COMPUTATIONAL OPTIMAL DESIGN AND UNCERTAINTY QUANTIFICATION OF COMPLEX SYSTEMS USING EXPLICIT DECISION BOUNDARIES

By

Anirban Basudhar

A Dissertation Submitted to the Faculty of the

DEPARTMENT OF AEROSPACE AND MECHANICAL ENGINEERING

In Partial Fulfillment of the Requirements

For the Degree of

DOCTOR OF PHILOSOPHY

WITH A MAJOR IN MECHANICAL ENGINEERING

In the Graduate College

THE UNIVERSITY OF ARIZONA

2011

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Dissertation Committee, we certify that we have read the dissertation prepared by Anirban Basudhar
entitled Computational Optimal Design and Uncertainty Quantification of Complex Systems Using Explicit Decision Boundaries
and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

Date: 29 April 2011
Samy Missoum

Date: 29 April 2011
Parviz Nikravesh

Date: 29 April 2011
Achintya Haldar

Date: 29 April 2011
Guzin Bayraksan

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to the Graduate College.
I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

Date: 29 April 2011
Dissertation Director: Samy Missoum

STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at the University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

SIGNED: Anirban Basudhar

ACKNOWLEDGEMENTS

I express my sincere gratitude and appreciation to my dissertation supervisor Dr. Samy Missoum for his inspiring guidance, valuable suggestions and constant support and encouragement. His drive for attaining perfection greatly enhanced the quality of the dissertation. It has been a great learning experience working with him.

I thank all the faculty members of the Aerospace and Mechanical Engineering, and Civil Engineering and Applied Mechanics Department of the University of Arizona for providing me immense opportunities to gain valuable academic knowledge through various courses and seminars. I thank Dr. P.Nikravesh, Dr. A.Haldar, Dr. T.Kundu, and Dr. G.Bayraksan for agreeing to become members of my Ph.D dissertation advisory committee. Their constructive suggestions from time to time and critical perusal of the thesis improved its quality. Although Dr. T.Kundu was unable to be part of the final oral examination, his contributions during all the earlier stages were valuable.

I thank my dissertation supervisor Dr. Samy Missoum and the Department of Aerospace and Mechanical Engineering for providing me with Research and Teaching Assistantships respectively, which enabled me to pursue my doctoral research and complete the same. The support of the National Science Foundation (award CMMI-0800117 and CMMI-1029257) is gratefully acknowledged.

I am also thankful to Dr. J.M.Bourinet for inviting me to France for an internship, and also for his valuable inputs to my research.

I thank all my colleagues of the research group for their help, discussion and company. I thank my friends namely Ron, Tamaki, Bharani, Ajoy, Sudib, Mayank, Gaurav, Avinash, Sagar, Niranjan, Gauri, Sanhita and many others, who made my stay at Tucson vibrant, memorable and pleasurable. They have been a great source of help and solace in difficult times. I would always cherish my company with them. I would always fondly remember the company of my elder sister Debashree during my stay here. She has been my friend and guide throughout my stay.

Finally, I want to thank my parents who have always been a great source of inspiration and support. They have always encouraged me to try my best to reach greater heights, but provided me with the freedom and flexibility to pursue my career as I like without exerting any parental pressure.

DEDICATION

To my parents and my sister who have always been there for support through good and bad times, and have always been a great source of inspiration and comfort.

TABLE OF CONTENTS

TABLE OF CONTENTS – *Continued*

TABLE OF CONTENTS – *Continued*

TABLE OF CONTENTS – *Continued*

TABLE OF CONTENTS – *Continued*

LIST OF FIGURES

LIST OF FIGURES – *Continued*

LIST OF FIGURES – *Continued*

LIST OF FIGURES – *Continued*

LIST OF TABLES

ABSTRACT

This dissertation presents a sampling-based method that can be used for uncertainty quantification and deterministic or probabilistic optimization. The objective is to simultaneously address several difficulties faced by classical techniques based on response values and their gradients. In particular, this research addresses issues with discontinuous and binary (pass or fail) responses, and multiple failure modes. All methods in this research are developed with the aim of addressing problems that have limited data due to high cost of computation or experiment, e.g. vehicle crashworthiness, fluid-structure interaction etc.

The core idea of this research is to construct an explicit boundary separating allowable and unallowable behaviors, based on classification information of responses instead of their actual values. As a result, the proposed method is naturally suited to handle discontinuities and binary states. A machine learning technique referred to as support vector machines (SVMs) is used to construct the explicit boundaries. SVM boundaries can be highly nonlinear, which allows one to use a single SVM for representing multiple failure modes.

One of the major concerns in the design and uncertainty quantification communities is to reduce computational costs. To address this issue, several adaptive sampling methods have been developed as part of this dissertation. Specific sampling methods have been developed for reliability assessment, deterministic optimization, and reliability-based design optimization. Adaptive sampling allows the construction of accurate SVMs with limited samples. However, like any approximation method, construction of SVM is subject to errors. A new method to quantify the prediction error of SVMs, based on probabilistic support vector

machines (PSVMs) is also developed. It is used to provide a relatively conservative probability of failure to mitigate some of the adverse effects of an inaccurate SVM. In the context of reliability assessment, the proposed method is presented for uncertainties represented by random variables as well as spatially varying random fields.

In order to validate the developed methods, analytical problems with known solutions are used. In addition, the approach is applied to some application problems, such as structural impact and tolerance optimization, to demonstrate its strengths in the context of discontinuous responses and multiple failure modes.

CHAPTER 1

INTRODUCTION

Optimization and uncertainty quantification, the main subjects of this dissertation, are of great importance while designing engineering systems. Design optimization refers to the process of selecting the best design alternative among various possible configurations. The definition of "best" alternative is subject to the designer's specifications. For example, it may refer to the lowest cost with certain minimum performance or high system performance with specified maximum cost. It is naturally desirable to have the best design alternative, which is why optimization is essential for any system. However, it is not sufficient to merely provide the best design alternative without considering the effect of uncertainties on its performance. Optimization is intended to minimize the amount of required resources while achieving certain performance from the system. This implies that the optimum design often lies at the boundary of the allowable design space, with a system performance equal to the limiting value. Therefore, even a slight variation of the design or the loading conditions can lead to unacceptable system performance or failure. As a result, such a design is not reliable and such behavior is highly undesirable. Safety factors are used traditionally to prevent failure. They are however chosen arbitrarily, and may not be sufficient or may lead to an overconservative design. To avoid such scenario, it is important to analyze the sensitivity of the design to uncertainties. The process of quantifying the reliability of a system, subject to uncertainties, is referred to as reliability assessment. The process of performing design optimization, while maintaining certain target level of reliability, is referred to as reliability-based design optimization (RBDO).

While the importance of optimization and reliability assessment is well understood, it may not always be straightforward to implement these for complex systems.

Especially, optimization under uncertainties can be quite involved, and is prohibitive in many cases. For complex engineering systems, the relation between system responses (e.g. stress, displacement etc.) and design/loading parameters (e.g. load, Young's Modulus etc.) is seldom available explicitly. For example, it is not possible to express the response of a vehicle subjected to impact in the form of analytical equations. In the absence of an explicit relationship, the most brute force optimization method is to generate a very large number of samples (design configurations) and pick the best solution after evaluating each sample. Similarly the most basic reliability assessment technique is Monte Carlo Simulation (MCS) (Metropolis and Ulam (1949); Melchers, R.E. (1999)), which also requires the generation of a very large number of random samples that follow some probability density function. Suppose the system under consideration is a vehicle subjected to impact; it is definitely not possible to perform a large number of crash tests on actual vehicles. The cost of such a procedure will be extremely high. To avoid tests on actual expensive systems, computer simulations have become popular in many fields. However, they can also involve high cost in the form of resource allocation and computation time. For example, let us consider a finite element model of a vehicle subjected to impact that takes 2 hours per simulation (in general it may be much more). Let us consider seven variables for optimization. To study the effect of each variable, let us consider 5 values per variable. The total number of design configurations will be 5^7 and the total computation time will be 2×5^7 hours, which is almost 18 years. If the probability of failure is calculated for each design configuration then the total time will be many times more. Such methods are definitely not practical. Therefore, the main focus of the design optimization and uncertainty quantification community is to develop efficient and accurate optimization and reliability assessment methods. This is also the subject of this dissertation. Some of the challenges that need resolution are:

- Computation cost per simulation may be high, thus restricting the number of samples.

- Boundary of allowable design space (limit state function or constraint) may be highly nonlinear, and difficult to approximate accurately with limited data.
- Responses can be non-smooth and discontinuous (Missoum, S. et al. (2007)).
- Only binary pass or fail information may be available (Layman, R. et al. (2010)).
- A system may be subject to multiple failure modes, making the process of identifying the failure domain more challenging (Arenbeck, H. et al. (2010)).

Several optimization and reliability assessment methods can be found in the literature. However, no single method presents a solution to all the aforementioned issues faced together.

Optimization methods can be of different types, such as heuristic methods or gradient-based methods. Several types of methods have been developed in each category. The heuristic methods, such as genetic algorithms (GAs), pattern search etc. (Weise (2009); Goldberg (1989); Audet et al. (2000)) are usually zero order methods that require only function values at the samples. Such methods are suitable for non-smooth and discontinuous responses. However, these methods can often require a large number of samples to find the optimum. Therefore, the associated cost is very high if the evaluation of system responses is expensive. Gradient-based methods (Vanderplaats (1984)) also use first order and sometimes second order information in addition to function values. The use of gradient information helps in expediting the optimization. However, these methods are only applicable to problems that have gradient information. If only zero order information is available then a large number of samples may be used to calculate the gradients using finite differences. Another limitation of gradient-based methods is that they are hampered by discontinuous responses, as the gradient is not defined at the discontinuities. Also, gradient based methods are usually limited to local optima.

In the context of reliability assessment also, several methods can be found in the literature. The simplest of the reliability assessment methods is the mean value method (Cornell (1969)) that calculates the probability of failure based on zero and first order information of the response at the mean configuration. First and second order reliability methods (FORM and SORM) (Haldar, A. and Mahadevan, S. (2000)) involve a search for "most probable point" for failure in the standard normal space. These methods are effective if the limit state function (boundary of the failure domain) is linear or quadratic in the standard normal space; however their accuracy in the general case is not reliable. Other methods, such as advanced mean value method (AMV) (Wu et al. (1990)) and two point nonlinear approximation (TANA) (Wang and Grandhi (1995)) have also been developed to treat nonlinear limit states. However, these methods can also lead to considerable errors in the case of complex multimodal limit state functions, as shown in Bichon, B.J. et al. (2007). Apart from the reliability assessment method, another source of error may lie in the quantification of uncertainties itself. Two types of representations are commonly used - random variables and random fields. For a problem with spatial variability (e.g., sheet metal thickness distribution), uncertainties should be represented with random fields as they provide a more realistic representation. However, literature dealing with random fields is relatively limited, and revolves around stochastic finite elements (SFE) (Ghanem and Spanos (2003); Stefanou (2009)). One of the major issues with SFE is that most methods are intrusive, and their implementation is, therefore, complicated. Non-intrusive SFE methods have also been developed (Ghiocel and Ghanem (2002); Berveiller et al. (2006); Huang et al. (2007)). However, one of the limitations of majority of these works lies in the assumption of a prior distribution for the random fields.

In all the optimization and reliability assessment methods, there is a compromise between the computation cost and their accuracy in the general case. A common approach to reduce computation costs is to replace the actual responses using a surrogate model, such as a response surface (Downing et al. (1985); Myers, R.H.

and Montgomery, D.C. (2002)) or metamodel (Wang and Shan (2007); Simpson, T. W. et al. (2008)). A surrogate model provides an approximation of the actual responses, and can be evaluated with a much lower (almost negligible compared to an actual evaluation) computation cost. However, in order to provide an accurate approximation of the responses, a surrogate needs to be "trained" first. This is achieved by first studying the actual system responses at specific samples or configurations in the space. The selection of samples themselves is a broad research area, referred to as design of experiments (DOE) (Montgomery, D.C. (2005); Kleijnen,J.P.C. et al. (2005); Liefvendahl, M. and Stocki, R. (2006); Kleijnen,J.P.C. (2008)). In classical DOE methods (Montgomery, D.C. (2005)), samples were selected based on an initial assumption about the nature of response approximation (e.g. second order polynomial). More recently, space filling designs and adaptive designs have gained prominence for design of computer experiments (Fang et al. (2006); Kleijnen,J.P.C. et al. (2005); Kleijnen,J.P.C. (2008)). Once the samples are selected, the surrogate is trained based on the response values at these samples. There are several choices for surrogate models that can be used. More details about different DOEs and surrogates are presented in Chapter 2. Once a surrogate model is trained, several different methods can be applied for optimization and reliability assessment. Several surrogate-based adaptive sampling techniques, specifically directed at optimization (Jones, D.R. et al. (1998)), reliability assessment (Wang, G.G. et al. (2005); Bichon, B.J. et al. (2007)) and RBDO (Bichon, B.J. et al. (2009)) have also been developed to accurately approximate nonlinear responses with limited samples. However, these methods also face problems when all the aforementioned issues are present together, especially discontinuous and binary responses, and multiple failure modes.

1.1 Motivation of present work

As mentioned in previous section, current optimization and reliability assessment techniques are faced with several challenges that limit their application. The main objective of this dissertation is to develop a general method that addresses these difficulties simultaneously. The work was originally motivated by the need to handle problems with discontinuous responses and highly nonlinear failure boundaries, as no current technique presents an efficient solution to such problems (Missoum, S. et al. (2007); Basudhar, A. et al. (2008)). In this research, such problems are handled by introducing a conceptual shift in the treatment of reliability assessment and constrained optimization problems. Unlike other methods, the proposed sampling-based method referred to as explicit design space decomposition (EDSD) does not require response values at the samples. It only requires binary classification of the samples, i.e. whether a sample is allowable or not. Unlike response approximation methods, only the zero-level contour of the responses is approximated (as opposed to the entire response distribution), which represents the decision boundary (i.e. optimization constraints or failure domain boundary). It is essential to understand this subtle difference, as it is the root cause of several advantages of the proposed method.

While presence of response discontinuities hampers traditional methods, the classification-based EDSD approach greatly simplifies the problem, as it does not require response values. The proposed method, therefore, remains unchanged for discontinuous problems. In the original EDSD method (Missoum et al. (2004); Missoum, S. et al. (2007)), decision boundaries were approximated using hyperplanes, ellipsoids and convex hulls. However, these methods could not be used for highly nonlinear and nonconvex decision boundaries. Therefore, the use of a machine learning technique referred to as support vector machines (SVMs) (Vapnik, V.N. (1998); Shawe-Taylor, J. and Cristianini, N. (2004); Gunn, S.R. (1998)) is proposed in this dissertation to approximate the boundaries. An SVM boundary can be

highly nonlinear, and can also correspond to multiple constraints or failure modes. Thus, the proposed method also provides a natural way to handle multiple failure modes using a single SVM (Arenbeck, H. et al. (2010)). One of the key issues in reliability assessment and optimization is to limit the computation cost. Therefore, in order to construct accurate SVMs with limited number of samples, several adaptive sampling techniques have also been developed as part of this dissertation.

1.2 Scope of work

The scope of this dissertation is the development of the SVM-based EDSD method, for use in reliability assessment, and deterministic or probabilistic optimization (Basudhar, A. et al. (2008)). The motivations for developing this new classification-based technique were discussed in Section 1.1. A major part of the dissertation is dedicated to adaptive sampling techniques. Specific adaptive schemes have been developed for probability of failure calculation, deterministic optimization and RBDO (Basudhar, A. and Missoum, S. (2008, 2010b, 2009a, 2010a)). It is understood that the accuracy of any sampling-based method depends on the quality of the samples, and it is, therefore, important to consider the possibility of errors. Therefore, probability of misclassification due to an incorrect SVM approximation is also considered in this research (Basudhar, A. and Missoum, S. (2010c)). This is performed using a probabilistic support vector machine (PSVM) (Vapnik, V.N. (1998); Wahba, G. (1999); Platt, J.C. (1999)). However, the current PSVM models do not take the spatial distribution of samples into account, resulting in non-intuitive misclassification probabilities at the evaluated samples. Therefore, an improved PSVM model is also developed in this work that accounts for such factors (Basudhar, A. and Missoum, S. (2010a)).

In terms of quantification of uncertainties, both random variable and random field representations are used in this work (Basudhar, A. and Missoum, S. (2009b)).

In the case of random fields, proper orthogonal decomposition (POD) (Liang,Y. C. et al. (2002)) is used to convert the problem into an equivalent random variable problem.

In order to demonstrate the efficacy of the developed methods, several analytical examples with known solutions are used. In addition, application of the approach to solve problems with discontinuous responses and multiple failure modes is also demonstrated. The methods are applied to linear and nonlinear buckling problems, structural impact problems, and tolerance optimization.

Organization of this dissertation is as follows. Chapter 2 presents a review of the literature in the fields of reliability assessment, optimization and RBDO. Chapter 3 provides an introduction to SVMs and PSVMs. In Chapter 4, an introduction to the basic notion of EDSD using SVMs is provided, along with an adaptive sampling to globally refine the SVM boundaries. Examples are presented to demonstrate the accuracy of the update and the application of EDSD to reliability assessment. To enhance the scalability of the approach, an RBDO method using locally refined SVMs is presented in Chapter 5. The RBDO method uses a specific update strategy for calculation of failure probabilities, which is also explained in the same chapter. In Chapter 6, a modified PSVM model developed in this work is presented. The model is used to calculate the probability of misclassification by SVM. This measure of uncertainty in SVM classification is then used to provide a relatively conservative PSVM-based probability of failure that compensates the errors in SVM prediction. Chapter 7 presents a method to perform constrained efficient global optimization using locally refined SVM constraints. The PSVM model introduced in Chapter 6 is used to guide the sample selection for optimization. In Chapter 8, the EDSD method, presented for random variables in earlier chapters, is extended to reliability assessment using random fields. Chapters 4-8 are supported with results of analytical examples with known solutions. In addition, some application problems are presented to demonstrate the usefulness of the methodologies developed in this

research. Finally, in Chapter 9, a summary of the contributions of this dissertation is presented along with a discussion on future scope for research in the field.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

This chapter presents the necessary background to various concepts that are part of this research, as well as a review of the literature. First, the basic concepts of design optimization and reliability assessment are presented. These are are followed by an introduction to the state of the art designs of experiments (DOE) that allow one to perform a study of system responses with respect to design and random variables. A DOE allows one to study system responses at discrete samples. Very often, these responses are used to construct a surrogate model that approximates the actual system responses. As mentioned in Chapter 1, use of surrogates is quite popular for the reduction of computation cost. A surrogate model provides an approximation of the responses that can be used to replace the actual function while performing optimization or reliability assessment. A review of various surrogate models is presented in Section 2.4. However, approximation of responses has its limitations that were the major motivation for the development of classification-based methods, which encompass the research performed in this dissertation. An introduction to classification-based methods is presented in Section 2.5. Finally, a detailed review of various reliability assessment, optimization and RBDO methods is presented in Sections 2.7, 2.6 and 2.8.

2.1 Design optimization

As already stated in Chapter 1, optimization is important in all fields of engineering. Design optimization refers to the process of enhancing system performance with minimal resources, i.e. to select the best design alternative. It is, therefore, naturally

desirable. In general, an optimization problem is defined as:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ s.t. \quad & \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \\ & \mathbf{h}(\mathbf{x}) = \mathbf{0} \\ & \mathbf{x}_{min} \leq \mathbf{x} \leq \mathbf{x}_{max} \end{aligned} \tag{2.1}$$

where $f(\mathbf{x})$ is an objective function that need to be minimized. $\mathbf{g}(\mathbf{x})$ is a vector of inequality constraints and $\mathbf{h}(\mathbf{x})$ is a vector of equality constraints. In this dissertation, the emphasis is on inequality constraints. The process of optimization can be illustrated with an example. Let us consider a two bar truss structure subjected to a point load F, with the area of cross-section A being a design variable (Figure 2.1).

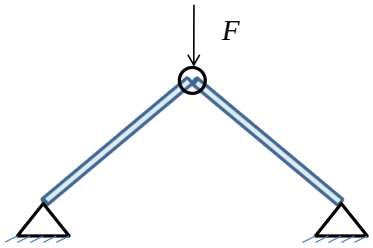

Figure 2.1: Two bar truss subjected to point load.

A design configuration (value of A) is considered allowable or feasible if it does not lead to buckling when subjected to the design load F. Any configuration that leads to buckling is unacceptable. Naturally, a configuration with lower cross-sectional area is more likely to buckle. Suppose the lowest area that does not lead to buckling is A^*. The feasible and infeasible domains of the one-dimensional design space are depicted in Figure 2.2. If the objective function function $f(A)$ is the weight of the structure then the optimum design will be the one with lowest area that does not lead to buckling, i.e. A^* (Figure 2.3).

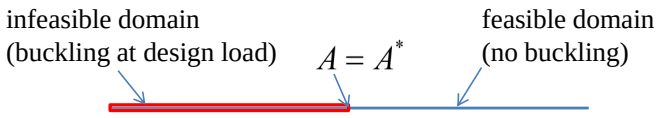

Figure 2.2: One dimensional feasible domain for two bar truss.

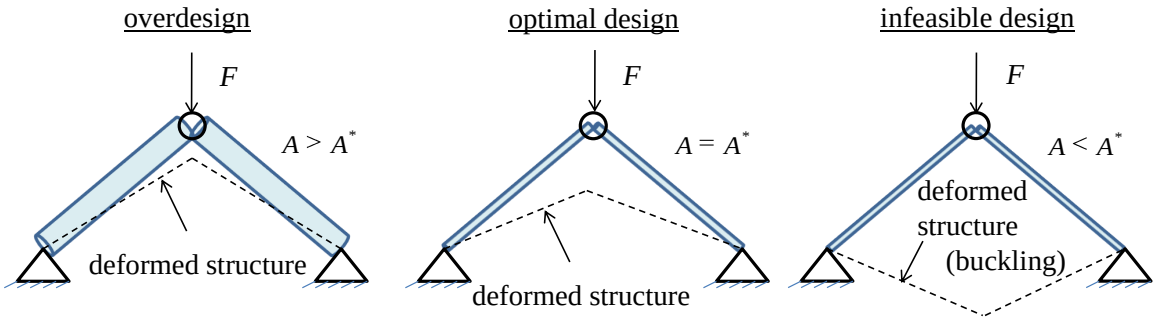

Figure 2.3: Optimal design of two bar truss.

In general, the relations between variables ($\mathbf{x}$) and system responses ($f(\mathbf{x}), \mathbf{g}(\mathbf{x}), \mathbf{h}(\mathbf{x})$) are not available explicitly. Therefore, finding the optimal solution may not be straightforward, especially when function evaluations are expensive. Additional difficulties may be encountered in the form of highly nonlinear system equations, response discontinuities etc. A review of optimization methods is provided in Section 2.6.

2.2 Reliability assessment

This section presents an introduction to the concept of reliability in design. The importance of reliability considerations for engineering systems is well known. A very small variation in the design or loading parameters can sometimes lead to failure. This can be illustrated with the help of the two bar truss example considered in Section 2.1. The load-displacement curve of the structure is shown in Figure 2.4. The left figure shows the point on the curve that corresponds to the optimal design subjected to the design load F. The figure on the right shows the effect of uncertainties on the displacement. Even a slight variation of the applied load can

lead to buckling of the structure.

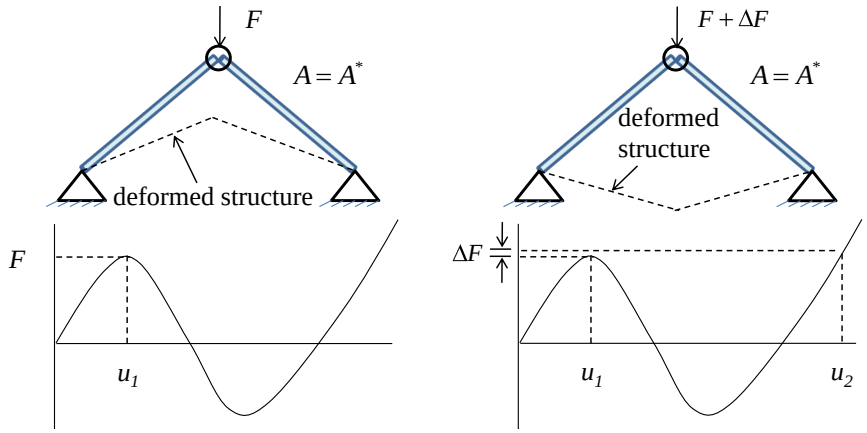

Figure 2.4: Effect of a small load variation on a two bar truss.

Clearly, the two bar truss structure displays high sensitivity to uncertainties. If the effect of uncertainties is not considered, then the resulting design will be extremely unsafe. Therefore, in general, it is imperative to account for uncertainties in loading and design variables.

There are several ways to quantify the reliability of a system. It can be represented in a deterministic way using a response value and a safety factor/margin or a stochastic way using a probability of failure or reliability index (Elishakoff (2004)). The conventional method is to specify a factor or margin of safety. Factor of safety is defined as:

$$SF = \frac{R}{S} \tag{2.2}$$

Safety margin is defined as:

$$SM = R - S \tag{2.3}$$

where R is the system resistance, e.g. yield strength, and S is the load applied to the system. A major limitation of above deterministic representations is that they do not provide a clear insight into the chances of failure or the factors affecting system

responses. Also, designs based on safety factors can often be over-conservative (Elishakoff (2004)). A much more widely accepted measure is the probability of failure. This involves consideration of R and S as random quantities with certain distribution (Figure 2.5). Failure occurs if S exceeds R (shaded region in Figure 2.5). Therefore, probability of failure P_f is:

$$P_f = P(R - S \leq 0) = P(z \leq 0) \tag{2.4}$$

where $z = R - S$ is the limit state function or the performance function. $z = 0$ represents the failure domain boundary.

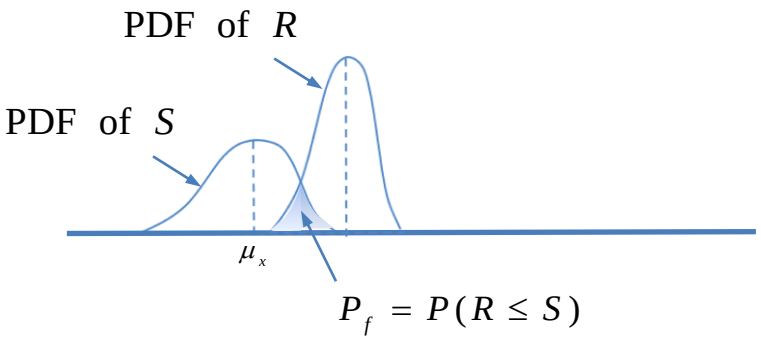

Figure 2.5: Probability of failure calculated as the probability of load S being greater than resistance R.

If the uncertainties in the system are represented using a vector of random variables $\mathbf{x}$ then the probability of failure can be expressed as:

$$P_f = P(g(\mathbf{x}) \leq 0) = \int_{\Omega_f} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \tag{2.5}$$

where $f_{\mathbf{X}}(\mathbf{x})$ is the joint probability density function of random variables x_1 up to x_m and Ω_f is the failure domain defined by the limit state function $g(\mathbf{x})$. By convention, the limit state function $g(\mathbf{x})$ is negative in failure region. The probability of failure is calculated as probability of $\mathbf{x}$ lying in the failure region (Figure 2.6). Unlike the safety factor and margin measures, probability of failure provides more insight

about the likeliness of failure. Systems can be designed based on specific target probabilities.

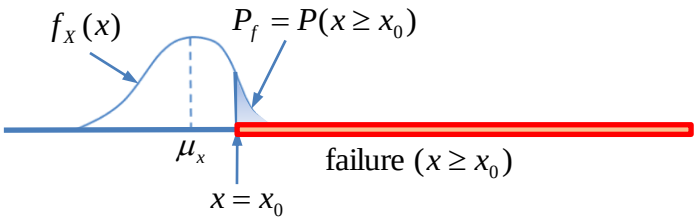

Figure 2.6: Failure probability due to uncertainty in random variable x.

The calculation of failure probability using Equation 2.5 is not always straightforward as the distribution of $g(\mathbf{x})$ or the boundary of failure domain Ω_f are not known explicitly. Very often, the only information available is the value of responses at discrete samples. Therefore, approximations are required to calculate the integral in Equation 2.5. The calculation of failure probability is especially challenging when the evaluation of samples is expensive. Therefore, the goal of most reliability assessment methods is to evaluate probabilities of failure with limited samples. A review of reliability assessment methods is presented in Section 2.7.

Similar to optimization, in the context of reliability assessment also, it is possible to replace the actual function with a surrogate to reduce computation cost. Surrogate models are constructed by studying the responses using a design of experiments (DOE). A review of DOEs and surrogate modeling is provided in following sections.

2.3 Design of experiments (DOE)

This section presents an introduction to designs of experiments (DOEs) (Montgomery, D.C. (2005); Kleijnen,J.P.C. et al. (2005); Kleijnen,J.P.C. (2008)), which allow one to study system responses based on a set of discrete samples. The method of selecting the samples is referred to as a design of experiments. Several types of DOEs exist with varying criteria for the selection of samples. The ultimate goal of a good DOE is to maximize the available information with limited samples.

The most basic and intuitive design of experiment is a full factorial design, in which each variable is divided into specified number of levels. Each possible combination of the variables is then treated as a sample. An example of a two variable full factorial design, with 3 levels per variable, is shown in Figure 2.7.

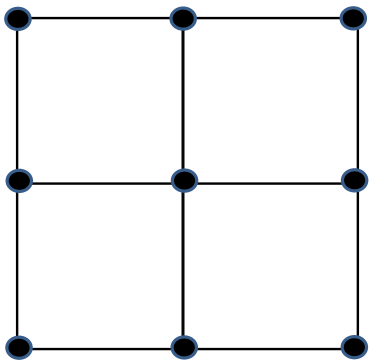

Figure 2.7: Full factorial design with 2 variables and 3 levels.

Although a full factorial design allows an exhaustive study with respect to to all the combinations of the different variable levels, the number of levels that can be used is limited. The method is not scalable to high dimensions because the number of samples will be very high. Fractional factorial designs address the issue to certain extent; however, these also require set levels for each variable.

In the context of response surfaces, non-factorial designs, such as central composite design ((Montgomery, D.C. (2005))) are useful. An example of CCD is shown in Figure 2.8. It consists of a center point, 2^m corner points and $2m$ axial points, m being the dimensionality. CCD allows the calculation of curvatures without evaluating a three level full factorial design, and is especially suited for second order approximation.

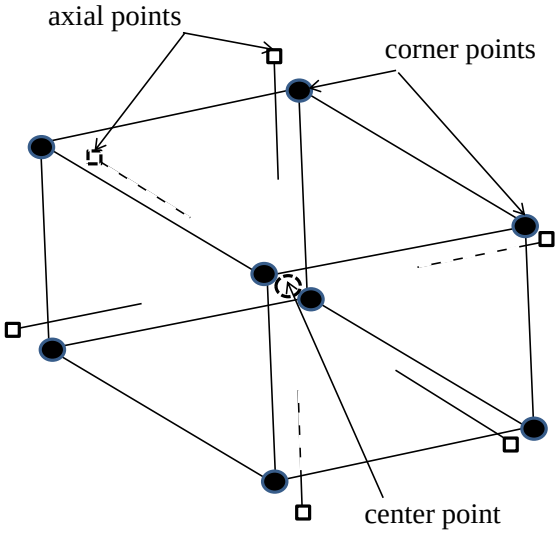

Figure 2.8: Center composite design with three variables.

More recent methods include space filling design, such as Latin Hypercube Sampling (LHS) (Butler, A. N. (2001)) and more uniform designs, such as Improved Distributed Hypercube Sampling (IHS) (Beachkofski, B.K. and Grandhi, R. (2002)), Optimal Latin Hypercube Sampling (OLHS) (Liefvendahl, M. and Stocki, R. (2006)), Centroidal Voronoi Tessellations (CVT)) (Romero, Vincente J. et al. (2006)) etc. Low discrepancy sequences such as Halton, Hammersely and Sobol are also used. An example of LHS is shown in Figure 2.9. The basic idea is that no two samples can have the same value or level for a particular variable. Thus it requires much fewer samples compared to factorial designs, and are more scalable. The number of samples is equal to the number of levels per variable.

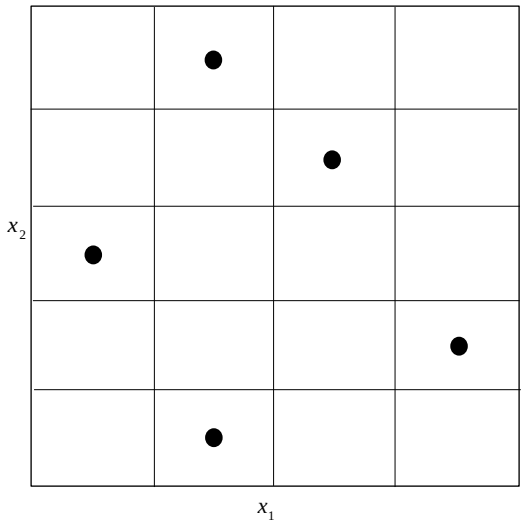

Figure 2.9: Latin hypercube sampling with two variables and five levels.

Although "latinization" of the samples has its benefits, there are several possibilities to construct an LHS. Some of these designs may not have uniform distribution of samples. The goal of OLHS is to provide an optimal LHS based on certain criterion. The criterion may be based on maximum minimum distance or minimum "potential energy" (Liefvendahl, M. and Stocki, R. (2006)). These criteria allow for a more uniform distribution of the samples.

Centroidal Voronoi Tessellations (CVT) also provide uniform sampling of the space. In CVT samples are placed such that they lie at the centroids of the respective Voronoi cells. The Voronoi cell corresponding to a sample is defined as the region where this sample is the closest one to any point within the cell. A uniform sample distribution is obtained when the samples coincide with the cell centroids. An example of CVT is shown in Figure 2.10. Although CVT gives a uniform sample distribution, it has limitations in high dimensions. In a high dimensional hypercube encompassing the DOE, most of the volume is contained in the corners. It may be useful to sample within a hypersphere, especially in the context of reliability

assessment Jiang, P. et al. (2011).

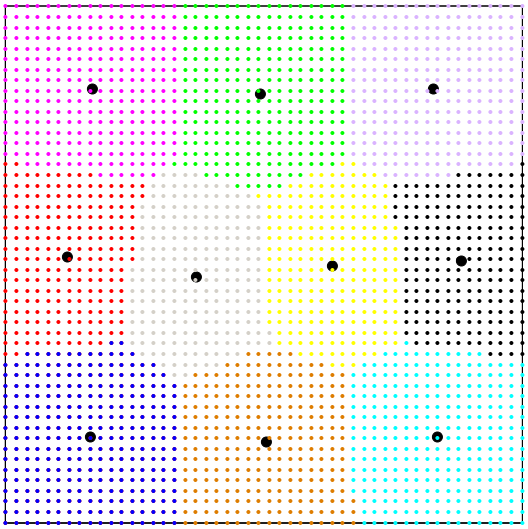

Figure 2.10: Centroidal Voronoi Tessellations with two variables and ten samples. Samples are represented by the black circles, and the corresponding Voronoi cells are shown in different colors.

2.4 Response approximation using surrogate models

In the context of expensive function evaluations, the focus of this dissertation, there are limitations on the number of function calls. In order to reduce function evaluation cost, the actual response is often replaced by a surrogate model, such as a response surface (Downing et al. (1985); Myers, R.H. and Montgomery, D.C. (2002)) or metamodel (Wang and Shan (2007); Simpson, T. W. et al. (2008)). Followed by the construction of a surrogate model with relatively few samples, the response at any sample can be evaluated using the approximated model.

2.4.1 Basic response approximation methodology

The basic steps for construction of a surrogate model are presented in this section. Once the surrogate is constructed, an approximation of the response function is obtained. Therefore, calculation of response at any sample is straightforward using the approximated function. The actual function can be replaced with the surrogate for performing optimization or reliability assessment. The key steps are:

- *Design of Experiments (DOE):* First, the space is sampled using a specified number samples or configurations. These samples are selected using a DOE (Montgomery, D.C. (2005); Kleijnen,J.P.C. et al. (2005); Kleijnen,J.P.C. (2008)).

- *Response Evaluation at DOE samples:* For each sample in the DOE, the system response is obtained using the actual function evaluator (e.g. finite element analysis (FEA) code). In the context of engineering applications, each function evaluation may be quite expensive, e.g. for crash analysis or fluid-structure interaction problems.

- *Choice of a surrogate model and calculation of unknown coefficients of the model:* There are several choices for a surrogate model (Section 2.4.3). Any model consists of a particular basis and a set of coefficients that need to be determined. These coefficients are determined using the information at DOE samples. Substitution of evaluated response at each sample into the expression for approximated function (e.g. Equations 2.8,2.9,2.12,2.14) provides an equation. This information is used to calculate the unknown coefficients. Different techniques exist to calculate the coefficients, depending on the type of surrogate (Section 2.4.2).

- *Prediction of unknown responses:* Once the coefficients are calculated, an approximation of the response function is obtained. Therefore, calculation of response at any sample is straightforward using the approximated function.

The surrogate model can then be used to replace the actual function while performing optimization or reliability assessment.

2.4.2 Calculation of surrogate model coefficients for response approximation

Any surrogate model consists of a basis and a set of unknown coefficients that need to be determined. There are various methods to determine the coefficients. Some of the methods are listed below.

- *Solution of a system of linear equations.* This method is used when the DOE is saturated, i.e. the number of samples is exactly same as the number of unknown coefficients in the model. Response evaluation at each sample gives an equation, and the system of equations using all samples is solved.

- *Least square and moving least square.* Determining the coefficients by solving a system of equations may lead to overfitting. A commonly used approach for finding the coefficients is to minimize the sum of square errors at the samples. The square error is:

$$\epsilon^2 = \sum_{j=1}^{N} \left(\hat{f}(\mathbf{x}_j) - f(\mathbf{x}_j) \right)^2 \tag{2.6}$$

  where $f(\mathbf{x})$ is the actual function and $\hat{f}(\mathbf{x})$ is the approximated one. The least square method gives equal weights to all the samples. Another approach is to give varying weights to the samples. In moving least square method, varying weights are given to the samples, with the weights depending on the distance to the point at which response is required. The moving least square is:

$$\epsilon^2 = \sum_{j=1}^{N} \left(\hat{f}(\mathbf{x}_j) - f(\mathbf{x}_j) \right)^2 w\left(\mathbf{x} - \mathbf{x}_j\right) \tag{2.7}$$

  where $\mathbf{x}$ is the point at which response is required. The weight w is higher for samples that are closer to $\mathbf{x}$.

- *Maximum likelihood approximation.* Maximum likelihood method is used for determining the coefficients in the context of probabilistic approaches, such as

Kriging. It is based on maximizing a known likelihood function expressed in terms of the surrogate model coefficients. Knowing the response outcomes at the samples, the likelihood of having a set of coefficients is defined. The likelihood is a function of the coefficient values. The maximum likelihood method is illustrated in the following section in the context of a Kriging surrogate.

2.4.3 Types of surrogate models

Several types of surrogate models can be found in the literature. A review of some of the common methods is provided in this section.

Polynomial response surface

The most basic method of approximating responses is to fit a polynomial response surface (Box and Wilson (1951)) to the function values:

$$\hat{f}(\mathbf{x}) = \alpha_0 + \alpha_1 x_1 + \ldots + \alpha_m x_m + \alpha_{m+1} x_1^2 + \ldots + \alpha_{2m} x_m^2 + \ldots \tag{2.8}$$

where α_i are unknown polynomial coefficients that are solved to obtain the approximation. Depending on the DOE used, the polynomial coefficients can be determined by solving a system of linear equations or based on a least square or moving least square type criterion. The use of second order polynomial response surface was first proposed in Box and Wilson (1951). A second order polynomial allows the calculation of first and second derivatives, and therefore, can be used to find the minima or maxima for optimization. However, in most situations, a second order approximation is not adequate as the relation between design variables and system responses may be highly nonlinear. Higher order polynomials may also be used, but they require more samples. Also the approximation is highly dependent on the degree of the polynomial.

Radial basis functions

A more flexible model to approximate responses consists of using radial basis functions instead of polynomials (Powell (1987)). Radial basis function is a function for which the value depends on the distance from the center. The nature of approximation depends on the number of basis centers used. The method of finding the coefficients for an RBF approximation depends on the number of basis centers. If all samples are used as basis centers then function values are interpolated, i.e., the approximation passes through the evaluated function value at each sample:

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i \psi_i(||\mathbf{x} - \mathbf{x}_i||) \tag{2.9}$$

where $\psi_i(\mathbf{x} - \mathbf{x}_i)$ is a radial basis function, α_i is the coefficient or weight for i^{th} basis function, and N is the number of samples. The RBF can have several forms, such as Gaussian, multiquadratic etc. For instance, Gaussian RBF is defined as $\exp(-\frac{||\mathbf{x}-\mathbf{x}_i||^2}{2\sigma^2})$. In Equation 2.9, there are N unknowns (α_i). N equations are obtained using function values at the samples:

$$\hat{f}(\mathbf{x}_j) = \sum_{i=1}^{N} \alpha_i \psi_i(||\mathbf{x}_j - \mathbf{x}_i||) \quad j = 1, 2, \ldots, N \tag{2.10}$$

This leads to a system of equations with N unknowns and N equations. The coefficients α_i are obtained as:

$$\boldsymbol{\alpha} = \boldsymbol{\Psi}^{-1}\mathbf{f} \tag{2.11}$$

where $\boldsymbol{\alpha}$ is the vector of unknown coefficients, $\mathbf{f}$ is the vector of function values at the samples, and $\boldsymbol{\Psi}$ is the matrix of radial basis functions calculated at the samples.

If all samples are used to calculate the RBF weights, it may lead to overfitting. This may cause problem if there is noise in the data. It is possible to train RBFs in a least square sense. In this case, only a subset n of the samples is used as basis centers. The approximation is given as:

$$\hat{f}(\mathbf{x}_j) = \sum_{i=1}^{n} \alpha_i \psi_i(||\mathbf{x}_j - \mathbf{x}_i||) \quad j = 1, 2, \ldots, N \tag{2.12}$$

The weights are calculated such that they minimize the sum of square errors ϵ^2 at the samples:

$$\epsilon^2 = \sum_{j=1}^{N} \left(\hat{f}(\mathbf{x}_j) - f(\mathbf{x}_j) \right)^2 = \sum_{j=1}^{N} \left(\sum_{i=1}^{n} \alpha_i \psi_i(||\mathbf{x}_j - \mathbf{x}_i||) - f(\mathbf{x}_j) \right)^2 \tag{2.13}$$

Kriging

Another popular method for response approximation is Kriging, which models system response as a random process:

$$\hat{f}(\mathbf{x}) = \mathbf{h}(\mathbf{x})^T \boldsymbol{\beta} + Z(\mathbf{x}) \tag{2.14}$$

where $\mathbf{h}$ is the trend of the model, $\boldsymbol{\beta}$ is the vector of trend coefficients, and Z is a stationary Gaussian process based on correlation between samples. The covariance between any two samples $\mathbf{a}$ and $\mathbf{b}$ is defined as:

$$cov[Z(\mathbf{a}), Z(\mathbf{b})] = \sigma_Z^2 R(\mathbf{a}, \mathbf{b}) \tag{2.15}$$

where σ_Z^2 is the variance of the process Z and R is the correlation function:

$$R(\mathbf{a}, \mathbf{b}) = e^{-\sum_{j=1}^{m} \theta_j |a_j - b_j|^{p_j}} \tag{2.16}$$

where θ_j is the scale parameter for the j^{th} dimension and the parameter p_j determines the smoothness of the correlation function and is set equal 2 for Gaussian correlation.

In ordinary Kriging the regression terms in Equation 2.14 are replaced by an unknown constant trend. It has been reported in literature that the correlation term itself is powerful enough to provide an approximation of the responses. The response at any point $\mathbf{x}$ is:

$$\hat{f}(\mathbf{x}) = \mu + Z(\mathbf{x}) \tag{2.17}$$

It should be noted that μ is an unknown that depends on the correlation between $\mathbf{x}$ and the evaluated samples, and needs to be solved. The total number of unknowns is $2m + 2$. For each of the m variables, θ_j and p_j are unknown. In addition, μ and σ_Z are also unknown. The unknowns in Kriging are are solved using maximum likelihood constructed based on the known actual function values and correlations for N evaluated samples. Because the model is considered as a correlated Gaussian process, the likelihood function is given as:

$$L(\mu, \sigma_Z, \boldsymbol{\theta}, \mathbf{p}) = \frac{1}{(2\pi)^{\frac{N}{2}}(\sigma_Z^2)^{\frac{N}{2}}|\mathbf{R}|^{\frac{1}{2}}} \exp\left[-\frac{(\mathbf{f} - \mathbf{1}\mu)'\mathbf{R}^{-1}(\mathbf{f} - \mathbf{1}\mu)}{2\sigma_Z^2}\right] \tag{2.18}$$

where $\mathbf{f}$ is the vector of actual function value at N evaluated samples and $\mathbf{R}$ is a $N \times N$ matrix containing the correlation function values between each sample pair. It should be noted that the likelihood function is given by the multivariate normal distribution, expressed as a function of the unknown distribution parameters. The values of μ and σ_Z that maximize the likelihood function can be obtained by simple differentiation, in terms of the other unknowns. These are given as:

$$\hat{\mu} = \frac{\mathbf{1}'\mathbf{R}^{-1}\mathbf{f}}{\mathbf{1}'\mathbf{R}^{-1}\mathbf{1}} \tag{2.19}$$

$$\hat{\sigma_Z}^2 = \frac{(\mathbf{f} - \mathbf{1}\hat{\mu})'\mathbf{R}^{-1}(\mathbf{f} - \mathbf{1}\hat{\mu})}{N} \tag{2.20}$$

Substituting the results from Equations 2.19 and 2.20 into Equation 2.18 provides the likelihood function as a function of the other $2m$ unknowns. These unknowns are determined using optimization, by maximizing the likelihood function. Similar to RBFs, Kriging has the ability to approximate highly nonlinear responses provided sufficient training data is present. In addition it provides a measure of estimation error by the surrogate. The mean Kriging prediction at a sample $\mathbf{x}$ is (Jones, D.R. et al. (1998)):

$$\hat{f}(\mathbf{x}) = \hat{\mu} + \mathbf{r}'\mathbf{R}^{-1}(\mathbf{f} - \mathbf{1}\hat{\mu}) \tag{2.21}$$

where $\mathbf{r}$ is a vector containing the correlation function values between $\mathbf{x}$ and the N training samples. The mean square error or variance of the Kriging prediction is

(Jones, D.R. et al. (1998)):

$$s^2(\mathbf{x}) = \sigma_Z^2 \left[1 - \mathbf{r}'\mathbf{R}^{-1}\mathbf{r} + \frac{(1 - \mathbf{1}'\mathbf{R}^{-1}\mathbf{r})^2}{\mathbf{1}'\mathbf{R}^{-1}\mathbf{1}} \right] \tag{2.22}$$

2.5 Response classification methods

The basic notions of design optimization and reliability assessment were presented in Sections 2.1 and 2.2. It was stated that the constraint functions and limit state function $g(\mathbf{x})$ are not known explicitly in general, and surrogate models are often used to approximate the function. However, it is interesting to note that in both optimization and reliability assessment, it is not actually the function $g(\mathbf{x})$ that is required. Only the zero-level contour of the function, $g(\mathbf{x}) = 0$, is required to define the decision boundaries. This is the central idea of classification-based methods presented in this section, and also of this dissertation. An approximation is built for the zero-level contour $g(\mathbf{x}) = 0$, which is then used to replace the actual decision boundary. Similar to response approximation methods, the system responses are first studied with discrete samples from a DOE. However, instead of fitting the responses, the samples are classified as allowable or not. An explicit boundary is then constructed that separates the two classes of samples. Optimization and reliability assessment can then be performed using the approximated classification boundary. A comparison of response approximation and classification methods is shown in Figure 2.11.

The first attempt to classify the space into safe and failure regions was made in (Missoum et al. (2004)). Hyperplanes and ellipsoids were used to approximate the failure domain. An improved version of the method was developed using convex hulls (Missoum, S. et al. (2007)). However, all these methods were limited to decision boundaries that represent convex domains. This limitation has been overcome in this dissertation by using Support Vector Machines (SVMs) (Vapnik, V.N. (1998); Gunn, S.R. (1998)) for decision boundary approximation. A comparison of the classification methods is shown in Figure 2.12.

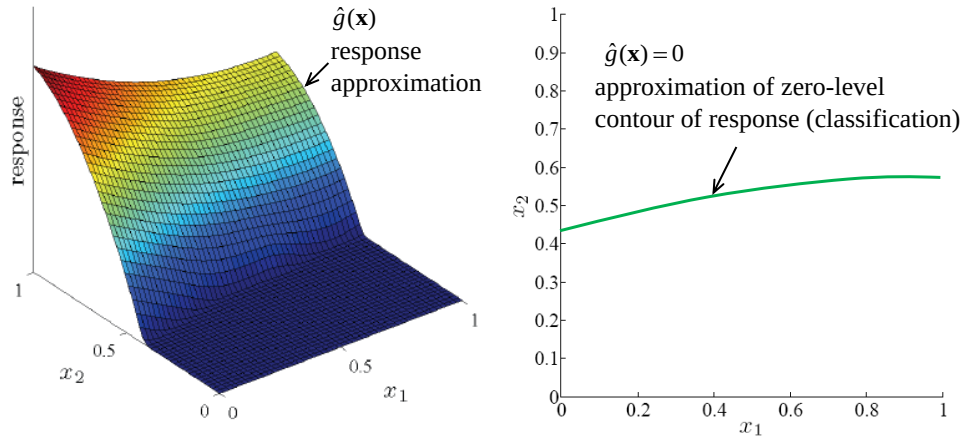

Figure 2.11: Comparison of response approximation and classification methods.

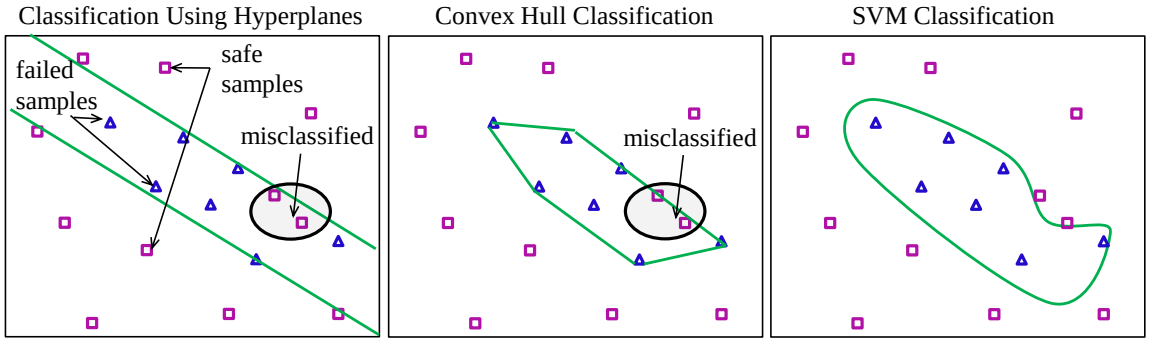

Figure 2.12: Classification using hyperplanes, convex hull and SVM. Among the three methods, only SVM is able to classify all samples correctly due to non-convex nature of the limit state function.

The use of SVMs in the context of reliability assessment was also independently proposed in Hurtado, Jorge E. (2004). In this dissertation, SVMs are used for reliability assessment as well as deterministic and probabilistic optimization, as explained in Chapter 5 and Chapter 7. Contribution of this dissertation also includes the development of several adaptive sampling techniques to refine SVM boundaries. A new method to quantify the prediction error of SVMs (Chapter 6) is also developed, which is used to provide relatively conservative failure probabilities

compared to a deterministic SVM.

2.6 Deterministic optimization methods

An introduction to the necessity of optimization and some of the difficulties encountered was provided in Section 2.1. This section provides a review of some of the optimization methods.

2.6.1 Gradient-based methods

Gradient-based methods are a category of optimization techniques that use the derivatives of responses for determining the search path. They are derived from the basic principles of Calculus. The most basic method is steepest descent method. It is known that value of a function reduces in a direction for which gradient is negative. The steepest descent method aims at finding the optimum by iteratively searching the direction with lowest gradient:

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} - \alpha \nabla f(\mathbf{x}^{(k)}) \tag{2.23}$$

where $\nabla f(\mathbf{x}^{(k)})$ is the search direction and α is a scalar that is adjusted to find minimum $f(\mathbf{x})$ in this direction. For constrained optimization, the objective function $f(\mathbf{x})$ in Equation 2.23 is replaced by a penalized merit function (Vanderplaats (1984)). More advanced methods exist, such as sequential linear programming (SLP) and sequential quadratic programming (SQP) (Boggs and Tolle (1995); Vanderplaats (1984)). In these methods, a series of linear or quadratic programming subproblems are solved to find the final solution.

In SLP, both the objective function and the constraints are linearized at the current iterate using Taylor series expansion. The next iterate is constrained to lie within the move limits of each variable, within which the linear approximation is

considered valid. The optimization subproblem is:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T(\mathbf{x} - \mathbf{x}_k) \\ s.t. \quad & g_j(\mathbf{x}_k) + \nabla g_j(\mathbf{x}_k)^T(\mathbf{x} - \mathbf{x}_k) \leq 0 \\ & x_i^{(l)} \leq x_i - x_{ki} \leq x_i^{(u)} \end{aligned} \tag{2.24}$$

where $\mathbf{x}_k$ is the current iterate, g_j is the j^{th} inequality constraint, and $x_i^{(l)}$ and $x_i^{(u)}$ are the lower and upper move limits for the i^{th} variable.

In SQP, the optimization subproblem to find the search direction $\mathbf{s}$ consists of a quadratic objective function and linear constraints:

$$\begin{aligned} \min_{\mathbf{s}} \quad & f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T\mathbf{s} + \frac{1}{2}\mathbf{s}^T\mathbf{H}\mathbf{s} \\ s.t. \quad & g_j(\mathbf{x}_k) + \nabla g_j(\mathbf{x}_k)^T\mathbf{s} \leq 0 \end{aligned} \tag{2.25}$$

where $\mathbf{H}$ is the Hessian matrix. If the Hessian is not available directly, it may require several function evaluations to determine it. To avoid that, an initial approximation of $\mathbf{H}$ is updated iteratively usually (Vanderplaats (1984)). Determination of the search direction is followed by a line search along that direction.

Any gradient-based method is prone to certain limitations:

- Gradient based methods often converge to local minima. Multiple starting points may be used to overcome this issue, but it increases the computation cost.
- Gradient-based methods are hampered by discontinuous and non-differentiable responses.

2.6.2 Heuristic methods

Unlike gradient-based methods, these methods are based on heuristics rather than solely the mathematical foundation of Calculus. Several types of algorithms

have been developed, which try to balance exploration (of the whole space) and exploitation (of a local region in space expected to contain optimum) in a specific manner. Some of the popular heuristic methods are generalized pattern search, Genetic Algorithms, Simulated Anealing (SA), and Particle Swarm Optimization (PSO) (Weise (2009)).

Generalized pattern search (GPS) methods (Audet et al. (2000)) are derived from coordinate search methods, and involve the definition of a positive spanning set of search directions. In other words, a positive linear combination of the search directions spans the space. The search in GPS is based on a "global search" step and a "local poll" step. An initial starting point is perturbed along each search direction by an initial step size. If a lower objective function is found then the step size is increased and a global search is started at the new point. Otherwise, the step size is reduced and a search is started at the previous point. Constraints are handled using pruning; objective function at infeasible points is set to infinity.

A popular class of heuristic methods are genetic algorithms (GAs) (Goldberg (1989); Weise (2009)). GAs are evolutionary algorithms that try to simulate evolution of genes in nature. Similar to natural gene evolution, GAs are stochastic in nature. They start with an initial population of consisting of individuals, each individual representing a sample in the context of optimization. Next generations of samples are created using three important operations - selection (among existing individuals), crossover (between selected samples from current population) and mutation:

- *Selection:* The first step in creating the next generation from the current population is to select individuals for "breeding". For this purpose, a fitness function is evaluated for each individual of the current population. The fitness values are then used to select individuals based on different criteria, such as tournament selection, roulette wheel selection, ranking methods etc.

Roulette wheel selection is one of the most intuitive methods. In this method, a probability of selection is assigned to the individuals, which is directly proportional to the fitness function value. This probability is obtained by dividing the fitness value of an individual by the sum of all fitness values in the population. The population is first sorted in decreasing order of this probability. A random number is then generated between 0 and 1. The selected individual is the first one with an accumulated probability greater than this number. The accumulated probability is the sum of probabilities for this individual and all previous individuals in the sorted list. This process is repeated until the required number of individuals for the next generation are obtained.

- *Crossover:* In crossover, two parents or individuals from the current population are combined to create children or individuals for the next generation. There are several methods, such as single point crossover, cut and splice crossover, uniform crossover etc. Before performing crossover, the individuals may be converted to their binary codes, i.e. expressed in terms of 0 and 1 (although it is not necessary to do so). For example, the binary code for 5 is 101. An example of single point crossover with binary coding of individuals is shown in Figure 2.13. In uniform crossover, individual bits of the parents are compared. They are swapped with a fixed probability. This probability is a parameter that needs to be defined by the user. A random number is generated between 0 and 1, and the bits are swapped if this number is less than the predefined probability value.

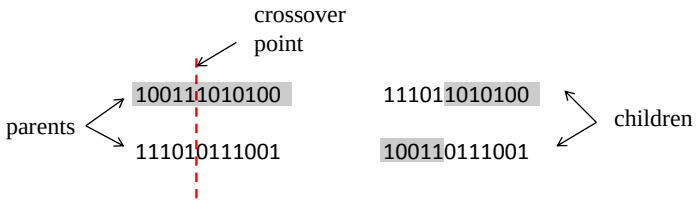

Figure 2.13: Single point crossover of binary coded individuals.

- *Mutation:* Mutation is performed to maintain diversity in the population. A small probability is provided that the children obtained after crossover may mutate. This is done so that all individuals in the population are not exactly the same. There are several methods for mutation, such as flip bit mutation, uniform mutation, gaussian mutation etc. For example, in flip bit mutation used in the case of binary coding, a bit may be flipped based on a predefined probability (usually low). A random number is generated, and the bit is flipped (from 0 to 1 or from 1 to 0) if this value is less than the predefined probability.

Apart from the above operations, specified number of best individuals may be designated as elite members for each generation. Elite members are automatically retained in the next generation without crossover or mutation. As the optimization progresses, the population converges to regions with minimum objective function value. In the case of constrained optimization, a penalized objective function is used to define the fitness function.

The use of genetic algorithms has several advantages. Because they require only the function values, and not their derivatives, GAs can handle discontinuities. Also, they can be used for both continuous and discrete variables. In addition, individuals of the population can be evaluated in parallel to reduce the computational cost. GA is a stochastic method, and provides a good possibility of finding the global optimum when several local optima are present. However, as a result of the randomness, different executions of the same problem may sometimes provide different solutions.

The GPS and GA methods reviewed above, as well as methods such as SA and PSO follow certain heuristics to balance exploration and exploitation. Although convergence to global optimum is not guaranteed, these methods are likely to find the global solution. However, all these methods face certain issues that limit their scope of application:

- Parameters of the optimization need to be fine tuned for efficient performance of these algorithms. Fine tuning the parameters is problem specific and needs a prior insight about the problem.
- The number of function evaluations can be high. Especially if each function evaluation is expensive, direct application of these methods may not be possible.

2.6.3 Surrogate-based adaptive sampling methods

In Section 2.4, the basic concepts of response approximation using surrogates were presented. It was stated that a surrogate can be used to replace the actual responses within any optimization method. Another important use of surrogates is that because they provide an insight into the variation of responses with respect to the variables, they provide methods for adaptive selection of samples. In particular, an adaptive sampling method referred to as Efficient Global Optimization (EGO) (Jones, D.R. et al. (1998)) using Kriging has gained significant popularity, in the context of optimization. At any sample $\mathbf{x}$, Kriging provides a mean prediction of function value as well as a variance. Because of the variance associated with the prediction, even a point with a higher predicted mean objective function value than the current best solution $f(\mathbf{x}^*)$ (or f^*) may have a non-zero probability of being lower than the current solution (Figure 2.14). This allows for the calculation of expected improvement (EI) of the objective function:

$$EI(\mathbf{x}) = E\left[\max(0, f^* - \hat{f})\right] = \int_{-\infty}^{f^*} (f^* - f) f_{\hat{f}}(\mathbf{x}) df \tag{2.26}$$

where f^* is the objective function value at the current best solution, $f_{\hat{f}}$ is the probability density function of the approximated values, and f is a realization of $f_{\hat{f}}$. The basic idea in EGO is to adaptively evaluate samples that maximize the EI, because such samples are expected to improve the objective function. Equation 2.26 can be integrated to express EI as follows.

$$EI(\mathbf{x}) = (f^\star - \mu_f)\Phi\left(\frac{f^\star - \mu_f}{\sigma_f}\right) + \sigma_f \phi\left(\frac{f^\star - \mu_f}{\sigma_f}\right) \tag{2.27}$$

where ϕ and Φ are the standard normal probability density function and the cumulative density function. Maximization of the EI balances the exploration of the unsampled space and the exploitation of the current model in the regions with low objective function values. The derivation of EI (Equation 2.27) is provided in Appendix B.

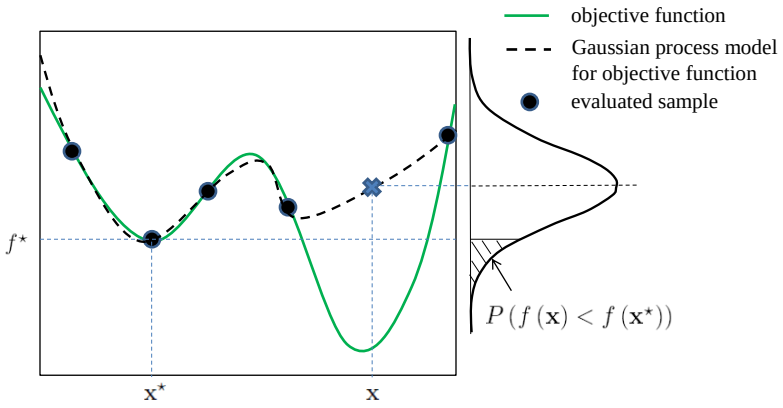

Figure 2.14: Depiction of the probability of improving the current best solution.

Several modifications of EGO have been reported based on modification of the sample selection criterion. For example, samples may be selected based on a generalized expected improvement (GEI) function (Sasena, M.J. et al. (2002)):

$$GEI(\mathbf{x}) = E\left[\max(0, (f^* - \hat{f})^q)\right] = \int_{-\infty}^{f^*} (f^* - f)^q f_{\hat{f}}(\mathbf{x}) df \tag{2.28}$$

The exponent q in the GEI expression governs the globality or locality of the search. For $q = 0$, the GEI reduces to the probability of improvement, which results in a very local search. For $q = 1$ GEI reduces to EI. For higher values of q, the search is more global. Several other sample selection criteria can also be found in the literature (Sasena, M.J. (2002); Forrester, A.I.J. et al. (2008)). Constrained formulations for EGO have also been developed (Schonlau, M. (1997); Sasena, M.J. (2002); Audet, C. et al. (2000); Forrester, A.I.J. et al. (2008)). Some of these are discussed in Chapter 7.

For many problems, use of surrogates for optimization is very efficient. However, optimization using surrogates is hampered by the presence of discontinuities and binary states. Also, handling of multiple constraints is a challenge. A method with classification-based constraint handling is presented in Chapter 7 to overcome such issues.

2.7 Reliability assessment methods

Several reliability assessment methods can be found in the literature, some of which are presented in this section. A major portion of the review is dedicated to reliability assessment methods for uncertainties represented using random variables, for which the literature is abundant. These methods assume the uncertainties to be spatially invariant. A brief review of methods considering spatial variation is also presented in latter sections. In addition, an introduction to the treatment of correlated random variables is also provided.

2.7.1 Probability of failure calculation with spatially invariant uncertainties

Mean value method

The mean value method (Cornell (1969)) is one of the simplest reliability assessment methods that is based on the first and second moments of the response function at a single point, which is the mean configuration of the random variables. The mean response value and variance are calculated based on first order Taylor expansion:

$$\mu_g \approx g(\boldsymbol{\mu}_{\mathbf{x}}) \tag{2.29}$$

$$\sigma_g^2 \approx \sum_{i=1}^{m}\sum_{j=1}^{m} \frac{\partial g}{\partial x_i}(\boldsymbol{\mu}_{\mathbf{x}})\frac{\partial g}{\partial x_j}(\boldsymbol{\mu}_{\mathbf{x}})COV(x_i, x_j) \tag{2.30}$$

Assuming the response distribution to be Gaussian centered at μ_g, the probability of failure is:

$$P_f = P(z \leq 0) = P(g(\mathbf{x}) \leq 0) = \Phi\left(\frac{0 - \mu_g}{\sigma_g}\right) \tag{2.31}$$

Equation 2.31 can also be written in terms of the "safety" or "reliability" index β, defined as the ratio between the mean value and the standard deviation of g:

$$\beta = \frac{\mu_g}{\sigma_g} \tag{2.32}$$

It can be seen from Equation 2.32 that reliability index is high for a large positive value of g, because a large g represents a safe configuration, negative g being failure by convention. Also, β is higher for low values of σ_g, because this represents less uncertainty in the response. Equation 2.31 can be rewritten as:

$$P_f = \Phi\left(\frac{0 - (\beta\sigma_g)}{\sigma_g}\right) = \Phi(-\beta) \tag{2.33}$$

The reliability index β can be interpreted as the number of standard deviations separating the mean μ_g and the threshold $g(\mathbf{x}) = z = 0$ (Figure 2.15). Increasing β decreases the probability of failure, irrespective of the distribution of $g(\mathbf{x})$. It is thus, an indicator of the reliability of a system. However, the probability of failure calculated using Equation 2.33 is correct only if $g(\mathbf{x})$ is normal. For non-normal distributions, it is more appropriate to only provide the β value as a measure of reliability.

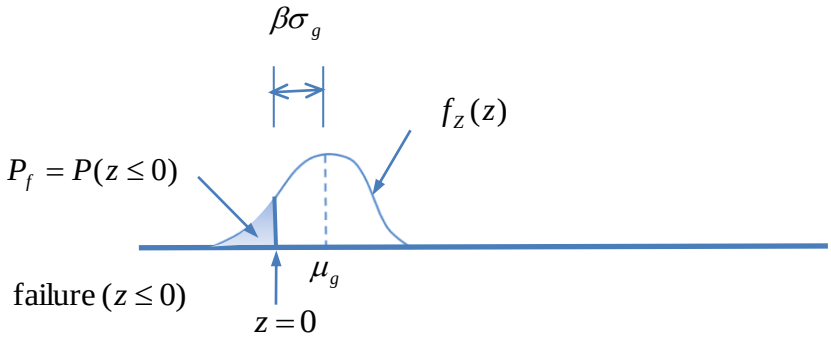

Figure 2.15: Failure probability and reliability index β.

MPP-based methods

Hasofer-Lind Method for normal Variables

One of the issues with Mean Value method lies in the variability of the reliability index β when different formulations of the same limit state function are used (Haldar, A. and Mahadevan, S. (2000)). This limitation was first overcome in the Hasofer-Lind method, which introduced a "generalized reliability index" (Hasofer and Lind (1974)). In this method, the random variables are transformed into standardized variables:

$$u_i = \frac{x_i - \mu_i}{\sigma_i} \tag{2.34}$$

Thus, the resistance and load variables R an S are transformed to:

$$\begin{aligned} u_R &= \frac{R - \mu_R}{\sigma_R} \\ u_S &= \frac{S - \mu_S}{\sigma_S} \end{aligned} \tag{2.35}$$

The limit state function is also transformed into the space of standardized variables:

$$\begin{aligned} & g(\mathbf{x}) = R - S \\ \Rightarrow \quad & g_u(\mathbf{u}) = \sigma_R u_R - \sigma_S u_S + \mu_R - \mu_S \end{aligned} \tag{2.36}$$

The reliability index in Equation 2.32 can be written as:

$$\beta = \frac{\mu_g}{\sigma_g} = \frac{\mu_{R-S}}{\sigma_{R-S}} = \frac{\mu_R - \mu_S}{\sqrt{\sigma_R^2 + \sigma_S^2}} \tag{2.37}$$

Because $g_u(\mathbf{u})$ (Equation 2.36) is linear in the $R - S$ space, the reliability index β in Equation 2.37 can easily be shown to equal the distance of $g_u(\mathbf{u}) = 0$ from the origin of the standardized space. If R and S are linear combinations of variables u_i, then g_u is linear with respect to these variables also. Equation 2.37 can then be generalized to the space of variables u_i. The generalized reliability index β is defined as the algebraic distance of the mean (origin in standardized space) to the closest point on the limit state, known as the design point or most probable point (MPP):

$$\beta = \text{sign}(g_u(\mathbf{u} = \mathbf{0}))||\mathbf{u}^*|| = \text{sign}(g_u(\mathbf{u} = \mathbf{0}))\sqrt{\sum_{i=1}^{m}(u_i^*)^2} \tag{2.38}$$

Being the closest point to the mean lying on the limit state, MPP is the most likely configuration at which failure can occur. Farther the MPP from the mean, greater is the reliability.

The Hasofer-Lind method provides a reliability index that is invariant with respect to the problem formulation. However, it does not include information about the distribution of z or the probability of failure. The probability of of failure can only be calculated in the case of normal random variables using Equation 2.33.

First and Second Order Reliability Methods for Non-normal Variables

As mentioned in previous section, Hasofer-Lind method can only be used to predict failure probabilities with normal variables. In addition, the probability of failure is accurate only for linear limit state functions. These limitations were overcome with the development of first and second order reliability methods respectively (FORM and SORM) (Hohenbichler and Rackwitz (1983); Hohenbichler et al. (1987); Haldar, A. and Mahadevan, S. (2000); Melchers, R.E. (1999)).

In FORM and SORM, all random variables are first converted to uncorrelated standard normal space or U-space. If the original variables are uncorrelated then this conversion is straightforward:

$$u_i = \Phi^{-1}F_{X_i}(x_i) \tag{2.39}$$

where Φ is the standard normal cumulative density function and F is the cumulative density function of the original random variables. A more detailed explanation of conversion of correlated variables to standard normal space is given in Section 2.7.3.

In order to calculate the probability of failure, the MPP is located in standard normal space. The reliability index is:

$$\beta = \text{sign}(g_u(\mathbf{u} = \mathbf{0}))\sqrt{\sum_{i=1}^{m}(u_i^*)^2} = \text{sign}(g_u(\mathbf{u} = \mathbf{0}))\sqrt{\mathbf{u}_i^{*T}\mathbf{u}_i^*} \tag{2.40}$$

MPP is located by solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{u}} \quad & \sqrt{\mathbf{u}_i^{*T}\mathbf{u}_i^*} \\ s.t. \quad & g_u(\mathbf{u}) = 0 \end{aligned} \tag{2.41}$$

where g_u is the limit state function in the standard normal space. The first order Taylor expansion of g' is given as:

$$g_u(\mathbf{u}) = b + \mathbf{a}^T\mathbf{u} = b + \left(\frac{\partial g_u}{\partial \mathbf{u}}\right)^T \mathbf{u} = 0 \tag{2.42}$$

In most practical situations, the function g_u or g is not available in closed form and several evaluations of the implicit limit state function are required. An iterative algorithm may be used to locate the MPP in the general case, starting from an arbitrary point. The procedure is simpler for a linear performance function. For a linear performance function, the starting point $\mathbf{u}_0$ may not lie on the limit state $g'(\mathbf{u}) = 0$, but it lies on a parallel line $g_u(\mathbf{u}) = c$ (hyperplane in multi-dimensional space). The minimum distance point can be found in a single step by searching in the direction of the constant gradient vector $\mathbf{a}$ (Figure 2.16) (Haldar, A. and Mahadevan, S. (2000)).

$$\begin{aligned} g_u(\mathbf{u}^*) &= g_u(\mathbf{u} + \mu\mathbf{a}) = 0 \\ \mathbf{u}^* &= \frac{1}{|\mathbf{a}|^2}[\mathbf{a}^T\mathbf{u}_0 - g_u(\mathbf{u}_0)]\mathbf{a} \end{aligned} \tag{2.43}$$

For the non-linear case, the gradient vector $\mathbf{a}$ is not constant. The minimum distance point is updated at each iteration using the previous gradient, and the new values of the limit state function and its gradient are calculated. The algorithm is repeated until convergence criteria are satisfied. Once the MPP is located, the first

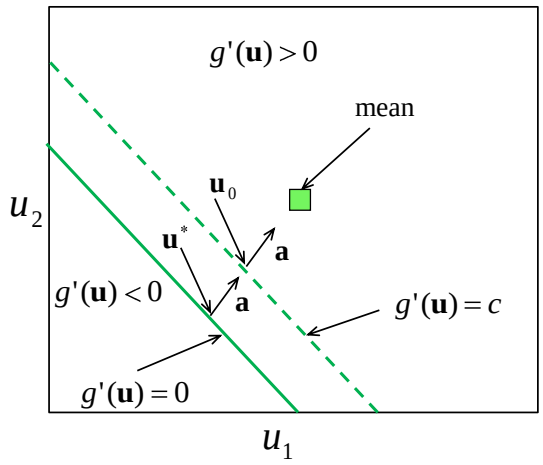

Figure 2.16: First Order Reliability Method (FORM).

order probability of failure is calculated using Equations 2.40 and 2.44.

$$P_f = \Phi(-\beta) \tag{2.44}$$

The first order probability of failure using Equation 2.44 is accurate only for linear limit state functions, as it does not include information about curvature (Figure 2.17).

Second order reliability method (SORM) (Hohenbichler et al. (1987); Haldar, A. and Mahadevan, S. (2000); Melchers, R.E. (1999)) extends the method to quadratic limit state functions by including second order information in the Taylor series expansion:

$$g_u(\mathbf{u}) \approx g_u(\mathbf{u}^*) + \sum_{i=1}^{d}(u_i - u_i^*)\frac{\partial g_u}{\partial u_i} + \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}(u_i - u_i^*)(u_j - u_j^*)\frac{\partial^2 g_u}{\partial u_i \partial u_j} \tag{2.45}$$

An approximation of the second order probability of failure (Breitung (1984)) is:

$$P_f \approx \Phi(-\beta)\prod_{i=1}^{d-1}(1 + \beta\kappa_i)^{-\frac{1}{2}} \tag{2.46}$$

where κ_i denotes the i^{th} principal curvature of the limit state at the MPP. The above approximation of P_f is, however, accurate only for large β values. Also, it

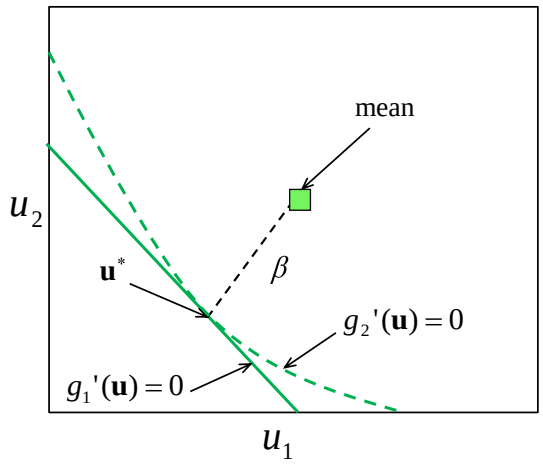

Figure 2.17: Limitation of FORM. Probability of failure using g_{u1} and g_{u2} will be same using FORM because it only depends on the position of MPP and not the curvature.

does not consider cross terms between variables (Haldar, A. and Mahadevan, S. (2000)).

The major limitation of FORM and SORM lies in the first or second order Taylor expansion of the limit state function. As a result, the probability of failure estimate can have significant errors when the limit state is highly nonlinear and consists of several MPPs. Modified FORM and SORM methods have been developed to account for multiple MPPs (Der Kiureghian and Dakessian (1998); Barranco-Cicilia, F. et al. (2009); Gupta and Manohar (2004)). The basic idea is to find the multiple MPPs successively and treat each MPP as if it corresponds to a component limit state. The probability of failure is calculated using a series system reliability analysis based on the component limit states. One method to solve for successive MPPs is to solve the following optimization problem (Barranco-Cicilia, F. et al. (2009)):

$$\begin{aligned} \max_{\mathbf{u}} \quad & I_g(\mathbf{u}) \prod_{i=1}^{m} \phi_{U_i}(u_i) \\ s.t. \quad & ||\mathbf{u}_j - \mathbf{u}_k^*|| \geq R \qquad \forall k, R \in [1,3] \end{aligned} \tag{2.47}$$

In Equation 2.47, once an MPP $\mathbf{u}_k^*$ is located, it is surrounded by an imaginary

hypersphere of radius R. Thus, the next optimization finds a different solution and not the current MPP. Although such methods can significantly improve the accuracy of failure probability, they also increase the cost of reliability analysis. Also, they still may not be accurate in the general case as the component limit states are based on first or second order approximations.

Combination of MPP search and response approximation

Mean value method, based on two first moments of the response function $z = g(\mathbf{x})$, and the MPP-based methods (FORM and SORM) are based on first and second order approximations of the limit state function. Therefore, their accuracy in the general case is questionable. Also, the search for MPP may require several function evaluations. More advanced methods based on combination of MPP search and approximation of response z have been developed to overcome these limitations. These include methods such as Advanced Mean Value (AMV) (Wu et al. (1990); Youn, B.D. et al. (2005)) and Two Point Adaptive Nonlinear Approximations (TANA) (Wang and Grandhi (1995); Wang and Grandhi (1994)).

The first step in AMV consists of finding the Mean Value solution. However, it is known that this is valid only in the case of linear limit state functions. Therefore, additional function evaluations are performed to improve the approximation of the probability density function for z. The main steps in AMV are:

1. Expand $g(\mathbf{x})$ as a linear function about $\boldsymbol{\mu}_{\mathbf{x}}$

2. Use the linear approximation of $g(\mathbf{x})$ to calculate reliability index β and MPP for k different values of z, i.e. for k translated limit states $z - z_0^{(k)}$

3. Reevaluate z at each design point. This provides k new pairs of β_i and z_i and thus gives an improved estimate of the CDF of z. This is referred to as the advanced mean value estimate.

The above method is hampered if multiple MPPs are present, because it is based on following the locus of the MPP for sample evaluation. Variations of the method

have been developed to overcome such issues. However, these methods are still not sufficient if several MPPs are present (Wu et al. (1990)).

Two-point Adaptive Nonlinear Approximation (TANA) method follows a similar idea of combining MPP search with response approximation. In order to account for nonlinearity of limit state functions, a nonlinearity index r_i is introduced in this method for the i^{th} random variable. The physical variables x_i are transformed to y_i as:

$$y_i = x_i^{r_i}, \quad i = 1, 2, \ldots, m \tag{2.48}$$

Starting from a Mean Value solution, MPP is updated until convergence. Approximation of response $g(\mathbf{x})$ is based on information at the last two MPPs $\mathbf{x}_1$ and $\mathbf{x}_2$, and is obtained by expanding about the current MPP $\mathbf{x}_2$ (Wang and Grandhi (1994)):

$$\begin{aligned} g(\mathbf{x}) &\approx g(\mathbf{x}_2) + \sum_{i=1}^{m} \frac{\partial g(\mathbf{x}_2)}{\partial y_i}(y_i - y_{i,2}) + \frac{1}{2}\sum_{i=1}^{m} \frac{\partial^2 g(\mathbf{x}_2)}{\partial y_i^2}(y_i y_{i,2})^2 \\ &\approx (\mathbf{x}_2) + \sum_{i=1}^{m} \frac{\partial g(\mathbf{x}_2)}{\partial x_i}\frac{\partial x_{i,2}^{1-r_i}}{r_i}(x_i^{r_i} - x_{i,2}^{r_i}) + \frac{1}{2}\epsilon_2 \sum_{i=1}^{m}(x_i^{r_i} - x_{i,2}^{r_i})^2 \end{aligned} \tag{2.49}$$

The nonlinearity indices r_i and the coefficient ϵ_2 are unknown. Thus, Equation 2.49 has $m+1$ unknowns that require $m+1$ equations for solution. These unknowns are calculated based on zero and first order information at previous MPP $\mathbf{x}_1$.

$$\begin{aligned} g(\mathbf{x}_1) &= (\mathbf{x}_2) + \sum_{i=1}^{m} \frac{\partial g(\mathbf{x}_2)}{\partial x_i}\frac{\partial x_{i,2}^{1-r_i}}{r_i}(x_{i,1}^{r_i} - x_{i,2}^{r_i}) + \frac{1}{2}\epsilon_2 \sum_{i=1}^{m}(x_{i,1}^{r_i} - x_{i,2}^{r_i})^2 \\ \frac{\partial g(\mathbf{x}_1)}{\partial x_i} &= \left(\frac{x_{i,1}}{x_{i,2}}\right)^{r_i - 1} \frac{\partial g(\mathbf{x}_2)}{\partial x_i} + \epsilon_2 (x_{i,1}^{r_i} - x_{i,2}^{r_i}) x_{i,1}^{r_i - 1} r_i \quad i = 1, 2, \ldots, m \end{aligned} \tag{2.50}$$

The first MPP estimate is based on the mean value solution. For this first iteration, the previous design point $\mathbf{x}_1$ is set equal to the mean. The MPP is updated iteratively until the reliability index β converges.

Sampling methods

Monte Carlo Simulations

All methods presented in previous sections are based on certain approximations

regarding the shape of limit state function. It is always possible to find counterexamples where these methods are unable to provide accurate probabilities of failure. The most basic reliability assessment method, which does not make any assumption regarding the shape of limit state, consists of numerical integration using Monte Carlo simulations (Metropolis and Ulam (1949); Melchers, R.E. (1999)). It is often used as a benchmark for validating the accuracy of other methods. The basic concept of MCS is shown in Figure 2.18 using two random variables. It consists of generating a large number of random samples based on the PDFs of random variables concerned. Response values are evaluated at all these samples, and the samples lying in the failure domain Ω_f are identified. For instance, the responses may be compared to a threshold response governing the failure criterion for this purpose. The probability of failure in Equation 2.5 can be written in discrete form as:

$$P_f = \frac{N_f}{N_{MC}} = \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} I_g(\mathbf{x}_i) \tag{2.51}$$

where N_f is the number of MCS samples lying in the failure domain Ω_f and N_{MC} is the total number of MCS samples. $I_g(\mathbf{x})$ is an indicator function given as:

$$I_g(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in \Omega_f \\ 0 & \text{otherwise} \end{cases} \tag{2.52}$$

MCS provides accurate probabilities of failure irrespective of the level of non-linearity, provided a sufficiently large number of samples is used. The accuracy of probability of failure depends on the level of probability and the number of samples N_{MC}. If there are N_f occurrences of failure among N_{MC} samples, the probability of obtaining N_f failures using MCS $P(N_f)$ is:

$$P(N_f) = \binom{N_{MC}}{N_f} P_f^{N_f} (1 - P_f)^{N_{MC} - N_f} \tag{2.53}$$

The probability of failure P_f has a distribution centered at $\frac{N_f}{N_{MC}}$. If N_{MC} is small, the variance of this distribution is high. This might result in inaccurate estimation of P_f. The variance tends to zero when N_{MC} tends to infinity. The coefficient of

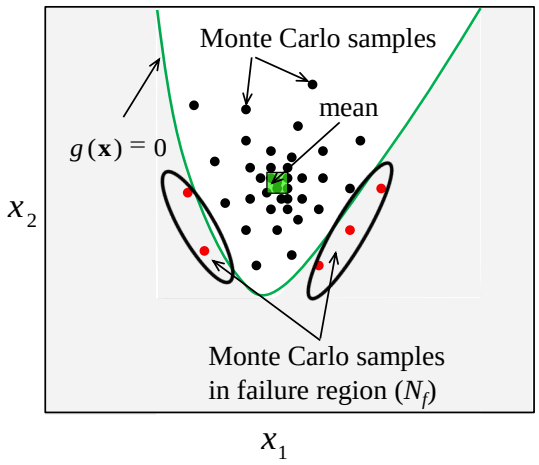

Figure 2.18: Monte Carlo Simulations. Failure probability is the fraction of samples in the failure region (shaded).

variation for the probability of failure estimate is:

$$COV = \frac{\sqrt{\frac{P_f^t(1-P_f^t)}{N_{MC}}}}{P_f^t} = \sqrt{\frac{1-P_f^t}{N_{MC}P_f^t}} \tag{2.54}$$

where P_f^t is the true probability of failure. The 95% confidence interval of the MCS estimate is:

$$95\%\ \text{CI} = \left[\left(P_f - 2P_f\sqrt{\frac{1-P_f^t}{N_{MC}P_f^t}}\right), \left(P_f + 2P_f\sqrt{\frac{1-P_f^t}{N_{MC}P_f^t}}\right)\right] \tag{2.55}$$

It is evident from Equation 2.54 and 2.55 that the coefficient of variation and confidence interval of probability estimate can be quite high if the actual probability of failure P_f^t appearing in the denominator is small. In many engineering problems, the cost of a single simulation can be quite high, e.g. for impact analysis or for fluid-structure interaction problems. Therefore, the high number of MCS samples required for an accurate probability of failure makes the method impractical to use in most situations.

Variance Reduction Techniques for Monte Carlo Simulations

Due to the large number of samples required, application of MCS directly is not

possible in many practical situations. Methods have been developed to reduce the number of MCS samples, using variance reduction techniques. Using these techniques, accurate results are obtained using relatively smaller MCS sample size. Also, they allow the calculation of much lower probabilities of failure compared to basic MCS. The general approach in these methods is to select the samples in a way that will reduce variance, instead of randomly generating samples in the whole space based on the original probability density functions of the random variables.

One of the popular variance reduction techniques is Importance Sampling (IS) (Harbitz (1986); Schueller and Stix (1987); Melchers (1989)), which aims to concentrate the distribution of Monte-Carlo samples in the region of most importance, i.e. the the region that contributes most to the failure probability. There are several ways to achieve this. One way is to select the samples outside the β sphere around the origin of the standard normal space (Harbitz (1986)). Because β is the distance to the closest point on the limit state function, there are no failures within the sphere. In another approach, the mean of the sampling points is translated to the MPP (Melchers (1989)). Because the distribution of samples is centered at the MPP, and not the origin of standard normal space, more samples lie in the failure domain. This helps in reducing the variance while calculating small probabilities of failure. In all IS methods, samples are generated using a modified probability density function referred to as the IS density. Therefore, the probability of failure cannot be calculated directly by calculating the fraction of samples in the failure domain (Equation 2.51). Instead, a weighing factor is required to relate the modified PDF to the original one. The probability of failure is calculated as:

$$P_f = \int_{\Omega_f} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} = \int_{\Omega_f} \frac{f_{\mathbf{X}}(\mathbf{x})}{p_{\mathbf{X}}(\mathbf{x})} p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} = \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} I_g(\mathbf{x}_i) \frac{f_{\mathbf{X}}(\mathbf{x}_i)}{p_{\mathbf{X}}(\mathbf{x}_i)} \tag{2.56}$$

where $p_{\mathbf{X}}$ is the IS density function.

A critical step in IS is to select an appropriate IS density. The issue with IS is that information about the optimal IS density (Ang et al. (1992)) not known a

priory. Adaptive Importance Sampling (AIS) (Givens and Raftery (1996); Au and Beck (1999)) can be used to overcome this issue. In AIS, the IS density function is updated as more information is added. An AIS method proposed in Au and Beck (1999) aims at approximating the optimal IS density using the Markov Chain Monte Carlo (MCMC) method (Gilks et al. (1996)) and Kernel Sampling Density (Ang et al. (1990, 1992)). The optimal importance sampling density is:

$$p_{opt}(\mathbf{x}) = \frac{I_g(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x})}{P_f} \tag{2.57}$$

The optimal IS density is not known a priori as P_f is unknown. However it can be approximated using MCMC, which requires only the ratio of $p_{opt}(\mathbf{x})$ at candidate samples. As a result, the constant of proportionality $\frac{1}{P_f}$ cancels out. Once a sufficiently large number of samples is selected using MCMC, an approximation of the optimal density is constructed using Kernel Sampling Density Methods. This approximation is then used to calculate the probability of failure using Equation 2.56.

Stratified sampling is another technique to reduce the variance (Haldar, A. and Mahadevan, S. (2000)). In this method the total domain is divided into several mutually exclusive domains. Specified number of samples is generated in each region. More samples are selected in the regions that contribute to the failure event. The required probability of failure is then calculated using the theorem of total probability, by considering the probabilities of the individual regions R_i, and the conditional probabilities of failure within those regions:

$$P_f = \sum_{i=1}^{N_R} \frac{N_f^{(i)}}{N_{MC}^{(i)}} P(R_i) \tag{2.58}$$

where $P(R_i)$ is the probability of region R_i, N_R is the number of mutually exclusive regions, $N_{MC}^{(i)}$ is the number of Monte Carlo samples in the region R_i, and $N_f^{(i)}$ is the number of failed Monte Carlo samples in the region R_i.

Another useful variance reduction technique for failure probability calculation is Subset Simulation (Au and Beck (2001); Au et al. (2007)). It is especially useful for calculating very low probabilities of failure. Instead of tackling the low failure probability in a single step, the calculation of probability of failure is performed in several steps with higher target probabilities of failure. Larger probability values are calculated more accurately using a relatively smaller sample size. The samples for the first step are generated using MCS. The threshold response $g_0^{(1)}$ for first step is selected such that the fraction of samples with $g(\mathbf{x}) \leq g_0$ is equal to the target probability. For subsequent steps, conditional samples are selected using MCMC. These samples are selected such that they lie in the failure region based on the previous threshold. The next threshold (lower $g_0^{(2)}$) is again calculated in the same manner based on the target failure probability. The process is repeated until $g_0^{(i)}$ is zero. First two steps of Subset Simulations Method are shown in Figure 2.19.

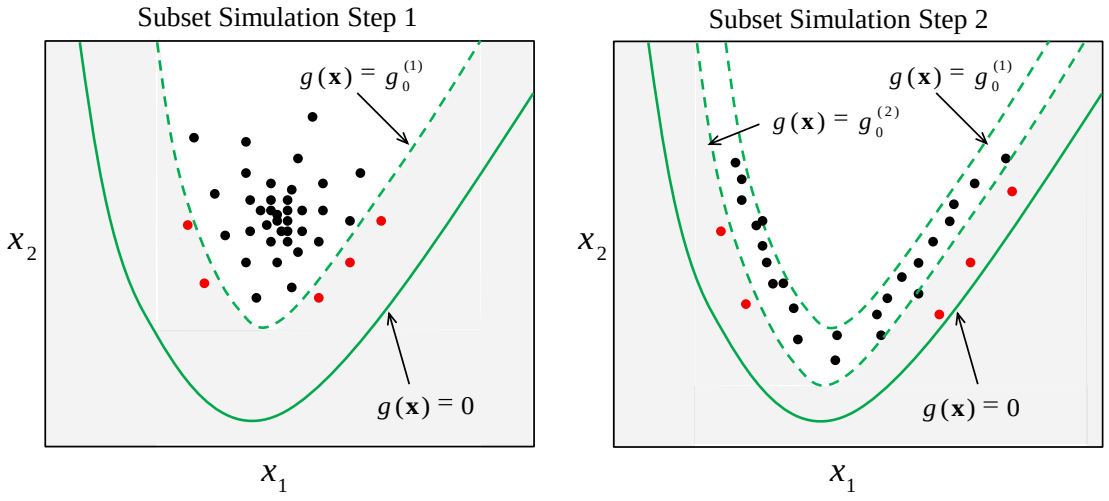

Figure 2.19: First two steps of Subset Simulations Method. The shaded regions in left and right figures are F_1 and F_2 ($F_2 \subset F_1$).

Representing failure in i^{th} substep by F_i ($F_i \subset F_{i-1}$), the probability of failure

is calculated as:

$$P_f = P(F_1) \prod_{i=2}^{N_{step}} P(F_i|F_{i-1}) \tag{2.59}$$

where N_{step} is the number of substeps required. The target probability in subset simulations is usually selected between 0.1 to 0.2.

The number of samples required is significantly reduced by the application of variance reduction techniques, but it can still be quite high. Actual function evaluation at each of these points can still be computationally expensive. This issue can be addressed using surrogate models, as explained in Section 2.7.1.

Surrogate-based adaptive sampling methods

The basic response approximation methods presented in Section 2.4 use a DOE to sample the space globally. The probability of failure can be calculated based on the approximated response, e.g. using MCS (Equation 2.51). The indicator function is:

$$I_g(\mathbf{x}) = \begin{cases} 1 & \hat{g}(\mathbf{x}) \leq 0 \\ 0 & \hat{g}(\mathbf{x}) > 0 \end{cases} \tag{2.60}$$

where $\hat{g}$ is the approximation of the response function g, and is evaluated efficiently. However, for an accurate probability estimate, the interest lies in certain regions of the space close to the limit state $g(\mathbf{x}) = 0$. Therefore, the procedure based on a static DOE may not be accurate. Adaptive sampling techniques have been developed to overcome such issues and construct an accurate response approximation in the vicinity of the limit state function (Wang, G.G. et al. (2005); Bichon, B.J. et al. (2007)). Of particular interest is the Kriging-based Efficient Global Reliability Analysis (EGRA) (Bichon, B.J. et al. (2007)). Kriging provides the variance of response approximation along with response values, which allows for an efficient sampling strategy. Using Kriging, a response function $g(\mathbf{x})$ is approximated using Equation 2.14. In EGRA, an initial Kriging approximation is built using a DOE.

The approximation is adaptively updated based on an Expected Feasibility Function (EFF):

$$EFF(\mathbf{x}) = \int_{\bar{z}-\epsilon}^{\bar{z}+\epsilon} [\epsilon - |\bar{z} - g|] f_{\hat{g}}(\mathbf{x}) dg \tag{2.61}$$

where $f_{\hat{g}}$ is the PDF of approximated limit state function values, g represents a realization of $f_{\hat{g}}$, ϵ is proportional to the standard deviation of the Kriging prediction σ_g, and $\bar{z}$ is the threshold response. For consistency of conventions $\bar{z} = 0$, and Equation 2.62 can be written as:

$$EFF(\mathbf{x}) = \int_{-\epsilon}^{\epsilon} [\epsilon - |g|] f_{\hat{g}}(\mathbf{x}) dg \tag{2.62}$$

Equation 2.62 can be integrated to express EFF in analytical form. The adaptive samples are selected such that they maximize the EFF. There are two factors that lead to a high EFF. It is high if $\mathbf{x}$ is close to the $g(\mathbf{x}) = 0$ contour or if the variance of Kriging predictor at $\mathbf{x}$ is high.

The EGRA method explained above, as well as other surrogate-based methods, work efficiently as long as responses are continuous. However, these methods are hampered by discontinuous and binary responses. Also, the method, in its original form, is not suitable for multiple failure modes. Recent attempts to extend surrogate-based reliability assessment for handling multiple failure modes have been made using active set methods (Bichon et al. (2010)). However, a more natural way of handling multiple failure modes is to treat reliability assessment as a classification problem instead of a response approximation one (Section 2.5). The conceptual shift from approximation to classification is the core idea in this dissertation, and it also enables the handling of discontinuous and binary responses.

2.7.2 Probability of failure calculation with spatially varying uncertainties

Most reliability assessment methods in the literature use random variables for representing uncertainties. However, random variables may not provide a realistic representation in cases with spatial variation of parameters. For example, in sheet

metal forming, thickness may vary from one sheet to another. However, even for a single sheet, thickness may not be constant everywhere. Such spatial variations are represented more realistically using random fields (Sudret and Der Kiureghian (2000)). These considerations are, therefore, important in the reliability assessment of such systems. Most of the literature pertaining to random fields is based on stochastic finite element methods (SFEM) (Ghanem and Spanos (2003); Stefanou (2009); Sudret and Der Kiureghian (2000)). This section presents a brief overview of these methods. The general approach for reliability assessment with random fields consists of two steps:

- Characterization of random field. The general practice is to discretize a random field and represent it using a few random variables.
- Reliability assessment using random variables representing the discretized random field.

Characterization of random fields

A random field can be interpreted as a collection of infinite random variables over the space. Such a representation, however, is not practical. The general practice to represent random fields for reliability assessment is to discretize them at selected points in the space. There are several ways of representing random fields. These include point estimate methods, average discretization methods and series expansion methods. Point estimation methods involve representation of random fields using values at specific points in the space, such as finite element centroids, nodes or integration points. Average discretization methods involve representation using average values at specific points, such as averaging of element values at nodal points. Using series expansion methods, a random field is represented exactly as a series, and an approximation is obtained by truncating the series. A commonly used method to represent random fields is Karhunen-Loeve (KL) expansion based on the eigenvalue decomposition of the autocovariance function (Ghanem and Spanos (2003)):

$$S = \bar{S} + \sum_{i=1}^{\infty} \sqrt{\lambda_i} \xi_i \phi_i \tag{2.63}$$

where $\bar{S}$ is the mean at a specific spatial point, ξ_i are stationary random variables, and λ_i and ϕ_i are the eigenvalues and eigenfunction obtained from the spectral decomposition of the autocovariance function associated with the random field. The discrete form of KL expansion is referred to as Proper Orthogonal Decomposition (POD) (Bui-Thanh, T. et al. (2003); Liang,Y. C. et al. (2002)). In POD, M snapshots are observed at N measurement points. The random field is expanded on the basis formed by the eigenvectors of the covariance matrix:

$$S = \bar{S} + \sum_{i=1}^{M} \alpha_i V_i \tag{2.64}$$

where V_i are the eigenvectors of covariance matrix and α_i are random variables. Usually only a few terms of the expansion, with the largest eigenvalues, are important. The random field is approximated as:

$$S = \bar{S} + \sum_{i=1}^{MS} \alpha_i V_i \tag{2.65}$$

where MS is the number of important features or eigenvectors and usually $MS << M$. The expansion of a random field on the basis of eigenvectors of covariance matrix is optimal in the sense that it gives a lower truncation error compared to the same number of terms with any other basis. This is what makes POD and KL expansion very attractive for representation of random fields.

Reliability assessment using random fields

The literature dealing with reliability assessment using random fields is largely dominated by stochastic finite element method (SFEM) (Ghanem and Spanos (2003)). SFE enables the propagation of uncertainties to obtain the distribution of system responses using polynomial chaos expansion (PCE). This involves introduction of uncertainties into the equilibrium equation by modifying the stiffness matrix, followed by representation of the inverted stiffness matrix using an expansion. In deterministic FEM, the nodal quantities $\mathbf{d}$ are solved from the following system of equations:

$$\mathbf{Kd} = \mathbf{F} \tag{2.66}$$

In SFEM, uncertainties are propagated to the responses by introducing them in the construction of stiffness matrix. For example, if the constitutive law is represented by a random field, the stiffness matrix is expressed as:

$$\mathbf{K} = \bar{\mathbf{K}} + \sum_{i=1}^{\infty} \sqrt{\lambda_i} \left(\int_{\Omega} \phi_i \mathbf{B}^T \bar{\mathbf{D}} \mathbf{B} d\Omega \right) \xi_i \tag{2.67}$$

where $\bar{\mathbf{D}}$ is the elasticity matrix based on the mean value over the space (i.e. without considering spatial variation), B is a matrix containing derivatives of shape functions, and Ω is the volume of the material. The nodal quantities can be approximated using Neumann series expansion (Yamazaki (1988)):

$$\mathbf{d} = \sum_{j=1}^{\infty} (-1)^j \left[\sum_{i=1}^{\infty} \bar{\mathbf{K}}^{-1} \mathbf{K}_i \xi_i \right]^j \bar{\mathbf{d}} \tag{2.68}$$

where $\bar{\mathbf{d}}$ is the solution of Equation 2.66. $\mathbf{K}_i$ is given as:

$$\mathbf{K}_i = \sqrt{\lambda_i} \left(\int_{\Omega} \phi_i \mathbf{B}^T \bar{\mathbf{D}} \mathbf{B} d\Omega \right) \tag{2.69}$$

The basis of expansion in Equation 2.68 is not orthogonal. The expansion of $\mathbf{d}$ can also be related to an orthogonal basis of Hermite polynomials.

SFE provides a rigorous framework of propagating uncertainties in the form of random fields. However, there are several difficulties that limit its use. It is quite evident that even for simple linear FEM problems, the use of SFEM is quite involved. Use of SFEM is mostly limited to linear problems. Most SFEM methods are intrusive, although non-intrusive methods have also been developed recently (Ghiocel and Ghanem (2002); Berveiller et al. (2006); Huang et al. (2007)). A portion of this dissertation also addresses the calculation of probabilities using random fields. The proposed method in Chapter 8 is non-intrusive, and is based on a combination of POD and SVM-based EDSD.

2.7.3 Probability of failure calculation with correlated random variables

The reliability methods in Section 2.7.1 were presented for independent random variables. Further, some of the methods are based on independent normal variables

only. In general however, random variables can have any probabilistic distribution type. Also, in real applications, random variables associated with a system are often correlated. Not accounting for such correlation can lead to erroneous probability of failure estimates. Therefore, there is a need to calculate probabilities of failure with correlated variables. The general approach for addressing problems with correlated random variables is to transform them into equivalent uncorrelated standard normal variables. Two of the common methods, Rosenblatt and Nataf transformations, are presented in this section.

Rosenblatt transformation

In Rosenblatt Transformations, variables are transformed from the original X-space to standard normal space (U-space) using conditional distributions. The transformation is performed one variable at a time:

$$\begin{aligned} u_1 &= \Phi^{-1} F_{X_1}(x_1) \\ u_2 &= \Phi^{-1} F_{X_2}(x_2|x_1) \\ &\vdots \\ u_m &= \Phi^{-1} F_{X_m}(x_m|x_1, x_2, \ldots, x_{m-1}) \end{aligned} \tag{2.70}$$

Conditional probability density function of i^{th} variable in the X-space is:

$$f_{X_i}(x_i|x_1, x_2, \ldots, x_{i-1}) = \frac{f_{X_1,X_2,\ldots,X_i}(x_1, x_2, \ldots, x_i)}{f_{X_1,X_2,\ldots,X_{i-1}}(x_1, x_2, \ldots, x_{i-1})} \tag{2.71}$$

The conditional cumulative density functions are calculated by integrating Equation 2.71. These are then substituted into Equation 2.70 to perform the transformation. There are two main issues in Rosenblatt Transformation. First, because conditional distributions are used, it depends on the order in which variables are transformed. Also, it requires the knowledge of the joint probability density function, which is not always available. These issues are overcome in Nataf Transformation presented in the following section.

Nataf transformation

The transformation of any instance of arbitrary correlated random variables x_i to uncorrelated standard normal samples u_i, using Nataf Transformation, is a two step process. First, a transformation to correlated standard normal space is performed. In the next step, a transformation to remove the correlation is performed.

- *Step 1 (Transformation to correlated standard normal space):* Suppose the marginal marginal cumulative distribution functions of variables x_i are known. Transformation to correlated standard normal variables is obtained as:

$$\begin{aligned} u_1^0 &= \Phi^{-1} F_{X_1}(x_1) \\ u_2^0 &= \Phi^{-1} F_{X_2}(x_2) \\ &\vdots \\ u_m^0 &= \Phi^{-1} F_{X_m}(x_m) \end{aligned} \tag{2.72}$$

  Unlike Rosenblatt Transformation, Nataf transformation does not require conditional joint probability distributions. However, it should be noted that the standard normal variables u_i^0 are correlated, and still need to be decorrelated. The correlation matrix ρ^0 in U^0-space is required for this purpose. However, it is not the same as ρ in X-space. A relation is required between the two, before decorrelating the standard normal variables. The relation between ρ^0 and ρ can be determined from the basic definition of correlation coefficient:

$$\begin{aligned} \rho_{ij} &= \frac{\mathrm{cov}(X_i, X_j)}{\sigma_{X_i}\sigma_{X_j}} \\ &= \frac{E(X_i X_j) - E(X_i)E(X_j)}{\sigma_{X_i}\sigma_{X_j}} \\ &= \frac{1}{\sigma_{X_i}\sigma_{X_j}} \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} x_i x_j f_{X_i,X_j}(x_i, x_j) dx_i dx_j \\ &\quad - \frac{1}{\sigma_{X_i}\sigma_{X_j}} \int_{-\infty}^{+\infty} x_i f_{X_i}(x_i) dx_i \int_{-\infty}^{+\infty} x_j f_{X_j}(x_j) dx_j \end{aligned} \tag{2.73}$$

  In order to relate ρ and ρ^0, the relation between joint probability density

functions in the X-space and U^0-space is used:

$$f_{X_i,X_j}(x_i,x_j) = \phi_2(u_i^0,u_j^0,\rho^0)\frac{f_{X_i}(x_i)f_{X_j}(x_j)}{\phi(u_i^0)\phi(u_j^0)} \tag{2.74}$$

where $\phi_2(u_i^0,u_j^0,\rho^0)$ is the bivariate standard normal probability density function with correlation coefficient of ρ_0. The Jacobian transformation (Lemaire et al. (2009)) from X-space to U-space is:

$$f_{X_i}(x_i)dx_i = \phi(u_i)du_i \tag{2.75}$$

The simplified expression for correlation coefficient is:

$$\begin{aligned}\rho_{ij} &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} u_i^0 u_j^0 f_{X_i,X_j}(x_i,x_j)dx_i dx_j \\ &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} u_i^0 u_j^0 \phi_2(u_i^0,u_j^0;\rho_{ij}^0)du_i^0 du_j^0\end{aligned} \tag{2.76}$$

Equation 2.76 is solved to find the ρ^0 corresponding to each ρ. This process is often replaced by a polynomial approximation (Kiureghian and Liu (1986)) to avoid solution of the double integral.

- *Step 2 (Decorrelation of standard normal variables):* The transformation from correlated standard normal space (U^0-space) to uncorrelated standard normal space (U-space) is linear:

$$U^0 = AU + B \tag{2.77}$$

Both U^0 and U here are standard normal variables and, therefore, have zero mean. Using Equation 2.77, the means are calculated as:

$$E(U^0) = E(AU+B) = AE(U) + B = 0 \tag{2.78}$$

Equation 2.78 is satisfied only if $B = 0$. The transformation coefficient A is obtained by solving the following system of equations:

$$AA^T = \rho^0 \tag{2.79}$$

A is lower triangular matrix of ρ^0 obtained by Cholesky decomposition.

Although Nataf transformation is simpler to implement compared to Rosenblatt, it has limitations that can lead to errors in the approximation of the joint cumulative density function (Celorrio, L. et al. (2009)). More recently, use of various copulas has gained popularity for performing the transformation to overcome some of these limitations. Nataf transformation is a special case that uses a Gaussian copula. In the work presented in this dissertation, correlation between variables has not been considered. However, the methods can be applied in an uncorrelated space, following a transformation using the techniques discussed in this section.

2.7.4 Calculation of system reliability

In engineering applications, very often systems with several components are encountered. Each component has one or more associated responses that govern its failure. The probability of failure for individual components can be calculated using the methods presented in previous sections. However, if system reliability is to be calculated then the knowledge of interactions between the components is necessary. A system can be a series system, a parallel system or a combination of the two. A series system is one, for which system failure is defined as the failure of one or more components. For a parallel system, it is defined as the failure of all components. The traditional method is to determine all the component probabilities of failure and then calculate the system reliability through postprocessing of these values (Lin, P.M. et al. (1976); Martz and Waller (1990)). For a series system, the system probability of failure is:

$$P_f^{sys} = 1 - \prod_{i=1}^{n_c}(1 - P_f^{(c_i)}) = 1 - \prod_{i=1}^{n_c}(1 - P(g_{c_i}(\mathbf{x}) \leq 0)) \tag{2.80}$$

where $P_f^{(c_i)}$ is the probability of failure of i^{th} component and n_c is the number of components. Probability of failure for a parallel system is:

$$P_f^{sys} = \prod_{i=1}^{n_c} P_f^{(c_i)} = \prod_{i=1}^{n_c} P(g_{c_i}(\mathbf{x}) \leq 0) \tag{2.81}$$

It is possible to derive the probability of failure expressions for mixed systems although they are more involved. The issue with treating each component separately for system probability of failure calculation is that it might lead to unnecessary response evaluations for all the components. Because the ultimate goal is to calculate the system reliability, calculation of accurate failure probabilities for each system may be unnecessary, and leads to additional computation cost. Bayesian methods have been developed to perform selective testing at component and sub-system levels, with the objective of minimizing the cost of evaluations (Sankararaman, S. et al. (2011); Salas, P. et al. (2011)). Response approximation methods that avoid unnecessary evaluations have also been developed. One method is to build a composite response approximation, e.g. based on the minimum of all component responses for a series system. However, because the individual responses can be quite different, this may lead to a discontinuous composite response. Approximation of such a response becomes complicated and may not be accurate. Another method, based on an extension of EGRA, was developed in Bichon et al. (2010) that does not require a composite response function. Because the handling of responses is based on classification using SVM in this dissertation, the proposed method is unaffected by the presence of discontinuities. A single SVM can therefore be used to represent the system-level limit state function, instead of approximating each component-level one. The same advantages apply in the context of multiple failure modes for one or more components.

2.7.5 Error margins in reliability assessment

Until now, several methods for reliability assessment under different conditions have been presented in this chapter. Methods with and without correlation between different sources of uncertainties were presented. Also, methods to consider spatial variations of uncertainties were presented. However, despite significant research activities in the area, reliability assessment of systems is still prone to errors. Errors can arise due to modeling errors, or due to approximations in the representation of uncertainties and in the failure probability calculation method. Therefore,

quantification of uncertainties in the reliability assessment methods themselves is also important. Conservative estimates can be used to reduce the chances of failure.

Several methods exist for providing conservative probability of failure estimates. One of the most basic approaches is to use a safety factor or safety margin while predicting responses. Conservative estimates of responses can be obtained as:

$$\hat{g}_c(\mathbf{x}) = \hat{g}(\mathbf{x}) + SM \tag{2.82}$$

$$\hat{g}_c(\mathbf{x}) = \hat{g}(\mathbf{x}) \times SF \tag{2.83}$$

where SM is a positive Safety Margin and SF is a safety factor greater than 1. Probabilities of failure calculated using responses $\hat{g}_c(\mathbf{x})$ given by Equations 2.82 and 2.83 will be more conservative compared to those using $\hat{g}(\mathbf{x})$. However, the choice of safety factor and safety margin is arbitrary, and they may not be effective always.

Another method to quantify uncertainties in predicting response values and corresponding failure probabilities, in the context of surrogate-based methods, is to use the variance of prediction. Confidence intervals for failure probabilities can be provided using regression or Kriging. Such a confidence interval is, however, based on prior assumptions on the error distribution of the surrogate. Another approach for quantifying prediction errors is to apply Bootstrap method, which does not assume any prior error distribution (Picheny, V. (2009)). In this dissertation, errors in calculation of failure probabilities using SVMs are quantified using a method based on PSVMs. The proposed method is presented in Chapter 6.

2.8 Reliability-based design optimization (RBDO) methods

An overview of various methods for deterministic optimization and reliability assessment was provided in previous sections. The importance of considering uncertainties in design was emphasized. This is achieved in Reliability-based Design Optimization (RBDO), in which uncertainties are considered during optimization. Unlike

deterministic optimization with a factor of safety, in which reliability of the system is generally unknown, RBDO is performed with a specified (usually low) level of failure probability. Typically, RBDO problems are formulated as:

$$\begin{aligned} &\min_{\bar{\mathbf{x}}} \quad f(\bar{\mathbf{x}}) \\ s.t. \quad &P(g(x) \leq 0) - P_T \leq 0 \end{aligned} \tag{2.84}$$

where f is the objective function, g is the limit state function and P_T is the target probability of failure. $\bar{\mathbf{x}}$ is the mean design configuration. The quantity $P(g(x) \leq 0)$ is the probability of failure. In the general case, the failure region may be represented by multiple limit state functions corresponding to different modes of failure. In many cases, instead of using the probability of failure directly, the RBDO problem is defined using reliability index β:

$$\begin{aligned} &\min_{\bar{\mathbf{x}}} \quad f(\bar{\mathbf{x}}) \\ s.t. \quad &\beta_T - \beta \leq 0 \end{aligned} \tag{2.85}$$

It is quite evident from Equations 2.84 and 2.85 that optimization and reliability assessment are coupled in an RBDO problems. The nature of implementation of this coupling separates one RBDO method from the others. There are three major RBDO frameworks that exist, although a combination of the methods can also be used. The RBDO methods differ further from each other based on the method used for optimization and reliability assessment. In Section 2.8.1, a brief overview of the three RBDO frameworks is provided. This is followed by a brief overview of some RBDO implementations with different reliability assessment methods in Section 2.8.2.

2.8.1 Reliability-based design optimization (RBDO) frameworks

There are three major RBDO frameworks based on how the coupling between optimization and reliability assessment is implemented. The three frameworks,

namely Double-loop Method, Sequential Method and Single-Loop Method, are presented in the following sections.

Double-loop RBDO

The double loop method is the most rigorous among the three RBDO frameworks (Choi and Youn (2001)). The basic idea is shown in Figure 2.20. The selected optimizer starts from an initial sample or a set of samples depending on the method. Subsequent samples are selected by the optimizer based on the objective function and the probabilistic constraint information at previous samples. Determining the probabilistic constraint information requires reliability assessment at each step of the optimization, which in itself is an iterative process (Section 2.7). Thus, the double-loop method involves a nested reliability assessment loop within the optimization loop. This implies that the computation time for optimization and reliability assessment are multiplied. Therefore, although this method is rigorous, it is also the most expensive one.

Sequential RBDO

In sequential optimization (Du and Chen (2004)), the reliability assessment loop is decoupled from the optimization loop. The basic concept of sequential RBDO is depicted in Figure 2.21. Although both loops are present, they are performed in series and are not nested within each other. This is achieved by converting the probabilistic optimization into an equivalent deterministic optimization at each step, based on a "shifted" threshold on the responses. After an optimum is obtained by solving the deterministic optimization at each step, reliability assessment is performed at the optimum to verify the satisfaction of the probabilistic constraint.

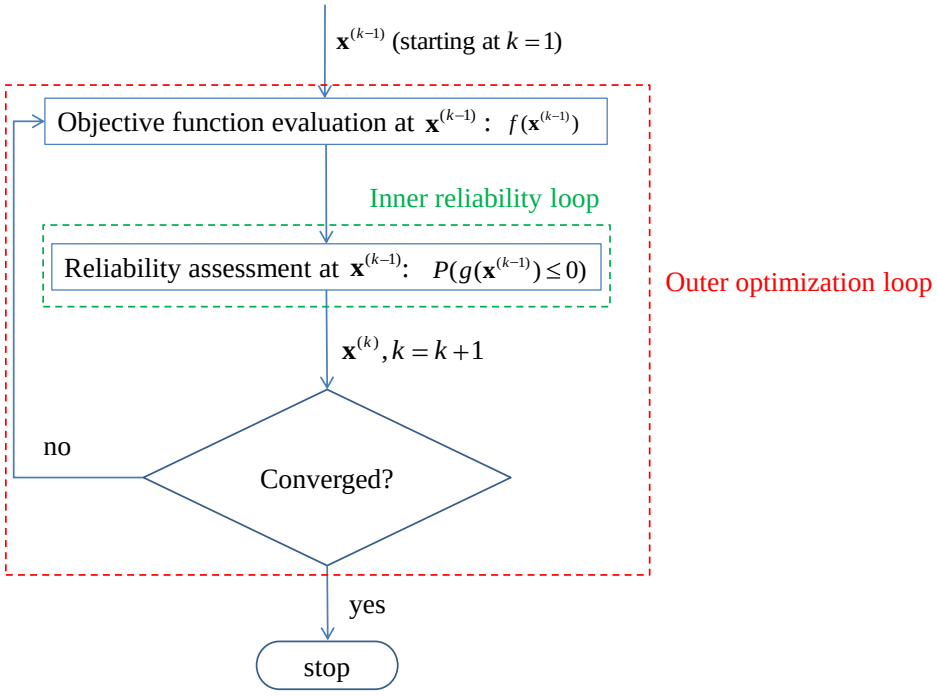

Figure 2.20: Double-loop RBDO.

Single-loop RBDO

Single-loop method (Agarwal (2004)) is the most efficient among the three RBDO frameworks. It is based on the idea of completely eliminating the reliability assessment loop. Instead of having two loops, one for optimization and the other for reliability assessment, the probability of failure is accounted for within the outer optimization problem using first order approximation. The RBDO formulation is:

$$\begin{aligned} \min_{\bar{\mathbf{x}}} \quad & f(\bar{\mathbf{x}}) \\ s.t. \quad & g(\mathbf{x}^*) \geq 0 \\ & \mathbf{x}^* = \bar{\mathbf{x}} - \beta_T \frac{\nabla g'(\mathbf{x}^*)}{||\nabla g'(\mathbf{x}^*)||} \approx \bar{\mathbf{x}} - \beta_T \frac{\nabla g'(\bar{\mathbf{x}})}{||\nabla g'(\bar{\mathbf{x}})||} \end{aligned} \tag{2.86}$$

where $\mathbf{x}^*$ is the inverse MPP, i.e. the point corresponding to MPP in the X-space. The approximation of $\mathbf{x}^*$ in Equation 2.86 assumes equal gradients at $\mathbf{x}^*$ and $\bar{\mathbf{x}}$. Therefore, the MPP can be calculated in a single step, using the constraint

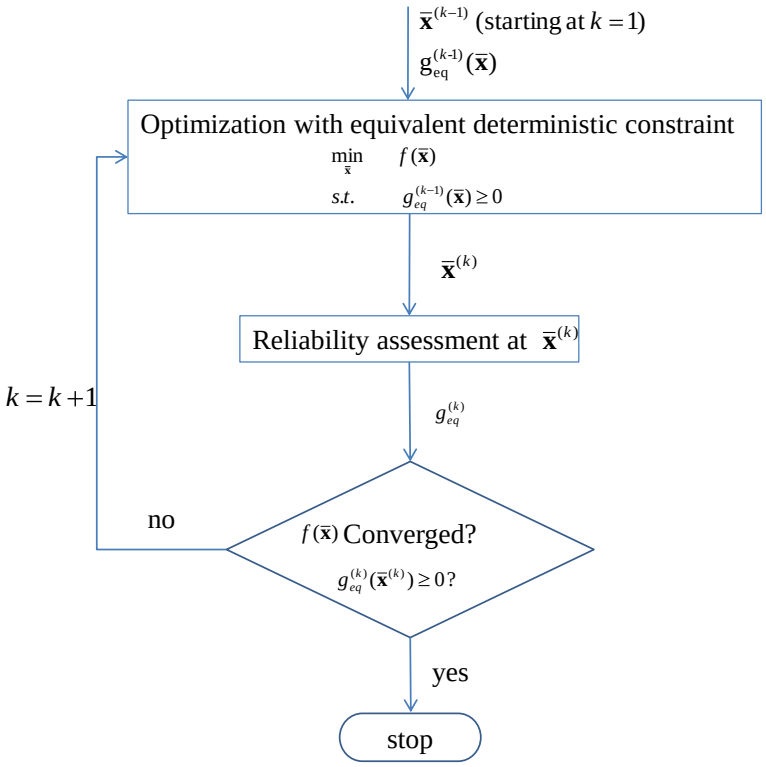

Figure 2.21: Sequential RBDO.

function and gradient information at the mean. As a result of the approximation, the inner loop for MPP search in reliability assessment is completely eliminated with the single-loop formulation. Although this approach can significantly reduce the computation cost for RBDO, it is limited in terms of its accuracy. The method is limited to first order reliability assessment, which is inaccurate in the general case.

2.8.2 Review of RBDO implementation methods

Several RBDO implementations are reported in the literature. These include implementations based on a single framework, as well as combinations of different frameworks. Except for the single-loop method, which is limited to first order reliability assessment, several different reliability assessment methods have been

implemented within RBDO frameworks.

Most of the double-loop and sequential RBDO methods found in literature require a search for the MPP. A major part of the computation time is required to located the MPP. Because RBDO involves repeated reliability assessment subproblems, efficiency of the MPP search is of critical importance in these methods. Two types of formulations are used to locate the MPP. The MPP search for reliability assessment is traditionally performed using Equation 2.41. The distance of the MPP to the mean (equal to the reliability index β) is minimized such that it lies on the zero-level of the limit state function. This approach is referred to as the reliability index approach (RIA) (Yu, X. et al. (1997)). In the context of RBDO, an alternate formulation, referred to as the inverse reliability approach or performance measure approach, is possible to locate the MPP (Youn, B.D. (2005)). It is based on the idea that the failure probability is usually active at the solution (i.e. $P_f = P_T$ or $\beta = \beta_T$). Therefore, the MPP search can be formulated as:

$$\begin{aligned} \min_{\mathbf{u}} \quad & g'(\mathbf{u}) = 0 \\ s.t. \quad & \beta = \beta_T \end{aligned} \tag{2.87}$$

Locating the MPP using PMA is more efficient than the RIA. Also, it is less prone to numerical instabilities. Both RIA and PMA implementations based on reliability assessment methods such as FORM, SORM, AMV, TANA etc. have been developed.

Although most methods are based on a specific RBDO framework, those with sequential combination of different frameworks have also been developed (Youn (2007)). Also, RBDO methods based on surrogate-based reliability assessment have also gained popularity to reduce computational cost and handle highly nonlinear limit state functions (Rais-Rohani and Singh (2004); Youn, B. and Xi, Z. (2009); Bichon, B.J. et al. (2009)). In Bichon, B.J. et al. (2009), the EGRA method presented in Section 2.7 was implemented within an RBDO framework to address multimodal functions. Other recent implementations include methods for handling correlated

input variables (Noh, Y. et al. (2009)). Methods for handling both discrete and continuous variables have also been developed (McDonald, M. and Mahadevan, S. (2008)). Dimensionality reduction methods have also been used for reducing the computation cost (Youn, B. and Xi, Z. (2009)).

2.9 Concluding remarks

An introduction to the basic concepts of optimization, reliability assessment and RBDO are presented in this chapter. An review of existing methods is also presented. The review is intended to cover the major ideas regarding variations in such methods, with the purpose of identifying areas that need improvement. Reliability assessment methods are classified as Mean Value, MPP-based, Sampling-based, Surrogate-based, and Classification-based methods. Several common limitations of the non-classification methods were identified, such as handling of discontinuous and binary responses, and multiple failure modes. In the context of optimization also, several methods are hampered by discontinuous and binary responses. Handling of multiple constraints is also a challenge if each constraint function is expensive to evaluate. In RBDO, all the difficulties associated with optimization and reliability assessment are coupled, making it a very challenging process. This dissertation proposes a new classification-based method to address many of the issues faced in current optimization, reliability assessment and RBDO methods.

CHAPTER 3

SUPPORT VECTOR MACHINES

Support vector machines (Vapnik, V.N. (1998); Shawe-Taylor, J. and Cristianini, N. (2004); Gunn, S.R. (1998)) are a class of machine learning techniques that can be used for classification or regression. In particular, SVMs have gained significant popularity as classification tools in the computer science community. They have widespread applications in pattern recognition, such as spam filtering, insurance decision making etc. As stated in Chapter 2, SVMs can be used in optimization and reliability assessment for approximating decision boundaries (failure domain boundaries or optimization constraints). The main feature of SVMs that makes them attractive for this research lies in their flexibility to define highly nonlinear decision boundaries that optimally separate two classes of samples. The purpose of this chapter is to provide an overview of the theory of SVMs for classification (Section 3.1). In addition, an introduction to probabilistic support vector machines (PSVMs) (Vapnik, V.N. (1998); Wahba, G. (1999); Platt, J.C. (1999)) is also provided. While an SVM provides binary classification of the space, a PSVM provides a probability of belonging to a specific class. The basic concept of PSVMs as well as a commonly used PSVM model are presented in Section 3.2.

3.1 Support vector machines as binary classifiers

This section presents an introduction to binary classification using SVMs. The basic SVM theory is derived for linearly separable data in Section 3.1.1. It is then generalized to the nonlinear case in Section 3.1.2.

3.1.1 Linear SVM boundary

To introduce SVM, we define a set of N training samples $\mathbf{x}_i$ in a m dimensional space. Each sample is associated with one of two classes characterized by a value $y_i = \pm 1$. In the case of linearly separable data, the two classes of samples can be separated using a hyperplane (straight line in two dimensional space). It should be noted that there are an infinite number of hyperplanes that can separate the samples (Figure 3.1).

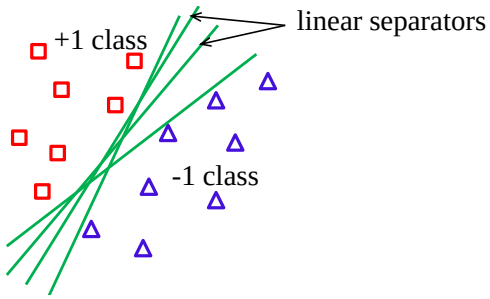

Figure 3.1: Multiple linear functions separating two classes.

The SVM algorithm finds the decision boundary that optimally separates the two classes of samples. In the linear case, the basic idea is to maximize the "margin" between two parallel hyperplanes that separate the data. This pair of hyperplanes is required to pass at least through one of the training points of each class, and there cannot be any points inside the margin (Figure 3.2). The points that these hyperplanes pass through are referred to as "support vectors". The optimum decision boundary is half way between these two previously described hyperplanes referred to as "support hyperplanes".

Equation of the linear SVM boundary is:

$$s(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b = 0 \tag{3.1}$$

where $\mathbf{w}$ is the vector of hyperplane coefficients, $\mathbf{x}$ is a point in space and b is the

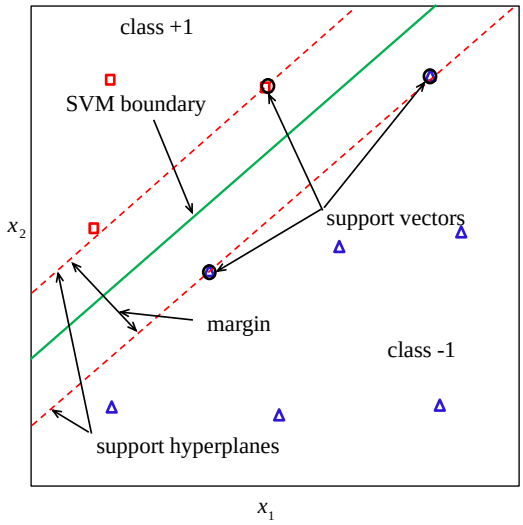

Figure 3.2: Linear SVM decision boundary.

bias. Equations of the two support hyperplanes are:

$$s(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b = \pm 1 \tag{3.2}$$

The perpendicular distance between the support hyperplanes or the margin is equal to $\frac{2}{||\mathbf{w}||}$. None of the samples can lie within the margin between support hyperplanes. This constraint can be represented as:

$$1 - y_i(< \mathbf{w}, \mathbf{x_i} > +b) \leq 0 \quad \forall i \in [1, N] \tag{3.3}$$

where $<,>$ is the inner product. The optimization problem to find the SVM boundary is:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}||\mathbf{w}||^2 \\ s.t. \quad & 1 - y_i(< \mathbf{w}, \mathbf{x}_i > +b) \leq 0 \qquad \forall i \in [1, N] \end{aligned} \tag{3.4}$$

The Lagrangian is given as:

$$\Phi(\mathbf{w}, b, \boldsymbol{\lambda}) = \frac{1}{2}||\mathbf{w}||^2 + \sum_{i=1}^{N} \lambda_i \left(1 - y_i(< \mathbf{w}, \mathbf{x}_i > +b)\right) \tag{3.5}$$

where λ_i are the Lagrange multipliers. It is easier to solve the dual formulation of Equation 3.4 (Gunn, S.R. (1998)):

$$\begin{aligned} \max_{\boldsymbol{\lambda}} \quad & \left(\min_{\mathbf{w},b} \Phi(\mathbf{w}, b, \boldsymbol{\lambda}) \right) \\ s.t. \quad & \boldsymbol{\lambda} \geq \mathbf{0} \end{aligned} \tag{3.6}$$

Minimization of the Lagrangian with respect to $\mathbf{w}$ and b provides the following relations:

$$\frac{\partial \Phi}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{N} \lambda_i y_i \mathbf{x}_i \tag{3.7}$$

$$\frac{\partial \Phi}{\partial b} = 0 \Rightarrow \sum_{i=1}^{N} \lambda_i y_i = 0 \tag{3.8}$$

Using Equations 3.6-3.5, the dual problem is given as (Gunn, S.R. (1998)):

$$\begin{aligned} \min_{\boldsymbol{\lambda}} \quad & \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j y_i y_j < \mathbf{x}_i, \mathbf{x}_j > - \sum_{k=1}^{N} \lambda_k \\ s.t. \quad & \lambda_i \geq 0 \\ & \sum_{i=1}^{N} \lambda_i y_i = 0 \quad \forall i \in [1, N] \end{aligned} \tag{3.9}$$

The SVM optimization is a Quadratic Programming (QP) that can be solved efficiently with available optimization packages. Following the Kuhn and Tucker conditions, only the Lagrange multipliers associated with the support vectors are strictly positive, while the rest are zero. Once the Lagrange multipliers are found, the optimal hyperplane coefficient vector $\mathbf{w}^*$ is calculated using Equation 3.7. Optimal value of the constant b is:

$$b^* = -\frac{1}{2} < \mathbf{w}^*, \mathbf{x}_+ + \mathbf{x}_- > \tag{3.10}$$

where $\mathbf{x}_+$ and $\mathbf{x}_-$ are any support vectors from $+1$ and -1 classes. The optimal SVM boundary is:

$$s(\mathbf{x}) = \sum_{i=1}^{NSV} \lambda_i y_i < \mathbf{x}_i, \mathbf{x} > + b = 0, \tag{3.11}$$

where NSV is the number of support vectors. In general, the number of support vectors is a small fraction of the total number of training points. The classification of any sample $\mathbf{x}$ is given by the sign of $s(\mathbf{x})$.

The SVM optimization problem may not always be feasible. The inequality constraints are then relaxed by the introduction of non-negative slack variables η_i which are minimized through a penalized objective function. The relaxed optimization problem is:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}||\mathbf{w}||^2 + C\sum_{j=1}^{N}\eta_j \\ s.t. \quad & 1 - \eta_i - y_i(<\mathbf{w},\mathbf{x}_i> + b) \leq 0 \qquad\qquad \forall i \in [1,N] \end{aligned} \tag{3.12}$$

where C is the penalty coefficient referred to as the misclassification cost. In the dual formulation, C becomes the upper bound for all the Lagrange multipliers (Gunn, S.R. (1998); Vapnik, V.N. (1998)):

$$\begin{aligned} \min_{\boldsymbol{\lambda}} \quad & \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_j y_i y_j <\mathbf{x}_i,\mathbf{x}_j> - \sum_{k=1}^{N}\lambda_k \\ s.t. \quad & 0 \leq \lambda_i \leq C \\ & \sum_{i=1}^{N}\lambda_i y_i = 0 \quad \forall i \in [1,N] \end{aligned} \tag{3.13}$$

3.1.2 Nonlinear SVM boundary

In Section 3.1.1, the SVM equation was derived for the linear case. In the general case, however, the training samples need not be linearly separable in the space $\{x_1, x_2, \ldots, x_m\}$. However, the samples can still be linearly classified in a higher dimensional space known as the "feature space". The dimensions or "features" of this space are denoted as $\{\phi_1(x_1), \phi_2(x_1), \ldots, \phi_{n-1}(x_m), \phi_n(x_m)\}$. Because the SVM is linear in feature space, the equations in Section 3.1.1 can be generalized to this

space. The dual problem to solve the SVM becomes:

$$\begin{aligned} \min_{\boldsymbol{\lambda}} \quad & \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_j y_i y_j < \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) > -\sum_{k=1}^{N}\lambda_k \\ s.t. \quad & 0 \leq \lambda_i \leq C \\ & \sum_{i=1}^{N}\lambda_i y_i = 0 \quad \forall i \in [1, N] \end{aligned} \tag{3.14}$$

It is however interesting to note that ϕ appears only within an inner product that results in a scalar function. Therefore, the inner product $< \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) >$ can be replaced with a kernel function. As a result, there is no need to solve for the SVM in the feature space, which can be high dimensional. Thus, the problem of finding the SVM becomes much simpler. This simplification is referred to as the "kernel trick". The optimization problem to find the SVM is:

$$\begin{aligned} \min_{\boldsymbol{\lambda}} \quad & \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{k=1}^{N}\lambda_k \\ s.t. \quad & 0 \leq \lambda_i \leq C \\ & \sum_{i=1}^{N}\lambda_i y_i = 0 \quad \forall i \in [1, N] \end{aligned} \tag{3.15}$$

The equation of SVM boundary is:

$$s(\mathbf{x}) = b + \sum_{i=1}^{NSV}\lambda_i y_i K(\mathbf{x}_i, \mathbf{x}) = 0 \tag{3.16}$$

where

$$\begin{aligned} < \mathbf{w}^*, \mathbf{x} > &= \sum_{i=1}^{NSV}\lambda_i y_i K(\mathbf{x}_i, \mathbf{x}) \\ b^* &= -\frac{1}{2}\sum_{i=1}^{NSV}\lambda_i y_i \left(K(\mathbf{x}_i, \mathbf{x}_+) + K(\mathbf{x}_i, \mathbf{x}_-)\right) \end{aligned} \tag{3.17}$$

It should be noted that it is not actually required to find $\mathbf{w}^*$ in order to obtain the SVM equation. The class of any sample is given by the sign of $s(\mathbf{x})$.

3.1.3 Kernel function and SVM parameters

In previous sections, the SVM equations for linear and nonlinear cases were presented. Two important quantities in the equation are the kernel function K and the misclassification cost C. The kernel function in Equation 3.16 can have several forms, such as polynomial, Gaussian radial basis function, splines, fourier series etc. The polynomial and Gaussian kernels are used in this dissertation. The polynomial kernel is given as:

$$K(\mathbf{x}_i, \mathbf{x}) = (1 + \langle \mathbf{x}_i, \mathbf{x} \rangle)^p \,, \tag{3.18}$$

where p is the degree of the polynomial kernel.

The Gaussian kernel is given as:

$$K(\mathbf{x}_i, \mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}\|^2}{2\sigma^2}\right) \tag{3.19}$$

where σ is the width parameter of the Gaussian kernel.

It is important to select appropriate values of the kernel parameters, e.g. degree of polynomial or width parameter of Gaussian kernel. A common method to select the kernel parameters is to use cross-validation techniques (Cawley and Talbot (2003)). Several cross-validation techniques exist, such as hold out, K-fold, and leave one out cross-validation (Cawley and Talbot (2003); Hamel (2009)). In K-fold cross-validation, the training samples are randomly partitioned into K sets. One out of the K sets is used for validation, and the remaining samples are used for training the SVM. The cross-validation process is then repeated K times, with each of the K sets used as the validation data once. This process is repeated for each candidate value of the kernel parameters. The kernel parameter value with the best validation results is then selected. In leave one out cross-validation, the validation set consists of only one sample and the rest $N - 1$ samples are used for training. The cross-validation process is repeated N times with each training sample as the validation data.

In this dissertation, another technique is used to select the kernel parameters. They are selected such that the boundary constructed is the "simplest" one without any training sample misclassification. For the polynomial kernel, this corresponds to the lowest degree polynomial that does not produce any training misclassification. For a Gaussian kernel it corresponds to the highest width parameter σ that does not produce training misclassification.

The misclassification cost C determines the penalty for violating the constraint of having an empty margin (Equation 3.3). A smaller value of C provides a larger margin. It can however lead to training misclassification. The parameter C can also be found using cross-validation. However, in this dissertation, C is set to infinity to avoid any training misclassification.

3.1.4 General features of SVM

An SVM has several features that make it useful for optimization and reliability assessment:

- It is multidimensional.
- It provides the optimal decision boundary by maximizing the margin.
- The boundary constructed using an SVM can be highly nonlinear and can represent several disjoint regions as well as multiple failure modes.

An example of three dimensional nonlinear SVM is demonstrated in Figure 3.3.

3.2 Probabilistic support vector machines (PSVMs)

Unlike deterministic SVMs, which assign a binary class label ± 1 to any point in the space, PSVMs provide the probability that a point will belong to the $+1$ class or

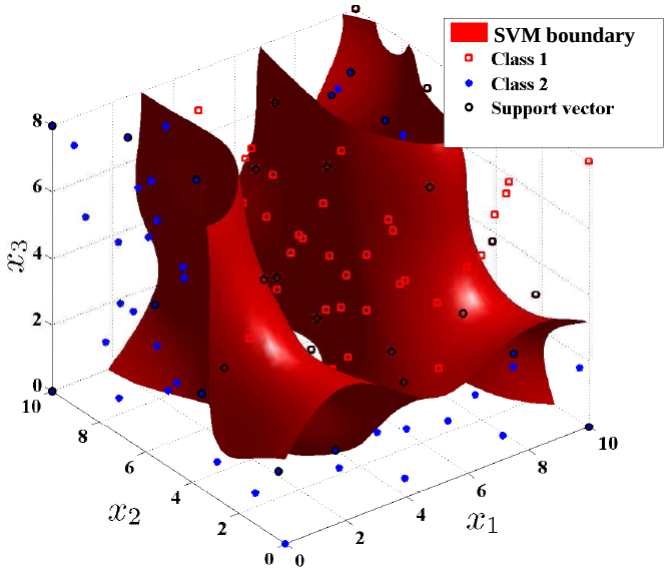

Figure 3.3: Three dimensional nonlinear SVM.

the -1 class. Thus, a PSVM also considers the variability in the construction of the SVM boundary $s(\mathbf{x}) = 0$ and quantifies the prediction error. That is, it provides the probability that a given point in the space will be misclassified by the SVM. An example of misclassification of the space by an SVM is shown in Figure 3.4.

An extensively used model for PSVM is based on the representation of probabilities using a sigmoid function (Platt, J.C. (1999)). The probability that a point $\mathbf{x}$ belongs to the $+1$ class is given by:

$$P\left(+1|\mathbf{x}\right) = \quad \frac{1}{1+e^{As(\mathbf{x})+B}} \tag{3.20}$$

The parameters $A(A < 0)$ and B of the sigmoid function are found by maximum likelihood, by solving the following problem (Platt, J.C. (1999)):

$$\min_{A,B} \quad -\sum_{i}^{N} t_i log\left(p_i\right) + \left(1-t_i\right) log\left(1-p_i\right), \tag{3.21}$$

$$\tag{3.22}$$

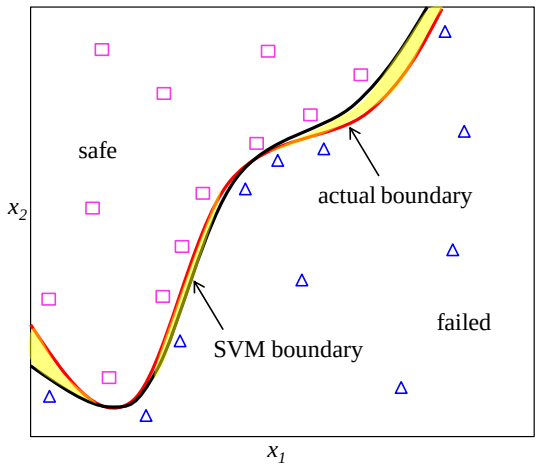

Figure 3.4: Misclassification of the space by an SVM. The shaded yellow regions are classified incorrectly by the SVM.

where N is the number of training samples, $p_i = P(1|\mathbf{x}_i)$ and t_i is given as:

$$t_i = \frac{y_i + 1}{2}, \tag{3.23}$$

where y_i are the class labels. Thus, $t = 1$ for the $+1$ samples and $t = 0$ for the samples belonging to the -1 class. Further details of solving the maximum likelihood problem can be found in Lin et al. (2007).

The basic sigmoid model presented in this section has some limitations that are overcome in a new PSVM model developed in this research. Further discussion on PSVMs is provided in Chapter 6. In Chapter 6, PSVMs are also used to provide a relatively conservative estimate of the probability of failure compared to deterministic SVM, to compensate for some of the consequences of an inaccurate SVM.

3.3 Concluding remarks

This chapter presents an introduction to the basic concepts of SVMs and PSVMs that are essential for this dissertation. They are used for constructing approximations of decision boundaries (failure boundaries and optimization constraints) in

the following chapters. The basic theory of constructing an SVM is presented for the linear case, before extending it to the nonlinear case. A commonly used sigmoid PSVM model is also presented that quantifies the prediction error of SVM. An improved PSVM model developed in this research will be presented in Chapter 6.

CHAPTER 4

EXPLICIT DESIGN SPACE DECOMPOSITION BASICS AND GLOBAL UPDATE OF SVMs

This chapter introduces the notion of explicit design space decomposition (EDSD) using SVMs, which is the fundamental idea around which all the methodologies developed in this dissertation revolve. A short introduction to classification-based methods for defining decision boundaries was presented in Chapter 2. The fundamentals of SVM classification were also presented in Chapter 3. Several features of SVMs were presented, such as their ability to optimally classify high dimensional data, definition of highly nonlinear boundaries etc. These features make it a flexible tool for defining limit state functions or constraint boundaries, together referred to as "decision boundaries". The process of constructing an explicit boundary that separates the space into distinct regions, e.g. failed and safe, or feasible and infeasible, is referred to as explicit design space decomposition (EDSD). The basic steps of constructing explicit decision boundaries using SVMs are presented in Section 4.1. This is followed by a brief introduction to the procedure of performing reliability assessment and optimization using explicit SVM boundaries (Sections 4.2 and 4.3). As was mentioned in Chapter 2, a critical research issue in any reliability assessment or optimization method is to limit the computation cost. For the same purpose, adaptive sampling methods have been developed in this research to construct SVM boundaries. Section 4.4 presents an adaptive sampling method that refines the SVM boundaries globally. In order to validate the developed method, analytical test examples with known solutions are presented in Section 4.5. Two application examples are also presented to show the usefulness of the approach. The first one consists of nonlinear buckling of an arch, with discontinuous responses, and the other involves tolerance optimization of a multibody system with several failure modes.

4.1 Basic explicit design space decomposition (EDSD) methodology using SVMs

The basic idea in EDSD, as the name suggests, is to decompose the space into regions of distinct behaviors using explicit decision boundaries. The boundary may represent a limit state function, in the context of reliability assessment, or the zero-level contour of optimization constraints. The use of SVMs is proposed for constructing the boundaries, because they have the ability to optimally classify highly nonlinear data. The main steps of EDSD are as follows.

- *Design of Experiments:* The first step is to sample the space using a DOE (Montgomery, D.C. (2005)). In order to extract information over the entire space, a uniform design of experiments such as Centroidal Voronoi Tessellations (CVT) (Romero, Vincente J. et al. (2006)), Latinized Centroidal Voronoi Tessellations (LCVT) or Optimal Latin Hypercube Sampling (OLHS) (Liefvendahl, M. and Stocki, R. (2006)) can be used. CVT and LCVT DOEs are used in this research. Examples of CVT DOEs with different number of samples in a two-dimensional space are shown in Figure 4.1.

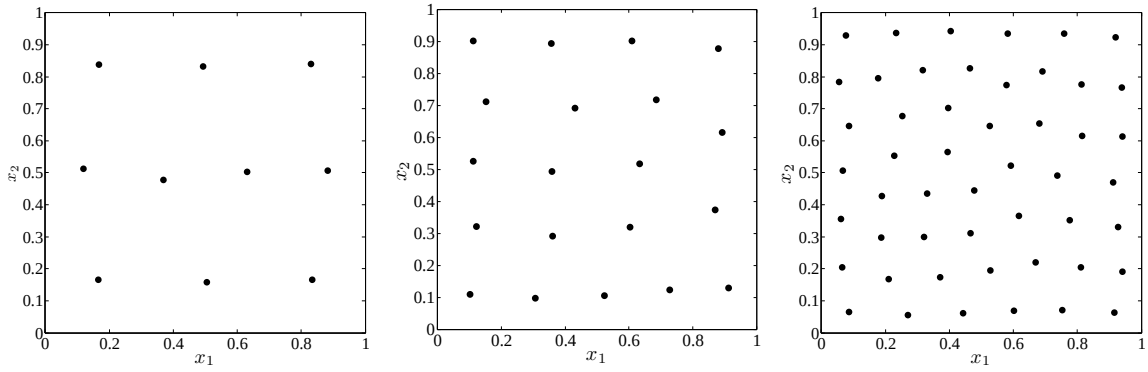

Figure 4.1: Example of two-dimensional CVT DOEs with 10, 20 and 50 samples.

- *Response Evaluation at DOE Samples:* At each of the DOE samples, system responses are evaluated. The evaluation may be performed using a computer code, e.g. finite element analysis, or through experiments (e.g. crash testing).

- *Response Classification at DOE Samples:* Once the system responses are available at the DOE samples, the samples are classified based on these values, e.g. safe and failed. The classification can be performed using a threshold response or using a clustering technique, such as K-means (Hartigan, J.A. and Wong, M.A. (1979)). Clustering is required when the system responses are discontinuous and there is no prior knowledge of a threshold response. It should be noted that the clustering is uni-dimensional, and is based on the response values. In the case of binary data, the classification information is available directly. Examples of classification using threshold response and clustering are depicted in Figure 4.2.

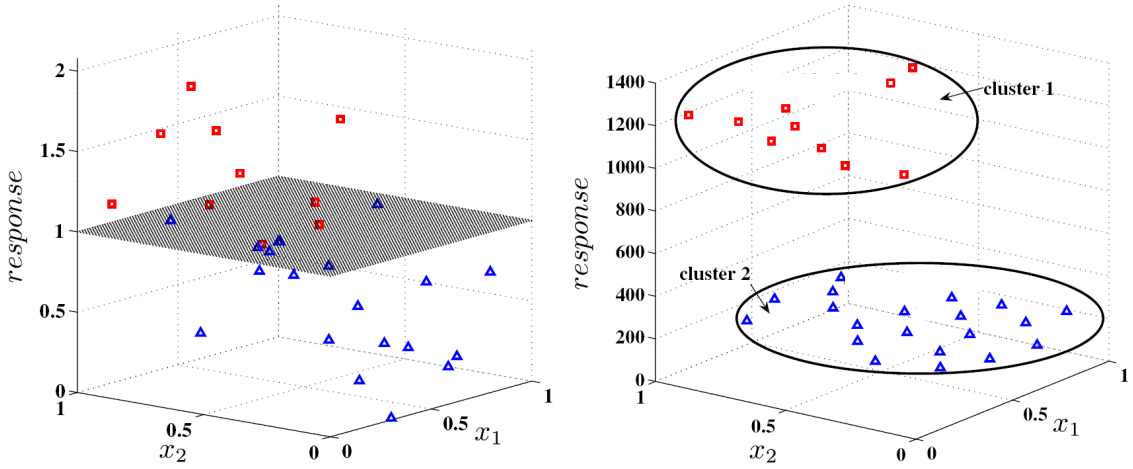

Figure 4.2: Example of classification using threshold response (left) and clustering (right).

- *Construction of the explicit boundary that separates the two classes:* An explicit boundary separating the samples belonging to distinct classes is constructed. In this work, the boundaries are constructed using SVMs (Chapter 3), due to their flexibility in defining highly nonlinear boundaries.

A summary of the basic SVM-based EDSD method is provided in Figure 4.3.

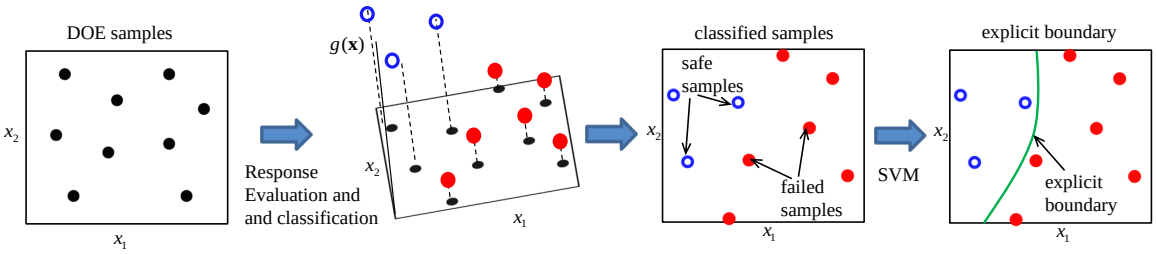

Figure 4.3: Summary of explicit design space decomposition using SVM.

4.2 Reliability assessment using explicit SVM boundaries

The methodology for constructing SVM boundaries separating two classes of samples was presented in Section 4.1. Once an SVM is constructed it provides an analytical approximation of the boundary separating the samples (Equation 3.16), which is the failure domain boundary in the context of reliability assessment. Because an analytical approximation of the failure domain boundary is available, any of the reliability assessment methods presented in Section 2.7.1 can be used to calculate the probability of failure. The most straightforward method is to perform Mote Carlo simulations (MCS) (Equation 2.51). Because an analytical approximation is available, a large number of Monte Carlo samples can be used. The class of any Monte Carlo sample ($\mathbf{x}_i$) is obtained using the sign of SVM value $s(\mathbf{x}_i)$ (Figure 4.4). Thus, the probability of failure is calculated as:

$$P_f = \frac{1}{N_{MCS}} \sum_{i=1}^{N_{MCS}} I_g(\mathbf{x}_i)$$

$$I_g(\mathbf{x}) = \begin{cases} 1 & s(\mathbf{x}) \leq 0 \\ 0 & s(\mathbf{x}) > 0 \end{cases} \tag{4.1}$$

For low probabilities of failure, methods such as importance sampling, MCMC, subset simulations etc. can be used.

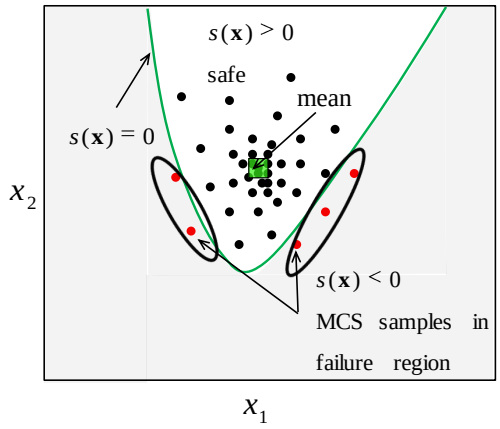

Figure 4.4: MCS based on SVM approximation of failure boundary.

4.3 Optimization using explicit SVM boundaries

Similar to failure boundaries in the context of reliability assessment, an SVM can also provide an analytical approximation of optimization constraints (Figure 4.5). In the context of multiple constraint problems, a single SVM can be used to represent the feasible space. Considering the case when the objective function is easy to evaluate (e.g. weight of a structural design), it is straightforward to perform optimization using the constraint zero-level contour approximation given by SVM. Any of the methods in Section 2.6 can be used. A method to handle cases where the objective function is also expensive is presented in Chapter 7.

For reliability-based design optimization, two types of approximations are required. One is the approximation of the failure boundary, separating safe and failed samples, and the other approximation is for the probabilistic constraint to identify the regions with allowable probability of failure (i.e. $P_f \leq P_T$). First, the explicit failure boundary is obtained using the method in Section 4.1. Once the SVM approximation for failure boundary is constructed, it allows the efficient calculation of probability of failure at any point in the space (Section 4.2). Therefore, the probability of failure can be calculated at a large number of samples. The next step is to

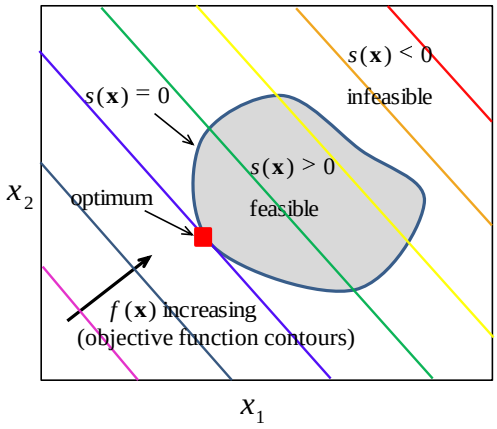

Figure 4.5: Optimization using SVM approximation of the zero-level of constraint function.

approximate the probabilistic constraint. There are two alternatives to do that:

- *Approximation method:* The probability of failure values at the large sample set are converted to the corresponding reliability index β values using the inverse of Equation 2.33:

$$\beta = -\Phi^{-1}(P_f) \tag{4.2}$$

  This is done because the variation of β is usually smoother than that of the corresponding P_f values. The β values are then fitted to a response surface or metamodel (Section 2.4). For example, Kriging or support vector regression (SVR) can be used.

- *Classification method:* The second alternative is to classify the large sample set into two categories based on the values of P_f calculated using Equation 4.1. The samples with $P_f \geq P_T$ are labeled as $+1$ and the ones with $P_f < P_T$ are labeled as -1. A second SVM boundary is then constructed that separates these two classes. This boundary approximates the zero-level contour of the probabilistic constraint. The procedure is discussed in more detail in Chapter 5.

4.4 Adaptive sampling for construction of accurate SVM boundaries

The basic EDSD method presented in Section 4.1 was based on a static DOE, and therefore, the SVM boundary may not provide an accurate approximation of the actual decision boundary, unless a large number of samples is used. Therefore, adaptive sampling is an important part of EDSD, as it is required to provide high accuracy of the approximated decision boundary with limited number of samples. A global update strategy to refine SVM boundaries is presented in this section. An initial SVM is constructed using a relatively sparse DOE. It is then updated using adaptively selected samples, until convergence. Two types of samples, referred to as primary and secondary samples, are used in the update. The overall methodology for constructing the boundaries is presented in Algorithm 4.1. For the sake of clarity, details of the scheme are presented in subsequent sections.

Algorithm 4.1: Methodology for global update of SVM boundaries

---

1: Sample the space with a CVT DOE.
2: Evaluate the system response at each sample (e.g. using a finite element code).
3: Classify the samples into two classes (e.g. safe and failed) based on the response values. The classification is performed using a threshold value or a clustering technique.
4: Set iteration $k = 0$
5: Select the parameters for constructing the SVM boundary (Section 3.1.3).
6: Construct the initial SVM boundary that separates the classified samples.
7: **repeat**
8: $k = k + 1$
9: Select a primary sample on the SVM boundary (Section 4.4.1) and reconstruct the SVM with the new information.

10: Re-execute line 9 to select another sample.

11: Select a secondary sample to prevent locking of the SVM (Section 4.4.1). Modify the SVM parameters (Section 3.1.3) and reconstruct the SVM boundary.

12: Calculate the convergence measure Δ_k.

13: **until** $\Delta_k \leq \delta_1$

4.4.1 Selection of samples to update the SVM

Details of sample selection for the update are presented in this section. As already mentioned in previous section, two types of samples are used. Details of these samples and the motivations for their selection are presented in this section.

Primary samples on the SVM boundary

The first type of samples used in the update are referred to as "primary samples". These samples are selected such that they lie on the SVM boundary in regions of space that are sparsely populated. The motivations of this choice are as follows:

- A sample on the boundary has high probability of misclassification. Such a sample may belong to either one of the two classes.
- It lies in the margin of SVM and, therefore, compels it to change. By construction, there cannot be any sample within the SVM margin.
- The selection of samples in sparsely populated regions avoids redundancy of data.

Two primary samples are selected at each iteration, in order to have higher frequency compared to secondary samples. The selection of a primary sample is performed by maximizing the distance to the closest training sample while lying on the SVM boundary (Equation 4.3). Figure 4.6 shows the selection of a primary

sample and the SVM boundary update due to it. The optimization problem to select a primary sample is:

$$\begin{aligned} \max_{\mathbf{x}} \quad & d_{min}(\mathbf{x}) \\ s.t. \quad & s(\mathbf{x}) = 0 \end{aligned} \tag{4.3}$$

where $d_{min}(\mathbf{x})$ is the distance to the closest training sample. The maxmin problem in Equation 4.3 is non-differentiable at the boundaries where the closest sample to $\mathbf{x}$ switches. The problem is made differentiable by reformulating it as:

$$\begin{aligned} \max_{\mathbf{x},z} \quad & z \\ s.t. \quad & ||\mathbf{x} - \mathbf{x}_i|| \geq z \quad \forall i \in [1, N] \\ & s(\mathbf{x}) = 0 \end{aligned} \tag{4.4}$$

Finding the maxmin distance sample is a global optimization problem. However, for the results in Section 4.5, the differentiable formulation of the global optimization problem (Equation 4.4) is solved using a local optimizer (sequential quadratic programming). Multiple starting locations given by the existing training samples are used for the optimization. It is also possible to use a global optimization method such as GA or branch and bound (Weise (2009)). GA can be used with the formulation in Equation 4.3, as it does not require differentiability of the functions. An implementation of GA is available in Matlab. Implementation of branch and bound method is available in a software referred to as DIRECT (Finkel (2003)).

Secondary samples to prevent locking of the SVM

In addition to primary samples, another sample, referred to as a "secondary sample" is evaluated at each iteration. Secondary samples are evaluated to prevent a phenomenon referred to as "locking" of the SVM, and to improve the convergence of the SVM. Although selection of samples on the SVM boundary compels it to

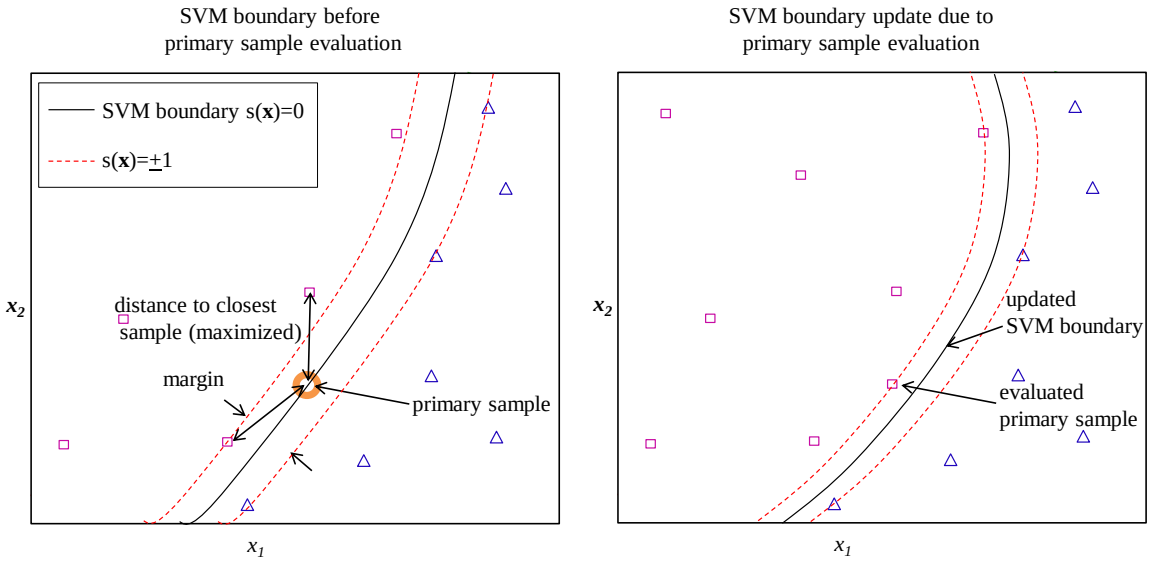

Figure 4.6: Selection of a new training sample on the SVM boundary while maximizing the distance to the closest sample. The right figure shows the updated SVM decision boundary.

change, the extent of this change may vary. Modification of SVM boundary due to a primary sample may be negligible if the margin (loosely, the local distance between $s(\mathbf{x}) = \pm 1$) is thin, thus wasting function evaluations. When locating a sample on the SVM within a thin margin, which by construction should not contain any sample, the change in boundary due to the update is inevitably small. If this small change occurs in a region with a relatively uniform amount of information from both classes in the vicinity of the added sample, then the SVM can be assumed to be locally accurate. However, if the data from one class is sparse, then the slow convergence rate becomes an issue. This is referred to as the "locking" of the SVM (Figure 4.7).

As a result of the locking phenomenon, the SVM boundary may not converge to the actual one with reasonable amount of data in certain localized regions. Therefore, in addition to primary samples selected on the SVM boundary, a secondary sample directed specifically at the prevention of SVM locking is evaluated. A

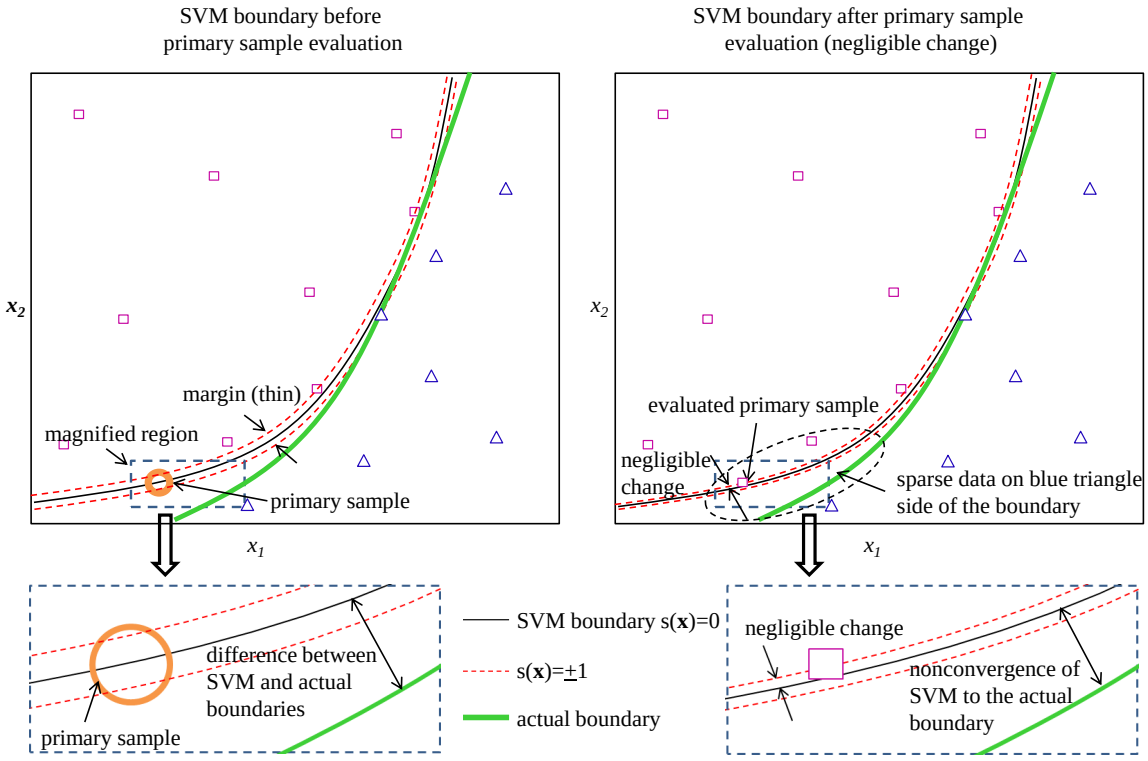

Figure 4.7: Locking of the SVM due to reduction of the margin (region between dashed red curves). A new sample on the boundary (brown circle), belonging to the magenta square class, produces negligible change although there is no sample belonging to the opposite class (blue triangles) nearby.

potential area for locking is identified as one in which data from one class is sparse in the vicinity of the boundary. If a secondary sample selected in such a region is misclassified by the current SVM then it results in significant change of the SVM boundary (Figure 4.8).

Selection of the sample is a two step process:

- Selection of center $\mathbf{x}_c$ and radius R of a hypersphere.
- Selection of a secondary sample within the hypersphere.

Two possible choices are presented to select a secondary sample, along with

their merits and demerits:

Secondary sample method 1: In this method (Figure 4.8), the center $\mathbf{x}_c$ is selected as the support vector farthest from existing samples of opposite class. The objective of secondary sample is to locate regions with nonuniform data belonging to the two classes in the vicinity of the SVM boundary. The motivation of selecting the support vectors as center is that it is known that these samples lie close to the boundary. Therefore, if the distance to the closest opposite class sample from a support vector is large, then it represents locally nonuniform data in the vicinity of the boundary. Radius of the hypersphere is given as:

$$R = \frac{1}{2} ||\mathbf{x}_c - \mathbf{x}_{opp}|| \tag{4.5}$$

where $\mathbf{x}_{opp}$ is the closest sample to $\mathbf{x}_c$ belonging to the opposite class. The secondary sample is selected within the hypersphere with center $\mathbf{x}_c$ and radius R. The sample is chosen so that it belongs to the opposite class of the support vector according to the current SVM prediction:

$$\begin{aligned} \min_{\mathbf{x}} \quad & s(\mathbf{x}) y_c \\ s.t. \quad & ||\mathbf{x} - \mathbf{x}_c|| \leq R \\ & s(\mathbf{x}) y_c \leq 0 \end{aligned} \tag{4.6}$$

where y_c is the class label (± 1) of $\mathbf{x}_c$. The objective function in Equation 4.6 also appears as a constraint in order to avoid an optimum solution with a positive objective function value, i.e. to avoid a solution for which the current SVM provides the same class for the support vector and the secondary sample. The optimization is solved using SQP starting from the center $\mathbf{x}_c$. If no feasible solution is found for Equation 4.6, the support vector with next highest value of R is selected as the center. It is noteworthy that the choice of R as one half of $||\mathbf{x}_c - \mathbf{x}_{opp}||$ will prevent the sample from being chosen too close to regions with existing samples, such as $\mathbf{x}_{opp}$ itself.

The limitation of selecting a secondary sample in this manner is that it is constrained to lie in regions surrounding the support vectors. Thus, it may ignore other regions with nonuniform distribution of samples. This limitation is overcome in a modified method to select secondary samples presented below. In this modified method, referred to as method 2, the center is not constrained to be chosen from the support vectors. However, an optimization problem needs to be solved to find the center. An advantage of selecting the center from the support vectors is that solving for the secondary sample is very efficient.

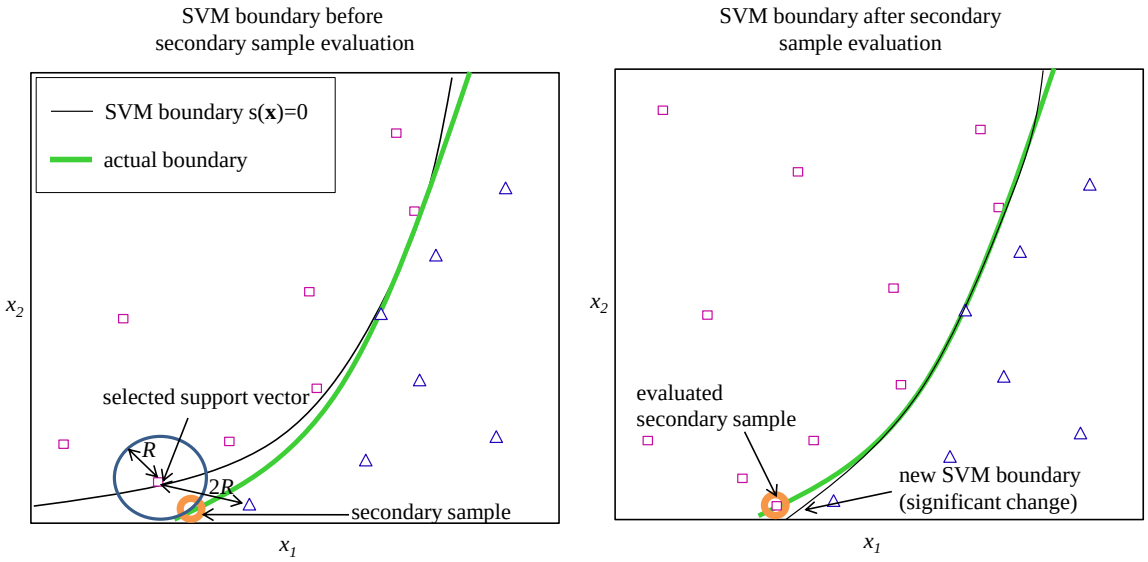

Figure 4.8: Evaluation of a secondary sample selected using method 1 to prevent locking of the SVM boundary.

Secondary sample method 2: In this improved method (Figure 4.9), the center $\mathbf{x}_c$ is not constrained to be chosen from the support vectors. Instead, it is located on the SVM boundary by solving an optimization problem:

$$\max_{\mathbf{x}} \quad (d_-(\mathbf{x}) - d_+(\mathbf{x}))^2$$
$$s(\mathbf{x}) = 0, \tag{4.7}$$

where $d_+(\mathbf{x})$ and $d_-(\mathbf{x})$ are the distances to the closest $+1$ and -1 samples. The optimization can be solved using SQP with multiple starting points given by the existing samples or using a global optimizer such as GA or DIRECT. Radius of the hypersphere R is proportional to the measure of unbalance $|d_-(\mathbf{x}) - d_+(\mathbf{x})|$:

$$R = \frac{1}{4}\,|d_-(\mathbf{x}_c) - d_+(\mathbf{x}_c)| \tag{4.8}$$

The value of radius is selected based on the idea that if $\mathbf{x}_c$ and the two closest opposite class samples are collinear then the mid point of these two samples lies at a distance $2R$ to the center $\mathbf{x}_c$. The secondary sample is selected within the hypersphere by solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \text{sign}(d_-(\mathbf{x}_c) - d_+(\mathbf{x}_c))s(\mathbf{x}) \\ & ||\mathbf{x} - \mathbf{x}_c|| - R \leq 0 \end{aligned} \tag{4.9}$$

Because the center $\mathbf{x}_c$ can lie anywhere on the SVM boundary, it does not ignore any region. However, unlike method 1, finding $\mathbf{x}_c$ requires solution of an optimization problem. In the context expensive function evaluations, however, the time required to locate the sample is just an overhead.

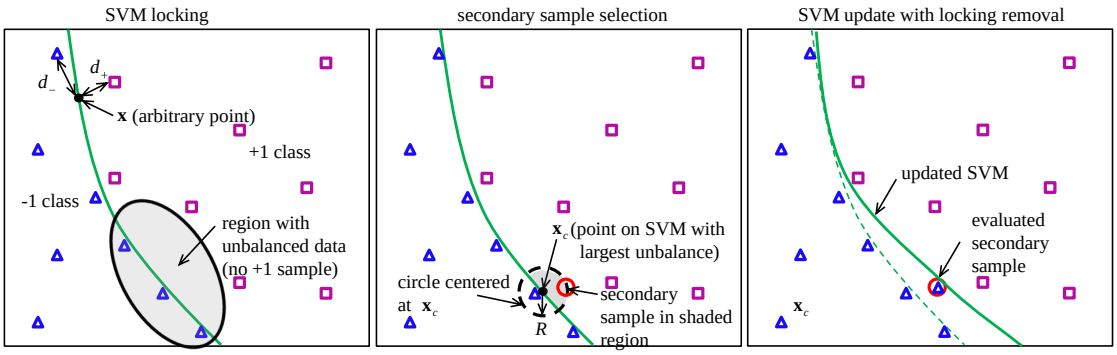

Figure 4.9: Locking of SVM with locally unbalanced data in the vicinity of SVM (left), selection of a secondary sample (mid), and update of SVM due to the evaluated secondary sample (right).

4.4.2 Convergence criterion

Because the actual function is not known in general, the convergence criterion for the update algorithm is based on the variation of the approximated SVM boundary between two consecutive iterations. Two types of convergence measures are presented in this section.

Polynomial coefficient based convergence measure

For the polynomial kernel used in most of the examples presented in this chapter, a rigorous quantification of the variation is possible based on the coefficients. In order to compare the polynomials at iterations $k-1$ and k, the coefficients are scaled such that the largest coefficient (absolute value) at iteration $k-1$ is 1. The corresponding coefficient for iteration k is also set to 1. The calculation of the convergence measure is implemented as follows:

- *Find the polynomial coefficients:* In order to find the polynomial coefficients a linear system of equations is solved. For a m-dimensional problem and a polynomial kernel of degree p, the number of coefficients is $\binom{m+p}{p}$. In order to find the coefficients, a set of $\binom{m+p}{p}$ points is selected from a CVT distribution and the corresponding SVM values are calculated. The coefficients are obtained as:

$$\boldsymbol{\alpha} = \mathbf{Q}^{-1}\mathbf{s} \tag{4.10}$$

where $\mathbf{s}$ is the array of SVM values. The i^{th} row of the matrix $\mathbf{Q}$ is given as:

$$\mathbf{R}_i = \left(1 \;\; x_1 \;\; x_2 \;\; \ldots \;\; x_d \;\; \ldots \;\; \ldots \;\; x_1^p \;\; (x_1^{p-1}x_2) \;\; \ldots \;\; x_m^p\right)\Big|_{\mathbf{x}_i} \tag{4.11}$$

Thus, the matrix $\mathbf{Q}$ is a square matrix of size $\binom{m+p}{p} \times \binom{m+p}{p}$. Note that the matrix $\mathbf{Q}$ is invertible and well conditioned as the samples to construct it are uniformly distributed with CVT.

Note that the coefficients could also be calculated using multinomial expansion (Ma, N. (2001)) and Equation 3.16. The coefficient of a general term $x_1^{p_1} x_2^{p_2} \dots x_m^{p_m}$, except for the constant term, in the SVM equation is given as:

$$\alpha_{p_1 p_2 \dots p_m} = \frac{p!}{\prod_{j=0}^{m} p_j!} \sum_{i=1}^{N} \left(\lambda_i y_i \prod_{j=1}^{m} x_j^{p_j} \right) \Bigg|_{\mathbf{x}_i} \quad \text{where} \quad \sum_{j=0}^{m} p_j = p \tag{4.12}$$

The constant term in the SVM equation is equal to b (Equation 3.16).

- *Comparison of the coefficients between iterations:* In order to compare the coefficients between successive iterations $k-1$ and k, the coefficients corresponding to different degrees are separated into distinct arrays. The array of coefficients corresponding to degree r is denoted as $\boldsymbol{\alpha}_r$. The evolution of the coefficients is studied separately for each degree (Figure 4.12). The reason for studying each degree separately is that an identical relative change for two coefficients, especially for the largest and smallest degrees, may not lead to the same change in the boundary. The relative change in the norm of $\boldsymbol{\alpha}_r$ is calculated for each degree and the maximum value is used as a measure of convergence. The convergence measure is given by:

$$\Delta_k = \max_r \left(\Delta_k^{(r)} \right) \tag{4.13}$$

where $\Delta_k^{(r)}$ is given as:

$$\Delta_k^{(r)} = \frac{\left|\left| \boldsymbol{\alpha}_r^{(k)} - \boldsymbol{\alpha}_r^{(k-1)} \right|\right|}{\left|\left| \boldsymbol{\alpha}_r^{(k-1)} \right|\right|} \tag{4.14}$$

Convergence point based measure

The convergence measure in previous section can be used for polynomial kernels. For other kernels, another measure based on "convergence points" may be used. For this purpose, a set of N_{conv} "convergence points" is generated using an LHS DOE. The fraction of convergence points for which there is a change of sign of $s(\mathbf{x})$ between two successive iterations is calculated. The number N_{conv} can be chosen to be quite

high because the calculation of SVM values using Equation 3.16 is inexpensive. Because the convergence points are generated using LHS, the generation of these samples is efficient. As a general rule, 10×5^m convergence points are used. By choosing a large set of convergence points, Equation 4.15 can be used to achieve an accurate estimate of the fraction up to a few dimensions, beyond which filling the space becomes impossible.

$$\Delta_k = \frac{\text{num}(|\text{sign}(\mathbf{s}_k) - \text{sign}(\mathbf{s}_{k-1})| > 0)}{N_{conv}} \tag{4.15}$$

where Δ_k is the fraction of convergence points for which the sign of the SVM evaluation changes between iterations $k-1$ and k. $\mathbf{s}_k$ and $\mathbf{s}_{k-1}$ represent vectors of SVM values at the convergence points at iterations k and $k-1$.

In order to implement a convergence criterion, the fraction of convergence points changing sign between successive iterations is fitted by an exponential curve:

$$\hat{\Delta}_k = Ae^{Bk} \tag{4.16}$$

where $\hat{\Delta}_k$ represents the fitted values of $\hat{\Delta}_k$. A and B are the parameters of the exponential curve. The value of $\hat{\Delta}_k$ at the last iteration k_c is checked after each training sample is added. The slope of the curve is also calculated. For the update to stop, the value of the fitted curve 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.

$$\begin{aligned} Ae^{Bk_c} &< \epsilon_1 \\ -\epsilon_2 < BAe^{Bk_c} &< 0 \end{aligned} \tag{4.17}$$

The convergence point based measure can be used for any type of kernel. However, it can only be used up to a few dimensions due to restrictions on filling the space, as already mentioned.

4.4.3 Error measures

The accuracy of an approximated SVM boundary is judged by its fidelity to the actual function. In practical problems, an error metric may not be available. However, error measures can be obtained in the case of academic analytical test functions. Two distinct error metrics are presented:

- *Based on "test" points:* The error may be quantified as the fraction of the spatial volume which is misclassified by the SVM boundary. For this purpose, a set of N_{test} uniformly distributed "test" points is generated to densely sample the whole space. The values of both the actual function and the SVM are calculated for each test point. Since the actual function is analytical, these function evaluations are efficiently performed. The number of test points being much larger than the number of sample points, the error can be assessed by calculating the fraction of misclassified test points (Basudhar, A. and Missoum, S. (2008)). A test point for which the SVM and the actual function provide different signs is considered misclassified. The error ϵ_k is given below:

$$\epsilon_k = \frac{\text{num}\left(s(\mathbf{x}_{test})y_{test} \leq 0\right)}{N_{test}} \tag{4.18}$$

  where $\mathbf{x}_{test}$ and y_{test} represent a test sample and the corresponding class value (± 1) based on the actual (known) decision function.

- *Based on polynomial coefficients of the SVM boundary:* ϵ_k is a good measure of the fraction of misclassified space if the space is sampled densely, but the approach is limited to a few dimensions due to constraints on computational resources. However, a measure based on polynomial coefficients is possible for actual decision boundaries represented by polynomials. The relative error E_k is given by:

$$E_k = \frac{\left|\left|\boldsymbol{\alpha}^{(act)} - \boldsymbol{\alpha}^{(k)}\right|\right|}{||\boldsymbol{\alpha}^{(act)}||} \tag{4.19}$$

  where $\boldsymbol{\alpha}^{(act)}$ is the array of the polynomial coefficients for the actual function.

4.5 Examples

Several test examples demonstrating the efficacy of the SVM-based EDSD method and the update methodology are presented. To validate the method, it is used to reconstruct highly nonlinear boundaries given by known analytical functions. The analytical decision functions are written in the form $g(\mathbf{x}) = 0$. In order to perform the SVM classification, the samples corresponding to $g(\mathbf{x}) > 0$ and $g(\mathbf{x}) < 0$ are labeled +1 and -1 respectively.

In Section 4.5.1, the application of the update scheme to high dimensional problems with up to seven variables is presented. The evolution of the polynomial coefficient-based convergence and error measures during the update are shown. Section 4.5.2 presents an example of SVM locking. In order to show the ability of secondary samples to remove SVM locking, a comparison of adaptive sampling with and without secondary sample evaluation is provided for this example. In both Examples 1 and 2, the method 1 of selecting secondary samples has been used.

In addition to the analytical test problems, an example with nonlinear buckling of an arch structure is presented in Section 4.5.3 to demonstrate the application of the SVM-based EDSD method to discontinuous responses. Also, an example with multiple failure modes for a multibody system is presented in Section 4.5.4. The results for the two application examples were generated using an earlier version of the update (Basudhar, A. and Missoum, S. (2008)) that did not have secondary samples. The stopping criteria for Examples 4.5.3 and 4.5.4 are based on convergence points.

The following notation will be used to present the results:

- $N_{initial}$ is the initial training set size.
- N_{total} is the total number of samples.

- $\epsilon_{initial}$ and ϵ_{final} are the test point-based errors associated with the initial and final SVM decision boundaries respectively.
- $E_{initial}$ and E_{final} are the errors associated with the initial and final SVM decision boundaries respectively, based on the comparison with the coefficients of the actual functions.

4.5.1 Example 4.1: Global update for high dimensional problems

This section presents the application of the update scheme to three analytical test functions of different dimensionality that are derived from the same general equation. The functions presented consist of three, five and seven variables, and represent non-convex and disjoint regions. The general equation written as a function of the dimensionality m is:

$$g(\mathbf{x}) = \sum_{i=1}^{m} (x_i + 2\beta)^2 - 3 \sum_{j=1}^{m-2} \prod_{l=j}^{j+2} x_l + 1$$
$$\beta = -1 \qquad \text{mod}(i,3) = 1$$
$$\beta = 0 \qquad \text{mod}(i,3) = 2$$
$$\beta = 1 \qquad \text{mod}(i,3) = 0 \tag{4.20}$$

For example, the decision function in a three-dimensional case (Equation 4.21) is obtained by substituting $m = 3$ in the general equation. The actual boundary (decision function) for the three-dimensional case is plotted in Figure 4.10. It forms several disjoint regions in the space. The failure domain boundary for this case is:

$$g(\mathbf{x}) = (x_1 - 2)^2 + x_2^2 + (x_3 + 2)^2 - 3x_1x_2x_3 + 1 = 0 \tag{4.21}$$

The polynomial kernel is used to construct the SVM boundary in each of the examples. The degree of the polynomial is automatically selected as explained in Section 3.1.3. As evident from Equation 4.20, the actual decision functions are polynomials of degree 3. It is observed that for all the examples the algorithm automatically selects a polynomial kernel of degree 3 to construct the SVM boundary.

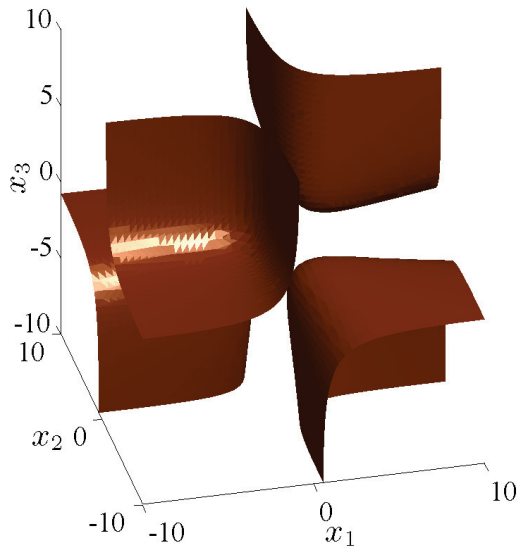

Figure 4.10: Three-dimensional problem with disjoint regions. Actual decision boundary.

Starting from relatively small CVT DOEs, the update algorithm is run up to a fixed number of iterations for each of the examples to study the evolution of the error and convergence properties of the algorithm. No actual convergence threshold is set for these problems. The initial and final values of the error measure ϵ_k are calculated using 10^7 uniformly distributed test points for all the examples. For the optimization problems in Equations 4.4 and 4.6, a convergence criterion of 10^{-3} was used on the objective function and the variables.

The results of the update for all three examples are listed in Table 4.1. The final SVM boundary for the three-dimensional case is plotted in Figure 4.11. The convergence plot for the three-dimensional example is depicted in Figure 4.12. The square root of the convergence measure Δ_k (Equation 4.13) is used for better readability of the plot by compressing the difference between the largest and the smallest values. The quantities $\Delta_k^{(1)}$, $\Delta_k^{(2)}$ and $\Delta_k^{(3)}$ (Equation 4.14) are also shown. The plot for $\Delta_k^{(1)}$ has a large peak in the beginning; however, being associated with

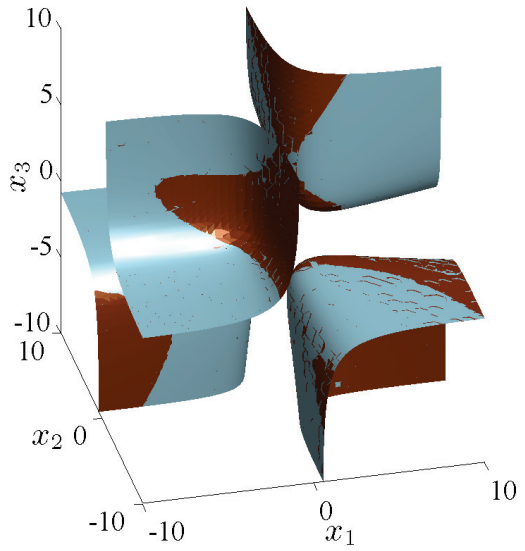

Figure 4.11: Three-dimensional problem with disjoint regions. Updated SVM boundary (light blue surface). The actual decision boundary is represented by the dark brown surface.

the linear terms, this may not correspond to the largest change in the SVM. At the end of the update all the quantities ($\Delta_k^{(1)}$, $\Delta_k^{(2)}$ and $\Delta_k^{(3)}$) converge to zero. The errors (E_k) for the three examples are plotted together in Figure 4.13. The initial and final values of the error measure ϵ_k are also provided in Table 4.1. The final error ϵ_{final}, which measures the discrepancy between the approximated and the actual boundary based a large number of test samples is lower than 0.1% even for the seven dimensional example. Similarly, the error E_{final}, based on the polynomial coefficients, are also low. It must be emphasized that the latter measure, although less intuitive than ϵ_{final}, allows one to quantify the error in higher dimensional spaces.

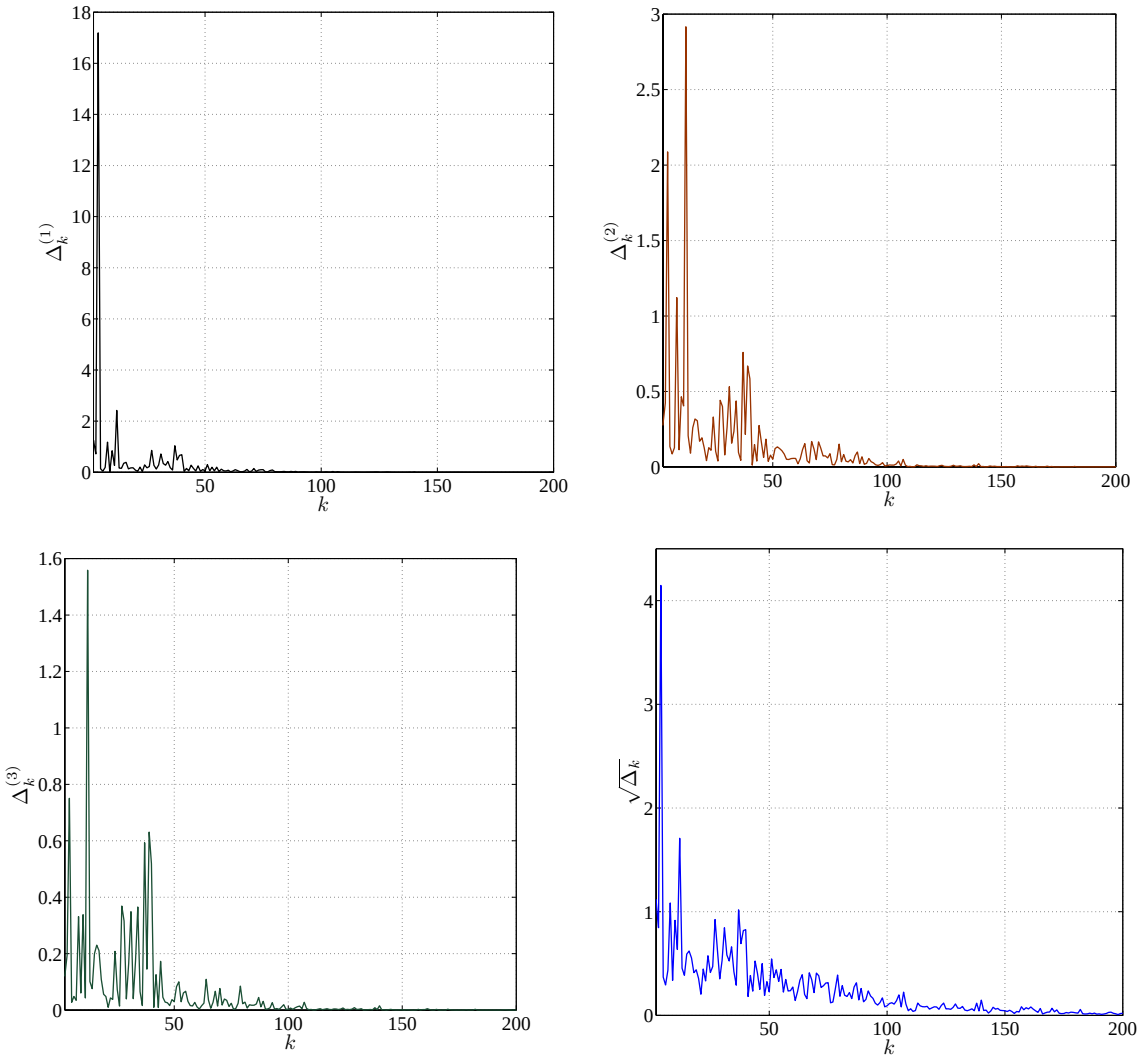

Figure 4.12: Three-dimensional problem. The bottom right figure shows the square root of the convergence measure. The other figures show the variation of polynomial coefficients corresponding to degrees 1,2,3.

d	$N_{initial}$	$E_{initial}$ (%)	$\epsilon_{initial}(\%)$	$Iterations$	N_{total}	E_{final} (%)	$\epsilon_{final}(\%)$
3	40	58.25%	9.4%	200	640	0.01%	$1.9 \times 10^{-3}\%$
5	160	55.93%	14.1%	400	1360	0.47%	$8.5 \times 10^{-3}\%$
7	640	46.44%	8.6%	800	3040	3.54%	$8.9 \times 10^{-2}\%$

Table 4.1: Number of samples and corresponding errors for the three examples.

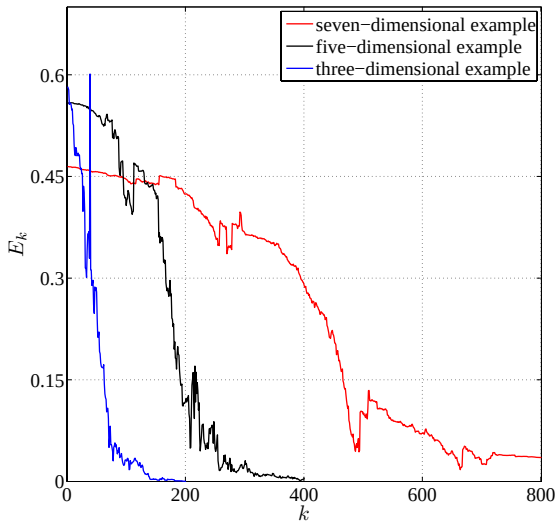

Figure 4.13: Comparison of the errors associated with the three, five and seven dimensional examples.

4.5.2 Example 4.2: Comparison of update schemes with and without secondary samples

In order to depict the importance of evaluating secondary samples, a two dimensional analytical test example is presented. The equation of the actual decision boundary is:

$$g(\mathbf{x}) = x_2 - 2\sin(x_1) - 5 \tag{4.22}$$

The initial SVM boundary is constructed using 20 CVT samples. The update is run up to 50 iterations and the final SVM boundary is constructed with a polynomial kernel of degree 4. In order to demonstrate the effect of secondary sample evaluations, the results are compared to the SVM boundary obtained after 50 iterations using primary samples only. The final SVM boundaries with and without secondary samples are plotted in Figure 4.14. The comparison of the evolution of the error measure ϵ_k is shown in Figure 4.15. The final decision boundary using both primary and secondary samples is close to the actual boundary whereas the scheme without secondary sample evaluation displays the locking phenomenon in some localized regions.

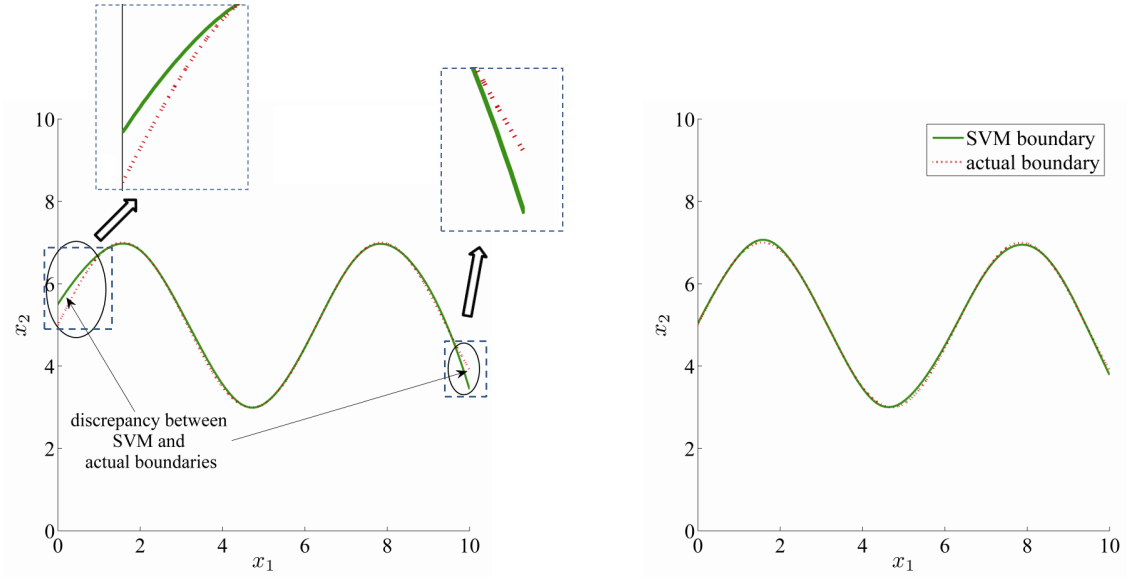

Figure 4.14: Comparison of update with and without secondary samples. Update using only primary sample results in locking (left). The regions where the SVM boundary differs from the actual boundary are circled. The boundary updated using both primary and secondary samples is very close to the actual boundary.

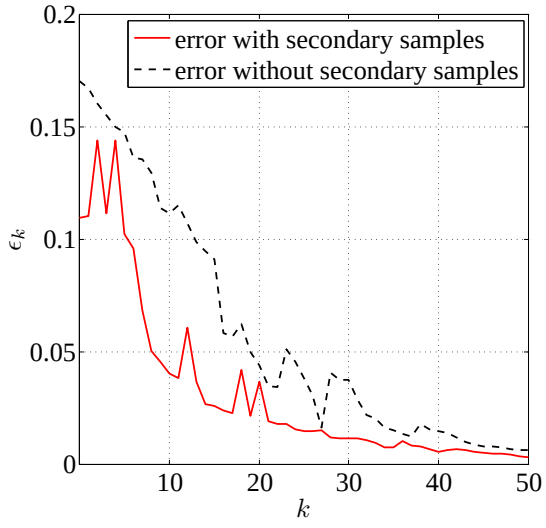

Figure 4.15: Comparison of the evolution of error measure ϵ_k with (solid red) and without (solid red) secondary sample evaluation.

4.5.3 Example 4.3: Explicit failure boundary approximation for nonlinear buckling of arch structure

The SVM-based EDSD method is applied to an arch structure subjected to a point load at the center (Figure 4.16). The objective of this example is to demonstrate the application of EDSD to problems with discontinuous responses. An arch is a typical example of a geometrically nonlinear structure exhibiting a snap-through behavior once the limit load is reached. A buckled structure is considered as failed in this example whereas configurations that do not lead to buckling are considered safe.

The arch considered in this example has a radius of curvature $R = 8$ m and subtends an angle $\theta = 14^\circ$ at the center of curvature. The thickness t, the width w, and the load F are random variables. The arch structure, simply supported at the ends, is modeled in ANSYS using SHELL63 elements. Due to the symmetries of the problem, only one fourth of the arch needed to be modeled. The range of values allowed for the design parameters are listed in Table 4.2.

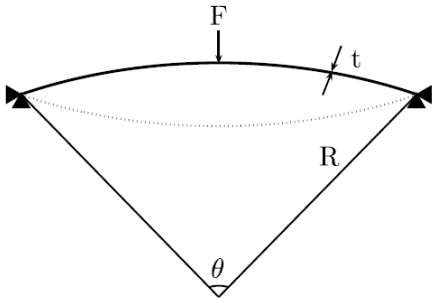

Figure 4.16: Arch geometry and loading.

	Thickness (t)	*Width* (w)	*Force* (F)
Min Value	3 mm	150 mm	2000 N
Max Value	10 mm	500 mm	8000 N

Table 4.2: Range of random variables for arch buckling

To construct the SVM decision function, first an initial LCVT (Romero, Vincente J. et al. (2006)) DOE consisting of 10 points is generated with thickness, width and load as the three variables. The variables are normalized by dividing the values by their respective maximum values. The studied response is the displacement of the central node which is solved for at each training sample (design configuration given by the LCVT DOE) using ANSYS. The response shows a clear discontinuity. The discontinuous variation of the displacement with respect to the thickness and width is depicted in Figure 4.17 for a fixed value of the applied load.

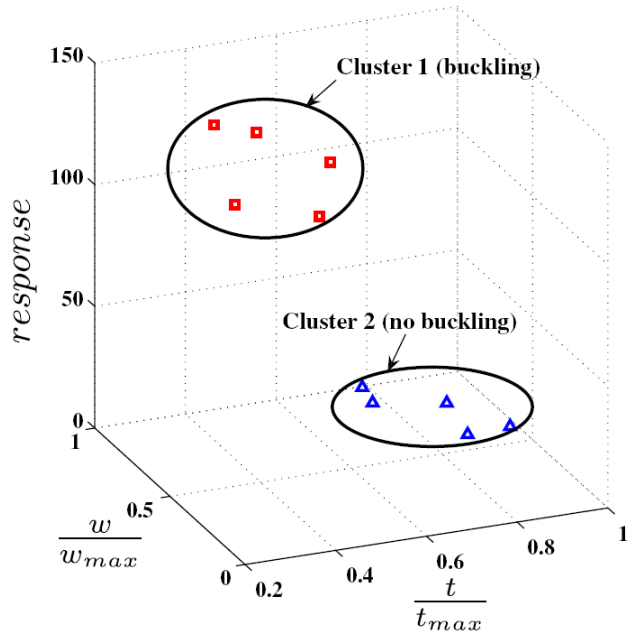

Figure 4.17: Discontinuous response of arch. The response (displacement) is obtained for a constant load $F = 6400$ N.

The discontinuity in displacement is used to separate the responses into two clusters using K-means clustering (Hartigan, J.A. and Wong, M.A. (1979)). One of the clusters corresponds to buckling (failure) while the other corresponds to design configurations which do not exhibit buckling. These two classes of samples in the design space are labeled as "+1" and "-1". This information is used to construct

the initial SVM boundary, which is then adaptively updated. A previous version of the adaptive sampling algorithm has been used for this example (Basudhar, A. and Missoum, S. (2008)). The Gaussian kernel is used, with a width parameter $\sigma = 2.2$. At every iteration the displacement of the new point is solved for. The new sample is added to the training set, and K-means clustering is then used again to reassign class labels to all the training samples based on their respective displacement values. After reassigning the class labels, SVM is reconstructed. The information is used for the selection of a new training sample in the next iteration, until the convergence criterion is met. The measure based on convergence points is used for this example.

The number of convergence points N_{conv} for the calculating Δ_k is $312,500$, and the values of ϵ_1 and ϵ_2 are 10^{-3} and 5×10^{-4} respectively. The number of training samples required to construct the final updated SVM decision function is 48. The initial and final SVM decision functions are shown in Figure 4.18. For comparison, an SVM decision function is also constructed using 48 LCVT training samples. Figure 4.19 shows that the decision function generated using 48 LCVT samples (dark brown surface) deviates from the updated SVM decision function (light grey surface). On the contrary, the updated decision function is very similar to the decision function (deep blue surface) constructed with a larger LCVT training set of 150 samples (Figure 4.19).

Convergence of the update algorithm is shown in Figure 4.20. The fraction of convergence points changing sign between successive iterations is plotted against the iteration number. Both the actual Δ_k values, and the fitted exponential curve are shown.

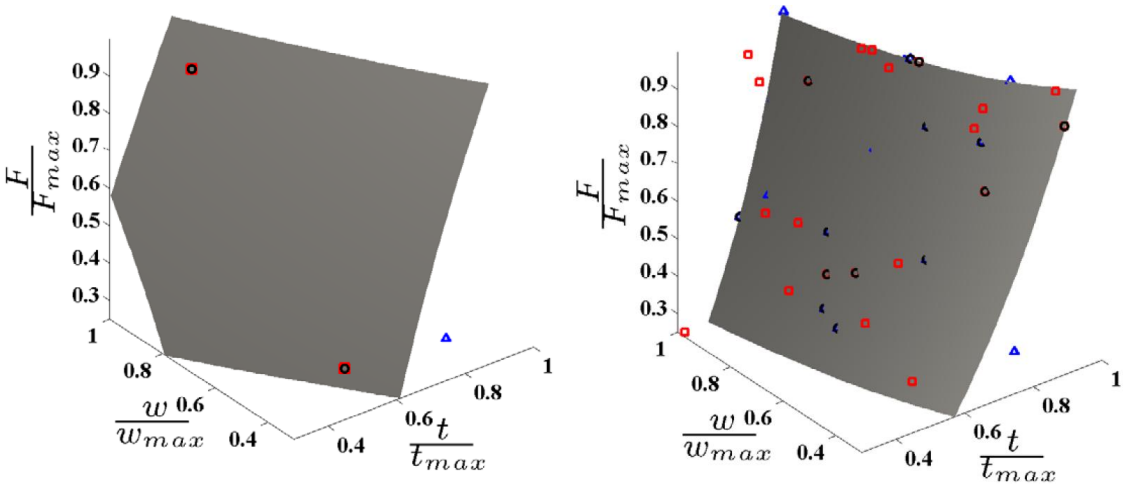

Figure 4.18: Arch problem. The left and right figures show the initial and final updated SVM decision functions constructed with 10 and 48 samples respectively.

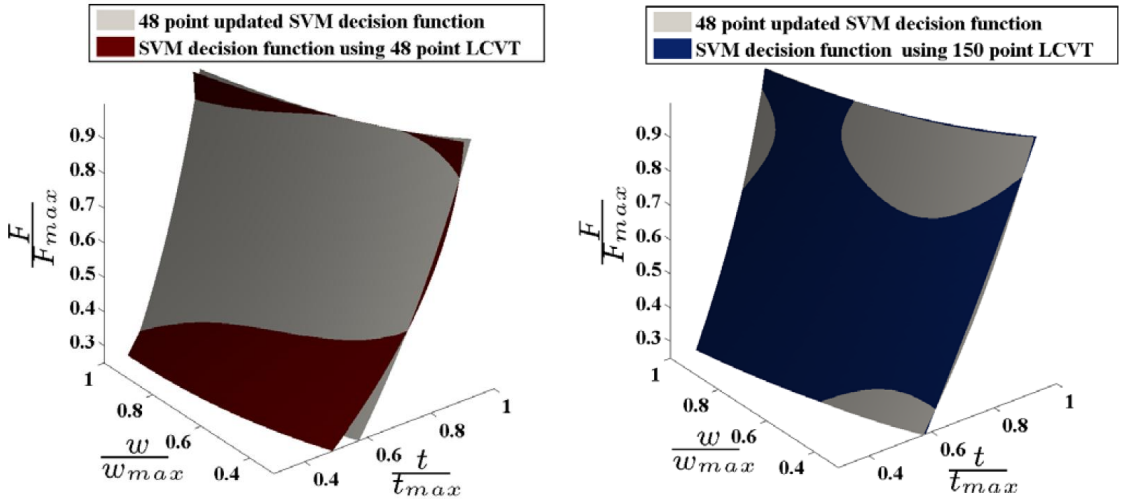

Figure 4.19: Arch problem. Comparison of SVM decision functions constructed using update algorithm and otherwise. The dark brown and light grey surfaces in the left figure are the decision functions using 48 LCVT samples and the update algorithm respectively. The deep blue surface in the right figure is the decision function constructed with 150 LCVT samples.

4.5.4 Example 4.4: Tolerance optimization for multibody system with multiple failure modes

In this section, an example of RBDO using SVM boundaries is presented. This example was formulated in the Masters thesis of Henry Arenbeck (Arenbeck, H.

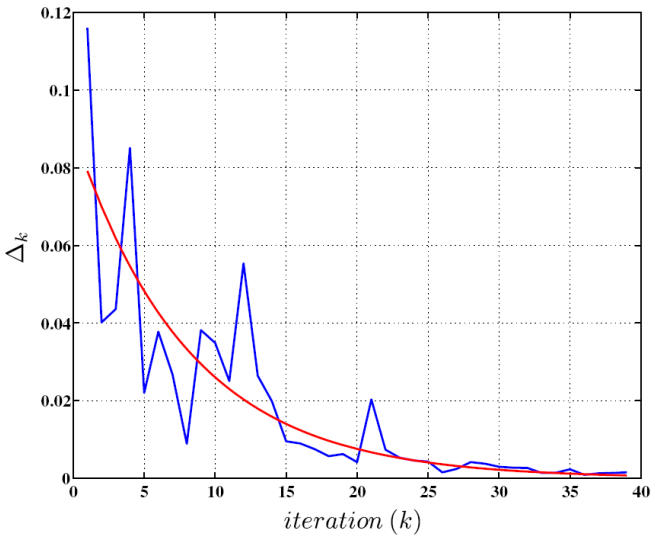

Figure 4.20: Arch problem. Convergence of the update algorithm.

(2007)). A portion of the work, concerning the construction of an adaptively refined SVM failure boundary, was performed as part of this dissertation. Therefore, only a brief summary of the problem is provided. Further details can be found in Arenbeck, H. et al. (2010).

The system considered is a web cutter mechanism (Figure 4.21) with multiple failure modes.

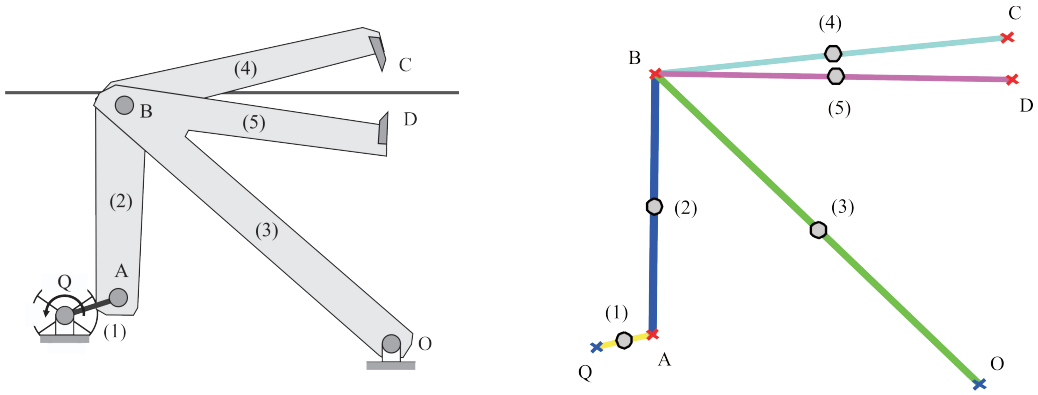

Figure 4.21: Web cutter mechanism.

Design variables are the lengths of members AQ (1), AB (2), and OB (3) as well as their respective tolerances. Lengths of the members are considered as random variables with truncated normal distributions. Width of the distributions are given by the respective tolerances. The tolerances are equal to two standard deviations of the corresponding link length distributions. The objective is to reduce the cost while maintaining certain target reliability. The cost is assumed to be a function of the tolerances only (inversely related). The optimization problem is:

$$\begin{aligned} \min_{\bar{\mathbf{x}},\mathbf{t}} \quad & C(\mathbf{t}) \\ s.t. \quad & P_f \leq P_T \\ & \bar{\mathbf{x}} \in [\bar{\mathbf{x}}_{min}, \bar{\mathbf{x}}_{max}] \\ & \mathbf{t} \in [\mathbf{t}_{min}, \mathbf{t}_{max}] \end{aligned} \tag{4.23}$$

where the target failure probability P_T is 5×10^{-4}. Failure is defined based on several criteria, such as stress, gap between the cutting blade edges, maximum web displacement, required working space etc. In total, there are 12 failure modes shown in Figure 4.22. The net failure domain is union of the individual ones. Ranges of the three link lengths are $\bar{x}_1 \in [0.091, 0.11]$, $\bar{x}_2 \in [0.682, 0.729]$, $\bar{x}_3 \in [0.986, 1.013]$. Ranges of the respective tolerances are $t_1 \in [10^{-5}, 0.016]$, $t_2 \in [10^{-5}, 0.037]$ and $t_3 \in [10^{-5}, 0.019]$.

An initial set of 40 LCVT samples is distributed uniformly over the three dimensional space consisting of x_1, x_2 and x_3. All the variables are scaled by their maximum values for constructing the SVM. The initial SVM boundary constructed with the LCVT samples is updated using adaptive sampling (Basudhar, A. and Missoum, S. (2008)). The final SVM boundary is constructed with 178 samples. Width parameter of the Gaussian kernel used for this example is automatically selected at each iteration, as mentioned in Section 3.1.3. A width parameter of 0.1 is used to construct the final SVM limit state function (Figure 4.23). In the figure, the large dots represent the samples used for training of the SVM-function. The small dots represent the falsely classified samples of a large reference dataset consisting

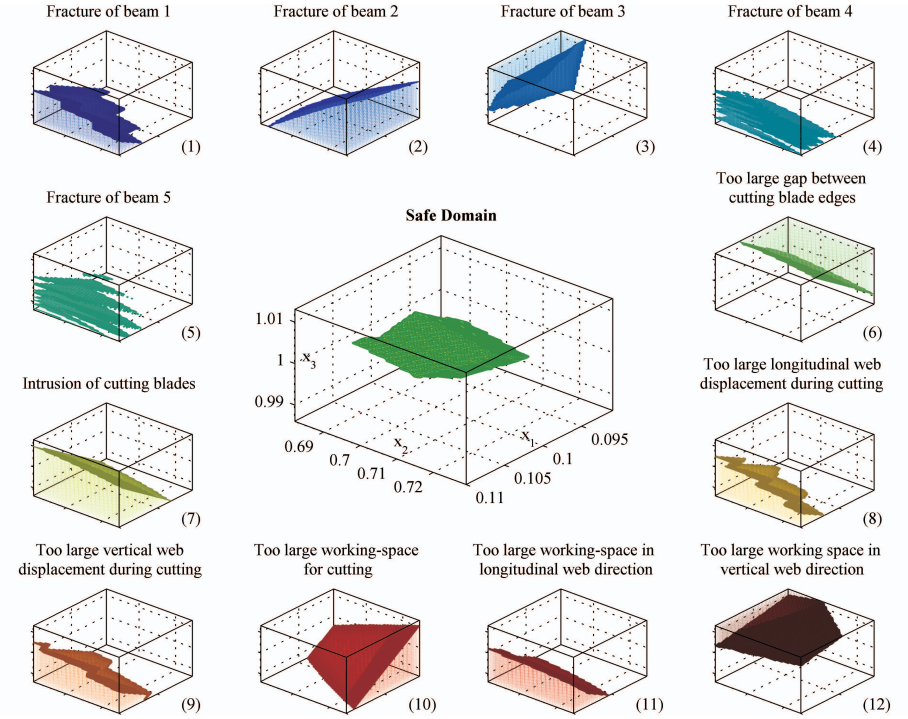

Figure 4.22: Failure modes of the web cutter and the net safe domain.

of 10^5 samples. These samples provide a measure of the error of the approximated limit state function. It is noticeable that the adaptive sampling scheme successfully yielded an increased sampling density in the vicinity of the limit state while avoiding clustering effects. The error of SVM classification is calculated to be 3.4%. It should be noted that an older version of the update algorithm is applied that does not have secondary samples. The accuracy is expected to be higher when the current update scheme consisting of both primary and secondary samples is used.

Although the SVM boundary is constructed in a three dimensional space consisting of the member lengths, the RBDO problem is defined with respect to six variables. This is because the remaining variables are the tolerances related to the same physical entities, i.e. the member lengths. For any configuration in the six dimensional space, the probability density functions of x_1, x_2 and x_3 are uniquely defined, with the tolerances defining the standard deviations. Therefore, the proba-

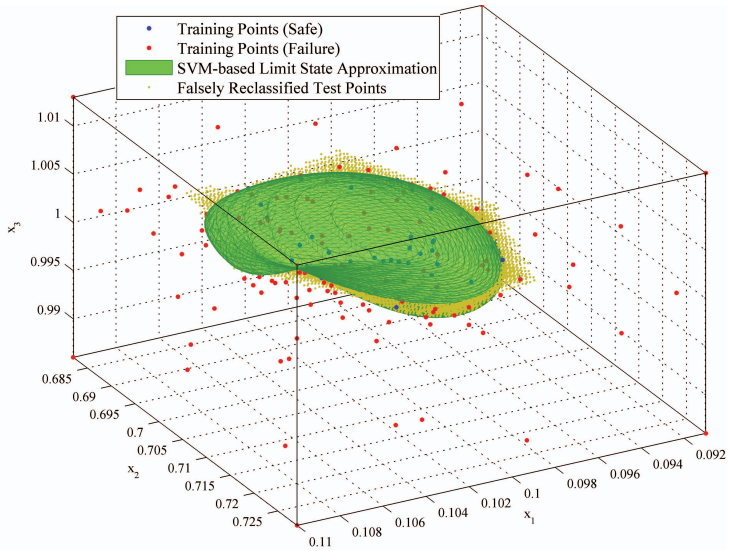

Figure 4.23: Tolerance Optimization Example. Failure boundary approximation using SVM.

bility of failure at any configuration in the six dimensional space can be calculated using the three dimensional SVM boundary. The probability of failure is calculated using MCS (Equation 2.51). To perform the RBDO, the probability of failure is calculated at 1000 uniform samples. These probabilities of failure are converted to the respective reliability indices β:

$$\beta = -\Phi^{-1}(P_f) \tag{4.24}$$

The reliability indices are then fitted to a surface using support vector regression (SVR) to get an analytical expression for β. The RBDO problem is then solved as:

$$\begin{aligned} \min_{\bar{\mathbf{x}},\mathbf{t}} \quad & C(\mathbf{t}) \\ s.t. \quad & \beta_T - \hat{\beta} \leq 0 \\ & \bar{\mathbf{x}} \in [\bar{\mathbf{x}}_{min}, \bar{\mathbf{x}}_{max}] \\ & \mathbf{t} \in [\mathbf{t}_{min}, \mathbf{t}_{max}] \end{aligned} \tag{4.25}$$

where β_T is the target reliability index (equal to 3.29) and $\hat{\beta}$ is the approximation of β using SVR. Details of the optimum results are omitted to avoid deviation from

the main message of this chapter, i.e. the construction of SVM. Details can be referred in Arenbeck, H. et al. (2010).

4.6 Discussion

This section presents a discussion on some of the features of EDSD and the SVM update. The effects of the sampling scheme on the update, as well as some possible improvements are discussed.

- *Application of EDSD to discontinuous responses and multiple failure modes:* The major difference in this dissertation compared to existing reliability assessment and optimization techniques is the use of a classification-based approach. The nonlinear arch buckling example with discontinuous responses demonstrates one of the major advantages of such an approach. The presence of discontinuities makes the application of response surface methods or other conventional methods difficult or inaccurate. However, the SVM-based method is undeterred by discontinuities, and it provides an explicit equation of the failure boundary that can be used for probabilistic optimization of the arch (Basudhar, A. et al. (2008)). The calculation of the probability of failure using MCS is made efficient, once the explicit SVM boundary is constructed. The value of indicator function in MCS (Equation 2.51) is obtained from the sign of SVM value $s(\mathbf{x})$ (Equation 3.16), and does not require actual expensive function evaluations. In addition to discontinuities, the EDSD method can also handle binary states and multiple failure modes. The limit state function for the tolerance optimization example is approximated with a single SVM. However, the problem has several failure modes that will hamper the conventional approximation methods.

- *SVM locking and convergence of the update:* The "SVM locking" phenomenon (Figure 4.7) results in a low rate of convergence of the SVM to the actual

boundary in localized regions of the space. It is noteworthy that although the term "SVM locking" may suggest that it entirely "stops" the SVM update, in reality, the update is believed to be convergent even without the locking removal step (secondary sample evaluation). However this would require a large number of samples, which would defeat one of the main purposes of the adaptive sampling scheme. The use of secondary samples enables one to reduce the number of necessary samples by efficiently reducing the local locking phenomena whose removal would otherwise require many function calls.

Another noteworthy feature of the locking phenomenon stems from the fact that it is a local phenomenon. For this reason, there might not always be a clear difference between the global convergence rates of the proposed scheme and the adaptive sampling scheme without secondary samples. However, these local errors in the SVM boundary construction might have a significant influence on the optimum solution or the probability of failure calculated using the SVM boundary. Therefore, it is important to remove the locking using secondary samples. Also, it is expected that the locking phenomenon may have a greater influence on the global convergence rate in higher dimensions. This needs a detailed study in the future.

- *Selection of secondary samples:* In this chapter, secondary samples are selected using the method 1 explained in Section 4.4.1. That is, the center of hypersphere, within which the sample lies, is selected as one of the support vectors. As explained in Section 4.4.1, this may ignore some regions of the space. Using the method 2 of selecting secondary samples within the global update scheme is expected to improve its efficiency. This method will be used in the following chapters within other adaptive sampling frameworks (Chapters 5 and 7).

- *Optimization of the sampling sequence:* Although the proposed sampling scheme has the ability to provide accurate decision boundary approximations, there is scope for further improvement of the approach. The frequency of evaluating secondary samples is not optimized in this work; there is no scheme to detect whether or not a secondary sample is required. Therefore, secondary samples are selected systematically (one for every two primary samples) in regions that are most likely to require a secondary sample. Such regions are identified as the ones where data from one class is sparse in the vicinity of the boundary. A scheme to detect whether a secondary sample needs to be evaluated may be useful. Such a scheme may be devised based on a critical distance from existing samples. However, ways to define the critical distance need to be studied.

- *Choice of the kernel:* As mentioned in Section 4.4.2, the polynomial kernel allows for a rigorous convergence measure based on the polynomial coefficients. However, the polynomial kernel is not necessarily superior to other kernels, such as the Gaussian kernel, in terms of the number of evaluations. If a Gaussian kernel is used, the convergence criterion based on "convergence points" may be used. However, the approach is not scalable, as filling the space with samples is possible only up to a few dimensions. A polynomial coefficient-based convergence criterion may be used with the Gaussian kernel by expanding it on a polynomial basis. However, the number of terms in the expansion of the Gaussian kernel may be crucial and needs to be studied. In terms of the methodology to select new samples, the update scheme is same irrespective of the kernel.

4.7 Concluding remarks

4.7.1 Summary

A novel method for explicit construction of decision boundaries (limit state functions and constraint boundaries) referred to as explicit design space decomposition (EDSD) is presented in this chapter. A machine learning technique referred to as support vector machines (SVMs) is used to construct the boundaries. The technique is particularly useful for problems exhibiting discontinuous and binary responses, disjoint failure domains, and multiple failure modes.

An adaptive sampling scheme for updating SVM decision boundaries is also developed. The ability of the method to accurately reconstruct analytical functions has been demonstrated for problems up to seven dimensions. The results from the application of the approach to highly nonlinear examples of up to seven variables are promising. The examples consist of decision boundaries that form multiple disjoint regions in the space. An arch buckling example with discontinuous responses is also presented. Application to multiple failure modes is also presented through the tolerance optimization example. The EDSD method presented in this chapter is the basic building block for the more advanced methods presented in latter chapters. While the update scheme in this chapter is global, a local update to further reduce the number of samples, and to make the method more scalable is presented in the next chapter.

4.7.2 Future work

The adaptive sampling scheme presented in this chapter could benefit from some relatively minor incremental changes as mentioned in the discussion section. Improvements to further reduce the number of samples are being considered. Specifically, a scheme to detect whether a secondary sample needs to be evaluated may be useful. The use of multifidelity models and competing approximations may also

be useful for the update. Also, the polynomial kernel has been used in this work as it provides a rigorous convergence criterion based on the polynomial coefficients. In the future, the method will be generalized by enabling the use of the polynomial coefficient based convergence criterion for the Gaussian kernel, as explained in the discussion section.

CHAPTER 5

RELIABILITY-BASED DESIGN OPTIMIZATION USING LOCALLY REFINED SVMs

In Chapter 4, an introduction to the SVM-based EDSD method was provided, along with a global update scheme to refine SVM boundaries. The use of updated SVM boundaries for reliability assessment and RBDO was also demonstrated. However, it is natural that the number of samples required for an accurate SVM will increase with the dimensionality. Therefore, to make the approach more scalable, a method to locally update SVM boundaries for RBDO is presented in this chapter (Basudhar, A. and Missoum, S. (2009a)). First, an adaptive sampling technique is proposed in order to construct an explicit limit state function approximation and obtain an accurate probability of failure (Section 5.1). This reliability assessment technique is then used as part of an RBDO algorithm in Section 5.2. The RBDO algorithm is based on the definition of two explicit boundaries - one for the approximation of the limit state function (LSF) and one for the approximation of the zero-level contour of probabilistic constraint. These boundaries are refined within a local "update region" whose size and position are modified iteratively. Finally, analytical examples to validate the methods are presented in Section 5.3.

5.1 Adaptive sampling for probability of failure calculation

It has already been demonstrated in Chapter 4 that EDSD can be used to generate explicit LSFs and calculate probabilities of failure. The calculation of failure probabilities with explicit LSF approximations is straightforward using Monte-Carlo simulations (MCS) (Section 4.2). However, the LSF approximations using the previously presented method could lead to substantial errors in the probability estimates. In this section, an adaptive sampling scheme dedicated to calculating accurate prob-

abilities of failure is presented. The proposed algorithm generates an accurate approximation of the LSF within a hyperbox whose bounds are defined based on the probability density functions (pdfs) of the variables. For example, for a variable with truncated distribution, the bounds of the hyperbox for the corresponding dimension are given by the lower and upper bounds of the pdf. Within this region, the probability of failure is assessed as summarized in Algorithm 5.1. Three types of samples are used as part of the adaptive scheme. The sample selection steps are explained in Section 5.1.1, following Algorithm 5.1.

Algorithm 5.1: probability of failure calculation

1: Define region for updating the LSF (based on the pdfs of the random variables).
2: Set $k = 0$
3: Construct the initial SVM approximation of the LSF ($s_d^{(0)} = 0$) using samples from a CVT DOE in the selected space.
4: Calculate the initial probability of failure estimate $P_f^{(0)}$ with the SVM approximation, using MCS (Equation 4.1).
5: **repeat**
6: $k = k + 1$
7: Adaptively select new samples using the three sample selection steps in Section 5.1.1, in order to refine the SVM approximation. Reconstruct SVM after each step.
8: Calculate the new probability of failure $P_f^{(k)}$.
9: **until** $\frac{\left|P_f^{(k)} - P_f^{(k-1)}\right|}{P_f^{(k-1)}} \leq \delta_1$ for $d + 1$ consecutive iterations

---

5.1.1 Sample selection steps for the update of probability of failure

The sample selection steps for updating the SVM LSFs are described in this Section. For the sake of clarity, the major points of the algorithm are presented while details are described in Appendix A.

Step 1: *Primary sample* ($\mathbf{x}_{mm}$) *on the SVM boundary with maximum minimum distance from existing training samples*

In step 1, a primary sample is selected within the update domain determined based on the PDFs of the random variables. Details of the method were presented in Chapter 4 (Equation 4.4). Selection of primary samples on the SVM boundary has been shown to be quite effective in updating the boundary in Chapter 4, as they lie in sparse regions with high probability of misclassification. Also, being located within the margin, they are bound to modify the SVM.

Although a primary sample ($\mathbf{x}_{mm}$) is bound to modify the SVM, the extent of this change may be negligible. In order to avoid unnecessary function evaluations, changes in SVM boundary are evaluated based on the probabilities calculated using MCS. These variations are calculated under the two hypothesis that the sample belongs to one class or the other. If one of these changes leads to a large variation of the probability of failure, then the sample is actually evaluated. Details are given in the Appendix A.

Step 2: *Secondary sample* $\mathbf{x}_s$ *to remove SVM locking*

Similar to the global update algorithm, a secondary sample to remove SVM locking (see Chapter 4) is evaluated. The method 2 for selecting secondary samples (Equations 4.7-4.9) is used. As was explained in Chapter 4, unlike method 1, the secondary samples selected using this improved method are not limited to regions surrounding existing support vectors.

Step 3: *Sample at closest point on the SVM boundary from the mean*
In step 3, a sample is selected at the closest point to the mean that lies on the SVM boundary (Figure 5.1). In a standard normal space, this sample is the "most probable point (MPP)". Such a point close to the mean has high probability density and a small change in the SVM in that region can cause a significant variation in the failure probability. Therefore, this step intends to have high accuracy of the SVM boundary in regions close to the mean. It must be understood that this sample becomes the actual MPP at convergence of the boundary only if the space is standard normal. The optimization problem is:

$$\begin{aligned} \min_{\mathbf{x}} \quad & ||\mathbf{x} - \mathbf{x}_0|| \\ s.t. \quad & s(\mathbf{x}) = 0, \end{aligned} \tag{5.1}$$

where $\mathbf{x}_0$ is the mean at which probability of failure is calculated. In order to reduce the number of function evaluations, the sample is evaluated only if the maximum possible change in the failure probability due to its evaluation is greater than 1% of the current estimate. The maximum change is evaluated by considering the two cases with class label $+1$ or -1 for the selected sample.

5.2 RBDO using locally refined SVMs

An adaptive sampling methodology for performing RBDO using locally refined SVM boundaries is presented in this section. Global update of SVMs for RBDO can require a large number of samples for accurate failure probabilities, especially for problems with high dimensionality. Therefore, in the method presented in this section, SVM boundaries are updated locally within an "update region". The RBDO problem is defined as:

$$\begin{aligned} \min_{\bar{\mathbf{x}}} \quad & f(\bar{\mathbf{x}}) \\ s.t. \quad & P(\mathbf{x} \in \Omega_f) \leq P_T, \end{aligned} \tag{5.2}$$

where $\bar{\mathbf{x}}$ is the mean value of $\mathbf{x}$, $f(\bar{\mathbf{x}})$ is the objective function, $P(\mathbf{x} \in \Omega_f)$ is the probability of failure and P_T is the target failure probability.

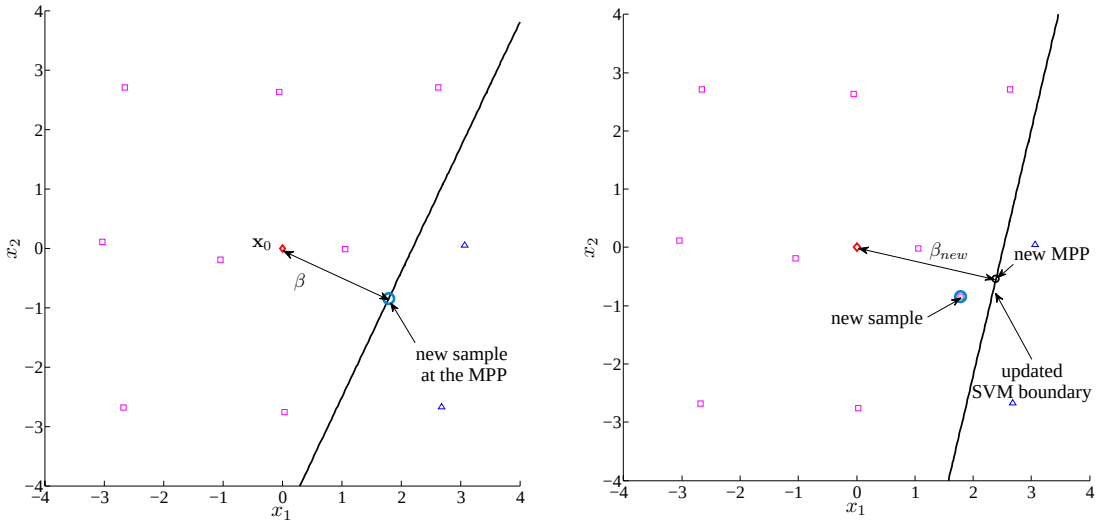

Figure 5.1: Change in the SVM boundary due to the evaluation of the closest point on the boundary. The closest point on the boundary is the MPP in a standard normal space.

A salient feature of the proposed RBDO methodology is the use of two SVM decision boundaries. One boundary ($s_d = 0$) approximates the limit-state $g(\mathbf{x}) = 0$ that defines the failure region Ω_f. The other SVM ($s_p = 0$) represents the zero-level of the probabilistic constraint ($P(\mathbf{x} \in \Omega_f) - P_T = 0$ or $P_f - P_T = 0$). The construction of an accurate s_d is important for locating the deterministic optimum $\mathbf{x}_d$ and correctly calculating the failure probabilities. The construction of s_p is important to locate the probabilistic optimum $\mathbf{x}_p$. Both the SVMs are important throughout the update to account for both deterministic and probabilistic optimum positions while guiding the sample selection. It should be noted that while defining the boundary $s_d = 0$ requires (expensive) function evaluations at the training samples, the classification information for constructing $s_p = 0$ is easily available. Once the approximated boundary $s_d = 0$ is constructed, it can be used to calculate the failure probability P_f at any sample. It is therefore straightforward to classify the samples based on the sign of $P_f - P_T$. The initial predictions of both $s_p = 0$ and $s_d = 0$ are built using CVT DOEs. The SVM boundaries and the corresponding

optima ($\mathbf{x}_d$ and $\mathbf{x}_p$) are updated with an adaptive scheme. The new samples for the update are selected within a local "update region" which is modified iteratively based on the optima.

The update is performed in two phases. The motivation is to reduce the number of samples while providing a final accurate result. The initial SVM approximation can be quite different from the actual function, thereby predicting a solution that is far from the actual optimum. Therefore, the first step is to find an approximate location of the optimum without wasting too many samples (phase 1). Phase 1 update is performed with relatively crude estimates of failure probabilities. However, a high accuracy is required for the final solution. Therefore, phase 1 is followed by a more rigorous update in phase 2. In phase 2, the failure probability at each iterate is calculated accurately using the update scheme developed specifically for the calculation of probabilities of failure (Algorithm 5.1). However, because the phase 2 update starts from a relatively good starting point (solution of phase 1), it is expected to require fewer steps to converge to the optimum. The convergence of the RBDO is based on the positions of $\mathbf{x}_d$ and $\mathbf{x}_p$. The actual interest while solving the RBDO lies in the probabilistic optimum $\mathbf{x}_p$, which is the basis for convergence of phase 2 update. However, because phase 1 update is based on relatively crude failure probability estimates, its convergence is based on the position of $\mathbf{x}_d$.

Summary of the RBDO methodology is given in Algorithm 5.2. The definition of two SVM boundaries and modification of the update region are shown conceptually in Figures 5.2 and 5.3.

Algorithm 5.2: RBDO Algorithm

1: Define the range of variables (extended beyond the optimization side constraints based on the probability density functions of the random variables).

2: Construct the initial SVM approximation $s_d^{(0)}$ of the LSF $g(\mathbf{x}) = 0$ using samples from a CVT DOE in the selected space.

3: Construct the initial SVM approximation $s_p^{(0)}$ which corresponds to the points of the design space for which $P_f^{(0)} = P_T$:

- Generate another (comparatively dense) CVT DOE and calculate the initial failure probability estimate $P_f^{(0)}$, using MCS, at each sample with the SVM approximation. It is noteworthy that this step does not require actual function evaluations.
- Classify these samples as ± 1, based on whether $P_f^{(0)} > P_T$ or $P_f^{(0)} \leq P_T$.

4: Find the initial deterministic and probabilistic optima $\mathbf{x_d^{(0)}}$ and $\mathbf{x_p^{(0)}}$ using the SVM boundaries $s_d^{(0)} = 0$ and $s_p^{(0)} = 0$ respectively.

5: Set iteration $k = 0$.

6: **repeat**

7: Phase 1 update: Define local update region for selection of new samples based on the optima. Select new samples for the refinement of SVM boundary $s_d^{(k)} = 0$ (Section 5.2.1).

8: Refine SVM boundary $s_p^{(k)} = 0$ (Appendix A).

9: Set $k = k + 1$.

10: Update $\mathbf{x_d^{(k)}}$ and $\mathbf{x_p^{(k)}}$ using the new SVM boundaries $s_d^{(k)} = 0$ and $s_p^{(k)} = 0$ respectively.

11: **until** $\left|\left|\mathbf{x}_d^{(k)} - \mathbf{x}_d^{(k-1)}\right|\right| \leq \epsilon_1$

12: **repeat**

13: Phase 2 update: probability of failure calculation at $\mathbf{x_p^{(k)}}$ using Algorithm 5.1. This leads to further refinement of $s_d^{(k)} = 0$.

14: Reconstruct SVM boundary $s_p^{(k)} = 0$.

15: Set $k = k + 1$.

16: Find the updated probabilistic optimum $\mathbf{x_p^{(k)}}$ using the boundary $s_p^{(k)} = 0$.

17: **until** $\left|\left|\mathbf{x}_p^{(k)} - \mathbf{x}_p^{(k-1)}\right|\right| \leq \epsilon_1$

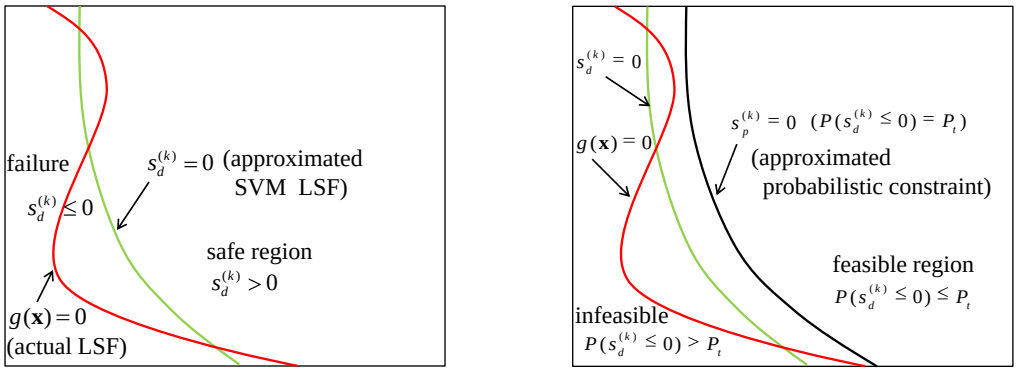

Figure 5.2: SVM approximations for the LSF and the probabilistic constraint in RBDO. The left figure shows the approximation $s_d = 0$ (green curve) for the LSF $g(\mathbf{x}) = 0$ (red curve). The right figure shows the approximation $s_p = 0$ (black curve) for the probabilistic constraint. The failure probabilities for the construction of $s_p = 0$ are calculated based on the SVM LSF $s_d = 0$.

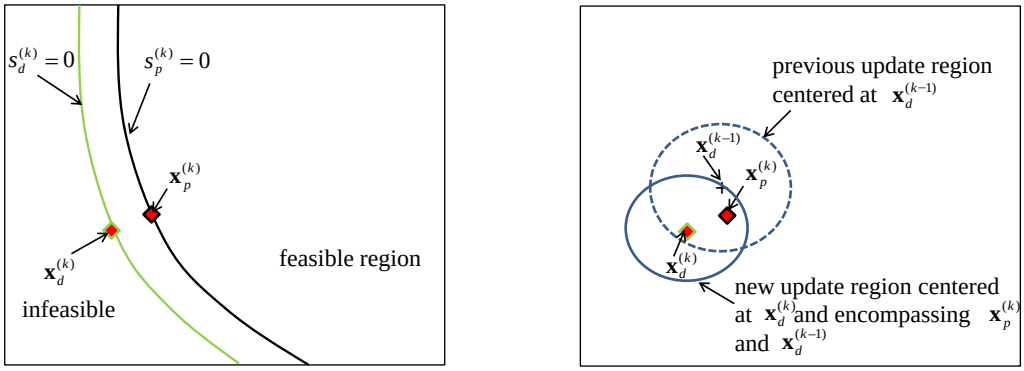

Figure 5.3: Modification of the update region based on the optima. The left figure shows the current deterministic and probabilistic optima $\mathbf{x}_d^{(k)}$ and $\mathbf{x}_p^{(k)}$. The right figure shows the update region. The dashed circle is the previous update region centered at the previous deterministic optimum $\mathbf{x}_d^{(k-1)}$ and circle with solid line is the new update region centered at the current optimum $\mathbf{x}_d^{(k)}$. The update region encompasses the current probabilistic optimum and the previous deterministic optimum.

5.2.1 Phase 1 update - local update region and sample selection steps

The SVM boundaries ($s_p = 0$ and $s_d = 0$) are refined only within a region of interest which, in the first phase, is not related to the probabilistic distributions of the variables. New samples are added within the update region to update the boundaries and the corresponding optima. For the first iteration ($k = 0$), the

entire space is the update region. Subsequently, the update region for phase 1 is a hypersphere of radius $R_u^{(k)}$ centered at $\mathbf{x_d^{(k)}}$ defined as:

$$R_u^{(k)} = \max\left(\Delta\mathbf{x}_d^{(k)}, \left|\left|\mathbf{x}_d^{(k)} - \mathbf{x}_{+1}\right|\right|, \left|\left|\mathbf{x}_d^{(k)} - \mathbf{x}_{-1}\right|\right|, \frac{1}{2}\left(\frac{V}{N}\right)^{\frac{1}{d}}, \left|\left|\mathbf{x}_d^{(k)} - \mathbf{x}_p^{(k)}\right|\right|\right), \tag{5.3}$$

where $\Delta\mathbf{x}_d^{(k)} = \left|\left|\mathbf{x}_d^{(k)} - \mathbf{x}_d^{(k-1)}\right|\right|$, $\mathbf{x}_{+1}$ and $\mathbf{x}_{-1}$ are the closest samples from $\mathbf{x}_d^{(k)}$ belonging to the +1 and -1 classes and V is the volume of the space. Therefore, the size of the update region is defined so as to include the optima $\mathbf{x}_d^{(k-1)}$ and $\mathbf{x}_p^{(k)}$, as well as at least one sample belonging to each class. The sample selection steps are listed below:

Step 1: Evaluate the deterministic optimum $\mathbf{x}_d^{(k)}$.

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ s.t. \quad & s_d^{(k)}(\mathbf{x}) \geq 0 \end{aligned} \tag{5.4}$$

Step 2: Evaluate the probabilistic optimum $\mathbf{x}_p^{(k)}$.

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ s.t. \quad & s_p^{(k)}(\mathbf{x}) \leq 0 \end{aligned} \tag{5.5}$$

Step 3: Evaluate the closest point to $\mathbf{x}_p^{(k)}$ lying on $s_d^{(k)} = 0$.

Step 4: Evaluate sample with maximum minimum distance from existing training samples, lying on the decision function $s_d = 0$ within the update region. This is similar to Equation 4.4, except for an additional constraint of the spherical update region.

Step 5: Select a secondary sample for SVM locking removal using the steps in Equations 4.7-4.9, similar to the step 2 of the probability of failure update (Section 5.1.1). However, there is an additional constraint of the local spherical update region.

5.2.2 Phase 2 update

In the phase 2 update, the probability of failure is calculated at $\mathbf{x}_p^{(k)}$ with the update methodology presented in Section 5.1. In this RBDO context, the variable ranges (i.e., the dimensions of the hyperbox) are twice of that used for a simple calculation of failure probability at a given point. This is done in order to obtain an accurate s_p boundary constructed from MCS-based probabilities of failure at the samples. The calculation of failure probability using the adaptive sampling technique leads to further refinement of the LSF $s_d^{(k)} = 0$ around the mean $\mathbf{x}_p^{(k)}$. As shown in Algorithm 5.2, the phase 2 update is repeated until the probabilistic optimum converges.

5.3 Examples

Three examples with analytical LSFs are shown in this section to demonstrate the efficacy of the proposed update methodology. The first two examples show the calculation of failure probability at a given point while the third example presents an RBDO problem with two critical failure modes. The analytical decision functions are written in the form $g(\mathbf{x}) = 0$. In order to construct the SVM failure boundary, the samples corresponding to $g(\mathbf{x}) > 0$ and $g(\mathbf{x}) \leq 0$ are considered as belonging to safe $(+1)$ and failed (-1) classes respectively.

For all examples, SVM boundaries are constructed using the polynomial kernel. Degree of the polynomial is automatically selected as explained in Chapter 4. For the probability of failure update, 10^6 samples are used for MCS. This allows for approximately 5% error in the calculation for failure probabilities of order 10^{-3} that are treated in this chapter. In order to verify the accuracy of the final probability of

error, the failure probabilities are also calculated using the known analytical LSFs. 10^7 MCS samples are used to compare the final values, as this allows for only 2% error due to MCS variance. The convergence criterion δ_1 as well as δ_2 (Appendix A) for the probability of failure update are varied within a reasonable range to study the effect of these parameters on the results.

The following notation will be used in the results section:

- N_i is the initial training set size.
- N_t is the total number of samples required at the end of the update.
- f_{opt} is the optimum objective function value.
- P_f^{SVM} is the probability of failure using the SVM LSF.
- P_f^{act} is the probability of failure using the actual function.
- $\Delta_{P_f}^{(k)}$ is the convergence measure for P_f^{SVM}, that is the maximum relative change in P_f^{SVM} between successive iterations over $d+1$ consecutive iterations.
- $E_{P_f}^{(k)}$ is relative error in P_f^{SVM} with respect to P_f^{act}.

5.3.1 Probability of failure. Example 5.1

The actual decision function in this example is a function of two random variables x_1 and x_2. Both the variables have standard normal distributions. The variable ranges for the update are chosen as four standard deviations ($[-4, 4]$). The equation of the LSF is:

$$g(x_1, x_2) = (5x_1 + 10)^3 + (5x_2 + 9.9)^3 - 18 = 0 \tag{5.6}$$

The probability of failure P_f^{act} with the actual LSF is calculated to be 5.7×10^{-3} using MCS. The 95% confidence interval (CI) for this MCS estimate is $[5.6, 5.8] \times 10^{-3}$.

The initial SVM is constructed using 10 CVT samples. Effect of the convergence parameter δ_1 on the update is studied. The parameter δ_2 (Appendix A) is varied between 0.1% and 5%. The results are listed in Tables 5.1-5.3. Figure 5.4 shows the initial SVM, the updated SVM with $\delta_1 = \delta_2 = 5 \times 10^{-3}$ and the actual LSF.

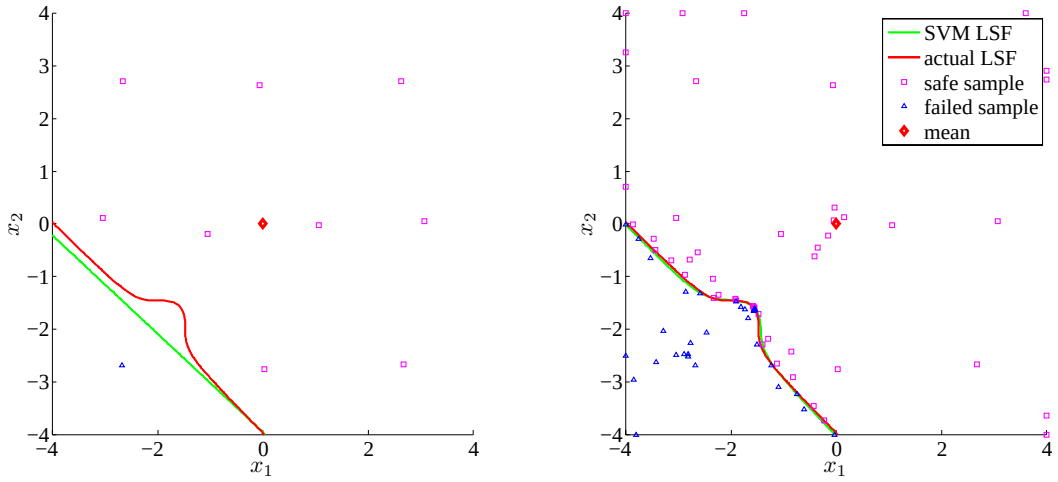

Figure 5.4: Example 5.1. Failure probability calculation at the origin for a two-dimensional non-convex LSF. The red curve represents the actual LSF. The green curves in left and right figures show the initial and final SVM approximations.

δ_2	N_t	P_f^{SVM}	E_{Pf}	P_f^{SVM} 95% CI
0.001	75	5.8×10^{-3}	1.6%	$[5.7, 5.8] \times 10^{-3}$
0.005	59	5.6×10^{-3}	2.2%	$[5.5, 5.6] \times 10^{-3}$
0.01	78	5.7×10^{-3}	0.7%	$[5.7, 5.8] \times 10^{-3}$
0.05	70	6.0×10^{-3}	4.7%	$[5.9, 6.0] \times 10^{-3}$

Table 5.1: Results for Example 5.1. $\delta_1 : 0.001$

5.3.2 Probability of failure - example 5.2

This example presents the probability of failure calculation with an LSF consisting of three random variables x_1, x_2 and x_3. All the variables have standard normal distributions. The variable ranges are selected as $[-4, 4]$. The equation of the LSF

δ_2	N_t	P_f^{SVM}	E_{Pf}	P_f^{SVM} 95% CI
0.001	54	5.5×10^{-3}	2.7%	$[5.5, 5.6] \times 10^{-3}$
0.005	79	5.8×10^{-3}	2.1%	$[5.8, 5.9] \times 10^{-3}$
0.01	72	6.0×10^{-3}	4.9%	$[5.9, 6.0] \times 10^{-3}$
0.05	70	6.0×10^{-3}	5.4%	$[6.0, 6.1] \times 10^{-3}$

Table 5.2: Results for Example 5.1. $\delta_1 : 0.005$

δ_2	N_t	P_f^{SVM}	E_{Pf}	P_f^{SVM} 95% CI
0.001	62	6.1×10^{-3}	7.8%	$[6.1, 6.2] \times 10^{-3}$
0.005	60	6.1×10^{-3}	7.2%	$[6.1, 6.2] \times 10^{-3}$
0.01	77	6.0×10^{-3}	5.1%	$[5.9, 6.0] \times 10^{-3}$
0.05	60	6.1×10^{-3}	7.9%	$[6.1, 6.2] \times 10^{-3}$

Table 5.3: Results for Example 5.1. $\delta_1 : 0.01$

is:

$$g(x_1, x_2, x_3) = (5x_1 + 10)^3 + (5x_2 + 9.9)^3 + (5x_3 + 5)^3 - 18 = 0 \qquad (5.7)$$

The probability of failure P_f^{act} using the actual LSF is 3.9×10^{-3}. The 95% confidence interval for P_f^{act} is $[3.9, 4.0] \times 10^{-3}$. The initial SVM constructed with 40 CVT samples, the updated SVM with $\delta_1 = \delta_2 = 10^{-3}$ and the actual function are shown in Figure 5.5. The convergence parameter δ_1 is set equal to 10^{-3} and δ_2 is varied from 0.1% to 5% to study its effect. The results are listed in Table 5.4. Figure 5.6 shows the convergence of the probability of failure Δ_{P_f} and the relative error E_{P_f}.

δ_2	N_t	P_f^{SVM}	E_{Pf}	P_f^{SVM} 95% CI
0.001	180	3.9×10^{-3}	1.0%	$[3.8, 3.9] \times 10^{-3}$
0.005	120	4.2×10^{-3}	8.2%	$[4.2, 4.3] \times 10^{-3}$
0.01	190	4.1×10^{-3}	4.3%	$[4.0, 4.1] \times 10^{-3}$
0.05	201	4.1×10^{-3}	5.6%	$[4.1, 4.2] \times 10^{-3}$

Table 5.4: Results for Example 5.2. $\delta_1 : 0.001$

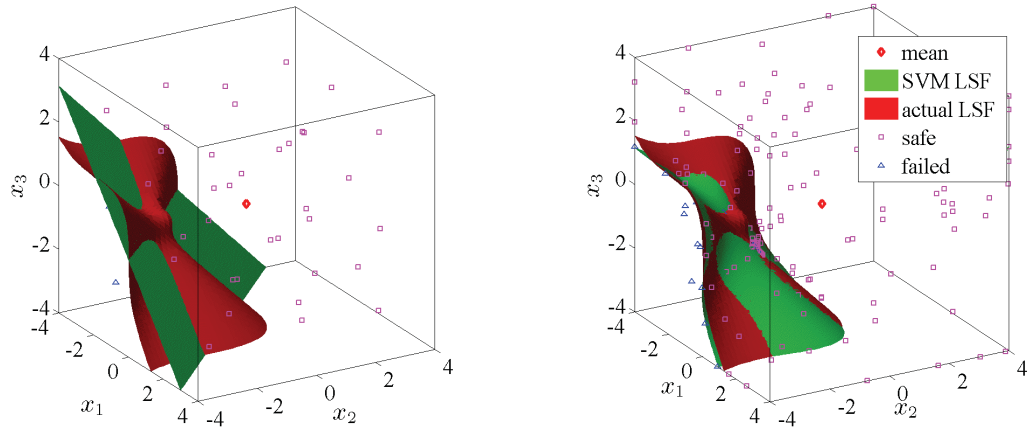

Figure 5.5: Example 5.2. Failure probability calculation at the origin for a three-dimensional non-convex LSF. The red surface represents the actual LSF. The green surfaces in left and right figures show the initial and final SVM approximations.

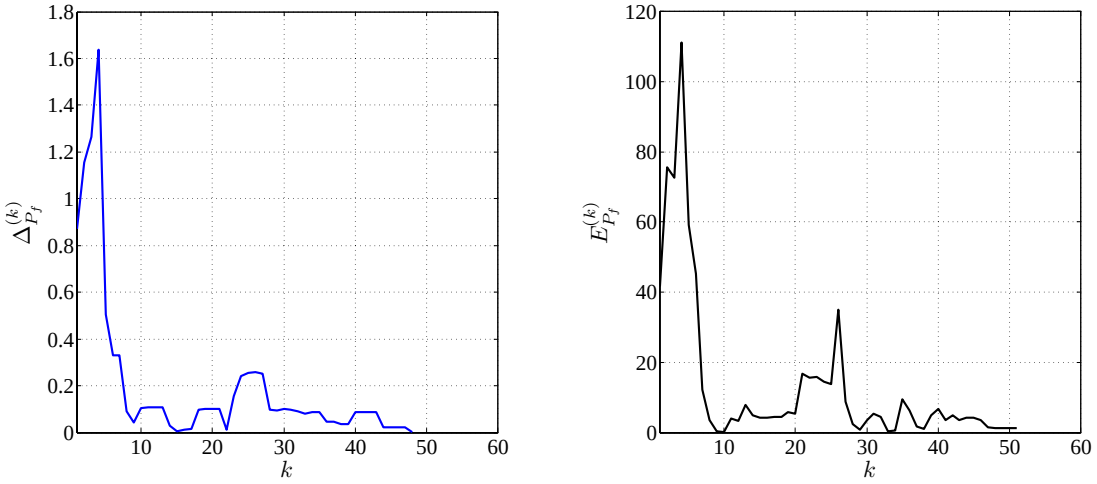

Figure 5.6: Example 5.2. Convergence (left) and relative error (right) of the probability of failure.

5.3.3 Example 4.3. RBDO example with two failure modes

This section presents a two-dimensional RBDO problem:

$$\begin{aligned} \min_{\bar{\mathbf{x}}} \quad & 2\bar{x}_1 + \bar{x}_2 \\ s.t. \quad & P_f = P(\mathbf{x} \in \Omega_f) \leq P_T \\ & 0 \leq \bar{x}_1 \leq 10 \\ & 0 \leq \bar{x}_2 \leq 10, \end{aligned} \tag{5.8}$$

where the failure domain Ω_f is the union of the regions defined by two inequalities (that is, two failure modes (Figure 5.7)):

$$\begin{aligned} \Omega_{f1} &: -x_1 + 2x_2 + 2 \leq 0 \\ \Omega_{f2} &: -x_1^2 + 6x_1 + x_2 - 8 \leq 0 \end{aligned} \tag{5.9}$$

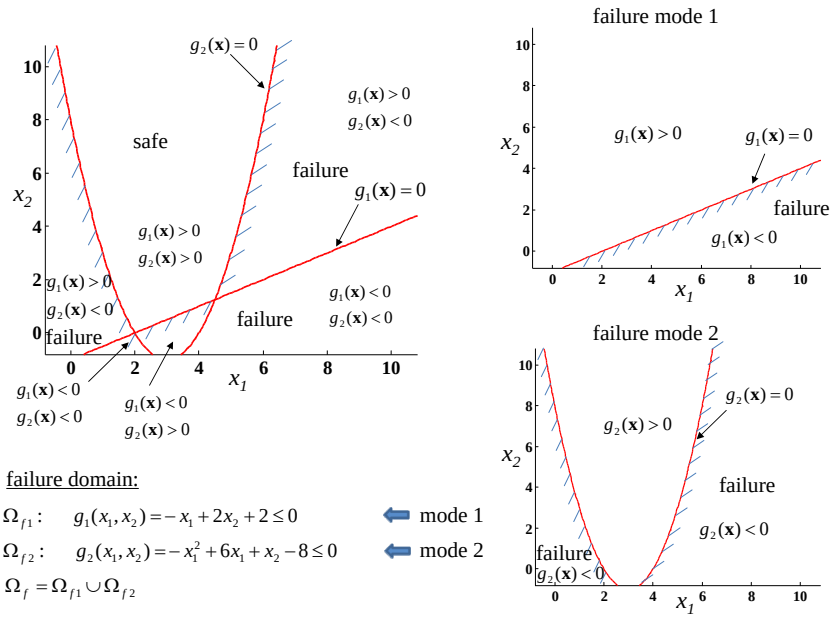

Figure 5.7: Failure domain for the RBDO example. The figures on right show the failure regions due to the two modes. The left figure shows the failure region due to combination of the two modes.

The target failure probability is $P_T = 10^{-3}$. Both variables x_1 and x_2 have normal distributions with standard deviation of 0.2. As explained in the methodology section, the ranges of both variables for the update are taken as $[-0.8, 10.8]$ in order to improve the accuracy of the failure probabilities at the design space boundaries. The initial DOE for s_d consists of 10 CVT samples. The training set for s_p consists of additional 40 CVT samples. However, it should be noted that the added samples for s_p do not require actual function evaluations. The value of ϵ_1 for convergence is set to 0.05. For the phase 2 update, δ_1 and δ_2 are set equal to 5×10^{-3}. Evolution of the objective function and constraint violation at the iterates, and the distances between successive iterates $\mathbf{x}_p^{(k)}$ and $\mathbf{x}_p^{(k-1)}$ are plotted in Figure 5.8. The results obtained after the end of phase 1 and phase 2 are given in Table 5.5. Figure 5.9 shows the updated SVM and the final optimum solution. In order to validate the usefulness of performing the update in two phases, the results of the proposed method are compared to an alternate scheme without the phase 1 update. The objective function values and constraint violations are compared at specific number of function evaluations using the two methods (Table 5.6).

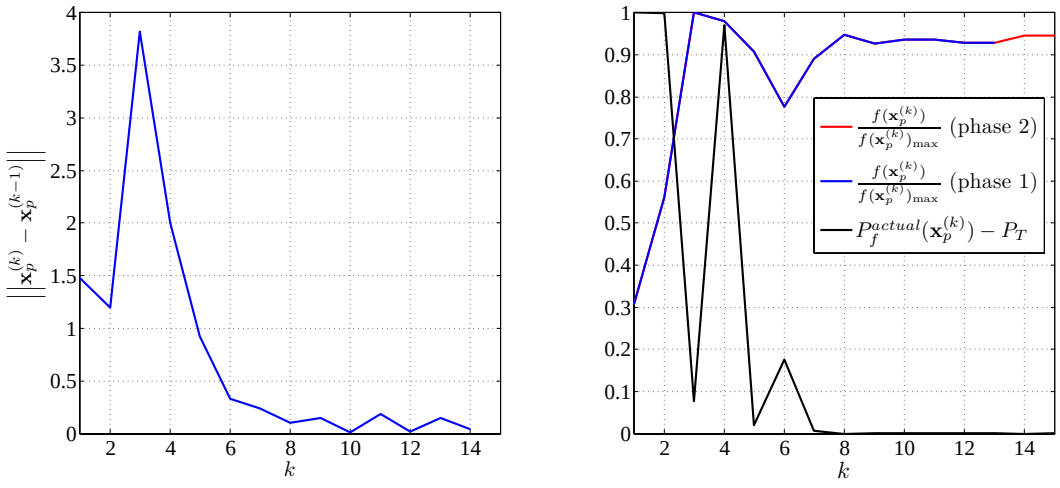

Figure 5.8: Example 5.3. Convergence of the distance between successive optima (left). Objective function and constraint violation (right) at $\mathbf{x}_p$. Objective function values are scaled by the maximum value among all iterations.

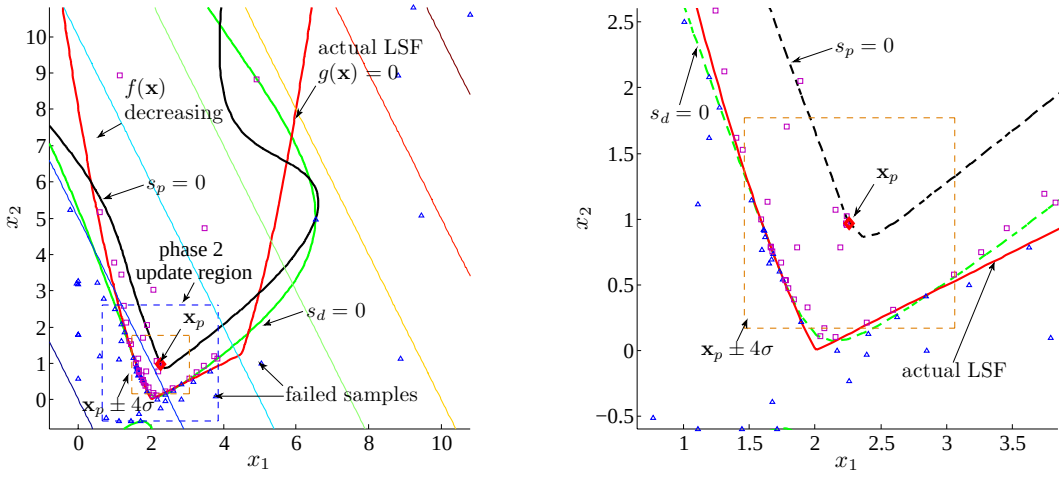

Figure 5.9: RBDO example. The left and right figures show the entire design space and the magnified region around $\mathbf{x}_p$ (final phase 2 update region). The $\pm 4\sigma$ region around $\mathbf{x}_p$ is also shown with the dashed brown lines.

Update stage	N_i	N_t	$\mathbf{x}_p$	f_{opt}	P_f^{SVM}	P_f^{act}
phase 1	10	70	$(2.136, 1.129)$	5.397	10^{-3}	2.6×10^{-3}
phase 2	70	88	$(2.262, 0.969)$	5.493	10^{-3}	10^{-3}

Table 5.5: Results for RBDO problem. Optimum solutions after phase 1 and phase 2 of the algorithm.

Evaluations	2-phase $f(\mathbf{x}_p)$	2-phase $P_f^{act}(\mathbf{x}_p)$	1-phase $f(\mathbf{x}_p)$	1-phase $P_f^{act}(\mathbf{x}_p)$
25	5.697	0.97	3.765	0.82
50	5.509	10^{-3}	4.22	1.0
75	5.397	2.6×10^{-3}	6.082	3.0×10^{-4}
88	5.493	10^{-3}	–	–
100	–	–	5.502	9.0×10^{-4}
113	–	–	5.506	10^{-3}

Table 5.6: Comparison of the proposed two-phase RBDO update to a scheme consisting of the phase 2 update only.

5.4 Discussion

The results in Section 5.3 show the efficacy of the update schemes to calculate probabilities of failure and to perform RBDO. The RBDO problem presents an

example of two critical failure modes. The use of the proposed classification-based approach presents an efficient method of handling multiple failure modes with a single SVM. This section presents a discussion on certain features of the update strategies.

- *Effect of performing RBDO in two phases:* As mentioned in the methodology section the objective of performing the RBDO update in two phases is to reduce the number of samples while retaining high accuracy. This is supported by the results in Table 5.6. The phase 1 update locates an approximate position of the optimum using limited samples. The region in the vicinity of the optimum is then further refined in phase 2 to provide accurate failure probabilities. On the contrary, if the phase 1 update is absent then more samples are required to obtain the final solution. This is because a comparatively larger number number of samples is spent during the initial iterations to refine regions of space that may be far from the optimum. The difference between the two methods is quite remarkable at the 50 and 75 function evaluation marks (Table 5.6). One of the consequences of locally refining the SVM and using relatively crude failure probabilities in phase 1 is that the global optimum may be missed in certain cases. That is, however, considered an acceptable risk with the objective of reducing function evaluations.

- *Effect of update scheme parameters:* The effect of parameters of the algorithm on the failure probability update is studied in Sections 5.3.1 and 5.3.2. The values of δ_1 and δ_2 are both varied between 0.1% to 5% for Example 5.1. The results suggest that value of more than 1% is too high or loose for the convergence criterion δ_1. The error increases for a higher value of δ_1, as expected. Increase in δ_2 for a fixed δ_1 also increases the error in general, for this example. For Example 5.2, the convergence parameter δ_1 is set equal to

10^{-3} and δ_2 is varied between 0.1% to 5%. However, no clear trend in the error variation with respect to δ_2 is seen in this case. The results of the two examples show that setting $\delta_1 = \delta_2 \leq 5 \times 10^{-3}$ provides reasonably accurate results.

- *Effect of Monte-Carlo samples:* The calculation of failure probabilities is performed using MCS. Failure probabilities are used to calculate the convergence measure during probability of failure update as well to construct the SVM s_p during the RBDO update. Therefore a high accuracy is required. 10^6 samples are used in this chapter, which allows an error of approximately 5% while calculating probabilities of the order 10^{-3}, based on 95% confidence level. The final probabilities are calculated using 10^7 MCS samples to provide a more accurate result. This allows for only 2% error for probabilities of order 10^{-3}. However, using 10^7 samples throughout the update can be time consuming and is avoided. An improvement to the proposed method can be obtained by using variance reduction techniques. Another important thing to note is that the MCS samples are not regenerated at every iteration. Instead, the same set of samples is used to calculate the probability at a given point. This is done in order to avoid confusing the change due to the SVM refinement and due to MCS error.

- *Looping of SVM:* Looping of SVM is a phenomenon that may occur in regions with lack of data. It leads to an additional artificial boundary, in the form of a loop, thus leading to potential misclassification of the space. Although such regions are efficiently identified and removed using the secondary samples, repeated occurrence of looping can lead to the waste of several function evaluations. Also, if looping occurs in a region with high probability content, it can lead to very high variations in the probability of failure, thus prolonging the convergence of the update. Such phenomenon is evident in the error plot

for Example 5.1 demonstrated in Figure 5.10. Removal of looping is a current issue that, if solved, can reduce the number of function evaluations for the update. One option is to populate certain regions of the space with dummy samples; however, such an approach has risks associated. Looping of SVM is, therefore, still an open issue that needs resolution in the future.

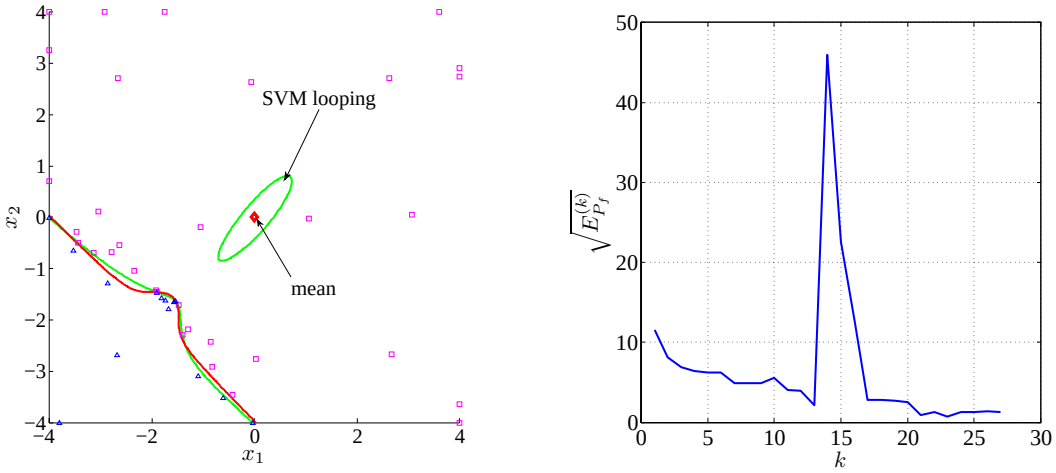

Figure 5.10: SVM looping in Example 5.1. The right figure shows the error with peak corresponding to looping. Square root is used to compress the plot for better visualization.

5.5 Concluding remarks

5.5.1 Summary

A methodology for failure probability calculation and RBDO is presented in this chapter. The methodology is based on the definition of explicit limit state functions using adaptively refined SVMs, and is applied to two and three-dimensional analytical problems. The probability of failure update scheme is implemented for predicting failure probabilities with highly nonlinear limit state functions. Efficacy of the proposed RBDO method is demonstrated using an example with two failure modes. The two limit state functions are represented by a single SVM boundary, which is refined through adaptive sampling. The decision boundaries for RBDO

are updated locally, which avoids function evaluations in the unimportant regions of the design space.

5.5.2 Future work

In the method presented in this chapter, limit state function evaluation is considered expensive whereas objective function evaluation is considered to be inexpensive. Possible improvements to the method include extension to expensive objective functions in the future. An approach similar to a deterministic optimization method presented in Chapter 7 can be used for this purpose. It is also possible to use PSVMs for the selection of samples. Specific criteria to identify intersections of limit state functions corresponding to different failure modes may also be used. One method to identify such regions could be based on the radius of curvature of the SVM boundary. In the future, the method will be applied to higher dimensional problems with several failure modes.

CHAPTER 6

RELIABILITY ASSESSMENT WITH MODIFIED PROBABILISTIC SUPPORT VECTOR MACHINES

In Chapter 4, the SVM-based EDSD method was introduced for limit state function approximation. Adaptive sampling techniques to enhance the accuracy of SVM boundaries were also presented in Chapters 4 and 5. However, in these methods, no attempt was made to quantify the prediction error of SVMs (except for known analytical functions). SVM-based limit-state functions may not always be accurate as they depend on the sampling, therefore leading to erroneous probability of failure estimates. This can be especially harmful if the probability of failure is underpredicted, as this may lead to an unsafe design. Considering the probability of misclassification by the SVM is, therefore, important. The method presented in this chapter quantifies the accuracy of the SVM classification model, using a probabilistic support vector machine (PSVM), and propagates this information to the calculation of the probability of failure. This modified, MCS-based, measure of probability, is constructed so that it is always more conservative compared to the probability of failure based on a deterministic SVM. Therefore, some of the consequences of an inaccurate limit-state function approximation are mitigated. The probability of failure estimate is based on a new sigmoid-based PSVM model along with the identification of a region where the probability of misclassification is large.

The organization of this chapter is as follows. Section 6.1 presents the PSVM-based reliability assessment method. Formulation of the failure probability accounting for the probability of misclassification by the SVM is explained in this section. Section 6.2 provides an overview of the conventional PSVM model and its limitations. In Section 6.3, the modified distance-based PSVM (DPSVM) model proposed

to overcome these limitations is presented. An error measure to compare the PSVM models is provided in Section 6.4. Finally, analytical test examples are presented in Section 6.5 to show the efficacy of the proposed method, followed by a discussion in Section 6.6.

6.1 Reliability assessment using probabilistic support vector machines (PSVMs)

In chapters 4 and 5, the methodology of calculating probabilities of failure using SVMs, based on MCS, was presented. The basic idea is to construct an explicit approximation of the failure boundary using SVM. The probability of failure based on a deterministic SVM is calculated using Equation 4.1. However, the construction of the SVM limit-state function is based on a DOE. Therefore, there is, in general, an error associated with the approximation of the boundary. This can result in an inaccurate probability of failure (Figure 6.1). Therefore, a PSVM-based method accounting for these errors is presented in this section.

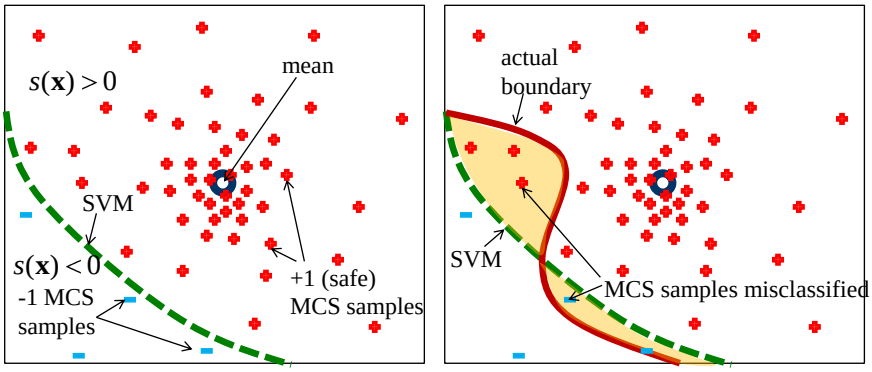

Figure 6.1: Calculation of the failure probability using an SVM (left). Misclassification of the MCS samples by the SVM (right).

In order to account for an inaccurate SVM, the probability of failure is calculated based on the probability of misclassification of the Monte-Carlo samples, in addition to the probability density functions of the random variables. The probability of misclassification is calculated using a PSVM. A modification of the basic sigmoid

model (Platt, J.C. (1999); Vapnik, V.N. (1998)), presented in Section 6.3, is used. A deterministic SVM only provides a binary classification. However, as explained in Chapter 3, a PSVM provides the probability that a particular configuration (or sample) will belong to a specific class (+1 or −1). This probability, which is the conditional probability of belonging to the +1 (resp. −1) class is denoted as $P(+1|\mathbf{x})$ (resp. $P(-1|\mathbf{x})$).

It is noticed in Equation 4.1 that the indicator function $I_g(\mathbf{x})$ can be interpreted as the probability of being in the failure class based on a deterministic SVM boundary. Therefore, the probability of being in the failure class −1 for any Monte-Carlo sample is either 0 or 1. It is equal to 0 for a sample lying in the safe or +1 class and equal to 1 for a sample lying in the failure or −1 class. The use of PSVM allows one to replace the binary indicator function by $P(-1|\mathbf{x})$, thus leading to:

$$P_f^{PSVM} = \frac{1}{N_{MCS}} \left(\sum_{i=1}^{N_{MCS}} P\left(-1|\mathbf{x}_i\right) \right) \tag{6.1}$$

A relatively conservative measure of the probability of failure is obtained if the probability of misclassification is considered only for the Monte-Carlo samples belonging to the safe class, that is, for $s(\mathbf{x}) > 0$:

$$P_f^{PSVM} = \frac{1}{N_{MCS}} \left(\sum_{i=1}^{N_{MCS}} \psi\left(-1|\mathbf{x}_i\right) \right),$$
$$\psi\left(-1|\mathbf{x}\right) = \begin{cases} 1 & s(\mathbf{x}) \leq 0 \\ P\left(-1|\mathbf{x}\right) & s(\mathbf{x}) > 0 \end{cases} \tag{6.2}$$

In Equation 6.2, the probability of misclassification is considered only for the samples lying in the safe domain based on the SVM ($s(\mathbf{x}) > 0$). Therefore, it would naturally provide a probability of failure that is greater than the one using the deterministic SVM (Equation 4.1). However, the failure probability estimate using Equation 6.2 may be over-conservative. Therefore, instead of considering a non-zero

$P(-1|\mathbf{x})$ for the entire safe domain, it is reasonable to consider it only in the regions with high probability of misclassification. Such regions can be identified as the ones "lacking data" in the vicinity of the SVM boundary. Therefore, the region Ω_{misc} for considering the probability of misclassification is a subset of the $s(\mathbf{x}) > 0$ regions. It is defined based on the distances to the closest $+1$ and -1 samples and the SVM margin, which does not contain any samples. Outside this region, the classification provided by the deterministic SVM is trusted (i.e. $P(-1|\mathbf{x})$ is either 1 or 0). The region Ω_{misc} for considering the probability of misclassification by the SVM is:

$$\Omega_{misc} = \Omega_{sd} \cap \Omega \left(|s(\mathbf{x})| < 1 \cup s(\mathbf{x})(d_+(\mathbf{x}) - d_-(\mathbf{x}) \geq 0) \right), \tag{6.3}$$

where Ω_{sd} is the safe domain based on the deterministic SVM, and $d_+(\mathbf{x})$ and $d_-(\mathbf{x})$ are the distances of $\mathbf{x}$ to the closest $+1$ and -1 training samples. Ω_{misc} consists of two kinds of regions in the $+1$ class. One is the SVM margin in the safe class and the other is the region with $d_+(\mathbf{x}) \geq d_-(\mathbf{x})$ (Figure 6.2). The probability of failure is given as:

$$P_f^{PSVM} = \frac{1}{N_{MCS}} \left(\sum_{i=1}^{N_{MCS}} \gamma\left(-1|\mathbf{x}_i\right) \right),$$

$$\gamma\left(-1|\mathbf{x}\right) = \begin{cases} 1 & \mathbf{x}_i \in \Omega_f \\ 0 & \mathbf{x}_i \in \Omega_{sd} - \Omega_{misc} \\ P\left(-1|\mathbf{x}\right) & \mathbf{x}_i \in \Omega_{misc} \end{cases} \tag{6.4}$$

The calculation of the failure probability using Equation 6.4 requires the calculation of $P(-1|\mathbf{x})$ in the region Ω_{misc}. The details of the PSVM models for calculating $P(+1|\mathbf{x})$ or $P(-1|\mathbf{x})$ are presented in the following sections.

6.2 Review of the basic sigmoid PSVM model

The basic sigmoid PSVM model given by Platt (Platt, J.C. (1999)) was presented in Chapter 3. In this model, the conditional probability $P(+1|\mathbf{x})$ is represented as

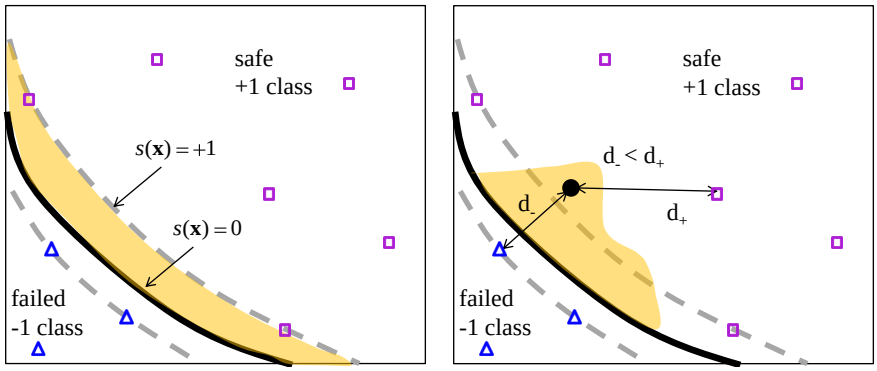

Figure 6.2: Definition of the region Ω_{misc} for considering the probability of misclassification. Ω_{misc} is the union of the two shaded regions in the left and the right figures.

a function of two parameters A and B:

$$P(+1|\mathbf{x}) = \frac{1}{1 + e^{As(\mathbf{x})+B}} \tag{6.5}$$

where $A < 0$. The conditional probability of the -1 class is equal to $1 - P(+1|\mathbf{x})$.

One of the limitations of the basic sigmoid model is that it depends only on the SVM values and not on the spatial distribution of the samples. As a result, if the classes of the evaluated samples are considered deterministic, it does not satisfy one of the conditions that requires $P(+1|\mathbf{x})$ to be either 0 or 1 at these samples. Instead, it provides a probability of misclassification even for the samples that have already been evaluated. An example of the probability of misclassification P_{misc} using Platt's sigmoid model is shown in Figure 6.3. It is seen that the probability of misclassification is high even for regions that are far from the boundary and consist of already evaluated samples. A modified PSVM model is presented in the following section to overcome this limitation.

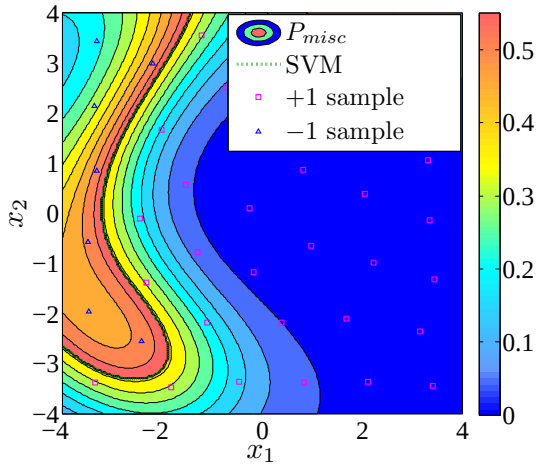

Figure 6.3: Map of the probabilities of misclassification using Platt's sigmoid model.

6.3 Improved distance-based probabilistic support vector machines (DPSVMs) using a modified sigmoid model

The biggest limitation of the basic sigmoid PSVM stems from neglecting the spatial distribution of the evaluated samples. To overcome this issue, a modified sigmoid model is presented in this section. The proposed model depends not only on the SVM values, but also on the distances to the evaluated samples used to train the SVM. Because the proposed model depends on the spatial distribution of the samples, it is also referred to as the distance-based probabilistic support vector machine (DPSVM). It is assumed in this model that the class of any evaluated sample is deterministic. The modified sigmoid model is defined as:

$$P\left(+1|\mathbf{x}\right) = \frac{1}{1+e^{As(\mathbf{x})+B(\frac{d_-}{d_++\tau}-\frac{d_+}{d_-+\tau})}} \qquad A < \frac{-3}{\min\left(s_{max}, -s_{min}\right)}, B < 0, \quad (6.6)$$

where d_- and d_+ are the distances to the closest -1 and +1 samples. τ is a small quantity (set equal to 10^{-100} in this work) added in order to avoid numerical issues at the evaluated training samples.

The proposed model satisfies the following conditions:

- $P\left(+1|\mathbf{x}\right) \to 1$ if $s(\mathbf{x}) \to \infty$ or $d_+ \to 0$

- $P(+1|\mathbf{x}) \to 0$ if $s(\mathbf{x}) \to -\infty$ or $d_- \to 0$
- $P(+1|\mathbf{x}) \to 0.5$ if $s(\mathbf{x}) \to 0$ and $d_- \to d_+$

The upper bound on A ensures that $P(+1|\mathbf{x})$ does not have a strong dependence on the distances away from the boundary. That is, the values of $P(+1|\mathbf{x})$ are close to 0 or 1 far from the boundary, irrespective of the influence of the distances. More specifically, for $B = 0$, the upper bound ensures $P(+1|\mathbf{x}) > 0.95$ at the point of maximum SVM value s_{max} and $P(+1|\mathbf{x}) < 0.05$ at the point of minimum SVM value s_{min}. The demonstration for the former is given below by setting $B = 0$:

$$\begin{aligned} \frac{1}{1+e^{As_{max}}} &> 0.95 \\ \Rightarrow A < \frac{ln\left(\frac{0.05}{0.95}\right)}{s_{max}} &= \frac{-2.94}{s_{max}} \approx \frac{-3}{s_{max}} \end{aligned} \tag{6.7}$$

The strict inequality bound on B ($B < 0$) ensures that $P(+1|\mathbf{x}) \to 1$ at the $+1$ samples and $P(+1|\mathbf{x}) \to 0$ at the -1 samples. The demonstration for the two cases is given below.

For the $+1$ samples, $d_+ \to 0$:

$$P(+1|\mathbf{x}) = \lim_{\tau \to 0} \frac{1}{1+e^{As(\mathbf{x})+B(\frac{d_-}{\tau})}} \approx \frac{1}{1+e^{As(\mathbf{x})-\infty}} \approx \frac{1}{1+e^{-\infty}} \approx 1 \tag{6.8}$$

For the -1 samples, $d_- \to 0$:

$$P(+1|\mathbf{x}) = \lim_{\tau \to 0} \frac{1}{1+e^{As(\mathbf{x})+B(-\frac{d_+}{\tau})}} \approx \frac{1}{1+e^{As(\mathbf{x})+\infty}} \approx \frac{1}{1+e^{+\infty}} \approx 0 \tag{6.9}$$

The training process for the DPSVM is as follows.

- The value of d_+ for a $+1$ sample and that of d_- for a -1 sample are zero. However, during the training process, these are assigned as the distances to the closest $+1$ and -1 samples other than the sample under consideration.

- The values of $s(\mathbf{x})$, d_- and d_+ at training samples are used to calculate the likelihood function, which is then maximized to find A and B. The optimization is solved using a genetic algorithm (GA).

Because the proposed DPSVM model accounts for both the SVM values and the spatial distribution of the evaluated samples, it overcomes the limitations of the basic SVM model mentioned in Section 6.2. A conceptual graphical comparison of the proposed model with Platt's sigmoid model is provided in Figure 6.4. A map of the probabilities of misclassification is shown in Figure 6.5.

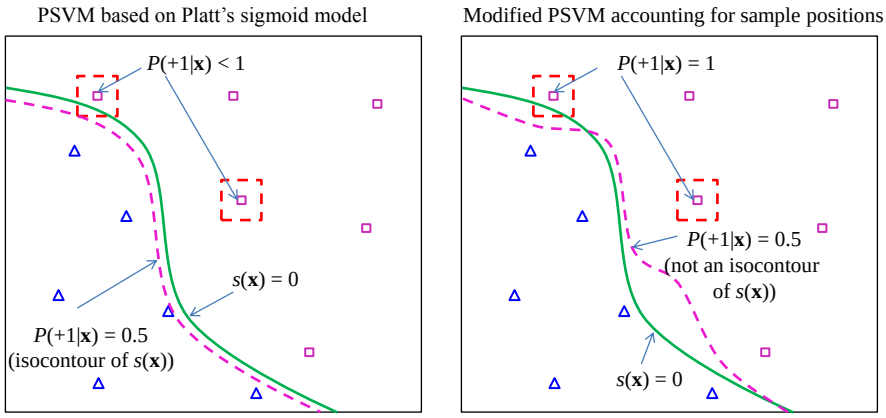

Figure 6.4: Comparison of the two PSVM models.

6.4 Error quantification of the PSVM model

In order to compare the proposed DPSVM model with Platt's PSVM model, a measure to quantify the error for the models is presented in this section. In the case where the actual limit-state function is known, $P(+1|\mathbf{x})$ is known for any point and is equal to 0 or 1. Therefore the error of the PSVM model can be calculated at any point. A large number of test points from a uniform grid are used for this purpose. Because the actual class of the all the test points is known, the probability

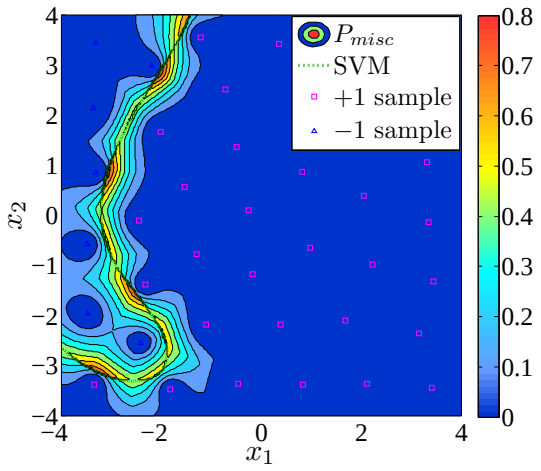

Figure 6.5: Map of the probabilities of misclassification using the modified distance-based sigmoid model. A comparison with the probabilities in the Figure 6.3 shows major differences.

of misclassification for the i^{th} point is:

$$P_{misc}(\mathbf{x}_i) = \begin{cases} 1 - P(+1|\mathbf{x}_i) & y_i = +1 \\ P(+1|\mathbf{x}_i) & y_i = -1 \end{cases}, \tag{6.10}$$

where $\mathbf{x}_i$ represents the i^{th} test point with class label y_i. A good PSVM model should provide a low probability of misclassification for the test points. The error E_{test} is defined as the mean probability of misclassification for all the test points:

$$E_{test} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} P_{misc}(\mathbf{x}_i) \tag{6.11}$$

6.5 Examples

This section presents analytical test examples to compare the PSVM models as well as to demonstrate the PSVM-based failure probability measure. The examples consist of two and three variables. For each example, the actual limit-state functions are approximated with SVMs constructed with CVT DOEs. Two studies are performed for each example - comparison of the two PSVM models and the study of the proposed failure probability measure. For each example, the ratio

of the probability of failure obtained with the PSVM models to the probability obtained with the deterministic SVM is provided. This ratio, referred to as the probability ratio (PR), provides a measure of the conservativeness of the PSVM model compared to the deterministic SVM. The size of the DOE is varied to study its effect on the PSVM and the probability of failure. As explained in Section 6.4, a uniform grid is used to quantify the efficacy of the PSVM models. For comparing the probabilities of failure, all the variables are assumed to have truncated Gaussian distributions with mean equal to 0 and standard deviation equal to 1.0. The lower and upper bounds of all the variables are -4.0 and 4.0. The probabilities of failure are calculated using 10^6 MCS samples for all the examples.

The following notations are used in this section:

- E_{test}^{Platt}: test point based error for the Platt PSVM model
- $E_{test}^{Modified}$: test point based error for the modified DPSVM model
- P_f^{Actual}: probability of failure calculated using the actual limit-state function
- P_f^{SVM}: probability of failure calculated using the SVM limit-state function
- P_f^{Platt}: PSVM-based probability of failure calculated using the Platt model
- $P_f^{Modified}$: PSVM-based probability of failure calculated using the DPSVM model
- $\epsilon_{P_f}^{SVM}$: relative difference between P_f^{SVM} and P_f^{actual}
- $\epsilon_{P_f}^{Platt}$: relative difference between P_f^{Platt} and P_f^{actual}
- $\epsilon_{P_f}^{Modified}$: relative difference between $P_f^{Modified}$ and P_f^{actual}
- PR: probability ratio $\frac{P_f^{Modified}}{P_f^{SVM}}$ or probability ratio $\frac{P_f^{Platt}}{P_f^{SVM}}$. Ratio of failure probability obtained with PSVM model to probability obtained with deterministic SVM.

6.5.1 Example 6.1 - two dimensional problem

This example consisting of two variables $x_1, x_2 \in [-4, 4]$ has two failure modes. The failure region is defined as:

$$\Omega_f = (-8(x_1 - 2) + x_2^2 \leq 0) \cap (x_2 - \tan\left(\frac{\pi}{12}\right)(x_1 + 7) + 4 \leq 0) \tag{6.12}$$

The two modes and the resulting limit-state function are shown in Figure 6.6. With the SVM-based approach, both the modes are represented by a single boundary.

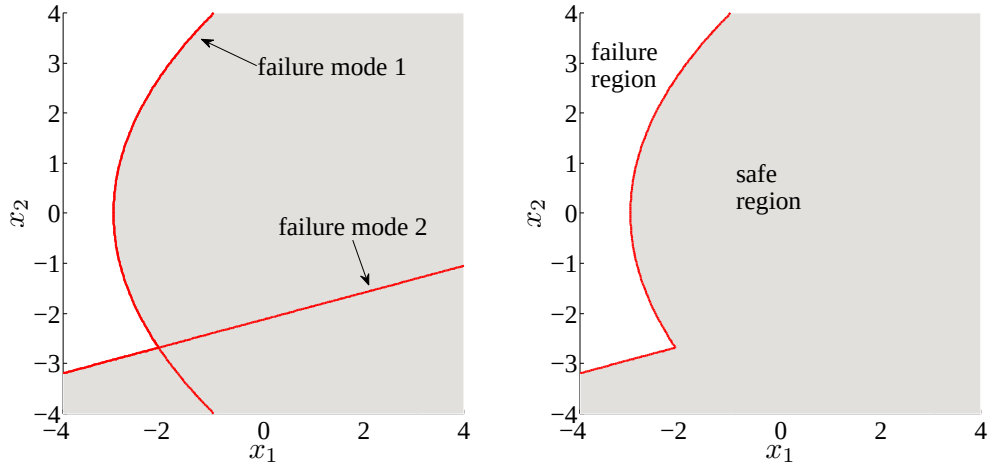

Figure 6.6: Example 6.1. The left figure shows the limit-state functions due to the two failure modes. The right figure shows the net failure region. For this example, the system is considered failed if it fails based on both modes.

In order to study the effect of the DOE, SVM approximations are constructed using 40 to 100 CVT DOE samples with increments of 20. A comparison of the Platt model and the proposed distance-based PSVM model is provided in Section 6.5.1. Additionally, the probabilities of failure are provided in Section 6.5.1.

Comparison of the PSVM models

The two PSVM models are compared in this section based on the measure provided in Section 6.4. 1600 grid points are used to calculate the errors due to the two

models. One example of the distribution of P_{misc} values in the space for 40 samples is shown in Figures 6.3 and 6.5. The errors E_{test}^{Platt} and $E_{test}^{Modified}$ are shown in Figure 6.7. Clearly, the proposed modified PSVM model provides lower errors irrespective of the size of the DOE.

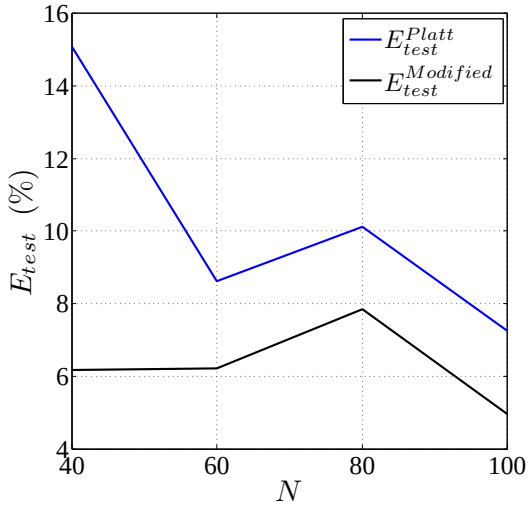

Figure 6.7: Example 6.1. Testing error for the PSVM models.

Comparison of the probabilities of failure

The probabilities of failure using varying sized DOEs are provided in Figure 6.8. The dashed-dotted green curve represents the probability of failure calculated using the deterministic SVMs. Clearly, there is significant variation in the failure probability depending on the DOE. The PSVM-based probabilities of failure and the relative differences with respect to P_f^{actual} using the two PSVMs are also shown in Figure 6.8 with the dashed blue and the solid black curves. It is seen from the figures that the deterministic SVM-based failure probability is less than the actual value (red) for several cases. The PSVM-based failure probabilities are always more conservative than the deterministic SVM case as demonstrated by the probability ratios depicted in Figure 6.9.

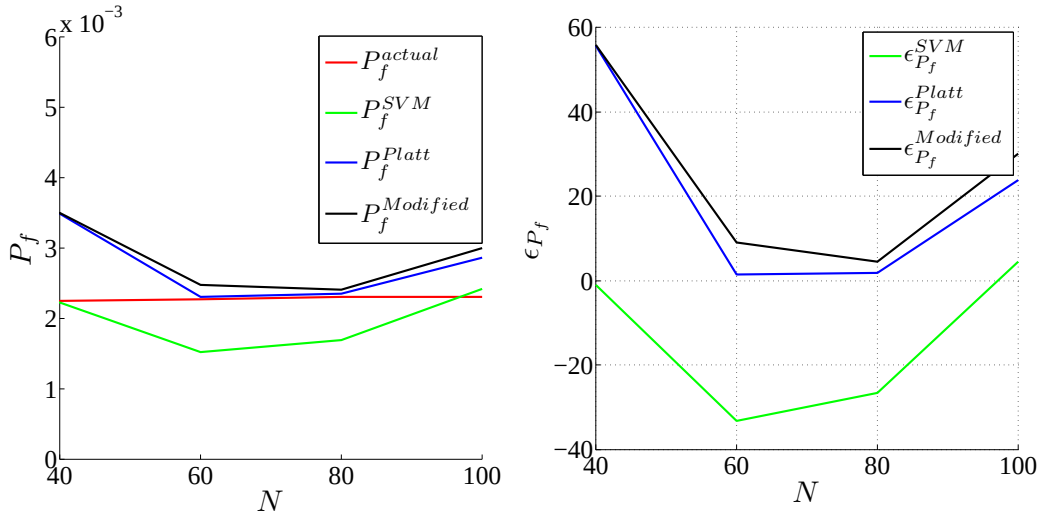

Figure 6.8: Example 6.1. Probabilities of failure.

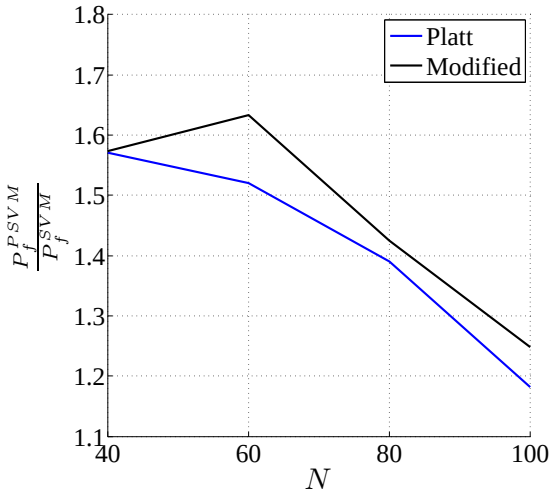

Figure 6.9: Example 6.1. Probability ratios with respect to the deterministic SVM-based failure probability.

6.5.2 Example 6.2 - three dimensional problem

A three variable example consisting of four disjoint regions in the space (Figure 6.10) is presented in this section. All the variables lie between -4 and 4. The

failure region is given as:

$$\Omega_f = (x_1 - 2)^2 + x_2^2 + (x_3 + 2)^2 - 3x_1x_2x_3 + 1 \leq 0 \tag{6.13}$$

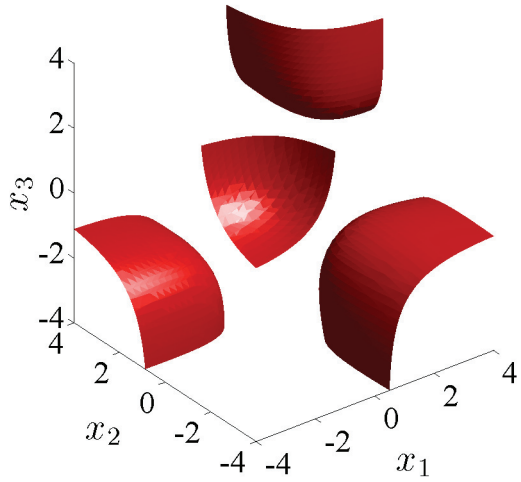

Figure 6.10: Example 6.2. Actual limit-state function.

The SVM approximation of the limit-state function is constructed using 40 to 1000 samples at an interval of 40. The corresponding PSVMs based on the two models are compared in Section 6.5.2. The probabilities of failure are provided in Section 6.5.2.

Comparison of the PSVM models

The comparison of the two PSVM models, based on testing points, is provided in this section. A grid consisting of 64000 points is used. The errors are shown in Figure 6.11. Similar to Example 6.1, the proposed modified PSVM model provides lower errors irrespective of the size of the DOE. This again shows the superiority of the proposed model. Additionally, the difference is especially prominent for smaller DOEs that represent lack of data in the space.

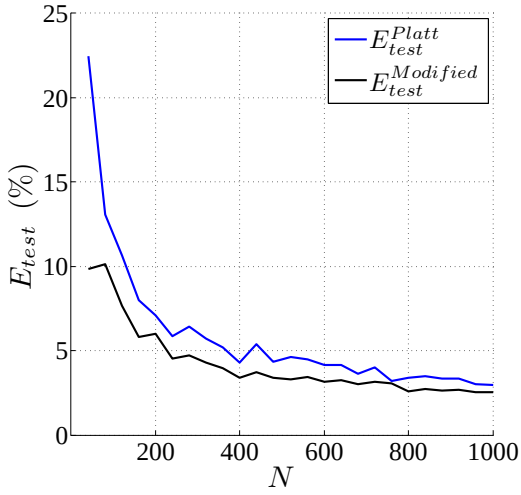

Figure 6.11: Example 6.2. Testing error for the PSVM models.

Comparison of the probabilities of failure

Similar to Example 6.1, it is seen in Figure 6.12 that the probability of failure calculated using the deterministic SVMs (dashed-dotted green) show significant variation with respect to the size of the DOE. Also, the failure probability is less than the actual value (red) in several cases. The PSVM-based failure probabilities are always more conservative than the deterministic SVM case as demonstrated by the probability ratios depicted in Figure 6.13. As expected, these ratios reduce with the number of samples because the confidence on the SVM increases.

6.6 Discussion

This section presents a discussion on the results presented in Section 6.5. First, the results of the comparison between the two PSVM models shows a very clear trend. Unlike the modified model, the Platt model does not satisfy the condition of having probabilities equal to 0 or 1 at the evaluated samples. The comparison of the errors E_{test}^{Platt} and $E_{test}^{Modified}$ also shows lower errors for the proposed model for both the examples. Both the errors reduce with the size of the DOE.

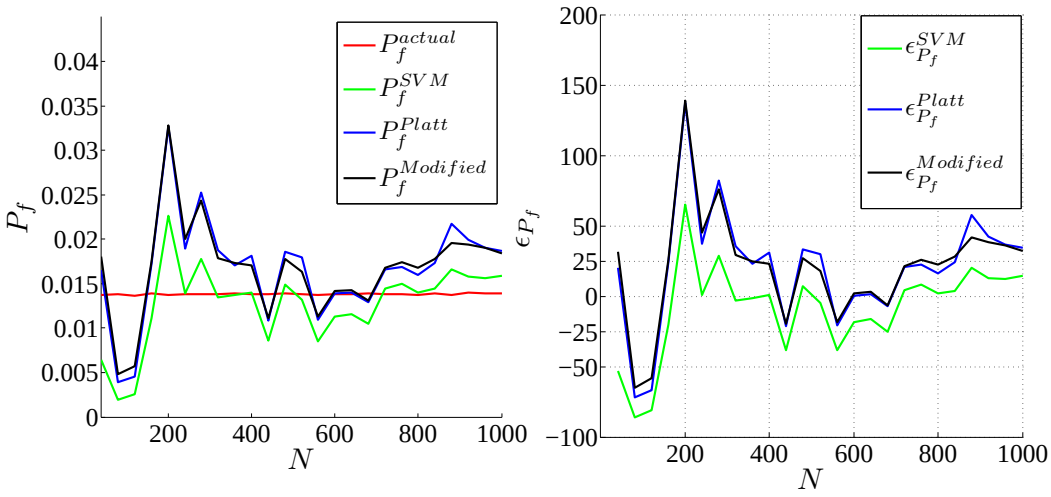

Figure 6.12: Example 6.2. Probabilities of failure.

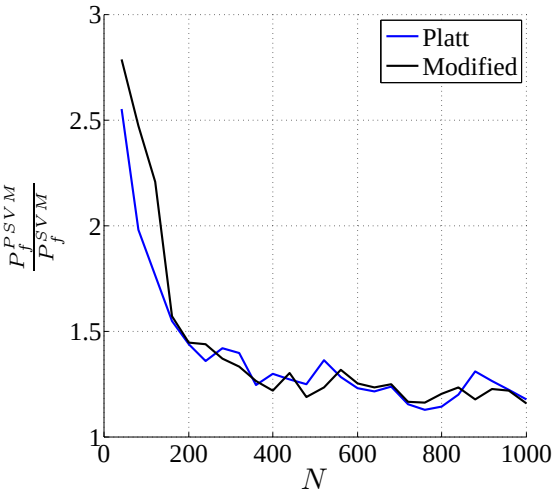

Figure 6.13: Example 6.2. Probability ratios with respect to the deterministic SVM-based failure probability.

A study of the failure probabilities using the deterministic SVMs shows significant variation with the DOE size. The uncertainty in the SVM approximations supports the need for considering the error associated with them. Therefore, the PSVM models are used to provide a probability of failure that accounts for these errors. The PSVM-based failure probability is calculated such that it is always more conservative than the one calculated with the deterministic SVM. Figures 6.9 and

6.13 show the ratio of the PSVM-based failure probabilities to the probabilities using the deterministic SVMs. The ratio is always greater than 1. In fact, the probability ratios are higher when the sparsity is greater and reduces with the size of the DOE. This indicates that the confidence in the SVM increases with the amount of data, as expected. It is, therefore much more meaningful than using an arbitrary constant safety factor. The comparison of the failure probabilities using the Platt and the modified PSVMs does not indicate a very large difference. This is because the probability of misclassification is considered only in the region Ω_{misc}, which is small when compared to the entire space. In general, the more conservative failure probability among the two (Platt and Modified PSVM) may be used. In addition to considering the probability of misclassification by the SVM, it is also useful to consider the variance of the MCS. Therefore, the 99% confidence intervals of the MCS failure probability estimates for Example 6.5.1 are also shown in Figure 6.14 .

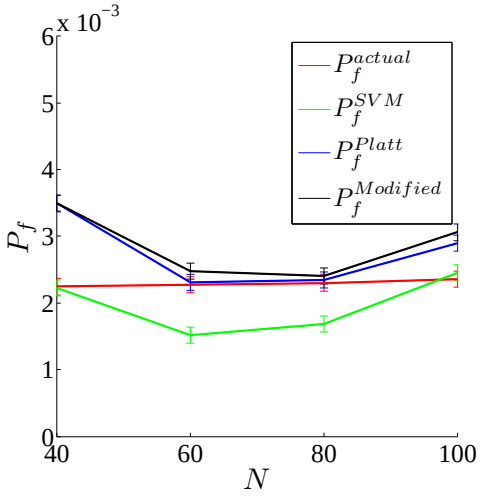

Figure 6.14: 99% confidence interval of the MCS failure probability estimates for Example 6.1.

In the studies performed in this chapter, significant variations of the failure probabilities were observed with respect to the size of the CVT DOEs. Apart from using PSVMs for quantifying the probability of misclassification, another option

to reduce the errors is to use adaptive sampling for the SVMs (Chapters 4 and 5). Adaptive sampling will increase the accuracy of the SVM, and also reduce the variation in the approximation by placing additional samples in the important regions. The probabilities of failure based on deterministic and probabilistic SVMs are expected to converge to the same value when a large number of adaptive samples is used. It should be noted that even with adaptive sampling, the SVM may not always be accurate as there might not be enough samples. If there are limitations on the number of samples due to high computation cost, then PSVM will allow the flexibility to use a relaxed convergence criterion. This is because the PSVM-based failure probability is always larger than the one using deterministic SVM, and even with fewer samples it is more likely to lead to a conservative estimate.

A new sampling scheme is also possible based on the proposed modified PSVM model. Because it provides the probability of misclassification, samples may be added in regions with high misclassification probability. One such scheme is used in the following chapter.

6.7 Concluding remarks

6.7.1 Summary

A method for reliability assessment using PSVMs was presented in this chapter. The main idea is to include the probability of misclassification of Monte-Carlo samples in the failure probability calculation. The proposed failure probability measure was shown to be more conservative than the deterministic SVM. Apart from the failure probability measure, an improved PSVM model was also presented in this chapter. Its efficacy to predict the probability of misclassification by the SVM was demonstrated through the example problems.

6.7.2 Future work

The next steps of this research will study higher dimensional examples. In addition, methods to use PSVM for adaptive sampling will be explored. The first steps towards a PSVM-based adaptive sampling scheme have been implemented in Chapter 7. Another possibility of improvement is in the definition of the region Ω_{misc}. Currently this region is defined based on a nearest neighbor algorithm. There is scope to improve this by using k nearest neighbours, or some gradient information to smoothen the boundaries of this region.

CHAPTER 7

CONSTRAINED EFFICIENT GLOBAL OPTIMIZATION USING SVMs

In previous chapters, the use of SVM for the construction of decision boundaries was demonstrated. These boundaries were used for reliability assessment and RBDO. However, the objective functions for optimization were given by analytical functions in these chapters. This may not be the case in general, i.e. evaluation of the objective function may also be expensive. In this chapter, a deterministic optimization method is presented that addresses both expensive constraints as well as objective functions. More specifically, a methodology for constrained Efficient Global Optimization (EGO) using Support Vector Machines (SVMs) is presented. While the objective function is approximated using Kriging as in the original EGO formulation (Jones, D.R. et al. (1998)), the "zero-level" of the constraints is approximated explicitly as a function of the design variables using an SVM. The use of SVM for constraint handling allows one to take advantage of the benefits of the EDSD approach, as outlined in Chapter 4. Because the constraint response is not approximated, this approach alleviates issues due to discontinuous or binary responses. More importantly, several constraints can be represented using one unique SVM, thus considerably simplifying constrained problems. In order to account for constraints, an SVM-based probability of feasibility is introduced in the optimization formulation. The probability is calculated using the modified probabilistic SVM (PSVM) model presented in Chapter 6. Several constrained EGO formulations are implemented and compared. For instance, formulations with global and local update regions of the SVM boundary are presented. The adaptive sampling scheme also consists of samples selected based on the probability of misclassification obtained using the PSVM.

The organization of this chapter is as follows. The previous EGO methodologies

are reviewed in Section 7.1. The proposed methodology for EGO with explicit SVM constraints, using PSVM-based probability of feasibility, is presented in Section 7.2. Finally, analytical example problems from the literature (Sasena, M.J. (2002)) are considered in Section 7.3 to validate the efficacy of the proposed methodology. For the sake of completeness, the details of the derivation of the expected improvement are given in the Appendix B, at the end of the dissertation.

7.1 Review of efficient global optimization (EGO)

The EGO formulation for unconstrained optimization was presented in Section 2.6.3. The original method is based on the maximization of expected improvement (EI) (Jones, D.R. et al. (1998)), but several variations of the sample selection method are also found in the literature (Sasena, M.J. (2002); Forrester, A.I.J. et al. (2008)). Constrained formulations of EGO have also been implemented, some of which are reviewed in this section.

In order to include constraints in the optimization, several EGO formulations have been proposed. In Schonlau, M. (1997), Schonlau proposed the multiplication of the EI with the probability of feasibility, calculated using the Kriging approximation of the constraint responses. In the case of multiple constraints, the probability is given by the product of the probability of feasibility of each constraint. A method based on the expected improvement of a penalized objective function was proposed in Sasena, M.J. et al. (2002). Both these methods had limitations and Sasena, M.J. (2002) proposed another approach which involves the maximization of the EI with sample constrained to lie in the feasible space. The feasible space was defined based on the mean values of the Kriging models for the constraint functions. Another way of handling the constraints involves the use of the expected violation (EV) Audet, C. et al. (2000). The EV is calculated in the same way as the EI and provides a measure of the expected amount by which a constraint is violated. It is then used to penalize the EI while locating the samples.

A common feature of all the above methods lies in the use of Kriging approximations for both the objective function, as well as for each constraint. Therefore, these methods would be hampered by discontinuous or binary constraint functions. Also, the propagation of the error in calculating the probability of feasibility is multiplicative for multiple constraint problems. The calculation of probability of feasibility is based on the assumption of independence of the constraints. In order to overcome these issues, the SVM-based approach presented in the following section is used in this work for approximating the constraints.

7.2 Constrained EGO using PSVMs

This section describes various candidate formulations of the proposed constrained EGO methodology. The core methodology is based on the EI as well as the probability of feasibility. In this work, the focus is on handling the constraints, i.e. to provide a method for assessing the probability of feasibility. This probability is calculated using the modified probabilistic SVM (PSVM) model presented in Chapter 6. $+1$ being the feasible class, the probability of feasibility is equal to $P(+1|\mathbf{x})$ (Equation 6.9). It should be noted that although the methods presented in this chapter are based on just the EI, in terms of handling the objective function, other more recent criteria (Sasena, M.J. (2002); Forrester, A.I.J. et al. (2008)) may also be used to improve the methodology. This would only require replacing the EI with the desired criterion. The optimization schemes implemented are divided into three categories:

- **Update scheme 1**. Constrained optimization problem formulated as an unconstrained problem that maximizes the product of the EI and the probability of feasibility.
- **Update scheme 2**. Maximization of the EI with a constraint based on an SVM approximation.

- **Update scheme 3**. Local approach with update region. Sequential two step approach involving the selection of a sample based on the formulations of update scheme 1 or 2 and a step to define samples dedicated to the local refinement of the SVM boundary.

The main characteristic of the proposed approach lies in the handling of the constraints. Instead of approximating each constraint response, an SVM boundary describing the "zero-level" contour of the constraint is constructed. It is critical to understand the difference between existing constrained EGO approaches (Section 7.1) and the proposed approach. While the former approach tries to approximate the actual constraint function values, an SVM is only interested in the isocontour that defines the boundary of the feasible region. In order to construct the SVM, the feasible class is labeled as $+1$ and the infeasible class is labeled -1. The use of an SVM for constraint handling has two major implications:

- Since this method eliminates the need to approximate the constraint function, it can handle problems with discontinuous or binary functions.

- A single SVM is used to define all the optimization constraints. Unlike the previous methodologies (Section 7.1), this method does not require the multiplication of the probabilities of feasibility for individual constraints. Therefore, the multiplicative error propagation in calculating the probability is avoided for multiple constraint problems. In addition, since only the classification of the samples is needed, the evaluation of all the constraint functions may not be required.

7.2.1 Update scheme 1

The update scheme 1 is similar to the probability adjusted EI method in Schonlau, M. (1997). The iterates are selected by solving an unconstrained formulation that maximizes the product of the EI and the probability of feasibility $P(+1|\mathbf{x})$. Thus, a sample is selected if it has a high EI and a high probability of being feasible.

However, unlike the original method Schonlau, M. (1997), the value of $P(+1|\mathbf{x})$ is calculated using a PSVM (Equation 6.9).

$$\max_{\mathbf{x}} \quad EI(\mathbf{x})P(+1|\mathbf{x}) \tag{7.1}$$

The objective function in Equation 7.1 can have several local optima. In this work, a Genetic Algorithm (GA) is used to solve the problem. The initial population is a CVT DOE consisting of $100m$ samples, m being the dimensionality. GA being a stochastic method, there can be variation in the optimization results. A deterministic method such as the branch and bound implementation in the DIRECT software may also be used.

7.2.2 Update scheme 2

This scheme is a modification of the method in Sasena, M.J. (2002), in which the solution of a constrained optimization was proposed to avoid the difficulties associated with the probability scaling of the EI. Unlike the update scheme 1, it simply uses the EI in the objective function. In Sasena, M.J. (2002), the constraints were approximated using Kriging models and the mean values of these models were used to guide the optimizer in the feasible space. In the SVM-based approach, the Kriging mean, which approximates the constraint responses over the whole space, is replaced by an SVM that approximates the zero-level contour of the constraint. Again, it is important to remember that only one SVM is needed for several constraints, whereas several Kriging models will be needed if the response approximation approach is chosen.

Clearly, the proposed approach will be highly dependent on the quality of the SVM approximation because the samples are selected only in the regions defined as feasible by the current constraint approximation. Therefore, if the constraint is not accurate, this may lead to a low rate of convergence to the actual optimum. To overcome such problems, a PSVM-based constraint is used to locate the samples.

The samples are selected in regions with given minimum probability of being feasible (Figure 7.1). In addition to the current SVM boundary, this probability also depends on the spatial distribution of the samples (Equation 6.9).

$$\begin{aligned} \max_{\mathbf{x}} \quad & EI(\mathbf{x}) \\ s.t. \quad & \delta_{pp} - P(+1|\mathbf{x}) \leq 0 \end{aligned} \tag{7.2}$$

where δ_{pp} is a given threshold of the probability of feasibility. The optimization is solved using a GA with the same starting population as update scheme 1.

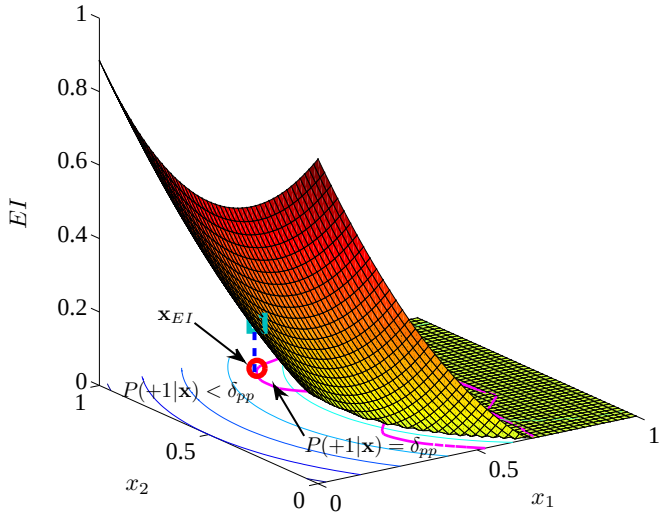

Figure 7.1: Selection of a sample using the maximization of EI in regions with a minimum threshold probability of feasibility.

7.2.3 Update scheme 3

The selection of samples using the schemes 1 and 2 is performed globally over the whole space. This third update scheme investigates the local refinement of the SVM constraint approximation. The basic idea is depicted in the Figure 7.2. After a sample from update scheme 1 or 2 is found, additional samples are added in the vicinity of the current iterate, with the purpose of refining the SVM boundary and improving the estimate of $P(+1|\mathbf{x})$. This scheme, consisting of an extra step to locally refine

the SVM, is referred to as the "update scheme 3" in the remainder of the paper. It is categorized as the scheme 3a or 3b depending on whether the first sample is selected using Equation 7.1 (update scheme 1) or Equation 7.2 (update scheme 2).

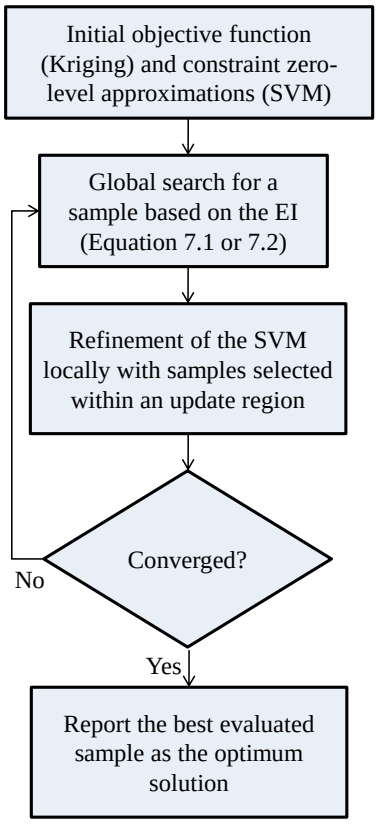

Figure 7.2: Basic summary of the update scheme 3.

While the EI maximization sample $\mathbf{x}_{EI}$ (Equation 7.1 or 7.2) explores the space globally, the additional samples, referred to as "primary" and "secondary" samples, are selected in a hyperspherical "update region" centered at the current iterate $x^\star$. The radius R_u of the update region is:

$$R_u = max\left(||\mathbf{x}_{EI} - x^\star||, d_+, d_-, 0.5\left(\frac{V}{N}\right)^{\frac{1}{m}}\right) \tag{7.3}$$

where d_+ and d_- are the distances to the closest $+1$ and -1 samples, V is the volume of the design space, N is the number of samples and m is the number of

design parameters.

The summary of the update scheme 3 is given in Algorithm 7.1. The objective of primary samples is to sample sparse regions with high probabilities of misclassification (incorrect class prediction) by the SVM (Figure 7.3). The secondary samples are evaluated to avoid non-uniformity of the samples belonging to the two classes in the vicinity of the SVM constraint (Figure 7.4). The selection of secondary samples is performed using Equations 4.7-4.9. The primary samples are selected using a new PSVM-based method. In order to avoid deviation from the main message of this chapter, details of the PSVM-based primary sample selection are presented in Appendix B. The primary and secondary samples ($\mathbf{x}_p$ and $\mathbf{x}_s$) can be evaluated in parallel with the EI maximization sample $\mathbf{x}_{EI}$ (Equation 7.1 or 7.2).

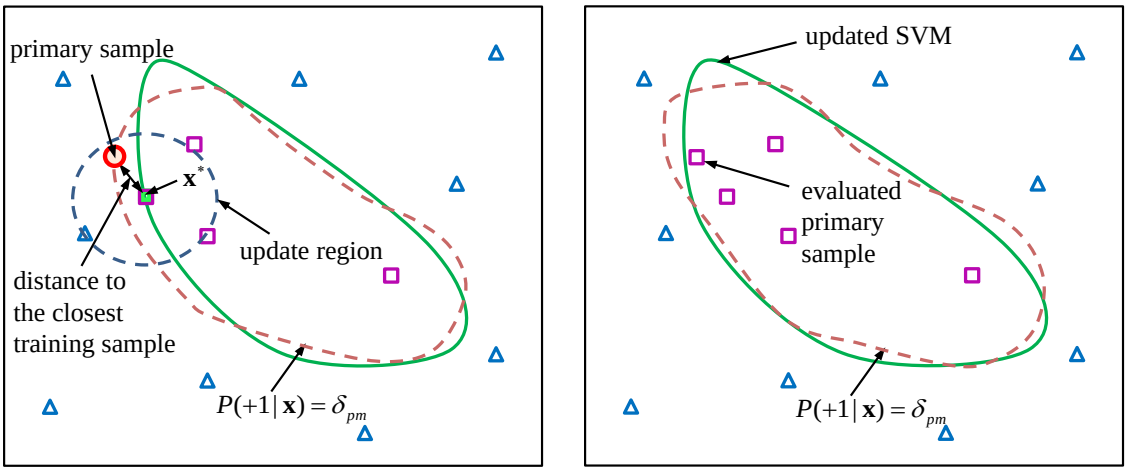

Figure 7.3: Update of the SVM constraint due to a primary sample.

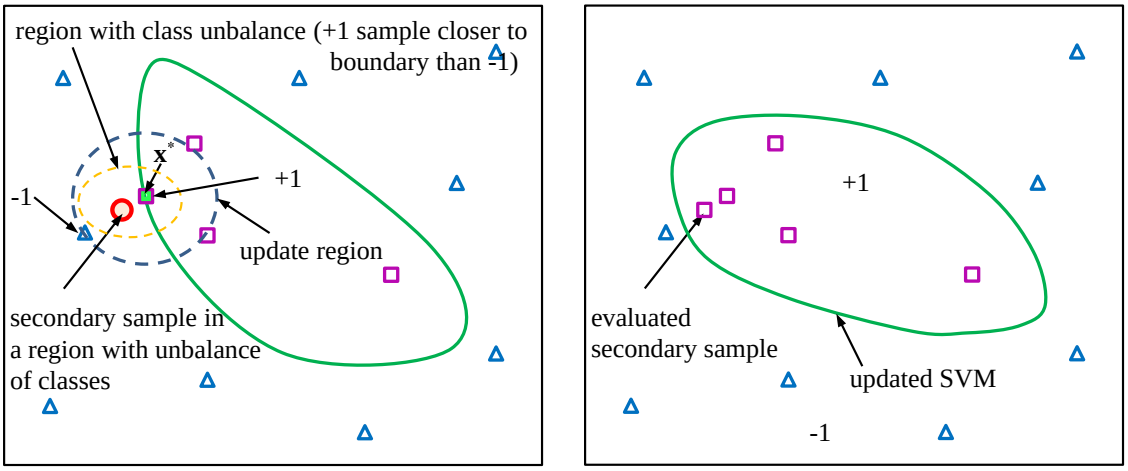

Figure 7.4: Update of the SVM constraint due to a secondary sample.

Algorithm 7.1: update scheme 3

1: Sample the space with a CVT DOE.
2: Evaluate the objective function at each sample.
3: Construct the initial Kriging model for the objective function.
4: Evaluate the constraint function(s) at each sample.
5: Classify the samples into two classes (e.g. feasible and infeasible) based on the constraint function values. The classification is performed using a threshold value or a clustering technique if discontinuities are present.
6: Construct the initial SVM boundary that separates the classified samples.
7: Calculate the parameters of the PSVM model using maximum likelihood.
8: Set iteration $k = 0$
9: **repeat**
10: Locate the current best solution among the evaluated samples. Set $f^{\star}$ equal to the objective function value at this sample.
11: Select a sample based on the EI and $P(+1|\mathbf{x})$ using the Equation 7.1 (scheme 3a) or the Equation 7.2 (scheme 3b).

12: Define the center and the radius of the "update region".
13: **for** $i = 1$ to $i = n_p$ **do**
14: Select a primary sample in the update region. For the first iteration, the entire space is the update region.
15: **end for**
16: **for** $i = 1$ to $i = n_s$ **do**
17: Select a secondary sample in the update region.
18: **end for**
19: Update the Kriging model for the objective function.
20: Evaluate the constraint function(s) at the selected samples and reconstruct the SVM and the PSVM with the new information.
21: $k = k + 1$
22: Calculate the convergence measure.
23: **until** convergence

7.3 Examples

In this section, two test examples are presented to validate the proposed methodologies. The examples presented are taken from the literature Sasena, M.J. (2002). The optimization problems consisting of analytical objective functions and constraints have the following form:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ s.t. \quad & g_i(\mathbf{x}) \leq 0 \qquad i = 1, 2, 3... \end{aligned} \tag{7.4}$$

For each problem presented, the actual optimal solution is known. A comparison of the different update schemes is provided for each example. For each example, the initial DOE consists of 10 CVT samples and the update is run for 50 iterations to study the convergence. The value of δ_{pp} in Equation 7.2 is initially set equal to 0.5.

If no solution with a non-zero EI is found, then δ_{pp} is decreased until a solution is found. The following notations are used in this section:

- k is the iteration number. The number of samples selected during each iteration depends on the update scheme.
- ϵ_k^f is the relative error of the optimum objective function value at the k^{th} iteration.
- $\mathbf{x}^\star$ and $f^\star$ are the predicted optimum and the corresponding objective function value.
- $\mathbf{x}^\star_{actual}$ and $f^\star_{actual}$ are the actual optimum and the corresponding objective function value.

7.3.1 Example 7.1: three constraint problem

This example consists of two design variables x_1 and x_2 with identical ranges $[0, 1]$. The optimization problem, consisting of three constraints, is defined as:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) = -(x_1 - 1)^2 - (x_2 - 0.5)^2 \\ s.t. \quad & g_1(\mathbf{x}) = ((x_1 - 3)^2 + (x_2 + 2)^2)e^{(-x_2^7)} - 12 \leq 0 \\ & g_2(\mathbf{x}) = 10x_1 + x_2 - 7 \leq 0 \\ & g_3(\mathbf{x}) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 - 0.2 \leq 0 \end{aligned} \tag{7.5}$$

Figure 7.5 shows the individual constraints and the resulting feasible region. The objective function contours and the optimum solution are also plotted. The problem has two optima at (0.2316,0.1216) and (0.2017,0.8332). The latter one is the global optimum with an objective function value -0.7483.

The update schemes 1-3 are used to find the optimum. The update schemes 3a and 3b are run with three combinations of the primary and secondary samples

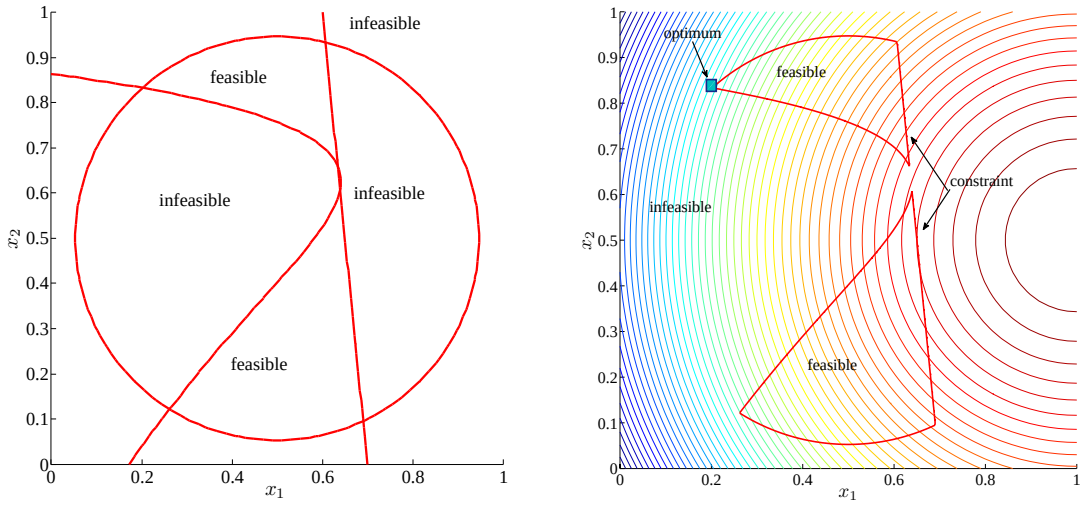

Figure 7.5: Example 7.1. Actual individual constraint boundaries and the resulting feasible and infeasible regions (left). Objective function contours, the constraint boundary and the the optimum solution (right).

($n_p = 2$ and $n_s = 0$,$n_p = 0$ and $n_s = 2$,$n_p = n_s = 1$). As a reminder, these samples are used to update (and refine) the approximation of the SVM boundary. The results are listed in Tables 7.1-7.3. The initial relative error in $f^\star$ is 51.2%. Three sets of results at several iterations are provided. It is seen that in most cases, the optimizer converges to the global optimum. However, there are a few cases where the local optimum is found. Figure 7.6 shows the evolution of $f^\star$. The SVM constraints after the update are plotted in Figures 7.7-7.10. For completeness, the results are compared to those using Kriging approximation of the constraints Sasena, M.J. (2002) in the Appendix B.

7.3.2 Example 7.2: two constraint problem

The feasible space for this problem is defined by two constraints. The objective function, the constraints and the optimum solution are shown in the Figure 7.11. Both the design parameters x_1 and x_2 lie in the range $[-2, 2]$. The actual optimum is at $(0.5955, -0.4045)$ with an objective function value of 289.85. The optimization

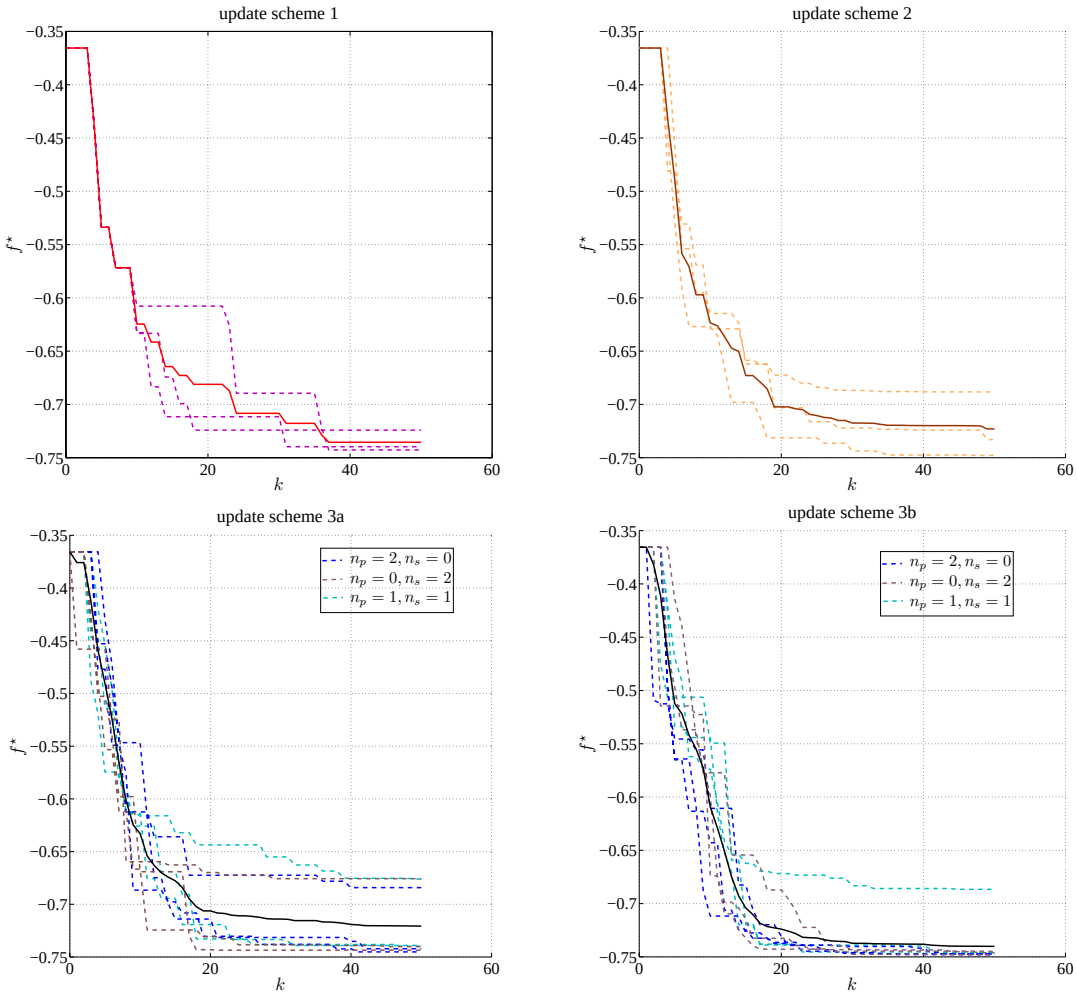

Figure 7.6: Example 7.1. The evolution of $f^\star$ using the update schemes $1-3$. The solid curves represent the mean of three runs for each case.

problem is:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) = (1 + A(x_1 + x_2 + 1)^2)(30 + B(2x_1 - 3x_2)^2) \\ where \quad & A = 19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2, \\ and \quad & B = 18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2 \\ s.t. \quad & g_1(\mathbf{x}) = -3x_1 + (-3x_2)^3 \leq 0 \\ & g_2(\mathbf{x}) = x_1 - x_2 - 1 \leq 0 \end{aligned} \tag{7.6}$$

The optimization is solved using the methodologies explained in the Section 7.2.

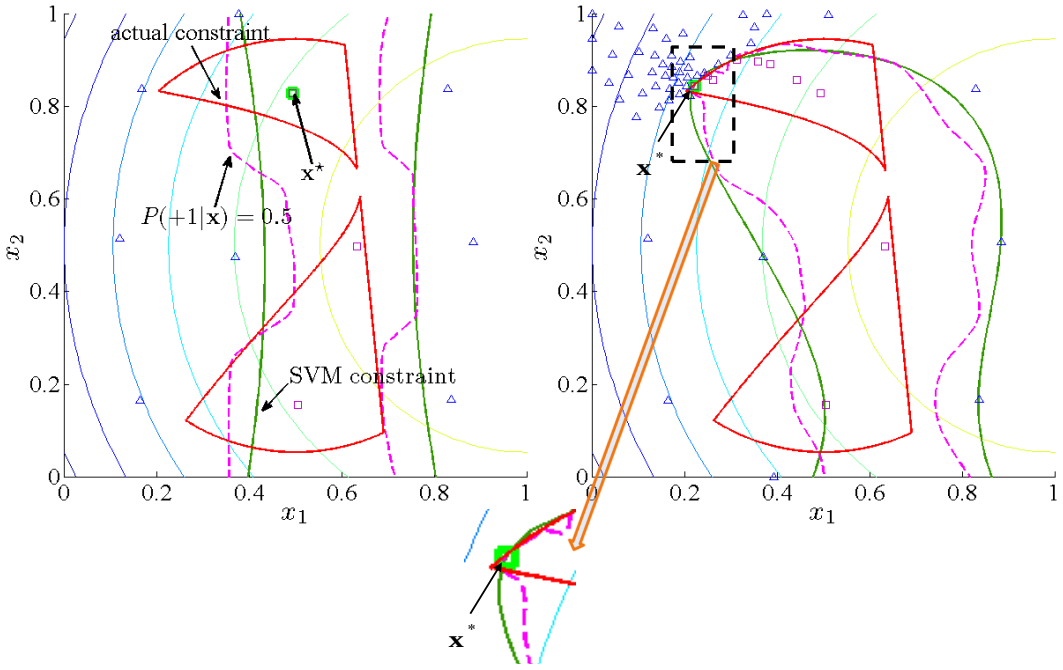

Figure 7.7: Example 7.1. Update scheme 1. SVM (green) and PSVM (dashed magenta) at $k = 0$ (left) and $k = 50$ (right) and the actual constraint (red).

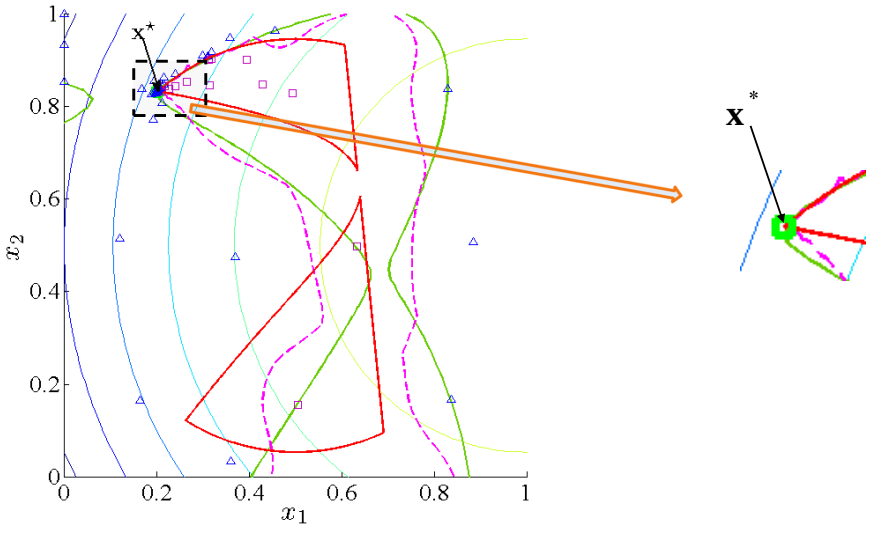

Figure 7.8: Example 7.1. Update scheme 2. SVM (green) and PSVM (magenta dashed) at $k = 50$ and the actual constraint (red).

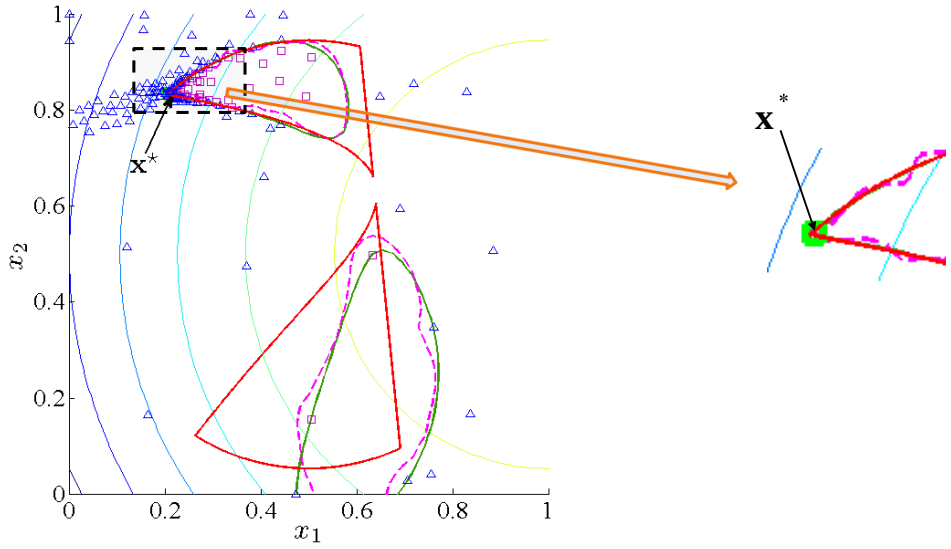

Figure 7.9: Example 7.1. Update scheme 3a with $n_p = 1$ and $n_s = 1$. SVM (green cunrve) and PSVM (mageta dashed) at $k = 50$ (left). Magnified view of the region in the vicinity of $\mathbf{x}^\star$ (right). The updated SVM is locally accurate.

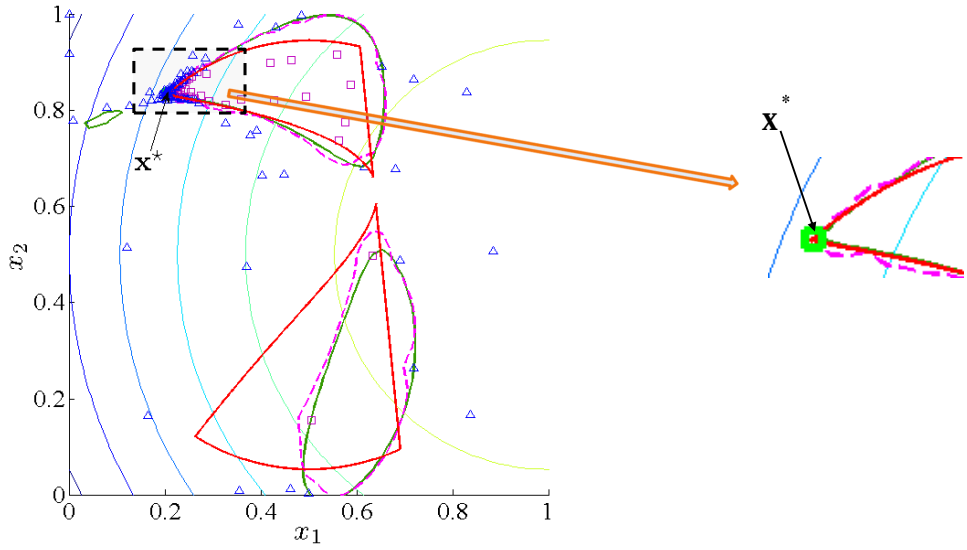

Figure 7.10: Example 7.1. Update scheme 3b with $n_p = 1$ and $n_s = 1$. SVM (green curve) and PSVM (magenta dashed) at $k = 50$ (left). Magnified view of the region in the vicinity of the optimum (right). The SVM boundary is very similar to the actual constraint (red curve).

Scheme	$\epsilon_{10}^f\%$	$\epsilon_{20}^f\%$	$\epsilon_{30}^f\%$	$\epsilon_{40}^f\%$	$\epsilon_{50}^f\%$
1	18.8	18.8	7.9	0.8	0.8
2	16.2	2.3	0.6	0.1	0.1
3a ($n_p = 2, n_s = 0$)	18.2	10.1	10.1	8.6	8.6
3a ($n_p = 0, n_s = 2$)	11.8	10.5	9.7	9.7	9.7
3a ($n_p = 1, n_s = 1$)	17.7	14.0	12.5	9.8	9.6
3b ($n_p = 2, n_s = 0$)	4.9	1.4	0.2	0.2	0.2
3b ($n_p = 0, n_s = 2$)	22.9	8.2	0.5	0.5	0.5
3b ($n_p = 1, n_s = 1$)	26.6	1.3	1.3	1.1	0.2

Table 7.1: Example 7.1. Relative error in $f^\star$ at specific iterations from first run.

Scheme	$\epsilon_{10}^f\%$	$\epsilon_{20}^f\%$	$\epsilon_{30}^f\%$	$\epsilon_{40}^f\%$	$\epsilon_{50}^f\%$
1	15.4	3.3	3.3	3.3	3.3
2	17.9	6.0	3.5	3.3	2.1
3a ($n_p = 2, n_s = 0$)	26.9	4.6	2.2	1.8	0.8
3a ($n_p = 0, n_s = 2$)	10.6	0.7	0.6	0.6	0.6
3a ($n_p = 1, n_s = 1$)	18.0	3.9	2.0	1.2	1.2
3b ($n_p = 2, n_s = 0$)	14.1	1.6	1.1	0.	0.1
3b ($n_p = 0, n_s = 2$)	20.1	2.1	0.7	0.6	0.3
3b ($n_p = 1, n_s = 1$)	23.3	10.2	8.7	8.4	8.2

Table 7.2: Example 7.1. Relative error in $f^\star$ at specific iterations from second run.

Scheme	$\epsilon_{10}^f\%$	$\epsilon_{20}^f\%$	$\epsilon_{30}^f\%$	$\epsilon_{40}^f\%$	$\epsilon_{50}^f\%$
1	15.4	4.9	4.9	1.2	1.2
2	15.9	10.1	8.2	8.0	8.0
3a ($n_p = 2, n_s = 0$)	8.2	2.4	1.4	0.8	0.4
3a ($n_p = 0, n_s = 2$)	10.9	2.3	1.3	1.2	1.1
3a ($n_p = 1, n_s = 1$)	16.4	2.1	1.4	1.4	1.1
3b ($n_p = 2, n_s = 0$)	18.4	2.4	0.3	0.1	0.01
3b ($n_p = 0, n_s = 2$)	9.9	0.7	0.1	0.1	0.05
3b ($n_p = 1, n_s = 1$)	27.3	1.4	0.4	0.4	0.3

Table 7.3: Example 7.1. Relative error in $f^\star$ at specific iterations from third run.

As explained before, the feasible region defined by the two constraints is approximated using a single SVM. Results of the optimization at several iterations are

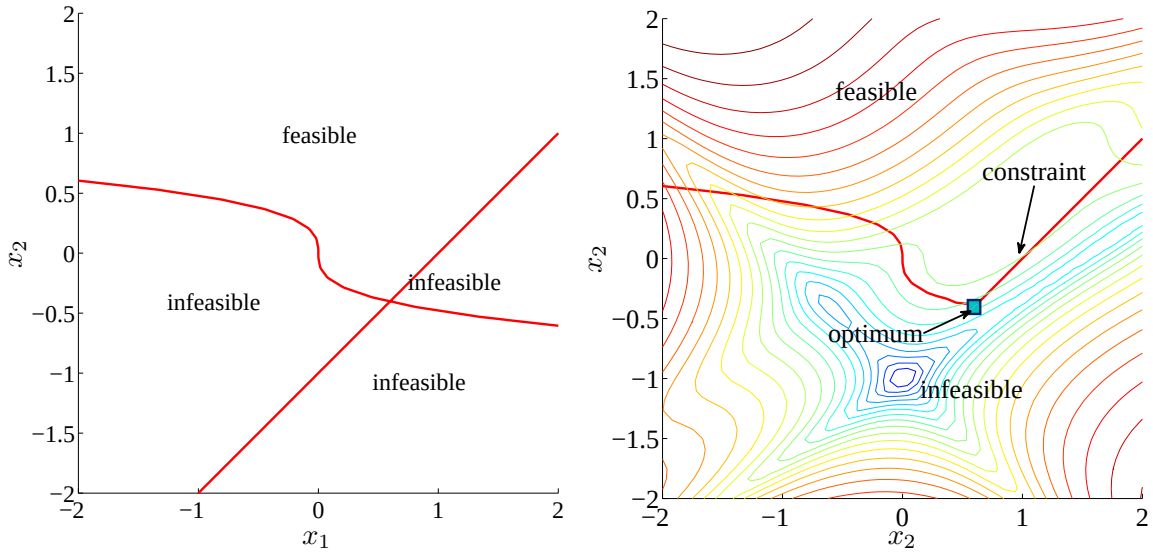

Figure 7.11: Example 7.2. Individual constraint boundaries and the resulting feasible and infeasible regions (left). Objective function contours, the constraint boundary and the the optimum solution (right).

listed in Tables 7.4-7.6. The initial error is 261.0%. The evolution of $f^\star$ using the different update schemes is plotted in Figure 7.12.

Scheme	$\epsilon^f_{10}\%$	$\epsilon^f_{20}\%$	$\epsilon^f_{30}\%$	$\epsilon^f_{40}\%$	$\epsilon^f_{50}\%$
1	202.6	33.2	33.2	33.2	6.2
2	219.7	20.6	17.3	5.4	2.0
3a $(n_p = 2, n_s = 0)$	126.8	32.8	22.1	22.1	22.1
3a $(n_p = 0, n_s = 2)$	128.0	31.4	18.0	11.4	7.8
3a $(n_p = 1, n_s = 1)$	61.5	29.6	14.0	11.6	11.6
3b $(n_p = 2, n_s = 0)$	53.7	6.7	0.3	0.3	0.3
3b $(n_p = 0, n_s = 2)$	39.4	20.2	3.7	3.7	3.7
3b $(n_p = 1, n_s = 1)$	111.8	15.6	6.9	1.0	1.0

Table 7.4: Example 7.2. Relative error in $f^\star$ at specific iterations from first run.

Scheme	$\epsilon_{10}^{f}\%$	$\epsilon_{20}^{f}\%$	$\epsilon_{30}^{f}\%$	$\epsilon_{40}^{f}\%$	$\epsilon_{50}^{f}\%$
1	202.6	33.3	33.3	6.8	6.8
2	224.3	83.9	75.4	10.8	1.8
3a ($n_p = 2, n_s = 0$)	97.7	28.0	6.9	6.9	6.9
3a ($n_p = 0, n_s = 2$)	145.3	12.6	12.6	6.2	2.6
3a ($n_p = 1, n_s = 1$)	116.3	31.5	31.5	15.5	5.9
3b ($n_p = 2, n_s = 0$)	40.7	4.7	0.1	0.1	0.1
3b ($n_p = 0, n_s = 2$)	52.9	9.3	4.3	0.1	0.1
3b ($n_p = 1, n_s = 1$)	126.7	6.2	6.2	1.5	1.2

Table 7.5: Example 7.2. Relative error in $f^\star$ at specific iterations from second run.

Scheme	$\epsilon_{10}^{f}\%$	$\epsilon_{20}^{f}\%$	$\epsilon_{30}^{f}\%$	$\epsilon_{40}^{f}\%$	$\epsilon_{50}^{f}\%$
1	202.6	32.0	32.0	32.0	13.9
2	180.7	58.3	40.1	30.8	29.0
3a ($n_p = 2, n_s = 0$)	83.6	29.2	26.2	4.5	4.5
3a ($n_p = 0, n_s = 2$)	88.3	27.9	25.4	11.9	11.9
3a ($n_p = 1, n_s = 1$)	112.5	50.8	15.9	15.9	15.9
3b ($n_p = 2, n_s = 0$)	38.5	6.3	6.3	2.9	0.6
3b ($n_p = 0, n_s = 2$)	126.6	27.9	3.6	0.8	0.4
3b ($n_p = 1, n_s = 1$)	54.4	2.8	2.8	0.7	0.3

Table 7.6: Example 7.2. Relative error in $f^\star$ at specific iterations from third run.

7.4 Discussion

This section presents a discussion on the results presented in Section 7.3. Apart from interpreting the results, the merits of the proposed technique are also discussed.

The results of the two example problems in Section 7.3 show the efficacy of the proposed method to locate the optimum. In particular, the examples show the applicability of the method to multiple constraint problems. The usefulness of the proposed methodology for handling multiple constraint problems is as expected because it simplifies such problems by replacing all the constraints with a single SVM constraint boundary approximation. For Example 1, the results of the proposed SVM-based method are compared to those using Kriging-based constraint

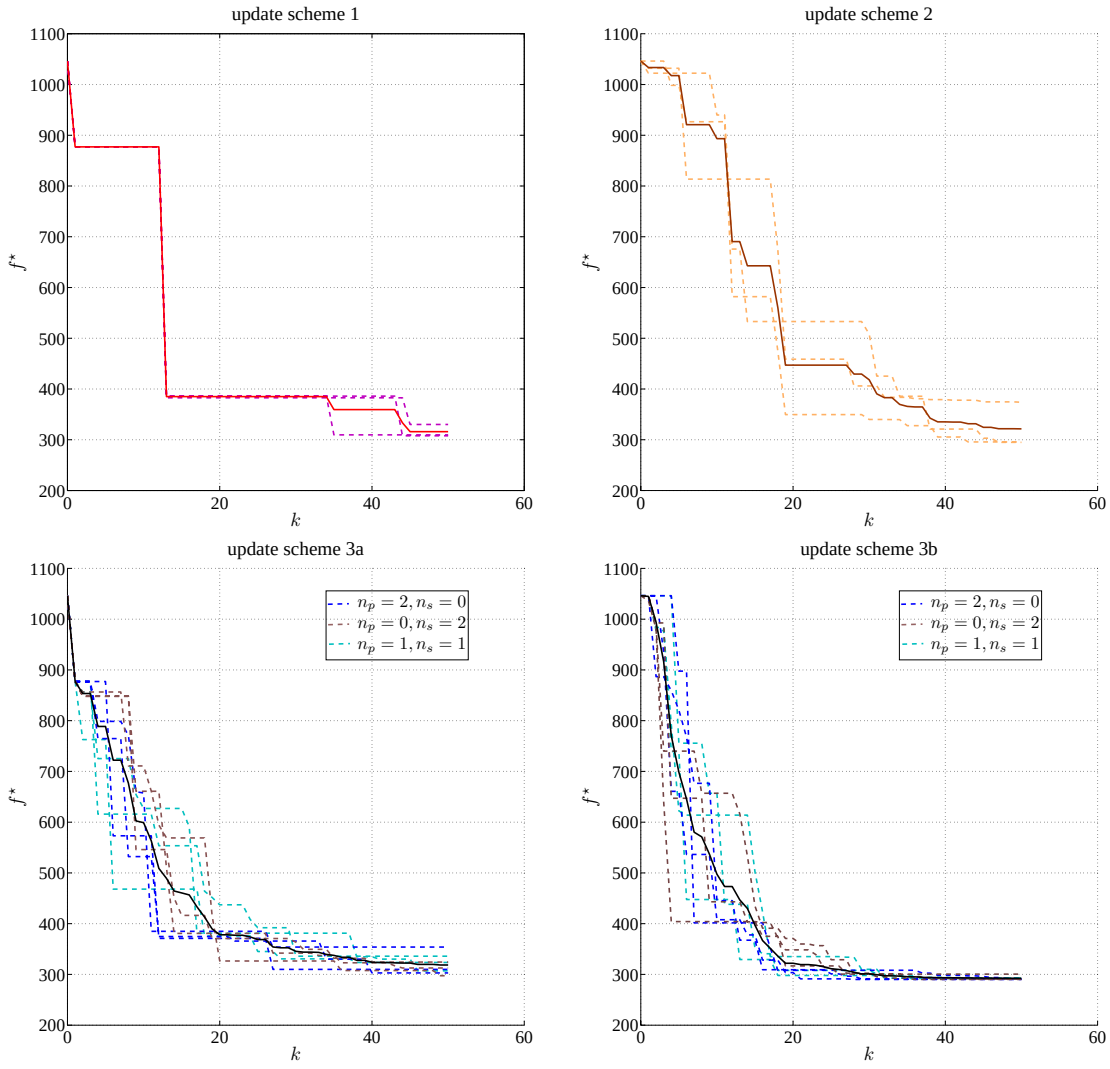

Figure 7.12: Example 7.2. Evolution of $f^\star$ using the update schemes $1 - 3$. The solid curves represent the mean of three runs for each case.

approximations with the same number of function evaluations (Appendix B). Both the SVM and Kriging approaches provide fairly accurate results, although the accuracy of the Kriging-based method is relatively higher at the same number of evaluations for this particular problem. However, the usefulness of the SVM-based approach is expected to be more evident when several other constraints are present. It should also be noted that in the case of multiple constraints it may not always be necessary to evaluate all the constraint responses to obtain the

classification information. If a sample is infeasible based on one of the constraints then the other constraint responses need not be evaluated. Thus, the proposed classification-based method can save several function evaluations if the constraint responses are obtained using different solvers. Another very important advantage of using the proposed SVM-based approach is that it can handle discontinuous and binary responses. Because of this, the proposed method is more general as it can be applied to a wider variety of problems. Also, being a relatively new approach, the SVM-based method still has some potential for improvement in terms of the efficiency.

Several formulations (or update schemes) of the constrained optimization problem are compared in this chapter. The final optimum solutions obtained with all the update schemes agree well with the actual known optimum. The number of function evaluations for the same error level are nearly the same for the different schemes. The function evaluations required by the schemes 3a and 3b are slightly higher for the problems that were studied. The difference in these schemes, compared to the schemes 1 and 2, lies in the evaluation of the additional primary and secondary samples. The evaluation of these samples is important as they refine the SVM approximation, which can accelerate the search for the optimum in some cases. As an additional advantage, the update schemes 3a and 3b also provide an accurate SVM boundary locally in the vicinity of the optimum, which is not the case with the other two update schemes (Figures 7.7-7.10). The locally refined SVM can, therefore, be used for calculating the probability of failure if uncertainties are present in the variables. Thus, the approach can be extended to reliability-based design optimization. It should also be noted that the primary and secondary samples can be evaluated in parallel with the EI maximization sample $\mathbf{x}_{EI}$. Therefore, if parallel processing capability is available (which is often the case in present days), these additional samples do not incur additional cost in terms of the wall time. For the same number of iterations, scheme 3b provides the highest accuracy. Especially, for Example 7.2, the accuracy is much higher using

the update scheme 3b. An interesting feature emerges from the study of the spatial distribution of the samples (Figures 7.7 to 7.10). Many of the samples are selected in the infeasible space when using the update schemes 1 and 3a. Such behavior stems from the issue of the relative scaling between the EI and the $P(+1|\mathbf{x})$. The results show the dominance of the EI over the $P(+1|\mathbf{x})$. Such issues are avoided in the constrained formulation of the optimization problem (schemes 2 and 3b).

7.5 Concluding remarks

7.5.1 Summary

A method for constrained global optimization using SVM constraints is presented in this chapter. The efficacy of the method is shown with two analytical examples with multiple constraints. Several formulations of the constrained optimization problem are proposed and compared in the results section. The use of SVM for constraint approximation allows the handling of discontinuous and binary constraint functions. Also, it simplifies multiple constraint problems, as a single SVM can be used even in the presence of several constraints. The optimization formulations require the calculation of "probability of feasibility". This is obtained using the modified PSVM model presented in Chapter 6. The PSVM model is also used in the selection of adaptive samples.

7.5.2 Future work

There are several avenues for future work in this research. First, the methodology will be extended to perform reliability-based design optimization. The local update used in the update scheme 3 provides an accurate SVM around the optimum and thus, will be useful for failure probability calculation and RBDO. In the future, the method will be applied to higher dimensional problems with multiple constraints. Another area of future study is the implementation of selective evaluation of the

objective function. The DOEs for evaluating the constraint and objective functions will be different in such a scheme.

CHAPTER 8

RELIABILITY ASSESSMENT USING RANDOM FIELDS

It is well known that the initial assumptions for the representation and quantification of uncertainties are of prime importance. These assumptions are as important as the process used to propagate uncertainties. In previous chapters, the SVM-based EDSD method was presented along with several adaptive sampling methods. The use of SVM for reliability assessment was demonstrated. However, the methods presented for the calculation of failure probabilities were based on the representation of uncertainties using random variables. For a problem with spatial variability (e.g., sheet metal thickness distribution), one should choose to describe the problem with random fields as they provide a more realistic representation than uncorrelated random variables (Missoum (2008)). A technique to incorporate random fields non-intrusively in probabilistic design is presented in this chapter. The approach is based on the extraction of the main features of a random field using a limited number of experimental observations (snapshots). An approximation of the random field is obtained using proper orthogonal decomposition (POD) (Liang,Y. C. et al. (2002); Bui-Thanh, T. et al. (2003)). For a given failure criterion, an explicit limit state function (LSF) in terms of the coefficients of the POD expansion is obtained using a support vector machine (SVM). Adaptive sampling (Chapter 4) is used to generate samples and update the approximated LSF. The coefficients of the orthogonal decomposition are considered as random variables with distributions determined from the snapshots. Based on these distributions and the explicit LSF, the approach allows for an efficient assessment of the probabilities of failure. In addition, the construction of explicit LSF has the advantage of handling discontinuous responses. Two test-problems are used to demonstrate the proposed methodology used for the calculation of probabilities of failure. The first example involves the linear buckling of an arch structure for which

the thickness is a random field. The second problem concerns the impact of a tube on a rigid wall. The planarity of the walls of the tube is considered as a random field.

8.1 Basic methodology

For the sake of clarity, this section summarizes the main steps of the approach, which are subsequently described in the following sections. The stages of the approach are (Figure 8.1):

- Collection of snapshots and construction of the covariance matrix.
- Selection of the main features of the random field.
- Expansion of the field on a reduced basis. Sampling of the coefficients using a uniform design of experiments (DOE).
- Construction of an explicit LSF using SVM in the space of coefficients.
- Refinement of the LSF using adaptive sampling.
- Fitting of the probability density functions (PDF) of the POD coefficients.
- Estimation of the probability of failure using Monte-Carlo simulations (MCS).

8.2 Random field characterization

8.2.1 Data collection and covariance matrix

The first step in the characterization of a random field is the collection of several observations of the random process output (e.g., a metal sheet after forming). The process generates a scalar random field $S(\overrightarrow{X})$, function of the position $\overrightarrow{X}$. M samples, outputs of this process, are obtained. On each sample, N measurements are

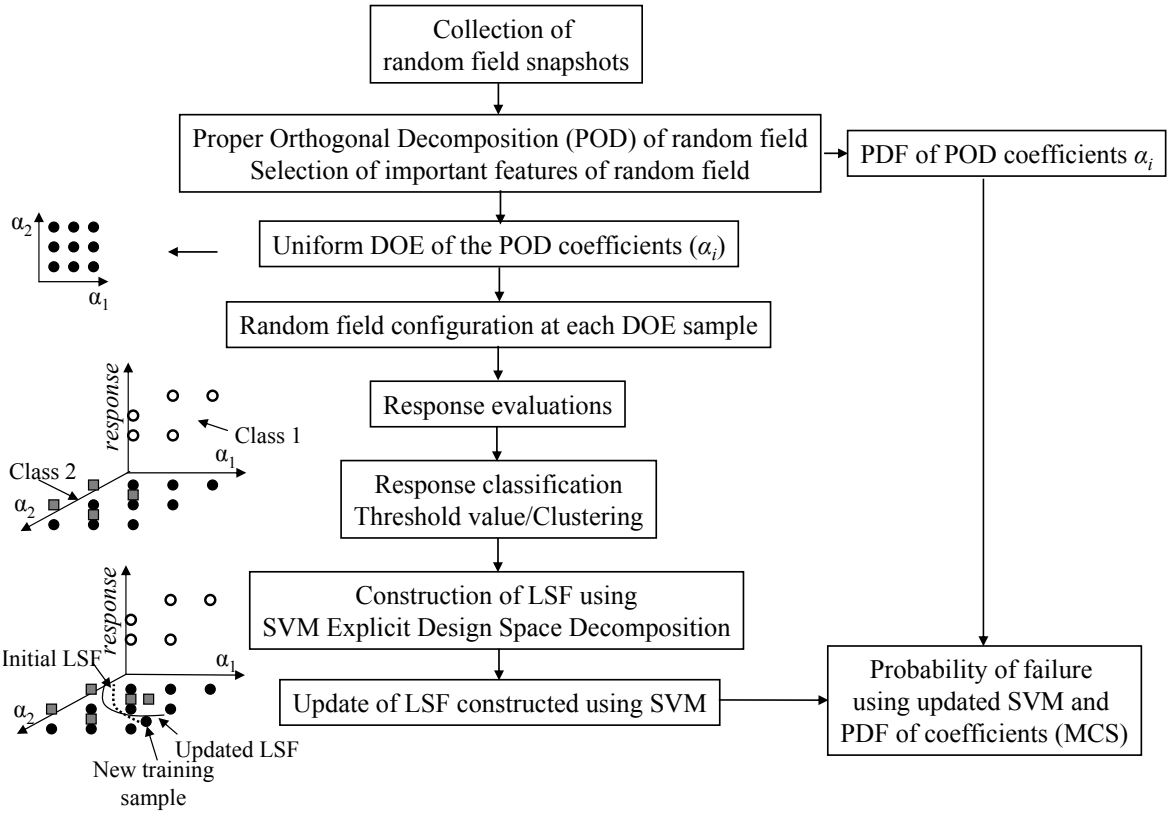

Figure 8.1: Summary of the proposed methodology for reliability assessment with random fields.

performed at distinct positions. An example of observations, referred to as snapshots, is provided in Figure 8.2. The snapshots can be condensed in the following matrix:

$$\mathbf{S} = \begin{pmatrix} S_{11} & \dots & S_{1M} \\ \vdots & \ddots & \vdots \\ S_{N1} & \dots & S_{NM} \end{pmatrix} \tag{8.1}$$

The general term S_{ij} is the i^{th} measurement for the j^{th} snapshot. A matrix $\mathbf{\Phi}$ is then defined, whose general term is:

$$\Phi_{ij} = S_{ij} - \bar{S}_i, \tag{8.2}$$

where $\bar{S}_i$ is the average snapshot vector at i^{th} measurement point, given by:

$$\bar{S}_i = \frac{1}{M} \sum_{j=1}^{M} S_{ij} \tag{8.3}$$

The covariance matrix $\mathbf{C}$, which is a square matrix of size N, is obtained as:

$$\mathbf{C} = \frac{1}{M} \mathbf{\Phi} \mathbf{\Phi}^T \tag{8.4}$$

Because the number of measurement locations N is usually high, the covariance matrix is large.

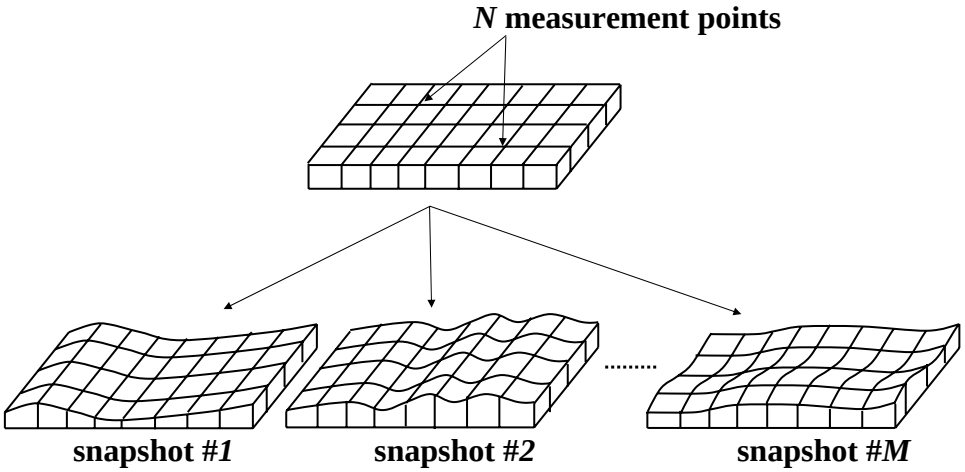

Figure 8.2: Example of M snapshots with N measurement points.

8.2.2 Feature extraction and selection POD

Proper orthogonal decomposition (POD) is used to decompose the random field on a basis made of the eigenvectors of the covariance matrix (Liang,Y. C. et al. (2002)). The random field is expressed in terms of the eigenvectors (i.e. features) as:

$$S_i = \bar{S} + \sum_{j=1}^{M} \alpha_{ij} V_j \tag{8.5}$$

S_i is the measurement vector of the i^{th} snapshot. V_j is the j^{th} eigenvector of the covariance matrix and α_{ij} are coefficients of the expansion. The eigenvectors being orthogonal, the general expression of the coefficients is obtained by projection:

$$\alpha_{ij} = \frac{\boldsymbol{\phi}_i \cdot V_j}{||V_j||^2} \tag{8.6}$$

where $\boldsymbol{\phi}_i = S_i - \bar{S}$ is the i^{th} centered snapshot. For normalized eigenvectors double vertical $||V_j|| = 1$, and the coefficients are:

$$\alpha_{ij} = \boldsymbol{\phi}_i \cdot V_j \tag{8.7}$$

If the size of covariance matrix is large then finding the eigenvectors might be difficult. If the number of snapshots M is lower than N, the eigenvectors can be obtained efficiently by defining a matrix $\mathbf{C}'$ as:

$$\mathbf{C}' = \frac{1}{M}\mathbf{\Phi}^T\mathbf{\Phi} \tag{8.8}$$

The eigenvectors of the covariance matrix $\mathbf{C}$ can then be found as (Sirovich (1987); Turk, M. and Pentland, A. (1991)):

$$V_i = \mathbf{\Phi}V_i' \tag{8.9}$$

where V_i' is an eigenvector of $\mathbf{C}'$. The dimensionality of the square matrix $\mathbf{C}'$ being M, the solution of the eigenvalue problem is computationally more efficient. Once the eigenvectors of the covariance matrix are obtained, the important features are selected by investigating the relative magnitude of the corresponding eigenvalues. The magnitude of an eigenvalue is proportionally related to the importance of the corresponding feature. Therefore by ranking the M eigenvalues, the MS most important features can be selected. This ranking is typically performed by inspecting the ratio ρ_i of the i^{th} eigenvalue ν_i to the sum of all eigenvalues (Bui-Thanh, T. et al. (2003)):

$$\rho_i = \frac{\nu_i}{\sum_{j=1}^{M}\nu_j} \tag{8.10}$$

The final expression of the expansion reads:

$$\widetilde{\boldsymbol{\phi}}_i = \sum_{j=1}^{MS}\alpha_{ij}V_j \tag{8.11}$$

where $\widetilde{\boldsymbol{\phi}}_i$ is the approximate reconstruction of the ith centered snapshot. It should be noted that the expansion containing less than M eigenvectors can only approximately reconstruct the original snapshots.

8.3 Probability density functions of the POD expansion coefficients

In order to perform probabilistic design, the PDFs of the coefficients need to be identified using data from the M snapshots. The coefficients are calculated for each snapshot using Equation 8.6. Thus, a distribution consisting of M discrete values is obtained for each of the MS coefficients. It is then possible to fit Weibull or Beta distributions to the data (Figure 8.5) that will be used subsequently for the calculation of the probability of failure.

8.4 Reliability assessment using explicit failure boundary approximation in a generalized space

The data from the snapshots allows the characterization of the random field, as well as the probability density functions of the POD expansion coefficients. Thus, the task of representing spatially varying uncertainties using a set of a few equivalent random variables is accomplished. In order to use this information for the calculation of failure probability, an explicit failure boundary is constructed using SVM. However, unlike previous chapters, the boundary is not constructed in a space consisting of physical entities. Instead, the SVM is constructed in a generalized space that consists of the POD coefficients.

For constructing the SVM, several instances of random fields (other than the snapshots) are created by using different linear combinations of the eigenvectors. The combinations are defined by selecting the POD coefficients using a DOE. It is important to note that the DOE is generated in the space of POD coefficients, and not in a space of physical variables (Figure 8.3). The bounds of the DOE are

defined by the maximum and minimum values of the coefficients obtained based on the snapshots. At this stage, the coefficients are sampled uniformly, and their PDFs are not yet taken into account. This is done to obtain information uniformly over the entire coefficient space. The DOE used for this study is generated by Latinized Centroidal Voronoi Tessellations (LCVT) (Romero, Vincente J. et al. (2006)).

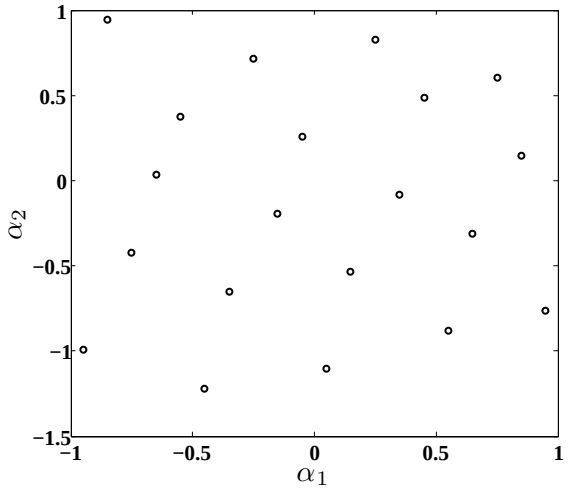

Figure 8.3: Example of LCVT DOE in the space of POD coefficients using 20 samples.

For each DOE sample $\boldsymbol{\alpha}_i$, an instance of the random field is created. Rest of the procedure is similar to the basic EDSD method presented in Chapter 4. The system response is estimated for the random field instance corresponding to each sample. The responses obtained for the DOE samples are then classified into failure or safe categories, based on a threshold response value or by using a clustering method such as K-means. The classification of responses into two distinct classes provides the information needed to construct the explicit failure boundary approximation using SVM:

$$s(\boldsymbol{\alpha}) = b + \sum_{i=1}^{N} \lambda_i y_i K(\boldsymbol{\alpha}_i, \boldsymbol{\alpha}) = 0 \tag{8.12}$$

Adaptive sampling is used to refine the initial SVM constructed with the uniform

DOE. The probability of failure is calculated using Monte-Carlo simulations in the space of POD coefficients:

$$P_f = \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} I_g(\boldsymbol{\alpha}_i) \tag{8.13}$$

The indicator function $I_g(\boldsymbol{\alpha})$ is:

$$I_g(\boldsymbol{\alpha}) = \begin{cases} 1 & s(\boldsymbol{\alpha}) \leq 0 \\ 0 & s(\boldsymbol{\alpha}) > 0 \end{cases} \tag{8.14}$$

8.5 Examples

Two examples are presented in this section to demonstrated the calculation of probability of failure in the presence of spatially varying uncertainties, using SVM-based EDSD. The first example consists of linear buckling of an arch structure. The second example consists of an tube impacting a rigid wall, for which the responses are discontinuous. The adaptive sampling scheme used for these results is a previous version of the global update scheme without secondary samples (Basudhar, A. and Missoum, S. (2008)).

8.5.1 Linear buckling of an arch structure

This section provides an example of the effect of a random field on the critical load factor of an arch structure. The structure is subjected to a unit pressure load on the top surface. The thickness of the arch should ideally be constant over the entire surface; however it may vary due to uncertainties in the manufacturing processes. These variations are represented, for this study, by an artificial analytical random field (as opposed to real experimental data). The arch has a radius of $R = 200$ mm, and it subtends an angle of $\theta_{max} = 60^\circ$ at the center (Figure 8.4). The width of the arch is $w = 600$ mm, and in the absence of any uncertainty it has a thickness $t = 3$ mm. The random field representing the deviation from the mean thickness is

assumed to have the following form.

$$h(\theta, z) = \frac{1}{4} \cos\left(\frac{K\pi\theta}{\theta_{max}}\right) \sin\left(\frac{L\pi z}{w}\right) \tag{8.15}$$

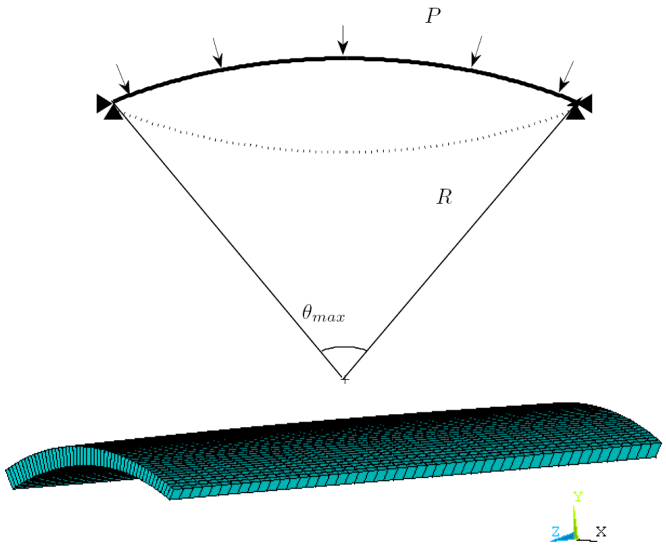

Figure 8.4: Geometry and loading of the arch structure. The bottom figure shows the spatial variation of thickness for one snapshot.

Following the creation of the snapshots, the important features are extracted based on the corresponding eigenvalue fractions. For example, if 200 snapshots are created, the four first ratios of eigenvalues as defined in Equation 8.10 are 0.7208, 0.1325, 0.0800, and 0.0634. The remaining ratios are clearly very small, and can be considered equal to zero. The analysis of the system is done by approximating the random field with three features.

Without the change in thickness introduced due to the random field, the critical load factor is 2.3086. When variations due to the random field are included, the critical load of the structure may increase or decrease. To quantify the uncertainty, the probability of having a critical load factor greater than 90% of the deterministic critical load factor is calculated. A critical load factor less than 90% of the deterministic critical load factor is considered as failure.

Random field approximation with three features

The first three features are used to approximate the random field. The corresponding eigenvalue fractions add up to 0.933. The random field is described as:

$$S(\alpha_1, \alpha_2, \alpha_3) = \alpha_1 V_1 + \alpha_2 V_2 + \alpha_3 V_3, \tag{8.16}$$

where α_1, α_2, and α_3 are the coefficients of the expansion. The minimum and maximum values of the coefficients, obtained from the snapshots, are given in Table 1. The PDFs of the coefficients are shown in Figure 8.5. Coefficients α_1 and α_2 are fitted to Beta distributions, while a Weibull distribution is used to fit α_3.

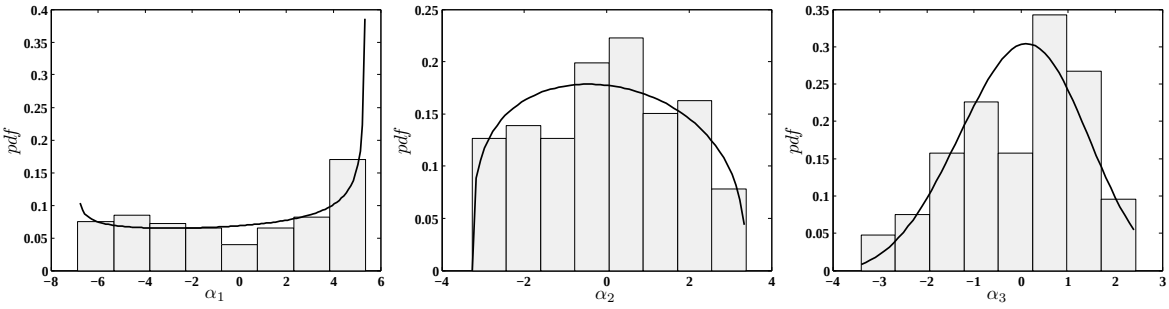

Figure 8.5: PDF for coefficients α_1, α_2, and α_3 corresponding to the first three features for the arch problem.

Once the random field is characterized, the coefficients α_1,α_2, and α_3 are sampled uniformly using 40 initial LCVT samples. The random fields corresponding to these configurations of the coefficients are reconstructed using Equation 8.16. The critical load factor for each configuration is then obtained using a finite element analysis (using ANSYS). The samples are then classified using the aforementioned failure criteria, and the classified configurations are the training samples for SVM to predict an initial LSF. It is then refined using the update algorithm, to construct the final LSF with 79 samples (Figure 8.6). The convergence of the SVM update algorithm is shown in Figure 8.7.

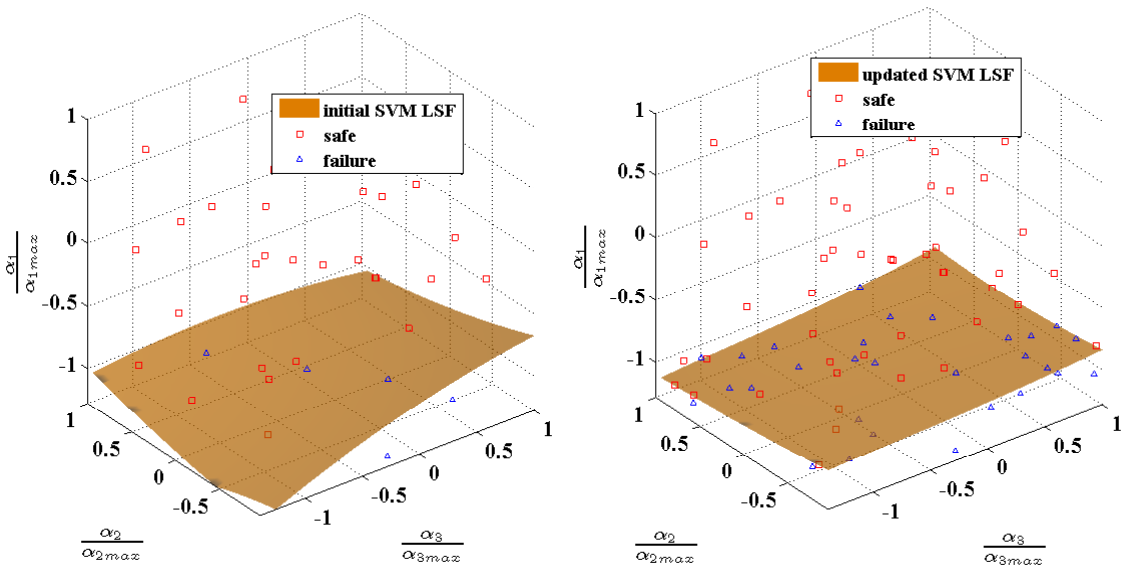

Figure 8.6: Initial (top) and final (bottom) SVM limit state function for the arch problem with three features. The brown surface is the limit state function separating failure (blue triangles) and safe (red squares) classes.

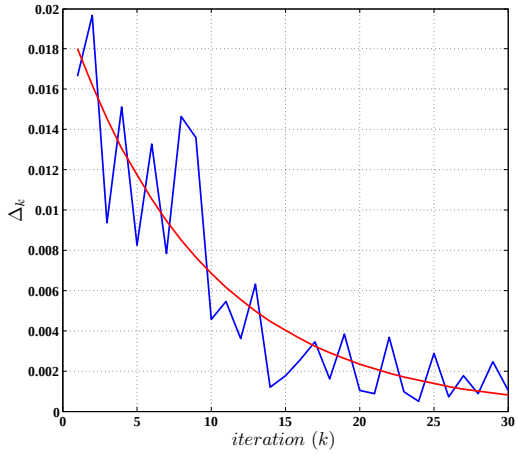

Figure 8.7: Convergence of the SVM limit state function update for the arch problem with spatially varying thickness.

After obtaining the explicit limit state function, MCS is carried out to calculate the probability of failure using the PDFs the coefficients (Figure 8.5). In order to validate the predicted value of P_f, MCS is carried out with 10^4 samples while varying

the values of K and L with their assumed distributions. Finite element analysis are carried out at each of these samples to find the actual probability of failure. The results are collected in Table 2.

Study of the influence of number of snapshots

In order to select the number of snapshots M, its influence on the eigenvalue fractions ρ_i are studied for the arch problem (Figure 8.8). It is seen that there is a significant change in the values of ρ_i initially. The values gradually stabilize around a constant value. A constant value suggests that adding snapshots does not provide much information to the random field. It is observed that the amplitude of the perturbations decreases gradually, and a value of $M = 200$ is selected.

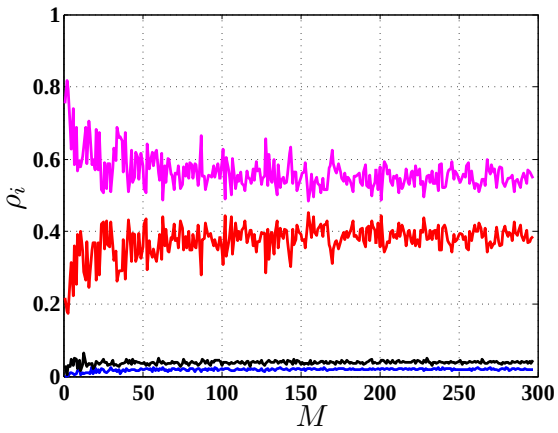

Figure 8.8: Influence of the number of snapshots (M), shown for the first four features of the arch problem. ρ_i ($i = 1, 2, 3, 4$) are the eigenvalue fractions corresponding to the first four features.

8.5.2 Tube impacting a rigid wall

In this section the application of the proposed methodology is shown on an impact problem. A tube of length $l = 1$ m (Figure 8.9) is made to impact a rigid wall with velocity 15 m/s, and the resulting behavior is analyzed. The cross-section of the

tube is a square with side $a = 7$ cm, and four masses of 25 kg each are attached to the four rear corners. The two bottom corners in the front of the tube are constrained in the transverse directions. The planarity of the walls of the tube are modified by a random field given by:

$$h(x_L, z_L) = \frac{A}{1000} \cos\left(\frac{3\pi x_L}{a}\right) \sin\left(\frac{L\pi z_L}{l}\right) \tag{8.17}$$

where x_L and z_L are the local coordinates at the four faces. x_L varies between $-\frac{a}{2}$ and $+\frac{a}{2}$, while z_L takes values between 0 and $-l$. A and L are uniformly distributed random variables with ranges $[0.25, 0.75]$ m and $[1, 2]$, respectively. The parameter A modifies the amplitude of the random field, while the frequency is modified by L. 200 snapshots of the random field are created, by varying A and L. The important features are then extracted using POD. The first three ratios (Equation 8.10) of eigenvalues are 0.6985, 0.2630, and 0.0383. Only the first two features are selected to characterize the random field.

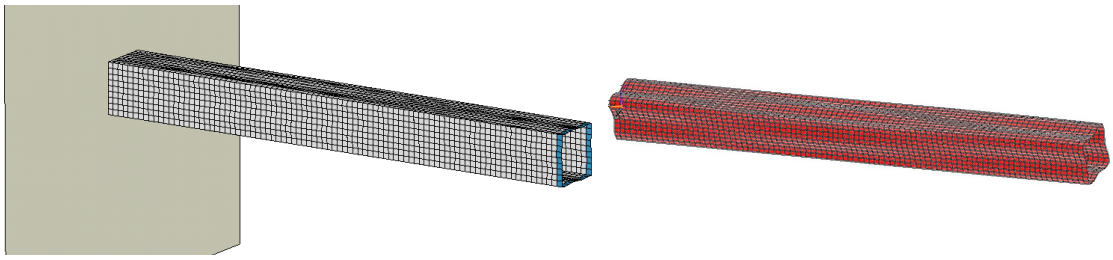

Figure 8.9: Tube impacting rigid wall. The right figure shows the effect of the random field.

The impact behavior of the tube falls into two main categories - crushing and global buckling (Figure 8.10). Due to the effect of the random field, the behavior can undergo sudden change from one state to the other. The discontinuous behavior of the tube is shown in Figure 8.11.

It is desired that the tube should display crushing, and there should not be any global buckling. In order to quantify the behavior, the maximum of the two

Figure 8.10: Crushing (left) and global buckling (right) of a tube subjected to impact.

absolute transverse displacements in x and y directions is studied. A low value of this quantity indicates crushing behavior, while a large value shows that global buckling has occurred. The probability of failure (global buckling), due to the effect of the random field, is studied with a thickness of 1.5 mm. The method can also be extended to carry out optimization by including the design variables as additional dimensions to the space (e.g., length or thickness).

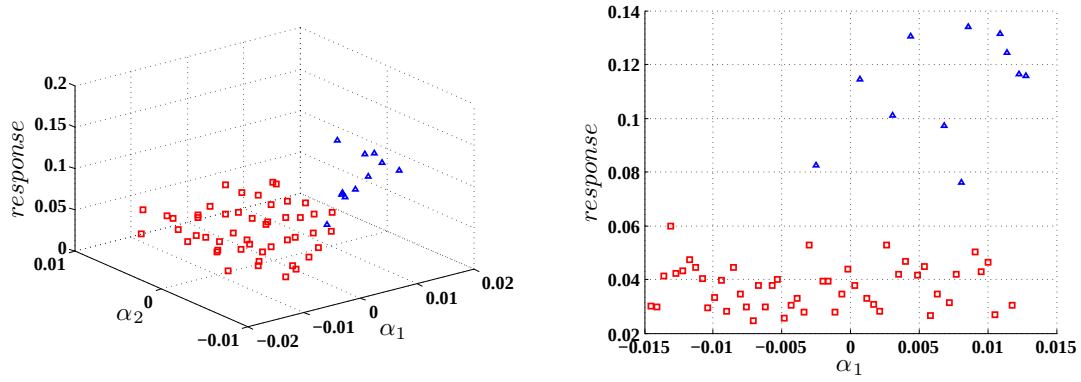

Figure 8.11: Discontinuous behavior of the tube response with respect to the coefficients of the expansion. The bottom figure shows a two-dimensional view of the top figure.

Following the random field characterization, the coefficients α_1, and α_2 are sampled uniformly using 60 LCVT samples, and the corresponding random field instances are constructed using Equation 8.11. The ranges of the two coefficients are $[-0.0148, 0.0130]$ and $[-0.0081, 0.0079]$. The analysis is done using ANSYS LS-

DYNA, to find the transverse displacements. The samples are then classified using K-means clustering, and the classified configurations are used as training samples for SVM to predict the explicit limit state function (Figure 8.12). After obtaining the explicit LSF, the probability of failure is calculated using the PDFs of the coefficients shown in Figure 8.13. The coefficient α_1 is fitted using a Weibull distribution, while α_2 is fitted with a Beta distribution. 106 MCS samples are used, and P_f is found as 0.1243.

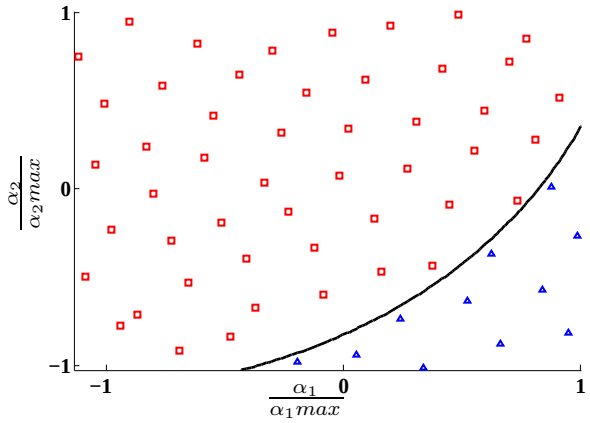

Figure 8.12: Tube impact problem. SVM failure domain boundary with two features. The black curve is the limit state function separating global buckling (blue triangles) and crushing (red squares).

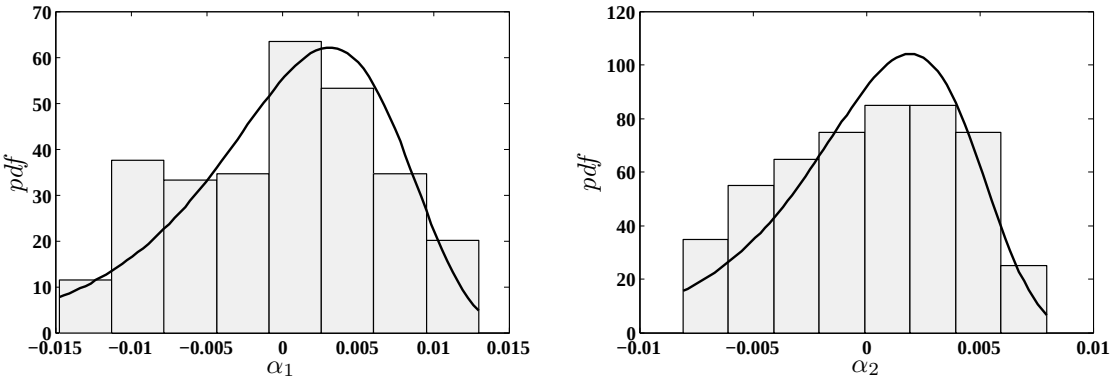

Figure 8.13: Tube impact problem. PDF for POD coefficients α_1, and α_2.

8.6 Concluding remarks

8.6.1 Summary

A technique for reliability assessment using random fields is presented in this chapter. A new sampling-based method is used for constructing various potential random field configurations. The method overcomes the need for assumption on the random field distribution by using snapshot data and Proper Orthogonal Decomposition. In addition the SVM-based method of constructing explicit LSFs enables one to address discontinuous system responses, which is successfully shown in the case of the tube impact problem.

8.6.2 Future work

One of the major limitations of the proposed method is that the correlation between POD coefficients is not considered. There is, however, no conceptual restriction on considering correlation in the proposed method. In the future, correlation will be considered using transformation methods, such as Nataf and Rosenblatt transformations. In future study, the method will also be extended for carrying out probabilistic optimization. This can be easily performed by adding the design variables as additional dimensions of the space while constructing the decision function. In the present study, analytical random fields have been used due to lack of experimental data; in the future, the methodology will be applied to data obtained from actual experiments.

CHAPTER 9

SUMMARY, CONCLUSION AND SCOPE OF FUTURE WORK

This chapter presents a summary of the research presented in this dissertation, and identifies certain avenues of future research generated from this work.

9.1 Summary and conclusion

Design of complex engineering systems often poses several challenges. The relation between design variables and the system responses are seldom known explicitly, and therefore, evaluation of several configurations or samples is required to gain an insight into the relationship. However, response evaluation at a single design configuration can be quite expensive, which limits the number of samples that can be evaluated. The decision boundaries (failure boundaries or optimization constraints) can be highly nonlinear, due to which several sample evaluations may be required to determine them accurately. This becomes even more challenging when multiple failure modes are present. Further challenges are faced in cases with discontinuous or binary responses that hamper most current techniques. As a result of these difficulties there is a need to develop newer methods for optimization or reliability assessment that can address all the issues together, which was the motivation for this research.

The main contribution of this work is the development of a new classification-based methodology that can be used for the approximation of failure domain boundaries and the zero-level contours of optimization constraints. The proposed approach referred to as explicit design space decomposition is a major shift from existing methods. In most current methods, the values of responses are important.

However, in the proposed method, only classification information of the responses is required for constructing the decision boundaries. As mentioned in Chapters 2 and 4, the classification-based approach has major advantages when the responses are discontinuous or binary.

A class of machine learning techniques referred to as support vector machines (Chapter 3) is used to construct the explicit decision boundaries. The boundaries constructed with SVM can be highly nonlinear. Also, a single SVM can be used to represent several failure modes, which is an advantage. Thus, the proposed SVM-based EDSD approach presents a natural way to handle problems with discontinuous and binary responses, as well as multiple failure modes.

In order to address the issue of limiting the computation cost, several adaptive sampling techniques have been developed in this research. The basic components of adaptive sampling were presented in Chapter 4. The basic EDSD method and a global update scheme to refine the SVMs was presented in this chapter. However, it may not be necessary to refine the decision boundaries over the entire space. Instead, identifying regions of importance can save a lot of samples. For this reason, a local update scheme for RBDO was presented in Chapter 5. An adaptive sampling scheme specifically designed to calculate probabilities of failure was also presented in this chapter. This update scheme was also used as part of the RBDO algorithm.

Because the construction of SVMs is based on a design of experiments, in general it has an error associated with the approximation of decision boundaries. This issue is addressed in Chapter 6, in which a modified probabilistic SVM model is presented that provides a measure of confidence on the SVM prediction. More specifically, it can be used to obtain the probability of misclassification by SVM at any sample. This information is used to provide an MCS-based probability of failure that is relatively conservative compared to the one using a deterministic SVM.

In Chapter 7, the modified PSVM model is also used in the development of a constrained efficient global optimization (EGO) method. The adaptive sampling schemes presented in Chapter 4 and 5 were focused on the accurate approximation of failure domain boundaries and optimization constraints. However, in general the objective function may also be expensive and require approximation. This issue is addressed in Chapter 7. The objective function is approximated using Kriging, whereas the zero-level constraint contours are approximated using SVM. The constrained EGO formulation requires the calculation of probability of feasibility, which is provided by the PSVM.

Finally, in Chapter 8, the SVM-based EDSD method is extended to the calculation of probability of failure in the presence of spatially varying uncertainties, represented with random fields. For this purpose, the random field is represented in terms of a few random variables, which are the coefficients of proper orthogonal decomposition (POD) expansion of the field. The SVM boundaries are then constructed in the space of POD coefficients. Probability density functions of the coefficients are determined from the observations.

Based on the studies made and reported in the different chapters, the following generalized conclusions are made:

- The proposed SVM-based EDSD method can provide a solution to various difficulties faced in optimization and reliability assessment. It handles problems with discontinuous and binary responses, and multiple failure modes. Application to discontinuous responses was demonstrated through the nonlinear arch buckling problem in Chapter 4, as well as the tube impact problem in Chapter 8. Application to multiple failure modes was demonstrated through the tolerance optimization example in Chapter 4, as well as through analytical test examples in Chapters 5 and 7.

- The adaptive sampling schemes in Chapters 4, 5 and 7 can accurately approximate highly nonlinear decision boundaries. Accuracy of the adaptive schemes was validated through several analytical test examples. The global update scheme has been tested up to seven dimensions.

- Identification of important regions in the space allows one to construct an accurate SVM in those regions, without wasting samples in other unimportant regions. This was demonstrated in Chapters 5 and 7. The examples showed the refinement of SVM boundaries in selected regions.

- Although there may be an error associated with an SVM, in general, it is possible to provide an error margin using a PSVM. PSVM can be used to provide a relatively conservative probability of failure compared to the one using a deterministic SVM. Unlike the traditional safety factor, which is usually constant, the measure of conservativeness referred to as probability factor depends on the amount of available information.

- It is possible to use SVM constraints within an EGO framework for performing optimization in problems with both expensive objective function and constraint functions. Because the constraint handling is based on SVM classification, it allows EGO to be performed for discontinuous and binary responses. Also it simplifies multiple constraint problems, as all the constraints can be approximated with a single SVM.

- In Chapter 8, it was shown that the SVM-based method is not limited to reliability assessment for spatially invariant uncertainties. It can also be used for reliability assessment with random fields.

9.2 Scope of future Work

The methods presented in this dissertation provide the first steps for classification-based reliability assessment and optimization. There are several improvements that

can be performed in the future research. These are discussed in this section.

9.2.1 Improvement of adaptive sampling schemes for refining SVMs

Several improvements are possible in the adaptive sampling process. One of the obvious extensions would be to use multifidelity models for the update. With complex engineering systems, very often different models with various levels of fidelity are available. For example, actual experimental testing or a finely meshed nonlinear finite element model can be examples of high fidelity. Lower fidelity models can obtained using different methods, such as using a coarse mesh, linearity assumptions, response approximations etc. Because a low fidelity model is cheaper to evaluate, several samples can be evaluated through it to get an initial idea. Selected samples can then be evaluated through the high fidelity model to update the approximated decision boundaries. If the low fidelity model is a response approximation, such as Kriging metamodel, then it is also possible to combine the sampling criteria for the approximation and classification approaches.

Apart from the use of multifidelity models, other improvements may also be possible in the sampling criteria. For example, the use of PSVMs for adaptive sampling needs to be explored in more detail. The threshold δ_{pm} in Equation B.6 can be be used to control how far or close to the current SVM boundary the adaptive sample will be selected. The threshold may be varied as the algorithm progresses. A study needs to be performed to assess the effect of the parameter.

9.2.2 Improvement of sampling criteria for constrained EGO

The constrained EGO formulation in Chapter 7 was based on the basic EI sampling criterion to asses the improvement of objective function. The focus of the chapter was to introduce a new method of handling constraints using PSVMs. However,

several criterion other than the EI can also be found in the literature (Sasena, M.J. (2002); Forrester, A.I.J. et al. (2008)). These criteria may be included in the algorithm to increase its efficiency.

9.2.3 Improvement of the method for providing error margin for SVM

In Chapter 6, a modified PSVM model was presented. The basic aim of including the information about spatial distribution of samples was achieved. While the comparison of errors for analytical examples shows significant improvement over previous models, further research may be performed to explore modifications of the model for achieving smoother variation of the conditional probability $P(+1|\mathbf{x})$ over the space. PSVM was also used to provide a relatively conservative measure of the probability of failure. A region Ω_{misc} for considering the probability of misclassification of Monte Carlo samples was identified for this purpose. This region was, however, defined based on the distance $d_+(\mathbf{x})$ and $d_+(\mathbf{x})$, which were calculated based on a single nearest neighbor. Use of K nearest neighbors or gradient information of SVMs may be useful to have a smoother region.

9.2.4 Consideration of correlation between variables

This dissertation was devoted to the development of the basic SVM-based EDSD method and adaptive sampling schemes to improve its efficiency. A very important issue has not been addressed, which is the consideration of correlation between variables. However, this does not limit the scope of application of the developed methods. Transformation methods exist for converting correlated variables to equivalent uncorrelated variables. Therefore, EDSD can be performed in the space of transformed variables. Initial work for calculating probabilities of failure with correlated variables has already been performed in Jiang, P. et al. (2011). Nataf transformation was used in this work to obtain equivalent standard normal variables. Copulas

other than the Gaussian one may be used in the future.

APPENDIX A

ADDITIONAL DETAILS OF UPDATE SCHEMES FOR PROBABILITY OF FAILURE CALCULATION AND RBDO

A.1 Determining whether a primary sample $\mathbf{x}_{mm}$ will be evaluated in probability of failure update

The steps to check whether to evaluate a primary sample $\mathbf{x}_{mm}$ in step 1 of the failure probability update (Chapter 5) are presented in Algorithm A.1. The maximum possible change in probability due to this sample is calculated by considering the two cases with class label $+1$ or -1. If the maximum change is less than certain threshold δ_2, this region is considered a candidate for SVM locking. Therefore, the possibility of selecting a sample to remove SVM locking (Section A.2), with $\mathbf{x}_{mm}$ as the center, is checked based on the same evaluation criterion. The locking removal sample is evaluated if it satisfies the threshold δ_2. However, even if the maximum probability change due to this sample is less than δ_2, it does not guarantee the absence of SVM locking. Also, if neither $\mathbf{x}_{mm}$ nor the locking removal sample are evaluated then it is likely that $\mathbf{x}_{mm}$ will be selected at the same position in the next iteration. Therefore, to avoid such scenario, if the locking removal sample does not satisfy the evaluation criterion then $\mathbf{x}_{mm}$ is evaluated irrespective of the change in probability.

Algorithm A.1

1: Estimate the new probabilities of failure $P_{f+}^{(k)}$ and $P_{f-}^{(k)}$ assuming that $\mathbf{x_{mm}}$ belongs to class +1 and -1 respectively.

2: **if** $max\left(\frac{\left|P_{f+}^{(k)}-P_f^{(k-1)}\right|}{P_f^{(k-1)}}, \frac{\left|P_{f-}^{(k)}-P_f^{(k-1)}\right|}{P_f^{(k-1)}}\right) > \delta_1$ **then**

3: Evaluate $\mathbf{x}_{mm}$.
4: **else**
5: Do not evaluate $\mathbf{x}_{mm}$.
6: Locate a sample $\mathbf{x}_{secondary}$ using the steps in Section A.2, with $\mathbf{x}_{mm}$ as the center. y_{mm} is assigned the sign of the closest sample to solve for the new sample.
7: Estimate the new probabilities of failure $P_{f+}^{(k)}$ and $P_{f-}^{(k)}$ assuming that $\mathbf{x}_{secondary}$ belongs to class +1 and -1 respectively.
8: **if** $max\left(\frac{\left|P_{f+}^{(k)}-P_f^{(k-1)}\right|}{P_f^{(k-1)}}, \frac{\left|P_{f-}^{(k)}-P_f^{(k-1)}\right|}{P_f^{(k-1)}}\right) > \delta_1$ **then**
9: Evaluate $\mathbf{x}_{secondary}$.
10: **else**
11: Evaluate $\mathbf{x}_{mm}$.
12: **end if**
13: **end if**

A.2 Refinement of SVM boundary $s_p^{(k)} = 0$

The construction of SVM boundary $s_p^{(k)} = 0$ requires information about the probability of failure at various configurations in the space. The probability of failure is calculated based on the SVM boundary $s_d^{(k)} = 0$ that approximates the limit state function. The steps for selecting samples for constructing $s_p^{(k)}$ are:

Step 1: In addition to the initial DOE, use all samples used for constructing $s_d^{(k)}$ in the approximation of $s_p^{(k)} = 0$ also. $P_f^{(k)}$ is evaluated at each of these samples with the SVM $s_d^{(k)} = 0$, using MCS.

Step 2: Define an update region for selecting additional samples. For phase 2, the update region is based on the probability density functions of the random variables. For phase 1, the update region is a hypersphere. Radius of the update region,

centered at $\mathbf{x}_p$, is:

$$R_u = max\left(\left|\left|\mathbf{x}_p^{(k)} - \mathbf{x}_p^{(k-1)}\right|\right|, \left|\left|\mathbf{x}_p^{(k)} - \mathbf{x}_{+1}\right|\right|, \left|\left|\mathbf{x}_p^{(k)} - \mathbf{x}_{-1}\right|\right|, 0.5\left(\frac{V}{N}\right)^{\frac{1}{d}}, \left|\left|\mathbf{x}_p^{(k)} - \mathbf{x}_d^{(k)}\right|\right|\right) \tag{A.1}$$

The radius is defined such that the update region encompasses the deterministic and probabilistic optima. It also requires the update region to consist of at least one sample from each class, which is a sufficient condition for the SVM $s_p^{(k)} = 0$ to pass through it. The critical value based on the volume V of the space and the number of samples ensures that the radius is not too small.

Step 3: Select primary and secondary samples within the update region, based on the SVM $s_p^{(k)} = 0$. Calculate $P_f^{(k)}$ at these samples, classify the samples and reconstruct SVM to update $s_p^{(k)} = 0$.

APPENDIX B

ADDITIONAL DETAILS OF SVM-BASED EFFICIENT GLOBAL OPTIMIZATION AND DERIVATION OF EXPECTED IMPROVEMENT

B.1 Derivation of expected improvement

Derivation of the expected improvement (EI), used in Chapter 7 is presented in this section. The improvement function $I(\mathbf{x})$ is defined as the difference between the objective function values at the current optimum and at $\mathbf{x}$:

$$\begin{aligned} I(\mathbf{x}) &= \max\left(0, f^\star - f\right) \\ &= \begin{cases} f^\star - f & f^\star \geq f \\ 0 & f^\star < f \end{cases} \end{aligned} \tag{B.1}$$

The expected improvement is calculated as the expectation of $I(\mathbf{x})$:

$$\begin{aligned} EI(\mathbf{x}) &= \int_{-\infty}^{f^\star} (f^\star - f)\hat{f}(\mathbf{x})df \\ &= \int_{-\infty}^{f^\star} (f^\star - \mu_f + \mu_f - f)\hat{f}(\mathbf{x})df \\ &= \int_{-\infty}^{f^\star} (f^\star - \mu_f)\hat{f}(\mathbf{x})df - \int_{-\infty}^{f^\star} (f - \mu_f)\hat{f}(\mathbf{x})df \\ &= \int_{-\infty}^{f^\star} (f^\star - \mu_f)\phi\left(\frac{f-\mu_f}{\sigma}\right)df - \int_{-\infty}^{f^\star} (f - \mu_f)\phi\left(\frac{f-\mu_f}{\sigma}\right)df \end{aligned} \tag{B.2}$$

Integrating by parts, the EI reduces to:

$$EI(\mathbf{x}) = (f^\star - \mu_f)\Phi\left(\frac{f-\mu_f}{\sigma}\right) - \sigma\int_{-\infty}^{f^\star} \frac{f-\mu_f}{\sigma}\phi\left(\frac{f-\mu_f}{\sigma}\right)df \tag{B.3}$$

For Gaussian distribution,

$$\phi(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}$$
$$\frac{d\phi}{dt} = -t\phi(t)$$
$$\phi = -\int_{-\infty}^{t} t\phi(t)dt \tag{B.4}$$

Substituting in Equation B.3,

$$EI(\mathbf{x}) = (f^\star - \mu_f)\Phi\left(\frac{f^\star - \mu_f}{\sigma_f}\right) + \sigma_f \phi\left(\frac{f^\star - \mu_f}{\sigma_f}\right) \tag{B.5}$$

B.2 Selection of primary samples based on PSVM

The procedure to select a primary sample $\mathbf{x}_p$ in the update scheme 3 of SVM-based EGO (Chapter 7) is presented in section. A primary sample is selected in a sparsely populated region with high probability of misclassification. However unlike the method in Chapter 4, in which the samples are selected on the SVM boundary, the regions of high misclassification probability are identified using the modified PSVM (Chapter 6). The optimization problem to locate $\mathbf{x}_p$ is:

$$\begin{aligned} \max \min_{\mathbf{x}} \quad & d(\mathbf{x}) \\ & \delta_{pm} - P_m(\mathbf{x}) \leq 0 \\ & ||\mathbf{x} - \mathbf{x}^\star|| - R_u \leq 0 \end{aligned} \tag{B.6}$$

where d is the distance to the closest training sample, P_m is the probability of misclassification. By default, the threshold δ_{pm} is equal to 0.5, but it is reduced until a feasible solution to the Equation B.6 is obtained. The misclassification probability P_m is calculated using the PSVM:

$$P_m(\mathbf{x}) = \begin{cases} 1 - P(+1|\mathbf{x}) & s(\mathbf{x}) \geq 0 \\ P(+1|\mathbf{x}) & s(\mathbf{x}) < 0 \end{cases} \tag{B.7}$$

The optimization problem in the Equation 4.3 is solved using a GA. The initial population is given by a local CVT with $100m$ samples, generated in a hypercube

circumscribing the update region centered at $\mathbf{x}^\star$.

B.3 Comparison of EGO results using classification and approximation based constraint handling

In this section, the results using the SVM-based EGO (Chapter 7) are compared to the ones using Kriging-based constraint approximation Sasena, M.J. (2002). The comparison is performed for the example in Section 7.3.1. Table B.1 lists the distances between the predicted and the actual optimum using the different methodologies at the same number of function evaluations. The mean values of $||\mathbf{x}^\star - \mathbf{x}^\star_{actual}||$ after 50 sequential evaluations are provided for each case. The solutions converging to the local optimum are omitted.

Several formulations are used for the Kriging-based approach. The first formulation uses the probability scaled EI (Schonlau, M. (1997)). The second one involves the constrained maximization of the EI. The constraints are based on the mean values of the Kriging models for the constraints. The final method is based on the expected violation of the constraints (Audet, C. et al. (2000)). For each of the methods, the constraints are treated in two ways for this multiple constraint problem. The first approach uses a Kriging model for each constraint (denoted by the subscript v) and the second one uses a single Kriging model for the maximum of all the constraints (denoted by the subscript s). It is seen that the results are fairly accurate using all the methods (SVM-based and Kriging-based). The accuracy of some of the Kriging-based methods is relatively higher for the same number of evaluations. However, the SVM-based method has the additional advantage of handling discontinuous and binary constraint functions. Also, the proposed SVM-based approach being in its early stages, there is ample scope for improvement. It should also be noted that the update scheme 3 also finds the optimum with similar number of function evaluations. However, because $\mathbf{x}_{EI}$, $\mathbf{x}_p$ and $\mathbf{x}_s$ are evaluated in parallel,

this only corresponds to 17 iterations of the update. In addition to the mean values listed in the Table B.1, the best solutions using the four SVM-based schemes are also noted. The errors are 4.1×10^{-3}, 7.0×10^{-4}, 2.4×10^{-2} and 5.5×10^{-3} using the update schemes 1, 2, 3a and 3b respectively.

Method	$\|\|\mathbf{x}^\star - \mathbf{x}^\star_{actual}\|\|$
Kriging $Probability_v$ Sasena, M.J. (2002)	2.2×10^{-4}
Kriging $Probability_s$ Sasena, M.J. (2002)	2.2×10^{-4}
Kriging $mean_v$ Sasena, M.J. (2002)	2.8×10^{-5}
Kriging $mean_s$ Sasena, M.J. (2002)	2.2×10^{-4}
Kriging EV_v Sasena, M.J. (2002)	1.8×10^{-1}
Kriging EV_s Sasena, M.J. (2002)	2.5×10^{-1}
SVM scheme 1	1.09×10^{-2}
SVM scheme 2	9.8×10^{-3}
SVM scheme 3a	3.45×10^{-2}
SVM scheme 3b	2.43×10^{-2}

Table B.1: Example 1. Distance between $x^\star$ and $x^\star_{actual}$ after evaluating 60 samples (10 initial). The SVM update schemes 3a and 3b evaluate 3 samples in parallel and the results after 17 iterations are given for these cases.

REFERENCES

Agarwal, H. (2004). *Reliability based design optimization: Formulations and Methodologies*. Ph.D. thesis, University of Notre Dame.

Ang, G., H. Alfredo, and W. Tang (1992). Optimal Importance-Sampling Density Estimator. *Journal of engineering mechanics*, **118**, p. 1146.

Ang, G., A. Ang, and W. Tang (1990). Kernel Method in Importance Sampling Density Estimation. *Structural Safety and Reliability*, pp. 1193–1200.

Arenbeck, H. (2007). *Reliability-Based Optimal Design and Tolerancing for Multibody Systems Using Explicit Design Space Decomposition*. Master's thesis, Department of Aerospace and Mechanical Engineering, University of Arizona, Tucson, AZ.

Arenbeck, H., Missoum, S., Basudhar, A., and Nikravesh, P.E. (2010). Reliability-Based Optimal Design and Tolerancing for Multibody Systems Using Explicit Design Space Decomposition. *Journal of Mechanical Design*, **132**(2), p. 021010. doi:10.1115/1.4000760.

Au, S. and J. Beck (1999). A new adaptive importance sampling scheme for reliability calculations. *Structural Safety*, **21**(2), pp. 135–158. ISSN 0167-4730.

Au, S. and J. Beck (2001). Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Engineering Mechanics*, **16**(4), pp. 263–277. ISSN 0266-8920.

Au, S., J. Ching, and J. Beck (2007). Application of subset simulation methods to reliability benchmark problems. *Structural Safety*, **29**(3), pp. 183–193. ISSN 0167-4730.

Audet, C., C. Audet, Jr., J. E. Dennis, and J. E. Dennis (2000). Analysis of Generalized Pattern Searches. *SIAM J. Optim*, **13**, pp. 889–903.

Audet, C., J.E. Dennis, Jr., Moore, D.W., Booker, A., and Frank, P.D. (2000). A surrogate based method for constrained optimization. In *8th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization. Paper number AIAA-2000-4891.*

Barranco-Cicilia, F., de Lima, E.C.P., and Sudati-Sagrilo, L.V. (2009). Structural Reliability Analysis of Limit State Functions With Multiple Design Points Using Evolutionary Strategies. *Ingenieria Investigacion y Tecnologia*, **10**(2). ISSN 1405-7743.

Basudhar, A. and Missoum, S. (2008). Adaptive explicit decision functions for probabilistic design and optimization using support vector machines. *Computers & Structures*, **86**(19–20), pp. 1904–1917.

Basudhar, A. and Missoum, S. (2009a). Local Update of Support Vector Machine Decision Boundaries. In 50^{th} *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. Palm Springs, California.

Basudhar, A. and Missoum, S. (2009b). A sampling-based approach for probabilistic design with random fields. *Computer Methods in Applied Mechanics and Engineering*, **198**(47-48), pp. 3647 – 3655.

Basudhar, A. and Missoum, S. (2010a). Constrained Efficient Global Optimization with Support Vector Machines (SVMs). In *13th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. Dallas, TX.

Basudhar, A. and Missoum, S. (2010b). An improved adaptive sampling scheme for the construction of explicit boundaries. *Structural and Multidisciplinary Optimization*, **42**(4), pp. 517–529. doi:10.1007/s00158-010-0511-0.

Basudhar, A. and Missoum, S. (2010c). Reliability Assessment using Probabilistic Support Vector Machines (PSVMs). In 51^{st} *AIAA/ASME/ASCE/AHS/ASC conference on Structures, Dynamics and Materials. Paper AIAA-2010-2764*. Orlando, Florida.

Basudhar, A., Missoum, S., and Harrison Sanchez, A. (2008). Limit state function identification using Support Vector Machines for discontinuous responses and disjoint failure domains. *Probabilistic Engineering Mechanics*, **23**(1), pp. 1–11.

Beachkofski, B.K. and Grandhi, R. (2002). Improved Distributed Hypercube Sampling. In *Proceedings of the* 43^{rd} *AIAA/ASME/ASCE/AHS/ASC conference on Structures, Dynamics and Materials. Paper AIAA-2002-1274*. Denver, Colorado, USA.

Berveiller, M., B. Sudret, and M. Lemaire (2006). Stochastic finite element: a non intrusive approach by regression. *Revue Europeenne de Mecanique Numerique*, **15**(1-2).

Bichon, B., J. McFarland, and S. Mahadevan (2010). Applying EGRA to Reliability Analysis Systems with Multiple Failure Modes.

Bichon, B.J., Eldred, M.S., Swiler, L.P., Mahadevan, S., and McFarland, J.M. (2007). Multimodal Reliability Assessment for Complex Engineering Applications using Efficient Global Optimization. In 48^{th} *AIAA/ASME/ASCE/AHS/ASC conference on Structures, Dynamics and Materials. Paper AIAA-2007-1946.* Honolulu, Hawaii.

Bichon, B.J., Mahadevan, S., and Eldred, M.S. (2009). Reliability-based Design Optimization using Efficient Global Reliability Assessment. In *Proceedings of the* 50^{th} *AIAA/ASME/ASCE/AHS/ASC conference on Structures, Dynamics and Materials. Paper AIAA-2009-2264.* Palm Springs, California.

Boggs, P. and J. Tolle (1995). Sequential quadratic programming. *Acta numerica*, **4**(-1), pp. 1–51. ISSN 1474-0508.

Box, G. and K. Wilson (1951). On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society. Series B (Methodological)*, **13**(1), pp. 1–45.

Breitung, K. (1984). Asymptotic approximations for multinormal integrals. *Journal of Engineering Mechanics*, **110**, p. 357.

Bui-Thanh, T., Damodaran, M., and Willcox, K. (2003). Proper orthogonal decomposition extensions for parametric applications in transonic aerodynamics. In 21^{st} *AIAA Applied Aerodynamics Conference, Orlando Florida*. Orlando, Florida.

Butler, A. N. (2001). Optimal and orthogonal latin hypercube designs for computer experiments. *Biometrika*, **88**(3), pp. 847–857.

Cawley, G. and N. Talbot (2003). Efficient leave-one-out cross-validation of kernel fisher discriminant classifiers. *Pattern Recognition*, **36**(11), pp. 2585 – 2592.

Celorrio, L., de Pison, E.M., and Bao, C. (2009). Efficiency analysis of Reliability Based Design Optimization approaches for dependent input random variables. In 1^{st} *ISSMO Internet Conference on Reliability-based Structural Optimization.*

Choi, K. and B. Youn (2001). Hybrid analysis method for reliability-based design optimization. In *27th ASME Design Automation Conference.*

Cornell, C. (1969). A Probability-Based Structural Code. In *ACI Journal Proceedings*, volume 66. ACI.

Der Kiureghian, A. and T. Dakessian (1998). Multiple design points in first and second-order reliability. *Structural Safety*, **20**(1), pp. 37–49.

Downing, D., R. Gardner, and F. Hoffman (1985). An examination of response-surface methodologies for uncertainty analysis in assessment models. *Technometrics*, **27**(2), pp. 151–163.

Du, X. and W. Chen (2004). Sequential optimization and reliability assessment method for efficient probabilistic design. *ASME Journal of Mechanical Design*, pp. 225–233. ISSN 1050-0472.

Elishakoff, I. (2004). *Safety factors and reliability: friends or foes?* Kluwer Academic Pub. ISBN 1402017790.

Fang, K., R. Li, and A. Sudjianto (2006). *Design and modeling for computer experiments*. CRC Press. ISBN 1584885467.

Finkel, D. (2003). DIRECT optimization algorithm user guide. *Center for Research in Scientific Computation, North Carolina State University*.

Forrester, A.I.J., Sobester, A., and Keane, A.J. (2008). *Engineering design via surrogate modelling: a practical guide*. Wiley. ISBN 0470060689.

Ghanem, R. and P. Spanos (2003). *Stochastic finite elements: a spectral approach*. Dover Pubns. ISBN 0486428184.

Ghiocel, D. and R. Ghanem (2002). Stochastic finite-element analysis of seismic soil-structure interaction. *Journal of engineering mechanics*, **128**(1), pp. 66–77.

Gilks, W., S. Richardson, and D. Spiegelhalter (1996). *Markov chain Monte Carlo in practice*. Chapman & Hall/CRC. ISBN 0412055511.

Givens, G. and A. Raftery (1996). Local Adaptive Importance Sampling for Multivariate Densities with Strong Nonlinear Relationships. *Journal of the American Statistical Association*, **91**(433).

Goldberg, D. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-wesley. ISBN 0201157675.

Gunn, S.R. (1998). Support vector machines for classification and regression. Technical Report ISIS-1-98, Department of Electronics and Computer Science, University of Southampton.

Gupta, S. and C. Manohar (2004). An improved response surface method for the determination of failure probability and importance measures. *Structural Safety*, **26**(2), pp. 123–139. ISSN 0167-4730.

Haldar, A. and Mahadevan, S. (2000). *Probability, Reliability, and Statistical Methods in Engineering Design*. Wiley and Sons, New-York.

Hamel, L. (2009). *Knowledge discovery with support vector machines*. Wiley and Sons. ISBN 0470503041.

Harbitz, A. (1986). An efficient sampling method for probability of failure calculation. *Structural safety*, **3**(2), pp. 109–115.

Hartigan, J.A. and Wong, M.A. (1979). A K-means clustering algorithm. *Applied Statistics*, **28**, pp. 100–108.

Hasofer, A. and N. Lind (1974). Exact and invariant second-moment code format. *Journal of the Engineering Mechanics Division*, **100**(1), pp. 111–121.

Hohenbichler, M., S. Gollwitzer, W. Kruse, and R. Rackwitz (1987). New light on first-and second-order reliability methods. *Structural Safety*, **4**(4), pp. 267–284.

Hohenbichler, M. and R. Rackwitz (1983). First-order concepts in system reliability. *Structural Safety*, **1**(3), pp. 177–188.

Huang, S., S. Mahadevan, and R. Rebba (2007). Collocation-based stochastic finite element analysis for random field problems. *Probabilistic engineering mechanics*, **22**(2), pp. 194–205.

Hurtado, Jorge E. (2004). An examination of methods for approximating implicit limit state functions from the viewpoint of statistical learning theory. *Structural Safety*, **26**, pp. 271–293.

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

Jones, D.R., Schonlau, M., and Welch, W.J. (1998). Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*, **13**(4), pp. 455–492.

Kiureghian, A. D. and P.-L. Liu (1986). Structural Reliability Under Incomplete Probability Information. *Journal of Engineering Mechanics*, **112**.

Kleijnen,J.P.C. (2008). Design of experiments: overview. In *Proceedings of the 40th Conference on Winter Simulation*, WSC '08, pp. 479–488. ISBN 978-1-4244-2708-6.

Kleijnen,J.P.C., Sanchez,S.M., and Lucas,T.W. (2005). State-of-the-Art Review: A Users Guide to the Brave New World of Designing Simulation Experiments. *INFORMS Journal on Computing*, **17**, pp. 263–289.

Layman, R., Missoum, S., and Geest, J.V. (2010). Simulation and probabilistic failure prediction of grafts for aortic aneurysm. *Engineering Computations*, **27**(1), pp. 84–105.

Lemaire, M., A. Chateauneuf, and J. Mitteau (2009). *Structural reliability*. Wiley Online Library. ISBN 1848210825.

Liang,Y. C., Lee.H. P., Lim, S. P., Lin, W.Z., Lee,K. H., and Wu, C.G. (2002). Proper orthogonal decomposition and its applications–part I: theory. *Journal of Sound and Vibration*, **252**(3), pp. 527 – 544.

Liefvendahl, M. and Stocki, R. (2006). A study on algorithms for optimization of Latin hypercubes. *Journal of Statistical Planning and Inference*, **136**, pp. 3231–3247.

Lin, H., C. Lin, and R. Weng (2007). A note on Platts probabilistic outputs for support vector machines. *Machine Learning*, **68**(3), pp. 267–276.

Lin, P.M., Leon, B.J., and Huang, T.C. (1976). A new algorithm for symbolic system reliability analysis. *IEEE Transactions on Reliability*, **25**(1), pp. 2–15.

Ma, N. (2001). Complete multinomial expansions. *Applied Mathematics and Computation*, **124**(3), pp. 365 – 370.

Martz, H. and R. Waller (1990). Bayesian reliability analysis of complex series/parallel systems of binomial subsystems and components. *Technometrics*, **32**(4), pp. 407–416. ISSN 0040-1706.

McDonald, M. and Mahadevan, S. (2008). Reliability-Based Optimization With Discrete and Continuous Decision and Random Variables. *Journal of Mechanical Design*, **130**(6), p. 061401.

Melchers, R. (1989). Importance sampling in structural systems. *Structural safety*, **6**(1), pp. 3–10.

Melchers, R.E. (1999). *Structural Reliability Analysis and Prediction*. John Wiley & Sons. ISBN 0471987719.

Metropolis, N. and S. Ulam (1949). The Monte Carlo Method. *Journal of the American Statistical Association*, **44**(247), pp. 335–341.

Missoum, S. (2008). Probabilistic optimal design in the presence of random fields. *Structural and Multidisciplinary Optimization*, **35**(6), pp. 523–530.

Missoum, S., S. Ben Chaabane, and B. Sudret (2004). Handling bifurcations in the optimal design of transient dynamic problems. In 45^{th} *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*. Palm Springs, CA.

Missoum, S., Ramu, P., and Haftka, R.T. (2007). A Convex Hull Approach for the Reliability-based Design of Nonlinear Transient Dynamic Problems. *Computer Methods in Applied Mechanics and Engineering*, **196**(29), pp. 2895–2906.

Montgomery, D.C. (2005). *Design and Analysis of Experiments*. Wiley and Sons. ISBN 0-471-48735-X.

Myers, R.H. and Montgomery, D.C. (2002). *Response Surface Methodology*. Wiley, 2^{nd} edition. ISBN 0471412554.

Noh, Y., Choi, K., and Du, L. (2009). Reliability-based design optimization of problems with correlated input variables using a Gaussian Copula. *Structural and Multidisciplinary Optimization*, **38**, pp. 1–16.

Picheny, V. (2009). *Improving accuracy and compensating for uncertainty in surrogate modeling*. Ph.D. thesis, University of Florida.

Platt, J.C. (1999). Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In *Advances in Large Margin Classifiers*, pp. 61–74. MIT Press.

Powell, M. (1987). Radial basis functions for multivariable interpolation: A review. In *Algorithms for approximation*, pp. 143–167. Clarendon Press. ISBN 0198536127.

Rais-Rohani, M. and M. Singh (2004). Comparison of global and local response surface techniques in reliability-based optimization of composite structures. *Structural and Multidisciplinary Optimization*, **26**(5), pp. 333–345.

Romero, Vincente J., Burkardt, John V., Gunzburger, Max D., and Peterson, Janet S. (2006). Comparison of Pure and "Latinized" Centroidal Voronoi Tesselation Against Various Other Statistical Sampling Methods. *Journal of Reliability Engineering and System Safety*, **91**(1266–1280).

Salas, P., Venkataraman, S., and Benson, D. (2011). Modeling Error Estimation in Response Prediction of Multilevel Composite Systems using Bayesian Networks. In 52^{nd} *AIAA/ASME/ASCE/AHS/ASC conference on Structures, Dynamics and Materials. Paper AIAA-2011-1906.* Denver, CO.

Sankararaman, S., Mahadevan, S., Bradford, S., and Peterson, L. (2011). Test Resource Allocation for Uncertainty Quantification of Multi-level and Coupled Systems. In 52^{nd} *AIAA/ASME/ASCE/AHS/ASC conference on Structures, Dynamics and Materials. Paper AIAA-2011-1904.* Denver, CO.

Sasena, M.J. (2002). *Flexibility and Efficiency Enhancements for Constrained Global Optimization with Kriging Approximations*. Ph.D. thesis, Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI.

Sasena, M.J., Papalambrose, P.Y., and Goovaerts, P. (2002). Exploration of metamodeling sampling criteria for constrained global optimization. *Engineering Optimization*, **34**, pp. 263–278.

Schonlau, M. (1997). *Computer experiments and global optimization*. Ph.D. thesis, Department of Statistics, University of Waterloo, Ontario, Canada.

Schueller, G. and R. Stix (1987). A critical appraisal of methods to determine failure probabilities. *Structural Safety*, **4**(4), pp. 293–309.

Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel Methods For Pattern Analysis*. Cambridge University Press.

Simpson, T. W., Toropov, V. V., Balabanov, V. O., and Viana, F. A. C. (2008). Design and Analysis of Computer Experiments in Multidisciplinary Design Optimization: A Review of How Far We Have Come - or Not. In *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. Reston, VA.

Sirovich, L. (1987). Turbulence and the dynamics of coherent structures, part 1: coherent structures. *Quarterly of Applied Mathematics*, **45**(3), pp. 561–571.

Stefanou, G. (2009). The stochastic finite element method: past, present and future. *Computer Methods in Applied Mechanics and Engineering*, **198**(9-12), pp. 1031–1051.

Sudret, B. and A. Der Kiureghian (2000). *Stochastic finite element methods and reliability: a state-of-the-art report*. Dept. of Civil and Environmental Engineering, University of California.

Turk, M. and Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, **3**(1), pp. 71–86.

Vanderplaats, G. (1984). *Numerical optimization techniques for engineering design: with applications*. ISBN 0070669643.

Vapnik, V.N. (1998). *Statistical Learning Theory*. John Wiley & Sons.

Wahba, G. (1999). Support vector machines, reproducing kernel hilbert spaces and the randomized GACV. In *Advances in Kernel Methods - Support Vector Learning*. MIT Press, Cambridge, MA, USA.

Wang, G. and S. Shan (2007). Review of Metamodeling Techniques in Support of Engineering Design Optimization. *Journal of Mechanical Design*, **129**(4), pp. 370–380.

Wang, L. and R. Grandhi (1994). Efficient safety index calculation for structural reliability analysis. *Computers & Structures*, **52**(1), pp. 103–111.

Wang, L. and R. Grandhi (1995). Intervening variables and constraint approximations in safety index and failure probability calculations. *Structural and Multidisciplinary Optimization*, **10**(1), pp. 2–8.

Wang, G.G., Wang, L., and Shan, S. (2005). Reliability Assessment Using Discriminative Sampling and Metamodeling. *SAE Transactions, Journal of Passenger Cars - Mechanical Systems*, **114**, pp. 291–300.

Weise, T. (2009). *Global Optimization Algorithms - Theory and Application*. Self-Published, second edition. Online available at http://www.it-weise.de/.

Wu, Y., H. Millwater, and T. Cruse (1990). Advance Probabilistic Structural Analysis Method for Implicit Performance Functions. *AIAA Journal*, **28**(9), pp. 1663–1669.

Yamazaki, F. (1988). Neumann expansion for stochastic finite element analysis. *Journal of Engineering Mechanics*, **114**, p. 1335.

Youn, B. (2007). Adaptive-loop method for non-deterministic design optimization. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, **221**(2), pp. 107–116.

Youn, B. and Xi, Z. (2009). Reliability-based robust design optimization using the eigenvector dimension reduction (EDR) method. *Structural and Multidisciplinary Optimization*, **37**, pp. 475–492.

Youn, B.D. (2005). Enriched Performance Measure Approach for Reliability-Based Design Optimization. *AIAA Journal*, **43**(4), pp. 874–884.

Youn, B.D., Choi, K.K., and Du, L. (2005). Adaptive probability analysis using an enhanced hybrid mean value method. *Structural and Multidisciplinary Optimization*, **29**(2), pp. 134–148.

Yu, X., Choi, K. K., and Chang, K.H. (1997). A mixed design approach for probabilistic structural durability. *Structural and Multidisciplinary Optimization*, **14**(2), pp. 81–90.