2013

Efficient sampling-based Rbdo by using virtual support vector machine and improving the accuracy of the Kriging method

Hyeongjin Song
University of Iowa

EFFICIENT SAMPLING-BASED RBDO BY USING VIRTUAL SUPPORT VECTOR MACHINE AND IMPROVING THE ACCURACY OF THE KRIGING METHOD

by

Hyeongjin Song

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

December 2013

Thesis Supervisor: Professor K.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

Hyeongjin Song

has been approved by the Examining Committee
for the thesis requirement for the Doctor of Philosophy
degree in Mechanical Engineering at the December 2013 graduation.

Thesis Committee:

K.K. Choi, Thesis Supervisor

David Lamb

Yong Chen

Jia Lu

Hiroyuki Sugiyama

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND SYMBOLS

AMV	Advanced Mean Value
$\boldsymbol{\beta}$	Regression Coefficients
β_t	Target Reliability Index
CBS	Constraint Boundary Sampling
CCE	Convergence Classification Error
CDF	Cumulative Density Function
CFD	Computational Fluid Dynamics
CMSE	Convergence Mean Squared Error
CMV	Conjugate Mean Value
CPF	Convergence Probability of Failure
CV	Cross-Validation
DDO	Deterministic Design Optimum
DKG	Dynamic Kriging
DoE	Design of Experiment
DRM	Dimension Reduction Method
$D(\mathbf{x})$	Nearest Distance from Existing Samples
EDSD	Explicit Design Space Decomposition
EFF	Expected Feasibility Function
EGRA	Efficient Global Reliability Analysis
EIS	Equal Interval Search
$\mathbf{f}$	Basis Functions
$\boldsymbol{\Phi}$	Standard Normal CDF
FORM	First Order Reliability Method
GA	Genetic Algorithm

γ	SVM Parameter
GPS	Generalized Pattern Search
GSS	Golden Section Search
$g(\mathbf{u})$	Performance Function in U-space
$G(\mathbf{X})$	Performance Function in X-space
HMV	Hybrid Mean Value
HSS	Hammersley Sequence Sampling
K	SVM Kernel Function
λ	Penalized MLE Parameter
LCVT	Latinized Centroidal Voronoi Tessellation
LHS	Latin Hypercube Sampling
LOOCV	Leave-One-Out Cross-Validation
MCS	Monte Carlo Simulation
MLE	Maximum Likelihood Estimation
MPP	Most Probable Point
MSE	Mean Squared Error
MV	Mean Value
N	Number of DoE Samples
nc	Number of Constraints
nd	Number of Design Variables
nr	Number of Random Variables
NRMSE	Normalized Root Mean Square Error
N_{stop}	Number of Test Points for Stopping Criterion
N_{test}	Number of Test Points for Error Measure
PDF	Probability Density Function

PF	Probability of Failure
PMA	Performance Measure Approach
PMLE	Penalized Maximum Likelihood Estimation
$\mathbf{R}$	Kriging Correlation Function
RBDO	Reliability-Based Design Optimization
RIA	Reliability Index Approach
SORM	Second Order Reliability Method
SVM	Support Vector Machine
$s(\mathbf{x})$	SVM Decision Function
TGS	Transformations Gibbs Sampling
$\boldsymbol{\theta}$	Kriging Correlation Function Parameter
UKG	Universal Kriging
VSVM	Virtual Support Vector Machine

CHAPTER 1

INTRODUCTION

This study presents development of efficient and accurate classification methods for reliability-based design optimization (RBDO) and several accuracy improvement strategies for the Kriging method. Previously the dynamic Kriging method (DKG) is developed and applied for sampling-based RBDO. However, reliability analyses using the Kriging model are expensive due to complicated matrix calculations in the Kriging method, and thus the efficiency needs to be improved. A new virtual support vector machine (VSVM) is proposed using an efficient classification methodology. Computer simulations are expensive in many engineering problems, thus the number of design of experiment (DoE) samples needs to be minimized in sampling-based RBDO to reduce the total computational cost. By applying four accuracy improvement strategies such as accurate parameter estimation in maximum likelihood estimation (MLE), penalized MLE (PMLE) for small DoE sample size, optimum correlation function selection and mean structure selection in the Kriging method, the accuracy of surrogates can be improved and consequently the number of DoE samples can be reduced.

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 Sampling-based RBDO and Surrogate Modeling Methods

Accurate reliability analysis is of great importance for solving engineering design problems. Inaccurate reliability analysis result can lead to an unreliable or overly conservative design. Numerous methods that are based on the most probable point (MPP)

are available in literature for carrying out reliability analyses of many engineering problems for which the sensitivity information can be obtained [Hasofer and Lind 1974; Hohenbichler and Rackwitz 1986; Breitug 1984; Ditlevsen and Madsen 1996; Haldar and Mahadevan 2000; Tu et al. 1999; Hou et al. 2004; Youn et al. 2005; Madsen et al. 2006; Rahman and Wei 2006; Lee et al. 2008; Valdebenito and Schueller 2010]. On the other hand, the sensitivity is often not available or difficult to obtain accurately in complex multi-physics or multidisciplinary design problems. Without sensitivity, an alternative to the MPP-based reliability analysis method is to directly perform the probability integration numerically by carrying out computer simulations at the Monte Carlo simulation (MCS) sampling points [Rubinstein 1981; Zeeb and Burns 1997; Haldar and Mahadevan 2000; Ching 2011]. However, this method requires a large number of response function evaluations and can be impractical in terms of computational cost.

Therefore, surrogate-based methods are used to reduce the computational cost [Booker et al. 1997, 1998] without requiring sensitivity analysis. The main advantage of the surrogate-based method is that a limited number of function evaluations can be used to construct surrogate models. Many different surrogates, such as polynomial response surface (PRS) [Barton 1994; Jin et al. 2001; Simpson et al. 2001b; Queipo et al. 2005; Wang and Shan 2007; Forrester et al. 2008; Forrester and Keane 2009; Fang et al. 2010; Zhao et al. 2011], radial basis function (RBF) [Barton 1994; Jin et al. 2001; Queipo et al. 2005; Wang and Shan 2007; Forrester et al. 2008; Forrester and Keane 2009; Zhao et al. 2011], multivariate adaptive regression spline (MARS) [Barton 1994; Jin et al. 2001; Wang and Shan 2007; Fang et al. 2010], moving least squares (MLS) [Kim et al. 2005, 2009; Forrester and Keane 2009; Kang et al. 2010], support vector regression (SVR) [Wang and Shan 2007; Forrester et al. 2008; Forrester and Keane 2009] and Kriging [Sacks et al. 1989a, b; Cressie 1991; Barton 1994; Jin et al. 2001; Simpson et al. 2001b; Queipo et al. 2005; Wang and Shan 2007; Forrester et al. 2008; Forrester and Keane 2009; Kleijnen 2009; Fang et al. 2010; Zhao et al. 2011], have been developed and

applied to engineering problems. These surrogates provide approximations of otherwise expensive computer simulations. Once an accurate surrogate model is generated, MCS can be carried out using the surrogate model to estimate the reliability. This method is called the sampling-based reliability analysis or sampling-based RBDO. In the sampling-based RBDO, probabilistic constraints are used. Thus, not only the probability of failure but also its sensitivity needs to be accurately estimated.

DKG was developed and applied successfully for sampling-based methods [Zhao et al. 2011]. However, when the response values are evaluated, all samples within the design space are used to calculate the trend and random component of the Kriging method. Furthermore, complex matrix calculations are also required. Therefore, response evaluations using the Kriging method could be computationally expensive since the sampling-based reliability analysis and RBDO need response evaluations at a very large number of MCS points to accurately estimate the probability of failure and its sensitivity. Furthermore, surrogate-based approaches usually obtain response function values over the entire domain. Therefore, the surrogate-based method requires a large number of samples even at the unnecessary regions to reach the target accuracy (i.e., mean squared error or R2), and thus they become inefficient [Hurtado and Alvarez 2003]. The computational burden becomes heavier in high-dimensional space due to the curse of dimensionality [Vapnik 1998; Cherkassky and Mulier 1998; Burges 1998]. Therefore, a classification method with simpler formulation needs to be investigated while achieving similar accuracy.

1.1.2 Virtual Support Vector Machine

The support vector machine (SVM) is a classification method, and thus it constructs only the decision (i.e., limit state) function, which maximizes the distance to the existing samples [Vapnik 1998, 2000; Cherkassky and Mulier 1998; Scholkopf 1999; Kecman 2001, 2005; Scholkopf and Smola 2002]. In SVM, only support vectors are used

instead of all samples for evaluations of responses and most calculations are performed through the kernel function. Thus, the response evaluation process is very efficient for MCS, compared to the Kriging method. Another advantage of the classification method is that they can deal with multiple constraints at once [Basudhar et al. 2012]. The SVM with a sequential sampling strategy, which is called the explicit design space decomposition (EDSD), is developed and applied to discontinuous and disjoint problems successfully [Basudhar et al. 2008, 2012; Basudhar and Missoum 2008, 2010]. Even though EDSD can be also used for continuous and differentiable problems, it often converges very slowly, and thus requires a large number of DoE samples. One of the main reasons for the inefficiency of EDSD for continuous problems is that it only uses the classification response function values rather than the actual function values to construct the decision function. Therefore, the accuracy needs to be improved for continuous problems.

Accurate probability of failure can be predicted using accurate limit state function, and accurate sensitivity can be calculated by using the score function [Lee et al. 2011]. It is interesting to note that the score function method depends on the derivatives of the input joint and marginal distributions. Therefore, the sampling-based method requires only an accurate decision function to evaluate the probability of failure and its sensitivity. That is, only the decision between a success and a failure is used instead of the function value. Thus, even though SVM, being a classification method, cannot be directly used for deterministic design optimization due to lack of surrogate model and thus design sensitivity, it is applicable for sampling-based RBDO.

In this research, a virtual SVM (VSVM) is proposed to improve the accuracy of SVM, while maintaining the desirable features of SVM, by using the available response function values. Unlike EDSD, VSVM is developed primarily for continuous design problems. The VSVM constructs the decision function rather than the surrogate model over the given domain. The proposed adaptive sampling method provides new samples near the limit state, which makes the method efficient. The proposed method provides an

explicit form of the limit state function, so it is efficient in obtaining response values at MCS points. Then, VSVM is applied to RBDO processes in this research. An efficient uniform sampling method is also proposed for hyper-spherical local window. For VSVM, a fixed large local window is used instead of moving small local windows as was done for the DKG surrogate model. Sample reuse strategy is also proposed to improve the stability of VSVM.

1.1.3 Accuracy Improvement Strategies for the Kriging Method

By using VSVM, the computational cost in reliability evaluations can be reduced. However, the computational cost can be reduced further if the number of expensive computer simulations is reduced. Less DoE samples are required to construct accurate surrogates, if the accuracy of surrogates is improved for given DoE samples. The Kriging method is one of the widely used surrogate modeling methods [Sacks et al. 1989a, b; Cressie 1991; Barton 1994; Simpson et al. 2001b; Forrester et al. 2008; Forrester and Keane 2009; Dubourg et al. 2011] and it is also used in VSVM to improve the accuracy of SVM. One advantage of the Kriging method is that it is an interpolation method and not a regression method. Thus, this method reproduces the same responses at given sample locations, and it is appropriate to approximate deterministic computer experiments. Another advantage is that it provides uncertainty information at the prediction point on un-sampled region, which has motivated a number of adaptive sampling methods [Sacks et al. 1989a, b]. To construct an accurate Kriging model, an appropriate form of the Kriging model should be selected and the correlation parameters should be estimated accurately. In geostatistics, a sample variogram is usually plotted first from given data [Bohling 2005a; Hengl 2007; Roustant et al. 2012]. Based on the variogram, an appropriate correlation model is selected and parameters are estimated. However, this process requires users' knowledge on the Kriging method and it is not

practical to check all directional dependence in the empirical variogram with high dimensional problems. Therefore, the correlation model is usually fixed, and the maximum likelihood estimation (MLE) approach is applied to estimate parameters in many engineering applications [Martin and Simpson 2003, 2005; Gano et al. 2006; Martin 2009; Deng et al. 2011; Zhao et al. 2011]. Correlation parameter estimation using cross-validation can be an alternative to MLE, but MLE usually outperforms CV [Martin and Simpson 2005; Refaeilzadeh et al. 2009].

Zhao et al. showed the importance of parameter estimation in the Kriging model, and the generalized pattern search (GPS) algorithm was used to find the optimal parameter in MLE [Lewis and Torczon 1999; Zhao et al. 2011]. They found that GPS performed better than the Hooke and Jeeves (H-J) method [Lophaven et al. 2002; Martin 2009], the Levenberg-Marquardt (L-M) method [Martin 2009; Zhao et al. 2011], or genetic algorithm (GA) [Forrester and Keane 2009; Zhao et al. 2011]. However, performances of GPS are influenced by initial parameter values, and so it is important to use better initial values for GPS. Therefore, GA is proposed in this research to find better initial parameters for GPS.

In many engineering applications, the Gaussian correlation function is the most commonly used spatial correlation function (SCF) since it provides a relatively smooth and infinitely differentiable surface, which can be beneficial for gradient-based optimization algorithms [Martin and Simpson 2005; Gano et al. 2006; Martin 2009; Deng et al. 2011; Zhao et al. 2011]. However, there could be many different data structures, and thus the fixed correlation model may not be able to describe the given data well [Bohling 2005a; Roustant et al. 2012; Lophaven et al. 2002; Koehler and Owen 1996]. MLE is used for identifying the best correlation model among seven different SCF types in the literature [Lophaven et al. 2002].

When enough samples are not provided, performance of MLE is often not satisfactory due to inaccurate log-likelihood function [Fan and Li 2001; Li and Sudjianto

2005; Ginsbourger et al. 2009; Roustant 2012; Kok 2012]. To overcome this problem, penalized MLE (PMLE) is introduced, which includes the penalty function in addition to the log-likelihood function [Fan and Li 2001; Li and Sudjianto 2005]. However, PMLE is computationally more expensive due to costly CV estimations, and this research found that PMLE is not always more accurate than MLE. Thus, an appropriate condition should be applied for the usage of PMLE, and, in this research, PMLE is only applied when the log-likelihood function has flat region near the optimum.

Zhao et al. also showed that the accuracy of the Kriging method can be enhanced by selecting appropriate basis functions instead of using all possible basis functions [Zhao et al. 2011]. Even though the idea is noble, it is found that the process variance is not appropriate for selecting better mean structures. In this research, leave-one-out cross-validation (LOOCV) [Refaeilzadeh et al. 2009] is used instead of the process variance. Finally, these proposed improvements are carefully integrated and implemented to propose a new method.

1.2 Objectives of the Proposed Study

The first objective of this study is to propose an efficient sampling-based RBDO using VSVM. To show advantages of the proposed method, comparison studies will be carried out with EDSD and DKG. EDSD is based on SVM, which is a classification method and only classification information is used in EDSD. DKG is more accurate compared with other surrogates such as polynomial response surface, radial basis function, universal Kriging and blind Kriging [Joseph et al. 2008]. Therefore, EDSD and DKG are compared with VSVM. Sequential samples based on VSVM are inserted near the limit state function. For initial DoE samples, Latin hypercube sampling (LHS) [McKay et al. 1979; Queipo et al. 2005] is widely used for generating uniform samples in the hypercube. However, it becomes inefficient for hyper-spherical windows due to the curse of dimensionality [Lee et al. 2011]. Therefore, the Transformation/Gibbs sampling

method (TGS) [Cumbus et al. 1996] is implemented to provide uniform initial DoE samples in hyper-spherical local windows. VSVM is a classification method so it requires a balance between success samples and failure samples. Therefore, enlarged fixed local windows are used rather than moving local windows in RBDO as was done for surrogate modeling methods.

The second objective is to improve the accuracy of surrogates. By improving the accuracy of surrogates, the number of required DoE samples can be reduced to construct surrogates with targeted accuracy. Parameter estimation in MLE is improved by using GA for generating initial correlation parameters for GPS. When DoE sample size is small, PMLE is often more accurate than MLE. Performances of the Kriging method are dependent on correlation functions and mean structures. Therefore, the performance of surrogates can be enhanced by employing better correlation functions and mean structures.

1.3 Organization of Thesis

Chapter 2 presents basic concepts of MPP-based RBDO and sampling-based RBDO.

Chapter 3 presents surrogate modeling methods and classification methods. The new virtual support vector machine is proposed and compared with EDSD and DKG.

Chapter 4 presents different initial and sequential sampling methods. Hyper-spherical local windows and TGS are introduced.

Chapter 5 presents sampling-based RBDO using VSVM. Several efficiency strategies to reduce the computational cost are introduced. The proposed method is compared with previous sampling-based RBDO using DKG.

Chapter 6 presents how to improve the accuracy of the Kriging method. By using mathematical and engineering examples, performance of each accuracy improvement strategy is compared with existing Kriging method results.

Chapter 7 presents conclusions of the study. Several future research topics are proposed to improve the performance.

CHAPTER 2

DESIGN UNDER UNCERTAINTY

2.1 Introduction

This chapter presents review of fundamental concepts in design under uncertainty including MPP-based and sampling-based RBDO methods. Sections 2.2 and 2.3 discuss basic ideas of reliability analysis and inverse reliability analysis, which are necessary for RBDO. In Section 2.4, MPP-based RBDO using the first order reliability method (FORM) and the dimension reduction method (DRM) is explained. Sampling-based RBDO will be used for problems without sensitivity information in Section 2.5. In sampling-based RBDO, the Monte Carlo Simulation (MCS) is used to calculate probabilistic responses and sensitivities without any assumption in calculating the probability of failure for the performance function.

2.2 Reliability Analysis

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

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

where $\mathbf{X}=\{X_1, X_2, \cdots, X_{nr}\}^{\mathrm{T}}$ is an *nr*-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 the joint probability density function (PDF) of the random variables X. In most engineering applications, the exact evaluation of Eq. (2.1) is very difficult or often impossible to obtain since $f_{\mathbf{X}}(\mathbf{x})$ is non-Gaussian in general and $G(\mathbf{X})$ can be highly nonlinear. To take care of the non-Gaussian $f_{\mathbf{X}}(\mathbf{x})$, a transformation from the original X-space into the independent standard normal U-space is introduced [Rosenblatt 1952]. For highly nonlinear $G(\mathbf{X})$, $G(\mathbf{X})$ is approximated using first order Taylor series expansion in the

first order reliability method (FORM); or second order Taylor series expansion in the second order reliability method (SORM).

2.2.1 Transformation

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

$$T:\begin{cases} u_1=\Phi^{-1}\left[F_{X_1}\left(x_1\right)\right] \\ u_2=\Phi^{-1}\left[F_{X_2}\left(x_2\middle|x_1\right)\right] \\ \quad\vdots \\ u_{nr}=\Phi^{-1}\left[F_{X_{nr}}\left(x_{nr}\middle|x_1,x_2,\cdots,x_{nr-1}\right)\right] \end{cases} \tag{2.2}$$

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

$$F_{X_i}\left(x_i\middle|x_1,x_2,\cdots,x_{i-1}\right)=\frac{\int_{-\infty}^{x_i} f_{X_1X_2\cdots X_i}(x_1,x_2,\cdots,x_{i-1},\xi)d\xi}{f_{X_1X_2\cdots X_{i-1}}(x_1,x_2,\cdots,x_{i-1})} \tag{2.3}$$

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

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

where ϕ is the standard normal PDF.

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

$$T^{-1}:\begin{cases} x_1 = F_{X_1}^{-1}\left[\Phi(u_1)\right] \\ x_2 = F_{X_2}^{-1}\left[\Phi(u_2|x_1)\right] \\ \vdots \\ x_{nr} = F_{X_{nr}}^{-1}\left[\Phi(u_{nr}|x_1, x_2, \cdots, x_{nr-1})\right] \end{cases} \tag{2.5}$$

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

$$f_{\mathbf{X}}(\mathbf{x}) = f_{X_1}(x_1) \times f_{X_2}(x_2) \times \cdots \times f_{X_{nr}}(x_{nr}) \tag{2.6}$$

where $f_{X_i}(x_i)$ are the marginal PDF's. Then, Rosenblatt transformation and the inverse transformation are simplified as

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

where $F_{X_i}(x_i)$ are the marginal CDF's. Table 2.1 shows five representative distributions and their transformations assuming random variables are independent.

Table 2.1 Probability Distribution and Its Transformation between X and U-space

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

Table 2.1. Continued

Uniform	$\mu = \frac{a+b}{2}$, $\sigma = \frac{b-a}{\sqrt{12}}$	$f(x) = \frac{1}{b-a}$, $a \le x \le b$	$X = a + (b-a)\Phi(U)$

2.2.2 First Order Reliability Method (FORM) and Second Order Reliability Method (SORM)

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 U-space as shown in Fig. 2.1. In this study, the performance function in 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

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

The distance from MPP to the origin is commonly called the Hasofer-Lind reliability index [Hasofer and Lind 1974] and denoted by β_{HL} . Using the reliability index β_{HL} , FORM can approximate the probability of failure using a linear approximation of the performance function as

$$P_F^{\mathrm{FORM}} \cong \Phi(-\beta_{\mathrm{HL}}). \tag{2.9}$$

The probability of failure also can be calculated using SORM, which uses a quadratic approximation of the performance function in U-space and the rotational

transformation from the standard normal U-space to the rotated standard normal V-space [Breitung 1984; Hohenbichler and Rackwitz 1988; Rahman and Wei 2006].

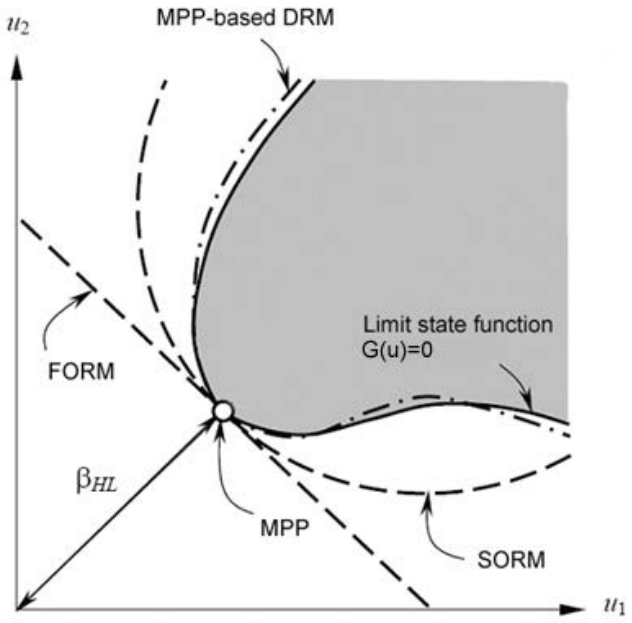

Figure 2.1 MPP and Reliability Index β_{HL} in U-Space

Source: Wei, D. "A Univariate Decomposition Method For Higher-Order Reliability Analysis And Design Optimization," Ph. D. Thesis, University of Iowa, 2006.

2.3 Inverse Reliability Analysis

The reliability analysis presented in Section 2.2 is called the reliability index approach (RIA) [Tu et al. 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. However, the inverse reliability analysis in the performance measure approach (PMA) [Tu et al. 1999, 2001; Choi et al. 2001; Youn et

al. 2003] is known to be numerically more efficient and stable compared to RIA. In PMA, the probability of failure is not calculated directly. Instead, PMA judges whether or not a given design satisfies the probabilistic constraint for a given target probability of failure P_F^{Tar}. The optimization problem for PMA is expressed as

$$\begin{aligned} &\text{maximize} \quad g(\mathbf{u}) \\ &\text{subject to} \quad \|\mathbf{u}\| = \beta_t \end{aligned} \tag{2.10}$$

where β_t is the target reliability index. Eq. (2.10) is the inverse problem of Eq. (2.8), therefore, this is called the inverse reliability analysis. 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 index β_t.

To find the MPP using the inverse reliability analysis with the given target reliability index β_t, there exist several methods available such as 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. 2005].

The MV method linearly approximates the performance function using the function and gradient information at the mean value in standard normal U-space. This MV method is a crude method to find MPP of the inverse reliability analysis. However, since it does not require additional function evaluation and sensitivity analysis, MPP by the 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 AMV method uses the MPP obtained by the MV method as its first iteration. AMV uses the gradient at the MPP obtained by the MV method to find next MPP candidate and the iteration continues until the approximate MPP converges to the correct

MPP. This AMV method is known as an efficient method when the constraint function is convex.

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

2.4 MPP-based RBDO

2.4.1 MPP-based RBDO Using FORM

The mathematical formulation of a general RBDO problem is expressed as

$$\begin{aligned} &\text{minimize} && \text{Cost}(\mathbf{d}) \\ &\text{subject to} && P[G_i(\mathbf{X}) > 0] \le P_{F_i}^{\text{Tar}},\ i = 1, \cdots, nc \\ & && \mathbf{d}^L \le \mathbf{d} \le \mathbf{d}^U, \quad \mathbf{d} \in \mathbf{R}^{nd} \text{ and } \mathbf{X} \in \mathbf{R}^{nr} \end{aligned} \tag{2.11}$$

where $\mathbf{d} = \{d_i\}^{\mathrm{T}} = \boldsymbol{\mu}(\mathbf{X}), i = 1 \sim nd,$ is the design vector; $\mathbf{X} = \{X_i\}^{\mathrm{T}}$ is the random vector; and *nc*, *nd* and *nr* 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

$$P[G_i(\mathbf{X}) > 0] - P_{F_i}^{\text{Tar}} \le 0 \Rightarrow G_i(\mathbf{x}^*) \le 0 \tag{2.12}$$

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

Using FORM, Eq. (2.11) can be reformulated to

$$\begin{aligned} &\text{minimize} && \text{Cost}(\mathbf{d}) \\ &\text{subject to} && P[G_i(\mathbf{X}) > 0] \le P_{F_i}^{\text{Tar}} = \Phi(-\beta_{t_i}),\ i = 1, \cdots, nc \\ & && \mathbf{d}^L \le \mathbf{d} \le \mathbf{d}^U, \quad \mathbf{d} \in \mathbf{R}^{nd} \text{ and } \mathbf{X} \in \mathbf{R}^{nr} \end{aligned} \tag{2.13}$$

where β_{t_i} is the target reliability index for the i^{th} constraint and the probabilistic constraint can be changed into

$$P[G_i(\mathbf{X}) > 0] - \Phi(-\beta_{t_i}) \le 0 \Rightarrow G_i\,(\mathbf{x}_{\text{FORM}}^*) \le 0 \tag{2.14}$$

where $\mathbf{x}_{\text{FORM}}^*$ is the FORM-based MPP.

To solve in Eq. (2.13), it is required to calculate the sensitivity of the probabilistic constraint in Eq. (2.14) 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

$$\frac{\partial G(\mathbf{x}^*)}{\partial \mathbf{d}} = \left.\frac{\partial G}{\partial \mathbf{d}}\right|_{\mathbf{x}=\mathbf{x}^*} = \sum_{i=1}^{nr} \left.\frac{\partial G}{\partial x_i}\right|_{\mathbf{x}=\mathbf{x}^*} \left.\frac{\partial x_i}{\partial \mathbf{d}}\right|_{\mathbf{x}=\mathbf{x}^*} = \left[\left.\frac{\partial \mathbf{x}}{\partial \mathbf{d}}\right]_{\mathbf{x}=\mathbf{x}^*}^{\mathrm{T}} \left.\frac{\partial G}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^*}\right. \tag{2.15}$$

and Eq. (2.15) can be further simplified as [Gumbert et al. 2003; Hou et al. 2004]

$$\frac{\partial G(\mathbf{x}^*)}{\partial \mathbf{d}} = \left[\left.\frac{\partial \mathbf{x}}{\partial \mathbf{d}}\right]_{\mathbf{x}=\mathbf{x}^*}^{\mathrm{T}} \left.\frac{\partial G}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^*} = \left.\frac{\partial G}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^*}\right. . \tag{2.16}$$

2.4.2 MPP-based RBDO Using DRM

The dimension reduction method [Xu and Rahman 2004, Rahman and Xu 2004; Wei 2006] approximates the multi-dimensional integration of a performance function using a function with reduced dimension. A univariate dimension reduction method is an additive decomposition of *nr*-dimensional performance function into one-dimensional

functions and an *nr*-dimensional performance function $G(\mathbf{X})$ can be additively decomposed into one-dimensional functions at the MPP of the random vector $\mathbf{X}$ as

$$G(\mathbf{X}) \cong \hat{G}(\mathbf{X}) \equiv \sum_{i=1}^{nr} G(x_1^*, \cdots, x_{i-1}^*, X_i, x_{i+1}^*, \cdots, x_{nr}^*) - (nr-1)G(\mathbf{x}^*) \qquad (2.17)$$

where $\mathbf{x}^* = \{x_1^*, x_2^*, \cdots, x_{nr}^*\}^{\mathrm{T}}$ is the FORM-based MPP of the performance function $G(\mathbf{X})$ obtained from Eq. (2.10) and *nr* is the number of random variables. This MPP-based univariate DRM provides more accurate reliability analysis result compared to FORM.

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, these approximations can be inaccurate and lead to unreliable designs. Furthermore, sensitivity information, which can be difficult or impossible to obtain in many real applications, is required to perform MPP-based RBDO. Therefore, a sampling-based RBDO is proposed and it 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 Sampling-Based Probability of Failure

In the 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

$$P_F(\boldsymbol{\psi}) \equiv P[\mathbf{X} \in \Omega_F] = \int_{\mathbb{R}^{nr}} I_{\Omega_F}(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}; \boldsymbol{\psi}) d\mathbf{x} = E\left[I_{\Omega_F}(\mathbf{X}) \right] \qquad (2.18)$$

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 $\mathbf{X} = \{X_1, \cdots, X_{nr}\}^{\mathrm{T}}$, $P[\bullet]$ 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 **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 $G_i(\mathbf{x})$, $\Omega_F \equiv \left\{\mathbf{x} : \bigcup_{i=1}^{nc} G_i(\mathbf{x}) > 0\right\}$ and $\Omega_F \equiv \left\{\mathbf{x} : \bigcap_{i=1}^{nc} G_i(\mathbf{x}) > 0\right\}$ for series system and parallel system reliability analysis of *nc* performance functions, respectively. $I_{\Omega_F}(\mathbf{x})$ in Eq. (2.18) is called an indicator function and defined as

$$I_{\Omega_F}(\mathbf{x}) \equiv \begin{cases} 1, & \mathbf{x} \in \Omega_F \\ 0, & \text{otherwise} \end{cases} \tag{2.19}$$

In this research, since the mean of **X**, $\boldsymbol{\mu} = \{\mu_1, \cdots, \mu_{nr}\}^{\mathrm{T}}$, is used as a design vector, the vector of distribution parameters $\boldsymbol{\psi}$ can be simply replaced with $\boldsymbol{\mu}$ for the computation of the probability of failure in Eq. (2.18).

2.5.2 Probabilistic Sensitivity Analysis

For the derivation of the sensitivity of the probability of failure, the following four regularity conditions need to be satisfied [Zhao 2011].

1. The joint PDF $f_{\mathbf{X}}(\mathbf{x};\boldsymbol{\mu})$ is continuous.
2. The mean $\mu_i \in \mathrm{M}_i \subset \mathbf{R}$, $i = 1, \cdots, nr$, where M_i is an open interval on **R**.
3. The partial derivative $\dfrac{\partial f_{\mathbf{X}}(\mathbf{x};\boldsymbol{\mu})}{\partial \mu_i}$ exists and is finite for all **x** and μ_i. In addition, $P_F(\boldsymbol{\mu})$ is a differentiable function of $\boldsymbol{\mu}$.
4. There exists a Lebesgue integrable dominating function $r(\mathbf{x})$ such that

$$\left| g(\mathbf{x}) \frac{\partial f_{\mathbf{X}}(\mathbf{x};\boldsymbol{\mu})}{\partial \mu_i} \right| \le r(\mathbf{x}) \tag{2.20}$$

for all $\boldsymbol{\mu}$.

With the four assumptions satisfied, take the partial derivative of Eq. (2.18) with respect to μ_i and use the interchangeability between the differential and integral

operators, then the sensitivity of the probability of failure can be obtained as [Lee et al. 2011]

$$\begin{aligned}\frac{\partial P_F(\boldsymbol{\mu})}{\partial \mu_i} &= \frac{\partial}{\partial \mu_i}\int_{\mathbb{R}^{nr}} I_{\Omega_F}(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x};\boldsymbol{\mu}) d\mathbf{x} \\ &= \int_{\mathbb{R}^{nr}} I_{\Omega_F}(\mathbf{x}) \frac{\partial f_{\mathbf{X}}(\mathbf{x};\boldsymbol{\mu})}{\partial \mu_i} d\mathbf{x} \\ &= \int_{\mathbb{R}^{nr}} I_{\Omega_F}(\mathbf{x}) \frac{\partial \ln f_{\mathbf{X}}(\mathbf{x};\boldsymbol{\mu})}{\partial \mu_i} f_{\mathbf{X}}(\mathbf{x};\boldsymbol{\mu}) d\mathbf{x} \\ &= E\left[I_{\Omega_F}(\mathbf{x}) \frac{\partial \ln f_{\mathbf{X}}(\mathbf{x};\boldsymbol{\mu})}{\partial \mu_i} \right].\end{aligned} \tag{2.21}$$

The partial derivative of the log function of the joint PDF in Eq. (2.21) with respect to μ_i is known as the first-order score function for μ_i and is denoted as

$$s^{(1)}_{\mu_i}(\mathbf{x};\boldsymbol{\mu}) \equiv \frac{\partial \ln f_{\mathbf{X}}(\mathbf{x};\boldsymbol{\mu})}{\partial \mu_i}. \tag{2.22}$$

As shown in Eq. (2.22), the proposed probabilistic sensitivity analysis using the first-order score function for μ_i does not depend on the sensitivity of the response $G_j(\mathbf{x})$. Instead, it uses the sensitivity of the input joint distribution, which can be obtained analytically. This can be shown in Fig. 2.2. Assume the horizontal axis represents the multiple dimensional random variable $\mathbf{x} = [x_1, x_2, ..., x_{nr}]^{\mathrm{T}}$ with $G_j(\mathbf{x}) > 0$ as the failure region for j^{th} constraint $G_j(\mathbf{x})$. The input joint PDF $f_{\mathbf{X}}(\mathbf{x};\boldsymbol{\mu})$ is shown in the Fig. 2.2. At the given design point **μ**, if we are considering the deterministic design optimization, the design sensitivity of the constraint function $G_j(\mathbf{x})$ at point A needs to be used. On the other hand, for the probabilistic constraint (i.e., probability of failure which is the volume of the gray shaded region under the input joint PDF), the input joint PDF $f_{\mathbf{X}}(\mathbf{x};\boldsymbol{\mu})$ will move as the design point **μ** moves and the rate change of the probabilistic constraint will

depend on the slope of the natural logarithm of the input joint PDF $f_{\mathbf{X}}(\mathbf{x};\boldsymbol{\mu})$ at point B as shown in Fig. 2.2 and Eq. (2.22).

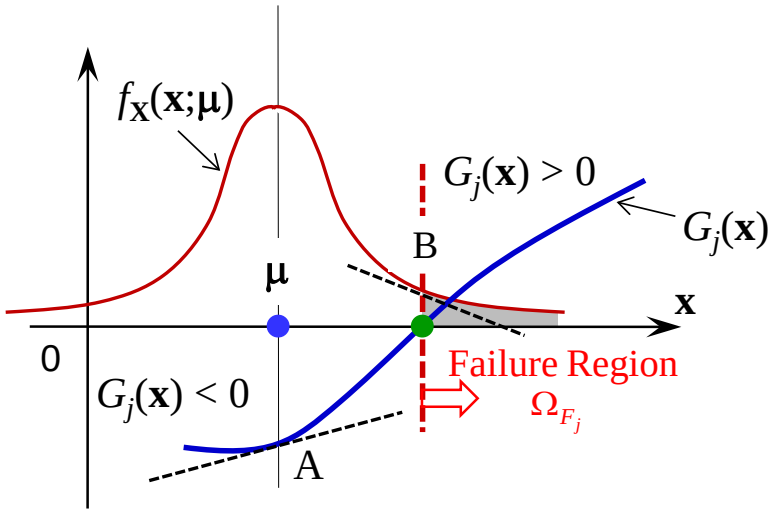

Figure 2.2 Probabilistic Sensitivity Analysis and First Order Score Function

For independent input random variables, the first order score function for μ_i in Eq. (2.22) can be expressed as

$$s_{\mu_i}^{(1)}(\mathbf{x};\boldsymbol{\mu}) \equiv \frac{\partial \ln f_{\mathbf{X}}(\mathbf{x};\boldsymbol{\mu})}{\partial \mu_i} = \frac{\partial \ln f_{X_i}(x_i;\mu_i)}{\partial \mu_i}. \tag{2.23}$$

Since the marginal PDF and the CDF are available analytically as listed in Table 2.2, the derivation of the first-order score function for the statistically independent random input is very straightforward as listed in Table 2.3 [Lee et al. 2011].

Table 2.2 Marginal PDF, CDF, and Parameters

	PDF, $f_X(x)$	CDF, $F_X(x)$	Parameters
Normal	$\frac{1}{\sqrt{2\pi}\sigma}e^{-0.5[\frac{x-\mu}{\sigma}]^2}$	$\Phi\left(\frac{x-\mu}{\sigma}\right)$	μ, σ
Log-normal	$\frac{1}{\sqrt{2\pi}x\bar{\sigma}}e^{-0.5[\frac{\ln x-\bar{\mu}}{\bar{\sigma}}]^2}$	$\Phi\left(\frac{\ln x-\bar{\mu}}{\bar{\sigma}}\right)$	$\bar{\sigma}^2=\ln[1+(\frac{\sigma}{\mu})^2]$, $\bar{\mu}=\ln(\mu)-0.5\bar{\sigma}^2$
Gumbel	$\alpha e^{-\alpha(x-\nu)-e^{-\alpha(x-\nu)}}$	$\alpha e^{-e^{-\alpha(x-\nu)}}$	$\mu=\nu+\frac{0.577}{\alpha}$, $\sigma=\frac{\pi}{\sqrt{6}\alpha}$
Weibull	$\frac{k}{\nu}(\frac{x}{\nu})^{k-1}e^{-(\frac{x}{\nu})^k}$	$1-e^{-(\frac{x}{\nu})^k}$	$\mu=\nu\Gamma(1+\frac{1}{k})$, $\sigma^2=\nu^2[\Gamma(1+\frac{2}{k})-\Gamma^2(1+\frac{1}{k})]$

Table 2.3 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	$\alpha_i-\alpha_i e^{-\alpha_i(x_i-\nu_i)}$
Weibull	$\frac{1}{k_i}\frac{\partial k_i}{\partial\mu_i}-\frac{1}{\nu_i}\frac{\partial\nu_i}{\partial\mu_i}+\frac{\partial k_i}{\partial\mu_i}\ln\frac{x_i}{\nu_i}-\frac{(k_i-1)}{\nu_i}\frac{\partial\nu_i}{\partial\mu_i}-(\frac{x_i}{\nu_i})^{k_i}\left(\frac{\partial k_i}{\partial\mu_i}\ln\frac{x_i}{\nu_i}-\frac{k_i}{\nu_i}\frac{\partial\nu_i}{\partial\mu_i}\right)$

For a bivariate correlated random input $\mathbf{X}=\{X_i,X_j\}^{\mathrm{T}}$, the joint PDF of $\mathbf{X}$ is expressed as [Noh et al. 2009; Noh et al. 2010; Lee et al. 2011]

$$f_{\mathbf{X}}(\mathbf{x};\boldsymbol{\mu})=\frac{\partial^2 C(u,v;\theta)}{\partial u\partial v}f_{X_i}(x_i;\mu_i)f_{X_j}(x_j;\mu_j)=C_{,uv}(u,v;\theta)f_{X_i}(x_i;\mu_i)f_{X_j}(x_j;\mu_j) \quad (2.24)$$

where C is the copula function, $u = F_{X_i}(x_i;\mu_i)$ and $v = F_{X_j}(x_j;\mu_j)$ are marginal CDF's for X_i and X_j, respectively, and θ is the correlation coefficient between X_i and X_j. The partial derivative of the copula function with respect to u and v is called the copula density function and denoted as

$$c(u,v;\theta) \equiv \frac{\partial^2 C(u,v;\theta)}{\partial u \partial v} = C_{,uv}(u,v;\theta). \tag{2.25}$$

Using Eq. (2.24), the first order score function in Eq. (2.22) for a bivariate correlated **X** are expressed as

$$s^{(1)}_{\mu_i}(\mathbf{x};\boldsymbol{\mu}) \equiv \frac{\partial \ln f_{\mathbf{X}}(\mathbf{x};\boldsymbol{\mu})}{\partial \mu_i} = \frac{\partial \ln c(u,v;\theta)}{\partial \mu_i} + \frac{\partial \ln f_{X_i}(x_i;\mu_i)}{\partial \mu_i} \tag{2.26}$$

The derivation of the first term of the right side of Eq. (2.26) is listed in Table 2.4. Therefore, sensitivity of the probability of failure in Eq. (2.21) can be obtained for bivariate correlated random input variables.

Table 2.4. Log-Derivative of Copula Density Function

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

Table 2.4. Continued

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

Since it is computationally very expensive to compute the probability of failure using true responses for Eq. (2.18), surrogate models are used. Denote the surrogate model for the j^{th} constraint function $G_j(\mathbf{x})$ as $\hat{G}_j(\mathbf{x})$. Then the probability of failure can be approximated as

$$P_{F_j} \equiv P\left[G_j(\mathbf{x})>0\right] \cong \frac{1}{M}\sum_{m=1}^{M} I_{\hat{\Omega}_{F_j}}(\mathbf{x}^{(m)}) \le P_{F_j}^{\text{Tar}} \tag{2.27}$$

where M is the number of MCS samples, $\mathbf{x}^{(m)}$ is the m^{th} realization of $\mathbf{X}$, and the failure region $\hat{\Omega}_{F_j}$ for the surrogate model is defined as $\hat{\Omega}_{F_j} \equiv \{\mathbf{x}:\hat{G}_j(\mathbf{x})>0\}$. The sensitivity of the probabilistic constraint in Eq. (2.21) can be approximated as

$$\frac{\partial P_{F_j}}{\partial \mu_i} \cong \frac{1}{M}\sum_{m=1}^{M} I_{\hat{\Omega}_{F_j}}(\mathbf{x}^{(m)}) s_{\mu_i}^{(1)}(\mathbf{x}^{(m)};\boldsymbol{\mu}). \tag{2.28}$$

Accuracy of the MCS is dependent on the number of MCS samples and the target probability of failure. Based on the 95% confidence interval of the estimated probability of failure, the percentage error can be defines as [Haldar and Mahadevan 2000]

$$\varepsilon_{MCS} = \sqrt{\frac{(1-P_F^{\text{Tar}})}{M\times P_F^{\text{Tar}}}}\times 200\%. \tag{2.29}$$

Therefore, for small target probability of failure, the number of MCS samples *M* needs to be increased to maintain target accuracy level.

In this Section, only component probabilities of failure are considered. However, if the failure set is appropriately defined for the system reliability, Eqs. (2.27) and (2.28) can be also used for the system RBDO. Therefore, only the component RBDO will be considered in this research without loss of generality. Equations (2.27) and (2.28) require only success or failure information based on the limit state obtained from the surrogate model $\hat{G}_j(\mathbf{x})$. Therefore, classification methods can be used effectively for RBDO.

CHAPTER 3
EFFICIENT SURROGATE MODELING METHOD FOR RELIABILITY ANALYSIS

3.1 Introduction

Many industrial engineering simulation models are computationally expensive or without sensitivity information. Therefore, surrogate models are often used to carry out reliability analysis and RBDO. The Kriging method is one of widely used surrogates, but its functional form is complicated. In this chapter, an efficient classification methodology is developed for reliability analysis and RBDO while maintaining an accuracy level similar to or better than existing response surface methods. The Kriging method and dynamic Kriging method are reviewed in Section 3.2. Support vector machines and virtual SVM (VSVM) are presented in Sections 3.3 and 3.4. They include basic ideas of SVM and VSVM. And, an adaptive sequential sampling for VSVM is explained in Section 3.5. In Section 3.6, comparison study results are shown using three different examples. Finally, conclusion is followed in Section 0.

3.2 Dynamic Kriging Method

3.2.1 The Kriging Method

The Kriging method became popular for constructing surrogates of deterministic computer simulations in recent years. In the Kriging method, responses are considered as a realization of a stochastic process. Consider *N* design of experiment (DoE) samples $\mathbf{x}_{DoE} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N)^{\mathrm{T}}$ with *N* responses $\mathbf{y} = (y_1, y_2, ..., y_N)^{\mathrm{T}}$, where $\mathbf{x}_{DoE} \in \mathbf{R}^m$ and *m* is the number of input variables. In a Kriging model, there are two parts, and the mathematical form can be expressed as,

$$\mathbf{y} = \mathbf{F}\boldsymbol{\beta} + \mathbf{Z} \tag{3.1}$$

where $\mathbf{F}\boldsymbol{\beta}$ is the mean structure of the response, $\mathbf{F} = [\mathbf{f}_j(\mathbf{x}_i)], i = 1, 2, ..., N,\ j = 1, 2, ..., K$ is an $N \times K$ design matrix, and $\boldsymbol{\beta} = [\beta_1, \beta_2, ..., \beta_K]^{\mathrm{T}}$ are the regression coefficients from the generalized least square regression method. **Z** is a model of a Gaussian random process with zero mean and covariance:

$$V(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2 R(\boldsymbol{\theta}, \mathbf{x}_i, \mathbf{x}_j) \tag{3.2}$$

where σ^2 is the process variance, R is the spatial correlation function (SCF), $\boldsymbol{\theta}$ is the correlation function parameter, and $\mathbf{x}_i, \mathbf{x}_j$ are two sampled sites. For engineering problems, the Gaussian function is widely applied for its SCF, since it is infinitely differentiable and provides a smooth response surface, which is beneficial with gradient-based optimization algorithms [Simpson et al. 2001b; Martin 2009]. The mathematical form of the Gaussian correlation function is expressed as

$$R(\boldsymbol{\theta}, \mathbf{x}_i, \mathbf{x}_j) = \prod_{l=1}^{m} \exp(-\theta^{(l)} (x_i^{(l)} - x_j^{(l)})^2) \tag{3.3}$$

The universal Kriging (UKG) method is defined with a set of basis functions:

$$\mathbf{f}(\mathbf{x}) = \{ f_1(\mathbf{x}),\ f_2(\mathbf{x}), \ldots,\ f_K(\mathbf{x}) \}^{\mathrm{T}}. \tag{3.4}$$

Polynomials are widely used for the basis functions in UKG. The ordinary Kriging (OKG) method is a special case of universal Kriging when its basis function is expressed as

$$\mathbf{f}(\mathbf{x}) = \{1\}^{\mathrm{T}}. \tag{3.5}$$

The Kriging prediction is the best linear unbiased estimator (BLUE) and can be expressed as

$$\hat{y}(\mathbf{x}) = \mathbf{f}^{\mathrm{T}}(\mathbf{x})\boldsymbol{\beta} + \mathbf{r}^{\mathrm{T}}(\mathbf{x})\mathbf{R}^{-1}(\mathbf{y} - \mathbf{F}\boldsymbol{\beta}) \tag{3.6}$$

where **y** is true responses at known samples, $\mathbf{r}(\mathbf{x}) = \{R(\boldsymbol{\theta}, \mathbf{x}, \mathbf{x}_1), R(\boldsymbol{\theta}, \mathbf{x}, \mathbf{x}_2), ...R(\boldsymbol{\theta}, \mathbf{x}, \mathbf{x}_N)\}^{\mathrm{T}}$; the generalized least-squares estimate of **β** is

$$\hat{\boldsymbol{\beta}}(\mathbf{x}) = (\mathbf{F}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{F})^{-1}\mathbf{F}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{y}. \tag{3.7}$$

The optimal correlation parameter **θ** is often chosen using the maximum likelihood estimation (MLE). Under the Gaussian assumption, the log-likelihood of the model parameters is defined as

$$L = -\frac{N}{2}\ln[2\pi\sigma^2] - \frac{1}{2}\ln[|\mathbf{R}(\boldsymbol{\theta})|] - \frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{F}\boldsymbol{\beta})^{\mathrm{T}}\mathbf{R}(\boldsymbol{\theta})^{-1}(\mathbf{y} - \mathbf{F}\boldsymbol{\beta}). \tag{3.8}$$

By taking the derivative of Eq. (2.1) with respect to **β** and σ^2 and using Eq. (3.7), σ^2 is estimated as

$$\hat{\sigma}^2 = \frac{1}{N}(\mathbf{y} - \mathbf{F}\hat{\boldsymbol{\beta}})^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y} - \mathbf{F}\hat{\boldsymbol{\beta}}). \tag{3.9}$$

Therefore, the best **θ** which maximizes Eq. (2.1) can be found using optimization algorithms.

3.2.2 Dynamic Kriging Method

There are a number of different methods to estimate the correlation parameter θ in MLE such as the downhill simplex method [Martin and Simpson 2005; Martin 2009; Deng et al. 2011], the Newton-Raphson method [Martin 2009; Deng et al. 2011], the quasi-Newton method [Martin and Simpson 2005; Gano et al. 2006; Deng et al. 2011], the Fisher scoring algorithm [Martin 2009; Deng et al. 2011], the adaptive simulated annealing [Gano et al. 2006; Deng et al. 2011], the genetic algorithm [Forrester and

Keane 2009; Zhao et al. 2011] and generalized pattern search algorithm [Gano et al. 2006; Deng et al. 2011; Zhao et al. 2011]. Gradient-based algorithms often perform poorly for highly nonlinear problems, thus global optimization algorithms are often used. The generalized pattern search algorithm (GPS) showed better performance compared with the Hooke and Jeeves method [Lophaven et al. 2002], Levenberg-Marquardt (L-M) method [Martin 2009], or genetic algorithm (GA) [Zhao et al. 2011], therefore, GPS is used in DKG.

For the UKG method, the basis function $\mathbf{f}(\mathbf{x})$ in Eq. (3.4) is fixed during the entire surrogate modeling process. However, the best basis function set is different for different data and problems; and a larger basis function set is not always better than the smaller basis function set. Therefore, in DKG, the best basis function selection process is expressed as

$$\begin{aligned} &\text{Find the basis function set} \\ &\text{to minimize the process variance } \hat{\sigma}^2 = \frac{1}{N}(\mathbf{y}-\mathbf{F}\hat{\boldsymbol{\beta}})^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-\mathbf{F}\hat{\boldsymbol{\beta}}). \end{aligned} \tag{3.10}$$

The GA is applied in this selection process, since the exhaustive algorithm is computationally expensive while performances are similar.

3.3 Support Vector Machines (SVM)

An SVM is a machine-learning concept used for the classification of data in pattern recognition [Vapnik 1998, 2000; Cherkassky and Mulier 1998; Scholkopf 1999; Scholkopf et al. 1999; Cristianini and Shawe-Taylor 2000; Kecman 2001, 2005; Scholkopf and Smola 2002; Ben-Hur and Weston 2008; Basudhar et al. 2012; Basudhar and Missoum 2008, 2010]. It has the ability to explicitly construct a multidimensional and complex decision function that optimally separates multiple classes of data. Even though SVM is able to deal with multi-class cases, only two classes – success or failure –

are used in reliability analyses, and thus only a two-class classification problem will be considered in this research. Good features for the high-dimensional problem make SVM an appropriate method for the formulation of the explicit limit state function. In this section, an overview of SVM is presented, including basic ideas and some important features.

3.3.1 Linear SVM

For the given multidimensional problem, *N* samples are distributed within the given window. Each sample $\mathbf{x}_i$ is associated with one of two classes characterized by values $yc_i = \pm 1$, which represents a success (−1) or a failure (+1). The SVM algorithm constructs the decision function that optimally separates two classes of samples. The corresponding linear explicit boundary function is expressed as

$$s(\mathbf{x}) = b + \sum_{i=1}^{N} \alpha_i yc_i (\mathbf{x}_i \cdot \mathbf{x}) \tag{3.11}$$

where *b* is the bias, α_i are Lagrange multipliers obtained from the quadratic programming optimization problem used to construct SVM, and **x** is an arbitrary point to be predicted. The classification of any arbitrary point **x** is given by the sign of *s*(**x**) in Eq. (3.11). The optimization process is used to solve for the optimal SVM decision function with a maximal margin. Figure 3.1 shows a linear SVM result, and the notion of margin can be easily noticed. In this case, the margin is the shortest distance between two parallel hyperplanes given by *s*(**x**) = ±1 in the design space. These hyperplanes are called support hyperplanes and pass through one or several samples, which are called support vectors. The SVM optimization process also does not allow any samples to exist between two hyperplanes.

The Lagrange multipliers associated with the support vectors are positive while the other Lagrange multipliers are zero. It means that the explicit SVM decision function

uses only support vectors in its formulation, and thus an SVM constructed only with support vectors is identical to the one obtained with all samples. Typically, the number of support vectors is much smaller than the number of samples *N*.

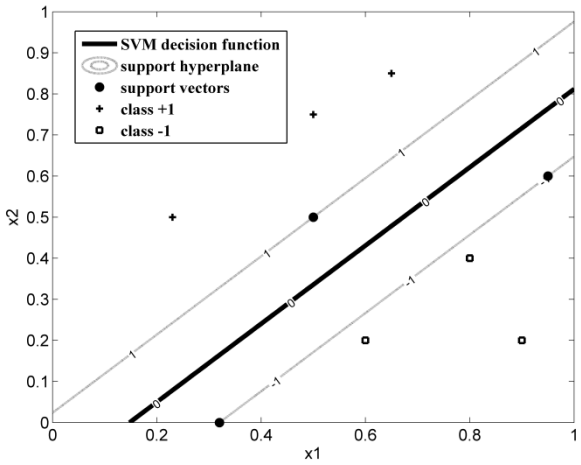

Figure 3.1 Linear Decision Function for Two-Dimensional Problem

3.3.2 Nonlinear SVM and Kernel Functions

To construct nonlinear decision functions, kernels are introduced in SVM. In the formulation of the SVM decision function, it is assumed that there exists a higher-dimensional space where the transformed data can be linearly separated. The transformation from the original design space to the higher-dimensional space is based on the kernel function $K(\mathbf{x}_i, \mathbf{x})$ in SVM. The nonlinear decision function is expressed as

$$s(\mathbf{x}) = b + \sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) \tag{3.12}$$

Instead of the linear function $(\mathbf{x}_i \cdot \mathbf{x})$ in Eq. (3.11), the nonlinear kernel function $K(\mathbf{x}_i, \mathbf{x})$ is used in the form of nonlinear decision function. Therefore, $s(\mathbf{x})$ is linear with respect to $K(\mathbf{x}_i, \mathbf{x})$ and nonlinear with respect to $\mathbf{x}$.

The kernel $K(\mathbf{x}_i, \mathbf{x})$ in the SVM equation can have different forms, such as polynomial, Gaussian, Sigmoid, etc. The Gaussian kernel is applied in this research, since the Gaussian kernel can deal with both linear and nonlinear problems [Lin and Lin, 2003; Hsu et al. 2004]. Furthermore, the number of hyper-parameters is relatively small and the Gaussian kernel has fewer numerical difficulties [Hsu et al. 2004]. For example, the kernel function K is equal to or less than 1 in contrast to polynomial kernels which may go to infinity or zero [Vapnik 2000; Hsu et al. 2004]. The form of the Gaussian kernel is given as [Vapnik 1998, 2000; Cherkassky and Mulier 1998; Scholkopf 1999; Kecman 2001, 2005; Scholkopf and Smola 2002; Ben-Hur and Weston 2008]:

$$K(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\frac{\left\| \mathbf{x} - \mathbf{x}_i \right\|^2}{2\gamma^2} \right) \tag{3.13}$$

where γ is the parameter of the Gaussian kernel. Figure 3.2 is an example of nonlinear SVM decision function with the Gaussian kernel for a two-dimensional problem. Even though the boundary is always linear in the transformed higher-dimensional space, the boundary is nonlinear in the original design space. In this research, *SVM and Kernel Methods Matlab* toolbox [Canu et al. 2005] is used for the formulation of SVM.

The SVM can deal with high-dimensional problems and can separate two classes of data with the maximal margin. The SVM decision function has an explicit form, and thus predictions based on SVM can be computed faster for MCS than those based on implicit surrogate methods such as the Kriging method. The prediction speed is important for sampling-based RBDO, since a very large number of MCS points are required in evaluating the probability of failure.

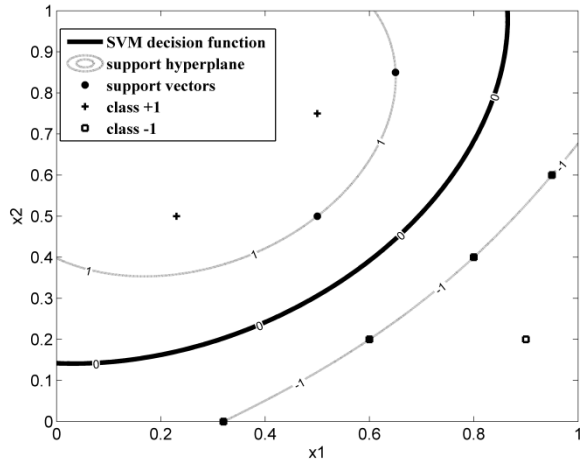

Figure 3.2 Nonlinear Decision Function for Two-Dimensional Problem

3.3.3 Parameter Estimation in SVM

It is very important to use appropriate kernel parameter value for γ in Eq. (3.13) because the nonlinearity of SVM model depends on the parameter γ. A number of parameter estimation methods have been proposed for SVM [Chapelle and Vapnik 1999; Seeger 2000; Vapnik 2000; Cawley 2001; Chapelle et al. 2002; Keerthi 2002; Momma and Bennett 2002; Staelin 2002; Ali and Smith 2003; Chung et al. 2003; Wang et al. 2003; Kulkarni et al. 2004; Frohlich and Zell 2005; Gold and Sollich 2005; Boardman and Trappenberg 2006; Huang and Wang 2006; Men and Wang 2008; Wang and Ma 2008]. There are mainly three ways to evaluate the performance of the kernel parameter. If DoE samples are cheap and enough, then samples can be divided into two groups: one for modeling and the other for testing. However, this is not possible for expensive computer simulations. Leave-one-out cross-validation (LOOCV) is also widely used but the computational cost may increase rapidly when there are a number of support vectors. Finally, upper bounds on the error expectation are available such as radius-margin bound

[Vapnik 1998; Chapelle et al. 2002; Wang et al. 2003; Men and Wang 2006] and Jaakkola-Haussler (J-H) bound [Chapelle et al. 2002].

The radius-margin bound is defined as

$$\text{LOOCV} \le \frac{1}{l}\frac{R^2}{M^2} \tag{3.14}$$

where l is the number of DoE samples, R is the radius of the smallest sphere containing all samples, and M is the margin of SVM defined in Section 3.3.1. Therefore, LOOCV is bounded by the right side term in Eq. (3.14) and this right side is minimized instead of calculating the LOOCV directly. The right side of Eq. (3.14) is approximated by Fisher linear discriminant function in Wang et al. 2003.

The Jaakkola-Haussler bound is defined as

$$\text{LOOCV} \le \sum_{i=1}^{l} \Psi(\alpha_i K(\mathbf{x}_i, \mathbf{x}_i) - 1) \tag{3.15}$$

where α_i are the Lagrange multipliers, Ψ is a step function, and K is the kernel function.

Using Iowa 2-D example, both error measures are estimated and compared to the true classification error. One hundred DoE samples and one million uniform test points are used. According to Figs. 3.3, 3.4, and 3.5, the shape of the J-H bound is similar to that of true classification error compared to the radius-margin bound. However, the location of the optimum still does not match exactly.

Deterministic computer simulations are our main target problem to deal with, and thus the limit state function with zero training error always exists. The training error is defined as the classification error with respect to the DoE and virtual samples, and thus zero training error means that there is no wrong identifications for the DoE and virtual samples. Therefore, the largest kernel parameter γ with the zero training error means that the corresponding VSVM model is the simplest model, which can describe given DoE

and virtual samples. And, in VSVM, virtual samples are forced to be close to the limit state function, and thus, a smaller parameter value can means an over-fitted model. Therefore, the largest kernel parameter γ with zero training error can be the optimal γ that minimizes the classification error.

Using the Iowa 2-D example, 9-D Rosenbrock function and 12-D Dixon-Price function, performances of the J-H bound and the largest kernel parameter γ with zero training error are compared. Examples are defined in Section 3.6. According to Figs. (3.7), (3.8), and (3.9), the largest kernel parameter γ with zero training error identified the optima correctly compared to the J-H bound. Therefore, the largest kernel parameter γ with zero training error is used as the optimal parameter in VSVM.

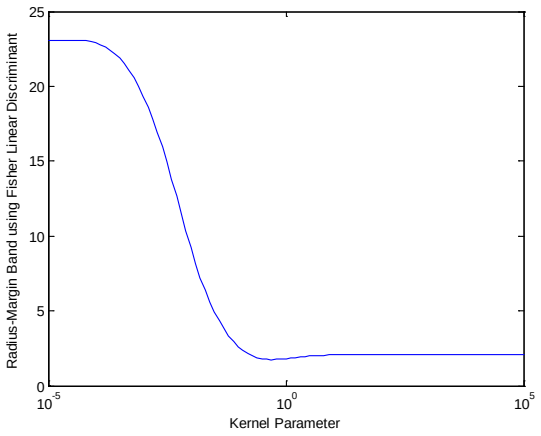

Figure 3.3 Radius-Margin Bound Using Fisher Linear Discriminant Function

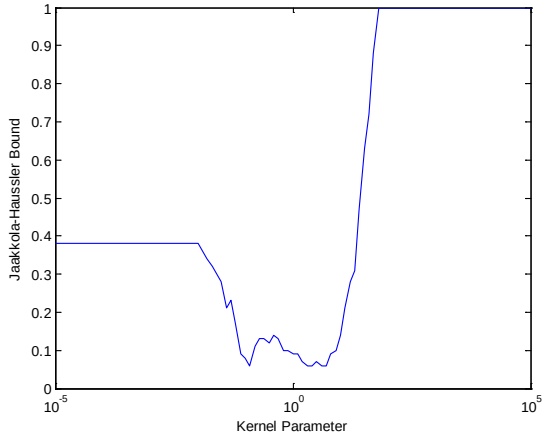

Figure 3.4 Jaakkola-Hausler Bound

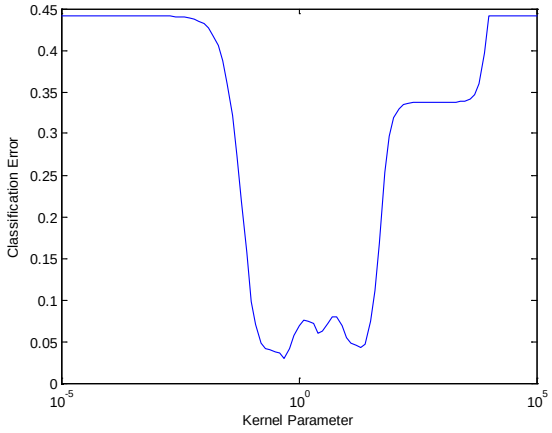

Figure 3.5 True Classification Error

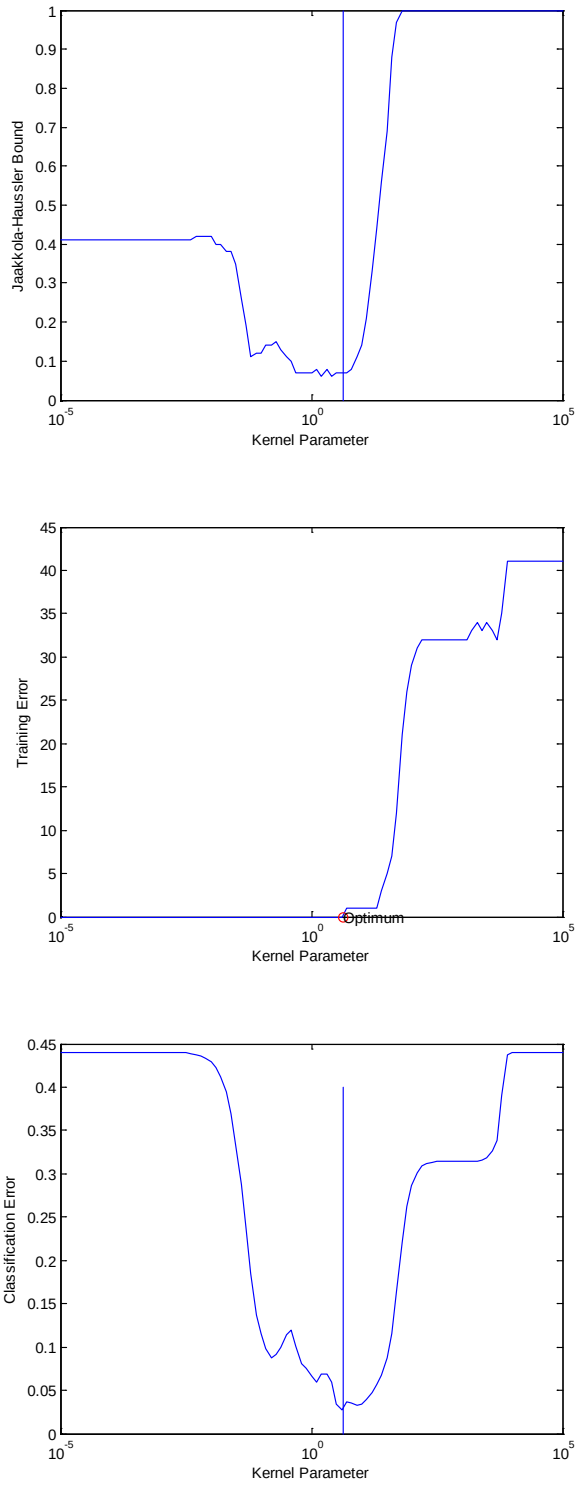

Figure 3.6 The J-H Bound, Training Error and Classification Error for Iowa 2-D Example

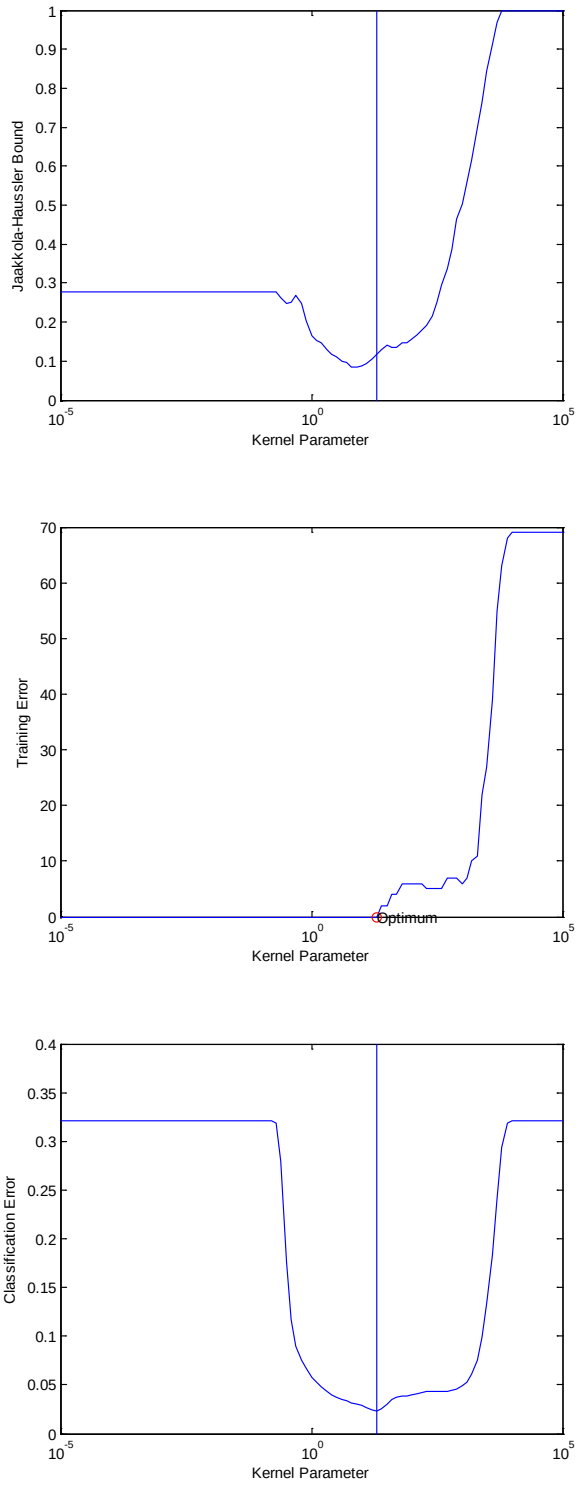

Figure 3.7 The J-H Bound, Training Error and Classification Error for 9-D Rosenbrock Example

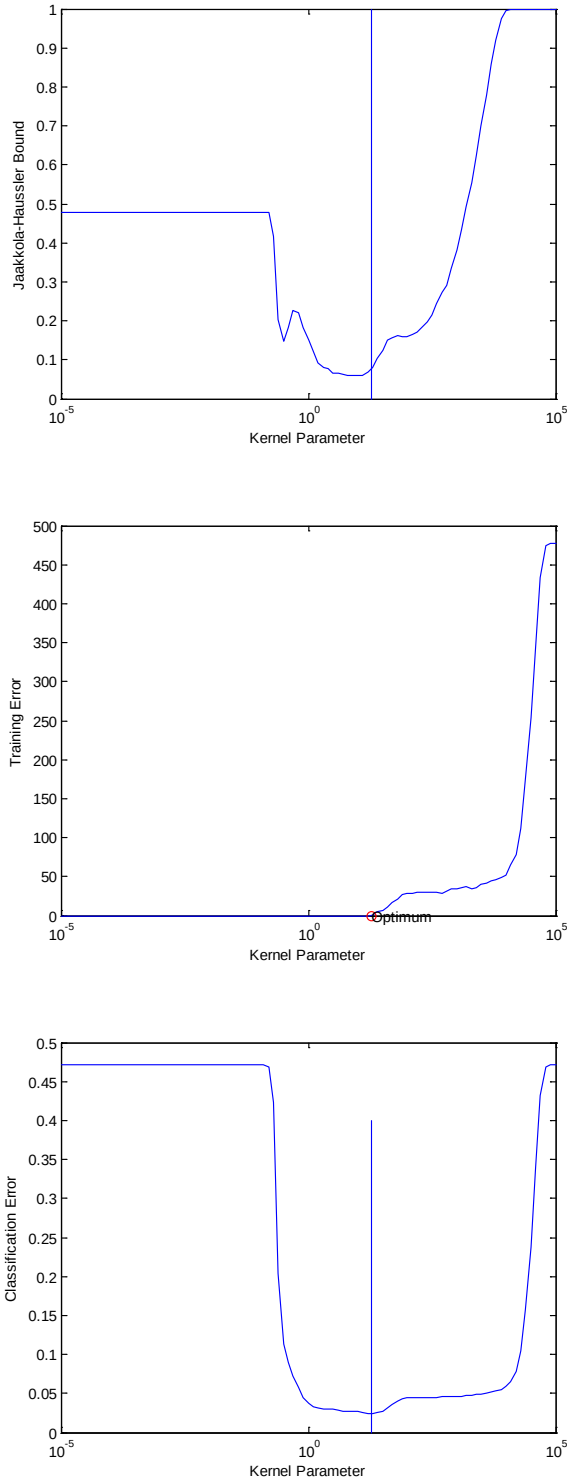

Figure 3.8 The J-H Bound, Training Error and Classification Error for 12-D Dixon-Price Example

Initially, equal interval search (EIS) algorithm [Arora 2004] is applied to find the optimal parameter. However, to improve the efficiency, golden section search (GSS) algorithm [Arora 2004] is used for the kernel parameter estimation process. The main steps of parameter estimation based on GSS are as follows:

Step 1. Define lb, ub and Δ where TE(lb) = 0, TE(ub)>0, and $ub - lb > \Delta > 0$.

TE is the training error.

Step 2. Evaluate TE(a) where $a = lb + 0.382\ (ub - lb)$.

If TE(a) > 0, then go to Step 3. Otherwise, go to step 4.

Step 3. $ub = a$.

If $(ub - lb) < \Delta$, then go to Step 5. Otherwise, go to Step 2.

Step 4. $b = lb + 0.618\ (ub - lb)$. Evaluate TE($b$).

If TE(b) > 0, then $lb = a$, $ub = b$. Otherwise, $lb = b$.

If $(ub - lb) < \Delta$, then go to Step 5. Otherwise, go to Step 2.

Step 5. The optimal kernel parameter $\gamma = lb$.

In Table 3.1, it is shown the golden section search performed better than the equal interval search.

Table 3.1 Performances of EIS and GSS

Problem	Initial Value	Search Method	No. of Training Error Estimations	Optimum Parameter	Time (Sec.)
9-D	19.87	EIS	2	19.87	0.77
		GSS	2	19.87	0.69
	1	EIS	19	19.95	7.95
		GSS	17	19.87	6.59
	100	EIS	17	19.95	6.44
		GSS	13	19.88	5.20

Table 3.1. Continued

12-D	18.50	EIS	2	18.50	0.84
		GSS	2	18.50	0.90
	1	EIS	19	18.62	11.14
		GSS	17	18.45	9.43
	100	EIS	17	18.62	10.47
		GSS	14	18.46	7.62

Note: EIS is the equal interval search algorithm and GSS is the golden section search algorithm.

3.4 Virtual Support Vector Machine (VSVM)

The explicit design space decomposition (EDSD) method, which is based on the conventional SVM, yields good performance for discontinuous limit state functions. However, there exists a limitation for continuous problems. Since EDSD does not use function values, EDSD converges very slowly, and consequently requires many samples in dealing with continuous problems. Therefore, the accuracy needs to be improved for continuous problems, which can be achieved by inserting virtual samples generated based on available function values.

3.4.1 Virtual Sample Generation and VSVM

For the construction of SVM, initial samples, which include both success and failure samples, should be given. Initial samples are generated by Latinized Centroidal Voronoi Tessellation (LCVT), since it shows very good uniformity and randomness [Basudhar et al. 2012; Saka et al. 2007].

Since SVM deals only with classification of responses, i.e., successes (−1) or failures (+1), the SVM decision function is usually located in the middle of opposite signed samples regardless of the function values of the given samples as shown in Figure 3.9. However, in reality, samples with small absolute function values are more likely to be located closer to the limit state function than those with large absolute function values.

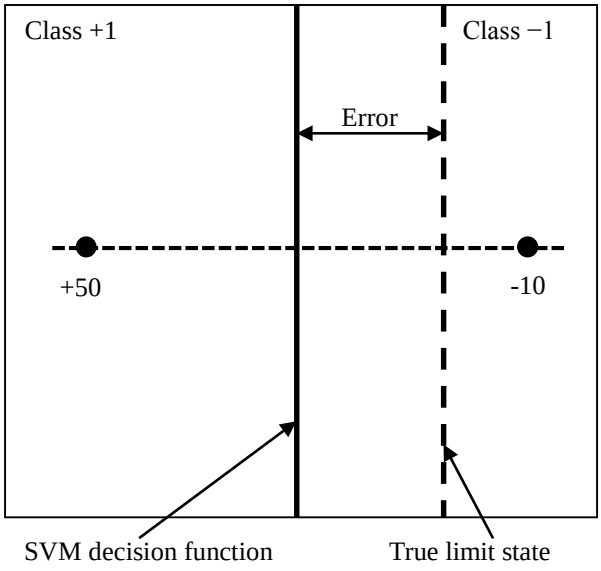

Figure 3.9 SVM Decision Function

In this research, two types of samples are used. The first type is real samples, which include initial samples and sequential samples. Sequential samples are inserted when the VSVM model does not meet the accuracy requirement. These real samples require function evaluations. The second type is virtual samples, which are generated using an approximation method to improve the accuracy of the VSVM decision function. Such virtual samples do not require real function evaluations and only have virtual signs. The basic idea of VSVM is to increase the probability of locating the decision function close to the true limit state function, by inserting two opposite signed virtual samples between the given two samples. These virtual samples play two major roles in VSVM. One is to make the predictions more accurate, and the other is to locate new sequential samples near the limit state function, which will be presented in Section 3.5. In Figure 3.10, the VSVM decision function is shifted towards the sample with a small absolute function value by inserting two virtual samples. The virtual samples with opposite signs

should be near the limit state function and be equally distanced from the limit state function to obtain the best VSVM decision function.

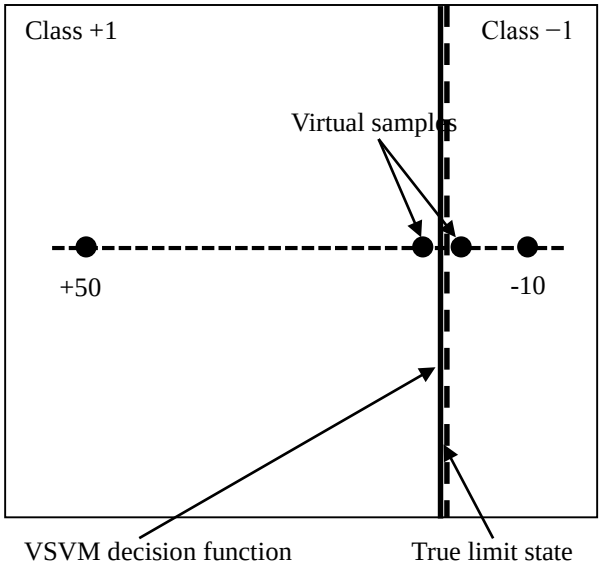

Figure 3.10 VSVM Decision Function with Virtual Samples

3.4.2 Informative Sample Set and Valid Distance

Virtual samples could be generated from the approximation method using any pair of real samples. However, it is desirable to use two opposite class samples. If both samples have the same sign, then finding the decision function requires an extrapolation, which is often inaccurate and the decision function is not located between two given samples. If two existing samples have opposite signs (+1 and −1), then the decision function should exist between the two samples for the continuous problem. Even though any pair of different class samples can be used in theory, if the distance between two

given samples is large or both samples are far from the limit state function, then accurate positioning of the zero point between two samples is difficult. Thus, at least one of two points should be close to the limit state function, and both should be close to each other to make approximations more accurate and useful. Therefore, an informative sample set, from which virtual samples are generated, is defined first. The original SVM is constructed first based on existing samples to identify support vectors. Support vectors are located near the limit state function, and thus they are included in the informative set. It is highly probable that some samples with small absolute values are also located close to the limit state function, even though they may not be support vectors. Therefore, all samples that have absolute response values that are smaller than the maximum absolute responses of the support vectors are chosen as members of the informative set. The informative sample set can be expressed as

$$\begin{aligned} f_{\max} &= \max_i(\left| f(\mathbf{x}_i^*)\right|),\ i = 1, \cdots, N_{\text{SV}} \\ \{\mathbf{x}_j \,|\, &\left| f(\mathbf{x}_j)\right| \le f_{\max},\ j = 1, \cdots, N\} \end{aligned} \tag{3.16}$$

where $\mathbf{x}_i^*$ is the i^{th} support vector, N_{sv} is the number of support vectors, $\mathbf{x}_j$ is the j^{th} sample, N is the number of samples, and f is the function value at the given sample.

From the previously chosen informative samples, the closest opposite signed samples are paired to generate virtual samples between each pair. However, there exist some pairs that can generate important virtual samples, even though they do not belong to the closest pairs. To solve this problem, a valid distance concept is introduced. Pairs can generate virtual samples if the distance between them is shorter than the valid distance. If the valid distance is too large, then there is a risk of including many unnecessary virtual samples and producing poor approximations. If the valid distance is too short, it may not include information that is more useful. To introduce more informative virtual samples while maintaining virtual samples from the closest pairs, the largest distance among all

distances between the closest pairs is defined as the valid distance. Figure 3.11 and Figure 3.12 show the influence of the valid distance concept in a two-dimensional example. By inserting three additional pairs of virtual samples, the accuracy is improved in the area near the new virtual sample pairs.

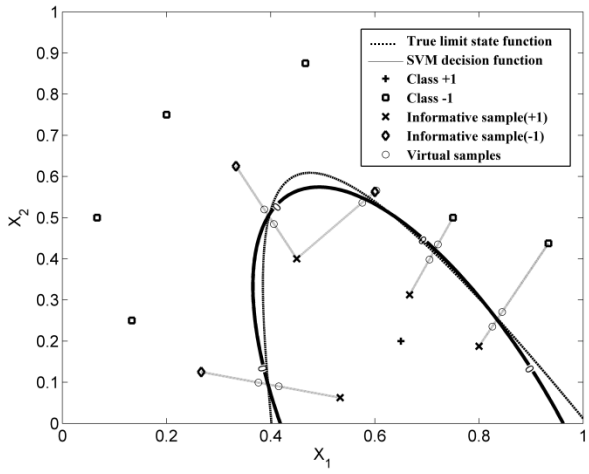

Figure 3.11 VSVM Decision Function without the Valid Distance Concept

3.4.3 Approximations for Zero Positions

Two additional steps are needed for generation of the virtual samples after the informative sample set and the valid distance are defined. Firstly, since the true limit state function is not known in general, a zero position is approximated from two different class samples by using approximation methods such as linear approximation, Kriging, moving lease squares (MLS) method [Kim et al. 2005, 2009; Kang et al. 2010], etc. A zero position means a point where the approximation value is zero among all the points on the line between two opposite signed samples. A linear approximation simply assumes that the function value between two given samples is linear and finds the zero point. The

linear approximation is very efficient and easy to apply but can be inaccurate for highly nonlinear functions.

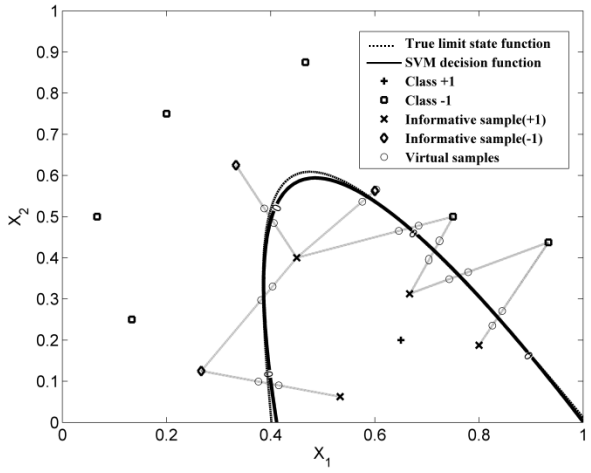

Figure 3.12 VSVM Decision Function with the Valid Distance Concept

The Kriging or MLS methods are accurate near existing samples, so they are appropriate to obtain better approximations, especially for highly nonlinear functions. In this research, the universal Kriging method is used by using all existing samples to approximate zero points between two opposite signed samples, and the SURROGATES toolbox [Viana 2010] is used for the construction of the universal Kriging model. The optimization problem for finding the zero position between two samples is expressed as

$$\begin{aligned} \min_{\mathbf{x}} \quad & \left|\hat{f}(\mathbf{x})\right| \\ \text{s.t.} \quad & \mathbf{x} = \mathbf{x}_i \cdot t + \mathbf{x}_j \cdot (1-t) \\ & 0 \le t \le 1 \end{aligned} \tag{3.17}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are original samples with opposite signs, $\mathbf{x}$ is a point on the straight-line connecting $\mathbf{x}_i$ and $\mathbf{x}_j$, and $\hat{f}(\mathbf{x})$ is an approximated value at $\mathbf{x}$ obtained by the universal Kriging method.

It requires a fair amount of computational time to solve Eq. (3.17) accurately. In particular, Kriging approximations take a large amount of time if approximations are calculated one by one due to its implicit formulation. Therefore, the line connecting two opposite signed samples $\mathbf{x}_i$ and $\mathbf{x}_j$ is divided into multiple elements, their Kriging approximations are evaluated all at once, and the position with the minimum absolute function value is chosen. The size of one element needs to be smaller than the virtual margin, which is the distance between a pair of virtual samples, to generate an accurate surrogate. Thus, the number of elements depends on the virtual margin. Then vector calculation can be carried out all at once rather than one-by-one calculation, which is more efficient in Matlab. In this way, the elapsed time for generating virtual samples is reduced from 39.94 sec. to 2.01 sec. per iteration for a twelve-dimensional problem.

To make the estimation process more efficient, the history of parameter changes was investigated to find that the new optimum correlation parameter $\boldsymbol{\theta}$ is close to the previous optimum $\boldsymbol{\theta}$ in general. If the current VSVM model is similar to the previous VSVM, then both optimum Kriging parameters are also close to each other. Therefore, the previous optimum Kriging correlation parameter $\boldsymbol{\theta}$ value is used as the initial value for the GPS method [Zhao et al. 2011]. By implementing this efficiency strategy, the elapsed time to find the optimum $\boldsymbol{\theta}$ is reduced by 90% per iteration on average.

3.4.4 Generation of Virtual Samples from Zero Positions

Secondly, two opposite signed virtual samples are generated near the zero point. One is located in the direction of the success sample, and the other is in the direction of the failure sample. Both virtual samples should be between the given two opposite signed samples and on the line that connects these points, as shown in Figure 3.10. Then, a new

SVM decision function based on the original and virtual samples will be located between the virtual sample pairs, because the virtual samples in each pair have different signs and are close to each other. If approximations for zero points are accurate, then both virtual samples and a new decision function will be near the limit state function.

One important question is how closely a pair of virtual samples should be located. If the distance between a pair of virtual samples is too long, then these virtual samples will not be chosen as support vectors and they become ineffective. To make the virtual samples effective, the distance should be short enough that the virtual samples are selected as support vectors so that the supporting hyperplanes are located near the decision function. If a smaller target error is required, the virtual margin needs to be reduced accordingly.

If many virtual samples are clustered together within a small region, the additional information from the most closely located virtual samples is negligible, and the computational time increases. In selection of virtual samples, the trade-off between the amount of additional information and the computational cost should be considered. After the valid distance is defined based on SVM with an initial sample set, virtual sample candidates are generated from two opposite samples within the valid distance. The first pair of virtual samples is the pair between a sample with the smallest absolute function value and its closest opposite signed sample, since they provide the most accurate information. Next pair is the candidate which has the longest distance from both real and previously selected virtual samples to prevent clustered virtual samples within a small region. Virtual samples are selected and added until new virtual samples are close to previously chosen virtual samples. In Figure 3.13, suppose that pair 2 is generated from the smallest absolute value. Then, pair 2 is the first pair of virtual samples. Pair 4 has the longest distance from pair 2 and thus pair 4 is selected as virtual samples next. By applying this method, virtual samples are spread out uniformly in the design space. Once

all virtual samples are selected, the VSVM decision function is constructed by using both existing and virtual samples.

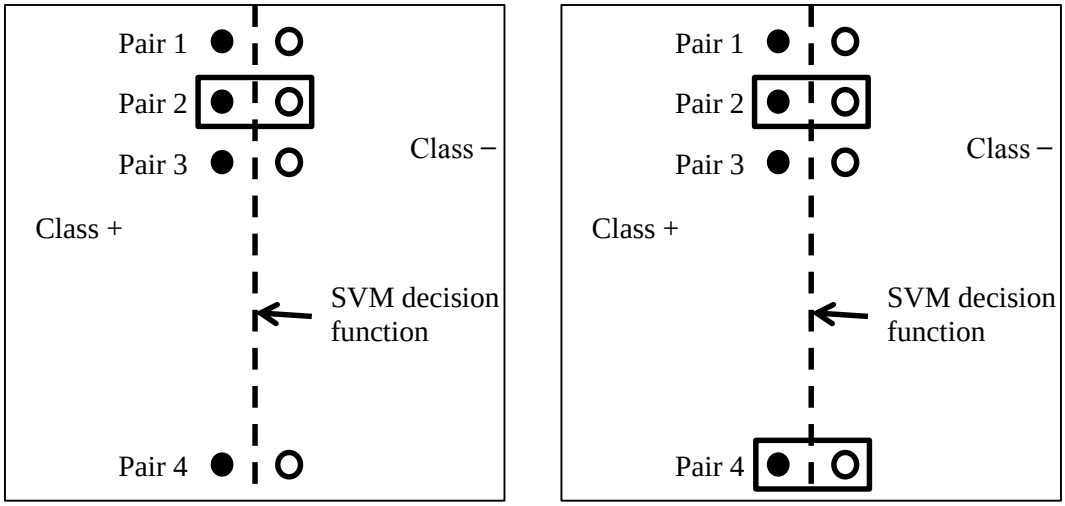

Figure 3.13 Selection of Virtual Samples – Pairs within Solid Squares Are Selected

As explained in Section 3.1, true function values at location of “zero point” and virtual samples are not evaluated. Their signs are decided by approximations. Even though signs and locations of virtual samples may not be accurate with initial samples, the accuracy is improved as new samples are inserted sequentially.

3.5 Adaptive Strategy for Sampling and Stopping Criteria

3.5.1 Adaptive Sequential Sampling

The surrogate-based methods construct a model that is accurate over the given domain, and thus samples tend to spread out on the given domain to satisfy the target accuracy. However, since only an accurate decision function is required for the sampling-

based reliability analysis and RBDO [Lee et al. 2011], samples near the limit state function are more informative than samples far away from the limit state function. Such efficiency cannot be achieved by using a uniform sampling strategy, and thus a sequential sampling method is proposed for better efficiency and accuracy.

In this research, new samples are selected such that they are located within the margin ($|s(\mathbf{x})|<1$), which is narrow since each pair of virtual samples are closely located near the decision function. In addition, new samples should have the maximum distance from the closest existing sample to maximize additional information from new samples. This strategy is similar to the sequential sampling method developed by Basudhar and Missoum, but the computational burden can be reduced by using the within-the-margin (i.e., inequality) constraint ($|s(\mathbf{x})|<1$) rather than the on-the-decision-function (i.e., equality) constraint ($s(\mathbf{x})=0$), which is more difficult to satisfy. On the other hand, this new constraint is effective since the virtual margin is very narrow compared with conventional SVM margins. A less strict constraint can be used with VSVM since new samples do not need to be on the limit state function by introducing virtual samples. In other words, if new samples are located near the limit state function, accurate virtual samples close to the limit state function can be obtained. The optimization problem is defined as

$$\begin{aligned} &\max_{\mathbf{x}} \quad \left\| \mathbf{x} - \mathbf{x}_{\text{nearest}} \right\| \\ &s.t. \quad \left| s(\mathbf{x}) \right| < 1 \end{aligned} \tag{3.18}$$

where $\mathbf{x}_{\text{nearest}}$ is the existing sample closest to the new sample $\mathbf{x}$. Since $\mathbf{x}_{\text{nearest}}$ changes as the position of new sample candidate $\mathbf{x}$ moves, Eq. (3.18) is a moving target problem. In Figure 3.14, a new sample is positioned by solving Eq. (3.18) and inserted into a region near the limit state function and where there is no existing sample nearby. In Figure 3.15, the VSVM decision function is improved significantly near the newly inserted sample.

Any efficient optimization method can be applied to solve Eq. (3.18). The gradient-based optimization methods such as trust-region-reflective algorithm [Coleman and Li 1994, 1996], active-set algorithm [Powell 1978a, b] or interior-point algorithm [Byrd et al. 2000; Waltz et al. 2006] can be used instead of the GPS method since they are faster than the GPS method [Lewis and Torczon 1999] without sacrificing much accuracy. In the research, active-set algorithm is used and initial points are selected among zero points in sparse region.

In the beginning, locations of virtual samples may not be accurate, because their signs are decided based on approximations. However, the accuracy is improved as new samples are inserted sequentially near the decision function.

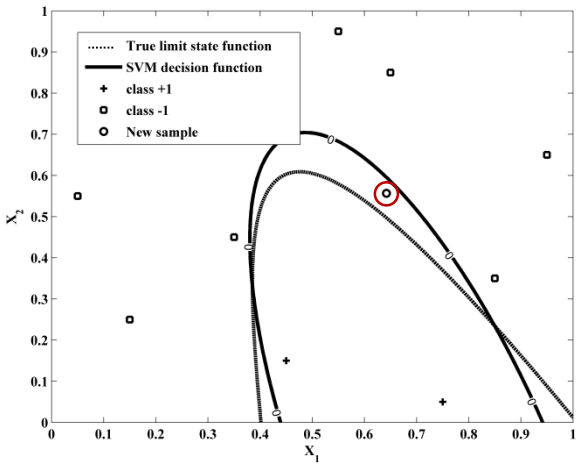

Figure 3.14 VSVM Decision Function and a Sequential Sample

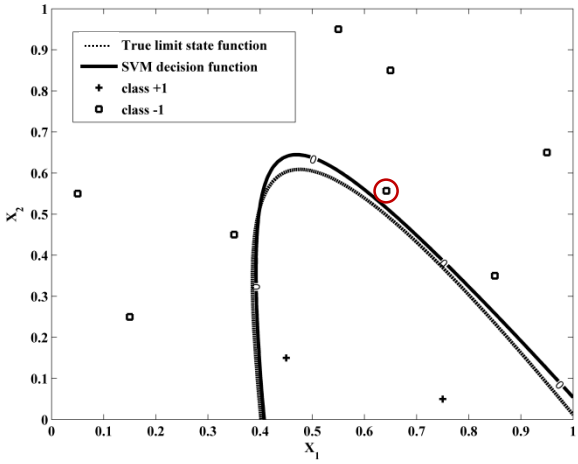

Figure 3.15 VSVM Decision Function with a New Sample

3.5.2 Stopping Criteria

Stopping criteria are required to determine when the decision function is converged. Since the true limit state function is not known, the criterion is based on the variations of the approximated decision function. A set of testing points is generated using input distributions because the MCS points are also generated in the same way for the sampling-based reliability analysis. In this research, ten thousand testing points for stopping criteria (N_{stop}) were used for all examples. The fraction of testing points that show different signs from the previous surrogate is calculated as [Basudhar and Missoum 2008]

$$\Delta_k = \frac{\sum_{i=1}^{N_{\text{stop}}} I_k(\mathbf{x}_i)}{N_{\text{stop}}} \times 100(\%) \tag{3.19}$$

where k is the current iteration number in the sequential sampling process, and Δ_k is the fraction of testing points for which the sign of the VSVM evaluation changes between $k-1^{th}$ and k^{th} iterations. $I_k(\mathbf{x}_i)$ in Eq. (3.19) is an indicator function defined as

$$I_k(\mathbf{x}_i) \equiv \begin{cases} 0, & sign(s_{k-1}(\mathbf{x}_i)) = sign(s_k(\mathbf{x}_i)) \\ 1, & \text{otherwise} \end{cases} \tag{3.20}$$

where $s_{k-1}(\mathbf{x}_i)$ and $s_k(\mathbf{x}_i)$ represent the VSVM value at x_i at the $k-1^{th}$ and k^{th} iterations, respectively. Changes in the VSVM decision function fluctuate and usually decrease as the number of iterations increases, as is shown in Figure 3.16.

In order to implement more stable stopping criteria, the fraction of testing points changing signs between successive iterations is fitted by an exponential curve as [Basudhar and Missoum 2008]

$$\hat{\Delta}_k = Ae^{Bk} \tag{3.21}$$

where $\hat{\Delta}_k$ represents the fitted values of Δ_k, and A and B are the parameters of the exponential curve. The value of $\hat{\Delta}_k$ and the slope of the curve are calculated when new samples are added. If Δ_k is large while $\hat{\Delta}_k$ is small, it means that a large change occurred in the model at k^{th} iteration, which $\hat{\Delta}_k$ did not catch. If Δ_k is small while $\hat{\Delta}_k$ is large, the situation is that the new sample is inserted into a region where zero-position approximations are already accurate, so there is a small change between recent two models but it may not be converged yet. Therefore, both Δ_k and $\hat{\Delta}_k$ should be small for more robust results. In addition, the slope of the curve needs to be small for stable convergence.

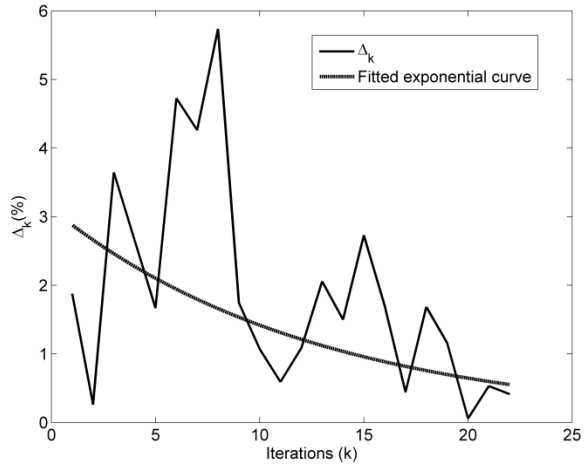

Figure 3.16 Changes of Δ_k and Fitted Exponential Curve

To stop the updating process, the maximum of Δ_k and $\hat{\Delta}_k$ should be less than a small positive number ε_1. Simultaneously, the absolute value of the slope of the curve at convergence should be lower than ε_2. Thus, the stopping criteria can be defined as

$$\begin{aligned} &\max(\Delta_k, \hat{\Delta}_k) < \varepsilon_1 \\ &-\varepsilon_2 < BAe^{Bk} < 0. \end{aligned} \tag{3.22}$$

where ε_1 and ε_2 are determined so that the target classification error level can be achieved. The target classification error is 2.0% in this research. For more accurate limit state function, smaller values can be used. Generally, ε_2 should be smaller than ε_1 for stable convergence. When parallel computing is available and thus multiple sequential samples can be used, changes in the VSVM decision function do not fluctuate much. Then, the slope of the curve in Eq. (3.22) may not be necessary anymore.

The overall procedure of the proposed VSVM method with the sequential sampling strategy is shown in Figure 3.17. Initial DoE samples are generated by LHS or

LCVT and their responses and classification information are evaluated. SVM is constructed and support vectors, informative sample set, and valid distance are estimated. VSVM is constructed using virtual samples. If VSVM is accurate, the process stops. Otherwise, new sequential samples are inserted near the limit state function and repeat the modeling process from function evaluations.

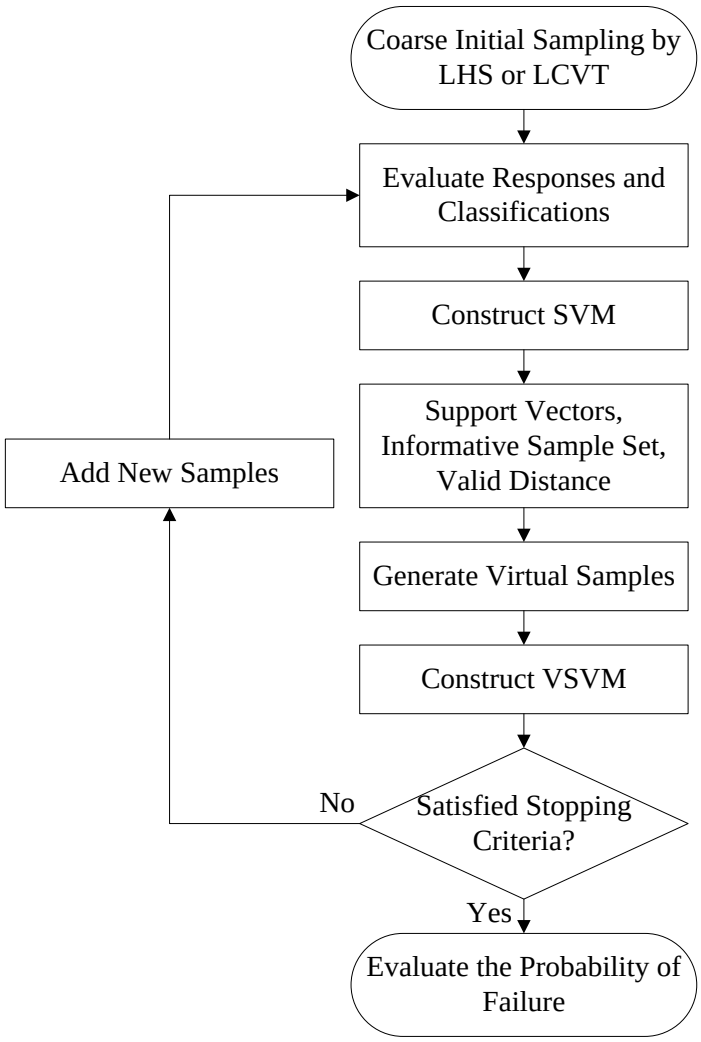

Figure 3.17 Flowchart of VSVM with Sequential Sampling Strategy

3.6 Comparison Study between VSVM and Other Surrogates

3.6.1 Comparison Procedure

Recently developed classification and surrogate modeling methods with sequential sampling schemes are selected for comparison with the proposed VSVM method. The classification method is the explicit design space decomposition (EDSD) method with an improved adaptive sampling scheme that is based on the conventional SVM [Basudhar et al. 2012; Basudhar and Missoum 2008, 2010]. The adaptive sampling method of EDSD has two kinds of methods to select new samples. One method is to select new samples which maximize the distance to the closest existing samples while lying on the SVM decision function. The other method is to increase the convergence and selects new samples in a region where data from one class is sparse in the vicinity of the boundary. For a fair comparison for both EDSD and VSVM, the same Gaussian kernel parameter γ in Eq. (3.13) is used.

The surrogate modeling method is the dynamic Kriging (DKG) method with a sequential sampling method [Zhao et al. 2011]. Zhao et al. showed that, when the same number of samples is used, DKG performs better compared with the polynomial response surface, radial basis function, support vector regression, and universal Kriging method. Therefore, DKG is chosen to compare the accuracy of VSVM. The sequential sampling method chooses new samples where the prediction variance is largest.

Three examples are used to test the performance of the adaptive sampling-based VSVM. One example is a low-dimensional problem, and the other two are high-dimensional problems. All three methods can be applied to both global and local windows. However, a global window usually requires unnecessarily many samples to achieve the target accuracy in RBDO compared with local windows. Therefore, all three

methods are applied to local windows of the original input domain. Since classification methods can be used when the limit state function exists within the local window, original functions are shifted appropriately to include both signed samples. In Sections 3.6.2, 3.6.3 and 3.6.4, the domains of interest are hyper-cube local windows, which are defined by the lower and upper bounds.

For the Gaussian kernel in Eq. (3.13), parameter γ should be provided. Selection of optimum γ is an ongoing research subject. In this research, a fixed γ value, which is small enough to maintain zero training error, is used. The training error is defined as the classification error with respect to the existing and virtual samples and not testing samples.

Since SVM is a classification method and only takes care of the decision function, the mean squared error (MSE) or R^2, which are widely used for the surrogate-based methods, is not appropriate for comparison. Therefore, the accuracy of the SVM decision function should be judged by its closeness to the true limit state function. In practical applications, the true limit state function is not known, and so is the error measure. However, the error measure can be obtained for academic analytical test problems. One million testing points for error measure (N_{test}) are generated based on input distributions because the MCS samples are also generated in the same way for the sampling-based reliability analysis. These testing points are used to calculate the classification error, which is the fraction of misclassified testing points over total number of testing points. A test point for which the sign of VSVM does not match the sign provided by the true limit state function is considered as misclassification [Basudhar and Missoum 2008]. Therefore, the classification error c is

$$c = \frac{\sum_{i=1}^{N_{test}} I(\mathbf{x}_i)}{N_{test}} \times 100(\%) \tag{3.23}$$

where $I(\mathbf{x}_i)$ is the indicator function defined as

$$I(\mathbf{x}_i) \equiv \begin{cases} 1, & s(\mathbf{x}_i) \cdot yc_i < 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.24}$$

where yc_i represents the corresponding classification value (±1) at $\mathbf{x}_i$, and $s(\mathbf{x}_i)$ is the VSVM approximation at $\mathbf{x}_i$.

Our purpose is to evaluate the probability of failure accurately and efficiently. The relationship between the probability of failure measurement error and the classification error is approximately proportional. Therefore, accurate probability of failure can be obtained by keeping the classification error small. In addition, the classification error represents the accuracy of the obtained limit state function, so the classification error is used as the error measure for comparison in this research.

3.6.2 Iowa 2-D Example

The analytic function is the second constraint of Iowa 2-D example function [Tu et al. 1999; Youn et al. 2005], which is expressed as

$$\begin{aligned} f(\mathbf{x}) = 1 + (0.9063 \cdot x_1 + 0.4226 \cdot x_2)^2 + (0.9063 \cdot x_1 \\ +0.4226 \cdot x_2 - 6)^3 - 0.6(0.9063 \cdot x_1 + 0.4226 \cdot x_2)^4 \\ -(-0.4226 \cdot x_1 + 0.9063 \cdot x_2) \\ 4.5 \le x_1 \le 6.5,\ 5.5 \le x_2 \le 7.5 \end{aligned} \tag{3.25}$$

Since the performance is influenced by sample positions, 20 different test cases are used. The number of initial samples, N_i, is 10, and parameters γ, ε_1, and ε_2 are 3, 0.8, and 0.3, respectively, for both EDSD and VSVM. To compare these methods, VSVM is performed first, and DKG and EDSD are applied later using the same number of additional samples, N_a, as VSVM. Each process is forced to stop when it reaches the same number of final samples. Since each method has its own sequential sampling

strategy, sample profiles of the final results are different except the 10 initial samples. According to Table 3.2, which shows averaged values of 20 test cases, EDSD is the fastest, but the classification result is not accurate. This clearly shows that EDSD converges slowly since it does not use the response function values. VSVM is the most accurate and requires a similar amount of time as DKG (33.1 sec. vs. 35.3 sec.) for modeling. However, VSVM is about 30 times faster than DKG (1.1 sec. vs. 32.5 sec.) in estimating response values at one million MCS points due to its simpler formulation. Better classification error (2.57% vs. 0.34%) is due to different sampling strategies.

Table 3.2 Average Results for 2-D Example (N_i=10, N_a=5.6, 20 Test Cases)

		DKG	EDSD	VSVM
Classification error (%)		2.57	15.34	0.34
Elapsed time (sec)	Modeling	35.3	3.2	33.1
	MCS	32.5	0.6	1.1

3.6.3 9-D Rosenbrock Example

The nine-dimensional extended Rosenbrock function [Dixon and Szego 1978; Viana et al. 2009] is used for the test, which is expressed as

$$f(\mathbf{x}) = \sum_{i=1}^{8}\left[(1-x_i)^2 + 100(x_{i+1}-x_i^2)^2\right] - 68000$$
$$-3 \le x_i \le -2, i = 1,\ldots,9. \tag{3.26}$$

The initial sample size is 20; and 20 different test cases are used. For both EDSD and VSVM, γ, ε_1, and ε_2 are 5, 0.5, and 0.03, respectively. Twenty-seven additional samples are used for all surrogate methods. In Table 3.3, EDSD requires less time than other

methods, but it is not accurate. The VSVM uses about half amount of time as DKG (103 sec. vs. 196 sec.) for modeling and results in better classification error. While VSVM is about twice faster than DKG in estimating response values for MCS at one million sample points, VSVM is about 15 times more efficient than DKG (3.4 sec. vs. 49.6 sec.).

Table 3.3 Average Results for 9-D Example (N_i=20, N_a=27, 20 Test Cases)

		DKG	EDSD	VSVM
Classification error (%)		2.31	6.79	1.78
Elapsed time (sec)	Modeling	196	60	103
	MCS	49.6	1.8	3.4

3.6.4 12-D Dixon-Price Example

For a twelve-dimensional example, the Dixon-Price function [Dixon and Szego 1978; Viana et al. 2009] is used and its mathematical expression is

$$f(\mathbf{x}) = (x_1 - 1)^2 + \sum_{i=2}^{12} i(2x_i^2 - x_{i-1})^2 - 36000$$
$$3 \le x_i \le 4, i = 1, \ldots, 12. \tag{3.27}$$

The initial sample size is 35 for 20 different test cases. Parameters γ, ε_1, and ε_2 are 15, 0.25, and 0.015, respectively. Thirty-three additional samples are used for all three methods. In Table 3.4, EDSD is very efficient but does not provide accurate results. The VSVM uses less time than DKG for both modeling and estimating response values for MCS and results in a better classification error.

Table 3.4 Average Results for 12-D Example (N_i=35, N_a=33.3, 20 Test Cases)

		DKG	EDSD	VSVM
Classification error (%)		2.02	8.88	1.67
Elapsed time (sec)	Modeling	289	64	169
	MCS	64.3	1.9	4.6

For another way of comparison, EDSD is performed using the same stopping criteria as VSVM so that EDSD can use more samples to construct the decision function. According to Table 3.5, the average number of additional samples of EDSD is 77.9, which is far more than 33.3 of VSVM. The EDSD also uses a similar amount of time as VSVM, and the classification error is still quite large. Clearly, VSVM is more accurate than EDSD.

Since DKG and VSVM use different stopping criteria, a smaller stopping criterion is used for DKG to achieve a classification error similar to that of VSVM. In Table 3.6, DKG can achieve a classification error level similar to that of VSVM after it uses about six more samples. Furthermore, the elapsed time of DKG is larger than that of VSVM for both modeling (341 sec. vs. 169 sec.) and estimating response values for MCS (65.1 sec. vs. 4.6 sec.).

Table 3.5 Average Results of EDSD and VSVM for the Same Stopping Criteria (ε_1 =0.25, ε_2 =0.015, 20 Test Cases)

		EDSD	VSVM
No. of additional samples		77.9	33.3
Classification error (%)		6.9	1.67
Elapsed time (sec)	Modeling	149	169
	MCS	1.9	4.6

In Figure 3.18, even though classification errors with initial samples are not satisfactory, they are decreasing as new samples are inserted. Compared with EDSD, the classification error of VSVM is significantly reduced. Both VSVM and DKG are accurate overall and their convergences are also stable. If new samples are inserted into the region which is already accurate, then new surrogate model will be almost identical to previous surrogate model. This is the reason some flat regions exist for all three methods as shown in Figure 3.18.

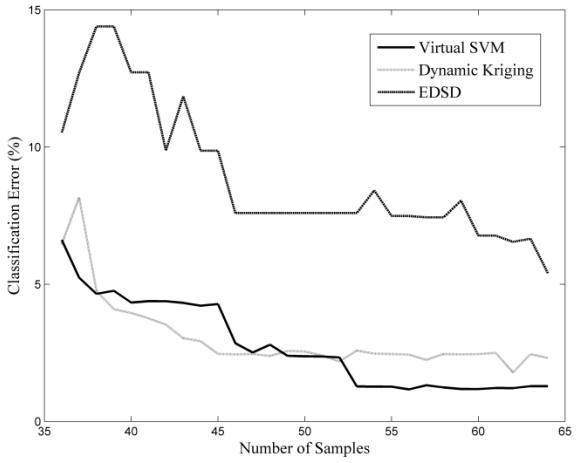

Figure 3.18 Classification Error Changes as VSVM Converges

The VSVM is more efficient than DKG in terms of the elapsed time for both surrogate modeling and estimating response values for MCS, while maintaining a better accuracy level. The EDSD converges very slowly and is inefficient in terms of the number of additional samples. This will be more problematic when the computer simulations at each sample point are expensive for practical application problems.

Table 3.6 Average Results of DKG and VSVM for Similar Classification Errors (20 Test Cases)

		DKG	VSVM
No. of additional samples		39.4	33.3
Classification error (%)		1.74	1.67
Elapsed time (sec)	Modeling	341	169
	MCS	65.1	4.6

3.6.5 Different Initial DoE Sample Sizes

Previously, one fixed initial sample size is applied for three different problems. Now, different initial sample sizes are applied. Constraint boundary sampling (CBS) method in Section 4.3.4 and convergence classification error (CCE) in Section 4.4 are applied with VSVM. In Table 3.7 and 3.8, more DoE samples are used for larger initial DoE samples, but they are not significantly different. Final classification error levels are also consistent for different dimensional problems. On the contrary, DKG with MSE-based sampling in Section 4.3.1 often prematurely converged with the same convergence criterion. Therefore, any reasonable initial DoE sample size is fine for VSVM. And, CBS with CCE is recommended for more stable performance.

Table 3.7 Average Results of DKG and VSVM for Different Initial DoE Samples (Rosenbrock, 20 Test Cases)

Problem	Method	Initial Samples	Final Samples	Classification Error	Response Evaluation Time
Rosenbrock	VSVM	20	169	6.47%	29.0
	DKG	20	29	14.19%	68.4
	VSVM	30	197	6.82%	29.8
	DKG	30	39	14.16%	69.3
	VSVM	40	216	7.43%	31.6
	DKG	40	49	14.07%	71.4

Table 3.8 Average Results of DKG and VSVM for Different Initial DoE Samples (Dixon-Price, 20 Test Cases)

Problem	Method	Initial Samples	Final Samples	Classification Error	Response Evaluation Time
Dixon-Price	VSVM	20	278	2.55%	37.8
	DKG	20	120	13.12%	381.3
	VSVM	35	275	2.39%	40.2
	DKG	35	142	6.09%	488.8
	VSVM	50	306	2.29%	40.8
	DKG	50	117	12.29%	386.0

3.7 Conclusion

A sequential sampling-based virtual support vector machine (VSVM) method is developed to efficiently construct the accurate decision function for the reliability analysis. Virtual samples are generated from real samples to improve the accuracy of the SVM decision function, and a sequential sampling method is developed to increase the efficiency and accuracy of VSVM by inserting new samples near the true limit state function.

The proposed method is compared with a classification method EDSD and a surrogate modeling method DKG with their own sequential sampling strategies. DKG can construct accurate surrogates with a relatively small number of samples, but it is inefficient for MCS since it has an implicit expression for response evaluations, and the dynamic basis selection process requires significant computational effort. For a low-dimensional problem, both VSVM and DKG are accurate and require similar modeling time. However, VSVM becomes more efficient than DKG or EDSD while achieving excellent accuracy for high-dimensional problems. VSVM is much more efficient than DKG in evaluating response values for MCS, and thus VSVM is preferred for sampling-based RBDO. In this comparison study, better classification error of VSVM compared with DKG is due to the fact that VSVM used the new sequential sampling method near

the constraint boundary. Therefore, sampling near the constraint is more efficient than sampling on the whole domain; and it is desirable to implement a constraint boundary sampling method [Bichon et al. 2008, 2011; Ranjan et al. 2008; Lee and Jung 2008; Bect et al. 2012; Viana et al. 2012] for DKG for sampling-based RBDO.

The proposed method is developed and applied to sampling-based RBDO using local windows. Therefore, whenever the design is changed, active/violated constraints are identified and DKG and VSVM are applied only for the active/violated constraints [Lee et al. 2011; Youn et al. 2005; Zhao et al. 2011]. Thus, both DKG and VSVM will use local windows. The RBDO problem usually has multiple constraints and the current VSVM method requires building VSVM model for each constraint. It would be more efficient if we can construct one VSVM model for multiple constraints. Another fact to point out is that response surface methods such as Kriging have advantage that they can describe not only the limit state but also the overall design space. However, only the limit stat information is required for the sampling-based RBDO so the classification method can be very effectively used.

CHAPTER 4

DESIGN OF EXPERIMENT

4.1 Introduction

The design of experiment (DoE) is the sampling plan in the design space and is of great importance for surrogates and VSVM. Many researchers have developed a number of sampling techniques for different surrogates and purposes. These sampling methods can be roughly divided into two groups: one-stage sampling and sequential sampling methods. Comprehensive reviews of different DoE's are available in the literatures [Simpson et al. 2002; Queipo et al. 2005; Wang and Shan 2007].

In one-stage sampling approaches, all DoE samples are generated at once to maximize the target criterion. Typical one-stage sampling approaches include orthogonal array design (OA) [Owen 1992; Koehler and Owen 1996; Hedayat 1999; Simpson et al. 2001a, b; Giunta et al. 2003; Queipo et al 2005; Fang et al. 2010], maximum entropy design [Shewry and Wynn 1987; Currin et al. 1991; Koehler and Owen 1996], Latin hypercube sampling (LHS) [McKay et al. 1979; Sacks et al. 1989b; Park 1994; Koehler and Owen 1996; Huntington and Lyrintzis 1998; Butler 2001; Simpson et al. 2001a; Giunta et al. 2003; Leary et al. 2003; Queipo et al 2005; Fang et al. 2010], Hammersley sequence sampling (HSS) [Kalagnanam and Diwekar 1997; Wong et al. 1997; Simpson et al. 2001a; Giunta et al. 2003], D-optimal design [Chaloner and Verdinelli 1995; Butler 2001; Goel et al. 2008] and Latinized Centroidal Voronoi Tessellation (LCVT) [Burkardt et al. 2002; Saka et al. 2006].

One-stage sampling methods focus on initial sampling in order to achieve certain space filling properties [Wang and Shan 2007]. On the other hand, sequential and adaptive sampling methods insert new samples into the design space to achieve the target accuracy of surrogates. They have become popular in recent years mainly because the appropriate sampling size is difficult to know a priori. In sequential and adaptive

sampling methods, locations of new samples are decided based on existing samples, current surrogates and the infill criteria. There are many different infill criteria available for the surrogate-based design optimization. Typical sequential and adaptive sampling approaches are error based exploration [Sacks et al. 1989a, b; Forrester et al. 2008], prediction based exploration [Forrester et al. 2008], statistical lower bound [Jones 2001; Lin et al. 2004; Forrester et al. 2008], probability of improvement [Jones 2001; Forrester et al. 2008], expected improvement [McKay et al. 1979; Jones et al. 1998; Jones 2001; Forrester et al. 2008] and conditional likelihood approaches [Jones 2001; Forrester et al. 2008].

4.2 Initial Sampling Methods

4.2.1 Uniform Sampling Methods in Hyper-Cube

Without any prior information, initial samples need to be distributed uniformly within the design space. Any one-stage sampling methods listed in Section 4.1 can be used for generating initial samples. Among many one-stage sampling methods, LHS and its variants are most popular, mainly because they do not require more samples for more dimensions and samples are non-collapsing. In this study, however, LCVT is usually applied for generating initial samples in hyper-cubic design spaces, since LCVT tends to provide more uniformly distributed samples within the design space [Romero et al. 2006; Saka et al. 2007; Basudhar and Missoum 2008]. Even though Romero et al. showed uniformity of HSS is better than LCVT or LHS for two-dimensional design space, the minimum statistical distance between samples is maximized with LCVT for higher dimensional design space as shown in Table 4.1.

.

Table 4.1 Mean of Minimum Distances between 100 Samples

Dimensions	LHS	HSS	LCVT
2	0.018	0.038	0.031
3	0.056	0.089	0.101
5	0.17	0.21	0.26
10	0.48	0.51	0.61
15	0.77	0.51	0.87

4.2.2 Uniform Sampling Method in Hyper-Sphere

For local or global windows, hyper-cubic windows have been widely applied. However, test points in the grey area in Figure 4.1 do not significantly affect the accuracy of the reliability analysis using surrogate models, and the volume of the hyper-cube is much larger than that of the hyper-sphere due to the curse of dimensionality, especially in high-dimensional spaces. Ratio of the volume of a hypersphere to that of a circumscribing hypercube is expressed as [Jiang et al. 2011]

$$r_V = \frac{(2R)^{nd}}{\frac{S_d R^{nd}}{nd}} = \frac{2^{nd}\, nd}{S_d}. \tag{4.1}$$

where R is the radius of the hypersphere, and the coefficient S_d for the number of dimensions nd is:

$$S_d = \begin{cases} \dfrac{2^{\frac{nd+1}{2}} \pi^{\frac{nd-1}{2}}}{(nd-2)!} & d \text{ odd} \\ \dfrac{2\pi^{\frac{nd}{2}}}{(\frac{nd}{2}-1)!} & d \text{ even} \end{cases}. \tag{4.2}$$

For example, according to Table 4.2, the volume of a hypercube is 3,057 times larger than that of the corresponding hyper-sphere for a 12-D problem. Therefore, the hyper-

cubic local window requires more samples to construct accurate limit state functions in the grey area, which is unnecessary. Thus, the number of required samples can be reduced by employing the hyper-spherical local window [Lee et al. 2011].

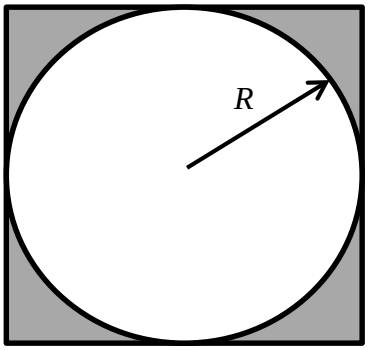

Figure 4.1 Hyper-Sphere and Hyper-Cube

Table 4.2 Volume Ratios of Hyper-Sphere and Hyper-Cube

Dimension	Volume Ratio (Hyper-cube / Hyper-sphere)
2	1.27
4	3.25
6	12.4
8	63.1
10	402
12	3057

To generate initial samples or test points for the calculation of stopping criteria, uniform sampling methods need to be applied. A disadvantage of factorial design and orthogonal arrays is that the user cannot specify an arbitrary number of samples [Owen 1992; Simpson et al. 2001; Giunta et al. 2003; Rai 2006]. Latin hyper-cube sampling

(LHS), Hammersley sequence sampling (HSS) and Latin Centroidal Voronoi Tessellations (LCVT) are available for generating uniform samples in hyper-cubic windows. However, since these sampling methods cannot be directly applied to hyper-spherical windows, the rejection algorithm needs to be used. In the rejection algorithm, uniform samples within a hyper-cubic window are generated first, and then the samples located outside of the hyper-spherical window are rejected. However, the rejection method is very inefficient, especially for high-dimensional problems according to Table 4.2. For example, 3057 times the target number of samples needs to be generated and rejected in a 12-D problem. Jiang et al. proposed a sampling method in a hyper-sphere that satisfies uniformity [Jiang et al. 2011]. In their proposed method, MCS samples are generated initially and the first sample in the DoE is selected as the farthest MCS sample from the mean. Subsequent samples are selected from the remaining MCS samples by maximizing the minimum distance to the already selected DoE samples. However, their local window size depends on the location of random MCS samples and thus, if any MCS sample happens to be far away from the design, the local window can be very large. Furthermore, distances between MCS samples and selected DoE samples should be calculated, and thus their uniform sampling method is inefficient. Therefore, the efficient Transformations/Gibbs sampling method (TGS) that generates uniform samples via latent variables and the Gibbs sampler is introduced [Cumbus et al. 1996]. For example, the Transformations/Gibbs sampling method is 235 times more efficient than the rejection methods for a 10-D problem.

4.3 Sequential Sampling Method

Currently, DoE samples are selected one-by-one in the sequential sampling method in Chapter 3. If parallel computing is available, multiple samples are more efficient than one-by-one sampling due to expensive computational cost for computer simulations. However, it is difficult to select multiple DoE samples using the sequential

sampling strategy in Chapter 3. For multiple samples, Eq. (3.16) needs to be solved in serial, thus the sampling process can become inefficient. Furthermore, since Eq. (3.16) has multiple local optimums, it is easy to fall into local optimums and the distance between new samples cannot be controlled. Therefore, Kriging-based constraint boundary sampling methods are investigated.

4.3.1 Sequential Sampling Method Using Mean Squared Error in the Kriging Prediction

In DKG, an error based sequential sampling method is used [Sacks et al. 1989a, b; Forrester et al. 2008]. An estimated error in the Kriging model is available and so it is possible to position new samples where the predicted error is the largest. The mean squared error (MSE) in the Kriging prediction can be expressed as

$$\hat{s}^2(\mathbf{x}) = \sigma^2\left[1-\mathbf{r}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{r}+(\mathbf{F}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{r}-\mathbf{f})^{\mathrm{T}}(\mathbf{F}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{F})^{-1}(\mathbf{F}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{r}-\mathbf{f})\right]. \tag{4.3}$$

where σ^2 is the process variance, $\mathbf{R}$ is the correlation matrix, $\mathbf{F}$ is a design matrix, $\mathbf{f}(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}),\ldots, f_K(\mathbf{x})\}^{\mathrm{T}}$ and $\mathbf{r}(\mathbf{x}) = \{R(\boldsymbol{\theta},\mathbf{x},\mathbf{x}_1), R(\boldsymbol{\theta},\mathbf{x},\mathbf{x}_2),..., R(\boldsymbol{\theta},\mathbf{x},\mathbf{x}_N)\}^{\mathrm{T}}$. And the infill criterion is to maximize the predicted error $\hat{s}^2(\mathbf{x})$.

4.3.2 Sub-Domain Sampling near Constraints

In Zhao 2011, the strategy is to insert new DoE sample at the weakest point. If multiple samples are selected using the same sampling strategy, new samples can be clustered near the weakest point. Therefore, new sampling strategy needs to be developed for multiple samples.

The basic concept of new sampling strategy is (1) to divide the whole domain into n_2 sub-domains (2) to select n_1 sub-domains which are located near the limit state function (3) to find the weakest point within each selected sub-domain. Therefore, first, n_2 uniform samples, where $n_1 < n_2$, are generated in the local window and all test points

can be divided into n_2 sub-domains using a Voronoi diagram [Voronoi 1908]. Then, n_1 sub-domains out of n_2 domains, for which mean response values at test points are close to 0, are selected. Within each selected subdomain, the weakest point is selected. Consequently, n_1 samples are selected near the limit state function. However, this methodology is not efficient as other constraint boundary sampling methods, since new samples are located near the limit state function instead of being located on the limit state function. Furthermore, n_2 affects the result and should be specified by the user.

4.3.3 Efficient Global Reliability Analysis (EGRA)

Given the constraint g(**x**) < 0, the feasibility at x is defined as

$$F(\mathbf{x}) = \max(\varepsilon(\mathbf{x}) - |G(\mathbf{x})|, 0) \tag{4.4}$$

where **x** is the sample location, $G(\mathbf{x})$ denotes the random variable representing $g(\mathbf{x})$ and measures the uncertainty in $G(\mathbf{x})$. The feasibility $F(\mathbf{x})$ is maximized when the uncertainty is largest and $G(\mathbf{x})$ is close to the limit state function. If $G(\mathbf{x})$ is a Gaussian process, then the expected feasibility function (EFF) is defined as

$$\begin{aligned} \mathrm{EFF}(\mathbf{x}) &= E\left[F(\mathbf{x})\right] \\ &= \int_{-\varepsilon}^{+\varepsilon} [\varepsilon - |G|] f_{\hat{g}} dG \\ &= \hat{g}(\mathbf{x})\left[2\Phi(u) - \Phi(u^+) - \Phi(u^-)\right] \\ &\quad - \sigma_G(\mathbf{x})\left[2\phi(u) - \phi(u^+) - \phi(u^-)\right] \\ &\quad + \varepsilon\left[\Phi(u^+) - \Phi(u^-)\right] \end{aligned} \tag{4.5}$$

where Φ and ϕ are the standard Gaussian CDF and PDF, respectively, $\hat{g}(\mathbf{x})$ is the Kriging prediction, $\sigma_G(\mathbf{x})$ is the square root of the Kriging prediction variance, $\varepsilon = \alpha\sigma_G(\mathbf{x})$, $u = \dfrac{-\hat{g}(\mathbf{x})}{\sigma_G(\mathbf{x})}$, $u^+ = \dfrac{\varepsilon - \hat{g}(\mathbf{x})}{\sigma_G(\mathbf{x})}$, and $u^- = \dfrac{-\varepsilon - \hat{g}(\mathbf{x})}{\sigma_G(\mathbf{x})}$. Bichon et al. 2008 recommended $\varepsilon = 2\sigma_G(\mathbf{x})$.

For multiple samples, a Kriging-believer EGRA is proposed by Viana et al. 2012. In this method, after one new point is obtained, the Kriging model is updated as if one sample is inserted at the selected location. Then, one more sample is selected using updated Kriging model. Therefore, if *n* samples are needed, Kriging models are constructed *n* times sequentially.

On the other hand, multiple sample points can be located on the constraint boundary by multiplying the nearest distance $D(\mathbf{x})$ from existing samples [Lee and Jung 2008]. Then EFF_2 can be expressed as

$$\begin{aligned}\mathrm{EFF}_2(\mathbf{x}) &= E\left[F(\mathbf{x})\right]\cdot D(\mathbf{x}) \\ &= \int_{-\varepsilon}^{+\varepsilon}[\varepsilon-|G|]\,f_{\hat{g}}\,dG\cdot D(\mathbf{x}) \\ &= \hat{g}(\mathbf{x})\left[2\Phi(u)-\Phi(u^+)-\Phi(u^-)\right]\cdot D(\mathbf{x}) \\ &\quad -\sigma_G(\mathbf{x})\left[2\phi(u)-\phi(u^+)-\phi(u^-)\right]\cdot D(\mathbf{x}) \\ &\quad +\varepsilon\left[\Phi(u^+)-\Phi(u^-)\right]\cdot D(\mathbf{x})\end{aligned} \tag{4.6}$$

where distance $D(\mathbf{x})$ is calculated in normalized space. Then, new samples are not selected on previously selected locations. In this method, previous EFF value does not change and only $D(\mathbf{x})$ changes as more samples are inserted. For example, if *n* samples are needed, *n* $D(\mathbf{x})$ need to be calculated sequentially.

These two methods are compared using Iowa 2-D (I-A) function, Branin-Hoo (B-H) function, Camelback (C-B) function, Hartmann3 (H3) function, Hartmann6 (H6) function, Rosenbrock (R-B) function and Dixon-Price (D-P) function. The number of variables, initial samples, sequential samples at once, and final samples are summarized in Table 4.3. According to

Table 4.4, EFF is more expensive compared to EFF_2 especially for high dimensional problems. However, classification errors are similar to each other. Therefore, EFF_2 will be used for EGRA in this research.

Table 4.3 Test Environments

Number	I-A	B-H	C-B	H3	H6	R-B	D-P
Variables	2	2	2	3	6	9	12
Initial Samples	4	4	4	9	150	150	150
Sequential Samples at Once	5	5	5	5	10	10	10
Final Samples	24	19	39	54	350	350	350

Table 4.4 Performances of EFF and EFF_2

	Methods	I-A	B-H	C-B	H3	H6	R-B	D-P
CE	EFF	4.29%	0.49%	1.11%	1.83%	7.33%	11.25%	10.34%
	EFF_2	0.70%	0.86%	0.50%	1.87%	7.64%	11.05%	10.42%
Time (sec.)	EFF	44.8	29.8	143	253	24,054	33,556	25,262
	EFF_2	4.7	3.58	11.3	18.7	99	215	142

Note: CE is the classification error.

4.3.4 Constraint Boundary Sampling (CBS)

If the failure region is defined as $G(\mathbf{x}) > 0$, the feasible probability of a Gaussian process is defined as follows:

$$F = \Phi\left(\frac{-\hat{G}(\mathbf{x})}{\sigma_G(\mathbf{x})}\right) \tag{4.7}$$

where Φ is the standard Gaussian CDF, $\sigma_G(\mathbf{x})$ is the standard deviation at $\mathbf{x}$ and $\hat{G}(\mathbf{x})$ is the Kriging prediction. Then a new sampling criterion by using standard normal PDF ϕ can be proposed as follows:

$$\text{CBS} = \begin{cases} \sum_{i=1}^{nc} \phi\left(\frac{-\hat{G}_i(\mathbf{x})}{\sigma_G}\right) \cdot D(\mathbf{x}) & \text{if } \hat{G}_i(\mathbf{x}) < 0 \ \ \forall i \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

where *nc* is the number of constraints and $D(\mathbf{x})$ is the nearest distance from the existing sample points. The proposed criterion will become large along constraint boundary, since PDF has the largest value at $\frac{-\hat{G}_i(\mathbf{x})}{\sigma_G} = 0$. By including $D(\mathbf{x})$ in the formulation, CBS becomes large when new candidate point is far away from existing samples. EGRA and CBS are conceptually similar and the difference between them is that CBS only chooses samples in the feasible regions.

After intensive tests using different numerical examples, CBS is chosen as the sequential sampling method for both VSVM and DKG.

4.4 Convergence Criterion

In DKG, prediction variance is used as the convergence criterion. However, prediction variance does not guarantee accurate response surface or decision function, since it shows large variation for different problems. If prediction variance is not used, there is no independent accuracy measure for surrogates or classification models. Therefore, convergence MSE (CMSE) is proposed, in which two consecutive surrogates are compared. Even though CMSE is more stable compared to the Kriging prediction variance, it is still affected by the scale of responses and it is not directly related to the probability of failure or its sensitivity. Therefore, it is very difficult to choose the threshold value for the convergence.

Next, convergence probability of failure (CPF) and convergence classification error (CCE) are proposed. CPF compares two consecutive probabilities of failures and it is directly related to the probability of failure but not to its sensitivity. On the contrary, CCE calculates the classification error between two consecutive decision functions, and CCE is directly related to the probability of failure and its sensitivity.

CMSE, CPF and CCE are intensively tested and compared using different mathematical examples. Overall, CPF and CCE perform more efficiently compared to CMSE. Table 4.5 shows RBDO results for Iowa 2D example. Two different types of test points are also compared. One is a MCS-based measure and the other is a hypersphere-based measure. In the MCS-based measure, MCS test points are used to calculate CPF or CCE. On the other hand, uniform test points in the hypersphere are used in the hypersphere-based measure. With MCS test points, more DoE samples are required to achieve similar accuracy in Table 4.5. And, there is no significant difference between CPF and CCE. Since CCE takes care of both the probability of failure and the sensitivity, CCE using hyper-spherical uniform test points will be used for the stopping criterion.

Table 4.5 CPF VS. CCE for Iowa 2D Example (Average over 10 Cases)

Stopping Criterion		X1	X2	Cost	No. Iter.	No. BB	No. FE
CPF(MCS)	1E-3	4.7372	1.5490	-1.9091	7.00	40.20	66.50
	5E-3	4.7357	1.5498	-1.9088	7.60	44.20	71.70
	1E-2	4.7389	1.5482	-1.9095	8.60	49.60	78.20
CCE(MCS)	1E-3	4.7362	1.5498	-1.9088	8.30	48.00	79.00
	5E-3	4.7381	1.5497	-1.9088	8.00	39.80	64.20
	1E-2	4.7365	1.5499	-1.9087	7.60	50.80	80.70
CPF(HS)	1E-3	4.7357	1.5504	-1.9085	7.30	28.70	49.20
	5E-3	4.7358	1.5509	-1.9083	8.40	34.40	56.40
	1E-2	4.7356	1.5502	-1.9086	7.40	32.80	53.40
CCE(HS)	1E-3	4.7359	1.5505	-1.9085	7.10	32.70	54.60
	5E-3	4.7357	1.5505	-1.9085	7.30	37.00	60.20
	1E-2	4.7356	1.5508	-1.9084	7.80	27.80	45.70

Note: MCS means MCS samples are used to calculate the convergence criterion. HS means uniform samples in the hypersphere are used to calculate the convergence criterion. BB means blackbox calls and FE means function evaluations.

CHAPTER 5

SAMPLING-BASED RBDO USING VSVM

5.1 Efficiency Strategies for Practical Use of Sampling-Based RBDO

Five strategies are introduced in this section to carry out the sampling-based RBDO accurately and efficiently, which are launching RBDO at a deterministic optimum design, applying hyper-spherical local window for surrogate model generation, filtering of constraints, reusing samples and generating virtual samples efficiently.

5.1.1 Launching RBDO at a Deterministic Optimum Design

Even though deterministic design optimization (DDO) optimum is not reliable, most probabilistic optima are located near the deterministic ones. Therefore, we can improve numerical efficiency by starting from deterministic optima and thus reducing the number of RBDO iterations [Du and Chen 2004; Youn et al. 2005; Lee et al. 2011]. In this research, if the probabilistic optimum candidate is very close to the deterministic optimum design, the previous VSVM model can be reused to save computational cost. However, probabilistic optimum design may not be very close to the deterministic optimum design. If so, a new local window and a new surrogate need to be constructed at the probabilistic optimum candidate to obtain accurate probabilistic optimum design.

5.1.2 Hyper-Spherical Local Window

Instead of using the global window, the local window concept is used for the generation of surrogate and VSVM models as shown in Fig. 5.1 to achieve accuracy and efficiency [Lee et al. 2011]. The radius R of the local window can be decided as

$$R = c_R \beta_t \tag{5.1}$$

where c_R is the coefficient for the size of local window, which is usually between 1.0 and 3.0, and β_t is the target reliability index of RBDO. However, Eq. (5.1) works only in the standard normal U-space where all random variables have the standard normal distribution, whereas the local window in X-space may not be a hyper-sphere. Hence, for the generation of samples in the X-space, samples are generated in the hyper-sphere with the radius *R* given in Eq. (5.1) in the normalized U-space, and the generated samples are transformed back to the X-space using the Rosenblatt transformation [Rosenblatt 1952]. If the target reliability index β_t is different for each constraint, then the maximum β_t can be used for Eq. (5.1).

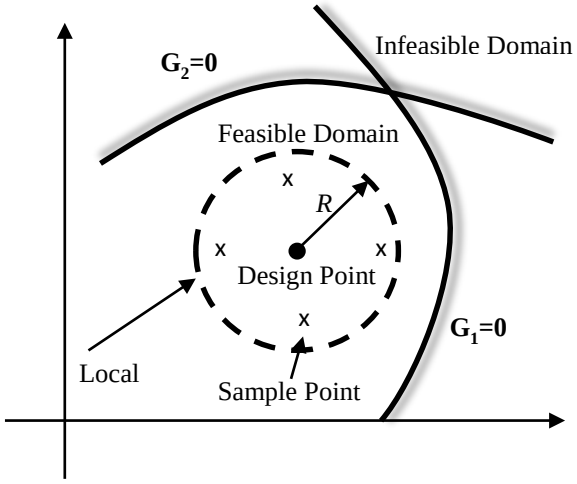

Figure 5.1 Hyper-Spherical Local Window for RBDO

Since VSVM is a classification method, VSVM performs better with a balanced sample set, which means there exists a similar number of positive and negative samples. Near the deterministic optimum, probabilities of failure of the active constraints are approximately 50%. Therefore, the local window for VSVM is centered at the

deterministic optimum, and c_R in Eq. (5.1) is set up to be reasonably large 2.5~3.0, so that the local window would include the RBDO optimum design and corresponding MCS samples in general. The distance between DDO optimum and RBDO optimum is approximately $1.0 \times \beta_t$ in the U-space in many problems. Thus, new local window with $c_R = 2.5 \sim 3.0$ can include new RBDO optimum and previous local window with $c_R = 1.2 \sim 1.5$. If c_R is small, we may need to construct a VSVM model whenever the design changes. If c_R is too large, we may end up with a number of DoE samples in unnecessary regions. In this way, the local window for VSVM does not have to be changed. On the contrary, Lee et al. used 1.2~1.5 for c_R and local windows are constructed whenever new RBDO optimum candidate is proposed [Lee et al. 2011]. By implementing a larger fixed local window, VSVM model is more stable and modeling time can be reduced. If the distance between the deterministic optimum and the RBDO optimum is large, then a new local window needs to be set up such that a new local window includes the RBDO optimum and corresponding MCS samples.

Instead of the hyper-cubic local window, the hyper-spherical local window is used in this research. As explained in Chapter 4, hyper-spherical local windows are more efficient than hyper-cubic local windows especially for high-dimensional problems due to the curse of dimensionality. Thus, the number of required samples can be reduced by employing the hyper-spherical local window [Lee et al. 2011]. To generate initial samples or test points for the calculation of stopping criteria in the hyper-spherical local window, Transformations/Gibbs sampling method is applied [Cumbus et al. 1996] in this research.

5.1.3 Filtering of Constraints

After computer simulations are carried out at DoE sample points in the local window, function values for each constraint are saved and used to determine if the constraints are feasible at those sample points or not. If function values for a certain

constraint are negative at all sample points on the local window, which means that the constraint is feasible because we define a constraint as failing if $G(\mathbf{X}) > 0$ in Eq. (2.1), then a surrogate model for the constraint is not generated because we can conclude that the probability of failure for the constraint will be zero without generating the surrogate model. Hence, if a constraint is identified as very feasible, -1 for the constraint value and 0 for the sensitivity of the constraint are assigned without generating a surrogate model to save computational cost [Lee et al. 2011].

5.1.4 Sample Reuse

At each iteration, the local window is scanned to check whether samples exist before generating N_r initial samples. If there are already existing samples whose number is denoted as N_e in the local window, and if N_e is less than N_r, then $N_r - N_e$ samples are generated in the local window instead of generating N_r samples. New samples are inserted in the sparse region first to avoid sampling in a clustered region. Even though N_e is larger than N_r, new samples are inserted in the sparse region only when existing samples are clustered within a small region.

During the deterministic optimization process, random parameters are ignored and only design variables are considered, whereas both random design variables and parameters are used during the RBDO process. Therefore, if there exist random parameters in the problems, samples generated during the deterministic optimization process cannot be reused during the RBDO process. Even though there is no random parameter in problems, samples from the DDO process can be clustered within the RBDO local window due to the size difference between DDO and RBDO local windows. And it is noted that such clustered DoE samples can cause numerical difficulty for the Kriging method [Bohling 2005b].

5.1.5 Generation of Virtual Samples

In Chapter 3, the original SVM is constructed first to identify informative samples before generating virtual samples. This procedure is necessary for local approximation methods such as simple linear approximation or moving least squares method (MLS), since local approximation results are only effective within a sub-local region and only DoE samples near the limit state function are effective. On the contrary, the Kriging method is a global approximation method, where all DoE samples are used in the modeling process. With the Kriging model, therefore, the informative sample set is not necessary and all combinations between positive and negative samples can be considered. However, it can be inefficient to use all combinations of samples since many virtual samples might be clustered within a small region as explained in Chapter 3. In selection of virtual samples, the trade-off between the amount of additional information and the computational cost is considered.

Figure 5.2 shows the overall algorithm flowchart of the sampling-based RBDO using VSVM for the limit state function and the probabilistic sensitivity analysis by the score function. At the deterministic optimum, the local window is defined and scanned to use existing samples within the local window. Additional DoE samples are generated if necessary and responses are evaluated at sample locations. Responses are converted into classification form and active and violated constraints are identified. Kriging models are constructed for active and violated constraints, virtual samples are generated, and virtual SVM is constructed. If current model is not accurate enough, sequential adaptive samples are inserted. Since sequential samples are located near limit states, the accuracy of Kriging model near the limit state is improved efficiently by new samples. Therefore, VSVM models are also accurate near limit states and RBDO optimum is searched for by using accurate VSVM models.

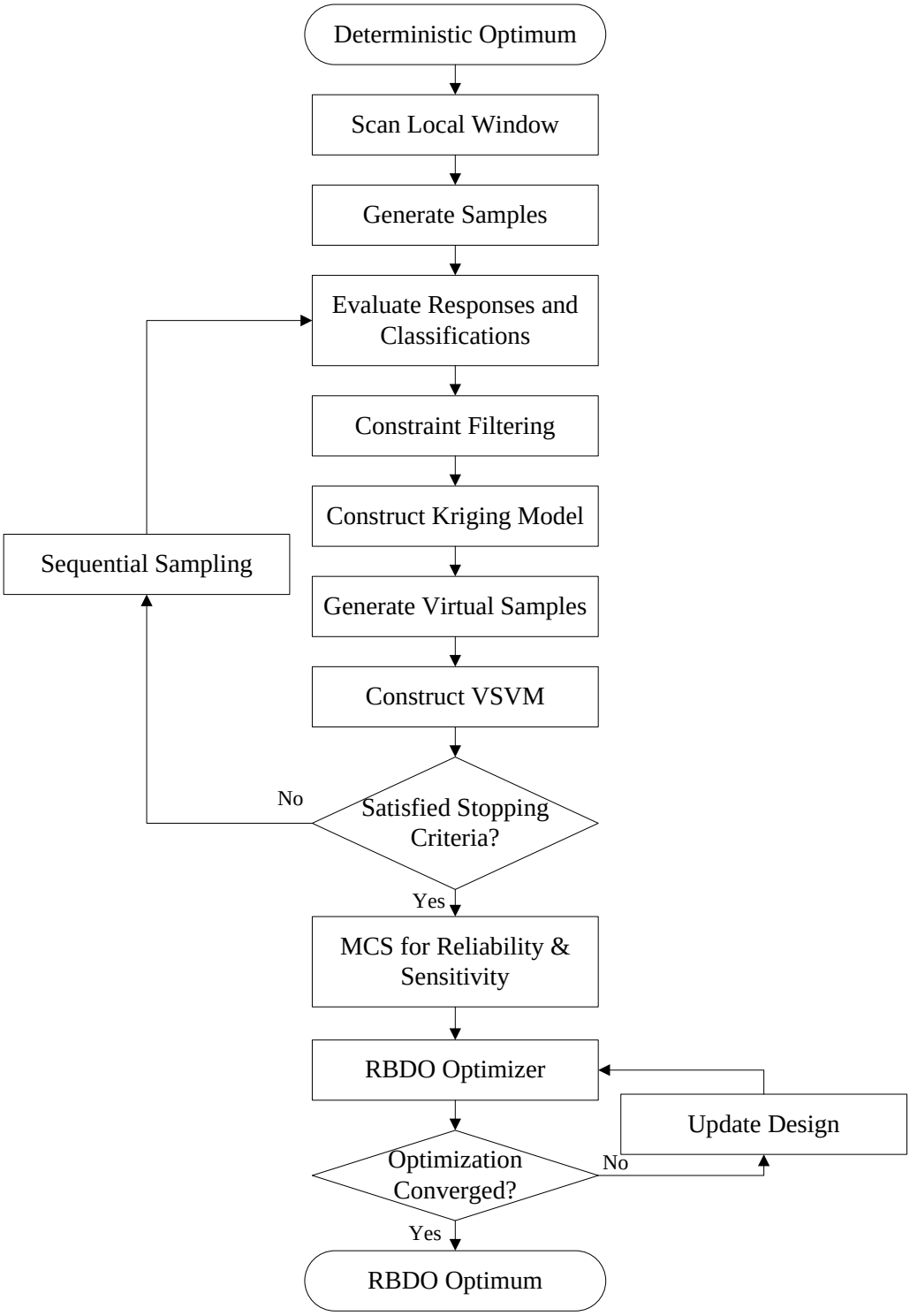

Figure 5.2 Flowchart of Sampling-Based RBDO Using VSVM

5.2 Comparison Study

5.2.1 Comparison Procedure

The sampling-based RBDO using the DKG method was compared with sampling-based RBDO using the VSVM. Two mathematical examples are used to compare performances of two sampling-based RBDO methods. In this research, a fixed γ value, which is small enough to maintain zero training error, is used for the Gaussian kernel in Eq. (3.13).

5.2.2 RBDO of Iowa 2-D Example

Consider a 2-D mathematical RBDO problem called the Iowa example [Tu et al. 1999; Youn et al. 2005; Lee et al. 2011], which is formulated to

$$\begin{aligned} &\text{minimize} \quad \text{Cost}(\mathbf{d}) = -\frac{(d_1 + d_2 - 10)^2}{30} - \frac{(d_1 - d_2 + 10)^2}{120} \\ &\text{subject to} \quad P(G_j(\mathbf{X}(\mathbf{d})) > 0) \le P_{F_j}^{\text{Tar}} = 2.275\%,\ j = 1 \sim 3 \\ &\qquad \mathbf{d}^L \le \mathbf{d} \le \mathbf{d}^U, \mathbf{d} \in \mathbf{R}^2 \text{ and } \mathbf{X} \in \mathbf{R}^2 \end{aligned} \tag{5.2}$$

where three constraint functions are expressed as

$$\begin{aligned} G_1(\mathbf{X}) &= 1 - \frac{X_1^2 X_2}{20} \\ G_2(\mathbf{X}) &= -1 + (Y-6)^2 + (Y-6)^3 - 0.6(Y-6)^4 + Z \\ G_3(\mathbf{X}) &= 1 - \frac{80}{X_1^2 + 8X_2 + 5} \end{aligned} \tag{5.3}$$

where $Y = 0.9063X_1 + 0.4226X_2$ and $Z = 0.4226X_1 - 0.9063X_2$, which are drawn in Fig. 5.3. The properties of two random variables are listed in Table 5.1, and they are independent. As shown in Eq. (5.2), the target probability of failure is 2.275% for all constraints. The sequential quadratic programming (SQP) algorithm is used for design optimization [Arora 2004].

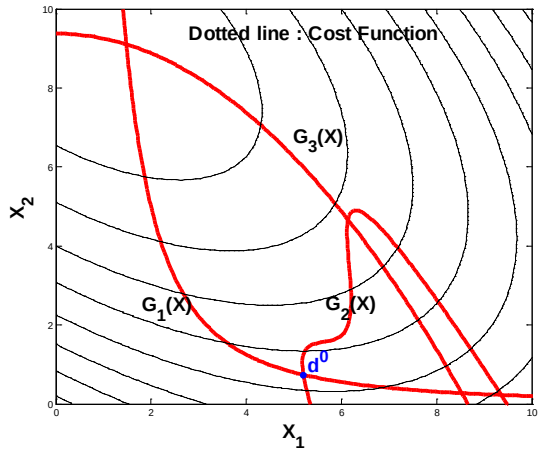

Figure 5.3 Shape of Cost and Constraint Functions

As shown in Fig. 5.4 and Table 5.1, the deterministic optimum design is used as initial design $\mathbf{d}_0 = (5.19, 0.74)^T$. To find the deterministic optimum, I-RBDO code, which is developed by the University of Iowa and contains the DKG method in addition to the sensitivity-based RBDO, is used. At the deterministic optimum design, the sampling-based RBDO using VSVM is launched with the local window coefficient $c_R = 2.5$. Using the same initial design, sampling-based RBDO using DKG is also performed. Table 5.2 shows the probabilistic sensitivities at the deterministic optimum. Both sampling-based RBDO results are close to probabilistic sensitivities with true functions, and thus both VSVM and DKG are accurate enough for sampling-based RBDO. Table 5.3 and Figs. 5.4 and 5.5 compare the numerical results of two different sampling-based RBDO methods. Sampling-based RBDO using VSVM results in better probability of failure while requiring a smaller number of samples, since the sequential sampling method for VSVM selects new samples near the limit state function as shown in Fig. 5.5. At each iteration, probability of failure is calculated at the current design using a large number of MCS points. Five design iterations are required for both sampling-based RBDO methods to

find the RBDO optimum from the DDO. In Table 5.3, sampling-based RBDO using VSVM requires 15 times less time due to its simpler explicit formulation. Therefore, sampling-based RBDO using VSVM is more efficient than sampling-based RBDO using DKG both in terms of the number of required samples and computational cost.

Table 5.1 Properties of Random Variables for Iowa 2-D Example

Random Variables	Distribution	$\mathbf{d}^L$	$\mathbf{d}^0$	$\mathbf{d}^U$	Standard Deviation
X_1	Normal	0.01	5.19	10.0	0.3
X_2		0.01	0.74	10.0	0.3

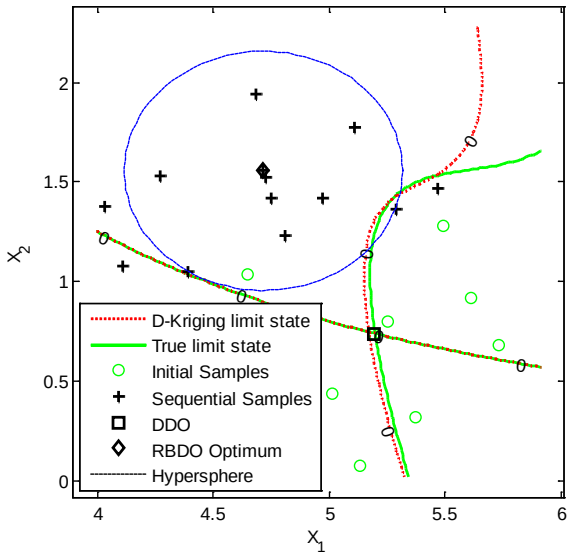

Figure 5.4 Result of Sampling-Based RBDO Using DKG

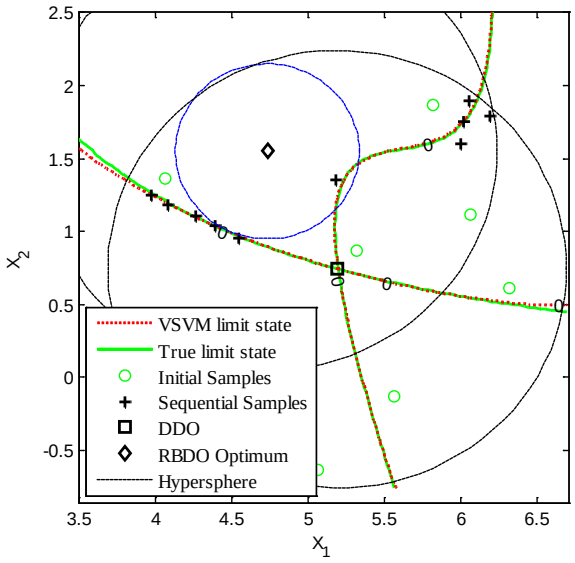

Figure 5.5 Result of Sampling-Based RBDO Using VSVM

Table 5.2 Probabilistic Sensitivities at the Deterministic Optimum for 2-D Example

Method	Using DKG		Using VSVM		Using True Functions	
Design Variables	X_1	X_2	X_1	X_2	X_1	X_2
Constraint 1	-16.2924	-56.0533	-16.5789	-56.0133	-16.2177	-55.9779
Constraint 2	57.7814	4.6555	57.6350	4.9174	57.7028	4.6091

Table 5.3 Comparison of Sampling-Based RBDO Using DKG vs. Using VSVM for 2-D Example

Sampling-based RBDO	Cost	Optimum Design		MCS (10,000,000)			
				P_{F_1}, %	P_{F_2}, %	Elapsed Time per Iteration (sec.)	Number of Function Evaluations
Using DKG	-1.9062	4.7153	1.5567	2.3399	1.9407	464	25
Using VSVM	-1.9093	4.7333	1.5489	2.3391	2.2623	31	20
Using True Functions	-1.9080	4.7380	1.5513	2.2441	2.3055	N.A.	N.A.

5.2.3 RBDO of 3-D example

Another mathematical example is used to compare two sampling-based RBDO methods [Arora 2004]. This can be formulated to

$$\begin{aligned} &\text{minimize} \quad \text{Cost}(\mathbf{d}) = (d_1 - 10)^2 + 5(d_2 - 12)^2 + d_3^4 \\ &\text{subject to} \quad P(G_j(\mathbf{X}(\mathbf{d})) > 0) \le P_{F_j}^{\text{Tar}} = 2.275\%, \\ &\qquad j = 1 \sim 4, \mathbf{d}^L \le \mathbf{d} \le \mathbf{d}^U, \\ &\qquad \mathbf{d} \in \mathbf{R}^3 \text{ and } \mathbf{X} \in \mathbf{R}^3 \end{aligned} \tag{5.4}$$

where four constraint functions are expressed as

$$\begin{aligned} G_1(\mathbf{X}) &= \frac{2X_1^2 + 3X_2^4 + X_3}{127} - 1 \\ G_2(\mathbf{X}) &= \frac{7X_1 + 3X_2 + 10X_3^2}{282} - 1 \\ G_3(\mathbf{X}) &= \frac{23X_1 + X_2^2}{196} - 1 \\ G_4(\mathbf{X}) &= \frac{4X_1^2 + X_2^2 - 3X_1X_2 + 2X_3^2}{20} - 1. \end{aligned} \tag{5.5}$$

There are three independent random design variables and four constraints. The deterministic optimum, which is obtained by the I-RBDO code, is used as the initial design $\mathbf{d}_0 = (2.881, 2.457, 1.000)^T$ as shown in Table 5.4 .

Table 5.4 Properties of Random Variables for 3-D Example

Random variables	Distribution	$\mathbf{d}^L$	$\mathbf{d}^0$	$\mathbf{d}^U$	Standard deviation
X_1	Normal	1	2.881	5	0.05
X_2		1	2.457	5	0.05
X_3		1	1.000	5	0.05

In Table 5.5, optimums obtained using both sampling-based RBDO methods are very close to the optimum using the sampling-based RBDO using true values. Their costs and probability of failures are also very close. However, sampling-based RBDO using VSVM requires less number of samples and less evaluation time than sampling-based RBDO using DKG. Six and five iterations are used for sampling-based RBDO methods using DKG and VSVM, respectively.

Table 5.5 Comparison of Sampling-Based RBDO Using DKG vs. Using VSVM for 3-D Example

Sampling-based RBDO	Cost	Optimum design			MCS (1,000,000)					Number of function evaluations
					P_{F_1}, %	P_{F_2}, %	P_{F_3}, %	P_{F4}, %	Elapsed time per iteration (sec.)	
Using DKG	517.66	2.7528	2.3653	1	2.2956	0	0	2.3150	94.6	51
Using VSVM	517.64	2.7519	2.3656	1	2.2982	0	0	2.2180	9.7	31
Using True Functions	517.67	2.7522	2.3653	1	2.2678	0	0	2.2756	N.A.	N.A.

5.2.4 RBDO of Correlated 2-D Example

Correlated Iowa 2-D example has the same formulation as Eq. (5.2) and Eq. (5.3). However, two input random variables are assumed to be correlated with the Clayton copula (τ = 0.5). With correlated input, RBDO optimum and corresponding MCS samples are located outside of the VSVM local window. Since VSVM may not show stable performances with unbalanced DoE samples, universal Kriging is used for designs far away from the deterministic optimum. In Table 5.6, both DKG and VSVM found reliable

optimal designs but sampling-based RBDO using VSVM required 25% less time and 18% less DoE samples. Relatively small difference in elapsed time is due to the fact that Kriging is also used for sampling-based RBDO using VSVM when RBDO optimum is far away from the deterministic optimum.

Both DKG and VSVM used 5 iterations in RBDO. With VSVM, unfortunately 4 designs out of 5 are located outside of the effective region of the VSVM local window. Therefore, 25% reduction in the computational cost in Table 5.6 is from one design within the effective region. Therefore, if DDO and RBDO optimums are located closer, the benefit will be more significant.

Table 5.6 Comparison of Sampling-Based RBDO Using DKG vs. Using VSVM for Correlated Iowa 2-D Example

Sampling-based RBDO	Cost	Optimum Design		MCS (1,000,000)		Elapsed Time for RBDO (sec.)	Number of Function Evaluations
				P_{F_1}, %	P_{F_2}, %		
Using DKG	-1.8859	5.0508	1.5904	2.3173	2.2632	328	60
Using VSVM	-1.8844	5.0595	1.5938	2.2649	2.2837	246	49

5.3 Conclusion

The virtual support vector machine (VSVM) is developed for sampling-based RBDO to improve the efficiency of the RBDO process. The number of samples is reduced to achieve accurate limit state function by employing the sequential sampling method. The evaluation time for probability of failure is also reduced due to the explicit formulation of VSVM.

The proposed method is compared with sampling-based RBDO using the DKG method. The DKG method can construct accurate surrogates with a relatively small number of samples, but it is inefficient for MCS since it has complicated expression for response evaluations, and the dynamic basis selection process requires significant computational effort. Since VSVM has simple explicit expression for limit state functions and only support vectors are used in response evaluations, sampling-based RBDO using VSVM can reduce the computational cost significantly. During the RBDO process, probability of failure is evaluated using MCS at each iteration, and thus the computational cost is proportional to the number of iterations. The number of required samples is also reduced by the efficient sequential sampling method. This is significant when expensive computer simulations are required for modeling and simulation. Therefore, the VSVM is more efficient than the DKG method in estimating response values for MCS and in reducing the number of required samples.

CHAPTER 6
ACCURACY IMPROVEMENT STRATEGIES FOR THE KRIGING METHOD

6.1 Introduction

Deterministic computer simulations of physical phenomena have been widely used in science and engineering for design guidance; and as more sophisticated and larger size models are used, they become computationally expensive. Therefore, it is of great importance to minimize the number of computer experiments in DDO and RBDO. The main advantage of the surrogate-based method is that a limited number of function evaluations can be used to construct surrogate models. Therefore, surrogate models are introduced to reduce the computational burden for solving problems without sensitivity. The Kriging method is one of the widely used surrogate modeling methods. To construct an accurate Kriging model, an appropriate form of the Kriging model should be selected and the correlation parameters should be estimated accurately. Therefore, the number of computer experiments (i.e., DoE) for surrogates can be reduced, if accuracy of surrogates is improved by using appropriate forms and parameters.

Section 6.2 briefly reviews accurate correlation parameter estimation methods for the Kriging method, and Section 6.3 explains penalized maximum likelihood estimation (PMLE) for small sample size problems. In Sections 6.4 and 6.5, appropriate correlation models and mean structures in the Kriging model are selected, and a combined method, which includes all previous improvement schemes, is proposed. Performances of proposed methods for numerical and engineering examples are shown in Section 6.6. Section 6.7 concludes the current study.

6.2 Accurate Parameter Estimation

Parameter estimation is the process of selecting the regression function coefficient, process variance, and correlation function parameters $\{\boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta}\}$ in Eq. (3.8).

To estimate accurate parameters, various statistical techniques are available, such as the variographic analysis (VA), Bayesian estimation (BE), cross-validation (CV), or MLE [Zimmerman and Zimmerman 1991; Martin and Simpson 2005; Dubourg et al. 2011]. In geostatistics, the empirical variogram is usually obtained first from the autocovariance structure of the data [LeMay 1995; Bohling 2005; Hengl 2007; Roustant et al. 2012]. However, this methodology requires users' interaction and knowledge on the Kriging method, so it is very sophisticated and not appropriate for our purpose. With the BE technique, prior information is needed, which might not be available [Dubourg et al. 2011]. In computer experiments, the MLE technique is most widely used, and Martin and Simpson showed that the MLE method outperformed the CV technique [Martin and Simpson 2005]. Therefore, the MLE technique will be mainly used.

In MLE, the log-likelihood function L in Eq. (3.8) is maximized. Since σ^2 and $\boldsymbol{\beta}$ can be estimated using Eqs. (3.7) and (3.9), the goal of the MLE method is to find optimal $\boldsymbol{\theta}$ that maximizes the likelihood function based on all observations. It is a global optimization problem, and various optimization algorithms have been applied, such as the downhill simplex method [Martin and Simpson 2005; Martin 2009; Deng et al. 2011], the Newton-Raphson method [Martin 2009; Deng et al. 2011], the quasi-Newton method [Martin and Simpson 2005; Gano et al. 2006; Deng et al. 2011], the Fisher scoring algorithm [Martin 2009; Deng et al. 2011], the adaptive simulated annealing [Gano et al. 2006; Deng et al. 2011], the genetic algorithm [Forrester and Keane 2009; Zhao et al. 2011] and generalized pattern search algorithm [Gano et al. 2006; Deng et al. 2011; Zhao et al. 2011]. The first four gradient-based methods are local optimization methods, so they are not appropriate for highly nonlinear problems. Adaptive simulated annealing is much more computationally intensive because it is a Monte Carlo method [Gano et al. 2006; Deng et al. 2011]. Zhao et al. showed that GPS performed better than H-J, L-M, or GA and the performance of GPS is influenced by its initial design point [Zhao et al. 2011]. Thus, if the initial design is far from true optimum, GPS is computationally

expensive and sometimes inaccurate. Therefore, GA can be used to find a better initial point for GPS. Even though GA is considered unreliable for continuous optimization problems, GA can provide useful initial design for GPS.

6.3 Penalized MLE (PMLE)

The log-likelihood function near the optimum may be flat, or it can give wrong information in some situations [Fan and Li 2001; Li and Sudjianto 2005; Kok 2012]. For example, true optimal correlation parameter in normalized root mean square error (NRMSE), which is defined in Eq. (6.10), is different from the optimal correlation parameter in MLE in Fig. 6.1. One way to solve this problem is to add a constraint $\theta < \theta^{UB}$ in MLE, but it is another problem to choose an appropriate threshold θ^{UB}. Therefore, penalized MLE (PMLE) is introduced for parameter estimation instead of MLE [Fan and Li 2001; Li and Sudjianto 2005; Roustant et al. 2012]. In PMLE, a penalty function is added to the log-likelihood function, such as

$$Q = -\frac{n}{2}\ln[2\pi\sigma^2] - \frac{1}{2}\ln[|\mathbf{R}|] - \frac{1}{2\sigma^2}(\mathbf{Y}-\mathbf{F}\boldsymbol{\beta})^T\mathbf{R}^{-1}(\mathbf{Y}-\mathbf{F}\boldsymbol{\beta}) - N\sum_{i=1}^{nr} p_\lambda(\theta_i) \qquad (6.1)$$

where *N* is the number of DoE samples, *nr* is the number of random variables, p_λ is the penalty function, and λ is the parameter in the penalty function. Three penalty functions are introduced; they are the L_1 penalty, $p_\lambda(\theta) = \lambda|\theta|$; the L_2 penalty, $p_\lambda(\theta) = 0.5\lambda|\theta|^2$; and the smoothly clipped absolute deviation (SCAD) penalty. The first derivative of the SCAD penalty is defined as

$$p'_\lambda(\theta) = \lambda\left\{ I(\theta \le \lambda) + \frac{(a\lambda-\theta)_+}{(a-1)\lambda} I(\theta > \lambda) \right\} \qquad (6.2)$$

where $a = 3.7$, $\theta > 0$, with $p_\lambda(0) = 0$ [Li and Sudjianto 2005]. The SCAD penalty function is recommended in the literature [Li and Sudjianto 2005]. However, other

penalty functions showed similar performances with global optimization algorithms, so the L_1 penalty is applied in this research. The amount of penalty is decided by the penalty function parameter λ. In the literature [Li and Sudjianto 2005], a set of grid samples for λ are chosen, and the best λ is selected based on CV error. However, when exhaustive grid samples are used, it is not easy to decide the grid size and the penalty parameter range for λ. If the grid size is too small, the computational cost becomes unnecessarily high. On the other hand, if the grid size is too large, the optimum λ may not be found. With grid samples, a large penalty parameter range also increases the computational cost. Therefore, an optimization algorithm needs to be applied to enhance the efficiency. Local optimization methods such as a golden section method [Arora 2004] or a gradient-based method are tested, but they often fail to find the optimum. Cross-validation error is not a smooth function and may have multiple local optima, so local optimization algorithms often fall into local optima. Therefore, GPS is applied to find optimal λ in PMLE.

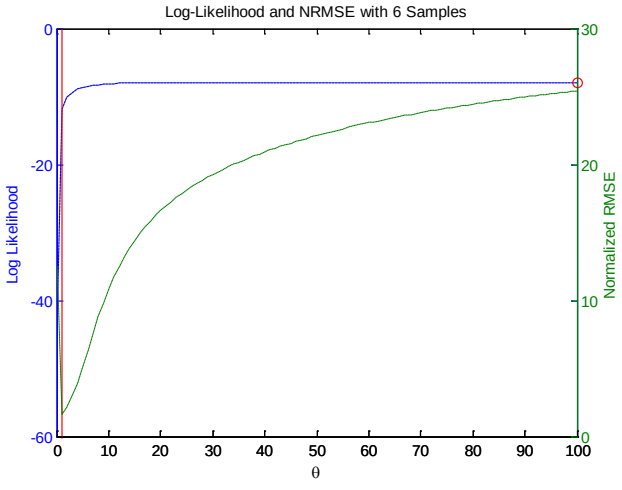

Figure 6.1 Log-Likelihood Function and NRMSE with Six Samples

6.4 Correlation Model Selection

The choice of the correlation model is crucial to the Kriging model. Gaussian correlation models are widely applied for many engineering problems, since corresponding Kriging models are infinitely differentiable and smooth. However, the best correlation model can be different for different data or problems. If mean structures are the same for given data, the residual information is also the same. Then, MLE can be used to select the best correlation function. Seven different correlation models are considered in Table 6.1 [Lophaven et al. 2002].

Table 6.1 Correlation Functions

Name	$R_j(\theta_j, x_1, x_2)$
Exponential	$\exp(-\theta_j \lvert x_2 - x_1 \rvert)$
General Exponential	$\exp(-\theta_j \lvert x_2 - x_1 \rvert^{\theta_{n+1}}),\ 0 < \theta_{n+1} \le 2$
Gaussian	$\exp(-\theta_j \lvert x_2 - x_1 \rvert^2)$
Linear	$\max\{0, 1-\theta_j \lvert x_2 - x_1 \rvert\}$
Spherical	$1-1.5\xi_j + 0.5\xi_j^3,\ \ \xi_j = \min\{1,\ \theta_j \lvert x_2 - x_1 \rvert\}$
Cubic	$1-3\xi_j^2 + 2\xi_j^3, \qquad \xi_j = \min\{1,\ \theta_j \lvert x_2 - x_1 \rvert\}$
Spline	$\begin{cases} 1-15\xi_j^2 + 30\xi_j^3, & \text{for } 0 \le \xi_j \le 0.2 \\ 1.25(1-\xi_j)^3, & \text{for } 0 < \xi_j < 1 \\ 0, & \text{for } \xi_j \ge 1, \end{cases}$ where $\xi_j = \theta_j \lvert x_2 - x_1 \rvert$

6.5 Mean Structure Selection

In DKG, the best mean structure is selected based on the minimum of process variance σ^2 in Eq. (3.9) [Zhao et al. 2011]. However, DKG tends to choose the model

with full basis functions for its mean structure, and chosen models are often not the best in terms of accuracy. For example, in Table 6.2, full basis functions (first order UKG) are selected based on σ^2, but the best model is ordinary Kriging method (OKG), whose mean value is a constant, in terms of NRMSE. This means that σ^2 may not be a good measure to select the better mean structure. On the contrary, cross-validation error identified the best model correctly in Table 6.2 and thus CV is applied instead of σ^2 to decide better mean structures. However, CV is computationally expensive, and it may not be sensitive enough to identify the difference among all basis function combinations. Therefore, the best mean structure is chosen among only OKG and first- and second-order UKG methods.

Table 6.2 σ^2, CV and NRMSE for 12-D Dixon-Price Problem

Basis Selection													σ^2	CV	NRMSE
1	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}			
1	0	0	0	0	0	0	0	0	0	0	0	0	6.12E10	7.24E11	12.80%
1	1	0	0	0	0	0	0	0	0	0	0	0	6.06E10	7.82E11	12.86%
1	1	1	0	0	0	0	0	0	0	0	0	0	6.06E10	8.98E11	12.86%
1	1	1	1	0	0	0	0	0	0	0	0	0	6.06E10	9.48E11	12.86%
1	1	1	1	1	0	0	0	0	0	0	0	0	6.03E10	1.05E12	12.85%
1	1	1	1	1	1	0	0	0	0	0	0	0	5.71E10	1.24E12	13.00%
1	1	1	1	1	1	1	0	0	0	0	0	0	5.63E10	1.32E12	13.09%
1	1	1	1	1	1	1	1	0	0	0	0	0	5.63E10	1.54E12	13.09%
1	1	1	1	1	1	1	1	1	0	0	0	0	5.61E10	1.76E12	13.09%
1	1	1	1	1	1	1	1	1	1	0	0	0	5.42E10	2.13E12	13.60%
1	1	1	1	1	1	1	1	1	1	1	0	0	5.42E10	2.21E12	13.60%
1	1	1	1	1	1	1	1	1	1	1	1	0	5.42E10	2.61E12	13.59%
1	1	1	1	1	1	1	1	1	1	1	1	1	5.32E10	2.91E12	13.36%

Note: 1 means selected and 0 means not selected.

6.6 Numerical Experiments

6.6.1 Analytical Examples

To test the effectiveness of the proposed methods, a set of analytical functions are employed [Tu et al. 1999; Youn et al. 2005; Dixon and Szego 1978; Lee 2007; Viana et al. 2008]. These are:

Iowa 2-D second constraint function (2 variables)

$$\begin{gathered} y(\mathbf{x}) = 1 + (0.9063 \cdot x_1 + 0.4226 \cdot x_2)^2 + (0.9063 \cdot x_1 + 0.4226 \cdot x_2 - 6)^3 \\ -0.6(0.9063 \cdot x_1 + 0.4226 \cdot x_2)^4 - (-0.4226 \cdot x_1 + 0.9063 \cdot x_2) \\ 0.01 \le x_1, x_2 \le 10 \end{gathered} \tag{6.3}$$

Branin-Hoo (2 variables)

$$\begin{gathered} y(\mathbf{x}) = \left(x_2 - \frac{5.1x_1^2}{4\pi^2} + \frac{5x_1}{\pi} - 6 \right)^2 + 10\left(1 - \frac{1}{8\pi} \right)\cos(x_1) + 10, \\ -5 \le x_1 \le 10, 0 \le x_2 \le 15. \end{gathered} \tag{6.4}$$

Camelback (2 variables)

$$\begin{gathered} y(\mathbf{x}) = \left(\frac{x_1^4}{3} - 2.1x_1^2 + 4 \right)x_1^2 + x_1x_2 + (4x_2^2 - 4)x_2^2, \\ -3 \le x_1 \le 3, -2 \le x_2 \le 2. \end{gathered} \tag{6.5}$$

Hartmann 3 (3 variables)

$$\begin{gathered} y(\mathbf{x}) = -\sum_{i=1}^{q} a_i \exp\left(-\sum_{j=1}^{m} b_{ij}(x_j - d_{ij})2 \right), \\ 0 \le x_j \le 1, m = 3, q = 4, \mathbf{a} = [1, 1.2, 3.0, 3.2], \end{gathered} \tag{6.6}$$

$$\mathbf{b} = \begin{bmatrix} 3.0 & 10.0 & 30.0 \\ 0.1 & 10.0 & 35.0 \\ 3.0 & 10.0 & 30.0 \\ 0.1 & 10.0 & 35.0 \end{bmatrix},$$

$$\mathbf{d} = \begin{bmatrix} 0.3689 & 0.1170 & 0.2673 \\ 0.4699 & 0.4387 & 0.7470 \\ 0.1091 & 0.8732 & 0.5547 \\ 0.03815 & 0.5743 & 0.8828 \end{bmatrix}$$

Hartmann 6 (6 variables)

$$y(\mathbf{x}) = -\sum_{i=1}^{q} a_i \exp\left(-\sum_{j=1}^{m} b_{ij}(x_j - d_{ij})2 \right),$$

$$0 \le x_j \le 1, m = 6, q = 4, \mathbf{a} = [1, 1.2, 3.0, 3.2],$$

$$\mathbf{b} = \begin{bmatrix} 10.0 & 3.0 & 17.0 & 3.5 & 1.7 & 8.0 \\ 0.05 & 10.0 & 17.0 & 0.1 & 8.0 & 14.0 \\ 3.0 & 3.5 & 1.7 & 10.0 & 17.0 & 8.0 \\ 17.0 & 8.0 & 0.05 & 10.0 & 0.1 & 14.0 \end{bmatrix}, \tag{6.7}$$

$$\mathbf{d} = \begin{bmatrix} 0.1312 & 0.1696 & 0.5569 & 0.0124 & 0.8283 & 0.5886 \\ 0.2329 & 0.4135 & 0.8307 & 0.3736 & 0.1004 & 0.9991 \\ 0.2348 & 0.1451 & 0.3522 & 0.2883 & 0.3047 & 0.6650 \\ 0.4047 & 0.8828 & 0.8732 & 0.5743 & 0.1091 & 0.0381 \end{bmatrix}$$

Extended Rosenbrock (9 variables)

$$y(\mathbf{x}) = \sum_{i=1}^{m-1} \left[(1 - x_i)^2 + 100(x_{i+1} - x_i^2)^2 \right], \tag{6.8}$$

$$-5 \le x_i \le 10, i = 1, 2, \cdots, m, m = 9.$$

Dixon-Price (12 variables)

$$y(\mathbf{x}) = (x_1 - 1)^2 + \sum_{i=2}^{m} i\left[2x_i^2 - x_{i-1}\right]^2, \quad -10 \le x_i \le 10, i = 1, 2, \cdots, m, m = 12. \tag{6.9}$$

Different numbers of samples are used depending on the number of variables in each example. Sample locations are generated using Latin Centroidal Voronoi Tessellations (LCVT), which has better uniformity compared with Latin hypercube sampling (LHS) [Saka et al. 2007]. For the CV error, leave-one-out CV (LOOCV) is applied. For the Kriging model, DACE MATLAB toolbox is used [Lophaven et al. 2002].

6.6.2 Engineering Example

As described by He et al., statistical analysis in irregular wave and uncertainty quantification in variable regular wave are presented for resistance and motions [He et al. 2012]. There are two input variables and they are zero-crossing period *T* and the significant wave height *H*. Deterministic regular head wave simulation for total/added resistance, heave and pitch motions, is studied. For each target performance function, mean, RMS, amplitude and period values are investigated. Therefore, there are twelve performance functions in the study. Figure 6.2 shows the shapes of all performance functions. One hundred twenty-nine true samples are evaluated using computational fluid dynamics (CFD) simulations. Among the 129 DoE samples, 9, 17, 33, and 65 samples are selected and used to construct Kriging models. The unused 120, 112, 96, and 64 samples, respectively, are used to evaluate the accuracy of surrogates. Therefore, there are 4×12=48 CFD test cases.

The normalized root mean square error (NRMSE) is used to compare the accuracies of different surrogates and is defined as

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{N_t}\sum_{i=1}^{N_t}\left[y_i - \hat{y}_i\right]^2}}{\max\left\{y_i\right\} - \min\left\{y_i\right\}} \tag{6.10}$$

where N_t is the number of test points, y_i is the i^{th} true response or the i^{th} CFD result, and $\hat{y}_i$ is the i^{th} prediction using Kriging models.

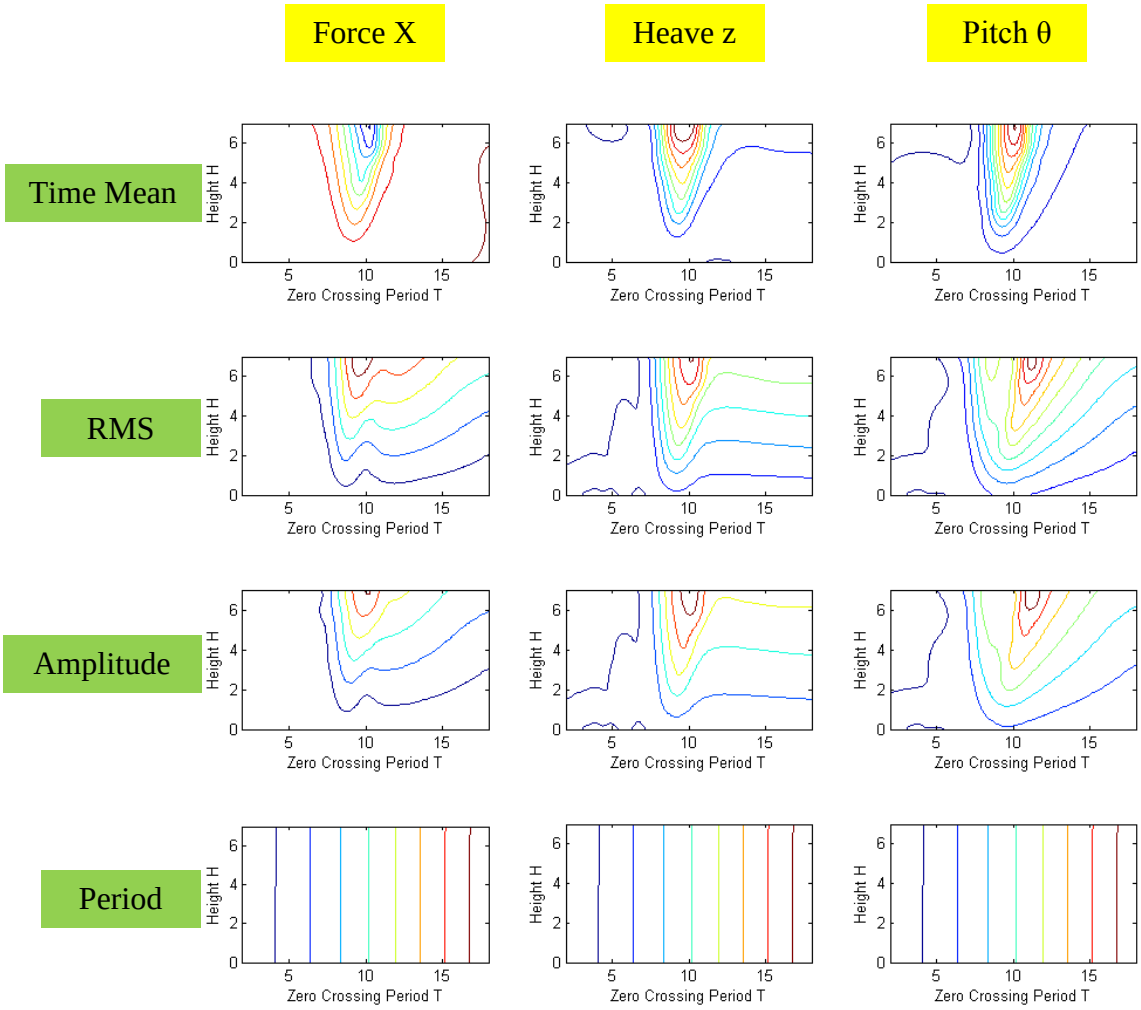

Figure 6.2 Responses of CFD Performance Functions

6.6.3 Parameter Estimation in MLE

The GA is proposed to find the initial point for GPS in Section 6.2. Using mathematical examples, H-J, GPS and GPS with GA are compared. The mathematical examples are Iowa 2-D, Branin-Hoo, Camelback, Rosenbrock, and Dixon-Price functions. The correlation function is fixed as Gaussian.

According to Table 6.3, GPS with GA performs better than H-J or GPS, especially for high-dimensional problems. Even though the computational cost is about twice that of GPS alone, it is relatively small compared with expensive computer simulation (DoE) time.

Table 6.3 NRMSE with Different Parameter Search Algorithms

Examples	Iowa 2-D	Branin-Hoo	Camelback	Rosenbrock	Dixon-Price
No. of Samples	10	10	50	50	60
H-J	10.57%	9.95%	0.04%	12.37%	12.35%
GPS	10.55%	9.80%	0.04%	12.37%	12.35%
GPS with GA	5.85%	9.80%	0.04%	0.70%	1.13%

6.6.4 Penalized MLE

When MLE is inaccurate, PMLE can be used as described in Section 6.3. Five mathematical examples are used to test the performance PMLE. There are three different penalty functions in PMLE: L_1 penalty function, L_2 penalty function and the SCAD penalty function. Performances are compared in Table 6.4. There is no one penalty function that is always the best. Among three penalty functions, L_1 penalty function

performs well in Table 6.4 and it has the simplest mathematical form. Therefore, L_1 penalty function is used in this study.

Table 6.4 NRMSE for Different Penalty Functions in PMLE

Penalty Function	Iowa 2-D	Branin-Hoo	Camelback	Rosenbrock	Dixon-Price
SCAD	8.76	7.95	14.16	13.82	13.02
L_1	9.13	7.95	11.56	10.23	10.59
L_2	9.26	6.15	11.79	15.13	11.11

Using L_1 penalty function, MLE and PMLE are compared. Five mathematical examples are applied and grid sampling method is used to find the best penalty function parameter λ in PMLE. For correlation parameter estimations, GPS with GA is applied for MLE and GPS is applied for PMLE. It is relatively easy to find optimum $\boldsymbol{\theta}$ of Q in Eq. (2.2) [Li and Sudjianto 2005], thus GPS is enough for PMLE. According to Table 6.5, PMLE performs better than MLE for relatively small sample size problems.

Table 6.5 NRMSE for MLE and PMLE

Examples	Iowa 2-D	Branin-Hoo	Camelback	Rosenbrock	Dixon-Price
No. of Samples	10	10	10	20	30
MLE	12.08%	9.35%	15.83%	15.21%	13.02%
PMLE	9.13%	9.21%	11.56%	10.24%	10.59%

Next, to find the best λ in PMLE, GPS is proposed instead of the grid sampling method. The grid sampling method, golden section search, and GPS are compared using five mathematical examples. The range of λ is [0, 10]. For the grid sampling method, 100 grid samples are used, and thus 100 LOOCV evaluations are required. For golden section search and GPS, stopping tolerance is set to 10^{-4}, which is smaller resolution than the grid sampling method.

In Table 6.6, GPS found true optima in all tests. Golden section search required 19~25 LOOCV evaluations, and GPS required 23~42 evaluations, whereas grid sampling method used 100 evaluations. Therefore, they are more efficient than the grid sampling method. However, golden section search often falls into local optimum in Table 6.6. Therefore, GPS is more efficient and accurate than other methods.

Table 6.6 Parameter λ from Different Search Algorithms

Examples	Iowa 2-D	Branin-Hoo	Camelback	Rosen-brock	Dixon-Price
No. of Samples	10	10	10	20	30
Grid Sampling	0.01	3.64	0.01	0.01	0.01
Golden Section	1.62	3.82	1.46	10.00	10.00
GPS	0.01	9.00	0.01	0.00	0.00

When the log-likelihood function is flat near the optimum or is misleading, PMLE performs better than MLE. However, PMLE performs worse than MLE for relatively large samples since the performance of PMLE is dependent on the CV error. Furthermore, PMLE is much more expensive than MLE due to CV evaluations, so application of PMLE should be limited to cases when PMLE is better than MLE. MLE has difficulty when the log-likelihood function is flat near the optimum or the optimum is

on the upper bound [Fan and Li 2001; Li and Sudjianto 2005; Kok 2012]. Therefore, PMLE is only applied when the optimal log-likelihood function value is the same as the log-likelihood function value at the upper bound.

6.6.5 Correlation Function Selection

In Section 6.4, correlation function selection by MLE is proposed. Five mathematical examples are used for comparison tests. As shown in Table 6.7, performances are different with different correlation functions, and correct correlation functions are identified by MLE except for the Branin-Hoo example. Even with the Branin-Hoo example, however, the difference in NRMSE is very small (0.58% vs. 0.53%), which means that those two correlation functions have similar shapes. Overall, better correlation models are identified by MLE, and the performance is improved.

Table 6.7 NRMSE for Different Correlation Functions

	Gaussian	Cubic	Exponential	Spline	Linear	Spherical	General Exponential
Iowa 2-D	**0.04%**	10.49%	6.16%	1.75%	5.85%	5.91%	0.69%
Branin-Hoo	**0.58%**	8.90%	4.54%	1.28%	3.87%	4.31%	0.53%
Camel-back	**0.04%**	10.55%	2.26%	0.76%	3.33%	2.29%	1.55%
Rosen-brock	12.35%	12.35%	12.35%	12.35%	12.35%	12.35%	**0.44%**
Dixon-Price	12.36%	12.36%	12.36%	12.36%	12.36%	12.36%	**0.73%**

Note: Bold numbers are the correlation functions chosen by MLE.

6.6.6 Mean Structure Selection

In Section 6.5, a mean structure selection method using CV is proposed instead of σ^2-based DKG. To compare two Kriging methods, Iowa 2-D, Branin-Hoo, Camelback, Rosenbrock, Dixon-Price, and CFD engineering problems are used. Two different sample sets are applied to each mathematical example, and four different samples sets are applied to engineering examples. Therefore, 58 cases are tested overall. To remove the effect of other factors, correlation functions are fixed as Gaussian and GPS with GA is applied for correlation parameter estimations. In this study, DKG using CV means the mean structure of the DKG method is selected by using CV error among many different basis function combinations.

In Table 6.8, DKG using CV is more accurate than DKG using σ^2. Therefore, the process variance σ^2 does not seem to be a good measure for mean structure selection. However, DKG using CV is very expensive since many different basis function combinations are examined. Furthermore, CV may not be accurate enough to identify the differences between all basis function combinations, especially with large sample size.

Table 6.8 Performances of DKG's Using σ^2 or CV

	Average NRMSE	Total Elapsed Time (Sec)
DKG using σ^2	7.91%	784
DKG using CV	6.85%	1,537

Therefore, it is proposed that the best model is chosen among the zeroth-, first- and second-order UKG methods based on CV error and it will be defined as CV-based DKG in this research. The Hartmann3 and Hartmann6 functions are included in this test, so 62 cases are tested. Correlation functions are fixed as Gaussian and GPS with GA is

applied for correlation parameter estimations. According to Table 6.9, the accuracy of DKG using CV is similar to that of the zeroth- and first-order UKG, but it is far more expensive. CV-based DKG is the most accurate, and the computational cost is much less than DKG using CV. Thus, CV does not seem to be accurate enough to identify all small differences between tested mean structures.

Table 6.9 Performances of Different Kriging Methods

Methods	0^{th} UKG	1^{st} UKG	2^{nd} UKG	DKG using CV	CV-based DKG
Average NRMSE	6.28%	6.55%	8.42%	6.45%	5.92%
Average Time (sec)	0.71	0.71	0.84	48.49	6.54

6.6.7 Combined Scheme

Four different schemes are introduced to improve the accuracy of Kriging. They are (1) better parameter search in MLE, (2) PMLE for small sample size, (3) better correlation model selection, and (4) better mean structure selection. They are combined into one process as shown in Fig. 6.3 and this combined process is defined as CV-based DKG with optimum correlation in this study. For given samples, the shape of the log-likelihood function is examined to see if it has a flat region near the optimum. If the log-likelihood function is inaccurate, PMLE is used in correlation function selection. Otherwise, the best correlation function is identified by MLE. Once a correlation model is selected, the best mean structure is chosen based on CV error. Parameters are estimated at each stage, and all parameter estimations are done by GPS with GA. With 62 tests, average NRMSE is 5.71%, and average elapsed time is 62.83 sec. Therefore, by employing the proposed CV-based DKG with optimum correlation, the accuracy of the

surrogates is improved but more time is required. However, in general, such modeling time is much smaller than conventional computer experiments such as CFD and FEA.

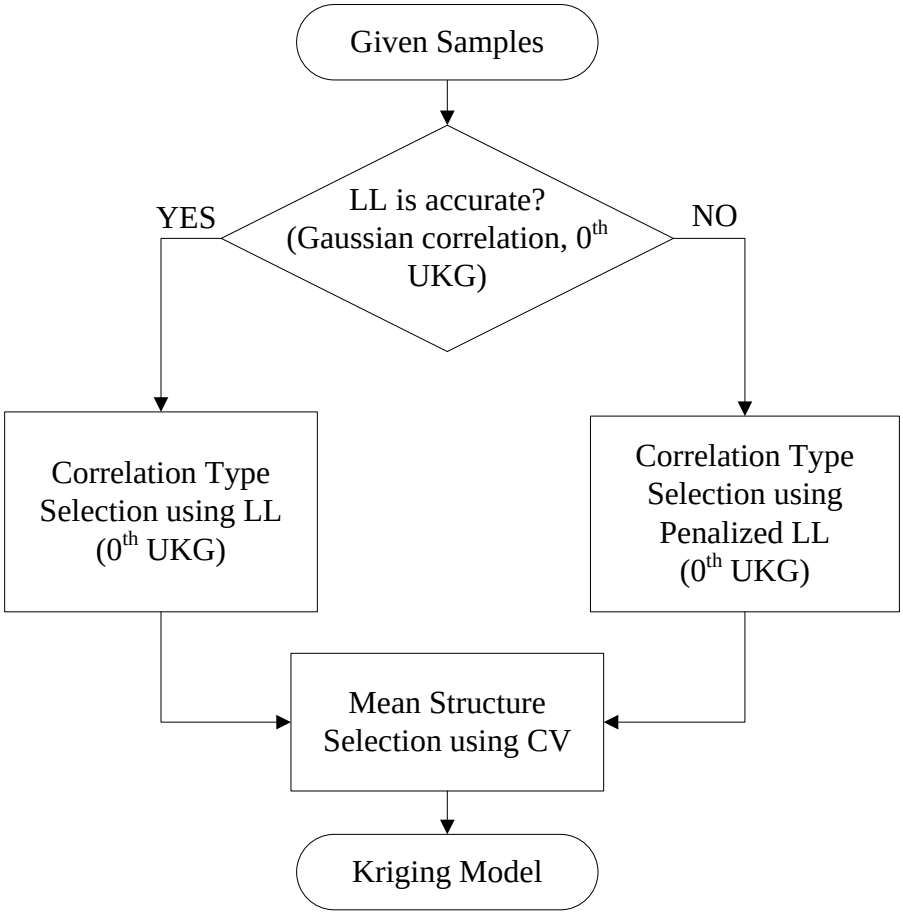

Figure 6.3 Flowchart of CV-Based DKG with Optimum Correlation (LL Means the Log-Likelihood Function)

6.7 Conclusion

The accuracy of the Kriging method is improved by employing four different methods: (1) GPS with GA for parameter search in MLE, (2) PMLE with small sample size, (3) correlation model selection by MLE, and (4) mean structure selection by CV

error. Parameter estimation in PMLE is also improved. Each method is shown to improve the accuracy of the Kriging model. All four methods are combined into one process, and the CV-based DKG with optimum correlation shows improved accuracy with mathematical and engineering examples.

CHAPTER 7

CONCLUSION, RESEARCH PROGRESS AND FUTURE WORK

7.1 Conclusion

The reliability-based design optimization (RBDO) methods that use sensitivity have been widely applied to various engineering applications. Using sensitivities, the most probable point (MPP)-based RBDO can be carried out for approximating the reliability of the system and searching for reliable designs. However, the sensitivity often is not available or is difficult to obtain in complex multi-physics or multidisciplinary simulation-based engineering design applications. In this case, the surrogate-based method can provide approximations of otherwise expensive computer simulations for reliability analysis and design optimization. In RBDO, once accurate surrogate models are constructed, Monte Carlo simulation (MCS) can be directly applied to the surrogate model for the reliability analysis and probabilistic sensitivity analysis for RBDO with affordable computational cost. This method is called sampling-based RBDO.

Previously, the dynamic Kriging (DKG) method is used for surrogates in sampling-based RBDO. However, response evaluations at MCS points using DKG are relatively expensive, because DKG uses all DoE samples and its formulation includes complicated matrix calculations. On the contrary, support vector machines (SVM) only use support vectors which is a part of the total DoE sample set and have a simpler formulation. Therefore, the virtual SVM (VSVM) is developed to improve the accuracy of SVM and achieve similar accuracy level as DKG. A comprehensive study has been carried out to show that VSVM is more efficient and accurate than existing methods such as DKG and the explicit design space decomposition (EDSD).

A sequential sampling method is proposed to construct the accurate decision function efficiently. The accuracy of the VSVM model is improved by inserting new samples near the limit state function. The classification error is used for convergence

criterion for VSVM because response values are meaningless in classification methods and only accurate classification results are needed for the sampling-based RBDO.

Hyper-spherical local windows are used to overcome the curse of dimension for high dimensional problems. The transformation/Gibbs sampling method is also introduced to generate initial DoE samples and convergence test points in hyper-spherical windows efficiently. Active and violated constraints are identified not to generate unnecessary VSVM models. For efficiency, existing samples are reused in different local windows. These efficiency strategies are implemented to perform a sampling-based RBDO using VSVM.

As for the DKG method, the total computational cost can be reduced by using less number of DoE samples for expensive computer simulations. Therefore, four accuracy improvement strategies are investigated. They are: (1) generalized pattern search (GPS) with genetic algorithm (GA) for parameter search in maximum likelihood estimation (MLE), (2) penalized MLE (PMLE) with small sample size, (3) optimum correlation model selection by MLE, and (4) mean structure selection by cross-validation (CV) error. These four methods are combined into one process to develop a new the CV-based DKG method, which shows improved accuracy with mathematical and engineering examples.

7.2 Research Progress

The main objective of this study is to develop an efficient and accurate classification method and apply it to sampling-based RBDO. Another objective is to improve the Kriging model accuracy. Four tasks have been carried out to meet these objectives of the study:

Task 1: Development of an efficient and accurate classification method;

Task 2: Development of an efficient sequential sampling method;

Task 3: Development of a sampling-based RBDO using the proposed method;

Task 4: Development of more accurate Kriging method.

7.3 Future Work

Performance of SVM can be unstable with unbalanced DoE samples and performance of VSVM is influenced by the location of DoE samples. To make the VSVM process more stable, universal Kriging responses at test points can be used in generating virtual samples. Since universal Kriging responses at test points are already evaluated for the constraint boundary sampling method, additional computational cost will be minimal.

The RBDO problem usually has multiple constraints and the current VSVM method requires building VSVM model for each constraint. It would be more efficient if we can construct one VSVM model for multiple constraints. In CV-based DKG with optimum correlation, PMLE is very conservatively applied. If a threshold condition, under which MLE is not accurate, can be identified accurately, then the value of PMLE can be significant. Finally, in selection of the mean structure, the proposed DKG method chooses among 0^{th} , 1^{st}, and 2^{nd} order universal Kriging methods. If a more accurate identification method is available, we can choose among various combinations of basis functions and more accurate DKG models can be obtained.

REFERENCES

Ali, S. and Smith, K. A., "Automatic Parameter Selection for Polynomial Kernel," *Information Reuse and Integration, IEEE International Conference,* 2003.

Arora, J.S., *Introduction to Optimum Design,* Elsevier/Academic Press, 2004.

Au, S. K., "Reliability-Based Design Sensitivity by Efficient Simulation," *Computers & Structures*, Vol. 83, No. 14, pp. 1048-1061, 2005.

Barton, R. R., "Metamodeling: A State of the Art Review," *WSC '94: Proceedings of the 26th conference on Winter Simulation,* San Diego, CA, USA, 1994.

Basudhar, A., Dribusch, C., Lacaze, S. and Missoum, S., "Constrained Efficient Global Optimization with Support Vector Machines," *Structural and Multidisciplinary Optimization*, Vol. 46, No. 2, pp. 201-221, 2012.

Basudhar, A. and Missoum, S., "An Improved Adaptive Sampling Scheme for the Construction of Explicit Boundaries," *Structural and Multidisciplinary Optimization*, Vol. 42, No. 4, pp. 517-529, 2010.

Basudhar, A. and Missoum, S., "Adaptive Explicit Decision Functions for Probabilistic Design and Optimization using Support Vector Machines," *Computers & Structures*, Vol. 86, No. 19-20, pp. 1904-1917, 2008.

Basudhar, A., Missoum, S. and Harrison Sanchez, A., "Limit State Function Identification using Support Vector Machines for Discontinuous Responses and Disjoint Failure Domains," *Probabilistic Engineering Mechanics*, Vol. 23, No. 1, pp. 1-11, 2008.

Bect, J., Ginsbourger, D., Li, L., Picheny, V. and Vazquez, E., "Sequential Design of Computer Experiments for the Estimation of a Probability of Failure," *Statistics and Computing*, Vol. 22, No. 3, pp. 773-793, 2012.

Ben-Hur, A. and Weston, J., *A User's Guide to Support Vector Machines,* Humana Press, 2008.

Bichon, B. J., Eldred, M. S., Swiler, L. P., Mahadevan, S. and McFarland, J. M., "Efficient Global Reliability Analysis for Nonlinear Implicit Performance Functions," *AIAA Journal,* Vol. 46, No. 10, pp. 2459; 10-2468, 2008.

Bichon, B. J., McFarland, J. M. and Mahadevan, S., "Efficient Surrogate Models for Reliability Analysis of Systems with Multiple Failure Modes," *Reliability Engineering & System Safety*, Vol. 96, No. 10, pp. 1386-1395, 2011.

Boardman, M. and Trappenberg, T., "A Heuristic for Free Parameter Optimization with Support Vector Machines," *Neural Networks, 2006. IJCNN '06. International Joint Conference on,* 2006.

Bohling, G., "Introduction to Geostatistics and Variogram Analysis," *Kansas Geological Survey C&PE*, 940, 2005a.

Bohling, G., "Kriging," *Kansas Geological Survey C&PE*, 940, 2005b.

Booker, A.J., Dennis Jr, J., Frank, P.D., "Optimization using surrogate objectives on a helicopter test example," TR97734-S, December 1997.

Booker, A.J., Dennis Jr, J., Frank, P.D., "A rigorous framework for optimization of expensive functions by surrogates," NASA Langley Research Center, 98-47, Hampton, Virginia, November 1998.

Bourinet, J. -., Deheeger, F. and Lemaire, M., "Assessing Small Failure Probabilities by Combined Subset Simulation and Support Vector Machines," *Structural Safety*, Vol. 33, No. 6, pp. 343-353, 2011.

Breitung, K., "Asymptotic Approximations for Multinormal Integrals," *Journal of Engineering Mechanics*, Vol. 110, No. 3, pp. 357-366, 1984.

Burges, C. J. C., "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, Vol. 2, No. 2, pp. 121-167, 1998.

Burkardt, J., Gunzburger, M., Peterson, J. and Brannon, R., "User Manual and Supporting Information for Library of Codes for Centroidal Voronoi Point Placement and Associated Zeroth, First, and Second Moment Determination," *SAND Report 2002-0099, Sandia National Laboratories,* Albuquerque, 2002.

Butler, N. A., "Optimal and Orthogonal Latin Hypercube Designs for Computer Experiments," *Biometrika*, Vol. 88, No. 3, pp. 847-857, 2001.

Byrd, R. H., Gilbert, J. C. and Nocedal, J., "A Trust Region Method Based on Interior Point Techniques for Nonlinear Programming," *Mathematical Programming*, Vol. 89, No. 1, pp. 149-185, 2000.

Canu, S., Grandvalet, Y., Guigue, V. and Rakotomamonjy, A., "SVM and Kernel Methods Matlab Toolbox," 2005.

Cawley, G. C., "Model Selection for Support Vector Machines Via Adaptive Step-Size Tabu Search," *Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms,* Prague, Czech Republic, 2001.

Chaloner, K. and Verdinelli, I., "Bayesian Experimental Design: A Review," *Statistical Science*, Vol. 10, No. 3, pp. 273-304, 1995.

Chapelle, O. and Vapnik, V., "Model Selection for Support Vector Machines." *NIPS*, 1999.

Chapelle, O., Vapnik, V., Bousquet, O. and Mukherjee, S., "Choosing Multiple Parameters for Support Vector Machines," *Machine Learning*, Vol. 46, No. 1, pp. 131-159, 2002.

Cherkassky, V. and Mulier, F., Learning from Data : Concepts, Theory, and Methods (Adaptive and Learning Systems for Signal Processing, Communications and Control Series), Wiley-Interscience, 1998.

Ching, J., Applications of Monte Carlo Method in Science and Engineering, InTech, 2011.

Choi, K. K., Tu, J. and Park, Y., "Extensions of Design Potential Concept for Reliability-Based Design Optimization to Nonsmooth and Extreme Cases," *Structural and Multidisciplinary Optimization*, Vol. 22, No. 5, pp. 335-350, 2001.

Chung, K., Kao, W., Sun, C., Wang, L. and Lin, C., "Radius Margin Bounds for Support Vector Machines with the RBF Kernel," *Neural Comput.*, Vol. 15, No. 11, pp. 2643-2681, 2003.

Coleman, T. F. and Li, Y., "An Interior Trust Region Approach for Nonlinear Minimization Subject to Bounds," *SIAM Journal on Optimization*, Vol. 6, No. 2, pp. 418-445, 1996.

Coleman, T. F. and Li, Y., "On the Convergence of Interior-Reflective Newton Methods for Nonlinear Minimization Subject to Bounds," *Mathematical Programming*, Vol. 67, No. 1, pp. 189-224, 1994.

Cressie, N.A.C., *Statistics for Spatial Data*, John Wiley & Sons, New York, 1991.

Cristianini, N., and Shawe-Taylor, J., *An Introduction to Support Vector Machines : And Other Kernel-Based Learning Methods*, Cambridge University Press, Cambridge, U.K. ; New York :, 2000.

Cumbus, C., Damien, P. and Walker, S., "Uniform Sampling in the Hypersphere Via Latent Variables and the Gibbs Sampler," Http://hdl.Handle.net/2027.42/35534, Vol. 9612, No. 36, 1996.

Currin, C., Toby Mitchell, Morris, M. and Don Ylvisaker, "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.

Deng, H., Shao, W., Ma, Y. and Wei, Z., "Bayesian Metamodeling for Computer Experiments using the Gaussian Kriging Models," *Quality and Reliability Engineering International*, 2011.

Ditlevsen, O. and Madsen, H.O., *Structural Reliability Methods*, Citeseer, 1996.

Dixon, L.C.W. and Szego G.P., *Towards Global Optimisation 2,* North-Holland Pub. Co, 1978.

Du, X. and Chen, W., "Sequential Optimization and Reliability Assessment Method for Efficient Probabilistic Design," *Journal of Mechanical Design (Transactions of the ASME),* Vol. 126, No. 2, pp. 225-233, 2004.

Dubourg, V., Sudret, B. and Bourinet, J., "Reliability-Based Design Optimization using Kriging Surrogates and Subset Simulation," *Structural and Multidisciplinary Optimization,* Vol. 44, No. 5, pp. 673-690, 2011.

Fan, J. and Li, R., "Variable Selection Via Nonconcave Penalized Likelihood and its Oracle Properties," *Journal of the American Statistical Association,* Vol. 96, No. 456, pp. 1348-1360, 2001.

Fang, K., Li, R. and Sudjianto, A., *Design and Modeling for Computer Experiments,* CRC Press, 2010.

Forrester, A., Sobester, A. and Keane, A., *Engineering Design Via Surrogate Modelling, A Practical Guide,* John Wiley & Sons, United Kingdom, 2008.

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

Frohlich, H. and Zell, A., "Efficient Parameter Selection for Support Vector Machines in Classification and Regression Via Model-Based Global Optimization," *Proceedings of the IEEE International Joint Conference on Neural Networks,* 2005.

Gano, S. E., Renaud, J. E., Martin, J. D. and Simpson, T. W., "Update Strategies for Kriging Models used in Variable Fidelity Optimization," *Structural and Multidisciplinary Optimization,* Vol. 32, No. 4, pp. 287-298, 2006.

Ginsbourger, D., Dupuy, D., Badea, A., Carraro, L. and Roustant, O., "A Note on the Choice and the Estimation of Kriging Models for the Analysis of Deterministic Computer Experiments," *Applied Stochastic Models in Business and Industry,* Vol. 25, No. 2, pp. 115-131, 2009.

Ginsbourger, D., Riche, R. and Carraro, L., "Kriging is Well-Suited to Parallelize Optimization," , Vol. 2,pp. 131-162, 2010.

Giunta, A. A., Wojtkiewicz, S. F. J. and Eldred, M. S., "Overview of Modern Design of Experiments Methods for Computational Simulations," *AIAA's 3rd Annual Aviation Technology, Integration, and Operations (ATIO) Forum,* 2003.

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.

Gold, C. and Sollich, P., "Fast Bayesian Support Vector Machine Parameter Tuning with the Nystrom Method," *Neural Networks, 2005. IJCNN '05. Proceedings. 2005 IEEE International Joint Conference on,* 2005.

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

Hasofer, A. M. and Lind, N. C., "Exact and Invariant Second-Moment Code Format," *Journal of the Engineering Mechanics Division*, Vol. 100, No. 1, pp. 111-121, 1974.

He, W., Diez, M., Peri, D., Campana, E. F., Tahara, Y. and Stern, F., "Uncertainty Quantification of Delft Catamaran Total Added Resistance and Motions in Irregular and Variable Regular Head Wave," *29th Symposium on Naval Hydrodynamics*, Gothenburg, Sweden, August 2012.

Hedayat, A., Sloane, N.J.A. and Stufken, J., *Orthogonal Arrays: Theory and Applications,* Springer Verlag, 1999.

Hengl, T., A Practical Guide to Geostatistical Mapping of Environmental Variables, Joint Research Centre, Luxembourg, 2007.

Hohenbichler, M. and Rackwitz, R., "Sensitivity and Importance 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," *Journal of Engineering Mechanics*, Vol. 114, No. 12, pp. 2195-2199, 1988.

Hou, G.J., Gumbert, C.R. and Newman, P.A., A most Probable Point-Based Method for Reliability Analysis, Sensitivity Analysis and Design Optimization, Citeseer, 2004.

Hsu, C.W., Chang, C.C., and Lin, C.J., "A Practical Guide to Support Vector Classification," Department of Computer Science and Information Engineering, National Taiwan University, Taiwan, 2004.

Huang, C. and Wang, C., "A GA-Based Feature Selection and Parameters Optimizationfor Support Vector Machines," *Expert Systems with Applications*, Vol. 31, No. 2, pp. 231-240, 2006.

Huntington, D. E. and Lyrintzis, C. S., "Improvements to and Limitations of Latin Hypercube Sampling," *Probabilistic Engineering Mechanics*, Vol. 13, No. 4, pp. 245-253, 1998.

Hurtado, J. E. and Alvarez, D. A., "Classification Approach for Reliability Analysis with Stochastic Finite-Element Modeling," *Journal of Structural Engineering*, Vol. 129, No. 8, pp. 1141-1149, 2003.

Jiang, P., Basudhar, A. and Missoum, S., "Reliability Assessment with Correlated Variables using Support Vector Machines," *52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Denver, 2011.

Jin, R., Chen, W. and Simpson, T. W., "Comparative Studies of Metamodelling Techniques Under Multiple Modelling Criteria," *Structural and Multidisciplinary Optimization*, Vol. 23, No. 1, pp. 1-13, 2001.

Jones, D. R., Schonlau, M. and Welch, W. J., "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, Vol. 13, No. 4, pp. 455-492, 1998.

Jones, D. R., "A Taxonomy of Global Optimization Methods Based on Response Surfaces," *Journal of Global Optimization*, Vol. 21, No. 4, pp. 345-383, 2001.

Joseph, V. R., Hung, Y. and Sudjianto, A., "Blind Kriging: A New Method for Developing Metamodels," *Journal of Mechanical Design*, Vol. 130, No. 3, pp. 031102, 2008.

Kalagnanam, J. R. and Diwekar, U. M., "An Efficient Sampling Technique for Off-Line Quality Control," *Technometrics*, Vol. 39, No. 3, pp. pp. 308-319, 1997.

Kang, S., Koh, H. and Choo, J. F., "An Efficient Response Surface Method using Moving Least Squares Approximation for Structural Reliability Analysis," *Probabilistic Engineering Mechanics*, Vol. 25, No. 4, pp. 365-371, 2010.

Kecman, V., Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models, MIT Press, Cambridge, Mass., 2001.

Kecman, V., *Support Vector Machines – an Introduction*, Springer Berlin / Heidelberg, 2005.

Keerthi, S. S., "Efficient Tuning of SVM Hyperparameters using radius/margin Bound and Iterative Algorithms," *Neural Networks, IEEE Transactions on*, Vol. 13, No. 5, pp. 1225-1229, 2002.

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, Nov. 2005.

Kim, B., Lee, Y. and Choi, D., "Comparison Study on the Accuracy of Metamodeling Technique for Non-Convex Functions," *Journal of Mechanical Science and Technology*, Vol. 23, No. 4, pp. 1175-1181, 2009.

Kleijnen, J. P. C., "Kriging Metamodeling in Simulation: A Review," *European Journal of Operational Research*, Vol. 192, No. 3, pp. 707-716, 2009.

Koehler, J. and Owen, A., "Computer Experiments," *Handbook of Statistics*, Vol. 13, No. 13, pp. 261-308, 1996.

Kok, S., "The Asymptotic Behaviour of the Maximum Likelihood Function of Kriging Approximations using the Gaussian Correlation Function," *EngOpt 2012 - International Conference on Engineering Optimization*, Rio de Janeiro, Brazil, 1-5 July 2012.

Kulkarni, A., Jayaraman, V. K. and Kulkarni, B. D., "Support Vector Classification with Parameter Tuning Assisted by Agent-Based Technique," *Computers & Chemical Engineering*, Vol. 28, No. 3, pp. 311-318, 2004.

Leary, S., Bhaskar, A. and Keane, A., "Optimal Orthogonal-Array-Based Latin Hypercubes." *Journal of Applied Statistics*, Vol. 30, No. 5, pp. 585, 2003.

Lee, I., Choi, K. K., Du, L. and Gorsich, D., "Inverse Analysis Method using MPP-Based Dimension Reduction for Reliability-Based Design Optimization of Nonlinear and Multi-Dimensional Systems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 198, No. 1, pp. 14-27, 2008.

Lee, J., "A Novel Three-Phase Trajectory Informed Search Methodology for Global Optimization," *Journal of Global Optimization*, Vol. 38, No. 1, pp. 61-77, 2007.

Lee, I., Choi, K. K. and Zhao, L., "Sampling-Based RBDO using the Stochastic Sensitivity Analysis and Dynamic Kriging Method," *Structural and Multidisciplinary Optimization*, Vol. 44, No. 3, pp. 299-317, 2011.

Lee, T. H. and Jung, J. J., "A Sampling Technique Enhancing Accuracy and Efficiency of Metamodel-Based RBDO: Constraint Boundary Sampling," *Computers & Structures*, Vol. 86, No. 13–14, pp. 1463-1476, 2008.

LeMay, N. E., "Variogram Modeling and Estimation," Ph. D. Thesis, University of Colorado at Denver, 1995.

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.

Li, R. and Sudjianto, A., "Analysis of Computer Experiments using Penalized Likelihood in Gaussian Kriging Models," *Technometrics*, Vol. 47, No. 2, pp. 111-120, 2005.

Lin, H., and Lin, C., "A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods," Department of Computer Science, National Taiwan University, 2003.

Lophaven, S.N., Nielsen, H.B. and Søndergaard, J., *DACE: A Matlab Kriging Toolbox*, Citeseer, 2002.

Madsen, H.O., Krenk, S. and Lind, N.C., *Methods of Structural Safety*, Courier Dover Publications, 2006.

Martin, J. D. and Simpson, T. W., "A Study on the use of Kriging Models to Approximate Deterministic Computer Models," *Proceedings of DETC*, 2003.

Martin, J. D., "Computational Improvements to Estimating Kriging Metamodel Parameters," *Journal of Mechanical Design*, Vol. 131, No. 8, pp. 084501, 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., Beckman, R. J. and Conover, W. J., "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.

Men, C. and Wang, W., "Selection of Gaussian Kernel Parameter for SVM Based on Convex Estimation," *Proceedings of the 5th international symposium on Neural Networks: Advances in Neural Networks*, Berlin, Heidelberg, 2008.

Momma, M. and Bennett, K. P., "A Pattern Search Method for Model Selection of Support Vector Regression," *In Proceedings of the SIAM International Conference on Data Mining*, 2002.

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, 2009.

Noh, Y., Choi, K. K. and Lee, I., "Identification of Marginal and Joint CDFs using Bayesian Method for RBDO," *Structural and Multidisciplinary Optimization*, Vol. 40, No. 1, pp. 35-51, 2010.

Owen, A. B., "Orthogonal Arrays for Computer Experiments, Integration and Visualization," *Statistica Sinica*, Vol. 2,pp. 439-452, 1992.

Park, J., "Optimal Latin-Hypercube Designs for Computer Experiments," *Journal of Statistical Planning and Inference*, Vol. 39, No. 1, pp. 95-111, 1994.

Picheny, V., Ginsbourger, D., Roustant, O., Haftka, R. T. and Kim, N., "Adaptive Designs of Experiments for Accurate Approximation of a Target Region," *Journal of Mechanical Design*, Vol. 132, 2010.

Powell, M.J., The Convergence of Variable Metric Methods for Non-Linearly Constrained Optimization Calculations, Academic Press, 1978a.

Powell, M., "A Fast Algorithm for Nonlinearly Constrained Optimization Calculations," , Vol. 630,pp. 144-157, 1978b.

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

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

Rai, R., *Qualitative and Quantitative Sequential Sampling*, Ph.D. Thesis, The University of Texas at Austin, 2006.

Ranjan, P., Bingham, D. and Michailidis, G., "Sequential Experiment Design for Contour Estimation from Complex Computer Codes," *Technometrics*, Vol. 50, No. 4, pp. 527-541, 2008.

Refaeilzadeh, P., Tang, L. and Liu, H., "Cross-Validation," *Encyclopedia of Database Systems*, Vol. 3, pp. 532-538, 2009.

Romero, V. J., Burkardt, J. V., Gunzburger, M. D. and Peterson, J. S., "Comparison of Pure and "Latinized" Centroidal Voronoi Tessellation Against various Other Statistical Sampling Methods," *Reliability Engineering & System Safety*, Vol. 91, No. 10, pp. 1266-1280, 2006.

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

Roustant, O., Ginsbourger, D. and Deville, Y., "DiceKriging, DiceOptim: Two R Packages for the Analysis of Computer Experiments by Kriging-Based Metamodeling and Optimization," *Journal of Statistical Software*, Vol. 51, No. 1, 2012.

Rubinstein, R.Y., *Simulation and the Monte Carlo Method*, Wiley, 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.

Sacks, J., Schiller, S. B. and Welch, W. J., "Designs for Computer Experiments," *Technometrics*, Vol. 31, No. 1, pp. 41-47, 1989.

Saka, Y., Gunzburger, M. and Burkardt, J., "Latinized, Improved LHS, and CVT Point Sets in Hypercubes," *International Journal of Numerical Analysis and Modeling*, Vol. 4, No. 3-4, pp. 729-743, 2007.

Schölkopf, B., Advances in Kernel Methods Support Vector Learning, MIT Press, Cambridge, Mass., 1999.

Schölkopf, B. and Smola, A.J., Learning with Kernels : Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, Cambridge, Mass. :, 2002.

Schölkopf, B., Bartlett, P., Smola, A. and Williamson, R., "Shrinking the Tube: A New Support Vector Regression Algorithm," 1999.

Seeger, M., "Bayesian Model Selection for Support Vector Machines, Gaussian Processes and Other Kernel Classifiers," *Advances in Neural Information Processing Systems*, Vol. 12,pp. 603-609, 2000.

Shewry, M. C. and Wynn, H. P., "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 Safety*, Vol. 2, No. 3, pp. 209-240, 2001a.

Simpson, T. W., Poplinski, J. D., Koch, P. N. and Allen, J. K., "Metamodels for Computer-Based Engineering Design: Survey and Recommendations," *Engineering with Computers*, Vol. 17, No. 2, pp. 129-150, 2001b.

Staelin, C., "Parameter Selection for Support Vector Machines," HP Laboratories, HPL-2002-354, Israel, 2002.

Tu, J., Choi, K. K. and Park, Y. H., "Design Potential Method for Robust System Parameter Design," *AIAA Journal*, Vol. 39, No. 4, pp. 667-677, 2001.

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

Valdebenito, M. and Schuëller, G., "A Survey on Approaches for Reliability-Based Optimization," *Structural and Multidisciplinary Optimization*, Vol. 42, No. 5, pp. 645-663, 2010.

Vapnik, V. and Chapelle, O., "Bounds on Error Expectation for Support Vector Machines," *Neural Computation*, Vol. 12, No. 9, pp. 2013-2036, 2000.

Vapnik, V.N., The Nature of Statistical Learning Theory, Springer, New York, 2000.

Vapnik, V.N., *Statistical Learning Theory*, Wiley, New York, 1998.

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., "SURROGATES Toolbox User's Guide," 2010.

Viana, F. A. C., Haftka, R. and Watson, L., "Sequential Sampling for Contour Estimation with Concurrent Function Evaluations," *Structural and Multidisciplinary Optimization*, Vol. 45, No. 4, pp. 615-618, 2012.

Voronoï, G., "Nouvelles Applications Des Paramètres Continus à La Théorie Des Formes Quadratiques. Deuxième Mémoire. Recherches Sur Les Paralléllοèdres rimitifs." *Journal Für Die Reine Und Angewandte Mathematik*, Vol. 134, pp. 198-287, 1908.

Waltz, R. A., Morales, J. L., Nocedal, J. and Orban, D., "An Interior Algorithm for Nonlinear Optimization that Combines Line Search and Trust Region Steps," *Mathematical Programming*, Vol. 107, No. 3, pp. 391-408, 2006.

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. 11, 2007.

Wang, W. and Ma, L., "An Estimation of the Optimal Gaussian Kernel Parameter for Support Vector Classification," *Proceedings of the 5th international symposium on Neural Networks: Advances in Neural Networks*, Berlin, Heidelberg, 2008.

Wang, W., Xu, Z., Lu, W. and Zhang, X., "Determination of the Spread Parameter in the Gaussian Kernel for Classification and Regression," *Neurocomputing*, Vol. 55, No. 3-4, pp. 643-663, 2003.

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

Wong, T., Luk, W. and Heng, P., "Sampling with Hammersley and Halton Points," *Journal of Graphics, Gpu, and Game Tools*, Vol. 2, No. 2, pp. 9-24, 1997,.

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

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

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 Multidimensional Integration in Stochastic Mechanics," *International Journal for Numerical Methods in Engineering*, Vol. 61, No. 12, pp. 1992-2019, 2004.

Youn, B. D., Choi, K. K. and Park, Y. H., "Hybrid Analysis Method for Reliability-Based Design Optimization," *Journal of Mechanical Design*, Vol. 125,pp. 221, 2003.

Youn, B., Choi, K. K. and Du, L., "Enriched Performance Measure Approach for Reliability-Based Design Optimization," *AIAA Journal*, Vol. 43, No. 4, 2005.

Zeeb, C.N., and Burns, P.J., "A Comparison of Failure Probability Estimates by Monte Carlo Sampling and Latin Hypercube Sampling," Sandia National Laboratories, Albuquerque, NM, 1997.

Zhao, L., "Reliability-Based Design Optimization using Surrogate Model with Assessment of Confidence Level," Ph. D. Thesis, University of Iowa, 2011.

Zhao, L., Choi, K. K. and Lee, I., "Metamodeling Method using Dynamic Kriging for Design Optimization," *AIAA Journal*, Vol. 49, No. 9, pp. 2034-2046, 2011.

Zimmerman, D. L. and Zimmerman, M. B., "A Comparison of Spatial Semivariogram Estimators and Corresponding Ordinary Kriging Predictors," *Technometrics*, Vol. 33, No. 1, pp. 77-91, 1991.