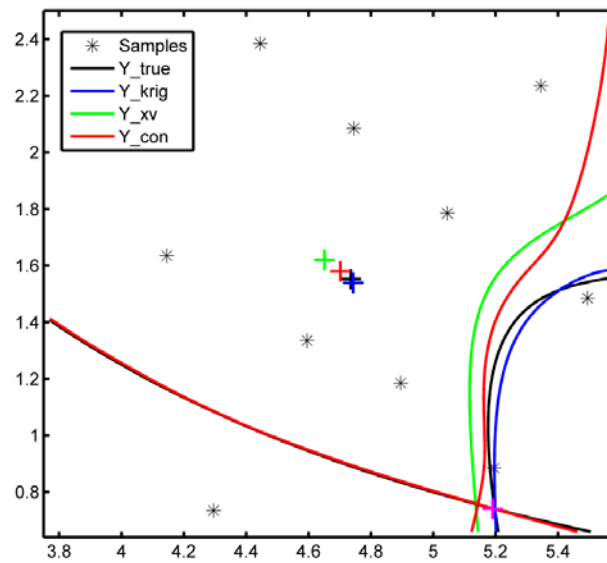


Figure 8. 6 Different Conservative Surrogate Models and Optimum Designs

Table 8. 1 Optimum Designs and the Probability of Failure at the Optimum

Methods	Cost	Optimum Design	MCS (5M)		
			P_{F_1} , %	P_{F_2} , %	
Surrogate Model	Dynamic Kriging	-1.9136	4.7432, 1.5391	2.3256	2.4992
	Constant Safety Margin	-1.8783	4.6510, 1.6205	1.8354	0.9455
	Weighted Kriging Variance	-1.8956	4.7020, 1.5801	2.0654	1.6313
Analytical	-1.9079	4.7351, 1.5518	2.2715	2.2633	

When using the surrogate model for sampling-based RBDO, the sample profile may affect the result as well. A good sample profile for dynamic Kriging may end up having a better surrogate model. Therefore, to investigate whether the proposed weighted Kriging variance for conservative surrogate model has a stable performance, a statistical study is carried out. In this statistical study, 50 sets of 10-LHS samples are generated. For each sample set, the sampling-based RBDO is carried out using both the weighted Kriging variance approach and the constant safety margin approach. The comparison for cost function value and the probability of failure at the optimum are shown in Table. 8.2.

Table 8. 2 Cost Function and Probability of Failure at Optimum Design (50 trials)

Methods	Cost (Median)	P_{F_1} , % (Median)	P_{F_2} , % (Median)	# of Violation G_1	# of Violation G_2
Dynamic Kriging	-1.9164	2.5982	2.3232	1	28
Constant Safety Margin	-1.8809	1.9002	0.9519	0	1
Weighted Kriging Variance	-1.8981	2.1352	1.6287	0	3

In Table 8.2, when using dynamic Kriging itself, the optimum designs violate the probabilistic constraint G_2 28 times out of 50 trials. It shows that the dynamic Kriging prediction either underestimates the true response or overestimates the true response at rough 50% chance for each way. By using the weighted Kriging variance for the conservative surrogate model, the obtained optimum designs violate the probabilistic constraint G_2 3 times. As a comparison, if the constant safety margin is used for conservative surrogate model, the number of violation for G_2 is 1 time out of 50 trials but at the same time the median cost function value at the optimum (which is -1.8809) is larger than the one by using the weighted Kriging variance (which is -1.8981). It is worth mentioning that since the conservativeness level in this example is set to be 90%, therefore both the constant safety margin approach and the weighted Kriging variance approach satisfy the conservativeness, and the weighted Kriging variance approach provides a better optimum in terms of the cost function value.

CHAPTER IX

CONCLUSION, RESEARCH PROGRESS AND FUTURE WORK

9.1 Conclusion

Reliability-based design optimization (RBDO) has been widely used recently for design of engineering applications. To achieve the optimal design with the target reliability of the performance function, the optimization algorithm and reliability analysis are implemented. However, in most large-scale complex engineering applications, the computer simulation of the performance function is compute-intensive and the sensitivity of the performance function is not available or extremely difficult to obtain. In such cases, the surrogate model is commonly used to represent the performance function as well as its sensitivity with respect to the design variables. However, traditional surrogate modeling methods are not accurate enough for reliability analysis use in RBDO; therefore the dynamic Kriging (DKG) is developed. In this DKG method, the traditional Kriging method is improved by using the pattern search for the correlation parameter estimation to obtain the accurate optimum; and selecting the best basis functions by a genetic algorithm based on a new accuracy criterion. A comprehensive study has been carried out to demonstrate that the DKG method can outperform other existing surrogate modeling methods in terms of generating an accurate surrogate model.

Based on the proposed DKG method, a sequential sampling method, that uses the bandwidth of the confidence interval of the surrogate model from the DKG, is applied and integrated with the DKG method to efficiently generate the accurate surrogate model. Numerical examples show that the sequential sampling method can lead to the accurate surrogate model faster than the Latin hypercube sampling does.

With the proposed sequential-sampling-based dynamic Kriging (SS-DKG) method, the MPP-based RBDO and the sampling-based RBDO using the surrogate model are carried out. In the MPP-based RBDO, both the performance function value and its sensitivity with respect to the design variables are approximated by the surrogate model from the SS-DKG. Three efficiency strategies are applied to efficiently and accurately generate the surrogate model in a local window at the current design for reliability analysis.

Numerical examples show that the MPP-based RBDO using the surrogate model can achieve the same RBDO optimum compared with the MPP-based RBDO using the analytical function and sensitivity. In the sampling-based RBDO, only the performance function value is approximated by the surrogate model from the SS-DKG. The sensitivity of the probabilistic constraint is obtained by using the scoring function and the MCS based on the surrogate model obtained by the SS-DKG. Three efficiency strategies are proposed to efficiently and accurately generate the surrogate model. Numerical examples show that the sampling-based RBDO using surrogate model can achieve a better RBDO optimum compared with the MPP-based RBDO using the analytical function and sensitivity since it does not require the approximation in probability of failure calculation, whereas FORM/SORM/DRM uses the approximation of the performance function to calculate the probability of failure in MPP-based RBDO.

To ensure the practical application of the sampling-based RBDO using surrogate model, a high performance computing (HPC) method by using the parallel computing is proposed. Three parallelization procedures in the sampling-based RBDO are applied: (1) the parallelization of the genetic algorithm in the DKG, (2) the parallelization of the

surrogate modeling for multiple constraints, (3) the parallelization of the MCS in reliability analysis. Numerical example shows that the computational time is significantly reduced after applying the HPC to the sampling-based RBDO using surrogate model.

When the number of samples used for surrogate modeling is small, the uncertainty from the surrogate model needs to be considered in RBDO using surrogate model to assure the obtained optimum satisfy the probabilistic constraints. A conservative surrogate model using the weighted Kriging variance is proposed. The weight assigned to each sample is calculated based on the proposed importance function according to the relative changes in AICc when current sample is omitted in surrogate modeling. By applying the proposed conservative surrogate model for sampling-based RBDO, the obtained design optimum can satisfy the probabilistic constraints and achieve a better optimum in terms of the cost function, compared to the optimum obtained by the constant conservative surrogate model.

9.2 Research Progress

The major of this study is to develop an accurate surrogate modeling method and apply it to the RBDO. Six tasks have been defined to meet the objective of the study:

- Task 1: Development of method for accurate surrogate modeling;
- Task 2: Development of method for sampling method which can be integrated into the surrogate modeling method;
- Task 3: Development of method for MPP-based RBDO using surrogate model;
- Task 4: Development of method for sampling-based RBDO using surrogate model;

- Task 5: Development of method using high performance computing for RBDO using surrogate model;
- Task 6: Development of method for generating conservative surrogate model for sampling-based RBDO.

9.3 Future Work

The SS-DKG method developed in this work can efficiently achieve an accurate surrogate model as demonstrated and has been successfully applied to sampling-based RBDO. However, in the sampling-based RBDO, the surrogate model is used to estimate the limit state of the performance functions while the SS-DKG method predicts the true function on the entire domain. Thus, when the number of dimension of the design problem increases, the computational effort increases exponentially and may become unaffordable. To efficiently characterize the limit state of the performance function, a classification method may perform more efficient than the SS-DKG method does. The support vector machine (SVM) is a popular method when dealing with classification problem. However, current SVM method is not accurate due to the non-used true function response information, whereas the performance of the prediction accuracy strongly depends on the sample profile. Therefore, a more accurate SVM method that can use the true function response and sequentially insert samples in the limit state region needs to be developed. In future work, more work should be focused on the development of the new SVM method to improve the efficiency of using the surrogate model for sampling-based RBDO.

REFERENCES

- Akaike, H., "A New Look at the Statistical Model Identification," *IEEE Transaction on Automatic Control*, Vol. 19, No. 6, pp. 716-723, 1974.
- Ang, A. H-S., and Tang, W., *Probability Concepts in Engineering Planning and Design. II: Decision, Risk, and Reliability*, Wiley, New York, NY, 1984.
- Atkinson K.E., *An Introduction to Numerical Analysis*, John Wiley & Sons, Chapter 5, New York, NY, 1989.
- Ba-abbad, M.A., Nikolaidis, E., and Kapania, R.K., "New Approach For System Reliability-Based Design Optimization," *AIAA Journal*, Vol. 44, No. 5, pp. 1087-1096, 2006.
- Bäck, T., *Evolutionary Algorithms in Theory and Practice*, Oxford Univ. Press, New York, 1996.
- Beachkofski, B., and Grandhi, R., "Improved Distributed Hypercube Sampling," *43rd Structures Dynamics and Materials Conference*, Denver, USA, 2002.
- Box, E.P., and Draper, N.R., *Empirical Model Building and Response Surfaces*, New York: Wiley, 1987.
- Breitung, K., "Asymptotic Approximations for Multinormal Integrals," *ASCE Journal of Engineering Mechanics*, Vol. 110, No. 3, pp. 357-366, 1984.
- Broomhead, D.S., Loewe, D., "Multivariate Functional Interpolation and Adaptive Networks," *Complex Systems*, Vol. 2, pp. 321-55, 1988.
- Buranathiti, T., Cao, J., and Chen, W., "A Weighted Three-Point-Based Strategy for Variance Estimation," *Proceedings of ASME Design Engineering Technical Conferences*, DETC2004-57363, 2004.
- Burkardt, J., Gunzburger, M., Peterson, J., and Brannon, R., "User Manual and Supporting Information for Library of Codes for Centroidal Voronoi Placement and Associated Zeroth, First, and Second Moment Determination," *Sandia National Laboratories Technical Report*, vSAND2002-0099, Feb. 2002.
- Burnham, K.P., and Anderson, D.R., *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*, Springer, 2002.
- Center for Computer-Aided Design, College of Engineering, *DRAW Concept Manual*, The University of Iowa, Iowa City, IA, 1999a.

Center for Computer-Aided Design, College of Engineering, *DRAW User Reference*, The University of Iowa, Iowa City, IA, 1999b.

Chandra, M.J., *Statistical Quality Control*, CRC Press, Chapter 3, Boca Raton, FL, 2001.

Choi, K.K., Tu, J., and Park, Y.H., "Extensions of Design Potential Concept for Reliability-Based Design Optimization," *Journal of Structural and Multidisciplinary Optimization*, Vol. 22, No. 3, pp. 335-350, 2001.

Chiles, J.P., and Delfiner, P., *Geostatistics:-Modeling Spatial Uncertainty*, Wiley, New York, 1999.

Clarke, S.M., Griebisch, J.H., and Simpson, T.W., "Analysis of Support Vector Regression for Approximation of Complex Engineering Analyses," *Journal of Mechanical Design*, Vol. 127, No. 6, pp. 1077-1087, 2005.

Cortes, C., and Vapnik, V., "Support-Vector Networks," *Machine Learning*, Vol. 20, No. 3, pp. 273-297, 1995.

Currin, C., Mitchell, T., Morris, M., and Ylvisaker, D., "Bayesian Prediction of Deterministic Functions, with Applications to the Design and Analysis of Computer Experiments," *Journal of the American Statistical Association*, Vol. 86, No. 416, pp. 953-963, 1991.

Denny, M., "Introduction to Importance Sampling in Rare-Event Simulations," *European Journal of Physics*, Vol. 22, No.4, pp. 403-411, 2001.

Dey, A., and Mahadevan, S., "Ductile Structural System Reliability Analysis Using Importance Sampling," *Structure & Safety*, Vol. 20, No. 2, pp. 137-154, 1998.

Ditlevsen, O., "Narrow Reliability Bounds for Structural Systems," *Journal of Structural Mechanics*, Vol. 7, No. 4, pp. 453-472, 1979.

Ditlevsen, O., and Madsen, H.O., *Structural Reliability Methods*, John Wiley & Sons Ltd., Chichester, 1996.

Du, X., and Chen, W., "A Most Probable Point-Based Method for Efficient Uncertainty Analysis," *Journal of Design and Manufacturing Automation*, Vol. 4, No. 1, pp. 47-66, 2001.

Du, X., Sudjianto, A., and Chen, W., "An Integrated Framework for Optimization Under Uncertainty Using Inverse Reliability Strategy," *ASME Journal of Mechanical Design*, Vol. 126, No. 4, pp. 562-570, 2004.

Du, X., and Huang, B., "Uncertainty Analysis by Dimension Reduction Integration and Saddlepoint Approximations," *Proceedings of ASME Design Engineering Technical Conferences*, DETC2005-84523, 2005.

Engelund, S. and Rackwitz, R., "A Benchmark Study on Importance Sampling Techniques in Structural Reliability," *Structural Safety*, Vol. 12, No. 4, pp.255-276, 1993.

Forrester, A., and Keane, A.J., "Recent Advances in Surrogate-Based Optimization," *Aerospace Sciences*, Vol. 45, No. 1-3, pp. 50-79, 2009.

Forrester, A., Sobester, A., and Keane, A.J., *Engineering Design via Surrogate Modelling: A Practical Guide*, Willy, New Jersey, 2008.

Goel, T., Haftka, R.T., and Shyy, W., "Error Measures for Noise-Free Surrogate Approximations," *AIAA Paper*, 2008-901, 2008.

Goel, T., Haftka, R.T., Shyy, W., and Watson, L.T., "Pitfalls of Using A Single Criterion for Selecting Experimental Designs," *International Journal for Numerical Methods in Engineering*, Vol. 75, No. 2, pp. 127-155, 2008.

Gorissen, D., Tommasi L.D., Crombecq, K., and DhaeneSpringer, T., "Sequential Modeling of a Low Noise Amplifier with Neural Networks and Active Learning," *Neural Computing & Applications*, Vol. 18, No. 5, pp. 485-494, 2009.

Gumbert, C. R., Hou, G.J.-W., and Newman P. A., "Reliability Assessment of a Robust Design under Uncertainty for a 3-D Flexible Wing," *AIAA Journal*, 2003-4094, 2003.

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

Hasofer, A. M., and Lind, N. C., "An Exact and Invariant First Order Reliability Format," *ASCE Journal of the Engineering Mechanics Division*, Vol. 100, No. 1, pp. 111-121, 1974.

Hedayat, A.S., Sloane, N.J.A., and Stufken, J., *Orthogonal Arrays*, Springer-Verlag, New York, 1999.

Hertog, D., Kleijnen, J., and Siem, A., "The Correct Kriging Variance Estimated by Bootstrapping," *Journal of the Operational Research Society*, Vol. 57, pp. 400-409, 2006.

Hohenbichler, M., and Rackwitz, R., "Sensitivity and Important Measures in Structural Reliability," *Civil Engineering Systems*, Vol. 3, No.4, pp. 203-209, 1986.

Hohenbichler, M. and Rackwitz, R., "Improvement of Second-Order Reliability Estimates by Importance Sampling," *ASCE Journal of Engineering Mechanics*, Vol. 114, No. 12, pp. 2195-2199, 1988.

Hou, G.J.-W., "A Most Probable Point-Based Method for Reliability Analysis, Sensitivity Analysis, and Design Optimization," *NASA CR*, 2004-213002.

Huang, B., and Du, X., "A Robust Design Method Using Variable Transformation and Gauss-Hermite Integration," *International Journal for Numerical Methods in Engineering*, Vol. 66, No. 12, pp. 1841-1858, 2006.

Hurvich, C.M., and Tsai, C.L., "Regression and Time Series Model Selection in Small Samples," *Biometrika*, Vol. 76, pp. 297-307, 1989.

Jin, R., Du, X., and Chen, W., "The Use of Metamodeling Techniques for Design under Uncertainty," *Structural and Multidisciplinary Optimization*, Vol. 25, No. 2, pp. 99-106, 2003.

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

Joseph, V.R., Hung, Y., and Sudjianto, A., "Blind Kriging: A New Method for Developing Metamodels," *Journal of Mechanical Design*, Vol. 130, No. 3, 2008.

Kalsi, M., Hacker, K., and Lewis, K., "A Comprehensive Robust Design Approach for Decision Trade-Offs in Complex Systems Design," *ASME Journal of Mechanical Design*, Vol. 123, No. 1, pp. 1-10, 2001.

Keane, A.J., "Design Search and Optimization Using Radial Basis Functions with Regression Capabilities," *Proceedings of the Conference on Adaptive Computing in Design and Manufacture*, Berlin, 2004.

Kim, C., Wang, S., and Choi, K. K., "Efficient Response Surface Modeling by Using Moving Least-Squares Method and Sensitivity," *AIAA Journal*, Vol. 43, No. 11, pp. 2404-2411, 2005.

Krige, D.G. "A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand," *Journal of the Chemical, Metallurgical and Mining Engineering Society of South Africa*, Vol. 52, No. 6, pp. 119-139, 1951.

Kyriakidis, P.C., and Goodchild, M.F., "On the Prediction Error Variance of Three Common Spatial Interpolation Schemes," *International Journal of Geographical Information Science*, Vol. 20, No. 8, pp. 823-855, 2006.

Lancaster, P., and Salkauskas, K., "Surfaces Generated by Moving Least Squares Methods," *Mathematics of Computation*, Vol. 37, No. 155, pp. 141-58, 1981.

Lee, I., Choi, K.K., and Du, L., "Alternative Methods for Reliability-Based Robust Design Optimization Including Dimension Reduction Method," *32nd ASME Design Automation Conference*, September 10-13, Philadelphia, PA, 2006.

Lee, I., Choi, K.K., Du, L., and Gorsich, D., "Dimension Reduction Method for Reliability-Based Robust Design Optimization," *Computers & Structures*, Vol. 86, No.13-14, pp. 1550-1562, 2008a.

Lee, I., Choi, K.K., Du, L., and Gorsich, D., "Inverse Analysis method using MPP-based Dimension Reductino for Reliability-based Design Optimization of Nonlinear and Multi-Dimensional Systems," *Computer Methods in Applied Mechanics and Engineering*, No. 198, pp. 14-27, 2008b.

Lee, I., Choi, K.K., Noh, Y., Zhao, L., and Gorsich, D., "Sampling-Based Stochastic Sensitivity Analysis Using Score Functions for RBDO Problems With Correlated Random Variables," *Journal of Mechanical Design*, 021003-3, Vol. 133, 2011.

Lee, I., Choi, K.K., and Zhao, L., "Sampling-Based RBDO Using the Dynamic Kriging (D-Kriging) Method and Stochastic Sensitivity Analysis," *Structural and Multidisciplinary Optimization*, 2011.

Lewis, R.M., and Torczon, V., "Pattern Search Algorithms for Bound Constrained Minimization," *SIAM Journal on Optimization*, Vol. 9, No. 4, pp. 1082-1099, 1999.

Liang, J., Mourelatos, Z. P., and Nikolaidis, E., "System Reliability-Based Design Using a Single-Loop Method," *SAE 2007 World Congress*, April 16-19, Detroit, MI, 2007.

Lin, C. Y., Huang, W. H., Jeng, M. C., and Doong, J. L., "Study of an Assembly Tolerance Allocation Model Based on Monte Carlo Simulation," *Journal of Materials Processing Technology*, Vol. 70, No. 1-3, pp. 9-16, 1997.

Lophaven, S.N., Nielsen, H.B., and Sondergaard, J., DACE: A Matlab Kriging Toolbox, *Technical Report IMM-TR-2002-12*, 2002.

Luna, S.D., and Young, A., "The Bootstrap and Kriging Prediction Intervals," *Scandinavian J Stat*, Vol. 30, pp. 175-192, 2003.

Madsen, H.O., Krenk, S., and Lind, N. C., *Methods of Structural Safety*, Prentice-Hall Inc., Englewood Cliffs, NJ, 1986.

Madsen, H.O. and Hansen, P.F., "A Comparison of Some Algorithms for Reliability Based Structural Optimization and Sensitivity Analysis," *Proceedings of 4th IFIP WG 7.5 Conference*, Munich, Germany, September 11-13, 1991.

- Marler, R.T., and Arora, J.S., "Survey of Multi-objective Optimization Methods for Engineering", *Structural and Multidisciplinary Optimization*, Vol. 26, No. 6, pp. 369-395, 2004.
- Martin, J.D., "Computational Improvements to Estimating Kriging Metamodel Parameters," *Journal of Mechanical Design*, Vol. 131, No. 8, 2009.
- Martin, J.D. and Simpson, T.W., "Use of Kriging Models to Approximate Deterministic Computer Models," *AIAA Journal*, Vol. 43, No. 4, pp.853-863, 2005.
- Mckay, M.D., Conover, W.J. and Beckman, R. J., "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, Vol. 21, No. 2, pp. 239-245, 1979.
- Mourelatos, Z. P. and Liang, J., "A Methodology for Trading-off Performance and Robustness under Uncertainty," *Proceedings of ASME Design Engineering Technical Conferences*, DETC2005-85019, 2005.
- Noh, Y., Choi, K.K., and Du, L., "Reliability-based design optimization of problems with correlated input variables using a Gaussian Copula," *Structural and Multidisciplinary Optimization*, Vol. 38, No. 1, pp. 1-16, 2007.
- Olsson, A. and Sandberg, G. E., "Latin Hypercube Sampling for Stochastic Finite Element Analysis," *ASCE Journal of Engineering Mechanics*, Vol. 128, No. 1, pp. 121-125, 2002.
- Owen, A., "Orthogonal Arrays for Computer Experiments, Integration, and Visualization," *Statistica Sinica*, Vol. 2, pp. 439-452, 1992.
- Palle, T. C., and Michael, J. B., *Structural Reliability Theory and Its Applications*, Springer-Verlag, Berlin, Heidelberg, 1982.
- Rackwitz, R. and Fiessler, B., "Structural Reliability under Combined Load Sequences," *Computers and Structures*, Vol. 9, No. 5, pp. 489-494, 1978.
- Picheny, V., "Improving Accuracy and Compensating for Uncertainty in Surrogate Modeling," *Ph.D Dissertation*, Univ. of Florida, Gainesville, FL, 2009.
- Rahman, S. and Wei, D., "A Univariate Approximation at Most Probable Point for Higher-Order Reliability Analysis," *International Journal of Solids and Structures*, Vol. 43, No. 9, pp. 2820-2839, 2006.
- Rahman, S., and Wei, D., "Design Sensitivity and Reliability-Based Structural Optimization by Univariate Decomposition," *Structural and Multidisciplinary Optimization*, Vol. 35, No. 3, pp. 245-261, 2007.

Rosenblatt, M., "Remarks on A Multivariate Transformation," *Annals of Mathematical Statistics*, Vol. 23, No. 3, pp. 470-472, 1952.

Rosenblueth, E., "Point Estimates for Probability Moments," *Proceeding of the National Academy of Sciences of the United States of America*, Vol. 72, No. 10, pp. 3812-3814, 1975.

Rubinstein, R.Y., *Simulation and Monte Carlo Method*, John Wiley & Sons, New York, 1981.

Sacks, J., Welch, W.J., Mitchell, T.J., and Wynn, H.P., "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. 4, No. 4, pp. 409-423, 1989.

Schueller, G.I. and Stix, R., "A Critical Appraisal of Methods to Determine Failure Probabilities," *Structural Safety*, Vol. 4, No. 4, pp. 293-309, 1987.

Shewry, M., and Wynn, H., "Maximum Entropy Sampling," *Journal of Applied Statistics*, Vol. 14, No. 2, pp. 165-170, 1987.

Simpson, T., Lin, D., and Chen, W., "Sampling Strategies for Computer Experiments: Design and Analysis," *International Journal of Reliability and Applications*, Vol. 2, No. 3, pp. 209-240, 2002.

Simpson, T.W., Toropov, V., Balabanov, V., and Viana, F.A., "Design and Analysis of Computer Experiments in Multidisciplinary Design Optimization: A Review of How We Have Come—or Not," *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Victoria, British Columbia, 2008.

Stein, M., "Large Sample Properties of Simulations using Latin Hypercube Sampling," *Technometrics*, Vol. 29, No. 2, pp. 143-151, 1987.

Su, J., and Renaud, J. E., "Automatic Differentiation in Robust Optimization," *AIAA Journal*, Vol. 35, No. 6, pp. 1072-1079, 1997.

Swanson Analysis System Inc., *ANSYS Engineering Analysis System User's Manual*, Vol. I, II, Houston, PA, 1989.

Queipo, N.V., Haftka, R.T., Shyy, W., Goel, T., Vaidyanathan, R., and Tucker, P.K., "Surrogate-Based Analysis and Optimization," *Progress in Aerospace Sciences*, Vol. 41, No. 1, pp. 1-28, 2005.

Taguchi, G., Elsayed, E., and Hsiang, T., *Quality Engineering in Production Systems*, McGraw-Hill, Chapter 2 and 3, New York, NY, 1989.

Tu, J., and Choi, K.K., "A New Study on Reliability-Based Design Optimization," *Journal of Mechanical Design*, Vol. 121, No. 4, pp. 557-564, 1999.

Tu, J., Choi, K.K., and Park, Y.H., “Design Potential Method for Reliability-Based System Parameter Design Using Adaptive Probabilistic Constraint Evaluation,” *AIAA Journal*, Vol. 39, No. 4, pp. 667-677, 2001.

Viana, F.A.C., “SURROGATES Toolbox User’s Guide,” <http://fcchegury.googlepages.com>, 2009.

Viana, F.A.C., Haftka, R.T., and Steffen Jr, V., “Multiple Surrogates: How Cross-Validation Errors Can Help Us to Obtain the Best Predictor,” *Structural and Multidisciplinary Optimization*, Vol. 39, No. 4, pp. 439-457, 2009.

Viana, F.A.C., Picheny, V., and Haftka, R.T., "Using Cross Validation to Design Conservative Surrogates," *AIAA Journal*, Vol. 48, No.10, pp. 2286-2298, 2010.

Walker, J. R., “Practical Application of Variance Reduction Techniques in Probabilistic Assessments,” *the Second International Conference on Radioactive Waste Management*, Winnipeg, Manito, Canada, pp. 517-521, 1986.

Wang, G.G., “Adaptive Response Surface Method Using Inherited Latin Hypercube Design Points,” *Journal of Mechanical Design*, Vol. 125, No. 2, pp. 210-221, 2003.

Wang, G.G., and Shan, S., “Review of Metamodeling Techniques in Support of Engineering Design Optimization,” *Journal of Mechanical Design*, Vol. 129, No. 4, pp. 370–380, 2007.

Wei, D., “A Univariate Decomposition Method for Higher-Order Reliability Analysis and Design Optimization,” *Ph. D. Thesis*, University of Iowa, 2006.

Wu, Y.T., Millwater, H.R., and Cruse, T.A., “Advanced Probabilistic Structural Analysis Method for Implicit Performance Functions,” *AIAA Journal*, Vol. 28, No. 9, pp. 1663–1669, 1990.

Wu, Y.T., “Computational Methods for Efficient Structural Reliability and Reliability Sensitivity Analysis,” *AIAA Journal*, Vol. 32 No. 8, pp. 1717–1723, 1994.

Wu, Y.T., Shin, Y., Sues, R., and Cesare, M., “Safety-Factor Based Approach for Probability-Based Design Optimization,” 42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference and Exhibit, *AIAA-2001-1239*, Seattle, WA, April 16-19, 2001.

Youn, B.D., and Choi, K.K., “Adaptive Probability Analysis Using Performance Measure Approach,” 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, *AIAA2002-5532*, Atlanta, Georgia, Sep. 4-6, 2002.

Youn, B.D. and Choi, K.K., "A New Response Surface Methodology for Reliability-Based Design Optimization," *Computers and Structures*, Vol. 82, No. 2-3, pp. 241-256, 2004.

Youn, B.D., Choi, K.K., and Du, L., "Adaptive Probability Analysis Using An Enhanced Hybrid Mean Value (HMV+) Method," *Journal of Structural and Multidisciplinary Optimization*, Vol. 29, No. 2, pp. 134-148, 2005a.

Youn, B.D., Choi, K.K., and Du, L., "Enriched Performance Measure Approach (PMA+) for Reliability-Based Design Optimization," *AIAA Journal*, Vol. 43, No. 4, pp. 874-884, 2005b.

Youn, B.D., Choi, K.K., and Park, Y.H., "Hybrid Analysis Method for Reliability-Based Design Optimization," *Journal of Mechanical Design*, Vol. 125, No. 2, pp. 221-232, 2003.

Youn, B.D., Choi, K.K., Yang, R.-J., and Gu, L., "Reliability-Based Design Optimization for Crash-worthiness of Vehicle Side Impact," *Structural Multidisciplinary Optimization*, Vol. 26, No. 3-4, pp. 272-283, 2004.

Youn, B.D., Choi, K.K., and Yi, K., "Performance Moment Integration (PMI) Method for Quality Assessment in Reliability-Based Robust Optimization," *Mechanics Based Design of Structures and Machines*, Vol. 33, No. 2, pp. 185-213, 2005c.

Youn, B.D., Xi, Z., Wells, L.J., and Lamb, D.A., "Stochastic Response Surface Using The Enhanced Dimension Reduction (eDR) Method For Reliability-Based Robust Design Optimization," *III European conference on Computational Mechanics*, Lisbon, Portugal, 2006.

Xiong, Y., Chen, W., and Tsui, K., "A New Variable Fidelity Optimization Framework Based on Model Fusion and Objective-Oriented Sequential Sampling," *Journal of Mechanical Design*, Vol. 130, No. 11, 2008.

Xu, H., and Rahman, S., "A Moment-Based Stochastic Method for Response Moment and Reliability Analysis," *Proceedings of 2nd MIT Conference on Computational Fluid and Solid Mechanics*, Cambridge, MA, July 17-20, 2003.

Xu, H., and Rahman, S., "A Generalized Dimension-Reduction Method for Multi-dimensional Integration in Stochastic Mechanics," *International Journal for Numerical Methods in Engineering*, Vol. 61, No. 12, pp. 1992-2019, 2004a.

Xu, H., and Rahman, S., "A Univariate Dimension-Reduction Method for Multi-dimensional Integration in Stochastic Mechanics," *Probabilistic Engineering Mechanics*, Vol. 19, No. 4, pp. 393-408, 2004b.

Zhao, L., Choi, K.K., and Lee, I., "Metamodeling Method Using Dynamic Kriging for Design Optimization," *accepted by AIAA Journal*, 2011.

Zhao, Y-G., Zhong W-Q., and Ang A.H-S., "Estimating Joint Failure Probability of Series Structural Systems," *Journal of Engineering Mechanics*, Vol. 133, No. 5, pp. 588-596, 2007.