AN EFFICIENT STRATEGY FOR RELIABILITY-BASED DESIGN OPTIMIZATION OF NONLINEAR SYSTEMS

by

Marcus H. Rademacher

A THESIS

Submitted to

Michigan State University in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Mechanical Engineering

ABSTRACT

AN EFFICIENT STRATEGY FOR RELIABILITY-BASED DESIGN OPTIMIZATION OF NONLINEAR SYSTEMS

by

Marcus H. Rademacher

Engineers are constantly challenged with the task of creating better products. Design optimization is one of several modern tools that have been developed to meet this challenge. A common issue in optimization is that of finding a reliable design. A design may nominally pass all requirements, but when it is put into service stochastic variations in performance may cause the design to fail more often than is acceptable. The preferred solution is to perform reliability-based design optimization (RBDO), which accounts for uncertainty. In order to evaluate the reliability of a given design, considerable computational cost is necessary.

The work presented in this thesis outlines a new strategy for performing RBDO where local response surfaces are used to reduce the computational burden of predicting the reliability. This strategy also takes advantage of the fact that the deterministic design optimization (DDO) and RBDO solutions are often spatially nearby each other, resulting in the opportunity to use inexpensive and well-worn DDO techniques to get a head start on finding the RBDO solution. The DDO study also provides ample data that can be used to fit response surfaces during the RBDO stage, without the need for additional evaluations. This new strategy is applied to several problems, including simple analytical functions and real engineering problems.

To my wonderful wife Victoria, for her steadfast support and encouragement.

ACKNOWLEDGEMENTS

I would like to acknowledge and thank my advisor Dr. Ron Averill for the support and guidance he has provided. I would also like to thank the other members of my thesis committee, Dr. Erik Goodman and Dr. Alejandro Diaz. Their support is much appreciated.

I would also like to thank Red Cedar Technology for their support, particularly Mr. Ranny Sidhu. He assisted considerably in overcoming research road blocks and in developing the strategy studied in this thesis.

TABLE OF CONTENTS

List of Tablesvii
List of Figures
1 Introduction 1 1.1 Motivation 1 1.2 Literature Review 7 1.3 Obstacles to Using RBDO 14 1.4 The Current Study 17
2 Statistical Foundation192.1 Random Variables192.2 Probability Density Function192.3 Cumulative Distribution Function222.4 Mean232.5 Variance and Standard Deviation242.6 Probability Distributions252.6.1 Gaussian Distribution262.6.2 Log-normal Distribution30
3 Measuring Reliability 32 3.1 Probability of Failure and the Reliability Index 32 3.2 Sampling Methods 33 3.2.1 Monte Carlo Sampling 34 3.2.2 Latin Hypercube Sampling 36 3.3 Most Probable Point Methods 37 3.3.1 The Hasofer-Lind Transformation and the MPP 37 3.3.2 Mean Value First-Order Second Moment Method 40 3.3.3 Hasofer-Lind Algorithm 41 3.3.4 Approximating Non-normal Distributions with Equivalent Normal Distributions 42 3.3.5 First-Order Reliability Method 45 3.4 Modern Methods 45 3.4.1 Performance Measure Approach 45 3.4.2 Sequential Optimization and Reliability Assessment 47
4 The Proposed Approach

 4.3 Intermediate Conclusions about RBDO	52 53 53 53 57 57 58 59 61 62 64
 5 Problem Descriptions and Results	
 6 Summary and Conclusions	

eferences

LIST OF TABLES

Table 3.1 Table of CI bounds for MCS where 1,000,000 simulations are performed36
Table 4.1. Genetic algorithm parameters used. 62
Table 4.2. Explanation of variables used in equation (4.1)
Table 5.1. Breakdown of the number of cycles performed in each stage of the studies performed
Table 5.2. Baseline design and applied variation for the cantilevered beam problem. 118
Table 5.3. Baseline design and applied variation. 154
Table 5.4. Breakdown of the number of cycles performed in each stage of the studies performed

LIST OF FIGURES

Figure 1.1. Example of a two variable problem which illustrates the concept of active constraints
Figure 1.2. Example of a two variable problem with variation
Figure 1.3. Flowchart illustrating the double-loop required to perform reliability-based design optimization
Figure 2.1 A histogram of sampled data with 32 bins. Since box heights are normalized in this case, it is also an approximation of the probability density distribution. Plot generated in Minitab 15. ^[26]
Figure 2.2 Histogram from Figure 2.1 with probability density function fit to the data overlaid. Plot generated in Minitab 15
Figure 2.3 Cumulative Distribution Function for data used in Figure 2.1 and Figure 2.2. Plot generated in Minitab 15
Figure 2.4 Normal (or Gaussian) distribution for several values of μ_{X,σ_X} . [28] For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this thesis
Figure 2.5. Log-normal distributions for several values of μ , σ . [29]
Figure 3.1 Hasofer-Lind transformation and location of MPP in U-space. ^[36]
Figure 3.2. An illustration of how the RF algorithm approximates a non-normal distribution with a normal one. ^[36]
Figure 4.1. Process for performing reliability analysis on a design. The "Fit RS" process is described in Figure 4.2
Figure 4.2. Process for fitting a response surface
Figure 4.3. Illustration of the parameters α and E and how they define the time when the switch from DDO to RBDO occurs

Figure 5.4. Absolute difference of P_f predicted by the linear response surface and the actual P_f for the best design found. The height of the colored bar is the average of 25 runs. The whiskers represent the range of the actual values. Note that in most cases the average is very close to the minimum, and typically only a single run had very high error. The "actual Pf" value compared against is the probability of failure as calculated using the same RA method, but with actual evaluations instead of a response surface.

Firmer F.F. Opens data as an assisted in Firmer F.4 with autoria and every hore (and

Figure 5.5. Same data as presented in Figure 5.4, without min and max bars (only average value is plotted). Recall that the reliability constraint limit was set to 0.0027...73

Figure 5.6. History plot for the simple function problem, objective F from 5 runs, α =0.25, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.......75

Figure 5.7. History plot for the simple function problem, constraint G from 5 runs, α =0.25, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage. 75

Figure 5.8. History plot for the simple function problem, objective F from 5 runs, α =0.25, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage......76

Figure 5.9. History plot for the simple function problem, constraint G from 5 runs, α =0.25, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage. 76

Figure 5.10. History plot for the simple function problem, objective F from 5 runs, α =0.50, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage. 77

Figure 5.11. History plot for the simple function problem, constraint G from 5 runs, α =0.50, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage. 77
Figure 5.12. History plot for the simple function problem, objective F from 5 runs, α =0.50, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage. 78
Figure 5.13. History plot for the simple function problem, constraint G from 5 runs, α =0.50, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage. 78
Figure 5.14. History plot for the simple function problem, objective F from 5 runs, α =0.75, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage
Figure 5.15. History plot for the simple function problem, constraint G from 5 runs, α =0.75, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage
Figure 5.16. History plot for the simple function problem, objective F from 5 runs, α =0.75, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage. 80
Figure 5.17. History plot for the simple function problem, constraint G from 5 runs, α =0.75, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage. 80
Figure 5.18. Value of F for the simple functions for best design found over the number of cycles and values of α studied. This data is the average of 25 runs. The magnitude is the percent difference the actual best design. P _f was calculated using a quadratic response surface based on available solution points
Figure 5.19. Same data as presented in Figure 5.18, rearranged to have alpha on the X-axis
Figure 5.20. Contours of F and the constraint limit $G = 0$ over $[X_1, X_2]$. The green dot represents the DDO optimum design, and the red dot represents the RBDO optimum design
Figure 5.21. Value of F for the peaks problem for best design found over the number of cycles and values of α studied. This data is the average of 25 runs. The magnitude is

Figure 5.23. Absolute difference of P_f predicted by the response surface and the actual P_f for the best design found. The height of the colored bar is the average of 25 runs. The whiskers represent the range of the actual values. Note that in most cases the average is very close to the minimum, and typically only a single run had very high error. The "actual P_f " value compared against is the probability of failure as calculated using the same RA method, but with actual evaluations instead of a response surface.

Figure 5.28. History plot for the peaks problem, constraint G from 5 runs, α =0.25, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage......95

Figure 5.32. History plot for the peaks problem, constraint G from 5 runs, α =0.50, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage......97

Figure 5.36. History plot for the peaks problem, constraint G from 5 runs, α =0.75, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage......99

Figure 5.45. History plot for the peaks problem, constraint G from 5 runs, α =0.25, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage....108

Figure 5.49. History plot for the peaks problem, constraint G from 5 runs, α =0.50, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage....110

Figure 5.53. History plot for the peaks problem, constraint G from 5 runs, α =0.75, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage....112

Figure 5.55. Contours of objective F overlaid with the constraint limit (G = 0, dotted line) for the peaks problem. The green dot is the DDO optimum and the red dot is the RBDO

Figure 5.63. Same data as presented in Figure 5.62, without min and max bars (only average value is plotted). A horizontal line is provided showing the reliability constraint limit value.

Figure 5.64. History plot for the cantilevered beam problem, objective Volume from 5 runs, α =0.25, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage. 126

Figure 5.66. History plot for the cantilevered beam problem, constraint MaxDisp from 5 runs, α =0.25, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.

Figure 5.67. History plot for the cantilevered beam problem, objective Volume from 5 runs, α =0.25, E=60. The blue dots are the DDO stage, while the red dots are the RBDO stage. 128

Figure 5.69. History plot for the cantilevered beam problem, constraint MaxDisp from 5 runs, α =0.25, E=60. The blue dots are the DDO stage, while the red dots are the RBDO stage. 129

Figure 5.70. History plot for the cantilevered beam problem, objective Volume from 5 runs, α =0.50, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage. 129

Figure 5.72. History plot for the cantilevered beam problem, constraint MaxDisp from 5 runs, α =0.50, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage. 130

Figure 5.73. History plot for the cantilevered beam problem, objective Volume from 5 runs, α =0.50, E=60. The blue dots are the DDO stage, while the red dots are the RBDO stage. 131

Figure 5.74. History plot for the cantilevered beam problem, constraint MaxStress from 5 runs, α =0.50, E=60. The blue dots are the DDO stage, while the red dots are the RBDO stage
Figure 5.75. History plot for the cantilevered beam problem, constraint MaxDisp from 5 runs, α =0.50, E=60. The blue dots are the DDO stage, while the red dots are the RBDO stage. 132
Figure 5.76. History plot for the cantilevered beam problem, objective Volume from 5 runs, α =0.75, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage
Figure 5.77. History plot for the cantilevered beam problem, constraint MaxStress from 5 runs, α =0.75, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage
Figure 5.78. History plot for the cantilevered beam problem, constraint MaxDisp from 5 runs, α =0.75, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage
Figure 5.79. History plot for the cantilevered beam problem, objective Volume from 5 runs, α =0.75, E=60. The blue dots are the DDO stage, while the red dots are the RBDO stage. 134
Figure 5.80. History plot for the cantilevered beam problem, constraint MaxStress from 5 runs, α =0.75, E=60. The blue dots are the DDO stage, while the red dots are the RBDO stage. 134
Figure 5.81. History plot for the cantilevered beam problem, constraint MaxDisp from 5 runs, α =0.75, E=60. The blue dots are the DDO stage, while the red dots are the RBDO stage. 135
Figure 5.82. Value of Volume for best design found over the number of cycles and values of α studied. This data is the average of 20 runs. The magnitude is the percent difference from the best design
Figure 5.83. Same data as presented in Figure 5.82, rearranged to have alpha on the X-axis
Figure 5.84. Absolute difference of P_f predicted by the response surface and the actual P_f for the best design found for the MaxStress response. The height of the colored bar is the average of 20 runs. The whiskers represent the range of the actual values. Note that

Figure 5.88. History plot for the cantilevered beam problem, objective Volume from 5 runs, α =0.25, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage. 143

Figure 5.90. History plot for the cantilevered beam problem, constraint MaxDisp from 5 runs, α =0.25, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage. 144

Figure 5.93. History plot for the cantilevered beam problem, constraint MaxDisp from 5 runs, α =0.25, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage. 146
Figure 5.94. History plot for the cantilevered beam problem, objective Volume from 5 runs, α =0.50, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage. 146
Figure 5.95. History plot for the cantilevered beam problem, constraint MaxStress from 5 runs, α =0.50, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage. 147
Figure 5.96. History plot for the cantilevered beam problem, constraint MaxDisp from 5 runs, α =0.50, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage. 147
Figure 5.97. History plot for the cantilevered beam problem, objective Volume from 5 runs, α =0.50, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage
Figure 5.98. History plot for the cantilevered beam problem, constraint MaxStress from 5 runs, α =0.50, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage. 148
Figure 5.99. History plot for the cantilevered beam problem, constraint MaxDisp from 5 runs, α =0.50, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage. 149
Figure 5.100. History plot for the cantilevered beam problem, objective Volume from 5 runs, α =0.75, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage. 149
Figure 5.101. History plot for the cantilevered beam problem, constraint MaxStress from 5 runs, α =0.75, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage. 150
Figure 5.102. History plot for the cantilevered beam problem, constraint MaxDisp from 5 runs, α =0.75, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage. 150

Figure 5.103. History plot for the cantilevered beam problem, objective Volume from 5 runs, α =0.75, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage
Figure 5.104. History plot for the cantilevered beam problem, constraint MaxStress from 5 runs, α =0.75, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage. 151
Figure 5.105. History plot for the cantilevered beam problem, constraint MaxDisp from 5 runs, α =0.75, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage
Figure 5.106. Geometric and loading details of the torque arm. The left-hand circle has pin boundary conditions applied (all displacement DOF constrained)
Figure 5.107. Value of Mass for best design found over the number of cycles and values of α studied. The magnitude is the percent difference from the best design found over all studies. The value for α = 0.75, E = 60 is the best design found, so it has a value of zero on this graph. 158
Figure 5.108. Same data as presented in Figure 5.107, rearranged to have alpha on the X-axis
Figure 5.109. Absolute difference of P_f predicted by the response surface and the actual P_f for the best design found for the MaxStress response. The dark line at 0.01 is the constraint limit
Figure 5.110. Value of Mass for best design found over the number of cycles and values of α studied. The magnitude is the percent difference from the best design found over all studies
Figure 5.111. Same data as presented in Figure 5.110, rearranged to have alpha on the X-axis
Figure 5.112. Absolute difference of P_f predicted by the response surface and the actual P_f for the best design found for the MaxStress response. The dark line at 0.01 is the constraint limit
Figure 5.113. Value of Mass for best design found over the number of cycles and values of α studied. The magnitude is the percent difference from the best design found over all runs performed

1 Introduction

1.1 Motivation

With rising global competition, engineers are increasingly required to produce better products with respect to cost, quality, and other performance measures. Simultaneously, they have less time and money to do this. These competing objectives of making better products faster and cheaper has forced many industries to change how they do product design. In recent years several technological revolutions in computing have occurred, facilitating a faster, cheaper, and better design process.

The first revolution that computing brought was Computer-Aided Design (CAD). CAD is used to create 2D and 3D digital representations of products in the computer. CAD enabled much more complex products to be designed and evaluated at a dramatically more rapid pace than traditional drafting [1]. Eventually this was extended to 3D CAD (or solid modeling), which is a prerequisite for the next revolution.

The second revolution in engineering enabled by computing technology was the use of computational simulation in evaluating product designs. Technologies like Finite Element Analysis (FEA), Computational Fluid Dynamics (CFD) and Computational Multi-Body Dynamics (MBD) allowed engineers to test their ideas virtually, skipping the time-consuming and expensive physical prototyping process until much later in the design cycle. Broadly, this is known as Computer-Aided Engineering (CAE) [2], which includes many sub-disciplines, including CAD and a host of methods for performing simulation.

The third revolution in engineering is still taking place in some industries and disciplines: automated optimization. This is enabled by the large-scale adoption of CAE technologies in industry. Automated optimization enhances an engineer's work in two ways:

- Automation of time-consuming model building, execution, and post-processing.
 The engineer can identify key parameters that govern the design's performance, and automate the evaluation of many designs with different parameter values.
- High quality optimization algorithms which are already implemented in commercial optimization tools. These algorithms control an iterative search process for the values of design parameters that meet all design goals.

The engineer now has the ability to spend significantly less time building simulation models, executing them, and post-processing the results. Additionally, through the use of modern optimization algorithms, the engineer can spend time identifying a promising design concept, and let the optimizer find the exact set of parameter values that yield the best performing design.

As the optimization revolution has progressed, one area of interest that remains a fertile topic of research is Reliability-Based Design Optimization (RBDO). RBDO is the process of optimizing a design with respect to performance criteria that have a reliability requirement. No product encounters exactly the environment it is tested in (virtually or physically), and no product is manufactured exactly as designed in CAD. This variation can be simulated as a stochastic variation on input parameters in the simulation model. An example might be the variation of the location of a feature, which is controlled to

within some tolerance range. Another example is the variation in loading applied to a structure.

Optimization performed without accounting for stochastic variation is called Deterministic Design Optimization (DDO). The standard DDO problem statement is:

$$\begin{array}{ll} \text{minimize} & F(X) \\ \text{subject to} & G_i(X) \ge 0, i = 1, 2, ..., n \\ & H_j(X) = 0, j = 1, 2, ..., m \\ \\ \text{by varying} & X^L \le X \le X^U \\ \text{where} & X = \left\{ x_1, x_2, ..., x_k \right\}^T \\ & x^L = \left\{ x_1^L, x_2^L, ..., x_k^L \right\}^T \\ & X^U = \left\{ x_1^U, x_2^U, ..., x_k^U \right\}^T \\ \end{array}$$

$$\begin{array}{l} \text{(1.1)} \\ \text{where} & X = \left\{ x_1, x_2, ..., x_k^L \right\}^T \\ & X^U = \left\{ x_1^U, x_2^U, ..., x_k^U \right\}^T \end{array}$$

The objective function, F(X), is being minimized, subject to inequality constraints, $G_i(X)$, and equality constraints, $H_j(X)$. This is done by varying the design variables Xindependently in the range X^L to X^U . Note that while in DDO the convention is to define the feasible region such that G_i has negative values, in RBDO this is often reversed. The convention of the feasible solutions being those which have G_i which are positive is used throughout this thesis in general statements. The example problems solved do not necessarily use this convention. A design that meets all of the constraints is said to be *feasible*, while a design that fails to meet one or more constraints is *infeasible*. For many optimization problems solved by engineers, there are no equality constraints, and this type of constraints will be ignored from this point on. Commonly, the solution to the above optimization problem is one where the value of one or more of the inequality constraints is exactly or nearly zero. These are called *active* constraints. This means that if the design variable values or other simulation input parameters were to change by a relatively small amount, the design might no longer satisfy the constraint(s). Similarly, if the active constraint(s) were relaxed somewhat, the new solution would have a better value for the objective function.



Figure 1.1. Example of a two variable problem which illustrates the concept of active constraints. For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this thesis.

As depicted in Figure 1.1, if constraint G_1 is applied then Opt_1 is the optimum design. If this constraint is relaxed to the location of constraint G_2 , then the objective function can be improved to the point Opt_2 .

A design that has active constraints, while technically optimal from a DDO perspective, may not be good to implement and sell because it could have a higher than desired likelihood of failing in circumstances where the service environment is not exactly the same as the simulated environment. This design would not be considered *reliable*.



Figure 1.2. Example of a two variable problem with variation.

As illustrated in Figure 1.2, a deterministic optimum is often found on the limit of an active constraint. Once stochastic variation of the variables is accounted for, it is obvious that many designs will fail to meet the constraint.

To account for this, the engineer could set the constraint with a safety factor in mind (since Ching [3] showed that the safety factor and probability of failure are functionally equivalent), knowing that this will result in a more reliable design. However, in many cases it is difficult to determine a suitable safety factor value. In these cases, it may be easier and more accurate to specify a number of allowable failures out of a population of manufactured products. This can be determined, for example, based on known costs for warranties and replacements of failed units. However, typically there is no way to equate a desired failure rate to a safety factor. Instead, one must evaluate the reliability of a given design directly using one of a number of methods, some of which will be described later.

So, the engineer really wants to find a design that has a failure rate no higher than a certain value, but which has a minimum value for the objective function. This is described in the sequel, and is called the standard RBDO problem statement:

$$\begin{array}{ll} \text{minimize} & F(X,B) \\ \text{subject to} & P[G_i(X,B) \leq 0] \leq P_{fi}, \ i=1,2,\ldots,n \\ \\ \text{by varying} & X^L \leq X \leq X^U \\ \text{where} & X = \left\{ x_1, x_2, \ldots, x_k \right\}^T \\ & X^L = \left\{ x_1^L, x_2^L, \ldots, x_k^L \right\}^T \\ & X^U = \left\{ x_1^U, x_2^U, \ldots, x_k^U \right\}^T \\ \end{array}$$

$$\begin{array}{l} \text{(1.2)} \\ \\ X^U = \left\{ x_1^U, x_2^U, \ldots, x_k^U \right\}^T \end{array}$$

where P[*] is the probability operator, and P_f is the maximum allowable probability of failure. *B* is the vector of stochastic parameters that also govern the performance of the system, along with the design variables *X*, which may be stochastic as well. Compared to using safety factors, an optimization problem statement of this form has the advantage of specifying the desired failure rate directly, which is usually easier to relate to real-world data. The main disadvantage, however, is that it requires each design to be evaluated for its nominal performance as well as its reliability. Typically, the latter calculation is much more computationally expensive than the former. Thus, it usually takes much longer to perform RBDO than DDO, sometimes prohibitively so.

1.2 Literature Review

The review of the relevant literature reveals a strong body of work relating to the use of response surfaces (RS), also called surrogate or meta-models. Bucher and Most [4] compared least-squares response surfaces with Artificial Neural Networks (ANN)

and Radial Basis Functions (RBF) with an eye toward reliability analysis (RA). For low numbers of sampling points, the error can be quite high (on order of 1000%). They came to the same conclusion as many others, which is that one cannot determine which type of response surface to use *a priori*. Which one is best depends on the problem being solved. Additionally, the response surface method was originally developed in the fields of chemical and industrial engineering, and so is not always well suited for use with RA, where the accuracy of the constraint limit surface is paramount.

Roux, Stander, and Haftka [5] attempted to use quadratic response surfaces to model global behavior of simple trusses. Unfortunately, they found that the accuracy of the RS was largely dependent on the regression scheme used and the scaling or inverting of variables. They were unable to create a global RS to effectively model the entire design space of a simple three-bar truss. Once they reduced the variable ranges to a smaller subregion, an acceptable fit could be found.

Taflanidis and Cheung [6] used a technique for fitting RS called Moving Least Squares (MLS). In this method, each support point has a neighborhood created around it. Weights are assigned to that point based on other points in the neighborhood. The neighborhood size used in this study is twice the minimum needed to fit a surface. The resulting surface is one that minimizes the error locally around each support point, and can capture a more complex function than standard global least squares fits. It does require more support points to do this, however.

Simpson et al [7] compared a quadratic RS with Kriging for modeling an aerospike nozzle modeled with CFD. These two different RS types yielded comparable accuracy on this problem, which might be thought to be more challenging initially.

Simpson et al [8] surveyed the literature on metamodeling and provide several useful recommendations for those applying metamodels to computer-based experiments. One key point is these experiments are deterministic. That is, the result of a simulation is the same each time it is run, provided the same inputs are used. The error in RS generally is made up of systemic error and random error. However, for computer-based experiments, the random error term is zero. Therefore, if there is no random error, statistical testing is not appropriate. This is significant, since some users of metamodels apply the general principles directly, and believe statistical measures of the metamodels' significance to be useful, when in fact they are measuring systemic error. This error is simply a measure of how ill-suited the metamodel in use is for modeling the underlying data with that particular fitting technique.

The authors also note that among aerospace engineers D-optimal arrays tend to be most popular, while orthogonal arrays are more popular among mechanical engineers. No potential reason for this is offered. Optimization seems to be the major driver of the use of DOE and RSM, since large FE and CFD models are so timeintensive.

The authors next recommend several additional measures of RS accuracy beyond the standard R² measure. These include max(abs(error)), avg(abs(error), and the leave-one-out-strategy. This last strategy entails leaving a single point out of the fit, and then evaluating the RS error at that point. Repeat this for each point in the set and take the maximum or average error.

In the end the authors recommend using the Kriging metamodel (as an alternative to polynomial RS) in situations where the number of factors is moderate to

large and/or the response has highly nonlinear behavior, since it is an interpolating technique and can have any underlying basis.

Some researchers focused on studying and adapting standard RS techniques in the context of RA. Among them, Rajashekhar and Ellingwood [9] found that including information about how the variables are distributed had similar accuracy and efficiency to standard RS fitting techniques. Additionally, sampling near the tails of the distributions was surmised to work better for RA since failure would typically happen "under extreme circumstances". This also, was not significantly better than standard methods.

Kaymaz and McMahon [10] presented a new technique for fitting RS with the intent to use an MPP-based algorithm like FORM. In the case of FORM, the accuracy of the RS is most important at the MPP. It is possible that poor RS accuracy elsewhere could cause FORM to diverge or converge to another local minimum. That aside, the estimation of β is entirely based on the final location in the last FORM iteration, so predicting the location of the MPP accurately will have the most influence on *P*_f prediction accuracy. The authors propose weighting the regression in proportion to each point's distance from the constraint limit. Additionally, the FORM or SORM algorithm is used within the RS fitting routine to develop points closer to the MPP. This method was more accurate than standard RS fitting techniques, though using FORM to estimate the MPP, then re-sampling about the approximate MPP and fitting a new RS yielded far more increase in accuracy than did the weighted regression technique.

Gomes and Awruch [11] compared polynomial RS to ANN in the context of RA. Since simple constraint limits with closed form solutions were used in the comparisons, it is not surprising that the accuracy of the two were comparable.

Gayton, Bourinet, and Lemaire [12] proposed a new way to fit an RS called CQ2RS (Complete Quadratic Response Surface with ReSampling) which is designed for use with MPP based RA. This technique first uses the engineer's estimate of the location of the MPP, then iteratively adds and removes points based on confidence intervals of where the MPP is thought to be. One interesting recommendation is that when fitting an RS for use with MPP-based RA, the sample matrix should be created in U-space, rather than X-space.

Alliax and Carbone [13] proposed a technique for fitting a RS for use with RA which is similar in some ways to other ideas. The first step is to use an MPP-based algorithm to estimate the location of the MPP, and then fit points around that. This step is similar to Bucher and Bourgund's idea. These authors also recommend designing the array in U-space, as Gayten et al also suggested. Finally, the authors use the sensitivities of the response at the MPP to align the DOE array to have one axis normal to the constraint limit surface. This method clearly requires more evaluations than a standard RS fit, however it was shown to be more accurate on a number of analytical examples.

Guan and Melchers [14] explored some of the potential pitfalls in using RS for RA. It is noted that this method of estimation may either under- or over-predict P_{f} , and there is little or no way to know for sure *a priori*. The paper goes on to show that when using standard RS fitting technique, perturbing the location of a sampled point can

reveal instability in the prediction of P_{f} . The authors note that a referee of this paper says that the issues with RS are well-known, though this paper attempts to examine the consequences of these issues.

Bucher and Bourgrund [15] note a well-known problem with FORM, that it is not well suited to solve many problems. These problems do not respond well to linearization, or may have too many variables for FORM to be efficient. They instead recommend a technique for fitting an RS where an iterative scheme is developed to move and scale the sample array to be more representative of the constraint limit. This is shown to give a more accurate prediction of P_f on the RS. While this technique does require considerable evaluations (twice as many, in fact, than standard RS fitting techniques), it is shown to be more accurate and more efficient than FORM some representative problems.

There has been extensive research into the use of FORM and SORM, much of it highlighting these algorithms' poor performance on a variety of both contrived and realworld problems. Valdebenito et al [16] studied the influence of the MPP on difficulty of calculating *P_f*. They found that for low numbers of random variables (< 20) the distance from the nominal design to the MPP had more influence than the number of variables. This distance was not as important for larger numbers of variables. They also found that for larger numbers of variables. They also found that for larger numbers of variables inherent in FORM and SORM produced considerable error. They recommend that for large numbers of variables these algorithms should only be used if the key assumptions for FORM are met, e.g. linearity (or near linearity) of the constraint and normally distributed variables.

Additionally, they looked at this issue with respect to structural analysis including nonlinearities. In situations where nonlinearities are included in the model and there are many variables, there may be multiple failure regions, which is not accounted for by FORM, and can lead to considerable error.

Huang and Griffiths [17] applied the FORM algorithm to the RA of a geomechanics problem. They found reasonably good agreement with direct RA methods when the variables were distributed normally, but poor agreement when they were distributed log-normally.

Liu and Der Kiureghian [18] explored several algorithms for performing RA based on the most probable point (MPP) approach, which will be explained in section 3.3. These included Gradient Projection, Penalty, Augmented Lagrangian, SQP, HL-RF, and Modified HL-RF (MHL-RF). Each of these methods is an MPP-based algorithm. These were tested on five different problems, and no clear winner emerged. In fact, several algorithms failed to converge on one or more of the problems, making them ill-suited for general use.

Finally, while many use FORM for RA, some do not fully understand the underlying assumptions of that algorithm. Koduru and Haukaas [19] examine those assumptions, and explain what the potential consequences are when they are violated. These include divergence of the algorithm, poor accuracy of the P_f prediction, and convergence to an incorrect MPP. Since it is so easy for FORM's key assumptions to be violated on real problems, FORM is not recommended for general use in RA.

Even the Performance Measure Approach (PMA, explained further in section 3.4.1) has similar pitfalls. Pretorius, Craig, and Haarhoff [20] showed that the HMV^[21] method, which is a recasting of the problem FORM attempts to solve, is not tolerant of noise on the constraint surface. This was shown through work where a Kriging surface was used with a submerged entry nozzle analyzed by 2D CFD.

1.3 Obstacles to Using RBDO

There are two major obstacles to using RBDO in an industrial setting:

 It can be difficult and/or expensive to determine how inputs vary stochastically in the service environment.^[22]

2. Evaluating the reliability of a design is time-consuming and/or expensive.^{[14][23]} Overcoming the first obstacle can be done through reasonable assumptions and investing in the necessary data gathering. The second obstacle, however, can be quite a challenge even if the engineer has access to the necessary data on the input variations and a predictive model. This is because of the double-loop problem.

A 'function evaluation' is defined as the required calculations necessary to fully evaluate the deterministic performance of a single design. By this definition, the nominal performance of a design requires a single function evaluation. Evaluating the reliability of a design, that is, the likelihood of a design failing to meet one or more constraints due to stochastic variation of input parameters, often requires many function evaluations. The double-loop problem stems from this issue.

To perform RBDO, two loops are needed.^{[24][25]} The outer loop is an optimization of the RBDO problem statement. At each iteration of the optimization search, the

nominal performance of the current design is evaluated (at a cost of one function evaluation). After that, the reliability of this design must be evaluated. This is performed by an inner loop, where many designs are iteratively evaluated (at a cost of one function evaluation per design). Once this inner loop has completed the reliability calculation, the iteration in the outer loop can end and move on to the next. In this way the many function evaluations needed to perform optimization in the deterministic sense are compounded by an expensive inner loop when reliability is included in the optimization. The double-loop is illustrated as a flow chart in Figure 1.3



Figure 1.3. Flowchart illustrating the double-loop required to perform reliability-based design optimization.
When function evaluations are expensive (e.g., many minutes or hours of CPU time), as is common in industry, RBDO can quickly become outrageously expensive when executed using this double loop process. Developing a more efficient process for RBDO is the main goal of this thesis.

1.4 The Current Study

The current study is intended to motivate, develop, and demonstrate an efficient way to perform reliability-based design optimization. As discovered by other researchers, the main challenge in RBDO is navigating the tradeoffs between the necessary computational expense to calculate P_f and the associated accuracy. The most accurate way to perform RBDO using millions of function evaluations is completely intractable on most industrial problems. As a result, some loss of accuracy must be accepted in return for more computationally feasible solutions.

One obvious approach is the use of response surfaces in place of the expensive inner loop evaluations. Global response surfaces have challenges and issues that are well-known, but local response surfaces are much more forgiving and have greater potential for generating sufficiently accurate solutions. The current approach will use local response surfaces to calculate the probability of failure of each design.

Moreover, these local response surfaces will be developed based on design data available from a previous deterministic design optimization study, which is intended to further increase the efficiency of the search by honing in on the general area of the reliability-based optimal design.

In the first chapter the motivation for this work was established, and a review of the relevant literature was performed. Key obstacles were also identified. In the second

17

chapter the statistical foundation necessary to understand RA and RBDO is developed. In the third chapter a few of the most common ways to perform RA are explained. In the fourth chapter the proposed approach for efficiently and accurately solving the RBDO problem is detailed. In the fifth chapter this strategy is explored through solving several example problems. The sixth chapter summarizes the results and concludes, and suggests future related research topics.

2 Statistical Foundation

In this section, the statistical and probabilistic concepts needed to understand how to perform reliability analysis will be presented.

2.1 Random Variables

A random variable, denoted *X*, can take on any real value *x*, such that $-\infty < x < \infty$. Herein, random variables are indicated by uppercase letters, and the value of a random variable is indicated by a lowercase letter.

Random variables can be continuous or discrete. Continuous random variables may take on any real value that is valid for their distribution. Discrete random variables may take on only a prescribed set of values, for example, $\{x_1, x_2, x_3, x_4, ..., x_n\}$. In the case of discrete random variables, each value from the set can have its own variation. For instance, the discrete set may be a set of materials, and the variation could be applied to multiple aspects of the materials' definitions. For example, Young's Modulus and yield stress may both have their own variation for a given material.

2.2 Probability Density Function

If a random variable is sampled a finite number of times, then one can create a *frequency diagram* or *histogram* that represents the probability that the variable will take a particular value. This graph is created by dividing the data into discrete intervals of equal size. Each interval is called a *bin*. A rectangle is drawn which has width equal to the bin size and height equal to the number of samples that occur in that bin. The height

of the rectangles should be normalized so the sum of all their areas is one. Once this is done, the histogram is an approximation of the *probability density distribution* of the data. An example is shown in Figure 2.1.



Figure 2.1 A histogram of sampled data with 32 bins. Since box heights are normalized in this case, it is also an approximation of the probability density distribution. Plot generated in Minitab 15.^[26]

The probability of a randomly selected sample falling in a particular bin can be determined by finding the area of the rectangle for that bin. Note that the height of the bin can be seen as the *probability density* of the bin.

$$Probability = Height of selected bin \times Width of selected bin$$
$$= Probability density \times Interval size$$
(2.1)

If a very large number of samples were taken and the bin widths were allowed to become infinitesimally small, the histogram would become a continuous curve. This is the *probability density function (PDF)*, and for the random variable *X* it is denoted as $f_X(x)$. The PDF is not defined for discrete random variables.



Figure 2.2 Histogram from Figure 2.1 with probability density function fit to the data overlaid. Plot generated in Minitab 15.

2.3 Cumulative Distribution Function

The *Cumulative Distribution Function* (*CDF*) is another way to describe random variables. The CDF is defined for $-\infty < x < \infty$ and varies from 0 to 1. The CDF is denoted as $F_X(x)$, and is equal to the probability that *X* is less than or equal to *x*, see Figure 2.3. To calculate $F_X(x)$, integrate the PDF for all values of *X* less than or equal to *x*, as shown below.

$$F_X(x) = \int_{-\infty}^{x} f_X(t)dt$$
 (2.2)

The probability of X having a value between *a* and *b* is thus:

$$F_X(a) - F_X(b) = \int_a^b f_X(x) dx$$
 (2.3)



Figure 2.3 Cumulative Distribution Function for data used in Figure 2.1 and Figure 2.2. Plot generated in Minitab 15.

2.4 Mean

The *mean* of a set of samples, also known as the *expected value* or *average*, describes the central tendency of a random variable. The mean of a random variable *X* is denoted as μ_X . This a weighted average of all the values a random variable may take. If a set of samples, { $x_1, x_2, x_3, x_4, ..., x_n$ }, is created, then the mean is calculated as:

$$\mu_X = \frac{1}{n} \sum_{i=1}^{n} x_i$$
 (2.4)

If a probability density function exists, then the mean of X can be calculated as:

$$\mu_X = E(X) = \int_{-\infty}^{\infty} x \cdot f_X(x) dx$$
 (2.5)

From equation 2.5, it can be seen that the mean value is the distance from the origin to the centroid of the PDF. Since the mean is the first moment of area of the PDF, the mean is also called the *first moment*.

Also introduced in equation 2.5 is the expectation operator, E[*], which is the defined as "the weighted average of all possible values that the random variable can take on" [27]. The expectation operator has several interesting properties, though only the linearity of the operator will be shown here for use in the sequel:

If
$$z = x_1 + x_2 + x_3 + x_4 + \dots + x_n$$
,
then $E(Z) = E(X_1) + E(X_2) + E(X_3) + E(X_4) + \dots + E(X_n)$ (2.6)

2.5 Variance and Standard Deviation

The *variance* of a PDF, also called the *second central moment*, is a measure of the variation of the data about the mean. The variance is defined as:

$$V(X) = E\left[\left(X - \mu_X\right)^2\right]$$

= $E\left(X^2\right) - 2E(X)\mu_X + \mu^2 = E\left(X^2\right) - 2\mu_X^2 + \mu_X^2 = E\left(X^2\right) - \mu_X^2$ (2.7)

As a geometric analogy, the variance represents the moment of inertia of the PDF about the mean. The standard deviation, σ_X , is another way to describe the variation about the mean, and is more commonly used than the variance because it has the same units as *X* and μ_X . The standard deviation can be calculated as:

$$\sigma_X = \sqrt{V(X)} \tag{2.8}$$

The *coefficient of variation (COV)*, δ_X , is the nondimensionalized form of the standard deviation:

$$\delta_X = \frac{\sigma_X}{\mu_X} \tag{2.9}$$

2.6 Probability Distributions

The probability density distributions of random variables can take many shapes, so there are many analytical probability distributions typically used to model random variables. This section describes the characteristics of two of the more commonly used distributions.

2.6.1 Gaussian Distribution

The Gaussian distribution (also called the *normal distribution*) is one of the most popular distributions for engineering applications due to its ease of use and the *central limit theorem* (*CLT*). The ease of use comes from being able to define the distribution by only specifying the mean and standard deviation. The *CLT* states that the sum of a set of randomly distributed, independent variables will be distributed normally. This accounts, in part, for the large number of phenomena that can be accurately described by the normal distribution. The PDF for the normal distribution is expressed as:

$$f_X(x) = \frac{1}{\sigma_X \sqrt{2\pi}} exp\left[-\frac{1}{2} \left(\frac{x - \mu_X}{\sigma_X}\right)^2\right], \quad -\infty < x < \infty$$
(2.10)

where μ_X and σ_X designate the mean and standard deviation of *X*, *respectively*. If *X* is described by a normal distribution, the notation N(μ_X , σ_X) can be used to express *X*. The PDF is shown in Figure 2.4.



Figure 2.4 Normal (or Gaussian) distribution for several values of $\mu \chi, \sigma \chi$. [28]

Due to the shape of the distribution curve, the PDF of a normally distributed variable is known as a *bell curve*. The normal distribution is symmetric about the mean, and has inflection points one standard deviation on either side of the mean, at $x = \mu_X \pm \sigma_X$. The fraction of values within one through six standard deviations from the mean are approximately 68.26%, 95.44%, 99.74%, 99.994%, 99.99994%, and 99.999998%, respectively. The normal distribution has the following properties as a result of the CLT.

1. Any linear sum of normally distributed random variables will also be normally distributed. For example, let $X_1, X_2, ..., X_n$ be independent, normally distributed random variables. If *Z* is defined as

$$Z = a_0 + a_1 X_1 + a_2 X_2 + \dots + a_n X_n$$
(2.11)

then Z will be normally distributed. The mean and standard deviation of Z can be calculated as

$$\mu_{Z} = a_{0} + \sum_{i=1}^{n} a_{i}\mu_{i}, \qquad \sigma_{Z} = \sqrt{\sum_{i=1}^{n} (a_{i}\sigma_{i})^{2}}$$
(2.12)

 Conversely, a nonlinear function of normally distributed random variables will not necessarily be normally distributed. Also, a function (linear or otherwise) of nonnormally distributed random variables will also not necessarily be normally distributed.

A normally distributed random variable can be normalized as follows:

$$z = \frac{x - \mu}{\sigma} \tag{2.13}$$

This is sometimes called the z-score or standard score. The normalized Gaussian distribution is called the *standard normal distribution*, N(0,1). The standard normal distribution is denoted $\varphi(*)$ and has the PDF:

$$\phi(z) = f_X(z) = \frac{1}{\sqrt{2\pi}} exp\left(-\frac{z^2}{2}\right), \quad -\infty < z < \infty$$
 (2.14)

The CDF of the standard normal distribution is denoted $\Phi(*)$:

$$\Phi(z) = F_X(z) = \int_{-\infty}^{z_0} \frac{1}{\sqrt{2\pi}} exp\left(-\frac{z^2}{2}\right) dz$$
 (2.15)

If $\Phi(z_p) = p$, then:

$$z_p = \Phi^{-1}(p)$$
 (2.16)

Furthermore, $\Phi(z)$ is only defined for positive values of *z*. For the case where *z* < 0, we have:

$$\Phi(-z) = 1 - \Phi(z)$$
 (2.17)

The CDF is symmetric about zero, so that:

$$z_p = \Phi^{-1}(p) = -\Phi^{-1}(1-p)$$
 (2.18)

2.6.2 Log-normal Distribution

The log-normal distribution is also commonly used, especially for quantities that cannot be negative. A random variable that is log-normally distributed has a logarithm that is normally distributed. The PDF of the log-normal distribution is given below. The two parameters of this distribution are typically denoted by μ and σ , however these are not the mean and standard deviation of the variable, but of the natural logarithm of the variable. This convention is used for this section only.

$$f_X(x) = \frac{1}{x\sqrt{2\pi\sigma^2}} exp\left[-\frac{(\ln(x) - \mu)^2}{2\sigma^2}\right], \quad -\infty < x < \infty$$
 (2.19)

PDFs of several log-normal distributions are shown in Figure 2.5.



Figure 2.5. Log-normal distributions for several values of μ , σ . [29]

3 Measuring Reliability

There are several methods for performing reliability analysis (RA) on an engineering system. All methods have the same goal: the accurate prediction of the probability of failure. For purposes of this discussion, the reliability is defined with respect to a single constraint. In a system with multiple deterministic constraints, the RBDO statement would have multiple stochastic constraints.

3.1 Probability of Failure and the Reliability Index

The objective of reliability analysis is to determine the probability that a system will violate a prescribed set of constraints that govern the system's behavior. The probability that one or more constraints will be violated is called the *Probability of Failure*, P_{f} . For example, the reliability of a steel bar is measured in terms of the probability that the stress in the bar exceeds the maximum allowable level.

Without loss of generality, the ensuing discussion will consider systems with only one constraint. For systems with more than one constraint, simply repeat the indicated steps for each constraint, assuming the constraints are statistically independent.

In a DDO problem, a constraint is defined in terms of a function, G(X), such that when G(X) < 0, the constraint is violated and the system fails. In an RBDO problem the value of P_f is defined as the probability that the constraint will be violated. So we express the constraint on P_f as $P[G(X, B) > 0] \le P_f$ For example, an automotive part designer can use the value of P_f to estimate the warranty costs due to anticipated part failures, and adjust the design as called for by the product requirements.

The *reliability index*, β , is another measure used to evaluate reliability. It is really just a different way of expressing P_{f} .

$$\beta = \frac{\mu g}{\sigma g}$$
(3.1)
$$P_f = 1 - \Phi(\beta) = \Phi(-\beta)$$

The reliability index is not valid for non-normally distributed constraints, which are quite common due to the nonlinear nature of many constraints. Because the reliability index is pervasive as the parameter of interest in reliability analysis, a method for locally approximating the normal distribution of a random variable with a non-normal distribution is desirable. This will be discussed in a later section, but this approach will not be used in the current study.

3.2 Sampling Methods

Sampling methods of RA involve sampling the random variables according to their distributions and using frequency of constraint violation to determine the probability of failure.

3.2.1 Monte Carlo Sampling

Monte Carlo sampling (MCS) [30] is the most straightforward way to estimate β when the constraint is an implicit function. Typically it also takes the greatest amount of CPU time and resources. MCS is performed by repeated sampling, where values for the random variables are randomly selected and *G*(*X*) is evaluated for each sample. The set of chosen values for each random variable conforms to the respective distribution. The probability of failure, that is, the probability that *G* will be violated, *P*_f, is given by:

$$P_{f} = \frac{number \ of \ samples \ where \ G \ is \ violated}{total \ number \ of \ samples \ evaluated}$$
(3.2)

If G is normally distributed, then the reliability index, β can be found as:

$$P_f = 1 - \Phi(\beta) = \Phi(-\beta) = \Phi(P_f)$$
(3.3)

As the number of samples increases, the accuracy of this method increases. In order to obtain a useful result, this method typically requires orders of magnitude more samples than the methods outlined in later sections. This is because the values of *X* at the tails of the distributions are far less likely to be sampled than those near the mean, but often the system in question fails only when *X* takes on lower probability values. Therefore, many evaluations are performed to get a reasonable number of samples near the tails of the distribution.

MCS is simple to implement, however, and is known to converge to the correct value of P_f for any problem given a sufficient number of samples ("sufficient number of samples" being problem dependent). If the problem has a very short evaluation time, or evaluations are performed on a surrogate model with a short evaluation time, this method can be used for reliability prediction in a reasonable timeframe.

Confidence Intervals (CI) for MCS can be found through a number of means. The "exact" formulation for CI bounds using the binomial distribution is known to give conservative estimates. That is, the actual CI bounds are known to be inside of the bounds given by the "exact" method. This method is considered too conservative, so as recommended by Brown [31], the Wilson score test interval is presented here. The CI is dependent on the number of simulations performed and the value of the P_f being predicted. This is illustrated in Table 3.1.

Table 3.1 Table of CI bounds for MCS where	1.000.000 simulations are performed.

Value of P _f to be predicted	P _f Low Bound	P _f High Bound	P _f Low (Percentage)	P _f High (Percentage)
1.000E-06	1.765E-07	5.665E-06	-82.35%	466.49%
1.000E-05	5.432E-06	1.841E-05	-45.68%	84.09%
1.000E-04	8.223E-05	1.216E-04	-17.77%	21.61%
0.001	9.399E-04	0.001064	-6.01%	6.39%
0.0027	0.002600	0.002804	-3.70%	3.84%
0.01	0.00981	0.01020	-1.93%	1.97%
0.05	0.04957	0.05043	-0.85%	0.86%
0.1	0.09941	0.10059	-0.59%	0.59%

3.2.2 Latin Hypercube Sampling

Latin Hypercube sampling (LHS) is a modification of MCS that reduces the number of samples needed to obtain the reliability index with respectable accuracy. This is performed by increasing the likelihood of samples being taken from the low probability regions of the distribution compared to MCS, while ensuring that each design is as likely to be chosen as all others.

As first described by McKay, et al [32], in LHS each random variable, X_i , is divided up into *N* strata, where each stratum has equal probability 1/*N*. Each stratum is then randomly sampled once for each variable. The components of each variable are matched at random with the other random variables to generate N sets of points where the constraint function should be evaluated. The value of P_f can be evaluated using the same formula as in MCS, (**3.2**), since all of the points sampled have the same probability of being chosen. Stein [33] showed that LHS has less variance than MCS, and in the limit MCS and LHS approach the same solution. While confidence intervals for MCS can be given via the binomial distribution, CI for LHS can only be given through empirical means. This because the rate at which LHS converges is dependent on the problem (the shape of the constraint limit surface) and the type of LHS used.^[34]

3.3 Most Probable Point Methods

Most Probable Point (MPP) methods for RA determine reliability by casting the problem as an optimization problem in which the objective is to determine the most probable point of failure, as described below.

3.3.1 The Hasofer-Lind Transformation and the MPP

The *Hasofer-Lind (HL)* transformation [35] is used to transform a normally distributed random variable *X* into a random variable *U* having a *standard* normal distribution. For normally distributed random variables, the transformation is:

$$U = \frac{X - \mu_X}{\sigma_X} \tag{3.4}$$

This transformation is useful in that the units of distance in *U*-space are standard deviations of the variables. For example, a point in *U*-space that is three units from the

origin is three standard deviations from the mean. This aspect of *U*-space is often utilized to calculate β , as outlined below.

The Most Probable Point of failure is often used in advanced methods for reliability analysis because it directly gives the value of β . The MPP, denoted as U^* , is the point that lies on the limit-state function (G(U) = 0) and is the minimum distance from the origin in *U*-space. It is sometimes, but not always true, that the MPP will be the minimum distance from the mean to the constraint limit in both the X- and U-spaces.

The value of β is calculated using the *U*-space transformation of the random variables:

$$\beta = \sqrt{\sum_{i=1}^{n} U_i^2} \tag{3.5}$$

where *n* is the number of random variables. This value is the spatial distance (in *U*-space) of the MPP from the origin. The MPP is illustrated in Figure 3.1 for a two variable problem.



[36] Figure 3.1 Hasofer-Lind transformation and location of MPP in U-space.

From this, it is clear that the reliability of a system can be calculated by finding the location of the MPP, which can be cast as an optimization statement:



One notable assumption inherent in MPP-based RA methods is that of a single failure region. If there are multiple failure regions, then the whole concept of an MPP breaks down.

3.3.2 Mean Value First-Order Second Moment Method

The Mean Value First-Order Second Moment (MVFOSM) method is the starting point from which many commonly used reliability analysis methods begin. The basic idea for the MVFOSM method is to solve the optimization problem specified in (**3.6**) by performing a Taylor series expansion about the nominal point [36].

$$\tilde{G}(X) = G(\mu_X) + \nabla G(\mu_X)^T \left(X_i - \mu_{X_i}\right)$$
(3.7)

where $\mu_X = \{\mu_{x_1}, \mu_{x_2}, \dots, \mu_{x_n}\}^T \mu_X = \{\mu_{x_1}, \mu_{x_2}, \dots, \mu_{x_n}\}^T \mu_X = \{\mu_{x_1}, \mu_{x_2}, \dots, \mu_{x_n}\}^T$, and $\nabla G(\mu_X)^T$ is the gradient of G evaluated at μ_X . The mean and standard deviation of the approximate constraint function is given below:

$$\mu_{\tilde{G}} \approx E[G(\mu_X)] = G(\mu_X)$$

$$\sigma_{\tilde{G}} = \left[\sum_{i=1}^{n} \left(\frac{\partial G(\mu_X)}{\partial x_i}\right)^2 \sigma_{x_i}^2\right]^{\frac{1}{2}}$$
(3.8)

and the reliability index is

$$\beta = \frac{\mu_{\tilde{G}}}{\sigma_{\tilde{G}}} \tag{3.9}$$

This method only produces accurate results if G(X) is linear or nearly linear. However, it is particularly efficient, since it only requires function evaluations for the nominal point and its gradient. Notably, it assumes that X is distributed normally.

3.3.3 Hasofer-Lind Algorithm

The Hasofer-Lind (HL) algorithm [35] further develops the concept from MVFOSM of a Taylor-series expansion as surrogate to the actual limit-state function. The concept of the HL transformation and the MPP are introduced to facilitate the reliability analysis strategy.

A single HL iteration can be described as Gradient Descent [36] where the step size is determined by the linearity assumption inherited from the MVFOSM method, so no line search is necessary. The steps proceed as follows:

- 1. Initialize the current point, X^* , at the mean point.
- 2. Calculate the value of $G(X^*)$ and the gradients of G at X^* .
- 3. Calculate the value of β using (3.8) and (3.9).
- 4. Convert X* to U* using (3.4).
- Calculate the new U* by stepping along the negative gradient of G by a distance of β.
- 6. Calculate the new value of X*.
- 7. Repeat steps 2 through 6 until β converges.

In the linear case, the Taylor-series expansion is exact, and so the computed β is the exact (not estimated) distance from the nominal point to the MPP. This is therefore

the step size along the steepest descent vector that should be used. In the case where *G* is nonlinear, then iteration is required to converge to the location of the MPP.

The HL iteration method offers equivalent or better accuracy than the MVFOSM method, indeed, the first iteration is equivalent to the MVFOSM method. It can, however, converge to the MPP for nonlinear problems. It is not guaranteed to converge for all problems, and it is known to diverge or oscillate between two incorrect solutions for certain problems. It has also been shown that equivalent problems which are simply transformed can get different results^[36]. Additionally, it assumes the random variables are normally distributed.

3.3.4 Approximating Non-normal Distributions with Equivalent Normal Distributions

The reliability index, β , while pervasive in reliability analysis, is only defined for normally distributed random variables. Since many of the most commonly used algorithms for reliability analysis try to find β , non-normally distributed random variables often cannot be used. This restriction is eased by normal approximations of nonnormally distributed random variables. This is commonly performed using a method called the normal tail approximation [38]. As outlined by Choi, et al [36], using this method has the following advantages:

- 1. It does not require complex integration of the PDF.
- When using this in the HL-RF algorithm (see next section), transformation of nonnormal variables into their approximate normal versions takes place before solving for the location of the MPP.

42

3. It often yields accurate results compared to the exact solution.

The transformation takes the following form:

$$U_{i} = \frac{x_{i} - \mu_{x_{i}'}}{\sigma_{x_{i}'}}$$
(3.10)

where

$$\sigma_{x_{i}'} = \frac{\phi\left(\Phi^{-1}\left[F_{x_{i}}(x_{i}^{*})\right]\right)}{f_{x_{i}}(x_{i}^{*})}$$

$$\mu_{x_{i}'} = x_{i}^{*} - \Phi\left[F_{x_{i}}(x_{i}^{*})\right]\sigma_{x_{i}'}$$
(3.11)

where $F_{xi}(x_i)$ is the CDF of the distribution to be approximated and $f_{xi}(x_i)$ is the PDF of the distribution to be approximated.

This method of normal tail approximation effectively generates a normal distribution that matches the value of its PDF and CDF to the original non-normal distribution at the point x_i^* . This is often referred to as the Rackwitz and Fiessler (RF) algorithm, after the authors of a paper implementing this algorithm for reliability analysis of non-normally distributed random variables [39].



Figure 3.2. An illustration of how the RF algorithm approximates a non-normal distribution with a normal one.^[36]

This method has a tendency to introduce considerable error into the RA, as shown by Huang and Griffiths [17] and Qin et al [40].

3.3.5 First-Order Reliability Method

The First-Order Reliability Method (FORM) is the combination of HL iterations with the RF transformation. It is also called the HL-RF algorithm, for that reason. FORM is one of the most commonly used methods for RA when accurate computational models exist. It is relatively accurate for many problems, and is much more efficient than both MCS and LHS when the number of random variables is small. Its main cost is in function evaluations when analytical gradients are not available. In these cases, it requires n+1 evaluations per iteration, where n is the number of random variables.

3.3.6 Second-Order Reliability Method

The Second-Order Reliability Method (SORM)^[41] is the natural extension of FORM using a second-order Taylor-series expansion about the design point. In this case, the SORM algorithm calculates reliability index of a quadratic *g* exactly, and can capture the curvature of highly nonlinear limit state functions better than FORM. It has the downside of requiring second-order derivatives to be calculated at each design point. For implicit functions, the finite-difference calculation requires 2n+1 function evaluations, versus n+1 for only first-order derivatives, where n is the number of variables.

3.4 Modern Methods

In recent years several advanced methods for performing RA have been developed. Two of the more commonly used methods are summarized here.

3.4.1 Performance Measure Approach

MPP-based approaches for RA described earlier are also called the Reliability Index Approach (RIA) to differentiate them from a method named the Performance Measure Approach (PMA). Tu and Choi [42] first published this idea, which takes the optimization problem statement used in the RIA and inverts it. Recall equation (**3.6**), which is the RIA problem statement, and is given again in equation (**3.12**).

45

minimize:
$$\beta(U_i)$$
 (3.12)
subject to: $g(U_i)=0$

This problem statement is instead inverted, as shown in equation (3.13).

minimize:	$g(U_i)$	
		(3.13)
subject to:	$\ \boldsymbol{U}\ = \boldsymbol{\beta}_{\boldsymbol{t}}$	

where β_t is the target reliability.

Tu and Choi first recommended using the Advanced Mean Value algorithm (AMV)^[43], and later introduced the Conjugate Mean Value algorithm (CMV). These two algorithms perform better than RIA on many problems. In particular, they efficiently find good solutions to problems where the RIA diverges or oscillates between solutions. Later, Choi and Youn [21] proposed combining them into the Hybrid Mean Value (HMV) algorithm, which determines which of the two to use based on a measure of the convexity of the constraint limit surface.

While the PMA approach has been shown to be more robust than the RIA approach, it still suffers from many of the same potential pitfalls. It assumes a single failure region, that the random variables are normally distributed, and can diverge on some problems.

3.4.2 Sequential Optimization and Reliability Assessment

The Sequential Optimization and Reliability Assessment (SORA)^[44] algorithm is a method for performing RBDO (and not strictly a reliability analysis method) which makes use of the fact that finding a reliable design is the same thing as finding a design with a factor of safety (FOS) applied to the constraint. This was suspected by many in the field, and given a sufficient condition for proof by Ching [3]. The challenge is to determine a relationship between the two, since any such relationship would be problem dependent. The SORA algorithm solves the RBDO problem by iteratively solving the DDO problem. First, a FOS is assumed for the constraint, and DDO is performed with that FOS applied. Then RA is performed on the best design found to determine how far off the FOS estimate was from the desired P_f limit. The FOS is then adjusted using a linear approximation of the relationship, and then DDO is performed again with the new FOS. This process is repeated until the chosen FOS matches the desired Pf. It is likely to be more efficient than performing the full double-loop RBDO, but the many evaluations from multiple DDO and RA runs makes this a very expensive strategy still. It does have the benefit of good accuracy, assuming the iterative process converges, which may not happen for all problems.

4 The Proposed Approach

4.1 Goal

There is an important class of optimization problems in which the time to evaluate a single design is large enough that accounting for the reliability of the system being designed would be computationally infeasible. Additionally, for problems of this class, the responses are usually implicit functions of the design variables.

The goal of this research is to develop a strategy that allows reliability-based design optimization (RBDO) to be performed on problems of that class with an acceptable level of accuracy and in significantly less time than required by standard RBDO techniques. These problems are typically the design, with reliability in mind, of systems where the analysis response(s) may be nonlinear, and the deterministic design optimization (DDO) problem may also be nonlinear. Often these systems are analyzed using numerical techniques such as finite element analysis (FEA) or computational fluid dynamics (CFD), where solution times for each design evaluation can range from several minutes to several days. It is also typical for these problems to have relatively low input variability. This means that typically the responses will have low variability as well.

The level of accuracy that is deemed "acceptable" is not easy to quantify. The computational model used will have some error, and this strategy cannot overcome an inaccurate model. The level of acceptable accuracy must be problem dependent, and is that which is accurate enough to allow the development of reliable designs.

48

4.2 Approaches to Solving the RBDO Problem

While there are many different strategies to solve the RBDO problem, presented below are two extremes in terms of accuracy of P_f prediction and overall efficiency.

1. One way to solve the RBDO problem is to perform a RA for each evaluation of the nominal values of the design. This was described earlier as the double-loop problem. Typically this means running many additional evaluations that are just as computationally costly as the standard evaluation, but which have design variable values that vary about the nominal design. Regardless of the method of RA chosen, the increase in computational cost is significant.

If an MPP-based algorithm like FORM is used then the number of additional evaluations for each nominal evaluation is related to the number of variables and the nonlinearity of the constraint limit function. As the number of variables increases the number of evaluations per FORM iteration goes up linearly. FORM also requires a greater number of iterations for constraints with a high degree of nonlinearity. Furthermore, in many situations where the constraints are nonlinear, FORM has been shown to converge to incorrect solutions, or not converge at all [36].

If instead a sampling approach like MCS or LHS is used, then the accuracy can be much better, but the computational cost is typically much larger, unless the number of variables is so large that the gradient calculations in FORM require more evaluations than the sampling approaches.

Since this method generates no response surfaces for use in RA, there is no error in P_f prediction from a response surface. The accuracy of the reliability prediction is wholly dependent on the fidelity of the computational model and the suitability of the RA method chosen.

2. An alternative approach to RBDO is to evaluate the nominal designs (all deterministic responses in the problem statement) using actual evaluations, but perform RA on a response surface. This response surface is generated from some sampling performed before the optimization begins, typically by Latin Hypercube designs (in the Design of Experiments (DOE) sense, not the RA method). As a result, there can be less overall cost, since the RA cost consists primarily of the additional evaluations performed before the optimization begins. Evaluations on a response surface are typically very efficient compared to actual evaluations. However, this approach may result in a considerable loss in accuracy of the reliability responses if the response surface is not accurate.

One obvious criticism is why, if the response surface is accurate enough to perform RA, is it not accurate enough to perform the entire optimization? Performing optimization on a response surface has known issues, a few of which are listed below:

- Response surfaces always suffer from some amount of error, and the error at any point is unknown until the predicted value is compared against the value from an actual evaluation.
- In order to fit a responses surface, many points must be evaluated.
 Typically an approach like Latin Hypercube designs is used to determine where to sample in the design space. This results in many points being evaluated that are in low performing regions of the design space.
- c. Relative to the overall design space, the variation in the high performing designs that the analyst is seeking will be small. This means that evaluations during RA will only take place in a small neighborhood about the nominal design. Therefore, for a given design the accuracy of the response surface used to calculate reliability is only important locally around the nominal design. Fitting a single RS at the beginning of the optimization will give a global RS that hopefully has relatively low aggregate error over the entire space. However, this does not mean that it will be accurate locally for a given design. In fact, the error in the RS usually varies considerably from location to location around the design space.

Both of these approaches are valid ways to solve the RBDO problem, but they present their own challenges. For some problems, one or both methods may not be well-suited. To further complicate the issue, whether a problem can be solved by the

51

latter approach or not often cannot be determined *a priori*, and the analyst may waste time with an attempt and get no solution.

4.3 Intermediate Conclusions about RBDO

Based on an analysis of the two most common approaches to RBDO, a few conclusions can be drawn:

- For many problems one cannot perform RA using actual evaluations. It is simply too computationally expensive. This is also borne out in the literature.^{[14][23]}
- 2. Therefore, if RBDO is to be performed on problems with very expensive function evaluations, some form of RS must be used to calculate reliability.
- The RS need not be accurate over the entire design space for all designs.
 Instead, it only needs to be locally accurate about the nominal design for which the reliability must be calculated.
- 4. Since the design variables usually have low variability relative to the range of the design space, the RBDO solution will be close to the DDO solution in most cases. Typically the DDO solution will lie on or very close to the active constraint limit(s), and the RBDO solution will lie along the line parallel to the gradient of the objective function a small distance away from the limit(s).

These conclusions form the basis for the proposed RBDO strategy described in this thesis. Similar conclusions regarding response surfaces were found by Jin et al [45]. As will be described further, the proposed RBDO strategy uses response surfaces which are designed to be accurate locally around the nominal design being evaluated. Additionally, a first stage of only DDO is performed to identify high performing regions of
the design space. A later stage where RBDO is performed is seeded with useful data from the DDO run.

4.4 Attributes of the Proposed RBDO Method

- The strategy proposed here is independent of reliability analysis (RA) method or algorithm. Some methods or algorithms may work better or worse with this strategy, as will be noted later.
- This strategy is independent of system analysis model. The accuracy of the solution can be no better than the accuracy of the system analysis model, however.
- 3. This strategy does not require any particular optimization algorithm. However, the algorithm should be a multi-point global search method that tends to cluster new evaluations near previous designs that are high performing.

4.5 RBDO Strategy

Based on the assumptions and the goal outlined above, the following aspects will be included in the new RBDO strategy:

- 1. A local response surface will be used to calculate reliability.
- The amount of computational effort used to evaluate low performing designs (both in the DDO and RBDO sense) will be minimized.
- 3. The strategy will endeavor to not need to evaluate additional design points strictly

for the purpose of creating a response surface, since it is not known until after RA is performed if the design is good in the RBDO sense.

The optimization is performed in three stages:

- 1. The first stage is DDO only. During this stage only deterministic responses are calculated, and for many problems it is expected that the optimization will identify some high quality designs that are near one or more constraint limits, meaning one or more constraints are active. In some applications, the algorithm may even identify the DDO optimum, and if so, will also likely identify many high quality designs near it. The data from this run (the design variable values and the corresponding constraint values) will be retained for use in later steps.
- 2. The second stage is an intermediate step to identify which designs identified by the DDO solution may also be reliable, and should therefore be explored in the third stage. During this stage the best design found in stage one and several feasible (in the DDO sense) designs near it are selected. Then, each design is reevaluated in an RBDO sense, meaning their respective reliability values are calculated using an RS. Nominal response values are already known, so they do not need to be calculated again. RA is performed using the steps illustrated in Figure 4.1 and Figure 4.2.

If an RS is accepted (because it is sufficiently accurate), then RA is performed for this design and a performance value is assigned based on the RBDO problem statement. If an RS cannot be found that meets the RS error criteria, the design

54

is marked as an error. The top performing designs are prepared for injection into the RBDO optimization study in the third stage.



Figure 4.1. Process for performing reliability analysis on a design. The "Fit RS" process is described in Figure 4.2.



Figure 4.2. Process for fitting a response surface.

3. The third stage is a full RBDO, where the reliability of each (deterministically) feasible design is evaluated using a response surface built in a similar fashion as in stage two. That is, for each feasible design a RS is built using previous design evaluation points in the neighborhood of that design. This results in a RS that is accurate locally, which is sufficient for RA about that design. Good design points from Stage two are injected into the run at the beginning to give the search a "head start", based on the information obtained by the DDO study. As new designs are evaluated, their deterministic response values are saved for potential use in building future response surfaces.

In the strategy as outlined above, DDO is performed first for two primary reasons:

- To locate an optimal or near optimal DDO design. This is expected to be nearby the RBDO optimum. This design and others around it are used to seed the RBDO stage with designs near the RBDO optimum. Especially when the design space is multi-modal or otherwise complex, the initial DDO study is an efficient way to focus the eventual RBDO study on the most relevant region of the space.
- As a result of identifying the DDO optimum design, many designs nearby (and elsewhere in the design space) will be evaluated and are available to be used to fit a local RS in stages two and three.

4.6 Details on Strategy Implementation

This strategy was implemented in the following way:

- 1. A Genetic Algorithm (GA) was used as the optimization method. GAs are commonly used and meet the necessary criteria outlined earlier.
- 2. During the DDO stage, designs that were the same as earlier designs were not reevaluated, since there would be no benefit to doing so. However, during the RBDO stage, it is possible that some designs that were evaluated early in the run might have their reliability predicted poorly as a result of not enough local data to fit a quality RS. Therefore, those designs that reoccurred during the RBDO run were reevaluated, with the hope that they would have more data points nearby and thus benefit from a better approximation of *P_f*.
- 3. During stage two, the RS was accepted when the R^2 was greater than 0.85.
- During stage three, the RS was accepted when the R² was greater than 0.9.
 Additionally, the R² RS error was added as a constraint that must be above a 0.95. This encouraged designs to have more accurate response surfaces.
- 5. Polynomial response surfaces of order one or two were used. All interaction terms up to the order of the response surface were included.
- During stage two, a larger RS error was allowed for RS acceptance than in stage three.
- 7. In all cases the response surfaces are fit using the least-squares [46] method.

4.7 When to Switch from DDO to RBDO: α

One key aspect of this strategy was when the switch from DDO to RBDO should occur. The parameters E and α are defined, where E is the total number of cycles to be

performed, and α is the percentage of completed cycles before RBDO will be performed. When a GA is used as the optimizer, the number of cycles is usually referred to as the number of generations. Once the DDO has completed α E cycles, then stage two begins, as outlined above. Then the RBDO study begins and runs for (1- α)E cycles. This is illustrated in Figure 4.3.



Figure 4.3. Illustration of the parameters α and E and how they define the time when the switch from DDO to RBDO occurs.

4.8 Desired Outcomes

The following should be true of the proposed strategy to deem it successful:

- 1. It will be more efficient or more effective than performing RBDO using traditional methods for the following reasons.
 - (a) Actual evaluations are not used to perform the RA.
 - (b) The response surface is calculated using existing data. Additional evaluations

are not performed as part of the response surface calculation.

- (c) RA is not performed early in the optimization, since early designs are typically lower performing than later designs. Designs that are low performing in the deterministic sense are likely to be low performing even after accounting for reliability. Additionally, even performing RA on a response surface incurs some computational expense. This expense is removed entirely during stage 1 of the optimization.
- 2. The reliability calculation will be accurate enough to yield a useful final design. A "useful design" is one which is significantly more reliable than the DDO solution.

4.9 Implementation

The problems above were solved using the commercial optimization software package, HEEDS MDO v6.1^[47]. HEEDS MDO was setup to handle the process automation and optimization. The calculation of needed response surfaces was developed in MATLAB R2011a^[48]. The probability of failure calculations were performed through a custom C program. Reliability was evaluated using Monte Carlo Simulation with 1,000,000 evaluations.

The selected optimization algorithm used was a *genetic algorithm (GA)*. The HEEDS MDO User's Manual describes a genetic algorithm in this way:

"A *genetic algorithm* is a stochastic multi-point search method, which is based on the evolutionary theory of the survival of the fittest. It searches a space of potential solutions (designs) driven by the information it has learned so far during its search. The search starts with the evaluation of a set of random designs (i.e., values of all designs are chosen randomly within their allowable ranges). Once a set of candidate designs has been evaluated, a subset of those designs is picked, based in some way on their relative quality (i.e., degree to which they satisfy the problem's constraints and objectives). Good designs are retained while bad designs are gradually eliminated as better design candidates are found.

During the run, new designs are created by either mutation or crossover. Mutation changes the value of one or more variables, based on previous results. Crossover takes a pair of designs and generates a new design by swapping variable values between the

61

selected pair. The creation and evaluation of each new population of designs is called a generation."^[49]

The parameters used are as follows in Table 4.1.

Table 4.1. Genetic algorithm parameters used.

Genetic Algorithm	Value		
Parameter			
Population Size	20		
Number of evolge	Varied during the		
Number of cycles	studies (E)		
Generations per cycle	1		
Crossover type	Two point		
Mutation type	Multi-field		
Selection type	Tournament		
Crossover rate	0.5		
Mutation rate	0.2		
Selection parameter	2		

4.9.1 Performance-Based Optimization Problem Statement

HEEDS MDO uses a performance-based optimization problem statement for single objective problems.^[49] This means that all of the objectives and constraints are

combined into a single value, which defines the relative fitness of each design. The performance can be calculated using the formula below:

Performance

$$= \sum_{i=1}^{N_{obj}} \left(\frac{LinWt_i \cdot Sign_i \cdot Obj_i}{Norm_i} \right)$$

$$- \sum_{j=1}^{N_{con}} \left(\frac{QuadWt_j \cdot ConstraintViolation_j^2}{Norm_j^2} \right)$$
(4.1)

Table 4.2. Explanation of variables used in equation (4.1).

Variable	Definition	Value Used
N _{obj}	Number of objectives in the optimization study.	1
LinWt _i	The linear weight for the i th objective.	1
Sign _i	The sign for the i th objective. The value is -1 for objectives being minimized and +1 for objectives being maximized.	problem dependent
Obji	The response value for the i th objective for a given design.	n/a
Norm _i	The normalizing value for the i th objective.	problem dependent
N _{con}	The number of constraints in the optimization study	problem dependent
ConstraintViolation _j	The amount by which the j th constraint is violated.	n/a
Norm _j	The normalizing value for the j th constraint.	problem dependent
QuadWtj	The quadratic weight for the j th constraint.	10,000

4.9.2 Resolution

HEEDS MDO discretizes continuous variables using the variable *resolution*. Each design variable is assigned a value that identifies how many values the design variable can take on in the range defined. Unless otherwise specified, all problems were solved using a resolution of 1,000,001. This means that there were 1,000,001 equally spaced increments between (and including) each variable's minimum and maximum values. The intent of this resolution value is to ensure that the variables are treated approximately like continuous variables. For typical engineering problems, finding the best values of design variables down to small fractions of a percent is not useful. For the problems in this study, the selected resolution level should not have an impact on the optimization until well beyond this threshold.

5 Problem Descriptions and Results

Several problems were investigated using the strategy outlined above. The problems and results are described below. For each problem solved, first DDO was performed. Next, an intermediate stage identifies potential high-performing and reliable designs. Some of these designs are injected into RBDO, which follows. During RBDO, a response surface is fit for each feasible design and RA performed to determine the design's reliability.

5.1 Simple Polynomial Functions

The first problem solved consists of two simple polynomial functions.

5.1.1 Problem Definition

This problem has two variables, denoted X_1 , and X_2 . The objective function, F, is a linear polynomial function. The single constraint, G, is a quadratic polynomial function. These functions are defined and plotted below. This problem was developed specifically to be of minimal complexity to show that the proposed strategy works on the simplest of problems. The coefficients for F and G are selected arbitrarily, with the coefficients of G adjusted to give the G = 0 contour varying curvature.

$$F = -1.74 + 4.5X_1 - 3.49X_2 \tag{5.1}$$

$$G = 1.9032 + 9.1X_1 - 1.2416X_2 + 0.87X_1X_2 - 1.28X_1^2 - 0.9673X_2^2$$
 (5.2)

The DDO optimization problem statement is:

minimize	$F(X_1, X_2)$				
subject to	G(X ₁ , X ₂)	≥ 0			
by varying	1.0 ≤	<i>X</i> ₁	≤	7.0	(5.3)
	1.0 ≤	X2	≤	7.0	

The corresponding RBDO problem statement is:

minimize	F(X ₁	, X ₂)				
subject to	P[(G	(X ₁ , X	2) < 0] :	≤ 0.002	27	
by varying	1.0	≤	<i>X</i> ₁	≤	7.0	(5.4)
	1.0	≤	<i>X</i> ₂	≤	7.0	
	X ₁ ,	<2~ No	$r(\mu, \sigma^2)$)		

 X_1 and X_2 have standard deviations of 4% of their nominal (mean) values. The reliability constraint limit corresponds to approximately 3 σ reliability.



Figure 5.1. Contours of F and the constraint limit G = 0 over [X₁, X₂]. The green dot represents the DDO optimum design, and the blue dot represents the RBDO optimum design.

Recall that this problem is a set of simple polynomial functions. There are two variables, X_1 and X_2 , a linear objective, F, and a quadratic constraint, G. In this problem G is active, meaning that it drives the solution and is the sole preventer of further gains in the objective, F. Since G is active, then the DDO optimum point is found on the constraint limit. Since G is close to linear locally around this solution, the probability of failure is approximately 50%. Near this solution, the contours of F are tangent to G, so the RBDO optimum point is along a vector normal to F and G at the DDO optimum. The RBDO optimum point is a sufficient distance from the DDO optimum point along this vector to meet the reliability constraint of $P_f \le 0.0027$.

This problem was solved for six different total numbers of cycles: 8, 12, 16, 32, 60, and 100. It was also solved for three values of α : 0.25, 0.5, and 0.75. To make this clear, the number of cycles performed in each stage of the study is summarized in Table 5.1.

Total	Number of cycles performed in each stage						
Number of	α = (α = 0.25		α = 0.5		α = 0.75	
Cycles	DDO	RBDO	DDO	RBDO	DDO	RBDO	
8	2	6	4	4	6	2	
12	3	9	6	6	9	3	
16	4	12	8	8	12	4	
32	8	24	16	16	24	8	
60	15	45	30	30	45	15	
100	25	75	50	50	75	25	

Table 5.1. Breakdown of the number of cycles performed in each stage of the studies performed.

Each of the 18 optimization runs was replicated 25 times with different random seeds to account for the variations inherent in stochastic methods like genetic algorithms. All results presented below are the average of these 25 replications, unless otherwise stated.

The reliability of each design was evaluated using MCS with 1,000,000 evaluations. This was implemented in a custom C program.





Figure 5.2. Value of F the simple functions for the best design found over the number of cycles and values of α studied. This data is the average of 25 runs. The magnitude is the percent difference from the actual best design. P_f was calculated using a linear response surface based on available solution points.



Figure 5.3. Same data as presented in Figure 5.2, rearranged to have alpha on the X-axis.

Note in Figure 5.2 that within 32 cycles the solution has nearly converged. However, as shown in Figure 5.4, the error in the P_f calculation on the response surface takes longer to converge even after the objective has converged. Regarding the error calculation, the percent difference is not shown because of two issues:

- Sometimes the actual *P_f* is equal to zero, resulting in an undefined percent difference.
- 2. Sometimes the actual P_f is very small. Since no response surface can be perfect (unless the response to be fit is actually a polynomial function of the same order), there will always be some error. For two designs with different P_f , one similar to the constraint limit for this problem and one very small, but which have response

surfaces of similar quality, the latter case will have much larger error in the calculation of P_{f} . In these situations, the percent difference does not accurately reflect the error.



Figure 5.4. Absolute difference of P_f predicted by the linear response surface and the actual P_f for the best design found. The height of the colored bar is the average of 25 runs. The whiskers represent the range of the actual values. Note that in most cases the average is very close to the minimum, and typically only a single run had very high error. The "actual Pf" value compared against is the probability of failure as calculated using the same RA method, but with actual evaluations instead of a response surface.





From the results it is clear that the error in calculation of P_f is higher for $\alpha = 25\%$ than for $\alpha = 75\%$ across most of the values for number of cycles. This is likely due to having fewer local points near the optimal solution when the value of α is low. Higher values of α would yield more local points to choose from in calculating the response surface, resulting in a more accurate RS (at least locally, where the accuracy is needed).

One interesting result not clear from the plots is that the value of P_f for the best design tended to be different as more cycles were allowed. For early numbers of cycles the best design tended to be highly reliable, that is, P_f very close or equal to zero. As more cycles were allowed, the reliability of the optimal solution moved closer to its

constraint limit. So for low numbers of cycles the error in P_f is quite low because it tends be in a region that is flat ($P_f = 0$ everywhere which is feasible and not near the constraint limit). As the number of cycles increases, there are more designs available to use to fit the response surface, resulting in higher performing but lower reliability designs.

In the figures below, the history plots of five separate runs are overlaid. The runs were selected arbitrarily to show a representative sampling. The blue dots are evaluations performed during stage one, DDO. The red dots are evaluations performed during stage two, RBDO. The lines represent the best design at that point in the optimization search. Note that typically the best design at the beginning of the RBDO stage is significantly better than the best design at the beginning of the DDO stage. This is due to the beneficial influence of designs injected at the beginning of the RBDO stage. Also note that the best designs at the end of the RBDO stage have G values that are pushed away from the constraint limit a small amount, in order to satisfy the reliability constraint. Only plots for E = 32 and 100 are shown, for brevity.



Figure 5.6. History plot for the simple function problem, objective F from 5 runs, α =0.25, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.7. History plot for the simple function problem, constraint G from 5 runs, α =0.25, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.8. History plot for the simple function problem, objective F from 5 runs, α =0.25, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.9. History plot for the simple function problem, constraint G from 5 runs, α =0.25, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.10. History plot for the simple function problem, objective F from 5 runs, α =0.50, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.11. History plot for the simple function problem, constraint G from 5 runs, α =0.50, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.12. History plot for the simple function problem, objective F from 5 runs, α =0.50, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.13. History plot for the simple function problem, constraint G from 5 runs, α =0.50, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.14. History plot for the simple function problem, objective F from 5 runs, α =0.75, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.15. History plot for the simple function problem, constraint G from 5 runs, α =0.75, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.16. History plot for the simple function problem, objective F from 5 runs, α =0.75, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.17. History plot for the simple function problem, constraint G from 5 runs, α =0.75, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage.



5.1.3 Solution Using a Quadratic Response Surface

Figure 5.18. Value of F for the simple functions for best design found over the number of cycles and values of α studied. This data is the average of 25 runs. The magnitude is the percent difference the actual best design. P_f was calculated using a quadratic response surface based on available solution points.



Figure 5.19. Same data as presented in Figure 5.18, rearranged to have alpha on the X-axis.

The problem with linear F and quadratic G was solved again with a quadratic response surface fit to the constraint G. The process for choosing points and fitting a response surface was the same as with the linear response surface. For these runs, the error in the P_f calculation was almost always zero, as expected, with a few cases in which the absolute error was 10⁻⁶ due to round-off error in the least squares fit. Effectively, these results are equivalent to the case in which the study is performed with actual evaluations for both the nominal and stochastic performance of the designs. It is considered here in order to verify the accuracy of the implementation and the validity of the approach in cases of high accuracy in the P_f calculations.

An important conclusion from this set of results is that convergence is faster for α = 25% than α = 75%, which is the reverse of the results using the linear response surface. This makes sense, however, since in the linear RS results the trends were due to the accuracy of the RS generated. For this set of results, the RS was always exact (aside from the round-off error). So, for low α , more cycles are spent performing RBDO (with exact values of *Pf*) than for high α . Because the ultimate goal is to find the RBDO solution and not the DDO solution, it makes sense that a lower value of α would result in faster convergence of the solution. In fact, were it computationally feasible to perform RA using actual evaluations of G (which is essentially equivalent to fitting a quadratic response surface to quadratic data, as performed here), then the value of α used should be zero to find the RBDO solution most efficiently.

When comparing Figure 5.2 and Figure 5.18, it should be noted that the rate of convergence is faster for the quadratic RS. It is not surprising that a less accurate prediction of P_f would result in slower convergence.

5.2 Peaks

5.2.1 Problem Definition

This problem has two variables, denoted X₁, and X₂. The objective function, F, is a variation on the Peaks problem.^[48] This function is continuous and continuously differentiable, and has several stationary points. The single constraint, G, is a quadratic polynomial function. These functions are defined and plotted below. The resulting DDO and RBDO problems are multimodal. The coefficients of G are selected arbitrarily, with the coefficients adjusted to give the G = 0 contour varying curvature and make the unconstrained optimum point infeasible. This problem was developed to show that the proposed strategy could solve multimodal problems.

$$F = -\frac{1}{3}exp\left(-(1-X_{1})^{2}-X^{2}\right)$$

+ $3exp\left(-X_{1}^{2}-(1+X_{2})^{2}\right)\left(1-X_{1}\right)^{2}$ (5.5)
 $-10exp\left(-X_{1}^{2}-X_{2}^{2}\right)\left(\frac{X_{1}}{5}-X_{1}^{3}-X_{2}^{5}\right)$

$$\begin{split} G &= -1.4265 + 0.22334X_1 - 0.62499X_2 + 0.23294X_1^2 \\ &\quad + 0.24135X_1X_2 + 0.20016X_2^2 \end{split} \tag{5.6}$$

The DDO optimization problem statement is:

minimize	F(X ₁ , X ₂)				
subject to	G(X ₁ , X ₂):	≤ 0			
by varying	<i>-3.0</i> ≤	<i>X</i> ₁	≤	3.0	(5.7)
	<i>-3.0</i> ≤	X2	≤	3.0	

The corresponding RBDO problem statement is:

minimize	F(X ₁ , X	(₂)				
subject to	P[(G(X	(₁ , X ₂)	> 0] ≤	0.005		
by varying	-3.0	≤	X ₁	≤	3.0	(5.8)
	-3.0	≤	X ₂	≤	3.0	
	X ₁ , X ₂	~ Nor	(μ, σ ²)			

 X_1 and X_2 have standard deviations of 0.1. The reliability constraint limit corresponds to approximately 2.5 σ reliability.



Figure 5.20. Contours of F and the constraint limit G = 0 over [X₁, X₂]. The green dot represents the DDO optimum design, and the red dot represents the RBDO optimum design.

The DDO optimum design has the following approximate values:

F	=	-6.46949
G	=	-6.11652*10 ⁻⁶
X ₁	=	0.244728
X_2	=	-1.549854

The RBDO optimum design has the following approximate values:

F	=	-5.02620
G	=	-0.290869
P[G > 0]	=	0.004999
X ₁	=	0. 315982
X ₂	=	-1.29081

This problem was solved for six total numbers of cycles: 8, 12, 16, 32, 60, and 100. It was also solved for three values of α : 0.25, 0.5, and 0.75 (these are the same parameter values as used in the previous example). Table 5.1 summarizes the list of runs performed.

Each optimization run was replicated 25 times with different random seeds to account for the variations inherent in stochastic methods like genetic algorithms. All results presented below are the average of these 25 replications, unless otherwise stated. The reliability of each design was evaluated using MCS with 1,000,000 evaluations. This was implemented in a custom C program.

5.2.2 Solution Using a Linear Response Surface

This problem was first solved using a linear response surface to locally approximate the constraint.



Figure 5.21. Value of F for the peaks problem for best design found over the number of cycles and values of α studied. This data is the average of 25 runs. The magnitude is the percent difference from the best design. P_f was calculated using a linear response surface based on available solution points.


Figure 5.22. Same data as presented in Figure 5.21, rearranged to have alpha on the X-axis.

For the low values of E, the value of α = 75% was more efficient than 25% and 50%. As the number of cycles increases this no longer holds, and α = 75% is the least efficient of the values of α studied. This reversal may be due to this problem being both multimodal and having an RBDO solution which is a significant distance away from the DDO solution. So even if the DDO solution finds the global DDO optimum, the RBDO solution is still a significant distance away, and so for this type of problem the proposed strategy may not accelerate the RBDO search as much as for other problems. There is still benefit from the DDO solution providing points for use in fitting response surfaces. The history plots shown later bear this out, where the starting best design for the RBDO phase is less often (but not always) similar in objective value to the starting best design from DDO.



Figure 5.23. Absolute difference of P_f predicted by the response surface and the actual P_f for the best design found. The height of the colored bar is the average of 25 runs. The whiskers represent the range of the actual values. Note that in most cases the average is very close to the minimum, and typically only a single run had very high error. The "actual P_f " value compared against is the probability of failure as calculated using the same RA method, but with actual evaluations instead of a response surface.





The P_f calculation error is reasonable, and trends downward between 60 and 100 cycles. As seen in previous problems, often there is a single run which has high error, which pushes the average upward.

In the figures below, the history plots of five runs are overlaid. The runs were selected arbitrarily to show a representative sampling. The blue dots are evaluations performed during stage one, DDO. The red dots are evaluations performed during stage two, RBDO. The lines represent the best design at that point in the optimization. In the previous study, the starting best design for the RBDO stage was typically much better than the starting best for the DDO stage. This was due to the DDO stage identifying high quality, reliable designs to seed the RBDO stage with. For this problem, since the

DDO solution found may not be the global well, or if it is, is still far from the RBDO solution, this seeding process did not always yield a significantly better solution to start off the RBDO stage with. This distance from the DDO solution to the RBDO solution can also be seen in the large cap between the best RBDO solutions found and the constraint limit line.

Also note the large number of designs clustered around the value of zero for the objective F. These designs are actually not necessarily near each other in design variable value, and are a result of the problem having a large area near the corners and edges where it is relatively flat and has a value of zero.



Figure 5.25. History plot for the peaks problem, objective F from 5 runs, α =0.25, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.26. History plot for the peaks problem, constraint G from 5 runs, α =0.25, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.27. History plot for the peaks problem, objective F from 5 runs, α =0.25, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.28. History plot for the peaks problem, constraint G from 5 runs, α =0.25, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.29. History plot for the peaks problem, objective F from 5 runs, α =0.50, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.30. History plot for the peaks problem, constraint G from 5 runs, α =0.50, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.31. History plot for the peaks problem, objective F from 5 runs, α =0.50, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.32. History plot for the peaks problem, constraint G from 5 runs, α =0.50, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.33. History plot for the peaks problem, objective F from 5 runs, α =0.75, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.34. History plot for the peaks problem, constraint G from 5 runs, α =0.75, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.35. History plot for the peaks problem, objective F from 5 runs, α =0.75, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.36. History plot for the peaks problem, constraint G from 5 runs, α =0.75, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage.

Since this is a 2D problem with multiple local optima it is useful to plot the location of the best design found by the RBDO stage.



Figure 5.37. Contours of objective F overlaid with the constraint limit (G = 0, dotted line) for the peaks problem. The green dot is the DDO optimum and the red dot is the RBDO optimum. The other plotted points are the best design found for each of the 25 runs performed. Only a subset of the data is plotted: E = 12 in cyan, E = 60 in white, and E = 100 in black. The results plotted here are for α = 0.25.



Figure 5.38. Contours of objective F overlaid with the constraint limit (G = 0, dotted line) for the peaks problem. The green dot is the DDO optimum and the red dot is the RBDO optimum. The other plotted points are the best design found for each of the 25 runs performed. Only a subset of the data is plotted: E = 12 in cyan, E = 60 in white, and E = 100 in black. The results plotted here are for α = 0.50.



Figure 5.39. Contours of objective F overlaid with the constraint limit (G = 0, dotted line) for the peaks problem. The green dot is the DDO optimum and the red dot is the RBDO optimum. The other plotted points are the best design found for each of the 25 runs performed. Only a subset of the data is plotted: E = 12 in cyan, E = 60 in white, and E = 100 in black. The results plotted here are for α = 0.75.

From the plots showing the location of the best design it can be noted that in some situations the best design found is not near the global optimum. This is not surprising, given that the feasible region near the global optimum which is also lower than the local well is rather narrow.

5.2.3 Solution Using a Quadratic Response Surface

Below are the results of using a quadratic response surface to locally approximate the constraint. Recall that the constraint in this problem is a quadratic function, so this section's results are effectively exact in terms of P_f calculation. The measured error was similar to the round-off error which occurs in the least-squares fit, so effectively nonexistent.



Figure 5.40. Value of F for the peaks problem for the best design found over the number of cycles and values of α studied. This data is the average of 25 runs. The magnitude is the percent difference from the best design. Pf was calculated using a quadratic response surface based on available solution points.



Figure 5.41. Same data as presented in Figure 5.40, rearranged to have alpha on the X-axis.

For these results, it does not appear that higher values of α are of any benefit. For low numbers of cycles the efficiency is similar for all values of α , while it is significantly better to have low α for higher numbers of cycles. These results are similar to those from the Simple Function, and the same conclusion holds. When the P_f calculation is exact or near exact, while also being computationally inexpensive, it makes sense to spend as much time as possible performing RBDO. So the best value of α in this situation is near zero.

In these results the low numbers of cycles find better designs than with the linear RS, but later there appears to be little difference in convergence rate. This may be because the accuracy of the linear response surface was "good enough" for the optimization to perform well once there are enough evaluations to fit a good local RS.

In the figures below, the history plots of five runs are overlaid. The runs were selected arbitrarily to show a representative sampling. The blue dots are evaluations performed during stage one, DDO. The red dots are evaluations performed during stage two, RBDO. The lines represent the best design at that point in the optimization. In the previous study, the starting best design for the RBDO stage was typically much better than the starting best for the DDO stage. This was due to the DDO stage identifying high quality, reliable designs to seed the RBDO stage with. For this problem, since the DDO solution found may not be the global well, or if it is, is still far from the RBDO solution to start off the RBDO stage with. This distance from the DDO solution to the RBDO solution can also be seen in the large cap between the best RBDO solutions found and the constraint limit line.

105

Also note the large number of designs clustered around the value of zero for the objective F. These designs are actually not necessarily near each other in design variable value, and are a result of the problem having a large area near the corners and edges where it is relatively flat and has a value of zero.



Figure 5.42. History plot for the peaks problem, objective F from 5 runs, α =0.25, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.43. History plot for the peaks problem, constraint G from 5 runs, α =0.25, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.44. History plot for the peaks problem, objective F from 5 runs, α =0.25, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.45. History plot for the peaks problem, constraint G from 5 runs, α =0.25, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.46. History plot for the peaks problem, objective F from 5 runs, α =0.50, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.47. History plot for the peaks problem, constraint G from 5 runs, α =0.50, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.48. History plot for the peaks problem, objective F from 5 runs, α =0.50, E=100 The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.49. History plot for the peaks problem, constraint G from 5 runs, α =0.50, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.50. History plot for the peaks problem, objective F from 5 runs, α =0.75, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.51. History plot for the peaks problem, constraint G from 5 runs, α =0.75, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.52. History plot for the peaks problem, objective F from 5 runs, α =0.75, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.53. History plot for the peaks problem, constraint G from 5 runs, α =0.75, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage.

Since this is a 2D problem with multiple local optima it is useful to plot the location of the best design found by the RBDO stage.



Figure 5.54. Contours of objective F overlaid with the constraint limit (G = 0, dotted line) for the peaks problem. The green dot is the DDO optimum and the red dot is the RBDO optimum. The other plotted points are the best design found for each of the 25 runs performed. Only a subset of the data is plotted: E = 12 in cyan, E = 60 in white, and E = 100 in black. The results plotted here are for α = 0.25.



Figure 5.55. Contours of objective F overlaid with the constraint limit (G = 0, dotted line) for the peaks problem. The green dot is the DDO optimum and the red dot is the RBDO optimum. The other plotted points are the best design found for each of the 25 runs performed. Only a subset of the data is plotted: E = 12 in cyan, E = 60 in white, and E = 100 in black. The results plotted here are for α = 0.50.



Figure 5.56. Contours of objective F overlaid with the constraint limit (G = 0, dotted line) for the peaks problem. The green dot is the DDO optimum and the red dot is the RBDO optimum. The other plotted points are the best design found for each of the 25 runs performed. Only a subset of the data is plotted: E = 12 in cyan, E = 60 in white, and E = 100 in black. The results plotted here are for α = 0.75.

From the plots showing the location of the best design it can be noted that in some situations the best design found is not near the global optimum. This is not surprising, given that the feasible region near the global optimum which is also lower than the local well is rather narrow.

5.3 Cantilevered I-Beam

5.3.1 Problem Definition

This problem is a cantilevered beam of length L with a tip load, P. It has an Ishaped cross-section, defined by four parameters, b1, h1, b2, and H, shown in the figure below. The beam is made of aluminum.



Figure 5.57. Analysis and geometric details of the cantilevered I-beam.

The DDO problem statement is described below:

minimize	Volume					
subject to	MaxStress ≤ 5000 psi					
	$MaxDisp \le 0.065$ in.					
by varying	2.0 in	≤	b1	≤	12.0 in.	(5.9)
	0.1 in.	≤	h1	≤	1.0 in.	
	0.1 in	≤	b2	≤	2.0 in.	
	3.0 in.	≤	Н	≤	7.0 in.	

The RBDO problem statement is described below:

minimize	Volume					
subject to	P[MaxStress > 5000 psi] ≤ 0.0027					
	<i>P[MaxDisp > 0.065 in.]</i> ≤ 0.0027					
by varying	2.0 in	≤	b1	≤	12.0 in.	(5.40)
	0.1 in.	≤	h1	≤	1.0 in.	(5.10)
	0.1 in	≤	b2	≤	2.0 in.	
	3.0 in.	≤	Н	≤	7.0 in.	
	b1, h1, b2, Η ~ Nor(μ, σ²)					

The responses are defined by the formulae below:

$$Volume = (2 \cdot h1 \cdot b1 + (H - 2 \cdot h1)b2)L$$

$$MaxStress = \frac{P \cdot L \cdot H}{2 \cdot I}$$

$$MaxDisp = \frac{P \cdot L^{3}}{3 \cdot E \cdot I}$$

$$I = \frac{1}{12}b2(H - 2 \cdot h1)^{3} + 2\left[\frac{1}{12}b1 \cdot h1^{3} + \frac{b1 \cdot h1(H - h1)^{2}}{4}\right]$$
(5.11)

For this problem, the following values were used: Young's Modulus = 10 Msi, L = 36 in., and P = 1000 lb_f. The baseline design is given in Table 8.2, along with the assumed standard deviation. All design variables are normally distributed and have constant standard deviations of 3% of their baseline values. Table 5.2. Baseline design and applied variation for the cantilevered beam problem.

Variable	Baseline	Standard		
Variable	Value	Deviation		
b1	4.0	0.12		
h1	0.1	0.003		
b2	0.25	0.0075		
Н	3.5	0.105		

The DDO optimum design has the following approximate values:

Volume	=	92.76 in ³ .
MaxStress	=	5000. psi
MaxDisp	=	0.06172 in.
b1	=	9.48387 in.
h1	=	0.1 in.
b2	=	0.1 in.
н	=	7.0 in.

The RBDO optimum design has the following approximate values:

Volume	=	98.81 in ³ .
MaxStress	=	4,643 psi
P[MaxStress > 5000]	=	0.01233
MaxDisp	=	0.05731 in.
P[MaxDisp > 0.065 in.]	=	0.001757
b1	=	9.510 in.

h1	=	0.1086 in
b2	=	0.1000 in.
Н	=	7.000 in.

This problem was solved using the same set of values for α and E as in the simple function problem, listed in Table 5.1, except for the linear response surface it was only run to E = 60. This problem was selected because it had more than two variables and more than one constraint, unlike the previous two example problems. Additionally, it is an engineering problem, which makes it similar to the types of problems this RBDO strategy is intended to solve.

Each optimization run was replicated 20 times with different random seeds to account for the variations inherent in stochastic methods like genetic algorithms. All results presented below are the average of these 20 replications, unless otherwise stated. The reliability of each design was evaluated using MCS with 1,000,000 evaluations. This was implemented in a custom C program.

5.3.2 Solution Using a Linear Response Surface

As in the previous problems, first the problem was solved using a linear RS. In this case separate RS were generated for each constraint.



Figure 5.58. Value of Volume for the cantilevered beam problem for best design found over the number of cycles and values of α studied. This data is the average of 20 runs. The magnitude is the percent difference from the best design.





This problem converged more slowly than the previous two problems, which is expected given the additional variables and constraint. It has not yet fully converged after 60 cycles.

For 8 cycles, $\alpha = 75\%$ is slightly more efficient than 25%. This holds for 12 cycles as well, though $\alpha = 50\%$ shows similar efficiency to 75%. For 32 cycles, the trend may be reversing so $\alpha = 25\%$ is slightly more efficient, though it is unknown if this is a true trend or some latent stochastic variation. If the trend is reversing, it would likely be due to the longer DDO stage producing enough designs to get RS which are accurate enough to avoid hampering the RBDO stage.



Figure 5.60. Absolute difference of Pf predicted by the response surface and the actual Pf for the best design found for the MaxStress response. The height of the colored bar is the average of 20 runs. The whiskers represent the range of the actual values. Note that in most cases the average is very close to the minimum, because typically only a single run had very high error. The "actual Pf" value compared against is the probability of failure as calculated using the same RA method, but with actual evaluations instead of a response surface.



Figure 5.61. Same data as presented in Figure 5.60, without min and max bars (only average value is plotted). A horizontal line is provided showing the reliability constraint limit value.



Figure 5.62. Absolute difference of Pf predicted by the response surface and the actual Pf for the best design found for the MaxDisp response. The height of the colored bar is the average of 20 runs. The whiskers represent the range of the actual values. Note that in most cases the average is very close to the minimum, because typically only a single run had very high error. The "actual Pf" value compared against is the probability of failure as calculated using the same RA method, but with actual evaluations instead of a response surface.




The trends in the error in P_f with respect to α are not clear, meaning there may be some significant stochastic error remaining in the data. Regardless, it is clear that the error goes down as the number of cycles increases. The error in P_f for MaxStress tends to be lower than that of MaxDisp. The average bars are clearly thrown off by the small number of replications which had high error. One indicator of this is that the median P_f Error for MaxStress was 0 for cycles 8, 12, and 16. For the P_f error for MaxDisp, the median averaged 0.000161 for cycles 8, 12, and 16, and rose to similar levels as MaxStress for the higher number of cycles. So it can be stated that typically the P_f error is relatively low, except isolated runs where the error can be much larger. This behavior is representative of one risk of using an automatically generated (no human intervention) RS.

In the figures below, the history plots of five runs are overlaid. The runs selected were arbitrary to show a representative sampling. The blue dots are evaluations performed during stage one, DDO. The red dots are evaluations performed during stage two, RBDO. The lines represent the best design at that point in the optimization. Note that typically the best design at the beginning of the RBDO stage is significantly better than the best design at the beginning of the DDO stage. This is due to the beneficial influence of designs injected to the RBDO stage. Also note that the best designs at the end of the RBDO stage have constraint values that are pushed away from the constraint limit a small amount.



Figure 5.64. History plot for the cantilevered beam problem, objective Volume from 5 runs, α =0.25, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.65. History plot for the cantilevered beam problem, constraint MaxStress from 5 runs, α =0.25, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.66. History plot for the cantilevered beam problem, constraint MaxDisp from 5 runs, α =0.25, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.67. History plot for the cantilevered beam problem, objective Volume from 5 runs, α =0.25, E=60. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.68. History plot for the cantilevered beam problem, constraint MaxStress from 5 runs, α =0.25, E=60. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.69. History plot for the cantilevered beam problem, constraint MaxDisp from 5 runs, α =0.25, E=60. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.70. History plot for the cantilevered beam problem, objective Volume from 5 runs, α =0.50, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.71. History plot for the cantilevered beam problem, constraint MaxStress from 5 runs, α =0.50, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.72. History plot for the cantilevered beam problem, constraint MaxDisp from 5 runs, α =0.50, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.73. History plot for the cantilevered beam problem, objective Volume from 5 runs, α =0.50, E=60. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.74. History plot for the cantilevered beam problem, constraint MaxStress from 5 runs, α =0.50, E=60. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.75. History plot for the cantilevered beam problem, constraint MaxDisp from 5 runs, α =0.50, E=60. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.76. History plot for the cantilevered beam problem, objective Volume from 5 runs, α =0.75, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.77. History plot for the cantilevered beam problem, constraint MaxStress from 5 runs, α =0.75, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.78. History plot for the cantilevered beam problem, constraint MaxDisp from 5 runs, α =0.75, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.79. History plot for the cantilevered beam problem, objective Volume from 5 runs, α =0.75, E=60. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.80. History plot for the cantilevered beam problem, constraint MaxStress from 5 runs, α =0.75, E=60. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.81. History plot for the cantilevered beam problem, constraint MaxDisp from 5 runs, α =0.75, E=60. The blue dots are the DDO stage, while the red dots are the RBDO stage.

5.3.3 Solution Using a Quadratic Response Surface

Below are the results of the cantilever beam study using a quadratic response surface to approximate the constraints.



Figure 5.82. Value of Volume for best design found over the number of cycles and values of α studied. This data is the average of 20 runs. The magnitude is the percent difference from the best design.



Figure 5.83. Same data as presented in Figure 5.82, rearranged to have alpha on the X-axis.



Figure 5.84. Absolute difference of Pf predicted by the response surface and the actual Pf for the best design found for the MaxStress response. The height of the colored bar is the average of 20 runs. The whiskers represent the range of the actual values. Note that in most cases the average is very close to the minimum, because typically only a single run had very high error. The "actual Pf" value compared against is the probability of failure as calculated using the same RA method, but with actual evaluations instead of a response surface.



Figure 5.85. Same data as presented in Figure 5.84, without min and max bars (only average value is plotted). A horizontal line is provided showing the reliability constraint limit value.



Figure 5.86. Absolute difference of Pf predicted by the response surface and the actual Pf for the best design found for the MaxDisp response. The height of the colored bar is the average of 20 runs. The whiskers represent the range of the actual values. Note that in most cases the average is very close to the minimum, because typically only a single run had very high error. The "actual Pf" value compared against is the probability of failure as calculated using the same RA method, but with actual evaluations instead of a response surface.



Figure 5.87. Same data as presented in Figure 5.86, without min and max bars (only average value is plotted). A horizontal line is provided showing the reliability constraint limit value.

When comparing Figure 5.58 and Figure 5.82 one difference between running with linear RS versus quadratic RS is that the quadratic actually performs worse at low numbers of evaluations. This is particularly evident at α = 0.25, where the quadratic RS run found significantly worse designs than the linear RS run. This is opposite of the results from the previous two problems. This is due to this problem having four variables, while the others had two, as well as the quadratic RS being exact for the last two problems. For this problem, a linear RS requires five points, while a quadratic surface requires fifteen points. For low numbers of evaluations, the run using a quadratic RS was not able to fit RS with acceptable accuracy until later in the run (when there are more points available). As the number of cycles increases, the runs using

linear and quadratic response surfaces have comparable performance. This indicates that there are enough local points in the larger number of cycle runs to get a local quadratic surface of sufficient accuracy, so the optimization is not hampered like it was with fewer cycles.

The trends in α seem to suggest that early on it is better to do more DDO (high α) and as the error in P_f decreases this reverses and it is better to do more RBDO (low α). The trends in the error in P_f with regard to α are not clear, meaning there may be some significant stochastic error remaining in that data. One obvious result is that the error is significantly lower for these results than those using the linear RS. Also, the error in P_f for MaxStress tends to be lower than that of MaxDisp.

In the figures below, the history plots of five runs are overlaid. The runs were arbitrarily selected to show a representative sampling. The blue dots are evaluations performed during stage one, DDO. The red dots are evaluations performed during stage two, RBDO. The lines represent the best design at that point in the optimization. Note that typically the best design at the beginning of the RBDO stage is significantly better than the best design at the beginning of the DDO stage. This is due to the beneficial influence of designs injected to the RBDO stage. Also note that the best designs at the end of the RBDO stage have constraint values that are pushed away from the constraint limit a small amount.

It's also clear that for α = 0.25, and 0.5 the DDO solution has not converged for E = 32. This is evident from the gap between the best design found during the DDO stage

142

and the constraint limits for the two constraints. For E = 100, the DDO solution is against the constraint limits, indicating that it may have found the DDO optimum.



Figure 5.88. History plot for the cantilevered beam problem, objective Volume from 5 runs, α =0.25, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.89. History plot for the cantilevered beam problem, constraint MaxStress from 5 runs, α =0.25, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.90. History plot for the cantilevered beam problem, constraint MaxDisp from 5 runs, α =0.25, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.91. History plot for the cantilevered beam problem, objective Volume from 5 runs, α =0.25, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.92. History plot for the cantilevered beam problem, constraint MaxStress from 5 runs, α =0.25, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.93. History plot for the cantilevered beam problem, constraint MaxDisp from 5 runs, α =0.25, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.94. History plot for the cantilevered beam problem, objective Volume from 5 runs, α =0.50, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.95. History plot for the cantilevered beam problem, constraint MaxStress from 5 runs, α =0.50, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.96. History plot for the cantilevered beam problem, constraint MaxDisp from 5 runs, α =0.50, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.97. History plot for the cantilevered beam problem, objective Volume from 5 runs, α =0.50, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.98. History plot for the cantilevered beam problem, constraint MaxStress from 5 runs, α =0.50, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.99. History plot for the cantilevered beam problem, constraint MaxDisp from 5 runs, α =0.50, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.100. History plot for the cantilevered beam problem, objective Volume from 5 runs, α =0.75, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.101. History plot for the cantilevered beam problem, constraint MaxStress from 5 runs, α =0.75, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.102. History plot for the cantilevered beam problem, constraint MaxDisp from 5 runs, α =0.75, E=32. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.103. History plot for the cantilevered beam problem, objective Volume from 5 runs, α =0.75, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.104. History plot for the cantilevered beam problem, constraint MaxStress from 5 runs, α =0.75, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage.



Figure 5.105. History plot for the cantilevered beam problem, constraint MaxDisp from 5 runs, α =0.75, E=100. The blue dots are the DDO stage, while the red dots are the RBDO stage.

5.4 Torque Arm

5.4.1 Problem Definition

This problem involves the optimization of a 2D planar torque arm. The torque arm has a cutout in the center, the shape of which is controlled by four variables: *x1*, *x2*, *r1*, and *r2*. These and the other dimensions of the torque arm are shown in Figure 5.106. The torque arm has the following properties: Young's Modulus = 207.4 GPa, v = 0.3, $\rho = 7,810 \text{ kg/m}^3$, $\sigma_Y = 800 \text{ MPa}$. The thickness of the torque arm is 3 mm. The left-hand circle is pinned, meaning all displacement degrees of freedom (DOF) are constrained. The right-hand circle has loading applied, as shown in Figure 5.106.



Figure 5.106. Geometric and loading details of the torque arm. The left-hand circle has pin boundary conditions applied (all displacement DOF constrained).

The DDO problem statement is described below:

minimize	Mass					
subject to	MaxStress	MaxStress ≤ 800 MPa				
by varying	90 mm	≤	x1	≤	140 mm	(5 40)
	40 mm	≤	x2	≤	210 mm	(5.12)
	5 mm	≤	r1	≤	40 mm	
	5 mm	≤	r2	≤	35 mm	

The RBDO problem statement is described below:

minimize	Mass					
subject to	<i>P[MaxStress > 800 MPa] ≤ 0.01</i>					
by varying	90 mm	≤	x1	≤	140 mm	
	40 mm	≤	x2	≤	210 mm	(5.13)
	5 mm	≤	r1	≤	40 mm	
	5 mm	≤	r2	≤	35 mm	
	x1, x2, r1, r2 ~ Nor(μ , σ^2)					

The baseline design is given in Table 5.3, with the applied standard deviation. All design variables were normally distributed and had constant standard deviations of 2.5% of their baseline values.

Variabla	Baseline	Standard		
variable	Value	Deviation		
x1	120	3.0		
x2	150	3.75		
r1	10	0.25		
r2	10	0.25		

Table 5.3. Baseline design and applied variation.

This problem was selected to be representative of a "real-world" RBDO problem. It was modeled in the commercial FEA code Abaqus v6.11-1^[50]. The CAD and FE preprocessing was performed in Abaqus/CAE. A linear static step was used, and the model was solved using Abaqus/Standard. CPS4 (linear plane stress continuum elements) with a mesh seed of 4 mm were used. Each design was modified through parametric CAD, and then remeshed. A new input deck was written for each new design. This entire process was automated through a python script, which was modified by HEEDS MDO to reflect that design's design variable values. The FE solver (Abaqus) read the input deck and calculated the solution. Finally, a python script was executed that reads the FE output file and writes a set of text files containing the pertinent responses for HEEDS MDO to read.

Since the analysis of this problem uses finite elements and each design is remeshed, there is some added noise to this problem. This increases the difficulty of solving this problem, since the constraint will be less smooth.

Since each evaluation of this problem takes much longer than for other problems, determining the best solution by performing MCS with 1,000,000 evaluations was computationally infeasible. Instead, at the end of each optimization run, a LHS RA was performed with 500 evaluations. This is the nominal value from which the error in P_f was calculated. As stated earlier, the variance in LHS is smaller than MCS. The 95% confidence interval for 500 evaluations to predict a P_f of approximately 0.01 is 0.004355 to 0.02327, if MCS was used. It can be stated that the CI for LHS is significantly tighter, but the reduction in variance is problem dependent and can typically only be determined by empirical means.

The DDO optimum design has the following approximate values:

Mass = 0.6879 kg MaxStress = 798.5 MPa

155

x1	=	91.5 mm
x2	=	210. mm
r1	=	30.2 mm
r2	=	15.2 mm

These values were determined by running this problem several times for up to 2000 evaluations.

Due to the time necessary to run this problem, no long running (full double-loop) RBDO studies were performed using actual evaluations. Instead, the best design found in any run performed is listed below:

Mass	=	0.7131 kg
MaxStress	=	776.1 MPa
P[MaxStress > 8	= [000	0.168
x1	=	90.00 mm
x2	=	205.5 mm
r1	=	30.18 mm
r2	=	11.66 mm

This problem was solved using a subset of the values of E as in the simple function problem. The new set of parametric studies is summarized in Table 5.4.

Total	Number of cycles performed in each stage					
Number of	α = 0.25		α = 0.5		α = 0.75	
Cycles (E)	DDO	RBDO	DDO	RBDO	DDO	RBDO
12	3	9	6	6	9	3
32	8	24	16	16	24	8
60	15	45	30	30	45	15
100	25	75	50	50	75	25

Table 5.4. Breakdown of the number of cycles performed in each stage of the studies performed.

Due to the required runtime on this problem, the results below have no replications, only a single run was performed at each set of parametric values.

5.4.2 Solution Using a Linear Response Surface

The plots below show the results of using a linear response surface to approximate the constraint for RA purposes. As noted above, the actual best design for the RBDO problem is not known due to the time required to run such a problem. So, the results below that refer to "best design" are using the best design found out of any of the runs performed.



Figure 5.107. Value of Mass for best design found over the number of cycles and values of α studied. The magnitude is the percent difference from the best design found over all studies. The value for α = 0.75, E = 60 is the best design found, so it has a value of zero on this graph.



Figure 5.108. Same data as presented in Figure 5.107, rearranged to have alpha on the X-axis.



Figure 5.109. Absolute difference of Pf predicted by the response surface and the actual Pf for the best design found for the MaxStress response. The dark line at 0.01 is the constraint limit.

5.4.3 Solution Using a Quadratic Response Surface

The plots below show the results of using a quadratic response surface to approximate the constraint for RA purposes. As noted above, the actual best design for the RBDO problem is not known due to the time required to run such a problem. So, the results below that refer to "best design" are using the best design found out of any of the runs performed.


Figure 5.110. Value of Mass for best design found over the number of cycles and values of α studied. The magnitude is the percent difference from the best design found over all studies.



Figure 5.111. Same data as presented in Figure 5.110, rearranged to have alpha on the X-axis.





The results of this study are difficult to interpret due to the lack of replications and sources of noise. One conclusion is that 100 cycles is likely not enough to converge on this problem. Additionally, the quadratic RS clearly performs better with regard to P_f prediction than the linear RS. The error is still rather large, so for this problem a different sort of response surface may give better results, such as a radial basis function or Kriging surface.

5.4.4 Subset Parametric Study with Replication – Linear RS

This problem was solved again with a different set of values. This was performed using $\alpha = 0.75$ and E = 60, 120. The additional evaluations from E = 120 are based on the previous study not being converged at 100 cycles. For this set of studies, 5 replications were performed. The results shown are the average of these replications, unless otherwise stated.



Figure 5.113. Value of Mass for best design found over the number of cycles and values of α studied. The magnitude is the percent difference from the best design found over all runs performed.



Figure 5.114. Absolute difference of P_f predicted by the response surface and the actual P_f for the best design found for the MaxStress response. The dark line at 0.01 is the constraint limit. The whiskers represent the max and min error values, while the bar is the average value.





For this study, the problem showed slow convergence. Regarding the error, it increased between 60 and 120 cycles, but the variation decreased. This difference in the mean may not be statistically significant.

5.4.5 Subset Parametric Study with Replication – Quadratic RS

This problem was solved again with a different set of values. This was performed using $\alpha = 0.75$ and E = 60, 120. The additional evaluations from E = 120 are based on the previous study not being converged at 100 cycles. For this set of studies, 5

replications were performed. The results shown are the average of these replications, unless otherwise stated.



Figure 5.116. Value of Mass for best design found over the number of cycles and values of α studied. The magnitude is the percent difference from the best design.





Again, this problem showed slow convergence. The error in P_f calculation was higher for 120 cycles versus 60 again, though this may be due to noise. The quadratic RS had lower error than the linear RS.

6 Summary and Conclusions

6.1 Summary and Conclusions

In this study, a new approach for reliability based design optimization was proposed, implemented and demonstrated. In this approach the following concepts play a key motivating role:

- Global response surfaces, especially least-squares, do not always have high accuracy for complex functions. However, these response surfaces can be accurate locally.
- RBDO is too computationally expensive to perform if the full double-loop method is employed on problems common to engineering today.
- 3. The RBDO optimum design is very often close to the DDO optimum.

This new approach has three stages. In stage one, only DDO is performed. This is to identify high-performing regions of the design space, and to provide data for fitting response surfaces later. In stage two the best designs are evaluated against the RBDO problem statement, using response surfaces created for each design from the existing data. The best designs are injected into stage three, RBDO. There, each feasible design is evaluated in the same way as in stage two, except as more data is obtained in high-performing regions, the accuracy of P_f tends to increase.

Based on the studies performed, the conclusions from this work are:

1. The proposed strategy for performing RBDO appears to be much more efficient than the traditional "double-loop" method. This is due to the enormous cost

associated with the inner loop, where many evaluations would have to be performed for each individual outer loop evaluation.

2. The proposed strategy for performing RBDO is less accurate than the doubleloop method, but provides sufficient accuracy to push the design toward reliable alternatives to the DDO solution.

Regarding the value of α that should be used, this decision is problem dependent. The results presented earlier showed that if there are not enough local points to fit a RS in high performing regions (which results in those evaluations being marked as errors), the optimization will tend toward lower performing, but highly reliable solutions. As more local points become available in the high performing regions, the optimization tends to produce designs closer to the constraint limit. The trends in α with respect to the optimization convergence indicate this as well, showing that it is more efficient to use more DDO at low numbers of cycles and more RBDO at high numbers of cycles. The number of cycles where this trend inflects is comparable to the time when the P_f error begins to drop. So the conclusion to be drawn for when to switch from DDO to RBDO is when there is enough data for accurate local response surfaces to be created around the highest performing designs found. If this is not possible at a certain stage in the run, it is prudent to continue performing DDO. This will continue to identify deterministically good designs, and generate more data around them for use in fitting response surfaces later.

Since the value of α that should be used is dependent on both the problem being solved and the number of available evaluations, it is recommended that future implementations of this strategy endeavor to set α dynamically.

170

6.2 Lessons Learned

6.2.1 Sensitivity of FORM to RS Error as Compared to MCS

In earlier studies on the Simple Function problem the runs were performed with FORM as the RA method and compared to results using MCS. Since the RA was being performed on an RS, the use of FORM was not as important, since the evaluations were very fast. It was discovered that FORM was much more sensitive to error in the RS than MCS, that is, the error in the P_f calculation was much higher when FORM was used than when MCS was used. These were only measures of the error in RS, since the comparison of P_f predicted and P_f actual were always compared using the same RA method.

This sensitivity is likely due to differences in how the two RA methods work. The accuracy of FORM is a function of far fewer points than MCS, which is the source of its efficiency. Least-squares fits are designed to minimize the error of the resulting response surface over all of the fitted points. For RA, the only aspect of the constraint that needs to be accurate is the surface that defines the constraint limit. FORM attempts to locate the nearest point on the constraint limit (the MPP), which may be poorly represented by the RS. Using the same response surface, MCS will sample more broadly, and thus would not be as sensitive to high error right at the MPP. Both of these methods will predict P_f more accurately when using RS fitting methods that favor points near the constraint limit, as mentioned in the literature review.

171

6.2.2 Issues Selecting Points for Fitting Local RS

The process for fitting a local RS, as summarized by Figure 4.2, may seem rather complex. Initially the strategy was to just select the points that were closest to the nominal design and use those directly. One issue that arose was that commonly these points would have some alignment or lack of good spacing. This resulted in the RS fit failing due to a poorly conditioned matrix, from the lack of linear independence of the equations resulting from aligned points. So the strategy used was to find more local points than were needed to fit a surface, that is, expand the size of the neighborhood around the nominal design. Then check several randomly selected subsets of points for the minimum Euclidian distance between the points in the subset. The subset with the maximum minimum distance would be more likely to have more spaced out points, and have less alignment. This does not guarantee better spacing, but it does encourage it.

6.2.3 Resolution Matters

It was noted in early studies of the Simple Function problem that the design variable resolution used can have a noticeable effect on the results. See Figure 6.1 and Figure 6.2, below.



Figure 6.1. Contours of F and the constraint limit G = 0 over $[X_1, X_2]$. The green dot represents the DDO optimum design, and the blue dot represents the RBDO optimum design. This is a reprinting of Figure 5.1 for comparison. For reference, these results use a resolution of 1,000,001.





The RBDO solution (blue point) is offset from where one would expect it to be when lower resolution is used. The constraint limit surface is not aligned with the design variables near the DDO optimum, resulting in the wrong solution being found. This illustrates the case where the resolution is coarse enough to cause the solution to be shifted noticeably from that where truly continuous variables are used.

6.3 Future work

Below are the suggestions for future work.

- 1. Since the recommendation for future use of this strategy is to set α dynamically and determining the error in P_f would be very expensive, additional research into inexpensive methods for estimating the error should be studied.
- The use of surrogate models that can capture more complexity than linear or quadratic response surfaces should be examined. Examples include Kriging surfaces and radial basis functions.
- 3. The value of R² used to accept the RS during stages two and three was kept constant throughout these studies. Further studies could be performed with this as a parameter. Additionally, other measures of RS accuracy could be used and studied for their effect on efficiency and accuracy of the solution.
- 4. No attempt was made to increase the efficiency or accuracy of the solution by modifying GA parameters. The RBDO problem has some distinct differences from the corresponding DDO problem, and so a study of which parameter values work better for each problem could be interesting.
- 5. This strategy could be implemented with different ways of fitting the RS which take into account that RA will be performed. Usually this is done by somehow favoring points near the constraint limit or the MPP, depending on the RA method which will be used, to get better RS accuracy at that location.
- More sophisticated methods for selecting points used to fit an RS could be used.
 One option is to fit existing data into a Latin Hypercube and sometimes evaluate additional points to get a more accurate RS.

REFERENCES

REFERENCES

- [1] Wikipedia contributors. "Computer-aided design." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 21 Apr. 2011. Web. 1 May. 2011.
- [2] Wikipedia contributors. "Computer-aided engineering." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 21 Apr. 2011. Web. 1 May. 2011.
- [3] Ching, Jianye. "Equivalence Between Reliability and Factor of Safety." *Probabilistic Engineering Mechanics* 24.2 (2009): 159–171. Web. 24 May 2012.
- [4] Bucher, Christian, and Thomas Most. "A Comparison of Approximate Response Functions in Structural Reliability Analysis." *Probabilistic Engineering Mechanics* 23.2–3 (2008): 154–163.
- [5] Roux, W. J., Nielen Stander, and R. T. Haftka. "Response Surface Approximations for Structural Optimization." *International Journal for Numerical Methods in Engineering* 42.3 (1998): 517–534. Web. 30 Mar. 2011.
- [6] Taflanidis, Alexandros A., and Sai-Hung Cheung. "Stochastic Sampling Using Moving Least Squares Response Surface Approximations." *Probabilistic Engineering Mechanics* 28.0 (2012): 216–224. Web. 24 May 2012.
- [7] Simpson, Timothy W. et al. "Comparison of Response Surface and Kriging Models for Multidisciplinary Design Optimization." *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. St. Louis, MO, 1998. Print.
- [8] Simpson, T.W. et al. "Metamodels for Computer-based Engineering Design: Survey and Recommendations." *Engineering With Computers* 17.2 (2001): 129– 150. Web. 30 Mar. 2011.
- [9] Rajashekhar, Malur R., and Bruce R. Ellingwood. "A New Look at the Response Surface Approach for Reliability Analysis." *Structural Safety* 12.3 (1993): 205– 220. Web. 2 Oct. 2009.
- [10] Kaymaz, Irfan, and Chris A. McMahon. "A Response Surface Method Based on Weighted Regression for Structural Reliability Analysis." *Probabilistic Engineering Mechanics* 20.1 (2005): 11–17. Web. 19 Feb. 2010.
- [11] Gomes, Herbert Martins, and Armando Miguel Awruch. "Comparison of Response Surface and Neural Network with Other Methods for Structural Reliability Analysis." *Structural Safety* 26.1 (2004): 49–67. Web. 23 Feb. 2010.

- [12] Gayton, N., J. M. Bourinet, and M. Lemaire. "CQ2RS: a New Statistical Approach to the Response Surface Method for Reliability Analysis." *Structural Safety* 25.1 (2003): 99–121. Web. 2 Feb. 2010.
- [13] Allaix, D.L., and V.I. Carbone. "An Improvement of the Response Surface Method." *Structural Safety* 33.2 (2011): 165–172. Web. 23 May 2012.
- [14] Guan, X.L., and R.E. Melchers. "Effect of Response Surface Parameter Variation on Structural Reliability Estimates." *Structural Safety* 23.4 (2001): 429–444. Web. 5 May 2012.
- [15] Bucher, C.G., and U. Bourgund. "A Fast and Efficient Response Surface Approach for Structural Reliability Problems." *Structural Safety* 7.1 (1990): 57– 66. Web. 9 June 2012.
- [16] Valdebenito, M.A., H.J. Pradlwarter, and G.I. Schuëller. "The Role of the Design Point for Calculating Failure Probabilities in View of Dimensionality and Structural Nonlinearities." *Structural Safety* 32.2 (2010): 101–111. Web. 19 Feb. 2010.
- [17] Huang, Jinsong, and D.V. Griffiths. "Observations on FORM in a Simple Geomechanics Example." *Structural Safety* 33.1 (2011): 115–119. Web. 23 May 2012.
- [18] Liu, Pei-Ling, and Armen Der Kiureghian. "Optimization Algorithms for Structural Reliability." *Structural Safety* 9.3 (1991): 161–177. Print.
- [19] Koduru, S.D., and T. Haukaas. "Feasibility of FORM in Finite Element Reliability Analysis." *Structural Safety* 32.2 (2010): 145–153. Web. 19 Feb. 2010.
- [20] Pretorius, C.A., K.J. Craig, and L.J. Haarhoff. "Kriging Response Surfaces as an Alternative Implementation of RBDO in Continuous Casting Design Optimization." Collection of Technical Papers -10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, August 30, 2004 -September 1, 2004. Vol. 4. Albany, NY, United states: American Institute of Aeronautics and Astronautics Inc., 2004. 2410-2419. Print. Collection of Technical Papers - 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference.
- [21] Choi, K. K., and B. D. Youn. "Hybrid Analysis Method for Reliability-Based Design Optimization." Proceedings of DETC'01 ASME 2001 Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Pittsburgh, Pennsylvania: ASME, 2001. Print.
- [22] Xu, Lin, and Gengdong Cheng. "Discussion on: Moment Methods for Structural Reliability." *Structural Safety* 25.2 (2003): 193–199. Web. 3 Feb. 2009.

- [23] Guan, X.L., and R.E. Melchers. "Effect of Response Surface Parameter Variation on Structural Reliability Estimates." *Structural Safety* 23.4 (2001): 429–444. Web. 5 May 2012.
- [24] Yang, R.J., and L. Gu. "Experience with Approximate Reliability-based Optimization Methods." *Structural and Multidisciplinary Optimization* 26.1-2 (2004): 152–159. Print.
- [25] Adams, B.M., M.S. Eldred, and J.W. Wittwer. "Reliability-based Design Optimization for Shape Design of Compliant Micro-electro-mechanical Systems." 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Sep 6-8 2006. Vol. 2. Reston, VA 20191, United States: American Institute of Aeronautics and Astronautics Inc., 2006. 1042–1056. Print. Collection of Technical Papers - 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference.
- [26] <u>Minitab</u> release 15. State College, Pennsylvania: Minitab Inc., 2011.
- [27] Wikipedia contributors. "Expected Value." *Wikipedia, the free encyclopedia* 15 June 2012. Web. 16 June 2012.
- [28] Wikipedia contributors. "Normal distribution." *Wikipedia, The Free Encyclopedia.* Wikipedia, The Free Encyclopedia, 23 Apr. 2011. Web. 27 Apr. 2011.
- [29] Wikipedia contributors. "Log-normal distribution." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 3 Apr. 2011. Web. 27 Apr. 2011.
- [30] Metropolis, Nicholas, and S. Ulam. "The Monte Carlo Method." *Journal of the American Statistical Association* 44.247 (1949) : 335-341. Web. 28 Apr 2011.
- [31] Brown, Lawrence D. "Interval Estimation for a Binomial Proportion." *Statistical Science* 16.2 (2001): 101–133. Web. 26 July 2012.
- [32] Mckay, M. D., R. J. Beckman, and W. J. Conover. "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code." *Technometrics* 42.1 (2000) : 55-61. Web. 18 Aug 2009.
- [33] Stein, Michael. "Large Sample Properties of Simulations Using Latin Hypercube Sampling." *Technometrics* 29.2 (1987): 143–151. Web. 24 Apr. 2012.
- [34] Olsson, A., G. Sandberg, and O. Dahlblom. "On Latin Hypercube Sampling for Structural Reliability Analysis." *Structural Safety* 25.1 (2003): 47–68. Web. 20 Aug. 2009.
- [35] Hasofer, Abraham M., and Niels C. Lind. "Exact and Invariant Second-Moment Code Format." *ASCE Journal of the Engineering Mechanics Division* 100.1 (1974): 111-121. Print.

- [36] Choi, Seung-Kyum, R. V. Grandhi, and Robert A. Canfield. *Reliability-based Structural Design*. 2007.
- [37] Avriel, Mordecai. *Nonlinear Programming: Analysis and Methods*. Courier Dover Publications, 2003. Print.
- [38] Hohenbichler, Michael, and Rudiger Rackwitz. "Non-Normal Dependent Vectors in Structural Safety." *ASCE Journal of the Engineering Mechanics Division*, 107.6 (1981) : 1227-1238. Print.
- [39] Rackwitz, Ruediger, and Bernd Fiessler. "Structural Reliability Under Combined Random Load Sequences." *Computers and Structures* 9.5 (1978): 489-494. Print.
- [40] Qin, Quan et al. "Effects of Variable Transformations on Errors in FORM Results." *Reliability Engineering & System Safety* 91.1 (2006): 112–118. Web. 23 Feb. 2010.
- [41] Fiessler, Bernd, Hans-Joachim Neumann, and Rudiger Rackwitz. "Quadratic Limit States in Structural Reliability." *Journal of the Engineering Mechanics Division* 105.4 (1979): 661–676. Print.
- [42] Tu, J., K. K. Choi, and Y. H. Park. "A New Study on Reliability-Based Design Optimization." *Journal of Mechanical Design* 121.4 (1999): 557–564. Web. 13 Apr. 2009.
- [43] Wu, Y.-T., T. A. Cruse, and H. R. Millwater. "Advanced Probabilistic Structural Analysis Method for Implicit Performance Functions." *AIAA Journal* 28 (1990): 1663–1669. Print.
- [44] Du, Xiaoping, and Wei Chen. "Sequential Optimization and Reliability Assessment Method for Efficient Probabilistic Design." ASME Design Engineering Technical Conferences 126 (2002): 225–233. Print.
- [45] Jin, R., X. Du, and W. Chen. "The Use of Metamodeling Techniques for Optimization Under Uncertainty." *Structural and Multidisciplinary Optimization* 25.2 (2003): 99–116. Web. 29 Apr. 2012.
- [46] Wikipedia contributors. "Least squares." *Wikipedia, The Free Encyclopedia.* Wikipedia, The Free Encyclopedia, 23 Apr. 2012. Web. 29 Apr. 2012.
- [47] <u>HEEDS MDO</u> version 6.1. East Lansing, Michigan: Red Cedar Technology, 2012.
- [48] MATLAB release 2011a. Natick, Massachusetts: The MathWorks Inc., 2011.
- [49] Red Cedar Technology, Inc. HEEDS MDO 6.1 User's Manual. East Lansing: Red Cedar Technology, 2011. Electronic.

[50] Abaqus v6.11-1. Providence, Rhode Island: Dassault Systèmes Simulia Corp., 2011.